

## Java and Database Interaction Analysis and Implementation

Xuewen PAN    Zhengde BAO    Chenxi LI

School of Computer and Software, Jincheng college, Sichuan University, Chengdu, Sichuan, 611731

### Abstract

In the process of learning the Java language, the most important thing is to make a system, such as the educational administration system, course scheduling system, library management system, of course, more important is the theory first, no theoretical derivation and summary is not good engineering, so it is necessary to discuss the idea behind the connection between the database. This data needs to establish a two-dimensional flat table to code rules, read into the database. Then we need to use Java instructions in the background such as Tomcat server to extract or modify or delete the table. Java and database are two independent content originally, the database is through SQL instructions to operate, the SQL language and analyzes the principle of Java code interaction in the is this article discusses the content, this helps to improve the programmer's understanding of itself, at the same time, the realization of the database table is simple, is in the process of the development path, which summarizes the developers often encountered obstacles, IT can help the IT staff in development experience for reference.

### Key Words

Java, JDBC, MySQL

DOI:10.18686/jsjxt.v1i2.683

## Java 与数据库交互的剖析与实现

潘学文    鲍正德    李晨曦

四川大学锦城学院计算机与软件学院, 四川成都, 611731

### 摘要

在学习 Java 语言过程中, 最重要的是做出一个系统, 例如教务系统、排课系统、图书管理系统等, 当然更加重要的是理论先行, 没有理论推导和总结是做不好工程的, 所以在连接数据库之间有必要讨论其背后的思想。这其中的数据需要建立二维平面表格以代码规则, 读取到数据库中。然后我们后台例如 Tomcat 服务器需要通过 Java 指令对其中的表进行提取或者修改删除等。Java 本来与数据库是两个独立的内容, 数据库是通过 SQL 指令来操作, 将 SQL 语言与 Java 代码交互的原理进行剖析是本文探讨的内容, 这样有助于提升编程人员对 Java 本身的理解, 同时数据库的实现建表很简单, 却是开发过程中必经之路, 其中总结了开发人员经常遇到的障碍, 这样可以帮助 IT 人员在开发中借鉴经验。

### 关键字

Java; JDBC; MySQL

### 1. JDBC 的由来

Java 从诞生到现在已经快 25 年, 刚刚诞生 Java 中没有自己的 API, 开发人员需要借用 C 语言的 ODBC。这使得 Java 在做数据库相关工作时常常十分繁琐, 限制了跨平台和它自身的面向对象性。当然 Java 本身的发展逐渐完善, 这主要是大家不断开发的经验, 使得连

接数据库 API 的步骤越来越简单。ODBC 相对于 Java 缺少面向对象的特点, 并且操作步骤, 因为数据库操作在 Java 中占十分重要的地位, 所以 Sun 自己在 JDK 版本 1.x 开发了 JDBC, 当时 JDBC 只是一个组件。在 JDK 版本 1.1 中, SQL 类包 (JDBC API) 成为 Java 语言的标准组件。

## 2. JDBC 体系结构

JDBC 体系结构就是所谓的接口, 通过这些工具我们就可以实现 SQL 语句, JDBC 最大的好处就是, 它的复用性很好, 这也是现在依然有许多人使用的原因, JDBC 可以使用固定的模板访问 Oracle、Mysql、SQLserver, 真正实现了编写依次复用多次, 大大提升了连接易用性、可操作性。

## 3. JDBC 接口剖析

通过使用 JDBC, 程序员可以连接到任何形式的数据库, Java 和 JDBC 的结合可以使编程人员实现跨平台的很好复用性<sup>[1]</sup>, 下面讨论 JDBC 应用程序的类和接口

### 3.1 DriverManager 类

是驱动程序方法的父类, 这个类需知道将要调用哪个类

DriverManger 类调用 Java.lang.Class 类的 forName() 方法指定驱动程序的名称和位置

```
Class.forName("\\公司名.驱动程序名.数据库名\\");
```

当调用 DriverManager.getConnection() 发出连接请求时, 此方法会进行根据提供的参数进行判断连接是否成功

```
Connection getConnection(String url,String user,String password)
```

用户和密码是登录数据库的用户和密码。

```
Jdbc: (subprotocol): (subnmae)
```

对于不同的库, 供应商提供的驱动程序和 URL 是不同的。MYSQL 驱动程序: org.git.mm.mysql.driver

```
URL: Jdbc: mysql: // ip: port: dataname
```

其中 ip、port 是 MYSQL 主机的 IP 地址和端口号, 而 databasemae 是库的名称。

### 3.2 Connectin 接口

Connction 是管理与数据库的连接, 不要忘记关闭连接。DriverMannger 的 GetConnect() 方法建立在 JDBC URL 中定义的数据库的 Connection 间接上:

```
Connection con =
DriverManager.getConnection(url,login,password);
Connection
```

close() 此方法关闭与数据库的连接 connectinname 是 Connection 类对象实例的名称。

```
Connectionnam.close();
```

