

# 用 Delphi 开发数据库的编程方法

耿静 王遵立 (中国科学院长春光学精密机械与物理研究所 130021)

**摘要:** Delphi 对关系型数据库提供全面的支持。Delphi 通过数据访问部件与数据库进行数据交换, 并提供一组数据控制部件用于显示数据库中各条记录的各个字段, 接受用户对数据的各种更新, 使用非常灵活和方便。本文主要讨论了 Delphi 中对本地数据库的编程方法, 以及编程时应注意的几个问题, 并提供了相应的解决方法。

**关键词:** 批量搬移(batchmove) 别名(alias) 独占 本地数据库 主表/明细表

## 引言

Delphi 是一门真正面向对象的开发工具, 并且完全的可视化, 由于它使用了本地编译器直接生成技术, 因此使程序的执行性能远远高出其他产品生成的程序。Pascal 语言的严谨加上可视化的优势和强大的数据库功能使 Delphi 素有 VB 杀手之称。

Delphi 对关系型数据库提供全面的支持。而关系型数据库将数据组织成相应的表, 表中的行对应着相应的每一个记录, 表中的列对应着数据库的字段, 通过使用一些简单的操作就能实现对数据的控制和管理。

## 1 Delphi 设计数据库应用程序的要点

### 1.1 使用数据库

数据库应用程序必须与数据库进行数据交换, 因此必须建立与数据库的联系。在 Delphi 中, 这个工作是通过数据访问部件(Data Access)来完成的。这些部件使用 DBE 来访问数据库, 并使其中的数据信息可以为用户接口的数据控制部件所使用。常用的数据访问部件有: TdataSource、Ttable、Tquery、Tdatabase、Tbatchmove。

BDE 中包含所支持数据库的驱动程序。这里的数据数据库种类有可以分为两种。一种是本地数据库: 如 Paradox、dBASE、FoxPro 和 Access 等, 另一种是远程数据库服务器: 如 Interbase、Oracle、Sybase、Informix、Microsoft SQL Server 和 DB2 等。

### 1.2 数据库结构

一个关系型数据库是由若干表组成。在 Delphi 中, 数据库概念对应到物理文件上是有一些不同的。对于 Paradox、dBASE、FoxPro 这三种数据库系统, 数据库对应于某一个子目录, 而其他类型如 MS Access、Btrieve 则是指某一个文件。

### 1.3 设计用户接口

Delphi 提供一组数据控制部件, 用来显示数据库中各条记录的各个字段, 接受用户对数据的各种更新。在数据控制部件页(Data Control Page)上的部件用来与用户交互, 显示、修改数据库中的数据。

本文将主要介绍用 Delphi 对本地数据库的应用编程, 选用的是 FOX 系列数据库(.dbf)。

## 2 Delphi 设计数据库应用程序的简单实例

表 Dept.dbf 是一个部门信息表, 它包括部门名称、部门编号、雇员人数、部门经理、部门预算等信息。存放在 'c:/data' 目录下。

表 Employee.dbf 是一个雇员信息表, 它包括隶属部门、雇员编号、姓名、薪水、职务名称等信息。存放在 'c:/data' 目录下。两表有一个共同的字段名: 部门名称 (Deptname)。

设计一个程序来显示 Dept.dbf 和 Employee.dbf 表中的内容, 各只需使用四个部件(两个 Table, 两个 DataSource, 两个 DBGrid, 两个 DBNavigator), 把控件放在表单 MainForm 上, 其属性设置如表 1 所示:

这样不用编写一行代码便可以构成一个简单的数据库应用程序。请试着添加、删除、修改一些记录, 在表格里都会得到及时的更新显示。

但在这个程序中存在问题:

(1) 在 Navigator1 工具条上删除按钮并不能将记录从数据库中真正删除, 而只是在记录头部作了删除标记。这对于需要经常更新的数据库来说, 表中存在大量被删除的无用记录, 是一种浪费。

(2) 当数据库表格移动路径后, 程序将出现错误。因为它在设定路径将查找不到数据集中设计的 TableName。

表 1

对象	属性	设定值	说明
Table	Name	Dept(Employee)	
	DatabaseName	C:/data	数据库文件所在路径或别名
	TableName	Dept.dbf(Employee.dbf)	实际连接的数据库表名
	Active	True	
Datasource	Name	Dsdept(Dsemployee)	
	AutoEdit	True	将数据集置于编辑状态
	Dataset	Dept(Employee)	说明部件从中获取数据的数据集的名字
DBGrid	Name	DBGrid1(DBGrid2)	可以网络形式显示数据库表中全部记录的所有字段信息的控制
	DataSource	Dsdept(Dsemployee)	数据浏览部件连接的数据源部件
DBNavigator	Name	Navigator1(Navigator2)	可以进行简单数据库操作(如: 编辑, 浏览, 插入, 删除)的控件
	DataSource	Dsdept(Dsemployee)	
	ShowHint	True	显示工具条上的提示

(3) Dept.dbf与Employee.dbf中有一个共同的字段: 部门名称(Deptname)。当Dept.dbf中的记录指针移动时, Employee.dbf中的记录并不发生相应的移动, 而当Employee.dbf中增添、删除记录时, Dept表中相应部门的雇员人数并不发生相应的改变。也就是说数据库表之间没有发生相关联系。

### 3 解决方案

针对以上问题, 我们需要一一解决。

(1) 首先需要了解数据集 Ttable 中的两个属性方法: Exclusive(独占)和 Batchmove(批量搬运)。Exclusive 属性是一个布尔型属性, 它表明是否以共享方式打开数据库表。如果 Exclusive 的值为 True, 当打开一个数据库表时, 其他用户就不能访问该表了, 即为独占。对于 SQL 数据库服务器上的数据库表, 当以独占方式被一个用户打开时, 其他用户可以读取表中的数据, 但不能修改表中的数据。Batchmove 方法可以对两个相同的数据库之间成批的记录或者整个数据库表进行移动、拷贝和删除操作。

