

基于 PB 数据管道的异构数据库转换系统设计与实现

Design and Implementation of Heterogeneous Database conversion System Based on Data Pipeline

张丽华 (嘉兴学院信息工程学院 浙江嘉兴 314001)

摘要:数据管道是完成数据转换与迁移的有效工具。结合 ODBC API 编程和动态数据窗口技术,探讨了通过程序动态创建数据管道对象,利用数据管道实现异构数据库通用数据转换与迁移的关键技术与实现方法。

关键词:数据管道 异构数据库 数据转换 ODBC API 数据窗口

1 引言

随着计算机和网络技术的不断发展,企业内部各个部门都开发出适用于自己部门的管理信息系统,这些系统在发挥一定功效的同时也逐渐显现出局限性:因采用不同的数据库管理系统,形成多种架构的系统在企业中并存但各自为政、信息不能互通的局面。如何最大限度地保护用户资源,实现异构数据库之间的连接、数据交换和数据共享,充分使用异构数据源中的大量信息,是应用程序开发中不可忽视的问题。

针对以上问题,笔者提出了一种基于 PowerBuilder 的异构数据库通用数据转换和迁移的方法。利用 ODBC API 编程^[1,2]和 PowerBuilder 的动态数据窗口技术^[3],在程序中动态创建数据管道对象,并利用它完成数据迁移,为异构数据库互连互通提供便利。

2 数据管道

数据管道(Data Pipeline)是 PowerBuilder 提供的

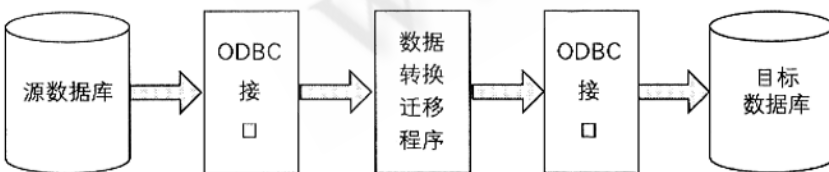


图 1 异构数据库转换系统模型

一个用于数据转换与迁移的智能对象,它提供了在数据库内部、数据库之间甚至数据库管理系统(DBMS)之间快速复制数据的途径。

一般在程序中使用数据管道进行数据转换与迁移的方法是:首先在 PB 开发环境中创建 Data Pipeline 对象,需要预先确定源数据库、目的数据库、源表、目的表、数据管道操作方式、运行数据管道时事务提交的频率、容许出现的最多错误数等属性;然后定义一个标准类 Pipeline 类型的用户对象,指定此用户对象的 DataObject 属性为预先建立的 Data Pipeline 对象;最后通过用户管道对象的 Start() 函数执行数据管道对象,将数据从源表传送到目标表^[4,5]。这种方法要求在 PB 开发环境中预先建立数据管道对象,存在一定的局限性,灵活性差。

笔者通过对 Data Pipeline 对象和标准类 Pipeline 进行分析,提出了一种新的使用数据管道的方法,允许用户根据需要设置源数据与目标数据库,根据用户的选择在程序中动态创建数据管道,最后执行数据管道进行数据转换与迁移。这种方法灵活性高,通用性强。

3 关键技术与实现方法

3.1 系统设计模型

通过 ODBC 为统一的数据库访问接口,同时连接源数据源和目的数据源,利用 PowerBuilder 的数据管道,根据一定的转换规则进行用户定制的数据转换与迁移,该模式的结构如图 1 所示。

通过 ODBC 可以实现对多种 DBMS 的数据进行转换。目前绝大多数关系数据库都提供了 ODBC 驱动程序,所以该转换系统应用面比较广。

在实施数据转换与迁移之前可以提供相关数据的显示和用户的定制功能,系统的互操作性强,灵活性大。

3.2 数据类型转换

由于不同的数据库管理系统(DBMS)使用的数据类型、数据格式不同,要进行异构数据迁移就需要解决数据类型的转换问题。在具体实现中,可以预先通过使用 ODBC API 函数 `SQLGetTypeInfo` 找出所连接的两种不同 DBMS 之间的数据类型映射关系,把各种 DBMS 之间的数据类型转换关系以数据表的形式保存在中间数据库中,在应用程序运行时根据源数据类型从类型转换表中找到并取出对应的目标数据类型。

