

C/S 应用系统中 PowerBuilder 与数据库的配合问题

王芳 李之棠 刘竞 (华中理工大学 430074)

目前开发中最常用的客户端开发工具为 POWER-BUILDER(PB), 服务器端采用 SYBASE 数据库管理系统。由于 SYBASE 和 PB 各自的功能都在不断加强, PB 功能向数据库方向伸展, 数据库功能向开发工具方向发展, 许多功能的实现相互交叉。因此在开发中对于某些特定问题或功能, 究竟在客户端用 PB 实现还是在服务器端由 SYBASE 完成是数据库设计人员和 PB 开发人员常遇到的问题。这里我们就高校医院管理系统的开发进行分析, 和大家共同探讨。

1. 数据库表间关系(相关完整性)的维护及实现

维护数据库表间关系的规则, 保持表间数据的合理完整性, 这一功能可以在 PB 中用嵌入式 SQL 语句实现, 也可以在服务器端用 SYBASE 中的 TRIGGER 给予实现。例如在教工医疗保健中, 每个教师每年进行体检就会保留一条记录, 当某教师记录因故从现有职工表中删除时, 相应多条体检记录也应从体检记录表中删除。这一功能可以在如下窗口事件处理程序中用 powerscripts 和嵌入 SQL 完成:

```
scripts for cb-delete
.....
row = dw-person.getrow()
cardno = dw-person.getitemstring(row, 'cardno')
dw-person.deleterow()
if dw-person.update()=1 then
    commit;
    DELETE tijian FROM tijian
    WHERE cardno = :cardno;
else
    rollback;
end if
```

但采用这种方法程序维护很困难。当反映现实规章制度的表间关系发生变化时, 需要重新修改相应的程序, 重新编译提交给用户, 对变化反应迟缓。若采用 TRIGGER 实现可避免此种问题。

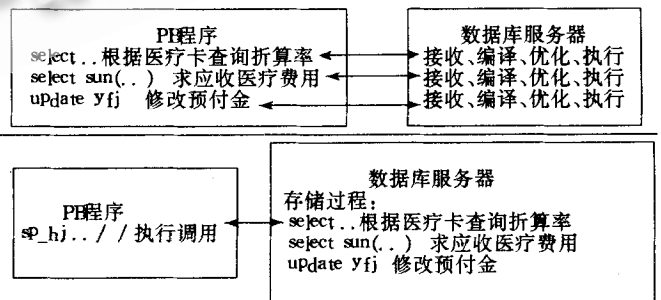
TRIGGER 是数据库对象之一, 采用这种方法实现, 体现表间关系的规章制度可由数据库集中维护控制, 规

则变化时只需修改相应 TRIGGER 既可体现变化, 使得系统易于维护, 提高了运行效率, 也使 PB 编程简单化。

```
create trigger del-person on tijian, deleted
for delete as
    delete tijian
    from tijian
    where tijian.cardno = deleted.cardno
```

2. 体现制度的数据运算的维护实现

此功能可以在 PB 中编程实现, 也可以用 SYBASE 的存储过程完成。采用 PB 实现方式, 当应用中多次出现相关的数据运算时, 需多次复制相应程序, 一旦数据运算规则发生变化, 难于及时一致地修改程序中的多次出现的数据运算, 这不利于系统维护。采用存储过程实现, 制度的变化只影响相应存储过程, 应用中使用数据运算的地方只需重复调用存储过程即可, 不用做任何修改, 减轻了 PB 编程复杂度。同时由于存储过程是预先在服务器端优化编译好的, 执行效率更高, 降低了网络通信量, 提高了整个系统运行效率。高校医院划价收费时根据病人身份(学生、教工、子女、离退人员)采用不同折算率求出药费总和并从预付金中扣除。按照常规 PB 编程需要多次查询求和及修改数据库内容, 用存储过程只要调用一次就可全部完成。



3. 视图与查询(QUERY)

视图是数据库对象之一, 查询是 PB 中预定义的常用查询语句。定义数据库窗口的数据源时, 可以选用视图或查询, 两者都能使应用数据窗口简单, 但又有各自不同的特点。

视图是从一个或多个基本表或其他视图导出的逻辑表,通过它可以看到用户感兴趣的内容。视图简化了数据库的存取,使用户眼中的数据结构简单而直接了当,大大简化用户的数据查询操作。视图提供了保持数据库安全性的手段。我们提供某一视图只让用户看到允许他看的数据而不是整个表。当视图和基表属主相同时,通过授权用户使用视图,屏蔽其对基表的操作权限,使用户只对特定的行列进行访问。

查询是 PB 中的对象,不存在授权机制,它直接对应于一个或多个表,不能象视图那样可以建立在其他查询之上,所以在表征复杂的数据变换上不如视图。当数据窗口的数据源为查询而建立在多表上时,可以通过一定技术修改对应表的数据;若数据窗口的数据源为视图时,禁止修改对应基表的数据。

4. 列约束条件

开发中实现列的约束条件有三种途径

- (1)数据库中定义列约束条件;
- (2)PB 的数据库画笔的列扩展属性中定义校验规则;
- (3)PB 的数据窗口中定义列校验规则。在数据库中定义的列约束条件是数据库对象,是维护数据库完整性的一种内部机制。不管数据输入途径是什么,只要进行数据提交送入数据库这些约束条件就会起作用,但这对于用户来说往往反应太迟了一点,并且返回的错误信息会让用户不知所措。PB 数据窗口中的列约束条件是以列扩展属性中校验规则为缺省设置的,只在数据录入时起作用。当用户在数据窗口中输入了错误数据时,应用程序可以及时检测并报告,让用户立即修改。所以通常在数据窗口中实现列约束,对于重要数据在数据窗口和数据库同时重复定义列约束条件,这样即可及时向用户报告错误信息又可防止用户使用其他工具绕过 PB 应用程序输入的错误数据。