

# SQL Server 大对象数据的访问技术及实现

廖新飞 (温州职业技术学院计算机系 温州 325035)

**摘要:**大对象是指 Word、图像、音频视频等非结构化的二进制数据。SQL3 标准定义的主要的大对象数据类型有: BLOB、CLOB 和 DBCLOB。本文就以 SQL Server2000 中图像文件的存取为例,说明了使用 BLOB 管理大对象数据的方法。

**关键词:**大对象 BLOB 二进制数据 SQL Server2000

## 1 引言

随着企业信息化的快速发展,越来越多的 WEB 数据库应用涉及到大型数据的管理,包括 PDF 文件、Word 文件、图像文件、音频文件、视频文件等。这些数据都是非结构化的二进制数据,称为大对象。对大对象数据的存储可以采用以下三种方式:

- ① 大对象数据以文件的形式保存在指定的路径中,数据库表中存储文件路径和名称;
- ② 在本地用 OLE 存储结构存储大对象数据;
- ③ 数据库中用 LOB 类型字段存储大对象数据。

前两种方式给数据的维护增加了难度,同时,也给数据的安全性带来一定的隐患。并且,大对象数据和普通数据(如 `String`、`int` 等)不同,不能直接在数据表内输入或者使用简单的 `INSERT` 语句插入到数据库中。下面我们就来研究如何利用第三种方法将大对象数据直接存储在数据库的关系表中。

## 2 LOB 简介

LOB(Large Object)的含义是大对象,用来存储大量非结构化的二进制数据。支持大对象(LOB)是目前关系型数据库的一个重要特征,SQL3 标准定义了许多用于管理大对象的数据类型, JDBC2.0 扩展支持 SQL3 新增的大对象数据类型包括:① `ARRAY`: 可将数组以列值存储;② `BLOB`(二进制大对象): `BLOB` 是变长字符串,以字节进行量度,最大长度可达 2 G 字节。`BLOB` 主要用来保存非传统数据,如图片、声音以及混合媒体等;③ `CLOB`(字符大对象): `CLOB` 用于存储大的单字节字符集数据,适于存储文本型的数据(如历史档案、文学名著等)。`CLOB` 被认为是字符串;④ `DBCLOB`(双字节字符大对象): `DBCLOB` 是最大长度可达 2G 字节的

双字节字符变长字符串(1 073 741 823 双字节字符)。`DBCLOB` 用于存储大的双字节字符集数据,如文档等。`DBCLOB` 被认为是图形字符串。

在 SQL Server 中,`BLOB` 可以是 `text`、`ntext` 或者 `image` 数据类型。`Image` 数据类型存储的是长度不确定的二进制数据,能包含最多达 2GB 的二进制数据。`Image` 数据中的数据被存储为位串 SQL Server 不对它进行解释。`Image` 列数据的解释必须由应用程序完成。`BLOB` 数据在 SQL Server 系统中的存储方式不同于普通数据类型,系统会开辟新的存储页面来存放这些数据,实际上存储在 `BLOB` 列中的不是真正的 `BLOB` 数据,而是一个指向 `BLOB` 数据的 16 个字节指针。该指针指向存放该条记录的 `BLOB` 数据的页面。此指针被称为“定位器”。这些所谓的定位器用于表示 `LOB` 值。当我们在一个 `ResultSet` 中检索数据时,将检索的是定位器而非这些定位器所表示的实际的 `BLOB` 值。

## 3 LOB 的操作

操作 LOB 的效率是个值得考虑的问题。由于 LOB 值可以很大,所以将它们从数据库服务器传送至客户机应用程序可能要花费一些时间。然而,一般一次处理 LOB 值的一部分,而不是将它们作为一个整体处理。对于应用程序不需要(或不想要)将整个 LOB 值存储在应用程序内存中的那些情况,应用程序可以通过大对象定位器变量引用此值。然后后续语句可以使用定位器对数据执行操作,而不必检索整个大对象。定位器变量用来减少应用程序的存储器需求,并通过减少客户机与服务器之间的数据流而改进性能。另一个机制是文件引用变量。它们用来直接对文件检索大对象或直接从文件来更新表中的大对象。文件引用变量用来减少应用程序的存储器需求,因为这些变量不

必存储大对象数据。

## 4 大对象应用实例

下面以 Web 应用程序中利用 SQL Server 处理图像文件来说明 BLOB 的具体应用。SQL Server 2000 数据库 picture\_db 中有一基本表 pic\_table ( title ( varchar (30) ), pic ( image ) )。

