

计算机高级编程技术



第九章 利用窗体维护数据





第9章 利用窗体维护数据

- 窗体（Form）是建立数据库应用程序必不可少的工具，可以将数据和窗体连接起来，可以利用窗体建立友好的界面。
 - 9.1 窗体对象
 - 9.2 利用窗体进行数据的操作
 - 9.3 字段对象
 - 9.4 使用多个数据表格
 - 9.5 使用Form Wizard





9.1 窗体对象

- Delphi的应用程序是从建立窗体开始的，因此熟练地掌握窗体的重要属性、方法和事件，对于开发数据库应用程序是非常重要的。
 - 9.1.1 基本属性
 - 9.1.2 窗体的常用事件和方法





9.1.1 基本属性

- 1. BorderStyle
 - BorderStyle属性控制窗体边框类型，是一个集合类型的属性。
- 2. BorderIcon
 - BorderIcon是一个属性的集合，这个属性是用来说明是否显示窗体顶部的控制菜单。
- 3. Ctl3D
 - 该属性是用来确定窗体是否具有立体效果。





9.1.1 基本属性

- 4. Caption
 - 该属性用来确定窗体的标题。
- 5. Name
 - 该属性为窗体指定了在程序中的引用名。
- 6. FormStyle
 - 该属性确定窗体的类型。





9.1.1 基本属性

7. Enable

- 该属性确定窗体是否可用，若其值为**False**，则不能用鼠标和键盘对它操作。

8. WindowState

- 该属性确定窗体所处的状态。

9. Cursor

- 该属性确定鼠标在窗体上移动时的光标形状。





9.1.1 基本属性

□ 10. Font

- 该属性用来描述显示文本中的字符对象，它是一个TFont类型的属性，在设计或运行时设定对象的属性可以决定窗体上显示的字体大小及颜色等。

□ 11. ActiveControl

- 该属性确定窗体上的哪个对象取得了焦点及是否触发键盘事件。





9.1.1 基本属性

- 12. KeyPreview
 - 设置为“是”，当窗体的任何对象有键盘输入时，都先触发窗体的键盘事件，然后再触发该对象的键盘事件。
- 13. Icon
 - 该属性确定窗口最小化时所显示的图标。
- 14. Handle
 - 该属性是只读属性，只有在运行时可以使用。





9.1.1 基本属性

□ 15. Hint和ShowHint

- ShowHint属性用于启动提示功能。Hint属性是用来设置提示信息文本的。





9.1.2 窗体的常用事件和方法

- 在数据库应用程序中常用的方法和事件是完成窗体的建立、显示和关闭功能的方法和事件。

- 1. Create方法和OnCreate事件
 - 调用Create方法可以建立并初始化一个窗体。
 - 调用Create方法可以触发OnCreate事件，在窗体建立时要实现某些操作，可以在该事件处理程序中进行。





9.1.2 窗体的常用事件和方法

- 2. Show和ShowModal方法与Onshow事件
 - Show方法用于显示窗体， ShowModal用于显示模态窗体，这两种方法都可以触发OnShow事件。

- 3. Close方法和OnClose事件
 - Close方法可以关闭窗体，同时触发OnClose事件。





9.2 利用窗体进行数据的操作

- 通过窗体可以对数据库的数据进行一些基本的操作，例如数据的排序，数据的查询，数据的修改等。
 - 9.2.1 数据的排序
 - 9.2.2 数据的查询
 - 9.2.3 对数据库中的数据进行修改
 - 9.2.4 数据的插入和删除





9.2.1 数据的排序

- ❑ 数据库中有大量的数据信息，通过设计表格可以浏览这些信息，在显示这些信息的时候，我们可以把数据以不同的顺序排列出来。
- ❑ 数据的排序要用到TTable组件的IndexFields属性，通过指定属性的值，就可以将信息以不同的排列顺序显示出来。





9.2.1 数据的排序

- 【例9-1】例子。
 - 建立一个新工程，修改窗体Caption的属性为Index，并将Name属性修改为FrmIndex。选取一个Table和一个DataSource放入窗体中，再选取DBGrid放入窗体中。
 - 向窗体中加入三个命令按钮，给这三个命令按钮的Caption属性和Name属性分别设置为City、CustNo、Name和CmdCity、CmdNo、CmdName。





9.2.1 数据的排序

属性	取值	属性	取值
Table1.Active	True	DataSource1.Data Set	Table1
Table1.DatabaseName	DBDEMOS	DBGrid1.DataSource	DataSource1
Table1.TableName	Custoly.db		





