一种将结果记录集转换为二维数组的方法及应用①

邓子云[1]，翦象慧[1]，谢英辉[2]

[1](湖南现代物流职业技术学院，长沙 410131)
[2](长沙民政职业技术学院，长沙 410004)

摘 要：针对数据库操作的连接释放、结果记录集对象占用资源、异常处理等棘手的问题，提出了“一种将结果记录集转换为二维数组的方法”，给出了解决问题的思路、数据结构和算法实现，并应用在物流信息系统集成中间件 LESB 的数据交换情形中。

关键词：结果记录集；二维数组；数据结构；算法实现

Method about ResultSet Converted to 2d Array and Application

DENG Zi-Yun[1], JIAN Xiang-Hui[1], XIE Yin-Hui[2]

[1](Hunan Modern Logistics Occupation Technical College, Changsha 410131, China)
[2](Changsha Social Work College, Changsha 410004, China)

Abstract:, To deal with problems like ResultSet object’s occupying resources and abnormal handling in the database connection release, this paper puts forward “a method about ResultSet converted to 2d array”. It gives solution to the problem, as well as data structure. It realizes the algorithm and applies it to data exchange cases about logistics information system integration middleware LESB.

Key words: ResultSet; two-dimensional array; data structure; realizing algorithm

1 问题提出

在应用程序开发的过程中，经常要面对打开和关闭数据库连接，处理 SQL 查询结果记录集等问题，对于这些问题，开发人员常常会遇到以下困难：

（1）打开的数据库连接没有及时释放，而连接数通常是有限的，一旦用完程序将不能再度创建连接。

（2）数据库驱动程序加载，连接对象的创建、关闭，结果记录集的指针处理等操作有可能会产生异常，为增强程序的健壮性，需要及时捕获这些异常，这样会使得代码冗长，且嵌套结构复杂。

（3）在 SQL 查询后得到的结果记录集占用了数据库连接资源，而结果记录集可能需要在较长的代码段中或需要跨对象使用，结果记录集很难满足这种开发需求。

开发人员处理好以上 3 个问题，可能要写相当多的重复而繁杂的代码，那么有没有办法来解决呢？

2 解决思路

以 JDBC 开发为例。生成一个数据库连接、关闭一个数据库连接都是十分消耗系统资源的操作，程序规模小、系统负载低还让人尚可接受，但随着规模的增长、系统交易量的增大，数据库连接的操作显得对系统的性能影响尤为突出[1]。

在对数据库表进行插入（insert）、更新（update）、删除（delete）等操作后，由于更新操作返回的是影响的记录条数，不必再行操作关系型数据，因此可在其语句之后马上加上关闭数据库连接的方法即可[2]。

但对于 ResultSet 来说就不好办了。因为 ResultSet 在进行了 select 操作后，仍然会占用数据库的连接，且此后往往还需编写代码来得到 ResultSet 中的数据。

下面是解决问题的思路：

（1）异常尽量在底层模块中就处理，这样高层模块只需专注于业务逻辑的开发。

① 基金项目:湖南省科技计划(2009GK3182)
收稿时间:2011-01-20;收到修改稿时间:2011-03-05

（2）创建连接池对象来共享数据库连接。

（3）得到 ResultSet 对象后，将 ResultSet 对象数据转储到一个二维数组中，即可关闭 ResultSet 了，而以后查询数据就可以查询这个二维数组就可以了。

基于以上解决问题的思路，可以编写一个数据库操作的类，在类的方法中集中处理异，和开关数据库的操作。

3 数据结构

从关系型数据库表的结构来看，可以将数据结构理解为一张二维的表格，如图 1 所示，实际上就是一个数据矩阵。

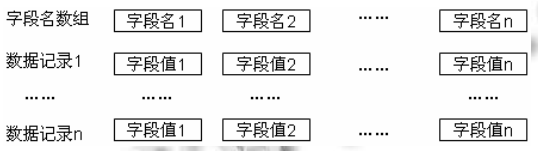

图 1　数据矩阵

在考虑程序中存储 ResultSet 数据的应该是一个二维的数组，但是直接用“[][]”这两个中括号的形式来声明二维数组的话，存在两个问题：(1)字段个数未知；(2)记录条数未知。这样将导致无法为二维数组初始化存储空间。

