

基于 ORACLE 数据库的多媒体数据 SQL 级操作探究

岳国华,赵静静

(西安科技大学,陕西 西安 710054)

摘要:探索 ORACLE 数据库平台下多媒体数据在 SQL 级直接操作的可行性。以 Oracle interMedia 为背景,运用多媒体数据 PL/SQL API 实现多媒体数据文件存取。虽然在关系数据库中对多媒体数据即非结构化数据进行操作时,和结构化数据的处理方式不同,但在 ORACLE 数据库平台下将基本 SQL 和多媒体数据 PL/SQL API 整合运用,屏蔽了多媒体数据处理的复杂性,对外呈现出一致的 SQL 级操作是可能的。在 ORACLE10g 的数据库平台上,通过分析 Oracle interMedia 体系结构和多媒体数据的特点以及在 ORACLE 中的存储方式,探索在 SQL 级实现常规结构化数据类型与表示多媒体内容的非结构化数据类型统一操作的可行性和实现途径。

关键词:多媒体数据;存储方式;SQL 级统一操作

中图分类号:TP319

文献标识码:A

文章编号:1673-629X(2011)04-0152-04

Research of Multimedia Data Type Operating in SQL Level Based on ORACLE Database

YUE Guo-hua, ZHAO Jing-jing

(Xi'an University of Science & Technology, Xi'an 710054, China)

Abstract: Explore ORACLE database platform multimedia data in SQL level directly the feasibility of the operation. With Oracle interMedia for background, use multimedia data PL/SQL API multimedia data files access. Although in the relational database of multimedia data points-unstructured data is operated, and structured data processing in different ways, but with the help of ORACLE database platform, make basic SQL and multimedia data PL/SQL API integration using multimedia data processing, shielding the complexity of foreign shows consistent SQL level operation is possible. According to the analysis of the Oracle interMedia system structure and characteristics of multimedia data as well as storage mode in ORACLE under ORACLE10g database platform, expound mainly the feasibility and implementation approaches that conventional structured data type and multimedia content which is the unstructured data type are unified operation in SQL level of oracle.

Key words: multimedia data; storage mode; unified operation in SQL level

0 引言

20 世纪 90 年代,多媒体技术和网络技术使人类迅速全面地进入信息社会。随着社会的信息化,承载信息的媒体也逐步表现出多样化、数字化、交互性和集成性等特点,如图形、图像、音频、视频、动画等,这就出现了这些信息资源的管理问题。

数据库的出现是为了解决如何更好地管理数据问题,同样,为了解决管理好多媒体数据,人们很容易地会想到使用数据库。但传统的数据库管理数据的基本

形式是二维表,其存放的都是结构化的数据,在处理文字和数值信息等方面很成功,在处理包含多媒体数据和非结构化数据方面有很大局限性,因此出现了多媒体数据库技术^[1,2]。目前多媒体数据库的实现模型主要采用扩充的关系模型^[3],也就是在较为成熟的关系模型的基础上增加 LOB 类型的数据类型。但是这种处理方式却不能直接在 SQL 级别上对多媒体数据进行操作与维护^[4],多媒体数据库在关系数据库中一般通过应用程序连接数据库,借助于图形界面将多媒体信息转化为二进制进行存取,当然 OLEDB, JDBC^[5]这些标准目前也不支持 SQL 级直接操作多媒体数据的功能^[6]。因此不能不遗憾的看到,现有数据库对一些多媒体数据所特有的操作能力还是很有限的。针对多媒体数据库的性能等现在也遇到了新的问题。虽然如

收稿日期:2010-08-10;修回日期:2010-11-19

基金项目:国家火炬计划立项项目(2008GH011595)

作者简介:岳国华(1962-),男,陕西兴平人,副教授,研究方向为数据仓库、数据库系统分布式应用系统。

此,多媒体数据库仍旧产生了广阔的应用价值和前景,例如网站内容管理、企业信息门户等广泛地采用了多媒体数据库技术。文中分析了 Oracle interMedia 体系结构及其对多媒体数据的支持,通过对传统的 SQL 语句 INSERT, UPDATE 进行扩充,探索了 ORACLE 环境下实现传统数据与多媒体数据 SQL 级别上统一操作的可行性。