解决不能彻底删除的方法就是: 引入一个空的临时表, 它具有与当前数据库表相同的结构, 然后把当前数据库表里的有效记录批量搬到临时表里, 把当前表清空, 再把有效记录从临时表中批量搬移回来, 这样, 我们就把标记成删除的记录真正删除了。需要注意的是, 在进行批量搬移前, 必须将数据集设为独占的, 在每次使用完临时表后, 应及时清空。临时表的生成可以使用 FOX 语言里的 Copy Structure to 命令, 并给它起一个特殊的文件名, 如: Temp.dbf。

(2) Ttable 中的属性 DatabaseName 时说明数据库应用程序的数据库的名字, 它可以是由 BDE 定义的数据库别名、显示说明的数据库文件所在的路径或者是由

Tdatabase 部件定义的一个数据库名。在这里, 建议用户选用由 BDE 定义的数据库别名。使用别名代替数据库实际所在路径和名字, 好处是当实际的数据库存放的位置发生变化时, 只需利用 BDE 简单的设置一下该数据库的别名, 而数据库应用程序无须修改。数据库别名的创建、修改可以通过数据库资源管理器(Database Explorer)来完成。另外还有一个窍门: 当数据库应用程序所用的数据库表与该应用程序处在同一路径下时, 可以缺省将 DatabaseName 属性设为空, 这样按 Delphi 的缺省查找路径, 它将首先查找应用程序当前所在路径下的数据库表。

(3) 如果想浏览多个相关的数据库表中的记录信息, 就必须创建具有多个表的数据库应用程序。在多表类型数据库应用程序中: 一个数据库表作为主表, 其中存放着综合信息; 其他的数据库表和主要数据库表相关联, 他们当中存放着更详细的信息, 称为明细表。在本例中, 我们将 Dept.dbf 作为主表, 将 Employee.dbf 作为明细表, 利用其中的字段 Deptname 把两个表联系起来, 这样当察看某个部门的信息时, 明细表将会显示该部门雇员的详细信息, 并可以通过编程实现当明细表添加或删除记录时, 修改相应主表的雇员人数(count)。

### 4 编程实现

在刚才的简单例子中, 添加一个 Ttable 对象存放 Temp.dbf, 并命名为 Temp, 添加三个命令按钮: Delbtn(删除主表记录)Appendbtn(添加明细表记录) Delbtn1(删除明细表记录)。删除 Navigator2, 修改、添加新老属性如表 2 所示:

注意: 在设置 Exclusive 之前, 请先将其数据集的 Active 设为 False, 等设置 Exclusive 为 True 后, 再将其 Active 设为 true。

表 2

部件	属性	属性值
Dept	Exclusive	true
Temp	Active	true
	Exclusive	true
	Tablename	Temp.dbf
Employee	IndexFieldNames	Deptname
	MasterField	Deptname
	MasterSource	DSDept

各事件处理函数如下:

```

procedure TMainForm.delbtnClick(Sender:TObject);//
删除主表记录
var currec:integer;
begin
    if messagedlg('是否确定删除该记录?',
    mtinformation,mbOkCancel,0)=mrok then
    begin
        Temp.EmptyTable; // 清空临时库表
        with Dept do
        begin
            currec:=RecNo; // 得到当前指针所指记录值, 以便
            恢复
            Delete; // 删除当前指针所指记录
            DisableControls; // 在修改记录的过程中使其他部件
            无效
            Temp.BatchMove(Dept,batCopy); // 把当前数据表里的
            有效记录批量搬移到临时表里
            Emptytable; // 把当前表清空
            BatchMove(Temp,batcopy); // 把有效记录从临时表
            中批量搬移回来
            first; // 恢复指针
            moveby(currec-1);
            EnableControls; // 恢复其他部件功能
            end;
        end;
    end;
end;
procedure TMainForm.appendbtnClick(Sender:TObject); //
添加明细表记录
var
    deptname:string;
    count:integer;
begin

```

```

    deptname:=Dept.FieldByName('Deptname').Asstring; //得
    到当前主表指针所指的部门名称
    Employee.Appendrecord([deptname,nil,nil,nil,nil]); // 添
    加该部门的一个新雇员
    with Dept do
    begin
        Edit; // 修改主表记录
        count:=FieldByName('Count').Asinteger; //得到该部
        门雇员人数
        FieldByName('Count').Asinteger:=count+1; //修改该部
        门雇员人数
        Post; // 保存修改
    end;
end;
procedure TMainForm.Delbtnl Click(Sender:TObject); // 删
    除明细表记录
var
    count:integer;
begin
    if messagedlg('是否确定删除该记录?',
    mtinformation,mbOkCancel,0)=mrok then
    begin
        Employee.Delete; //这里只简单作了删除标志, 如要彻
        底删除, 请参照 DelbtnClick
        with Dept do
        begin
            Edit; // 修改主表记录
            count:=FieldByName('Count').Asinteger; // 得到该
            部门雇员人数
            FieldByName('Count').Asinteger:=count-1; // 修改该
            部门雇员人数
            Post; // 保存修改
        end;
    end;
end;
end;

```

参考文献

- 1 孙琨, 曹勇, 利用 CGI 技术实现 WEB 服务器与 DBF 数据库文件的连接, 计算机系统应用, 1998, 9(9): 39-42
- 2 刘韬, 肖永顺, 王宇等, Delphi 4.0 数据库编程, 人民邮电出版社, 1999.