9.2.1 数据的排序

```
procedure TFrmIndex.CmdCityClick(Sender: TObject);  
begin  
    Table1.IndexFieldNames:='City'; {以City为索引}  
end;  
procedure TFrmIndex.CmdNoClick(Sender: TObject);  
begin  
    Table1.IndexFieldNames:='CustNo'; {以CustNo为索引}  
end;  
procedure TFrmIndex.CmdNameClick(Sender: TObject);  
begin  
    Table1.IndexFieldNames:=  
    'Last_Name;First_Name'; {以Name为索引}  
end;  
end.
```





9.2.1 数据的排序

The screenshot shows a window titled "Index" containing a table with three columns: CustNo, Last_Name, and First_Name. The table lists 11 customer records, sorted by CustNo. Below the table are three buttons labeled "City", "CustNo", and "Name". The "CustNo" button is highlighted with a dashed border, indicating it is the active sorting criterion.

CustNo	Last_Name	First_Name
1	Ferguson	Marilyn
2	Montegiordano	Donatella
3	Constantinopolous	Marulla
4	Parker	Geoffrina
5	Sawyer	Thomas L.
6	Blue	Jack M.
7	DuDiver	Madelline
8	O'Parr	Ruth
9	Delgado De Cabeza	Enrique
10	Marmot	Herbert A.
11	Charstein	Peter & Francoise





9.2.2 数据的查询

- 数据的查询是数据库应用程序的一项重要功能。要建立程序功能，首先要确定查询的字段，一般都是数据表格的主关键字段或辅助索引。
- 在程序中可以通过下面几种方法移动记录指针。





9.2.2 数据的查询

方法	内容	方法	内容
First	移至数据表的第一个记录	Next	移至下一个记录
Last	移至数据表的最后一个记录	MoveBy	移动指定数目的记录
Prior	移至上一个记录		





9.2.3 对数据库中的数据进行修改

- 用户可以对数据库中的数据进行修改。修改数据首先要将TTable组件设置为编辑状态，用Edit方法实现。要实现对实际数据库中的数据修改，就必须调用Post方法将修改的结果写入数据的数据库中。





9.2.4 数据的插入和删除

- 在数据库工作平台可以对数据库的记录进行插入和删除，但在实际中，应该建立一个友好的界面根据用户的最终命令来完成这类操作。
- Delphi提供了Insert和Append两种方法。Insert方法是在当前记录的指针处插入记录，Append方法是在数据库表格的结尾处。使用Insert方法的基本程序段如下：





9.2.4 数据的插入和删除

```
begin  
  Insert; //插入空记录  
  {依次添加数据;}  
  Post; //将添加的记录传送给数据库  
end;
```





9.2.4 数据的插入和删除

- 可以使用InsertRecord方法一次来完成上述功能，InsertRecord方法把记录的各个字段合成一个字段数组作为它的参数。
- 调用Delete方法可以删除记录。记录删除后，该记录下面的记录依次向前移动，记录指针指到下一个记录。通常在删除记录之前，会弹出一个对话框，确认是否删除。





9.3 字段对象

- 字段对象是对象的一种，在数据库编程时用得很多，使用也很灵活。
- TFields对象是在打开数据库时动态产生的，在数据库关闭时自动消失。通过字段编辑器可以记录永久字段对象列表，创建后的字段对象保存到应用程序中。





9.3 字段对象

- 9.3.1 字段对象的数据类型
- 9.3.2 创建永久字段对象
- 9.3.3 字段对象的属性和方法
- 9.3.4 字段对象的使用





9.3.1 字段对象的数据类型

- 字段对象对应着数据表格中的字段，字段对象可以在应用程序运行的过程中动态地产生，也可以利用字段编辑器创建成为永久字段。
- 由于字段对象对应数据表格中的字段，数据表格中的字段的数据类型比较多，所以字段对象的数据类型也比较多。





9.3.1 字段对象的数据类型

字段对象的数据类型	适用情况
TBLOB	对应大二进制的对象
TbooleanField	对应布尔类型数据，可以将其他数据类型转化为该类型的数据
TcurrentyField	对应通用类型的数据，适用于储存大的数字
TintegerField	对应整数类型的数据，可以将其他类型的数据转化为该数据类型
TstringField	对应字符串类型的数据，可以将其他类型的数据转化为该数据类型





