

数据库应用系统中 LOB 应用技术研究

张 静, 王永敏

(西北核技术研究所, 陕西 西安 710024)

摘要:为解决某专项技术数据库应用系统中多类型文档数据、图像数据等大对象数据的组织、存储、管理和高效访问技术问题,研究了应用 Oracle 9i 数据库管理系统对大对象类型数据进行集中存储和统一管理的技术,以及基于 JSP 的网络数据库应用系统对大对象数据的数据访问和数据显示技术。提出采用 Oracle 9i 中 LOB 和 Long Raw 数据类型对大对象数据进行存储和管理,以数据流的形式实现大对象类型数据的输入和输出,并基于 JSP 语言设计客户端软件,提供方便的图形用户接口,有效解决了数据库应用系统中大对象数据的存储、访问与显示问题。

关键词:数据库;大对象;Oracle 9i

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2011)02-0166-04

Research on Application Technology of LOB in Database Application System

ZHANG Jing, WANG Yong-min

(Northwest Nuclear Technique Institute, Xi'an 710024, China)

Abstract: In order to efficiently organize, store, manage and access the large object data of a specific database application system such as multi-document data, image data, etc, studies unified storage and management of LOBS of database application system based on Oracle 9i. This paper presents the solution of accessing and displaying of LOB using LOB and Long Raw data type in Oracle 9i and give out the in-out method of LOB using datastream. The client software based on JSP is implemented which provide convenient user interface. At last, the accessing and displaying problem of LOB in database application system are solved effectively.

Key words: database; LOB; Oracle 9i

0 引言

在数据库应用系统的开发过程中,经常要处理如图片、Office 等带格式文档、视频、音频等复杂数据类型。为了方便数据的管理,通常将其集中存储于数据库中进行统一管理。在数据库中这类复杂数据统称为大对象数据 LOB (Large Objects)。由于 LOB 不同于其它的简单数据类型,它是长度可变的二进制流或字节流,数据的存储和显示方法均具有一定的特殊性,本文结合某专项技术数据库应用系统的开发,研究了基于 Oracle 9i 的数据库应用系统中大对象数据的存储和管理方法,以及该类数据的数据访问和数据显示技术。

1 系统体系结构与开发环境

某专项技术数据库系统采用三层 B/S (Browser/

Server) 开发体系结构,包括数据库服务器、Web 服务器和客户浏览器^[1]。数据库服务器采用 Oracle 9i 数据库管理系统构建。Web 服务器端采用基于 Java EE 的 MVC 三层框架模型,分别为模型层、控制层和视图层。使用 JavaBean 作为模型层的应用,用来存储应用的状态;使用 Jsp 来进行视图层的应用,将模型中的应用展现在用户面前;使用 Servlet+Jsp 作为控制层的应用,将视图层用户的输入请求提交给指定的模型,并将经过模型解释处理过的结果返回给视图。

在 Windows XP 服务器操作系统上,采用 Apache 公司的 Tomcat 5.0.19 提供 Jsp+Servlets 的运行平台,实现 JSP/Servlet 引擎功能和 Web 服务,JVM (Java Virtual Machine) 由 Java 开发工具 JDK 1.6 提供,集成开发环境采用 MyEclipse Enterprise Workbench 6.5.0。

2 大对象类型数据访问机制和存储性能

Oracle 9i 数据库管理系统中提供了 Blob、Clob、Nclob、Bfile 和 Long Raw 数据类型。这些大对象数据类型根据 Oracle 中对象的存储位置不同,可以将其分

收稿日期:2010-05-26;修回日期:2010-08-24

基金项目:国防预研基金项目(51331040302)

作者简介:张 静(1975-),女,硕士研究生,研究方向为计算机仿真、数据库。

为内部 LOB 和外部 LOB^[2,3]。内部 LOB 包括 Blob、Clob、Nclob 和 Long Raw, Oracle 9i 将这些数据存储在数据库内部,可以执行读取、存储、写入等特殊操作;Bfile 是在数据库外部保存的大型二进制对象,这种外部的 LOB 类型,通知数据库记录变化情况,但是数据的具体保存是在数据库外部进行的,数据的完整性由操作系统维护。