### 3.3 Statement 接口

此接口用来管理具体的对数据库操作, 此类中包含了查询方法, 修改方法, 更新方法。用户需要的时候直接调用就可以。

```
Connection conn =
DatabaseConnecton.getConnection();
Statement stm = conn.createStatement();
ResultSet rs = tem.executeQuery("select*from Admin_Info");//查询管理员这个表
Connection conn =
DatabaseConnection.getConnection();
PreparedStatement pstmt =
conn.prepareStatement("insert into Admin_Info values(?, ?, ?, ?)");//插入或者更新具体的数值
pstmt.setInt(1, 1);
Pstm.setString(2, "twm");
Pstm.setString(3, "boy");
Pstm.setInt(4, 25);
Pset.executeUpdate();
```

Statement 对象的常用方法如下, (1) execute(): 运行语句, 返回一个是否有结果的

### 3.4 ResultSet 接口

介绍几种常用的 ResultSet 类方法: (1) absolute(); 移动查询时, 记录指针可以上下移动当前指针例如: rs.sbsolute(int line); //rs 为 ResuletSet 对象实例的名称。其中参数 line 是记录指针的位置, 如果大于 0, 则从记录的开头移动指定的数字; 否则表示倒数。

### 3.5JDBC 连接数据的实现

建立数据源<sup>[2]</sup> (建表如表 Admin\_Info) 2 引入 Java.sql 包 3 加载驱动程序 4 创建与数据库的连接 5 创建语句对象 6 编写 SQL 语句 7 执行 SQL 语句 8 处理数据 9 关闭相关对象 10 异常处理完整代码如下

```
String className ="com.mysql.Jdbc.Driver";
String url ="Jdbc:mysql://localhost:3306:test";//创建与数据库的连接
```

```
String username = "root";//用户名
String userpwd="root";//密码
String tableName ="Admin_Info";//要操作的表名
Connection con =null;//建立连接接口
PreparedStatement pstmt =null;
ResultSet = null;//用于查询获取的结果集对象
    Try{
Class.forName(classname);//加载驱动程序
con =
DriverManager.getConnection(rul,username,pword);// 创
建与数据库连接
pstmt= con.perpareStatement("select *
from"+tableName);//创建语句对象
rs. = pstmt.executeQuery();
    While(rs.next());//处理数据， next 方法判断是否有
记录，
    {
for(int i=0;i<rs.getMetaData().getColumnCount();i++)
System.out.println(rs.getString(i+1));//getString() 从 结果
集从获取信息
System.out.println();
    }
}
catch(Exception ex)//异常处理
{
System.out.println(ex.toString());
    }finally{
Try {pstmt.close();} catch(Exception ex) {}//关闭相关对象
Try {con.close();} catch(Exctption ex) {}//关闭相关对象
}
}
```

#### 4.连接池原理

掌握 Java 语言的重要环节是 Java 与数据库连接，是编写后台代码与浏览器交互的重要环节，再怎么深入理解和优化都不为过，传统的连接数据库的方法，采用 JDBC ，但是这种方式会浪费和占用无用的连接<sup>[3]</sup>，会减慢服务器运行速度，而近来业界比较常用的上述技术即连接池的使用提高数据库的连接效率。

连接池的主要原理在于，Java 推出的 JNDI（Java Naming and Directory Interface）即 Java 命名和目录接口，JND 可以比喻为一个将命名统一的盒子，盒子里可

以容纳各种的命名服务，主要作用可以集中使用者的信息，使信息共享。与命名服务不同的是它可以有自身属性，用户可以通过不同的客户端来访问，也可以根据这些属性进行搜索，JNDI API 简单可以理解为一个映射，JNDI 的出现可以很好的实现了 Java 与命名服务的连接。而 JNDI 集合不仅提升了其自身效能还提升了自身的容错机制和负载均衡能力。当然每种接口的设计离不开具体代码的实现，JNDI 也不例外其提供了许多组件执行查询的入口。JDBC 最大的好处是使得数据层与应用程序直接分离，JNDI 有应用程序和服务供应接口组成，并且应用程序和服务商提供了各自需要应用的接口。

#### 4.1 连接池代码

```
public class DB {
    public Connection connect() {
        Connection conn = null;
        Context ctx;
        ctx = new InitialContext();
        DataSource ds = (DataSource)
ctx.lookup("Java:comp/env/mypool");//连接池，性能优于
JDBC，从连接池中看有没有被释放的连接。
        conn = ds.getConnection();
        return conn;}
}
```

