

Cyberspace_TechNode

虚心学习 专注 豁达

[博客园](#) [首页](#) [新随笔](#) [联系](#) [访问](#) [管理](#)

通过JDBC进行简单的增删改查（以MySQL为例）

目录

[前言：什么是JDBC](#)

[一、准备工作（一）：MySQL安装配置和基础学习](#)

[二、准备工作（二）：下载数据库对应的jar包并导入](#)

[三、JDBC基本操作](#)

[\(1\)定义记录的类（可选）](#)

[\(2\)连接的获取](#)

[\(3\)insert](#)

[\(4\)update](#)

[\(5\)select](#)

[\(6\)delete](#)

[四、测试](#)

[五、代码分析](#)

[六、思考问题](#)

前言：什么是JDBC

维基百科的简介：

公告

since July 10, 2013

昵称：五岳

园龄：10年6个月

粉丝：826

关注：15

[+加关注](#)



2021年6月						
日	一	二	三	四	五	六
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

搜索

105

0

[回到顶部](#)

Java 数据库连接，（Java Database Connectivity，简称JDBC）是Java语言中用来规范客户端程序如何来访问数据库的应用程序接口，提供了诸如查询和更新数据库中数据的方法。JDBC也是Sun Microsystems的商标。它JDBC是面向关系型数据库的。

简单地说，就是用于执行SQL语句的一类Java API，通过JDBC使得我们可以直接使用Java编程来对关系数据库进行操作。通过封装，可以使开发人员使用纯Java API完成SQL的执行。

一、准备工作（一）：MySQL安装配置和基础学习

使用JDBC操作数据库之前，首先你需要有一个数据库。这里提供了3个链接供读者自学，如果曾有过SQL语言的使用经历（包括在学校中的课堂学习），前两个链接足以上手。

- 1.安装和配置：[mysql安装图解](#) [mysql图文安装教程\(详细说明\)](#)
- 2.基本操作：[21分钟 MySQL 入门教程](#)
- 3.简易命令查询：[一千行MySQL学习笔记](#)

建议边看入门教程，边练习，在练习insert、update、select、delete等基本操作的同时，将后面要用的表建好。

下图是我接下来用于演示的数据库的表。

```
mysql> use samp_db
Database changed
mysql> select * from students;
+----+-----+-----+-----+
| id | name  | sex  | age |
+----+-----+-----+-----+
| 1  | Ender | male | 8   |
| 2  | Bean  | male | 7   |
| 3  | Petra | fema | 9   |
| 4  | Peter | male | 9   |
| 5  | _Graff | male | 40  |
| 6  | GOD   | fema | 255 |
+----+-----+-----+-----+
6 rows in set (0.00 sec)
```

二、准备工作（二）：下载数据库对应的jar包并导入

使用JDBC需要在工程中导入对应的jar包。数据库与JDBC包的对应关系可以参考[各种数据库对应的jar包、驱动类名和URL格式](#)。在Eclipse下的导入方法：

在工程的图标上右击，选择“ Properties” ，在“ Java Bulid Path” 中选择“ Add External JARs...” ，选择下载并解压后获得的jar包。

找找看

谷歌搜索

我的标签

算法(13)

Java(8)

面试(8)

操作系统(6)

编程之美(5)

现代操作系统(4)

nginx(4)

Linux(3)

爬坑(2)

细节(2)

更多

积分与排名

积分 - 287941

排名 - 2250

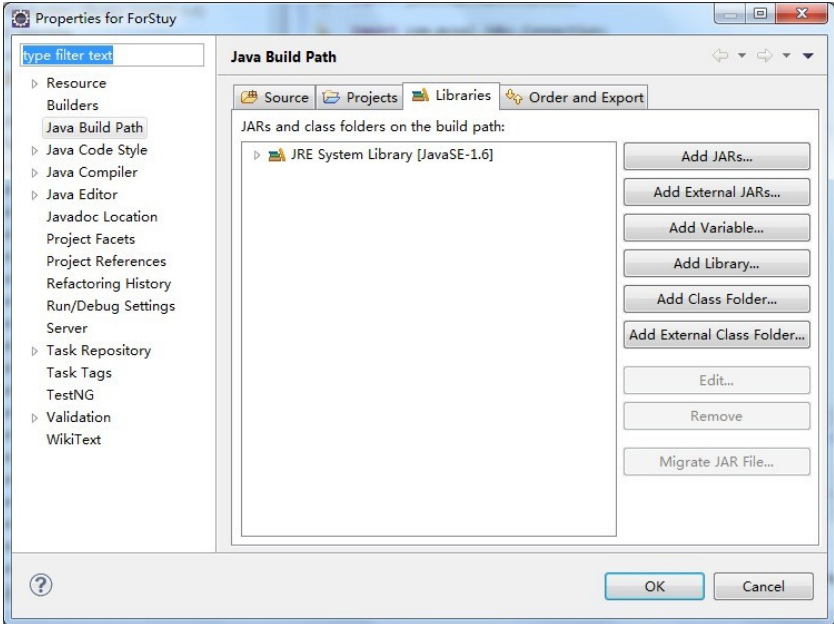
随笔分类 (316)