本文应用的 Blob、Long Raw 和 Clob 三种大对象数据类型,在存储容量和存储内容上存在差别。Blob 和 Clob 最大可以存储 4G 的数据,而 Long Raw 最大只能存储 2G 的数据,另外, Blob 和 Clob 没有 Long Raw 一个表最多只能有一列的限制^[4]。从存储内容的角度来讲, Clob 只能存储单字节字符数据,多用于存储长文本数据; Blob 和 Long Raw 用于存储无结构的二进制数据,多用于存储图像,视频、音频以及 Word 文档等带格式数据。

LOB 数据的访问与一般类型数据的访问方式是不同的。一般数据类型可以直接通过 select 返回,而大对象数据类型 select 返回的则是定位符,然后以数据流的形式进行访问,其中 Clob 和 Nclob 以字节流的形式访问,而 Blob 和 Long Raw 则以二进制流的形式进行输入和输出。

为了验证 Blob、Long Raw 和 Clob 三种大对象数据类型在存储性能上是否存在差别,本文作了以下性能测试:①将同一文本数据 test. rtf(152kB) 分别写入 Blob 和 Clob 字段,写入前者仅用了 16 毫秒,而写入后者则用了 63 毫秒;②将同一图像数据 test. jpg(465kB) 分别写入 Blob 和 Long Raw 字段,结果表明二者用时相等。实验表明 Blob 和 Long Raw 在存储性能上优于 Clob,但是 Clob 字段在提高长文本数据的检索速度方面存在优势,所以建议用 Clob 存储长文本数据。Blob 和 Long Raw 虽然在存储二进制数据时性能相当,但是 Long Raw 存在单列限制,建议采用 Blob 存储二进制数据。

3 数据库设计

为实现数据的统一管理,某专项技术数据库系统中使用 Blob、Clob 和 Long Raw 三种内部 LOB 数据类型, Blob 用于存储超长带格式文本数据,如 Word 文档、PPT 文件等数据。Clob 用于存储长文本数据, Long Raw 用于存储图片数据。数据库的创建脚本如下。

(1) 创建表 OSI_DEV001 的 DLL

```
CREATE TABLE "SCOTT"."OSI_DEV001" (  
  "NAME" VARCHAR2(100 byte),  
  "DPICTURE" LONG RAW,  
  "MADEUNIT" VARCHAR2(200 byte),
```

```
"THEORY" BLOB DEFAULTEmpty_blob(),  
"TEVALUATE" VARCHAR2(500 byte),  
"COMPONENT" BLOB DEFAULTEmpty_blob(),  
"REMARK" VARCHAR2(300 byte),  
"ID" NUMBER(10) NOT NULL  
)
```

(2) 创建数据表 CLOB

```
CREATE TABLE "SCOTT"."CLOB" (  
  "ID" NUMBER(10),  
  "CLO1" VARCHAR2(10 byte),  
  "CLO2" CLOB DEFAULTEmpty_clob()  
)
```

4 数据访问方法的设计与实现

4.1 数据库的连接

Oracle 提供了 jdbc、ado、bde 等多种数据访问接口,本文采用 jdbc 访问 Oracle 数据库,主要包括连接数据库、发送 SQL 语句和处理请求结果^[4,5]。方法如下:

(1) 建立连接。

首先装载数据库驱动程序类,然后创建 oracle. jdbc 类库中的 Connection 类对象,并通过调用 DriverManager. getConnection 方法建立与数据库的连接^[6]。

(2) 发送 SQL。

本文使用 jdbc 提供的 Statement、PreparedStatement 两个类向数据库发送 SQL 语句。Statement 类实例应用 Connection 类的 createStatement 方法创建, PreparedStatement 类实例由 preparedStatement 创建。

(3) 获取信息。

建立与数据库的连接后就可以运行 SQL 语句对数据库进行操作,用来从数据库中获取数据库元素的信息。

4.2 LOB 数据的输入输出方法设计

4.2.1 Java 类的设计

根据需要操作数据的数据类型不同,需要分别定义完成 Clob、Blob 和 long Raw 数据处理类,实现长文本、带格式 Office 文档和图片数据的存储和读取。数据处理类的设计思想^[7,8]如下。

