

基于多事务对象的数据库互操作的实现

电子工程学院系统集成技术研究所
安徽省粮油进出口公司

张冠华 胡荣贵 陆余良 邓晓鹏
张新民

▲ 本文针对 MIS 开发中经常出现的数据库互操作问题,提出了一种基于多事务对象的解决方案,并给出了应用 PowerBuilder 开发的一个实例。

一、引言

数据库是 MIS 的核心,对数据的操作则是 MIS 开发及应用的关键。许多基于小型本地数据库的应用系统要求对关键性的数据进行备份,以确保其安全;在某些大型的 MIS,尤其是跨地区的网络 MIS 中,由于各个部分可能独立运行基于某种数据库的系统,而定期通过网络向 MIS 中心上报一些数据,往往采用多种异构数据库,这就要求系统能对不同的数据源进行互操作。上述这些问题的解决实际上就是使得多数据源(包括同构的和异构的)能进行互操作。

PowerBuilder 是 Sybase 公司基于 Client/Server 体系结构的一种数据库开发工具,它支持多种网络环境,提供多数据源(包括异种数据源)、多事务处理,对数据库操作简单方便,并给出了对两个以上数据库互操作的有效机制,非常适合开发基于数据库的 MIS。下面具体介绍如何利用 PowerBuilder 进行开发,实现多数据库互操作。

二、事务对象

PowerBuilder 事务对象(Transaction Object)一个内置的不可见类(Class),用于

充当应用程序与后端 DBMS 之间的通信区域。它可以有多个实例,每个实例对应于一个数据库连接,从而应用多事务对象可对多个数据源进行互操作。

事务对象包含属性、事件和函数。它有 10 个属性可用于连接数据库,究竟用其中的哪些属性取决于具体的 DBMS。一般说来,连接某个数据库只要用到其中的几个属性。表 1 列出了事务对象的所有属性及其使用说明。

表 1 中的后 5 个属性用于传递作为 SQL 命令执行结果的返回信息。每当 SQL 命令执行之后,PowerBuilder 都会根据产生的结果更新这 5 个属性中的一个或多个的内容。

SQLCode 属性是 PowerBuilder 一个比较关键,比较活跃的属性,它使得 PowerBuilder 的代码与平台无关。每当执行 SQL 命令时,PowerBuilder 都会根据执行结果更新 SQL Code 属性的值。程序中常检查 SQL Code 的值以判断执行 SQL 命令后的状态。

● 操作成功,SQLCode 置为 0。

● 一条合法的 SELECT 语句没有返回任何数据,SQLCode 置为 100。这可能由于没有找到与检索条件相匹配的数据,也可能由于利用游标

(cursor) 取数据到了最后一行,SQLCode 被置为 100,表明到了数据表的末尾。

● 出现 DBMS 错误,SQLCode 置为 -1。

三、数据管道

数据管道(Pipeline)提供了数据复制功能,它允许从一个 DBMS 向另一个 DBMS 拷贝表和数据。数据管道功能可以同 PowerBuilder 所支持的所有数据库一起工作。

数据管道不仅仅是复制

为选项。当管道运行时,SQL 语句被执行,目标表被创建并装载抽取出来的数据。SQL SELECT 语句是复制数据的基础,可以用来连接操作的多张源表。通过它,用户和开发人员可以在各种数据源(包括同一数据库、同种数据库和异种数据库)之间进行数据操作。

数据管道提供了创建、覆盖、刷新、追加和更新目标表的功能选项。

● 创建(Create)——把表加

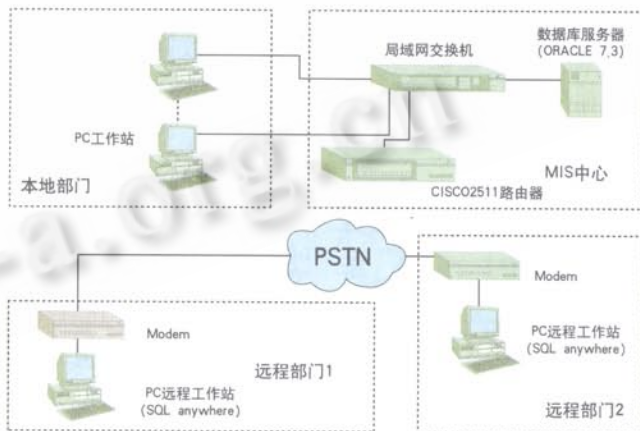


图 1 某集团公司网络拓扑结构及数据库分布

DBMS 表。开发人员完全可以控制哪些应包含或不包含在目标表中的数据段,以及使用哪些字段来生成目标表的索引。这通过为源表创建一个 SQL SELECT 语句来实现,该语句有可选的检索参数,并且使用数据管道画板(painter)来说明关键域和行