9.3.2 创建永久字段对象

- 使用字段编辑器可以创建永久性字段对象。
- 下面通过一个例子来演示如何创建永久性字段对象。
 - ① 创建一个基于**Animals.dbf**的数据库应用程序。
 - ② 选择**Table1**组件，双击鼠标左键，可以打开字段对象编辑器。
 - ③ 单击鼠标右键，在弹出的对话框中选择**Add Fields**选项，从中选择需要的字段，加入到字段编辑器中。
 - ④ 单击**OK**按钮，结束创建永久字段对象的过程。





9.3.2 创建永久字段对象

- 在创建完永久字段对象后，在Object Inspector窗口中就可以看到Table1Name、Table1Aera等对象。
- 发现有不需要的字段对象想删除它，可以在对象编辑器中选择需要删除的字段对象，然后单击鼠标右键，选择Delete选项，就可以删除该字段对象了。





9.3.3 字段对象的属性和方法

属性	含义
Alignment	说明字段值的显示方式
Calculated	说明该字段是否可以由其他字段的值计算得来，设置为true时，显示数据库表格中字段的值
DisplayLabel	说明字段在网格的顶部显示的标题，默认时显示该字段名
DisplayWidth	说明字段在网格的顶部显示的列宽度
DisplayFormatandEditMask	说明字段在显示和编辑状态时的显示格式和过滤条件
FieldName	说明在数据表格中对应该字段对象的字段名
Index	指定该对象在数据集中的逻辑位置
Name	说明字段对象的名称，以在程序中引用字段对象
ReadOnly	说明字段是否能够被修改
Visible	在网格中是否显示该字段





9.3.3 字段对象的属性和方法

事件	内容
OnChange	当字段对象对应的字段的值改变时，触发该事件
OnGetText	当字段对象获得字段的值时，触发该事件
OnSetText	当字段对象被设置字段值时，触发该事件
OnValidata	当字段值被修改或插入新的字段值时，对字段进行有效性检验时，触发该事件





9.3.4 字段对象的使用

- 字段对象有动态生成的，也有在字段编辑器中创建的永久性字段对象，它们拥有自己的属性、方法和事件，我们可以在程序中访问和控制字段对象。
- 由于动态字段对象没有自己的名称，而永久性字段对象有自己的名称，所以两者在程序中的使用方法是不同的，下面我们就分别介绍一下这两个字段对象的用法。





9.3.4 字段对象的使用

- 1. 动态字段对象的使用
 - 动态字段对象是TTable组件和Query的一部分，每个动态字段对象都分别对应着数据表格中的字段。TTable或Query组件中的所有字段对象都存在于Fields属性中，Fields中的字段对象分别对应着一个索引号。





9.3.4 字段对象的使用

- 2. 永久性字段对象的访问
 - 永久性字段对象的访问同其他类型对象的访问基本一样，在程序中可以直接使用字段对象的属性进行访问。

- 3. 字段对象的输出
 - 使用字段对象的Value属性，可以将字段的值输出到需要的地方。





9.3.4 字段对象的使用

□ 4. 计算字段对象

- 有时数据表格中的现有字段不能满足用户的需求，需要增加一些数据表格中没有的字段，这些字段可以根据数据表格中的字段计算出来。





9.4 使用多个数据表格

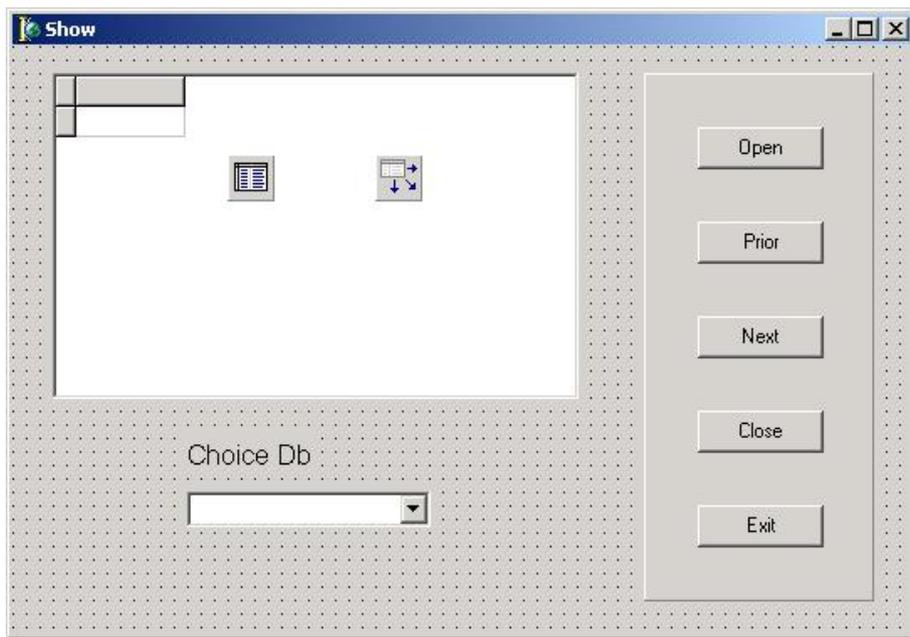
- 前面讲的内容都是使用一个数据表格，我们还可以在一个窗体中浏览多个数据库表格。