可以先利用 ResultSetMetaData 对象的 getColumnCount()方法得到字段的个数，以及字段的名称，用 String 数组来存储字段名数组。然后生成一个 ArrayList，其中的元素均为一维的 String 数组。ArrayList 的第一个元素为字段名数组，第二个元素开始为 ResultSet 中的数据，使用 ArrayList 的原因是因为不需要知道记录集中记录的条数，而且 ArrayList 可动态地增加元素。

另外还要考虑数据矩阵中的数据用什么数据类型。在数据库中的数据类型有很多种，为通用起见，可使用 Java 中的 Object 类型，但是又要另外编写数据类型的代码。为简化起见，可设数据矩阵中的数据均为 String 类型，String 类型可由数据库中的大多数数据类型转换而来。

4 算法实现

可设计一个 RsArray 对象，其属性和方法如图 2 所示。

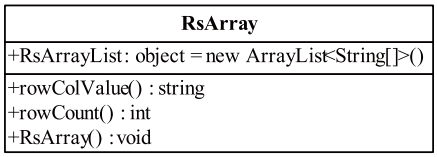

图 2　RsArray 的属性和方法

RsArray 有一个属性 ArrayList，声明如下：

public ArrayList<String[]> RsArrayList=new ArrayList<String[]>();

关键是看其它的操作方法如何实现。一是构造函数 RsArray()，要实现将 ResultSet 对象中的数据转换到 RsArrayList 属性中；二是 rowColValue()方法用于得到指定记录的指定字段的值；三是 rowCount()方法用于得到总记录条数。

先来看构造函数的实现。

```
public RsArray(ResultSet rs){
if(rs==null) return;
try {
  //取得结果集的元元素
  ResultSetMetaData metaData =
    rs.getMetaData();
  //取得所有列的个数
  int colCount = metaData._
    getColumnCount();
  //得到字段名数组
  String[] fieldNameArray=
    new String[colCount];
  for(int i=0;i<colCount;i++){
    fieldNameArray[i]=new String(
    metaData.getColumnName(i+1));
  }
  RsArrayList.add(fieldNameArray);
  //得到 ResultSet 中的数据
  while(rs.next()){
    String[] tempArray=
      new String[colCount];
    for(int i=0;i<colCount;i++)
     tempArray[i]=rs.getString(
    fieldNameArray[i]);
    RsArrayList.add(tempArray);
}
```

```
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

如果要得到某一行某一列的数据，则可参见 rowColValue()方法的实现代码：

```
//参数 rowNo 表示第 rowNo 行，colName 表示列名
public String rowColValue(int rowNo,String colName){
//如果行号比总记录条数还大
if(rowNo>this.rowCount()) return null;
int colNum=0;//列数
boolean findOk=false;//是否找到数据
//找到 colName 列所在的列号
for(int i=0;i<RsArrayList.get(0)._
length;i++){
        if(RsArrayList.get(0)[i]._
        equalsIgnoreCase(colName)){
            colNum=i;
            findOk=true;
        }
    }
    if(!findOk) return null;
    return RsArrayList.get(rowNo)[colNum];
}
```

如果要得到总记录条数，可参见 rowCount()方法的实现代码：

```
public int rowCount(){
    if(RsArrayList==null) return 0;
    if(RsArrayList.size()<=1) return 0;
    return RsArrayList.size()-1;
}
```

为什么记录条数是“RsArrayList.size()-1”呢？因为还有一个字符串数组是记录的字段名。

5 应用分析

在物流信息系统集成中间件 LESB 中，由于该中间件要起到数据交换的作用，每天有大量的货源、车源、专线、仓储等数据要进行交换[3]，这就需要一种结果记录集转换为二维数组的方法，以解决前文中提出的问题。使用情况如图 3 所示。

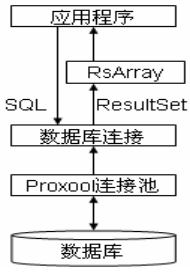

图 3 RsArray 的应用情况

不失一般性，以一次数据交换为例：

```
//查询出前 100 条要交换的数据
sqlStr="select * from exchange_out where status=0
and ROWNUM <= 100 ORDER BY ROWNUM ASC";
RsArray oracleRsArray=oracleDb._
executeQueryArray(sqlStr);
//如果有数据要交换
if(oracleRsArray.rowCount()>0){
    for(int k=1;k<oracleRsArray._
    rowCount()+1;k++){
    String exchangeOutId=
    oracleRsArray.rowColValue(k,"exchange_out_i
d")
    … …
    }
}
```

这样在应用程序中不必再为 SQLException 而烦恼，同时也不必再担心数据库连接没有释放的问题，还可方便地存取数据。

物流信息系统集成中间件 LESB 的数据交换情形如图 4 所示。

图 4 数据交换的情形

(下转第 199 页)

版本下,将图二所示的 XML 文档解析后，其解析结果如下所示：

