第十一讲 Delphi数据库技术

吕松茂

lsmtech@163.com

内容提要

- Delphi数据库组件
- TField对象的使用
- 数据集的操作

内容提要

- Delphi数据库组件
  - TField对象的使用
  - 数据集的操作

一、 Delphi数据库组件

Delphi使用VCL将BDE封装成组件，为数据库应用程序提供统一的访问接口，组件面板的BDE页、Data Access页和Data Controls页提供了不同的数据库组件。其中，BDE页和Data Access页的组件为非可视组件，Data Controls页的组件为可视组件。

一、 Delphi数据库组件

- BDE组件集（BDE页）：提供了以BDE方式访问数据库的数据表和查询等数据集组件，如TTable、TQuery、TStoredProc等；
- 数据访问组件集（DataAccess页）：提供了数据源等连接组件；
- 数据控制组件集（DataControls页）：用来浏览和编辑数据，为用户使用数据库提供接口，如TDBGrid、TDBEdit、TDBMemo等。

一、 Delphi数据库组件

1.1 TTable组件

❁TTable组件用于连接一个数据表并对数据表的各种状态进行控制，它通过BDE从一个数据库表格中取得数据，并通过TDataSource组件将数据传递给一个或多个数据控制组件，或者反之，从数据控制组件得到的信息通过BDE传递给数据库。

❁TTable组件既可以访问本地的数据库，也可以访问ODBC数据库，还可以访问远程数据库。

一、 Delphi数据库组件

TTable组件连接数据表的一般步骤

- 步骤一：把一个TTable组件放到窗体上，设置DatabaseName属性指定要访问的数据库。
- 步骤二：设置TableName属性指定要访问的表。
- 步骤三：把一个TDataSource组件放到窗体上，设置DataSet属性指向该TTable组件。
- 步骤四：把一个数据控制组件放到窗体上，设置DataSource属性指向该DataSource组件。
- 步骤五：把TTable组件的Active属性设为True。

一、 Delphi数据库组件

字段组件的访问

字段组件对应着数据表中实际的字段，读写数据表中的字段值是通过访问相应的字段组件进行的。一般采用的方法有两种：

❁使用数据集组件的Fields属性

可以通过Fields属性的下标（即索引号）来访问各字段，索引号从0开始。例如，Table1.Fields[0]表示当前记录的第一个字段的数据。如：

```
Table1.Fields[0].AsString:=Edit1.Text;
Edit1.Text:=Table1.Fields[0].AsString;
```

一、 Delphi数据库组件

- 使用数据集组件的FieldByName方法

用这种方法访问字段组件时，必须知道数据表中各个字段的名字。

例如：

```
Edit1.Text:=Table1.FieldByName
                ('Name').AsString;
```

一、 Delphi数据库组件

记录的增加与删除

- Insert方法：在当前记录之前插入一条新的记录。
- Append方法：在文件的最后追加一条新的记录。
- Edit方法：修改当前记录。
- Delete方法：删除数据表中的当前记录。
- Post方法：当数据集处于编辑状态时，将当前记录的修改写回数据库表。

一、 Delphi数据库组件

1.2 TDataSource组件

TDataSource组件是联系数据集组件与数据控制组件的桥梁。其常用属性如下：

❁AutoEdit属性：该属性值是一个布尔值，用于说明是否将数据集置于编辑状态，为Ture时允许用户编辑数据集中的数据。

❁DataSet属性：指明与当前数据源组件相联系的数据集组件对象的名字。

❁Enabled属性：该属性值是一个布尔值，决定了与此数据源组件相连的数据控制组件是否显示数据。为True（默认值）时，数据控制组件将显示数据。

一、 Delphi数据库组件

1.3 数据控制组件

- 数据控制组件也称为数据感知组件，它应用程序的接口，让用户能浏览和操作数据库。
- 数据控制组件在Data Controls页上。
- 数据控制组件与数据集组件之间的联系是通过数据源组件TDataSource来实现的。

一、 Delphi数据库组件

数据控制组件常用属性：

- DataSource：将此属性设置为窗体上数据源组件的名称，建立数据控制组件与数据源组件之间的联系。
- DataField：用来确定访问的是数据集中的哪一字段。TDBGrid组件和TDBNavigator组件访问数据集中所有的字段，所以没有这一属性。
- Enabled：决定数据控制组件能否接受来自鼠标、键盘和定时器事件的消息。
- ReadOnly：确定是否可以在数据浏览中编辑修改字段的值。默认值为False，也就是可以在其中编辑修改字段的值，反之，则不可以。