1 ORACLE 对多媒体的支持

1.1 多媒体数据特点

传统的数据是用数字或文字来描述,它们具有相同的层次或网络结构,称之为结构化数据,多媒体数据(如声音、图形、动画和视频等)所拥有的信息量特别大,与传统的数据相比,这类数据是以二进制数位表示,它们无法用数字或者统一的结构表示,称之为非结构化数据^[7]。由于非结构化数据不具备严格的结构,较之结构化信息更难以标准化,管理起来较为困难。

1.2 Oracle 对多媒体的支持

ORACLE 数据库平台从 8.0 版以后提供了对大数据对象(LOB)的存储管理支持,特别是 ORACLE10g 对大数据对象数据类型 ORDSYS. ORDAudio、ORDSYS. ORDImage、ORDSYS. ORDVideo 的支持功能进一步得到提升^[8]。对于 LOB,可分为二进制大数据对象(BLOB)、字符大数据对象(CLOB)、Unicode 字符大数据对象(NCLOB)、文件大数据对象(BFILE)。这四种大数据对象皆可作为某个字段数据的声明类型,其中 BLOB、CLOB、NCLOB 为几个内部 LOB,数据通常在数据库中存放,BFILE 为外部 LOB,所存储的只是指向外部操作系统文件的指针。

在访问 BFILE 类型的数据时,只能读取,不能对其进行修改。ORACLE 提供了 DBMS_LOB 包和 UTL_FILE 包可对 LOB 数据类型存取,具体操作见下。

2 Oracle interMedia 体系结构与多媒体数据 PL/SQL API

2.1 Oracle interMedia 体系结构

ORACLE 数据库对多媒体数据的支持是通过 in-

terMedia 结构框架实现的,如图 1 所示^[9],在这个架构下,多媒体数据像传统的结构化数据一样被存储在数据库中。这些多媒体内容和数据能安全地被多个应用程序共享,同时数据易于管理和维护。

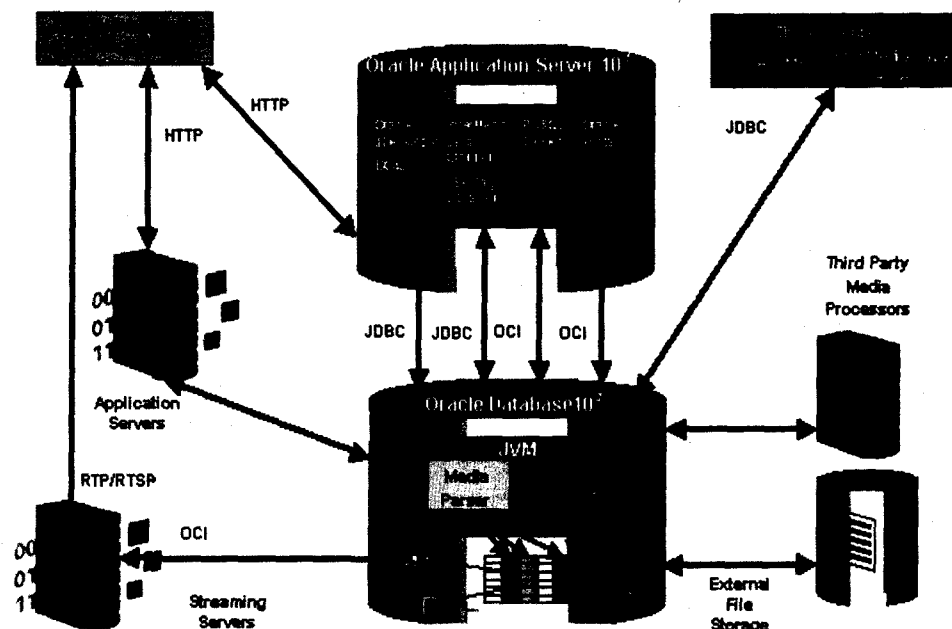


图 1 Oracle interMedia 体系结构框架

ORACLE interMedia 架构是一个三层结构: Oracle 数据库层(Oracle Database 10g); Oracle 应用服务器层(Oracle Application Server 10g); 客户端层(瘦客户端或胖客户端)。

