

# 在 PowerBuilder 中对嵌套使用游标的优化

程 岑 (武汉水利电力大学 430072)

**摘要:**本文介绍了利用 PowerBuilder 的数据窗口和数据库系统的存储过程的特性,对嵌套使用游标的不足之处提出的两种改进方法,并给出了应用实例和相关的源程序代码。

**关键词:**PowerBuilder 嵌入式 SQL 游标 数据窗口 存储过程

## 1. 问题的提出

随着信息产业的高速发展,人们希望提高整体管理水平,建立全集团、全公司的企业级信息管理系统时发现,以往已完成的各部门系统采用的是不同的数据库管理系统(DBMS)。为了充分利用已有的信息系统,必须选择一种兼容多种数据库的前台开发工具。

PowerBuilder 正是这样一种面向对象、独立于数据库的可视化前台开发工具,是当前功能最强大的客户机/服务器模式的数据库开发系统之一。它通过对不同数据库采用不同接口的形式同时支持多种关系数据库,并提供了数据管道(Data Pipeline)允许在两个相同或不同的 DBMS 管理的数据库之间复制表的结构和表中的数据,这样原有的数据库和表就可以完全为新系统所使用了,但是适用于原有数据库系统的表结构并非对新的 DBMS 合理,例如没有主键和规范的字段作为外键建立多表之间的关联,即使 PowerBuilder 中提供的一种易用而高效的信息访问方式—DataWindow(数据窗口),这时也显得无能为力。这样对数据的查询、整理和统计将会带来一定的麻烦,但 PowerBuilder 支持嵌入式的 SQL 语句可以有效的帮助我们完成数据的操纵。

本文就游标的嵌套使用结合自己在编程中的实践介绍两种改进方法。

在 FOXPRO 数据库中有一条 SELECT 语句,可以选择不同的工作区,同时打开多个表,在多表中灵活操作记录指针实现对数据的访问。而在 SQL 语句中提供了游标,其本身是由结果集和指向结果集中特定记录的游标位置组成,通过游标的循环可以对结果集中的记录逐行进行处理,若使用嵌套游标,便可完成多表间的同时操作(不过请大家注意使用嵌套游标时,应尽量避免对同一表多次定义和打开不同的游标进行修改,这样容易造成游标交互时的错误操作)。虽然嵌套游标实现起来方便,但主要存在两方面的问题:一是在应用程序中嵌套 SQL 语

句,程序执行速度慢,对于记录数和反复循环次数多的嵌套游标,这种问题就更加突出了,而 PowerBuilder 提供的数据库窗口和大多数数据库所支持的存储过程能很好地提高程序的执行效率;二是很多数据库系统所支持的游标不具备双向滚动的功能,这对数据的操纵有很多的不便之处,而 PowerBuilder 提供的强大的数据窗口函数可有效的解决这个问题。

## 2. 改进方法及实例

本文用两个应用实例具体介绍两种能同时打开多表的方法,从而优化了游标的嵌套使用。

(1)游标和隐含数据窗口的配合使用。用游标打开一表后,再同时操作多个隐含的数据窗口,就可达到同时操作多个表的目的。数据窗口具有丰富的函数,可以使记录指针按需要任意滑动,比游标操作起来更灵活、方便,且查询效率比嵌入式的 SQL 编码效率高。

具体操作是在当前数据窗口中设置一隐含的数据窗口控制部件,即将其 visible 属性设为 false,将第二个要打开的表导入一数据窗口中,并与此隐含数据窗口控制部件相关联,在 PowerBuilder 中的脚本(script)如下:

```
string psuc-1
datetime u1, uss-date-1
int i
dw-2.DataObject = "dw-ndmjs0"
//指明与隐含数据窗口控制部件关联的数据窗口名字
dw-2.SetTransObject(SQLCA) //指定事务变量 SQLCA
DECLARE cur-ndmzb Cursor
FOR select psuc, uss-date
from ndmzb;
open cur-ndmzb;
fetch cur-ndmzb into :psuc-1, :uss-date-1;
do while sqlca.sqlcode = 0
insert ndmjs0(psuc, uss-date, ec)
```

```

select psuc, uss-date, ec
  from ndmnjs where psuc = :psuc-1;
dw-2.retrieve()
i = dw-2.Find("es>'a'", 1, dw-2.rowcount())
  //找到第一个 es 标识大于 a 的记录号
u1 = dw-2.GetItemDatetime(i, uss-date)
update ndmzb set uss-date = :u1 where current of cur-nd-
mzb;
  //改变表 ndmzb 当前行的 uss-date 字段的值
fetch cur-ndmzb into :psuc-1, :uss-date-1;
loop
close cur-ndmzb;