以标准 ODBC SQL 类型为参考,找出不同 DBMS 之间的数据类型映射关系。应用程序分别连接源数据源与目标数据源,调用 `SQLGetTypeInfo` 函数,使用参数 `SQL_ALL_TYPES`,可得到相应数据源支持的所有数据类型的信息结果集。比较这两个结果集,找出 `DATA_TYPE` 字段值相同的记录,取出这些记录各自对应的 `TYPE_NAME` 字段的值,就可以得到源和目标这两个不同的 DBMS 的数据类型映射关系^[6]。可能出现一个源数据类型对应多个目标数据类型的情况,在创建数据管道对象时可以让用户根据具体需要进行选择。

3.3 ODBC API 编程

ODBC(Open Database Connectivity,开放式数据库互连)是 Microsoft 公司提供的用于数据库访问的应用最广泛的应用程序编程接口。对于不同的数据库,ODBC 提供了一套统一的 API,使应用程序可以使用这些 API 来访问任何提供 ODBC 驱动程序的数据库。利用 ODBC API 编目函数,很容易得到数据源的数据表和字段的各元数据信息。如源表名可使用函数 `SQLTables` 取得;源表各字段名、字段在数据库中的数据类型、是否为主键、是否允许为空值等属性可以用函数 `SQLColumns` 获得^[6]。

ODBC API 编程主要有以下几个步骤:

- (1) 申请环境句柄,建立 ODBC 环境;
- (2) 申请分配一个连接句柄,ODBC API 根据用户提供的数据源名,用户名,密码等信息,建立到数据源的连接;
- (3) 申请分配一个语句句柄,通过它执行 ODBC 的 SQL 命令;

(4) 执行 SQL 命令,返回结果集;

(5) 释放语句句柄,断开同数据源的连接,释放连接句柄,最后释放 ODBC 环境。

3.4 动态数据窗口

数据窗口是 PowerBuilder 提供给数据库开发人员的强有力的快速开发工具,而动态数据窗口技术是更加高效、灵活地使用数据窗口的一种方法,能大大提高应用程序的灵活性和实际运行效率。动态创建数据窗口对象的一般步骤为:

- (1) 创建数据窗口控件,用于放置动态创建的数据窗口对象;
- (2) 根据实际需要构造一条 SQL SELECT 语句;
- (3) 利用事务对象的对象函数 `SyntaxFormSQL()` 生成符合数据窗口对象语法的字符串;
- (4) 使用数据窗口控件或数据存储的对象函数 `Create()` 创建数据窗口对象;
- (5) 利用新创建的数据窗口对象对数据库进行操作。

在该数据转换系统中,多处采用这种动态数据窗口技术,如为获得选中字段的 `pdtype`、`key`、`initial_value` 等属性而创建的数据窗口;对源数据表进行数据筛选时的条件设置窗口中的字段名称下拉数据窗口等。

4 实例

4.1 建立数据连接

首先由用户选择源数据库和目标数据库,为源和目标库分别创建 ODBC 数据源。然后定义两个事物对象变量并创建对象实例,通过这两个事物对象分别与源数据库和目标数据库建立连接。实例中将数据从 Adaptive Server Anywhere 数据库转换复制到 MS SQL Server 数据库。

4.2 生成数据管道的语法字符串

在程序中创建数据管道对象的关键是生成数据管道的语法字符串。应用程序连接源和目标数据库后显示界面如图 2。通过 ODBC API 编程,调用函数 `SQLTables()` 得到源数据库表结果集,再配合使用函数 `SQLBind()` 和 `SQLFetch()` 获得源数据库中各数据表名称。用类似的方法调用函数 `SQLColumns()` 得到指定表的字段结果集,可以获得表中各字段的字段名称等属性。对于二进制大对象类型的字段需要作特殊处理,所以

另设列表框显示。

用户选择源数据表与相关字段,系统根据源数据库

name = " ole. id" , dbtype = " char (20)" , key = yes, nulls_allowed = no))