在数据库层多媒体内容被存储在数据表里,通过数据库内嵌的 JVM 这些多媒体数据在服务器端被解析处理;通过使用 interMedia 提供的诸多方法,多媒体内容可在数据库和操作系统文件之间导入、导出,也可以使用 RTSP 协议对流媒体进行处理^[10,11]。

在 Oracle 应用服务器层 Oracle interMedia 提供了丰富的 Java 类能够存取、修改存储在数据库中的音频、视频、图像数据;这些 Java 类对多媒体数据的上传、抽取提供了强有力的支持,SQL 开发者能在 PL/SQL 环境下开发 Web 应用程序,编写 PL/SQL servlet。

在客户端层 ORACLE interMedia 提供了丰富的 Java 类以及 JAI 和 JMF 提供的一组 API 调用接口;来自客户端的任何多媒体数据类型均可直接处理。

2.2 多媒体数据 PL/SQL API

为了处理多媒体数据 ORACLE 提供了丰富的多媒体数据处理 API,这些 API 以 PL/SQL Functions/Procedures^[12]的形式处理和修改 BLOB、CLOB、NCLOB 类型的值,见表 1 和表 2。

3 对多媒体数据的操作

这里所说的多媒体数据包括 image, audio, video 三

个部分^[12],在 ORACLE 中可以直接将多媒体数据看成对象来进行操作,一般使用 ORACLE 自带的 PL/SQL 对其操作,PL/SQL 是 ORACLE 对关系数据库语言 SQL 的扩充,是过程化语言,它集成了现代软件工程特色,将数据库技术和过程化程序设计语言结合起来,是一种高效的 ORACLE 数据库平台下的内嵌式开发语言。

表 1 PL/SQL: DBMS_LOB 过程修改
BLOB, CLOB 和 NCLOB 值

函数/过程	说明
APPEND()	为 LOB 对象附加 LOB 值
COPY()	复制全部或部分 LOB 值到另外一个 LOB
ERASE()	从指定位置开始删除 LOB 值的内容
LOADFROMFILE()	加载 BFILE 数据到内部 LOB 字段
LOADCLOBFROMFILE()	从文件中读字符数据到 LOB 字段
LOADBLOBFROMFILE()	从文件中读二进制数据到 LOB 字段
TRIM()	LOB 值尾部取空
WRITE()	写数据到 LOB 数据指定位置处
WRITEAPPEND()	写数据到 LOB 数据的末端

表 2 PL/SQL: DBMS_LOB 操作 BFILEs 数据

函数/过程	说明
FILECLOSE()	关闭文件
FILECLOSEALL()	关闭以前打开的所有文件
FILEEXISTS()	检查文件是否在服务器上存在
FILEGETNAME()	获得目录名和文件名
FILEISOPEN()	检查文件是否打开
FILEOPEN()	打开一个文件

利用 DBMS_LOB 包和 UTL_FILE 包进行 LOB 数据操作。SQL 仅可以操纵整个 LOB,而不能操纵数据片。DBMS_LOB 包则放开了这个限制,它提供对 LOB 数据中数据片的操作。由于篇幅有限,这里只介绍几个主要的函数:

`v_FileLocator := BFILENAME (v_Directory, v_Filename)` 为读文件初始化一个 BFILE 定位符;

`fileopen()` 用指定的方式打开一个操作系统文件;

`DBMS_LOB. LOADFROMFILE (v_CLOBLocator, v_FileLocator, DBMS_LOB. GETLENGTH (v_FileLocator))`;将整个操作系统文件装入 LOB 中,这个函数用来往目标 LOB 字段填充一个操作系统文件,很有实用价值,因为有许多 LOB 内容在 SQL 语句中是不可描述的,例如一个二进制音频文件,而用这个函数可以将文件内容导入到数据库。