C(31)

C++(3)

DB(3)

Git(1)



如果对MySQL进行操作，这时下面的import就不会报错了：

```
import com.mysql.jdbc.Connection;
import com.mysql.jdbc.PreparedStatement;
```

除此以外，还需要JDBC的包，直接import即可。

```
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
```

三、JDBC基本操作

为了简单起见，与数据库相关的操作、命令、参数都被硬编码了。有兴趣的读者可以对这些进行探索，降低数据与操作的耦合性。

先看具体代码并实践，本文第五部分对用到的API稍作了研究。

下面的所有方法和数据成员都在public class JDBCOperation内部。

(1)定义记录的类（可选）

这样做主要是为了便于操作和接口定义，是非必须的。

```
static class Student {
    private String Id;
    private String Name;
    private String Sex;
    private String Age;

    Student(String Name, String Sex, String Age) {
        this.Id = null; //default
        this.Name = Name;
        this.Sex = Sex;
        this.Age = Age;
    }

    public String getId() {
        return Id;
    }
}
```

Java(23)
JVM(6)
Linux/Unix(31)
Linux内核(7)
nginx(4)
Python(3)
Simulink(6)
笔试面试题(14)
编程之美(5)
操作系统(10)
机器学习/神经网络(3)
旧坑归档(2)
面向对象(2)
嵌入式(2)
软件开发(15)
算法(28)
网络编程(20)
学习笔记(65)
珠玑之棧(7)
资料收集(25)

随笔档案 (142)	105	0
2021年4月(1)		

```
    }

    public void setId(String Id) {
        this.Id = Id;
    }

    public String getName() {
        return Name;
    }

    public void setName(String Name) {
        this.Name = Name;
    }

    public String getSex() {
        return Sex;
    }

    public void setSex(String Sex) {
        this.Sex = Sex;
    }

    public String getAge() {
        return Age;
    }

    public void setage(String Age) {
        this.Age = Age;
    }
}
```



(2)连接的获取

在操作前必须先获取与数据库的连接。

driver、url的格式同样可以参考[各种数据库对应的jar包、驱动类名和URL格式](#)。

```
private static Connection getConn() {
    String driver = "com.mysql.jdbc.Driver";
    String url = "jdbc:mysql://localhost:3306/samp_db";
    String username = "root";
    String password = "";
    Connection conn = null;
    try {
        Class.forName(driver); //classLoader,加载对应驱动
        conn = (Connection) DriverManager.getConnection(url, username,
password);
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return conn;
}
```



2021年2月(1)

2020年7月(1)

2020年6月(4)

2020年5月(3)

2020年3月(2)

2019年12月(1)

2019年11月(3)

2019年8月(1)

2019年7月(1)

2019年3月(2)

2018年12月(2)

2018年11月(3)

2018年10月(1)

2017年6月(1)

2015年11月(1)

2015年5月(1)

2015年4月(1)

2015年3月(2)

2015年2月(2)

2015年1月(1)

2014年7月(1)

2014年4月(1)

105

0

(3)insert

```
private static int insert(Student student) {
    Connection conn = getConn();
    int i = 0;
    String sql = "insert into students (Name,Sex,Age) values(?,?,?)";
    PreparedStatement pstmt;
    try {
        pstmt = (PreparedStatement) conn.prepareStatement(sql);
        pstmt.setString(1, student.getName());
        pstmt.setString(2, student.getSex());
        pstmt.setString(3, student.getAge());
        i = pstmt.executeUpdate();
        pstmt.close();
        conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return i;
}
```

(4)update

```
private static int update(Student student) {
    Connection conn = getConn();
    int i = 0;
    String sql = "update students set Age='" + student.getAge() + "'
where Name='" + student.getName() + "'";
    PreparedStatement pstmt;
    try {
        pstmt = (PreparedStatement) conn.prepareStatement(sql);
        i = pstmt.executeUpdate();
        System.out.println("result: " + i);
        pstmt.close();
        conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return i;
}
```