- **【例9-6】**下面通过一个实例显示如何创建一个简单的浏览多个数据表格的窗体。
 - 在图所示的对话框中选择要打开的数据表格，右边的命令按钮分别是浏览记录和重新选择数据表格。





9.4 使用多个数据表格





【例9-6】

- 程序的代码如下：

```
procedure TFrmshow.CmdOpenClick(Sender:TObject);  
begin  
    Table1.Close;{关闭打开的表格}  
    Table1.TableName:=CmbChoice.Text;{选择要打开的表格}  
    Table1.Open;{打开表格}  
end;  
procedure TFrmshow.CmdPriorClick(Sender:TObject);  
begin  
    Table1.Prior;  
end;  
procedure TFrmshow.CmdNextClick(Sender:TObject);  
begin
```





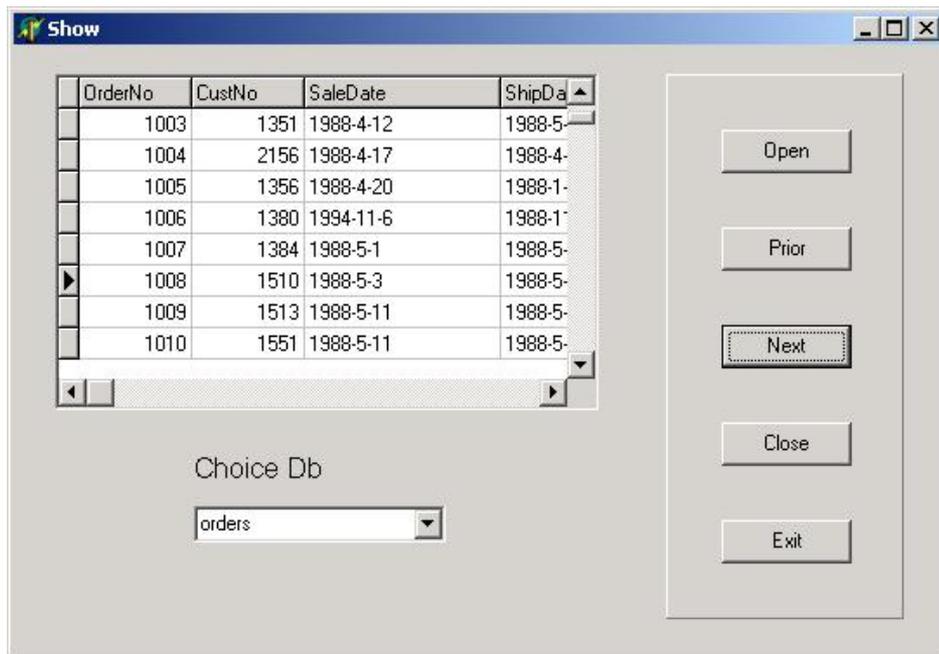
【例9-6】

```
Table1.Next;  
end;  
procedure TFrmshow.CmdCloseClick(Sender:TObject);  
begin  
    CmbChoice.Text:="";  
    Table1.Close;  
end;  
procedure TFrmshow.CmdExitClick(Sender:TObject);  
begin  
    Application.Terminate;  
end;
```





【例9-6】





9.5 使用Form Wizard

- Form Wizard是Delphi提供的快速创建数据库应用程序的工具，利用这个工具可以快速、方便地创建简单的数据库应用程序。使用可以创建单个数据表格的应用程序，也可以创建一对多的应用程序。



9.5.1 使用Form Wizard创建单个数据表格的应用程序



- ① 在主菜单中选择Database选项中的FormWizard选项。
 - 在Form Options框中，选择Create a simple form表示创建单个数据表格的应用程序。在DataSetOptions框中，选择Create a form using TTable objects表示选择TTable对象。



9.5.1 使用Form Wizard创建单个数据表格的应用程序



Database Form Wizard

What type of form do you want the wizard to create?

