

2011年计算机等级考试二级Delphi辅导讲义：简单数据库应用的创建及MASTAPP介绍 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/646/2021_2022_2011_E5_B9_B4_E8_AE_A1_c97_646918.htm 导读：本章主要介绍用Delphi开发简单的数据库应用程序的一般方法和步骤，首先让读者对Delphi强劲的数据库应用开发工具有一个直观的印象，然后在此基础上进行复杂的数据库应用程序的设计。>>>>点击查看此系列辅导讲义汇总第十四章 简单数据库应用的创建及MASTAPP介绍 Delphi中嵌入的数据库应用开发工具

如Database Form Expert具有很强大的功能，我们不需要编写任何程序代码便可以快速地创建一个简单的数据库应用程序，甚至还能创建基于多个数据库表的主要 明细型数据库应用程序。本章主要介绍用Delphi开发简单的数据库应用程序的一般方法和步骤，首先让读者对Delphi强劲的数据库应用开发工具有一个直观的印象，然后在此基础上进行复杂的数据库应用程序的设计，本章主要包括以下内容： 创建数据库应用窗体 包括用Database Form Expert 或手工方式创建简单的无需编写程序代码的应用程序或者利用多个部件并编写功能复杂的程序代码创建主要 明细型数据库应用程序。

在应用程序中控制字段有关的属性 描述怎样读写数据库表中字段的值和控制字段的显示格式等。本章所介绍的例子中用到的窗体、数据库表以及相关的文件都是在安装Delphi时缺省安装在C:\DELPHI\DEMOS\DB\MASTAPP目录中，并且用别名DBDEMOS表示这一子目录。在本章例子中，除特殊声明外，所有的TTable和 TQuery 部件的 DatabaseName 属性都

设置为DBDEMOS。 14.1 简单的基于单表的数据库应用

用Decphi创建显示一个数据库表中的内容的应用非常简单和方便，只需要三个部件，只要将这三个部件通过相关的属性相互联系起来，不需要编写任何程序代码便可以实现。例如，用户想查看数据库表Customer.DB中的内容时，可以按下面步骤来实现：

14.1.1 选择相关的部件：选择菜单Project/New开始一个新工程，并修改Form1的Caption属性为CustomerFrom1并把Name属性设置为CustomerForm1，然后从部件选择板上的Data Access 页上选取一个Datasource部件和一个Table部件放到窗体的左上角，它们是非可见的部件，在窗体中我们看到的只是部件的图标；从Data Control页上选取DBGrid部件放到窗体中前两个部件的下面。完成这些工作之后，窗体如图14.1所示。图在CustomerFrom1窗体中放置三个部件

14.1.2 设置部件的属性 为了使TDBGrid部件能够显示数据库表Customer.DB中的客户信息，我们必须修改窗体三个部件相关的属性，这些属性的设置如表14.1所示。表14.1 CustomerFrom1窗体中三个部件的属性设置

属性	属性值
DataSource1.AutoEdit	False
DataSource1.DataSet	Table1
Table1.DatabaseName	DBDEMOS
Table1.TableName	CUSTOMER.DB
Table1.Active	True
DBGrid1.DataSource	DataSource1

这里要注意的是：DBDEMOS是Delphi缺省安装时C:\Delphi\DEMO\DB\MASTAPP目录的别名，而且数据库表Customer.DB存在该目录下，用户在使用这一例子时，

请注意这两项设置都是正确的。另外 Datasource1.Dataset , Table1.TableName和DBGrid1.Datasource属性都有下拉式列表框允许用户从可能的值列表中选择它们的值，这样能方便我们进行属性的设置，而且不容易出错。

Datasouce1.AutoEdit属性设置为False是为了防止用户修改数据库表中的数据，在下面的讨论中我们将详细地进行说明。

Table1.Active设置为True时,Delphi会打开Table1.TableName所指定的数据库表。如果这个数据库表不存在（或表中什么也没有，即空表），Delphi会弹出出错信息并且Table1.Active变成False。当Table1.Active被设置成True之后，Table1 部件的一些属性就不能再修改了，如Table1.DatabaseName和Table1.Tablename属性。若要修改它们，必须首先要将Table1.Active属性设置为False，然后再进行修改，否则

，Delphi会弹出错误信息“Cannot perform this operation on an open database”。当看到这个错误信息时，只需

把Table1.Active置成False，完成相关的修改后，再把Table1.Active属性设置为True。当我们把DBGrid1.DataSource的值设置成DataSource1时，Delphi会把Customer.DB中的数据填充到DBGrid1部件中，并且可以用DBGrid1中的滚动条来浏览数据库表中的所有记录。

14.1.3 运行程序 保存文件，命名代码单元为Cust.pas，命名工程名为CustPRJ.DPR,然后按F9编译并运行程序。程序执行之后，我们可以使用滚动条或键盘移动键在字段和记录间移动。但不能修改表中的数据，因

为Datasouc1.AutoEdit1属性已被设置为False。Cust程序中的三个部件都有各自的特殊用途，三个部件的相关属性在内部相互联系生成最终的应用程序。TTable部件连接磁盘上的实际