- 建立数据库连接类 connection 和 statement 类实例。

- 创建 LOB 类实例。

- 创建 ResultSet 类实例,并调用相应成员方法获取 LOB 定位符。

- 创建 LOB 数据的输入和输出类实例,以数据流的形式将源数据写入数据库,读取时则将数据流形式的数据转化成字符信息输出到客户端。

4.2.2 Java 类的实现

(1) Clob 数据处理类

Clob 数据处理类 ClobToolwr 类成员方法如表 1 所示。

表 1 ClobToolwr 类成员方法表

成员方法名	具体说明
ClobToolwr()	ClobToolwr 类的构造函数
readClob (String tableName, String clobName, String condition)	实现从 Clob 字段中读取数据, 参数 tableName 表示表名, condition 是查询条件, 用来确定表中要读取信息的行, 参数 clobName 是 Clob 字段名
writeClob (String tableName, String clobName, String strIn, String condition)	实现将信息内容 strIn 写入指定表的指定位置, 参数 tableName 表示表名, condition 是查询条件, 用来确定表中要写入信息的行, 参数 clobName 是 Clob 字段名

①readClob 的关键代码:

```
StringstrOut = null,sql2;  
java.sql.Clob clob1 = null;  
sql2 = "select " +clobName + " from scott." +  
tableName + " where " + condition;  
ResultSet rs1 = stmt.executeQuery(sql2);  
if (rs1.next()) { clob1 = rs1.getClob(1);}  
if (clob1 == null)  
{  
conn.close();  
return null;  
}  
else  
{  
Readerreader = clob1.getCharacterStream();  
StringBuffer strBuf = new StringBuffer();  
int len = 0;  
char[] b = new char[1024];  
while ( (len = reader.read(b)) != -1)  
{strBuf.append(b, 0, len);}  
strOut=new String(strBuf);  
}  
rs1.close();  
conn.close();  
returnstrOut; //返回输出字符
```

②writeClob 的关键代码:

```
oracle.sql.CLOB clob = null;  
String sql1="updatescott." + tableName + " set "  
+ clobName +"=empty_clob() where " + condition;  
conn.setAutoCommit(false);  
java.sql.ResultSet rs1 = stmt.executeQuery  
(sql1);  
String sql2 = "select " +clobName + " from scott." +  
tableName + " where " +condition + " for update";
```

```
ResultSet rs2=stmt.executeQuery(sql2);  
if (rs2.next()) {  
clob=(oracle.sql.CLOB)rs2.getClob(clobName);  
Writerwriter = clob.getCharacterOutputStream();  
StringReader strRead = new StringReader(strIn);  
char[] b = new char[1024];  
int len = 0;  
while ( (len = strRead.read(b)) != -1)  
{ writer.write(b, 0, len);}  
strRead.close();  
writer.close();  
rs2.close();  
conn.commit();  
(2)Blob 数据处理类
```

Blob 数据的输入输出主要应用 java.io 包中的 InputStream 和 OutputStream 类实现^[9,10]。设计输入方法时,首先获得 Blob 数据的定位符,创建 InputStream 类的子类 FileInputStream 的类实例,并创建 File 类和 OutputStream 类的实例对象,然后从 File 对象创建输入流,调用 FileInputStream 对象的 read 成员方法读取数据,用 OutputStream 的 write 方法将数据写入数据库。输出方法则首先调用 Blob 对象的 getBinaryStream 方法获得 InputStream 对象,并调用 read 方法读取数据,并用标准输出流 system.out 将其输出到客户端。

Blob 数据输入关键代码:

```
oracle.sql.BLOB myblob=null;  
java.io.File outfile=null;  
OutputStream outstream=null;  
ResultSet rs= statement.executeQuery(sql);  
if(rs.next())  
{  
myblob=((OracleResultSet)rs).getBLOB(field-  
name);  
outstream = myblob.getBinaryOutputStream();  
outfile = new java.io.File(filename);  
FileInputStream fin = new FileInputStream(out-  
file);  
byte[] b = new byte[myblob.getBufferSize()];  
int len=0;  
while((len=fin.read(b))!=-1)  
{outstream.write(b,0,len);}  
fin.close();  
outstream.flush();  
}  
outstream.close();  
Blob 数据输出方法关键代码:
```

```
java.io.InputStream fin = myblob.getBinaryStream  
( );  
int i = fin.read(), lg = fin.available();  
while(i != -1)  
{  
    system.out.write(i);  
    i = fin.read();  
}
```