到目标数据库中,如果说明了主关键字则创建索引,并且按照所选择的字段和 where 子句说明装载数据。

● 覆盖(Replace)——删除目标数据库中的表并创建该表。目标表必须已经存在。

● 刷新(Refresh)——这种选项假定目标表已存在。如果

表1 事务对象属性

属性	数据类型	说明
DBMS	String	后端数据库平台的供应商名(如 "ODBC", "O73ORACLE7.3")
Database	String	数据库实例名称
UserID	String	用户ID或用户名
DBParm	String	连接过程中用来向数据库传递参数的字符串
DBPass	String	用户ID对应的数据库口令
Lock	String	当连接到数据库时被用来定义隔离层的锁定值
LogID	String	服务器用户ID(可能与数据库ID不同)
LogPass	String	数据库服务器所在的网络名称,即数据库服务所在网络的地址名 (x: =SQLSPX, t: =TCPIP)
AutoCommit	Boolean	自动提交标记
	TRUE	执行一条SQL命令后自动执行COMMIT命令
	FALSE	只在执行COMMIT命令时提交事务
SQLcode	Long	指示最近操作的成功或失败的编码:
		0 成功
		100 无返回数据
		-1 出错
SQLNRows	Long	选中或更新的记录数
SQLDBCode	Long	DBMS返回的错误码
SQLErrMsg	Long	DBMS返回的错误消息
SQLReturnData	String	视DBMS的不同,返回不同的数据(例如, Informix 利用该属性来返回插入行的序列号)

目标表中存在某个记录,则该记录被删除然后再插入,新记录直接插入。

●追加 (Append) ——将新记录插入到目标表,使用该选项可能在目标表中插入重复记录。

●更新 (Update) ——更改目标表中已经存在的记录。新记录直接插入。对于Update选项,必须指定主关键字。对于Create和Replace选项,主关键字是可选项,若指定则创建索引。

数据管道对象可在数据管道画板里生成,也可根据需要动态生成。

四、数据库互操作的实现

下面给出一个实例,说明如何实现多数据库的互操作。

1. 网络拓扑结构及数据库配置

假定某单位拥有若干部门和一个MIS中心,有些部门和MIS中心地理上分割,其拓扑结构和数据库分布如图1所示,具体配置为:

●MIS中心。数据库服务器操作系统采用UNIX,数据库选用大型系统数据库ORACLE; CISCO2511提供远程拨号访问服务。PC工作站采用ORACLE的CLIENT/SERVER结构工作。

●本地部门。本地部门的PC工作站操作系统为Windows 98,采用ORACLE的CLIENT/SERVER结构工作。

●远程部门。远程部门的PC工作站操作系统为Windows 98,采用本地数据库SQL anywhere,同时安装ORACLE客户端程序,通过PSTN与MIS中心的数据库服务器互连。

各远程部门的PC工作站可独立的在本地SQL

anywhere上工作,根据需要可将要上报的一批数据传至总公司的ORACLE数据库服务器或从总公司下载数据至本地。这种工作方式与实时的在ORACLE上工作相比有如下优点:在实时性要求不高的情况下(大多数MIS都能满足),可节省高额的长途电话费,提高数据传输的可靠性(Modem在短时的连接中掉线的可能性小)。

2. 远程数据传送流程

本文讨论如何从分公司向总公司传送客户资料表。此表(custom)表结构如表2所示。

将custom中的录入日期insert [[CD#*2]] date作为源表的检索参数,使满足检索条件的客户资料表的数据传送到总公司。具体实现框图如图2所示。

3. 创建数据管道

在数据管道画板中设计数据管道对象p custom upload。其实现某一个远程部门向MIS中心传送客户资料表。源数据库为SQL anywhere,目的数据库为ORACLE。源数据库表为custom,对源数据的获取设置一个参数,为Insert date,对应于新客户的增加日期,对目标表进行Update操作。

4. 创建标准的类用户对象

利用用户对象画板(user object painter)创建一标准的类用户对象,其继承至PowerBuilder的内置的系统对象pipeline,为其取名为pipe wmeter。