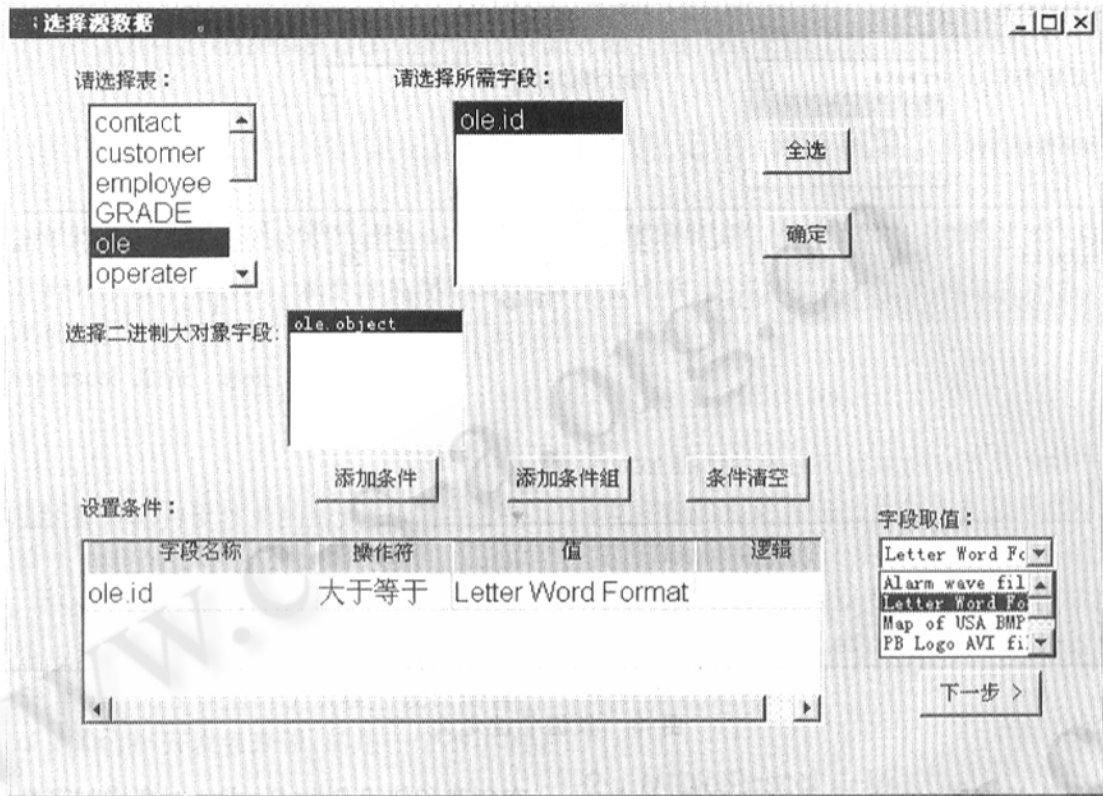


图 2 选择源数据

类型、目标数据库类型和所选源字段的数据类型从类型转换中间库中找到目标数据库的对应数据类型,并把相关信息,如源字段名称、源字段类型、目标字段名称、目标字段类型和是否关键字等信息写入临时中间表,以便组织数据管道语法字符串时使用。系统提供条件设置对源表进行数据筛选,允许进行多个条件组合筛选。单击“下一步”按钮,显示数据管道定义窗口,如图 3 所示。用户定义目标表名、主键名、管道执行方式等,允许用户修改目标字段名和数据类型等信息。

应用程序根据用户的选择和定义,组织生成数据管道的语法字符串。以下是实例中从 Adaptive Server Anywhere 数据库迁移数据到 MS SQL Server 数据库的执行成功的数据管道的语法字符串:

```
PIPELINE( source_connect = mydb, destination_connect = mydata, type = create, commit = 100, errors = 100, keyname = " oleid" )
```

```
SOURCE ( name = " ole" , COLUMN ( type = char,
```

```
RETRIEVE ( statement = " select ole. Id from ole where ( ole. Id > = Map of USA BMP' )" )
```

```
DESTINATION ( name = " ole" , COLUMN ( type = char, name = " Id" , dbtype = " char ( 20 )" , key = yes, nulls_allowed = no ) )
```

```
BLOB( source_table = " ole" , keylist = " id" , SOURCE_COLUMN ( type = blob, name = " object" , dbtype = " long binary" ) DESTINATION_COLUMN ( type = blob, name = " object" , dbtype = " image" ) )
```

该语法字符串主要有 5 部分内容组成:

(1) PIPELINE 段定义源数据库、目的数据库、管道执行方式等属性;

(2) SOURCE 段定义源表名和源字段属性;

(3) DESTINATION 段定义目的表名和目的字段属性;