将外部 OS 下的文件装载到数据库的 LOB 字段中,首先要用 `EMPTY_LOB()` 函数初始化 LOB 字段数据为空,获得将要插入的 LOB 字段定位符并锁定该行,并以只读的方式打开要写的文件,最后用 PL/SQL 的 DBMS_LOB 包和 UTL_FILE 包来将外部的文件读入到 LOB 字段。为了方便描述,假设已经存在一个

PERSON 表,它的信息如下:

```
CREATE TABLE PERSON (
  P_Key NUMBER(10) not null,
  P_Name VARCHAR2(50) null,
  P_Photo BLOB default null,
  P_Video BFILE default null,
  CONSTRAINT PK_PERSONID PRIMARY KEY (P_Key));
CREATE OR REPLACE PROCEDURE FILELOAD(
  V_Key number ( 10 ), V_Directory IN VARCHAR2, V_
  Filename IN VARCHAR2 ) IS
  v_FileLocator BFILE; --用 BFILE 数据类型变量来储存文件
  路径
  V_Blob BLOB; --临时存储 LOB 数据
  V_Length BINARY_INTEGER; --文件长度
  Begin
  v_FileLocator := BFILENAME( V_Directory, v_Filename );
  INSERT INTO PERSON ( P_Key, P_Photo ) VALUES ( V_Key,
  EMPTY_BLOB() );
  Select P_Photo INTO V_Blob from PERSON where P_Key = V_
  Key;
  --初始化 LOB 数据并获得 LOB 定位符
  DBMS_LOB. FILEOPEN ( v_FileLocator, DBMS_LOB. FILE_
  READONLY ); --以只读方式打开外部文件
  V_Length := DBMS_LOB. GETLENGTH ( v_FileLocator ); --获
  得外部文件长度
  DBMS_LOB. LOADFROMFILE ( V_Blob , v_FileLocator, V_
  Length ); --将外部文件装载到 LOB 类型中
  DBMS_LOB. FILECLOSE ( v_FileLocator ); --关闭外部文件
  COMMIT
  END FILELOAD
```

4 SQL 级实现传统数据与多媒体数据统一操作

目前对多媒体数据在关系数据库中的存储处理是通过应用程序链接关系数据库,借助于图形界面将图形数据转化为二进制进行存取,没有一个直接在 SQL 级操作与维护多媒体数据的方法,可以通过 JAVA 语言,实现一个扩展的 SQL 命令扩展处理接口(和 SQL/PLUS 界面类似),设计出 INSERT 和 UPDATE 语句扩展语法,实现常规数据和多媒体内容数据操作方式的统一,例如:

ESQL>INSERT INTO PERSON(P_Key, P_Name, P_Photo) values(1, '张水成', 'zsc. bmp')/M: P_Photo;

这条执行时系统对其进行语法分析,发现其中有多媒体存储字段 P_Photo,因此文件“zsc. bmp”的内容将插入数据表 PERSON 中,而不是将字符串“zsc. bmp”插入到 P_Photo 字段。这条 INSERT 语句就是对基本的 SQL 中的 INSERT 语句的扩充,实现了多媒体数据和常规数据操作方法的统一(同样 UPDATE 语句

也有类似的扩展语法)。

语句的实现算法是:当得到一条 SQL 语句后,首先用正则表达式来判断是否有 LOB 类型数据的操作(/M 选项),如没有,直接用 executeupdate() 方法执行 SQL 命令,如果有,就需要按多媒体数据来处理,可以调用 ORACLE 存储过程,让其数据的存储在数据库端自动实现,ORACLE 的内嵌 JVM 机制可以执行 JAVA 代码调用存储过程,实现多媒体数据的载入。例如上面已经有了一个名叫 FILELOAD 的存储过程,当系统执行过程中调用方法 LoadMediaData 实现多媒体数据载入。

```
public static void LoadMediaData ( int docId, String dirPath,
String fileName) throws SQLException{
    Connection con = null;
    CallableStatement proc = null;//创建一个 CallableStatement
    对象 proc
    try {
        con = connectionPool.getConnection();//连接数据库
        proc = con. prepareCall ( " { call FILELOAD ( ?, ?, ? )
        } " );//调用名为 FILELOAD 的存储过程
        proc.setInt(1,docId);
        proc.setString(2, dirPath);//传入文件路径
        proc.setString ( 3, fileName );//传入文件名称
        proc.execute();//执行存储过程的操作
    } finally {
        try { proc.close(); } //关闭存储过程
        catch ( SQLException e ) {}
        con.close();//关闭数据库
    }
}
```