Row	123
9666	234
Con	345
0444	456
Erre	567
Row	678
9666	789

Form Options

- Create a simple form
- Create a master/detail form

DataSet Options

- Create a form using ITable objects
- Create a form using TQuery objects

Help < Back Next > Cancel



9.5.1 使用Form Wizard创建单个数据表格的应用程序



- ② 选择要创建应用程序的数据表格。
 - 单击**Next**按钮，弹出选择数据表格的窗口。选择C:\Program Files\Common Files\Borland Shared\Data\Customer.db作为应用程序的数据库表格，在该数据库表格的基础上创建应用程序。





9.5.1 使用Form Wizard创建单个数据表格的应用程序



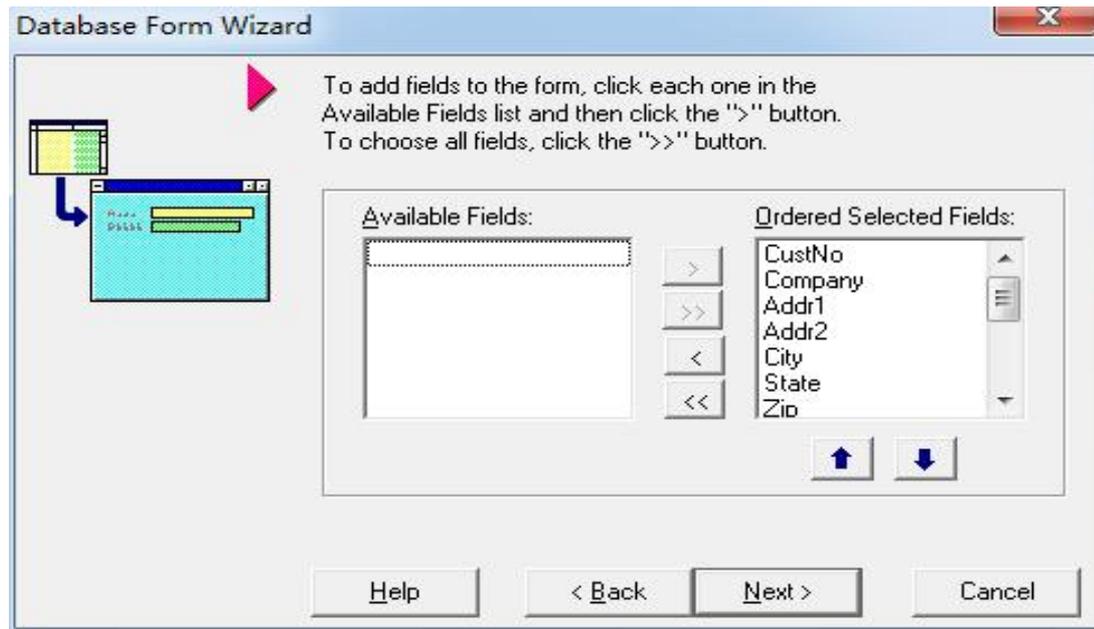
9.5.1 使用Form Wizard创建单个数据表格的应用程序



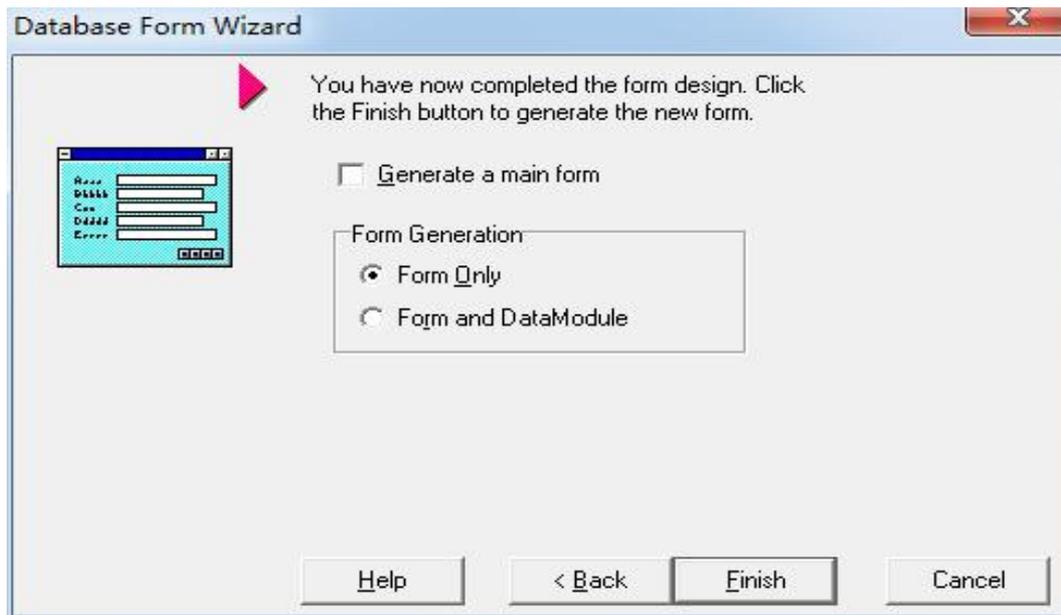
- ③ 选择需要的字段。
 - 在选择了数据库表格后，单击**Next**按钮，就弹出选择字段的对话框。在左边**Available Fields**对话框中是数据表格的所有字段，从中选择需要的字段，单击**>**按钮添加到右边的对话框中。如果所有的字段全选，单击**>>**按钮。



