

JDBC 中操作 Blob、Clob 等对象

江苏 无锡 缪小东

很多网友都不知道 JDBC 中如何操作 Blob、Clob 等对象，其实这是不难的！操作这些对象就象操作其它 String、int 一样。不信！你就看下面的文章。下面的例子会带你首先创建包含 Clob、Blob 对象的数据库的表，然后写入这些对象，最后再读取并操作这些对象！

一、关于 Derby

www.open-open.com 是这样介绍 Derby 的“Apache Derby 是一种高质量的、纯 Java™ 的嵌入式关系数据库引擎，IBM® 最近已将其捐献给开放源码社区。Derby 数据库基于文件系统，具有高度的可移植性，并且是轻量级的，这使得它非常便于发布。”据说它的源代码价值几十亿美元哦！我一般使用 Derby，因为使用它可以很多其它开源数据库不能完成的操作，它是一个与 JDBC 标准完全兼容的数据库产品，再说它是 IT 界泰斗 IBM 的开源项目，能不去好好研究，多多学习吗？！

下载该软件你可以去 www.open-open.com、www.sourceforge.net。下载下来后将其 db-derby-10.XXXX-lib.zip 解压放到一个目录下（看到了吧，基本都是 jar 文件，我放在 c:\Derby 目录中），同时让你的 classpath 中包含 derby.jar 文件，这样我们就可以在程序中使用该库了！

简单讲述 path 和 classpath 吧！设置 path 的目的是：当你使用 java、javac、rmic 命令时系统会到指定的 path 下查找这些.exe 或.com 文件然后执行它，我们要使用 java 中的这些工具，当然在 path 环境变量中要包含它们所在的目录了。classpath 也差不多，当你在 java 中要调用其它已经有的类时，必须包含该类所在的目录，这样就可以调用了！例如在使用 javac 编译 servlet 时，sun 提供的 servlet 的包必须要在你的 classpath 中。

下面就开始我们的例子吧！

二、一个简单例子

本例中先创建一个包含 Blob 和 Clob 对象的数据库的表，然后向此表中写入 Blob 和 Clob 对象，最后再将这些数据库表中的数据读出并处理。其中使用了 PreparedStatement 对象、运用了一些和 IO 相关的知识。我们在此程序里使用了两种流的处理方式。下面是源代码：

```
import java.io.*;
import java.sql.*;

public class BlobClobExample {
    public static void main(String[] args) {
        try {
            String url = "jdbc:derby:clobberyclob:create=true";
            //Derby 中的 URL，后面是附加参数。表示数据库中没有此数据库时，会自动建一个
            //其它数据库提供商的产品可以查阅其手册
            Class.forName("org.apache.derby.jdbc.EmbeddedDriver").newInstance();
            //加载驱动。关于加载驱动的细节，请阅读本博客的《JDBC 中驱动加载的过程分析（上）》
            //和《JDBC 中驱动加载的过程分析（下）》
            //在数据库 Mysql 的驱动加载时也建议在 Class.forName()后，继续调用 newInstance()方法
```

//没有本质区别

Connection conn = DriverManager.getConnection(url);

//建立连接

Statement s = conn.createStatement();

//创建 Statement 对象

s.executeUpdate("drop table documents");

//这句是为了避免再次创建表 documents

s.executeUpdate("CREATE TABLE documents(id INT, text CLOB(64 K), photo Blob(1440 K))");

//以上就是创建包含 Clob 和 Blob 对象的表的过程

//和将一般的字段差不多，后面括号中代表该对象的大小，其语法为：

// CLOB (length [{K |M |G}]) 表示长度为 K、M、G 等，没有带表示比特

conn.commit(); //提交操作

//以下是将本地文件中，本程序的源文件加载为一个流，用于向数据库中写入 Clob 字段

File file1 = new File("BlobClobExample.java");

int len1 = (int) file1.length();

InputStream fis1 = new java.io.FileInputStream(file1);