(5)select

以select * from XXX为例。

```
private static Integer getAll() {
    Connection conn = getConn();
    String sql = "select * from students";
    PreparedStatement pstmt;
    try {
        pstmt = (PreparedStatement)conn.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
    }
}
```

2014年2月(3)
2013年12月(2)
更多

相册 (31)
《大话设计模式》配图(24)
malloc配图(2)
VC维(3)
vi(1)
博客贴图(1)

常用资料
MATLAB中文论坛
在线进制转换
Linux Cross Reference
Linux在线手册
在线LaTex转换
在线C定义解释
在线Shell (1)
在线Shell (2)
Markdown语法
Markdown在线编辑器

友链	105	0
----	-----	---

```
int col = rs.getMetaData().getColumnCount();
System.out.println("=====");
while (rs.next()) {
    for (int i = 1; i <= col; i++) {
        System.out.print(rs.getString(i) + "\t");
        if ((i == 2) && (rs.getString(i).length() < 8)) {
            System.out.print("\t");
        }
    }
    System.out.println("");
}

System.out.println("=====");
} catch (SQLException e) {
    e.printStackTrace();
}
return null;
}
```



(6)delete



```
private static int delete(String name) {
    Connection conn = getConn();
    int i = 0;
    String sql = "delete from students where Name='" + name + "'";
    PreparedStatement pstmt;
    try {
        pstmt = (PreparedStatement) conn.prepareStatement(sql);
        i = pstmt.executeUpdate();
        System.out.println("resutl: " + i);
        pstmt.close();
        conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return i;
}
```



四、测试

在测试前，需要在系统中打开对应数据库的服务。MySQL在Windows下的启动命令为

```
net start mysql
```

测试代码



```
public static void main(String args[]) {
    JDBCOperation.getAll();
    JDBCOperation.insert(new Student("Achilles", "Male", "14"));
    JDBCOperation.getAll();
    JDBCOperation.update(new Student("Bean", "", "7"));
}
```

dariusdong

Shen Fan(范深)

最新评论

1. Re:轻松记住大端小端的含义（附对大端和小端的解释）

我试着用【好懂】和【好记】的方法来解释下。顺便提一嘴，我自己对技术名词的翻译很在乎，翻译的不好，会让学习者很难受，影响理解。big-endian、little-endian翻译成大端、小端，这个翻译...

--lison

2. Re:《大话设计模式》Python版代码实现（新增多年实践的心得）

纸上得来终觉浅...

--从零开始的程序员生活

3. Re:DDD实践反思

@Flaming、淡蓝 不过现在好多微服务本身里面用的就还是DDD...

--luacloud

阅读排行榜

1. 通过JDBC进行简单的增删改查（以MySQL为例）(560990)

2. 常见的动态规划问题分析与求解(179850)

3. 轻松记住大端小端的含义（附对大端和小端的解释）(83759)

4. 手把手教你编写一个具有基本功能的shell（已开源）(73913)

5. 全面解析回溯法：算法 105 习题 0 (674)

回到顶部

```
JDBCOperation.delete("Achilles");
JDBCOperation.getAll();
}
```



Eclipse中的输出



```
=====
1   Ender      male    8
2   Bean       male    6
3   Petra      fema    9
4   Peter      male    9
5   _Graff     male    40
6   GOD        fema    255
=====
=====
1   Ender      male    8
2   Bean       male    6
3   Petra      fema    9
4   Peter      male    9
5   _Graff     male    40
6   GOD        fema    255
7   Achilles   Male    14
=====
resutl: 1
resutl: 1
=====
1   Ender      male    8
2   Bean       male    7
3   Petra      fema    9
4   Peter      male    9
5   _Graff     male    40
6   GOD        fema    255
=====
```



五、代码分析

在上述对数据库进行增删改查的过程中，可以发现其共性部分，即通用的流程：

- (1)创建**Connection**对象、SQL查询命令字符串；
- (2)对**Connection**对象传入SQL查询命令，获得**PreparedStatement**对象；
- (3)对**PreparedStatement**对象执行**executeUpdate()**或**executeQurey()**获得结果；
- (4)先后关闭**PreparedStatement**对象和**Connection**对象。

可见，使用JDBC时，最常打交道的是**Connection**、**PreparedStatement**这两个类，以及select中的**ResultSet**类。查阅Java API手册可以了解其具体的意义和方法。

评论排行榜

1. 通过JDBC进行简单的增删改查（以MySQL为例）(24)
2. 手把手教你编写一个具有基本功能的shell（已开源）(23)
3. 如何写出正确的二分查找？——利用循环不变式理解二分查找及其变体的正确性以及构造方式(21)
4. DDD实践反思(16)
5. malloc()参数为0的情况(15)

