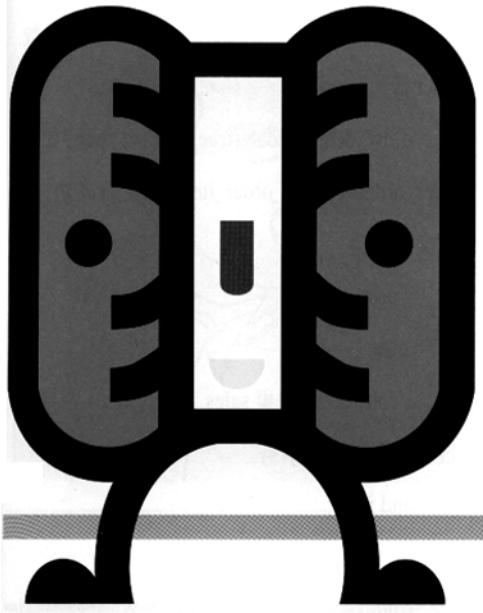


主细表更新在PowerBuilder中的实现

Implement on Updating Master-detail Table with PowerBuilder

陈渝 （昆明理工大学 管理与经济学院 650093）

摘要：本文介绍了在PowerBuilder开发环境中如何实现数据库中主细表数据完整性的维护。

关键词：事务 完整性 主细表更新 POWERBUILDER

1 引言

在数据库应用系统的开发中，为了维护系统中实体间数据的完整性，经常在表结构的定义中使用参照完整性，即使用外码来保证参照实体间数据一致。有参照关系的表有时称为主细表。现在许多软件开发工具中均能够较好的实现主细表中数据的一致性。但是数据库应用系统比较突出的开发工具PowerBuilder则必须构造一段程序来实现主细表中数据更新。下面介绍这段程序如何构造。

2 实现步骤

2.1 主细表的定义

在数据库系统设计时，必须考虑每个实体(ENTITY)和实体间的联系(RELATIONSHIP)，这样就能使系统形成一个整体。但同时也应考虑实体完整性（每个实体的主码不能是空值）和参照完整性（参照实体间的数据依赖关系），这样系统中的数据才是一致和有用的，能够避免“垃圾”数据的存在。下面以一个简化的产品定单来详细描述数据库中的主细表定义。

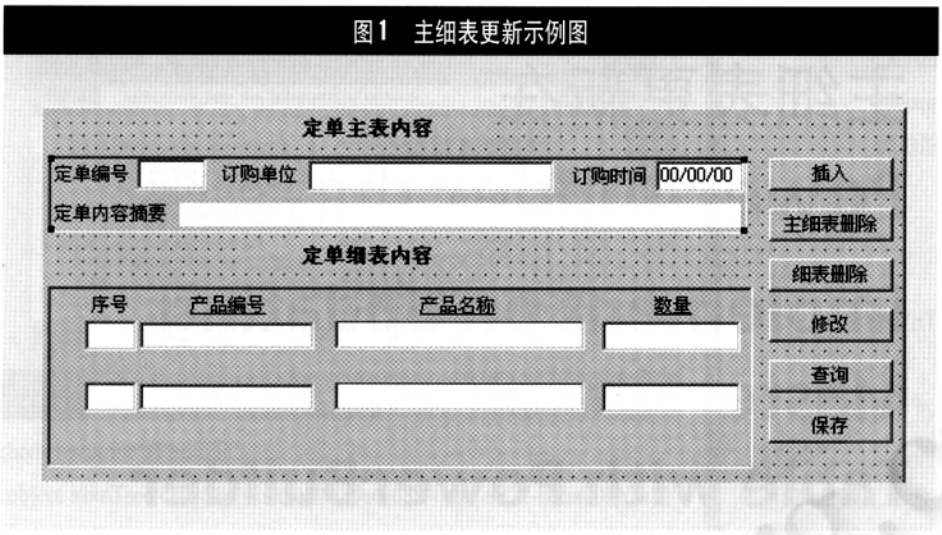
```
CREATE TABLE sales_order
(order_id varchar(10) NOT NULL primary key,
/*order_id 定单编号 */
order_dept varchar(30) NOT NULL,
/*order_dept 定单订购单位 */
order_date date NOT NULL,
/*order_date 定单时间 */
abstract varchar(40)
/*abstract 定单内容摘要 */
);
CREATE TABLE sales_order_items
(order_id varchar(10) NOT NULL,
/*order_id 定单编号 */
item_id smallint NOT NULL,
/*item_id 定单细表行编号 */
product_id integer NOT NULL,
/*product_id 产品编号 */
quantity integer NOT NULL,
/*quanlity 数量 */
PRIMARY KEY (order_id,item_id),
FOREIGN KEY sales_order_items_fk_1 (order_id)
REFERENCES sales_order
);
```