其中源表或目的表的字段属性包括:该字段的 PB 数据类型 (type)、字段名 (name)、数据库中的类型

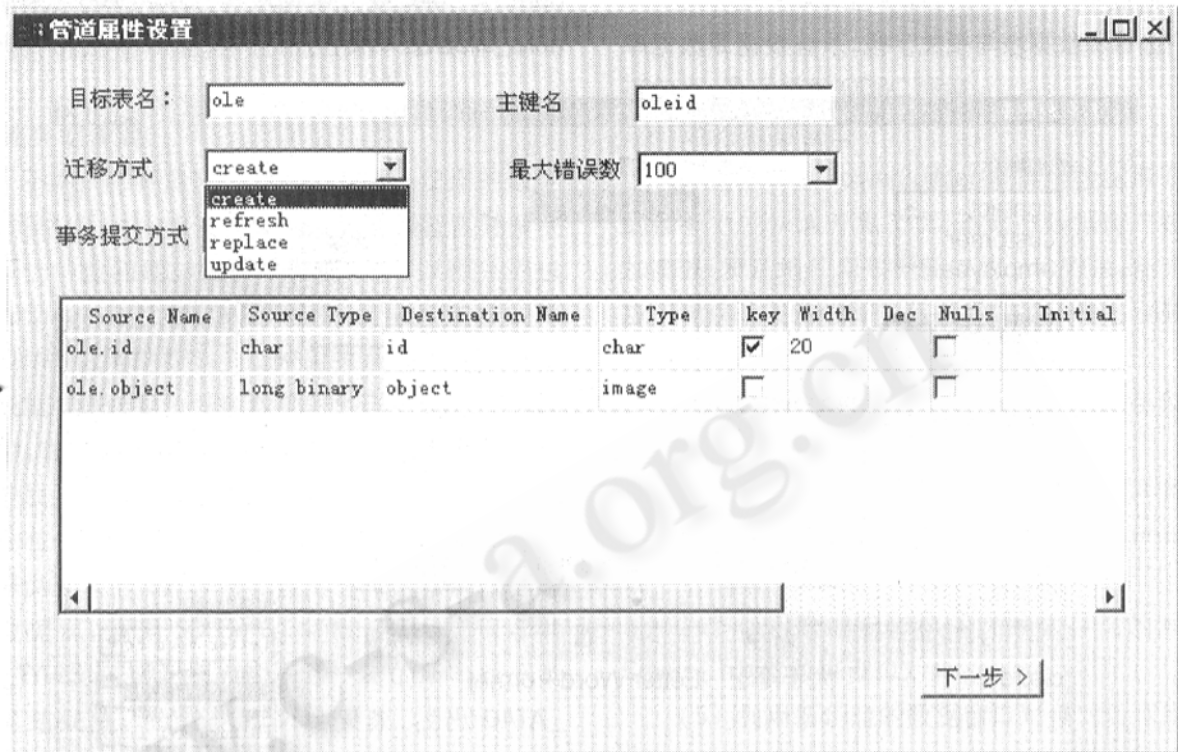


图 3 数据管道定义

(dbtype)、是否为关键字(key)、是否允许空值(nulls_allowed)等。

(4) RETRIEVE 段定义对源数据表进行数据筛选的 SQL 语句;

(5) BLOB 段定义 blob 类型的字段的转换,若源数据中不包括 blob 类型的字段,则数据管道语法中就不包含 BLOB 段。

4.3 执行数据管道

定义标准类 Pipeline 类型的用户对象 u_pipe,把以上字符串赋值为 u_pipe 的 syntax 属性值,然后用 u_pipe 的 start() 函数执行数据管道,完成数据的转换与迁移。若在数据转换与迁移的过程中出现数据的不唯一、数据类型的不一致等错误,出现的错误提示保存在与数据管道对象相关联的数据窗口中,用户可以在数据窗口中修改数据,然后将修改结果传送到目标数据库。

5 结束语

有效使用异构数据源中的大量信息是目前很多应用系统开发中要面临的问题。PowerBuilder 的数据管道技术为异构数据库数据转换和迁移提供了方便,动

态创建和使用 Data Pipeline 对象增加了数据转换和迁移的灵活性。ODBC 通过使用数据库驱动程序获得数据库独立性,屏蔽了底层数据库系统的不同,避免了应用程序随数据库改变而改变的情况,使程序具有较好的通用性。

参考文献

- 1 张宏林等, Visual C++ 6.0 程序设计与开发技术大全[M], 人民邮电出版社, 2004。
- 2 马贵安等, PowerBuilder Win32 API 程序设计(基础卷)[M], 清华大学出版社, 2004。
- 3 杨昭等, PowerBuilder 9.0 数据窗口技术[M], 中国水利水电出版社, 2004。
- 4 赵祥等, 使用 PB 数据管道对象实现多数据库备份与更新[J], 计算机工程与应用, 2004. 40(26): 163~165。
- 5 张少中等, 基于 PowerBuilder 的异构分布式数据库访问[J], 计算机工程与应用, 2003. 39(18): 199~201。
- 6 ODBC API Reference, <http://msdn.microsoft.com>