9.5.1 使用Form Wizard创建单个数据表格的应用程序



9.5.1 使用Form Wizard创建单个数据表格的应用程序



9.5.1 使用Form Wizard创建单个数据表格的应用程序



- ④ 选择应用程序的格式。
 - 在选择完字段后，单击**Next**按钮，弹出选择应用程序的格式对话框。**Horizontally**是水平方式显示表格中的数据，**Vertically**是垂直方式显示表格中的数据，**In a grid**是网格方式显示表格中的数据。



9.5.1 使用Form Wizard创建单个数据表格的应用程序



- ⑤ 选择应用程序的生成方式。
 - 单击**Next**按钮就会弹出选择应用程序的生成方式对话框。在这个对话框中我们选择只生成窗体。

- ⑥ 生成应用程序。
 - 单击**Finish**按钮就完成了使用**Form Wizard**生成应用程序的过程。生成的应用程序的窗体如图所示。



9.5.1 使用Form Wizard创建单个数据表格的应用程序



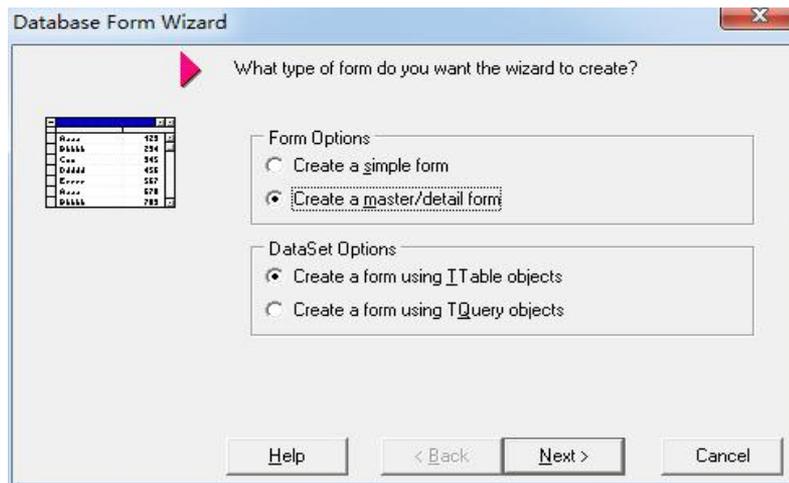
CustNo	Company	Addr1	
EditCustNo	EditCompany	EditAddr	
Addr2	City	State	
EditAddr2	EditCity	EditState	
Zip	Country	Phone	FAX
EditZip	EditCountry	EditPhone	EditFAX
TaxRate	Contact	LastInvoiceDate	
EditTaxRate	EditContact	EditLastInvoiceDate	