//以下是将当前源代码目录的子目录“11”下的“1.JPG”文件加载为一个流，

//用于后面向数据库中写入 Blob 字段

File file2 = new File("c:\\11\\1.JPG");

int len2 = (int) file2.length();

InputStream fis2 = new FileInputStream(file2);

//以上两个简单吧！！

//创建一个 PreparedStatement 对象，用于批量插入内容

//使用 PreparedStatement 对象可以节省，数据库编译 SQL 指令的时间，

//因为在使用 PreparedStatement 对象时，该 SQL 语句是预先编译好了，

//对于某些变化的参数使用占位符（Place Holder）代替

//这对于以后将要执行多次的同一操作，该操作仅仅是参数不同，是极其有利

//比如在网页上输入客户信息时，用户要填入的数据基本是一样的（你可以填也可以不填）

//这时后台的数据库可以使用 PreparedStatement 对象，每次都是设置参数，执行操作

PreparedStatement ps = conn.prepareStatement("INSERT INTO documents VALUES (?, ?, ?)");

ps.setInt(1, 250);

ps.setAsciiStream(2, fis1, len1);

ps.setBinaryStream(3, fis2, len2);

//以上三步是设置占位符的数值

ps.execute(); //执行操作

//以上四步可以重复执行。PreparedStatement 就是为了适用于此用途

conn.commit();

//以上完成了数据的写入

//以下是数据的读出

ResultSet rs = s.executeQuery("SELECT text, photo FROM documents WHERE id = 250");

while (rs.next()) {

Clob aclob = rs.getClob(1);

//和提取一般对象一样

InputStream is = rs.getAsciiStream(1);

//特殊的，对于与得到 Clob 的流

//这是得到 Clob、Blob 流的第一种方法

```

//以下是对流进行处理的过程。Clob 本身是包含大字符的对象
//顺其自然，以下是使用 java IO 中读取字符流的方法读取它
BufferedReader br = new BufferedReader(new InputStreamReader(is));
String line = null ;
while(null != (line = br.readLine())){
    System.out.println(line);    //将其输出至屏幕，实际你可以按照需要处理
}
is.close();

java.sql.Blob ablob = rs.getBlob(2);    //和操作其它基本类型的字段一样
System.out.println(ablob.length());    //简单地操作 Blob 的实例
//实际你可以使用 Clob 的 API 对其进行任何它允许的操作，请查 API
InputStream bis = ablob.getBinaryStream() ;    //得到 Blob 实例的字节流
//这是操作 Clob、Blob 等对象的第二种方法，也是最自然的用法
OutputStream os = new FileOutputStream("11.jpg");
//用于将数据库中的 Blob 存放到目前目录的“11.jpg”文件中
int b = bis.read();    //以下就是象操作其它 java 字节流一样操作了
while (b != -1 ) {
    os.write((char)b);
    b = bis.read();
}
os.flush();
os.close();
bis.close();
}
} catch (Exception e) {
    System.out.println("Error! "+e);
}
}
}

```

简单吧！实际上写入 Blob 和 Clob 对象时一般只能使用 PreparedStatement 对象，一般使用其 setXXXStream 写入流。写就是这么简单！看完了，研究一下 API 吧！读出时一般通过 getClob()、getBlob() 方法，和其它提取基本数据类型字段的方法完全相同，接着你就可以得到此 Clob、Blob 实例的流了，有了流就可以象处理其它流一样处理这些流了！

简单吧！关键是一定要对基本流程了解！对其原来了解！对什么 IO、net、Collections....基础知识必须精通！

运行的结果是本程序的源代码会存储在数据库中，同时会在屏幕上显式，源代码目录的子目录“11”下的 1.jpg 会存储到数据库中，同时也会在当前目录下有一个该文件的副本。（图片的相对位置一定要搞对哦！）

更多精彩请关注：

<http://blog.163.com/miaoxiaodong78/>