一、 Delphi数据库组件

TDBGrid组件

- TDBGrid组件的作用是一个将数据集记录显示在网格中，并且可以对其中的数据进行编辑修改。
- Columns属性。指定TDBGrid对象中各栏目的特性，如栏目的标题、栏目的宽度及颜色、在该栏目中显示的字段的名称、数据在栏目中的位置方式等。

一、 Delphi数据库组件

TDBNavigator组件

TDBNavigator组件是用来在数据集中浏览数据和编辑数据的，它由一组控制按钮组成，通过这些控制按钮，用户可以完成在数据集中移动记录指针，插入、删除、编辑、确认、取消、刷新记录数据等。它一般与其他数据控制组件（如TDBGrid或TDBEdit）一起使用。

一、 Delphi数据库组件

- 设置按钮

TDBNavigator组件中提供了10个按钮，设置各按钮的VisibleButtons属性可以控制按钮的显示与否。

- 设置按钮提示字符串

TDBNavigator组件的Hints属性和ShowHint属性就用来设置和显示提示字符串。

- 设置按钮的操作方式

CanfirmDelete属性。该属性值是一个布尔值，用于控制执行删除操作时是否弹出确认对话框缺省值为True。

一、 Delphi数据库组件

TDBText组件

- TDBText组件是用来显示数据集里当前记录中一个特定字段的值，随着记录指针的移动，其显示的内容也不断变化，这是它与TLabel组件不同的地方。用TDBText组件显示的字段值是只读属性的。如果用户要修改数据，就需要使用TDBEdit组件或者TDBMemo组件。
- TDBText组件的重要属性有DataSource和DataField属性。

一、 Delphi数据库组件

TDBEdit组件

TDBEdit组件用来显示和编辑数据集里当前记录中一个特定字段的值，随着记录指针的移动，其显示的内容也不断变化。

- 常用属性有DataSource、 DataField、Enabled和ReadOnly。

一、 Delphi数据库组件

TDBMemo组件

- TDBMemo组件用来显示和编辑数据集中的多行文本，
- TDBMemo的Text属性代表了该字段的内容。

TDBMemo组件除了具有其他数据控制组件所具有常用属性以外，还具有以下不同的属性：

- AutoDisplay属性：决定是否自动显示数据集中备注类型的字段值。
- Alignment属性：说明文本的对齐方式。
- MaxLength属性：限定可以输入的最大字符数。值为0时表明不限制最大字符数。
- WordWrap属性：指明文本是否自动换行。

一、 Delphi数据库组件

TDBCheckBox组件

- TDBCheckBox组件是允许用户选择或不选择一个值的数据控制组件，TDBCheckBox组件适合于表达布尔类型数据的字段。
- TDBCheckBox组件除了完成CheckBox组件的功能外，主要是用来显示和修改一个布尔类型数据的字段值。

一、 Delphi数据库组件

TDBListBox组件

TDBListBox组件用来提供给用户一个可选的列表，用户可以从中选取适当的值来修改当前记录中的特定字段值。它与TListBox组件类似。

一、 Delphi数据库组件

TDBComboBox组件

TDBComboBox组件与TDBListBox组件的功能相似，其用法与TDBListBox组件相应的属性一样。

TDBRadioGroup组件

TDBRadioGroup组件是允许用户在一组选项中选取唯一一项的数据控制组件，它必须指向数据集中一个特定的字段。TDBRadioGroup组件与普通单选框用法相似。

一、 Delphi数据库组件

TDBImage组件

TDBImage组件用来显示和编辑当前记录中BLOB类型的图形字段。该组件除了可以在窗体中显示数据集里的图形数据外，还允许通过剪贴板对图形数据进行编辑操作。其特殊的属性有：

- Picture属性：保存DBImage对象中的图形数据。
- Center属性：控制是否将图形自动显示在对象的中心。该属性默认值为False，图形显示在DBImage对象的左上角。

内容提要

- Delphi数据库组件
- **TField对象的使用**
- 数据集的操作

二、 TField对象的使用

- 动态字段对象和永久字段对象
- 设置字段属性
- 对字段进行格式化
- 字段的有效性检查
- 创建查找字段
- 创建计算字段

二、 TField对象的使用

2.1 动态字段对象和永久字段对象

- 用TField字段对象可以方便地访问数据集中的字段。当把一个数据集放到窗体上并且打开它时，Delphi会为数据集中的每一个字段自动生成一个动态的字段对象。当数据集关闭时，这些字段对象也跟着消失。
- 动态字段对象的最大特点是适应性强，动态字段对象的不足之处是：要想改变字段的显示属性、数据格式，就要编写代码。不能把某些字段暂时隐去，也不能增加新的字段。