在上述表结构的定义中，表Sales_order称为主表（master table），表sales_order_items称为细表（detail table），两个表之间是一对多的关系，通过在表sales_order_items中的字段order_id上定义外码（foreign key）来设置两表的参照完整性。表sales_order_items中的字段order_id中需插入或修改的数据必须是在表sales_order的字段order_id中已经存在的数据，否则在sales_order_ items中不能保存数据，同时，数据库系统会提示错误信息。因此数据更新时在两个表的字段order_id上必须保持数据完整性，避免出现"有头无尾"或"有尾无头"的数据存在。由于有数据关联限制，在数据的插入过程中，必须先插入主表的数据，然后再插入细表的数据，删除时则先删除细表的数据后，才能删除主表的数据。

2.2 PowerBuilder中实现主细表更新

在数据库的更新（insert,update,delete）中，事务是保证数据间完整性的前提，对于主细表

图1 主细表更新示例图



的更新必须是将这一更新过程定义为一个事务，这样就保证数据更新要么全部成功要么全部失败。事务一旦成功则提交（commit）修改数据到数据库中，否则回退（rollback）修改数据。在PowerBuilder中和数据库相连的控件是数据窗口（datawindow），在数据窗口中判断数据更新是否成功执行的函数是datawindowname.update({accept,resetflag})，其中可选参数accept是布尔型变量，该变量是指执行数据更新前DataWindow是否自动执行函数AcceptText（接受更新数据），缺省值为True即自动执行函数AcceptText。ResetFlag也是可选的布尔型变量，指定在DataWindow是否自动重新设置更新标志，缺省为TRUE，即每次更新均要修改数据库，当为False时是每次更新不修改数据库。因此，在PowerBuilder中实现主细表同时更新，关键是对函数datawindowname.update()作出合理设置和判断，只有这样才能保证数据更新成功的提交到数据库中。

以上面定义的产品定单为例，说明如何同时维护主细数据窗口的数据更新，窗口的界面如图主细表更新示例图。在窗口中有两个数据窗口dw_master和dw_detail，其中dw_master对应表是sales_order，dw_detail对应表sales_order_items。在窗口中包含命令按钮插入（cb_insert，主细表数据同时插入，当焦点从主表数据窗口进入细表数据窗口时，需要相应判断，当在细表中增加记录时，需判断是细表的数据窗口做数据插入），命令按钮主细表删除（cb_deleteall，主细表数据同时删除），命令按钮细表删除（cb_delete_detail，细表数据删除，但是当细表中只有一条记录时不能再删除，否则细表为空记录即数据"有头无尾"），命令按钮保存（cb_save，主细表数据同时修改并保存到数据库），命令按钮查询（cb_retrieve，主细表数据同时查询到数据窗口，数据查询时要注意会触发数据窗口中的事件rowfocuschanging），命令按钮修改（cb_ update，修改主细表中查询出来的数据）等。在数据的插入和修改可以使用同一的数据保存提交语句，数据的删除则需要改变数据窗口的修改顺序。

在保存命令按钮的clicked事件中主要有以下脚本语句完成数据窗口的数据修改：

```
...
//将主表中order_id的值赋予细表中的每个order_id，使得两者保持一致，细表中的字段order_id将属性变成隐藏
dw_detail.object.order_id [i] =dw_master.object.order_id [dw_master.getrow()]
...
//以下是数据插入和修改时，采用的数据提交的语句
if dw_master.update(true,false)=1 then// 先判断主表，后判断细表，其中dw_master.update()中的参数reset必须是false，这样保证主细表数据窗口全部修改后才反映到数据库，随后将参数reset置为true.
    if dw_detail.update(true,false)=1 then//如果sales_order和sales_order_items都更新成功，则提交该事务
        commit;
    else
        rollback;// 如果sales_order_items更新不成功，则回退该事务
    end if
else
    rollback;// 如果sales_order更新不成功，则回退该事务
end if
//以下是数据删除时，采用的数据提交的语句
if dw_detail.update(true,false)=1 then// 先判断细表，后判断主表，其余的类似数据插入时的语句
    if dw_master.update(true,false)=1 then
    ...
```

在实际的编程过程中，还需要判断焦点是否进入到dw_master或dw_detail并设置相应的全局变量，同时判断是数据插入还是数据删除，这样才能很好的控制主细表数据协调一致的提交到数据库中，避免出现数据不一致的情况发生。关于主细表中的数据插入和数据删除的语句写法可以参看PowerBuilder在线例子中的窗口w_add_ sales_order中的脚本语句，但该例子不是完整意义上的主细表数据更新。

参考文献

1 晓通数据库研究所，PowerBuilder 6.0开发手册［M］，晓通数据库研究与发展中心，1998年。

2 王珊、陈红，数据库系统原理教程［M］，清华大学出版社，1998年。