```

若在数据窗口中要实现对表中记录的登、删、改操作并需保存时,不宜采用此方法。因为保存修改时要用到的事务处理语句 commit 和 rollback,在执行时会关闭所有打开的游标和存储过程。

在 PowerBuilder 5.0 中也可以采用 DataStore 对象来代替隐含数据窗口完成上述功能。DataStore 系统对象是一种数据窗口控制,它可以用来检索和更新数据库,但该数据对象不具有数据窗口的可视属性,是一种隐含的数据窗口。DataStore 系统对象建立步骤如下:

```

datestore xt-2 //定义 datestore 变量 xt-2
xt-2 = CREATE datestore //产生实例
xt-2.DataObject = "dw-ndmjs0" //连接数据窗口对象
xt-2.SetTransObject (SQLCA) //指定事务变量 SQLCA
xt-2.retrieve() //至此可以对实例进行操作了
DESTROY xt-2 //注销 xt-2 实例
xt-2 控件的其余使用方法和上面的 dw-2 控件一样。

```

(2)利用存储过程嵌套打开游标。存储过程是存储在数据库里保存的以 SQL 编写的代码段,在首次运行时被编译,并存储在过程高速缓存的内存中,由于它们已预先经过编译和优化,只要通过传输存储过程名及其参数到 DBMS 就可执行,所以该过程大大降低了网络通信量。若这些代码由应用程序发送,则网络通信量和数据库系统对 SQL 语句的语法检查、编译、分析和优化的工作量都将是相当大的。若 SQL 语句较复杂或循环次数多时,应尽量使用存储过程,它的执行工作都是在数据库服务器上进行的,故能得到最高的执行效率。

为灵活地处理结果集,可以在存储过程中使用多层

游标。本文以使用的 SYBASE 数据库为例,该操作分两步完成:

①在数据库中创建存储过程。在 Data Painter 工作窗口单击 Painter Bar 上的 Admin(数据库管理)按钮或选择菜单栏中 Objects 菜单下的 Database Administration 选项,则打开了数据管理画板(Database Administration Painter),此时便可输入 SQL 语句创建存储过程,并使用流控语句如:IF...ELSE、BEGIN...END、WHILE、DECLARE、GOTO 等来组织程序,输入完毕后,单击画板中的 Execute(执行)按钮或选中 Objects 下拉式菜单中的 Execute SQL 选项,PowerBuilder 就将 SQL 语句提交给 DBMS 了。

②在 PowerBuilder 中执行存储过程。在 PowerBuilder 脚本(script)中写如下程序执行存储过程。

```
DECLARE lsp-dc Procedure for proc-dc
```

```
@u1 = :u1-date; //定义一个 Procedure 类型的变量 lsp-dc
```

```
Execute lsp-dc; //执行已定义的存储过程
```

```
Close lsp-dc; //关闭存储过程
```

此外,像任何其他编程语言一样,存储过程的代码应以结构化和易于维护的方式进行构造,存储过程可以调用另外的存储过程,即过程嵌套,过程嵌套被限制为 16 层,通过观察全局系统变量 @@nestinglevel 就能确定当前的嵌套层。

### 3. 结束语

虽然在 PowerBuilder 中,SQL 语句进行数据操纵只是作为数据窗口的一种补充手段,但在应用 PowerBuilder 与其他数据库程序作接口或进行多表之间的复杂统计查询时,SQL 语句就显得必不可少,它的执行效率问题也就变得尤为重要了。本文就此提供了两种可行的方法。总之,在开发应用程序中若能充分利用 PowerBuilder 所提供的强大功能和数据库系统的特点,结合实际需要灵活应用,定能使开发工作达到事半功倍的效果。

### 参考文献

- [1] Peter Hazlehurst 著,周太保等译。Sybase System XI 实用大全。北京科海培训中心,1997。
- [2] 徐东晖、桑大勇著。PowerBuilder 4.0 使用精解。西安电子科技大学出版社,1996。

(来稿时间:1998年8月)