二、 TField对象的使用

- 永久字段对象的最大好处是可以在设计时设置它的属性。此外，永久字段对象还具有以下优势：选择部分字段；增加新的字段，包括“计算字段”等；在永久字段对象列表中删除某些需要保护的字段，避免用户访问这些特定的字段；在数据库查询或特定数据表的字段基础上定义新字段，代替现存的字段；改变原有字段的显示和编辑属性。
- 对于同一个数据集来说，动态字段对象和永久字段对象不能同时存在。

二、 TField对象的使用

2.2 设置字段属性

为数据集创建永久TField对象步骤如下：

- 步骤一：设置一个数据集组件（如TTable）和数据表关联；
- 步骤二：双击TTable组件，或右击Ttable组件，选择“Fileds Editor”菜单命令，将打开字段编辑器；
- 步骤三：右击字段编辑器窗口的空白处，弹出快捷菜单，单击“Add Fields”菜单命令，打开“Add Fields”对话框；
- 步骤四：选择一个或几个字段，单击OK按钮。Delphi将根据选择的字段创建永久字段对象。

二、 TField对象的使用

TField对象常用属性 ：

- Alignment：设置字段在数据组件中的对齐方式
- Currency：仅用于数字字段，True表示按货币格式显示
- DisplayFormat：设置字段在数据组件中的显示格式
- DisplayLabel：设置字段在数据表格中的列标签
- EditFormat：设置字段在编辑时的显示格式
- EditMask：设置用户编辑数据时必须遵守的规则
- FieldKind：指定字段的生成类型
- FieldName：指定字段的名称

二、 TField对象的使用

TField对象常用属性 ：

- Index：指定字段在数据集中的序号
- Max Value：指定字段的最大值
- Min Value：指定字段的最小值
- Name：指定永久字段对象的内部名称。
- Size：指定字段的长度（以字符为各单位）

二、 TField对象的使用

2.3 对字段进行格式化

操作步骤如下：

❁步骤一：双击TTable组件，在字段属性编辑器中添加字段后，选定BIRTHDAY字段。

❁步骤二：在对象浏览器（Object Inspector）窗口中双击“EditMask”属性，将打开“Input Mask Editor”对话框。

❁步骤三：为选定的字段设置掩模码并测试，然后单击“OK”按钮关闭对话框。

二、 TField对象的使用

2.4 字段的有效性检查

字段的有效性可以保证字段中的数据与预定义的数据格式、限定范围、条件等相符。可以使用TField对象的Validation事件来控制字段的有效性规则。

二、 TField对象的使用

2.5 创建查找字段

使用查找字段可以在表格中添加一个下拉列表框，使得用户可以在数据集对象设定的一些选项中选择，而不必盲目地用键盘输入数据。用这种方法可以确保用户输入的总是有效数据，从而不必总是要进行有效性检查。

二、 TField对象的使用

2.6 创建计算字段

Delphi的TField对象还允许创建另一种特殊的字段——计算字段。计算字段是根据数据表中的字段数据计算出来的，用户可以通过计算字段很直观地看到几个字段的计算结果，但数据表不会将计算字段的数据存入内存，因此计算字段不会被存储在数据表中，它只是用来显示某些计算结果的。

内容提要

- Delphi数据库组件
- TField对象的使用
- 数据集的操作

三、 数据集的操作

- 数据集的打开和关闭
- 数据集的状态
- 移动记录指针
- 限制记录集
- 查找记录
- 数据集的修改
- 建立数据表的主从关系

三、 数据集的操作

3.1 数据集的打开和关闭

1. 设置Active属性

Active属性用来说明数据表文件的打开状态。通过设置Active属性来决定一个数据集组件与数据表数据之间的联系。

2. 调用Open和Close方法

使用Open和Close方法也能打开和关闭一个数据集。

3.2 数据集的状态

数据集的状态（State属性）决定了当前能够对数据集进行的操作。在程序运行期间，可以通过检测数据集的只读属性State来确定其当前状态。

三、 数据集的操作

3.3 移动记录指针

- 在对数据表操作时，系统使用一个记录指针指向当前正在访问的记录。浏览数据表就是通过移动记录指针来查看数据表中的记录。
- Bof属性：Bof属性为布尔值。为True表明当前记录指针所处的位置为数据集的第一条记录；
- Eof属性：Eof属性为布尔值。为True表明当前记录指针所处的位置为数据集的最后一条记录；
- First方法：将记录指针移至数据集的第一条记录处，并使之成为当前记录，同时将Bof属性值设置为True。