上面详细阐述了 JDBC 的原理即，DriveManager 类、Connectin 接口、Statement 接口、ResultSet 接口四大接口以及其中的常用方法。和业界最常用的连接池技术，用户建立连接时需先到连接池取。如果连接被占用，则用户排队，知道有新的被上个用户释放的连接。这样很好的提升系统的性能，避免无效的浪费。由于本文最要讨论数据库技术，所以连接好数据库后我们开始详述如何建立 MySQL 表的操作。

### 5.数据库实现

#### 5.1 操作数据库准备工作

在实现 Java 与 Mysql 之间需要一些准备工作，就是启动数据库，步骤虽然简单但是体现出电脑操作最基本的思想，其中包含本人开发中遇到的一些障碍，下面介绍最常用两种连接 Mysql 数据库方法，第一种为 Mysql 包直接下载到 D 盘的，另一种情况是下载到其他

盘的。第一种情况，命令提示符，进入 Mysql 包中的 bin 命令，然后点击 mysql.exe 文件，会发生闪烁，说明 Mysql 已经启动，这时就可以连接了，输入 mysql -u root -p,这是会提示 Enter password:,然后输入密码。第二种情况，连接时先启动 Mysql,和上文步骤一样，如果 Mysql 没有下载到 C 盘，例如下载到 D 盘。因为 mysql.exe 文件在 bin 目录下，系统得先进入此路径才能启动 mysql.exe 文件。这时输入 D:代表进入 D 盘根目录，然后输入，Cd D:\mySql\mysql-5.7.23-winx64\bin,注意，cd 必不可少，并且 cd 后有空格。

## 5.2 数据库物理结构设计

在后台数据库的建立过程中,需要建立一个数据库<sup>[4]</sup>,在 MySQL 中,SQL 语言使用十分重要,这个数据库基础上对数据库的内容进行增加或删减。我们使用上文的 Admin\_Info 即管理员首先使用命令 create database C2C; create table Admin\_Info(a\_id(15) primary key auto\_increment,a\_name varchar(20) not null,password varchar(20) not null ,email varchar(50) not null,a\_emmo varchar(50) null);回车此五行六列的表创建成功,输入命令, describe Admin\_Info 可以查看此表结构。

序号	Field	Type	长度	Null	Extra
1	a_id	Int		No	Auto_increment
2	a_name	Varchar	20	No	
3	a_password	Varchar	20	No	
4	a_email	Varchar	50	No	
5	a_emmo	Varchar	50	Yes	

表 1 Admin\_Info

同样，如果要删除此表，可以使用命令：drop+表格名称 来删除这个数据库表。在建表之后，就可以在表格中进行插入记录、删除记录等操作。如插入记录：Insert into Admin\_Info() values(100,"panxuwen"12345","12345@qq.com","好评")。这是表格就会存在一个 panxuwen 的用户信息。显示表格中的记录可以使用 select 语句，如 select\*from Admin\_Info,这时就会显示 Admin\_Info 表中的所有记录信息，UPDATE Admin\_Info SET password = '54321' WHERE a\_id = 100;注意此语句几个点，第一，大写的地方不能小写。第二，WHERE 前无空格。此语句的作用是通过 id 来修改管理员信息。

## 6.结束语

通过分析了 JDBC 的连接方式，与基于 JNDI 的连接池方法，我们可以得出结论，连接池连接数据库更加优化。编写 JavaWeb 程序时，我们常常用到连接池方

法<sup>[5]</sup>，其中不仅提供了连接数据库的代码，还提供了对数据库进行增删改查的具体 SQL 代码，虽然技术发展速度十分迅速，但是不变的是原理与计算机基本实现思想，所以不断学习是作为 IT 人员的基本要求。

## 参考文献

- [1] 许峰.JavaWeb 整合开发全程指南[M].第四版本.电子工业出版社,2009,182-184.
- [2] 杨光.JavaWeb 实战开发[M].清华大学出版社, 2014, 178-180.
- [3] 陈虹君. Java Web 与数据库连接池技术的应用研究与实践[D].电子科技大学,2008.
- [4] 张永梅. MySQL 数据库技术在公民健康信息管理系统中的应用[D].西安电子科技大学,2010
- [5] 胡峰. 基于 JAVA 的 B2C 电子商城设计与实现[D].西安电子科技大学,2017.

## 作者简介

第一作者：潘学文（1996-），男，汉，河北省张家口市，本科，四川大学锦城学院，研究方向：J2EE。

第二作者（通讯作者）：鲍正德（1989-），男，汉，黑

龙江哈尔滨，研究生，四川大学锦城学院，研究方向：电子商务。

第三作者：李晨曦（1998-），男，汉，四川省贵阳市，本科，四川大学锦城学院，研究方向：大数据。