上述过程对于用户是透明的,用户只需在命令行直接使用扩展后的 SQL 命令,类似于普通 SQL 语句一样执行 INSERT/UPDATE,不再需要对 LOB 字段的非结构化数据做特殊的处理,而这个处理过程由 ORACLE 来自动完成。对用户来说,不需要考虑用高级语言编写一个程序处理这些二进制数据内容,操作变得相对简单,同时也提高了多媒体数据的安全性。

5 结束语

文中主要是基于 ORACLE 数据库环境对多媒体数据的操作的探究,虽然多媒体数据的存储在 ORACLE 端由存储过程实现,但这并不影响应用端的操作,对应用端而言,只需要掌握 SQL 语言即可。通过 ORACLE 数据库环境下实现 SQL 级对多媒体数据和普通的结构化数据统一操作,简化了不同数据类型操作的复杂度,它们在 SQL 级操作方法的统一为后台数据管理和维护提供了方便,提高了数据维护效率,也从而降低了用户的学习使用成本。

参考文献:

- [1] 李逸波. 多媒体数据库技术[M]. 北京:机械工业出版社, 2004.
- [2] 黄志军, 曾 斌. 多媒体数据库技术[M]. 北京:国防工业出版社, 2005.
- [3] 潘江波, 冯兰萍. 在 Oracle 数据库中如何处理多媒体数据[J]. 福建电脑, 2005(3): 41-42.
- [4] 李 波, 朱庆生. 一种多媒体数据存储、查询、检索的新方法[J]. 计算机工程与科学, 2000(2): 97-99.
- [5] Time 研究室, 萧仁惠, 陈锦辉. JDBC 数据库程序设计[M]. 北京:中国铁道出版社, 2004.
- [6] Zukowski J. Java2 从入门到精通[M]. 邱仲潘, 译. 北京:电子工业出版社, 2002.
- [7] 岳国华. ORACLE InterMedia 多媒体数据存取技术与应用[J]. 西安科技大学学报, 2007(3): 467-471.
- [8] 孙风栋, 闫海珍. Oracle 10g 数据库系统性能优化与调整[J]. 计算机技术与发展, 2009, 19(2): 83-86.
- [9] Oracle10g Database Documentation Readme(CD)[M]. California: Oracle Corporation, 2007.
- [10] Narasimhalu A D. Multimedia databases[J]. Multimedia Systems, 1996, 4: 226-249.
- [11] Santini S, Gupta A. Principles of schema design for multimedia databases[J]. IEEE Transactions on Multimedia, 2002, 4(2): 248-259.
- [12] Allen C. Oracle Database 10g PL/SQL 入门教程[M]. 北京:清华大学出版社, 2005.

(上接第 151 页)

- [6] 潘洁珠, 吴共庆, 胡学钢, 等. 基于领域知识的预警规则发现研究[J]. 计算机技术与发展, 2008, 18(7): 30-35.
- [7] 窦万春, 苏 丰, 蔡士杰, 等. 面向知识应用和交互的工作流系统建模与控制[J]. 计算机研究与发展, 2003(2): 66-67.
- [8] 张德海, 沙月林. 基于本体与工作流的知识服务系统[J]. 计算机工程, 2009(19): 75-79.
- [9] 王 浩, 武 凌, 张 海, 等. 基于 Ontology 的工作流知识管理系统的应用研究[J]. 计算机技术与发展, 2008, 18

(6): 23-28.

- [10] 孙 超, 张仰森. 面向综合语言知识库的知识融合与获取研究[J]. 计算机技术与发展, 2010, 20(8): 28-29.
- [11] 陈 志. 对商业银行现行个人贷款信用风险管理的思考[J]. 北方经贸, 2005(4): 61-63.
- [12] Walker F. CrossRef and SFX: complementary linking services for libraries[J]. New Library World, 2002, 103 (1174): 83-89.