### 4.1 将图像文件存储到数据库中

以下是将图像文件写入到 SQL Server 数据库中的代码 ( write\_picture.jsp )。

```
<% @ page language = " java" import = " java. util.
*" pageEncoding = " gb2312" % >
<% @ page import = " java. sql. *" % >
<% @ page import = " java. util. *" % >
<% @ page import = " java. text. *" % >
<% @ page import = " java. io. *" % >
<html >
<head >
<title > 将图像文件存储到数据库中 </title >
</head >
<body >
<% Class.forName ( " com. microsoft. jdbc.
sqlserver. SQLServerDriver" ). newInstance ( );
String url = jdbc: microsoft: sqlserver: //local-
host: 1433 ; DatabaseName = picture_db ;
String user = " sa" ;
String password = "" ;
Connection conn = DriverManager. getCon-
nection( url, user, password ) ;
// picture_filename 可以从表单传递过来
的图像的 URL。
String picture_filename = " d: \test. jpg" ;
// 转化为单字符的字符输入流 in
FileInputStream in = new FileInputStream ( pic-
ture_filename ) ;
String sqlcmdtext = " insert into pic_table( title,
pic) values( " test" , ?)" ;
PreparedStatement pstmt = conn. prepareS-
tatement( sqlcmdtext ) ;
pstmt. setBinaryStream ( 1, in, in. available
```

```
());
pstmt. executeUpdate ( ); // pstmt 将记录插
入到数据库中。
pstmt. close ( ) ;
conn. closef ( ) ;
% >
<br >
</body >
</html >
```

说明:

① 获取 BLOB 数据。Image 类型的字段可以存储 exe、jpg、mpg、pdf、wav 等类型的大对象数据。在将这些类型的大对象数据存储到数据库之前, 首先要将文件转换成二进制数据流。代码如下:

```
FileInputStream in = new FileInputStream ( "
filename" );
```

其中, filename 是文件在客户端的实际存储位置。

② 将 BLOB 数据插入到数据库。将得到的大对象二进制数据流 in 插入到数据库的 image 类型字段 pic 中。代码如下:

```
String sqlcmdtext = " insert into pic_table( title, pic)
values ( " test" , ?)" ; // 插入记录的 sql 语句。
```

```
PreparedStatement pstmt = conn. prepareState-
ment( sqlcmdtext ) ;
```

```
pstmt. setBinaryStream ( 1, in, in. available
()); // 设置第一个参数的值
```

```
pstmt. executeUpdate ( ); // pstmt 将记录插入
到数据库中。
```

### 4.2 从 SQL Server 数据库中读出图像

以下是从 SQL Server 2000 数据库中读出图像的代码 ( read\_picture.jsp )。

```
<% @ page language = " java" import = " java. util.
*" pageEncoding = " gb2312" % >
<% @ page import = " java. sql. *" % >
<% @ page import = " java. util. *" % >
<% @ page import = " java. text. *" % >
<% @ page import = " java. io. *" % >
<html >
<head >
<title > 从数据库中读取图像 </title >
```

```

</head >
<body >
  <%
    try{
      Class.forName ( " com. microsoft. jdbc. sqlserver.
SQLServerDriver" ). newInstance ( ) ;
      String url = " jdbc: microsoft: sqlserver: //lo-
calhost: 1433; DatabaseName = picture_db" ; String user
= " sa" ;
      String password = " " ;
      Connection conn = DriverManager. getCon-
nection( url , user , password ) ;
      PreparedStatement ps = conn. prepareStatement
( " select * from pic_table where title = ?" ) ;
      ps.setString( 1 , " test" ) ;
      ResultSet rs = ps. executeQuery ( ) ;
      if( rs. next( ) ) {
        //获取 image 类型字段的二进制流
        InputStream in = rs. getBinaryStream ( "
pic" ) ;
      }
      System. out. println( in. available( ) ) ;
      //将读出的图像存储到文件 test2. jpg 中
      FileOutputStream out = new FileOutput-
Stream( " d: \test2. jpg" ) ;
      byte[ ] b = new byte[ 1024 ] ;
      int len = 0 ;
      while ( ( len = in. read( b ) ) != -1 ) {
        out. write( b , 0 , len ) ;
        out. flush ( ) ;
      }
      out. close ( ) ;
      in. close ( ) ;
      rs. close ( ) ;
      ps. close ( ) ;
    }
  }
  catch ( Exception ex ) {
    ex. printStackTrace( System. out ) ;
  }
  finally{

```

```

    try { conn. close ( ) ; }
    catch ( Exception ex ) { }
  }
}
% >
<br >
<img src = " d: \test. jpg" width = " 200" height
= " 200" >
</body >
</html >

```

说明:从 SQL Server 2000 中读出 BLOB 对象有两种形式。

① 以二进制数据流的形式读出。代码如下:

```
InputStream in = rs. getBinaryStream( " pic" );
```

使用 setBinaryStream 方法存取大对象数据,执行速度快,使用又方便。

② 利用 Java 的 BLOB 对象读出。代码如下:

```
Blob blob = rs. getBlob( 2 ); //获取表中第二
个字段
```

File fileOutput = new File( " d: \test. jpg" ); //生成输出信息到文件的输出流,但是它没有实际的信息,只是定向了输出流的方向

```
InputStream inputStream = blob. getBinaryS-
tream( ); //得到要输出信息的实际信息流
```

```
FileOutputStream fo = new FileOutputStream
( fileOutput );
```

## 5 结束语

本文就以 SQL Server2000 中图像的存取为例,说明了大对象数据的管理。其他大对象数据(如 WORD、音频视频文件等)的存取方法都与此类似。

### 参考文献

- 1 刘彬编著, JSP 数据库高级教程[M], 北京:清华大学出版社, 2006.
- 2 杨钰、卢银娟等, JSP 网络开发技术[M], 北京:人民邮电出版社, 2001.
- 3 张晓东等编著, JAVA 数据库高级教程[M], 北京:清华大学出版社, 2004.