9.5.2 使用Form Wizard创建一对多的程序

- ① 在主菜单中选择Database选项中的Form Wizard选项。
- 在弹出的窗口中选择创建一对多的应用程序。





9.5.2 使用Form Wizard创建一对多的程序

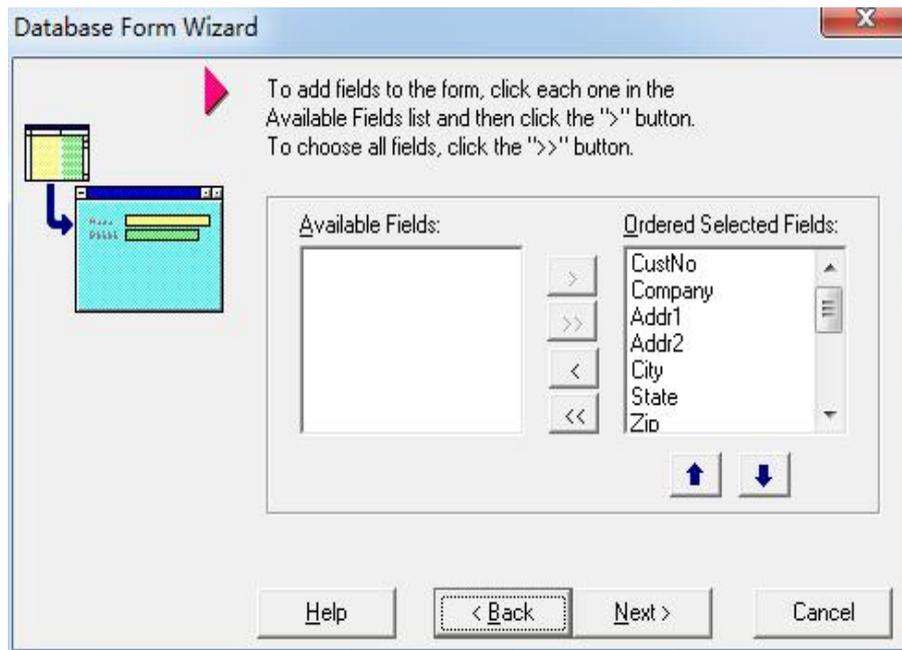
- ② 选择要创建应用程序的主数据表格。
 - 选择应用程序的主数据表格为Customer.db。

- ③ 选择主数据表格的字段。
 - 选择需要的字段。单击>>按钮，选择所有的字段。





9.5.2 使用Form Wizard创建一对多的程序





9.5.2 使用Form Wizard创建一对多的程序

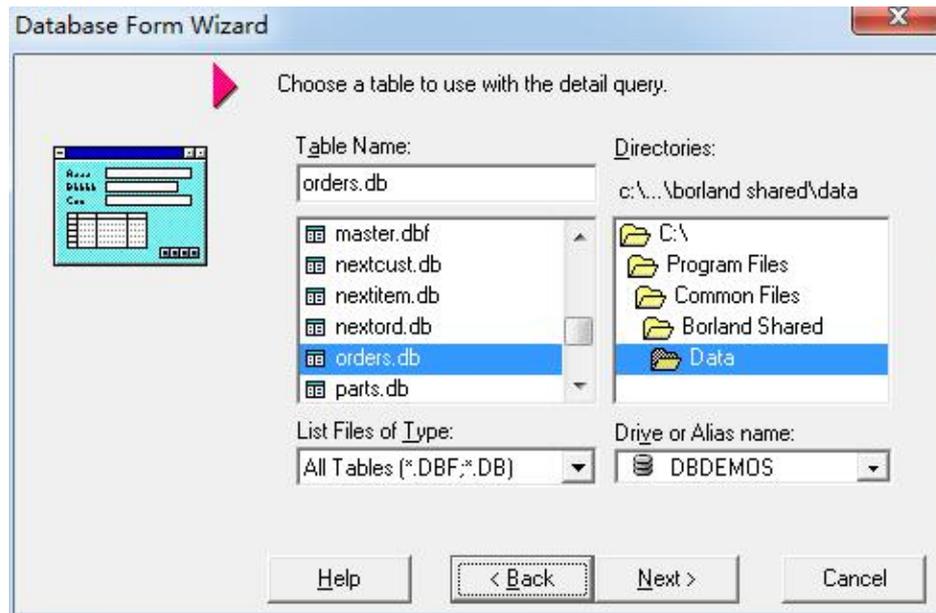
- ④ 选择主数据表格的显示格式。
 - 选择主数据表格的显示格式In a grid显示方式。