```
cnumber=0  pre_xml=null  current_xml=vod/vod1.xml  next_xml=vod/vod2.xml
PositionX=65  PositionY=90  Width=150  Height=10
StreamName=Laislabonita_H264.ts  href=rtsp://192.168.12.68/Alizee3_1.5Mb.ts
```

图 3 解析结果

3.2 解析流程

整个解析流程如下图所示：

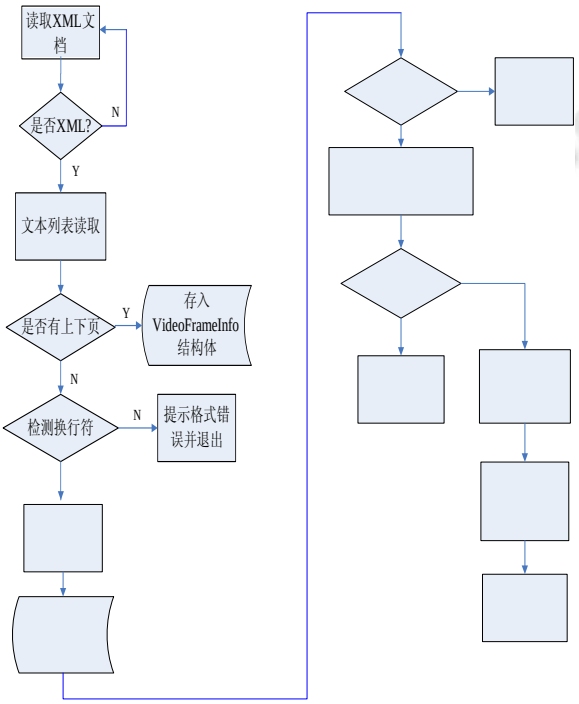

图 4 解析流程图

由如上结果对照图 2 可见，用此方法对按照本文所定义的 XML 文档格式进行文档解析，解析结果完全正确，解析出来的内容可供上层的 API 函数调用，如需更改流媒体内容或界面内容，只需按照 XML 文档格式更改相应内容即可，让服务器端的操作变得轻松，简单，快捷。

4 结论

本文应用 C 语言编程解决了基于 DOM 模型的 XML 解析工作，成功为上层 API 函数提供了所需要的数据，使得整体 IPTV 设计工作大大简化，从而减轻了开发的工作量，加快了研发进程。为 IPTV 的改进和开发提供了一种有效方法。

参考文献

1 谢质文.IPTV 产品,营及案例.北京:电子工业出版社,2008.
2 倪继利.Linux 内核分析及编程.北京:电子工业出版社,2005.
3 李善平,刘文山,王焕龙.Linux 与嵌入式系统.北京:清华大学出版社,2006.
4 Wall K, Watson M, Whitis M. GNU-Linux 编程指南.自由软件基金会,1991.
5 勤研工作室.U-boot 运行机制.http://www.kiyi.com.cn.
6 Salzman PJ, Burian M, Oripomerantz. The Linux kennel module programming Guide, 2005.
7 Prate S. C primer plus.第五版.中文版.北京:人民邮电出版社,2005.
8 冯进,丁博,史殿习,张曈熹,许凯.XML 解析技术研究.计算机工程与科学,2009,31(2):120–124.

(上接第 247 页)

6 结语

本文中给出的思路、数据结构和算法实现，实际上就是“一种将结果记录集转换为二维数组的方法”这种方法已经在物流信息系统集成中间件 LESB 的数据交换情形中进行了应用，较好地解决了前文中提出的数据库操作的连接释放、结果记录集对象占用资源、异常处理等棘手的问题。

这种“将结果记录集转换为二维数组的方法”在各种应用系统开发的过程中具有普遍而实用的价值。

参考文献

1 梁伍七.基于 JDBC 的 Web 应用程序数据库连接技术研究.合肥学院学报(自然科学版),2010,20(4):29–32.
2 梁清翰,沈占锋,骆剑承,吴亚,明令申,李均力.构建 LBS 系统的数据库连接池技术研究.计算机工程,2006,32(18):39–41.
3 邓子云,黄友森,杨晓峰,陈玉林,罗涛.基于 SOA-BPM 组合架构的第三方物流XML 信息集成平台.计算机系统应用,2010,19(3):1–5.