推荐排行榜

1. 通过JDBC进行简单的增删改查（以MySQL为例）(105)
2. 常见的动态规划问题分析与求解(54)
3. 《大话设计模式》Python版代码实现（新增多年实践的心得）(31)
4. 全面解析回溯法：算法框架与问题求解(28)
5. 从《编程之美》买票找零问题说起，娓娓道来卡特兰数——兼爬坑指南(25)

下面引用的Java API的资料出自
<http://download.oracle.com/technetwork/java/javase/6/docs/zh/api/index.html>。

Connection

java.sql 接口 Connection

所有超级接口：

Wrapper

```
public interface Connection extends Wrapper
```

与特定数据库的连接（会话）。在连接上下文中执行 SQL 语句并返回结果。
Connection 对象的数据库能够提供描述其表、所支持的 SQL 语法、存储过程、此连接功能等等的信息。此信息是使用 `getMetaData` 方法获得的。

PreparedStatement

java.sql 接口 PreparedStatement

所有超级接口：

Statement, Wrapper

所有已知子接口：

CallableStatement

```
public interface PreparedStatement extends Statement
```

表示预编译的 SQL 语句的对象。

SQL 语句被预编译并存储在 PreparedStatement 对象中。然后可以使用此对象多次高效地执行该语句。

常用方法

```
boolean execute()
```

在此 PreparedStatement 对象中执行 SQL 语句，该语句可以是任何种类的 SQL 语句。

```
ResultSet executeQuery()
```

在此 PreparedStatement 对象中执行 SQL 查询，并返回该查询生成的 ResultSet 对象。

```
int executeUpdate()
```

在此 PreparedStatement 对象中执行 SQL 语句，该语句必须是一个 SQL 数据操作语言（Data Manipulation Language, DML）语句，比如 INSERT、UPDATE 或 DELETE 语句；或者是无返回内容的 SQL 语句，比如 DDL 语句。

ResultSet

java.sql 接口 ResultSet

所有超级接口：

Wrapper

所有已知子接口：

CachedRowSet, FilteredRowSet, JdbcRowSet, JoinRowSet, RowSet, SyncRes
olver, WebRowSet

```
public interface ResultSet extends Wrapper
```

表示数据库结果集的数据表，通常通过执行查询数据库的语句生成。

六、思考问题

1.每次SQL操作都需要建立和关闭连接，这势必会消耗大量的资源开销，如何避免？

分析：可以采用连接池，对连接进行统一维护，不必每次都建立和关闭。事实上这是很多对JDBC进行封装的工具所采用的。

2.Java代码中，传入的数据格式与数据库定义不同怎么办？如把Java的String对象赋值给数据库的tinyint属性。

分析：在执行SQL语句时，数据库会尝试进行转换。根据我的实验，如果内容为纯字母的String对象传入tinyint的age属性时，会被转化成0。具体转化规则应该和数据库有关。

作者：五岳

出处：<http://www.cnblogs.com/wuyuegb2312>

对于标题未标注为“转载”的文章均为原创，其版权归作者所有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

分类：Java, DB, 学习笔记

标签：Java, JDBC, MySQL

好文要顶

关注我

收藏该文



五岳

关注 - 15

粉丝 - 826

+加关注

« 上一篇：优化程序性能的几个方法（来自于《深入理解计算机系统》）

» 下一篇：EditPlus远程编辑、语法高亮、编译运行源代码设置

posted @ 2014-07-28 12:49 五岳 阅读(560990) 评论(24) 编辑 收藏 举报

登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) [博客园首页](#)

- 【推荐】百度智能云618年中大促, 限时抢购, 新老用户同享超值折扣
- 【推荐】大型组态、工控、仿真、CAD\GIS 50万行VC++源码免费下载!
- 【推荐】阿里云爆品销量榜单出炉, 精选爆款产品低至0.55折
- 【推荐】限时秒杀! 国云大数据魔镜, 企业级云分析平台
- 【推荐】华为应用软件专题日 | 生态市场企业特惠GO

园子动态:

- 致园友们的一封检讨书: 都是我们的错
- 数据库实例 CPU 100% 引发全站故障
- 发起一个开源项目: 博客引擎 fluss

最新新闻:

- 风波不断, 留给这届家长的在线教育机构不多了
 - 95后职场图鉴: 躺平、裸辞、焦虑、拼命, 对PUA大胆说不
 - 维基百科不缺钱, 为什么募捐的时候总要道德绑架?
 - 翻红的摩尔庄园, 曾是5000万95后的第一款网游
 - 牵手阿里云! 奇瑞发布首款汽车生态品牌
- » [更多新闻...](#)