- ⑤ 为主数据表格选择一个从数据表格。
 - 选择orders.db为从数据表格。





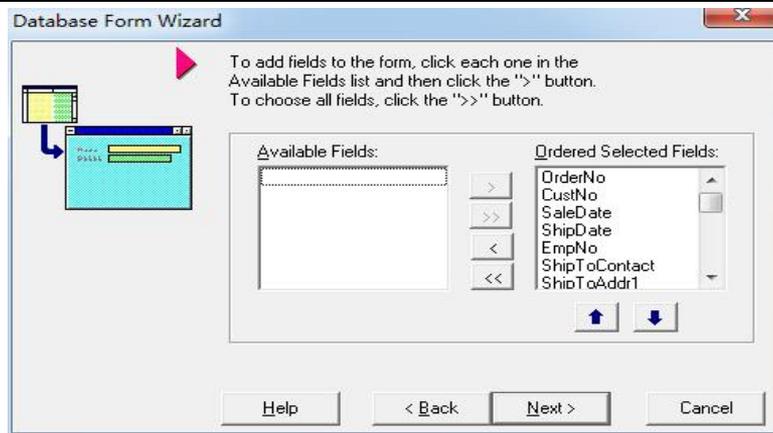
9.5.2 使用Form Wizard创建一对多的程序





9.5.2 使用Form Wizard创建一对多的程序

- ⑥ 选择从数据表格的字段。
 - 单击Next按钮选择从数据表格的字段。单击>>按钮，选择所有的字段。





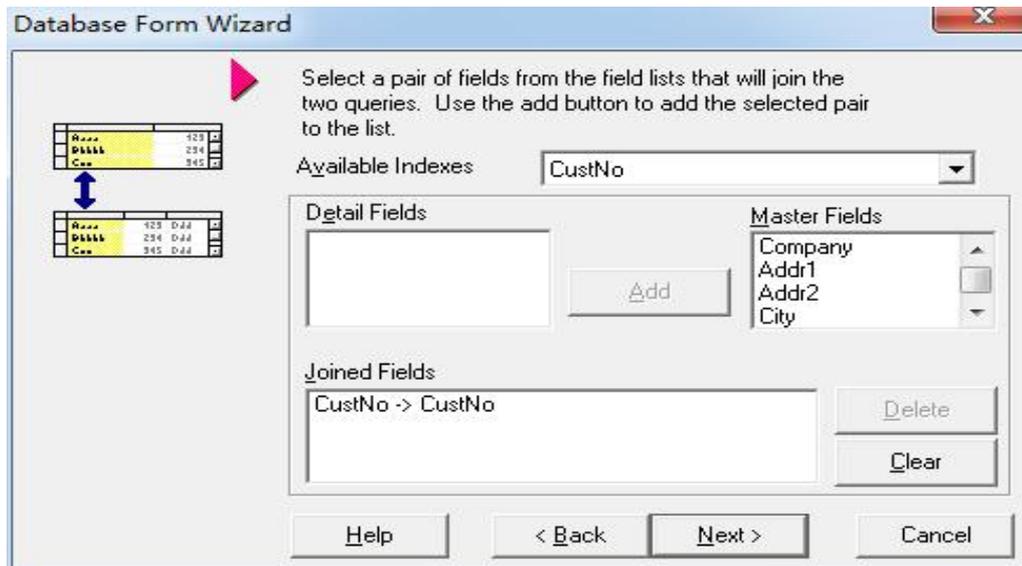
9.5.2 使用Form Wizard创建一对多的程序

- ⑦ 选择从数据表格的显示方式。
 - 单击Next按钮，为从数据表格选择In a grid显示方式。
- ⑧ 选择主从数据表格的链接字段。
 - 单击Next按钮，选择主数据表格和从数据表格之间链接的字段。
 - 在Available Indexes列表框中选择CustNo，然后在Master Fields列表框中选定CustNo，单击Add按钮。





9.5.2 使用Form Wizard创建一对多的程序





9.5.2 使用Form Wizard创建一对多的程序

⑨ 完成创建应用程序。

单击**Next**按钮，选择应用程序的创建方式。选择**Generate a main form**选项和**Form only**选项，单击**Finish**按钮，就完成了使用**Form Wizard**创建一对多应用程序的过程。完成后的应用程序的窗体如图所示。





9.5.2 使用Form Wizard创建一对多的程序

Form1

CustNo	Company	Addr1	Addr2
1221	Kauai Dive Shoppe	4-976 Sugarloaf Hwy	St
1231	Unisco	PO Box Z-547	
1351	Sight Diver	1 Neptune Lane	
1354	Cayman Divers World Unlimited	PO Box 541	
1356	Tom Sawyer Diving Centre	632-1 Third Frydenhoj	
1380	Blue Jack Aqua Center	23-738 Paddington Lane	Su

OrderNo	CustNo	SaleDate	ShipDate	EmpNo	Ship
1023	1221	1988-7-1	1988-7-2	5	
1076	1221	1994-12-16	1989-4-26	9	
1123	1221	1993-8-24	1993-8-24	121	
1169	1221	1994-7-6	1994-7-6	12	
1176	1221	1994-7-26	1994-7-26	52	
1269	1221	1994-12-16	1994-12-16	28	