5. 连接

首先创建类型为transaction的两个实例,对应于SQL anywhere本地数据库和

表 2 客户资料表 (custom) 结构

字段名	类型	注释
inser date	date	录入日期
custom ID	integer	客户号 (关键字)
name	varchar(8)	客户姓名

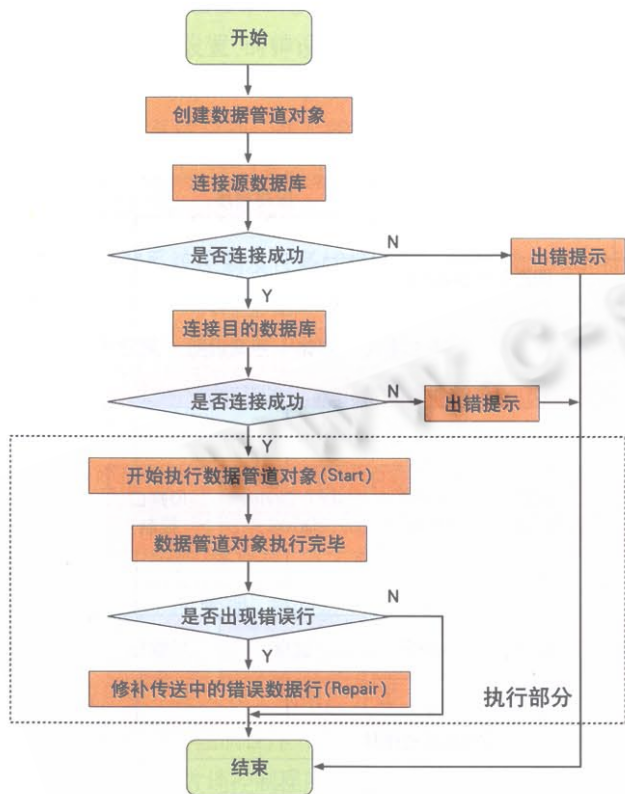


图 2 实例中的总体设计方案

ORACLE 数据库, 代码如下:

```

// Create an instance of
// Create an instance of type transaction on oracle
type transaction on SQL transaction ltran oracle
anywhere transaction tran
ltran sqlany tran ltran oracle tran =
ltran sqlany tran = CREATE transaction
CREATE transaction

// Populate the attribute
// Populate the attribute ltran oracle tran,DBMS =
ltran sqlany tran,DBMS = "O73 ORACLE 7.3"
"ODBC" ltran oracle tran,LogID =
ltran sqlany tran,DbParm "dev"
= "Connectstring='DSN= ltran oracle
example'" tran,LogPassword = "dev"
ltran oracle

```

```

tran.ServerName = @tns:wgs
server orcl

```

```

// which user defined
to display the error columns
during execute process

```

除了上述的描述方法外, 也可将数据库的连接信息写入应用程序的配置文件, 然后用 ProfileString 函数获取。

创建之后, 对两个事务对象分别进行连接, 即可对两个数据库进行互操作。

```

CONNECT USING ltran
oracle tran;

```

```

CONNECT USING ltran
sqlany tran;

```

6. 执行

在程序中利用数据管道进行数据操作时需要创建一个 p pipe wmeter 类用户对象的实例 uo_pipe, 并设置其 dataobject 属性, 使其与在数据管道画板中预先设计的数据管道对象连接。其代码为:

```

p_pipe_wmeter uo_pipe
uo_pipe = CREATE
p_pipe_wmeter
uo_pipe.dataobject =
"p_custom_upload"

```

数据管道为数据管道的执行提供了三个函数, 下面具体谈一下它们的用法。

Start——执行一数据管道对象。它有三个主要参数: 源数据源事务对象, 目的数据源事务对象, 显示传送错误的窗口对象。若数据管道对象设计时设置了 SQL SELECT 的检索参数, 则将它们依次列在上述三个参数的后面。本例中将执行以下代码:

```

// the dw_error argument
of this function is a
datwindow

```

```

uo_pipe.start
(ltran_sqlany_tran,
ltran_oracle_tran, dw_error,
l_insert_date)

```

Cancel——取消执行数据管道对象, 没有参数。在数据管道已开始执行但没执行完的情况下可以用此函数取消执行。

Repair——修补在报错数据窗口中出现的错误, 只有一个参数, 为目的数据库的事务对象。在 Start 函数执行后, 如返回参数表明数据传送过程中有出错的行, 则用此函数进行修补。

五、结束语

PowerBuilder 提供的数据管道基于多事务对象, 为实现多数据库的互操作提供了一个高效、方便、快速的方法。它不仅为编程开了绿灯, 也为开发人员进行数据转换提供了一个很好的环境。笔者在进行 MIS 开发的过程中充分认识到了这一点。

参考文献

- [1] Jason Coombs, Ted Coombs, Ed Ashley 著, 舒少文, 屈健译。PowerBuilder 4.0 深入编程。北京: 电子工业出版社, 1997
- [2] Kent March, Bruce Braunstein 著, 廖卫东, 李奇, 吴洁明译。PowerBuilder 4 应用程序开发指南。北京: 清华大学出版社, 1996