三、 数据集的操作

- Last方法：将记录指针移至数据集的最后一条记录处，并使之成为当前记录，同时将Eof属性值设置为True。
- Next方法：将记录指针后移一条记录，并使之成为当前记录。如果记录指针指向了数据集的最后一条记录，将Eof属性设置为True。
- Prior方法：将记录指针前移一条记录，并使之成为当前记录。如果记录指针指向了数据集的第一条记录，将Bof属性设置为True。
- MoveBy方法：将记录指针从当前记录开始向后或向前移动若干条记录。格式如下：

```
function MoveBy(Distance:Integer):Integer;
```

3.4 限制记录集

Delphi提供了以下两种常用的限制记录集的方法。

1. SetRangeStart方法和SetRangeEnd方法

❁SetRangeStart方法和SetRangeEnd方法可以过滤记录，SetRangeStart方法用来限制记录集的开始，SetRangeEnd方法用来限制记录集的结束，调用ApplyRange方法使限制生效。调用CancelRange方法取消为数据表设定的限制范围。这种限制记录集的方法要求指定数据集的检索字段。

三、 数据集的操作

2. 使用数据集的Filter属性

Delphi提供了一种更简便的限制记录集的方法：用数据集的Filter属性。Filter属性值是一个用来指明数据表过滤标准的字符串，Filtered属性决定了过滤器是否起作用。当使用Filter属性时，不必指定数据集的检索字段，各字段或字段的组合都可以运用该属性。

3.5 查找记录

1. 使用FindKey方法

使用TTable的FindKey方法或FindNearest方法，可以查找数据表中满足条件的记录，要查找的表必须按查找关键字建立了索引。其定义如下：

```
function FindKey(Const KeyValues:
  Array of Const):Boolean;
    procedure FindNearest(Const KeyValues:
      Array of Const);
```

三、 数据集的操作

2.搜索特定记录（Locate）

FindKey和FindNearest方法只能在TTable组件中使用，如果使用的是TQuery或TStoreProc组件，就要使用Locate方法来查找记录。

Locate函数适合于所有数据集组件，也适合于TTable组件。Locate函数可以在任何表中基于任何类型的字段来查找记录，而不管表格是否建立了索引。Locate函数定义如下：

```
function Locate(const KeyFields:string;
    Const KeyValues:Variant;Options):Boolean;
```

三、 数据集的操作

3.6 数据集的修改

与数据集修改有关的属性和方法：

- ReadOnly属性：确定相应的数据集是否为只读方式。True表示只读，不能修改。
- Exclusive属性：确定相应的数据集是否为独占方式。如果设定为True，则其他程序不能使用该数据集，否则可同时使用。
- CanModify属性：确定相应的数据集是否可以改变。当以只读方式打开数据集时，该属性的返回值为False。
- Modified属性：标记相应的数据集是否已被修改过。当其值为True时，表示已被修改过。

三、 数据集的操作

- Delete方法：删除数据集中的当前记录，并使当前记录指向下一条记录。
- Edit方法：使数据集处于编辑状态，此时才可对数据集进行编辑修改操作。
- Insert方法：在当前记录之前插入一个新的空记录，并将新记录作为当前记录。用户向记录的字段中输入数据后，通过调用Post方法将这些更改提交给数据库。
- Post方法：使当前对数据集所作的修改有效，即对数据集所作的修改写入到数据库中。调用了Edit、Insert或Append等方法后，需要调用Post方法将修改的数据写回数据库。
- Cancel方法：使当前对数据集所作的修改无效，即取消对数据集所作的修改。

三、 数据集的操作

❁AppendRecord方法与InsertRecord方法：这两个方法分别与Append方法和Insert方法相似，它们都是用于在表中插入一条新记录，但AppendRecord方法和InsertRecord方法比Append方法和Insert方法更简单、更方便一些，不需调用Post方法。其过程形式如下：

procedure AppendRecord(Const Values: Array of Const);

procedure InsertRecord(Const Values: Array of Const);

三、 数据集的操作

❁SetFields方法：以参数的形式来修改当前记录中的多个字段值。其过程形式如下：

procedure SetFields(Const Values: Array of Const)

❁Refresh方法：从数据库中取出数据来更新数据集的内容。它保证了应用程序拥有数据库的最新数据。

3.7 建立数据表的主从关系

❁设置表之间的主从关系是通过设置从表的MasterSource属性和MasterField属性来实现的。且从表必须是按MasterField属性中指定的字段建立了索引的。

总结

- Delphi数据库组件
- TField对象的使用
- 数据集的操作

谢谢大家