5 应用实例

Java 类定义好后,需要将包含 .class 文件的 Java 包保存在应用系统的 WEB-INF 目录下的 classes 子目录下,在数据访问的 JSP 文件中用 import 命令将其引入,创建类实例并为参数赋值后,就可以调用类成员函数进行数据访问了^[11,12]。

某专项技术数据库系统中对 Clob 数据的存储和显示代码如下。

Clob 数据存储方法应用:

```
ClobToolwr clobtool1 = new ClobToolwr();  
int i = 0;  
while(i < clnum) {  
    try {  
        Stringss = strIn[i].replace(" \r\n", "<br>");  
        ss = ss.replace(" ", "&nbsp;");  
        StringtableName = tableid0, clobName = cname[i],  
        condition = "id = " + rs.getInt("maxid"); clobtool1.  
writeClob(tableName, clobName, ss,  
        condition);  
    } catch (Exception e) { out.print(e.toString()); }  
    i++;  
}
```

在增加数据记录模块中,系统根据数据库定义的字段类型,自动生成不同的用户接口,并调用相应类型数据的数据存储方法存储数据。

Clob 数据读取方法应用:

```
ClobToolwr clobtool1 = new ClobToolwr();  
StringtableName = request.getParameter("tableid");  
StringclobName = rstmd.getColumnname(column);  
Stringcondition = "id = " + rst.getString("ID");
```

```
Stringss = clobtool1.readClob(tableName, clobName,  
condition);
```

```
out.print(ss);
```

Blob 数据和 Long Raw 数据的访问方法同 Clob 数据的访问方法类似,这里不再赘述。

6 结束语

文中研究了基于 Oracle 数据库管理系统的大对象数据存储与访问技术。运用 Java 语言,采用面向对象的编程思想,通过设计并实现大对象数据的数据库操作类,解决了 JSP+Oracle 网络应用系统中长文本数据、图片、带格式文档等复杂数据的数据库集中统一管理和数据访问问题,该技术成功应用于某专项技术数据库系统中,取得了比较好的应用效果。

参考文献:

- [1] Walsh A, Fronckowiak J. Java Bible[M]. 北京:电子工业出版社,2001.
- [2] 王传胜,张凌燕. 信息集成中大对象的研究与实现[J]. 微计算机信息,2009(18):107-108.
- [3] Oracle Corporation. Oracle9i Documentation[EB/OL]. 2009-06. <http://technet.oracle.com/docs/index.htm>.
- [4] 黄天云. 基于 Blob 和嵌入 SQL 的数据库外壳程序设计与实现[J]. 计算机工程与设计,2007(10):2250-2253.
- [5] 李晓黎,刘宗尧. Oracle 10g 数据库管理与应用系统开发[M]. 北京:人民邮电出版社,2007.
- [6] 盖国强. 循序渐进 Oracle—数据库管理、优化与备份恢复[M]. 北京:人民邮电出版社,2008.
- [7] 谷庆华,李成贵. 基于 Java 语言实现数据库的访问[J]. 计算机技术与发展,2008,18(2):5-8.
- [8] Sun Microsystems. JavaTM Platform Standard Edition 6 Development Kit API Document[EB/OL]. 2009-04. <http://java.sun.com/javase/6/docs>.
- [9] 刘玉平. Java 输入输出流及在网站开发中的应用探讨[J]. 电脑编程技巧与维护,2009(18):15-18.
- [10] Mensah K. Oracle Database Programming using Java and Web Services[M]. [s.l]:Digital Press,2006.
- [11] 聂红梅,赵建民. Oracle 数据库中 Clob 大字段的查询优化技术研究[J]. 计算机技术与发展,2006,16(8):33-34.
- [12] 袁健美. 基于 JSP 的 B/S 动态网站开发及数据库连接[J]. 计算机技术与发展,2007,17(6):8-10.

本刊可优先安排发表的四类来稿

- (1) 代表计算机学科最高学术水准,具有重大创新理论与技术的原创性学术文稿。
- (2) 获国家自然科学基金、省(部)以上重点项目计划或其他重要基金资助项目以及重大奖项的文稿(注明基金名称和项目编号)。
- (3) 能反映本学科领域最新发展趋势、最新热点技术,具有独创性、新颖性、探索性、实用性的文稿。
- (4) 凡第一作者为 CCF(中国计算机学会)会员的文稿。