数据库表和应用程序中其他部件的通道。TTable部件具有打开和关闭、读取、更新以及其他处理磁盘数据库文件的方法。TDataSource部件是连接TTable部件和数据浏览部件如TDBGrid部件的桥梁。TDBGrid部件用于显示数据库表中的数据信息，它为应用程序提供一个直观的界面。图14.2阐述了这三个部件之间的关系。Cust程序中三个部件之间的内部关系 TDBGrid 部件的奇妙之处在于它知道如何去获取数据库表中的下一条或前一条记录，我们使用滚动条或箭头键便可以完成这项任务。TDBGrid部件不知道如何增加、删除和修改记录。如果想让 Cust 程序能够修改数据库表中的记录，只要把 DataSource1 部件的AutoEdit属性设置成True，并重新编译和运行程序就可以达到目的。使用箭头键，把DBGrid的高亮度条定位到某一个字段上，然后键入新值，该字段中的值将被键入的新值所取代，并且当移动到另一条记录时，键入的信息会自动写入数据库表中。如果想放弃所做的改动，只需在离开该字段前按一下Escape键。如果想在表中增加新记录，可以把高亮度条移到网格底端的空白记录上并输入新记录的有关字段值。也可以在用户指定的某一条记录的后面插入一条新记录，只要把高亮度条定位到指定的记录上，按Ins键，使可以在该记录的后面插入新记录。删除某一条记录时，把高亮度条定位在想删除的记录的任意字段上，按Ctrl del键,这时会出现保护信息，我们可以确认是否真的想删除该项记录。TDBGrid为用户提供了较完备的功能，用于控制是否编辑、增加或删除记录。若想禁止对数据库表作任何修改，设置TDBGrid部件的ReadOnly属性为 True，并设置 Option.dgEDiting为False(这将为我们提供一个只读的数据库表

浏览器而不是数据库编辑器，但它隐含着增加、编辑和删除记录的能力)。TDBGrid部件的这些属性和Option属性其它选项的各种不同组合可以让我们很方便地对数据库表进行有效的浏览、编辑等操作。如果我们经常使用像电子表格那样的界面来显示和编辑数据记录，TDBGrid 部件便是一个很方便的工具，但那并不是最友好的用户界面，如果想拥有更优美更直观的界面，我们还可以使用单独的数据浏览部件来显示数据库表中各个字段的值，并利用TDBNavigator部件控制对数据库表的存取。

14.2 利用TDBNavigator部件创建存取程序

我们可以改进一下Cust程序以便它一次只在对话框中显示一个客户的记录信息，并用一个TDBNavigator部件控制对记录存取。允许我们选择一个记录来显示或编辑以及增加和删除记录。完成的应用窗体。

14.2.1 创建应用程序窗体

我们可以非常迅速地创建起来，因为到目前为止我们对创建窗体的方法已经比较熟悉，我们首先把所有的部件都放到窗体中，然后再设置它们的属性。开始一个新工程，设置窗体Form1的Name 属性为 Customerform2 ， Caption 属性为 CustomerForm2。然后从部件选择板上的Data Access页上选取一个DataSource部件和一个Table部件放在窗体的右上角。再从Data Controls页上选取DBNavigator部件放在窗体的左上角。窗体中其余的部件如图14.3所示。它们是TDBEdit和TLabel 部件，按图14.3 所示创建并布置部件，分别命名DBEdit部件为EditCustno、 Editcompany、 EditAddr1、 EditAddr2、 EditCity、 EditState、 EditZip、 EditCountry、 EditPhone、 EditFAX、 EditTaxRate、 EditContact。现在我们来连接TTable部件和 TDataSource 部件，然后连接所有的数据浏览

部件和DataSource部件。设置TBNavigator部件和TDBEdit部件的属性，它们的DataSource属性都设置为DataSouce1。我们最后要做的事是连接窗体中各个TDBEdit 部件和它们在数据库表中对应的字段名，通过设置TDBEdit 部件的 DataField 属性来完成。例如要连接命名为EditCustNo的TDBEdit部件和数据库表中的CustNo字段，具体步骤如下： 选中窗体中的EditCustNo部件。 在Object Inspector窗体中，单击DataField属性右边的箭头。 从下拉列表中选中CustNo字段名。对窗体中的其他TDBEdit部件执行以上操作连接到其对应的字段，然后保存文件。命名代码单元名为Cust2.pas，命名工程名为Cusprj2.DPR。

14.2.2 使用TDBNavigator部件移动记录指针

上述程序运行之后，在数据浏览部件中会显示数据库表中的第一条记录。利用Tab 键可以在字段之间移动，但是不能编辑字段。因为我们为了防止意外修改，设置了Table1的AutoEdit属性值为False。如果想对数据库表中的记录进行编辑、插入和删除操作或者想显示数据库表中另一条记录，需要按 TDBNavigator 部件上这些功能所对应的功能按钮。TDBNavigator部件上的按钮和它们的功能如图14.4所示。TDBNavigator中的按钮 TDBNavigator部件的绝大多数功能都可以根据其按钮的图标能够很容易地识别出来，而且TDBNavigator部件本身能感知到很多事情，如当前指针是否在数据库表的开头或尾部。如果用户正在查看数据库表中的最后一个记录，Next和Last按钮将会变灰成为非活动状态。同样，如果用户当前正在浏览数据库表中的第一条记录，TDBNavigator 上的 First 和Previous按钮会变灰而成为非活动状态。有关各个按钮的作用的更详细说明请查看联机帮助。如

果用户想修改当前的记录，单击TDBNavigator部件的Edit按钮，然后完成需要做的修改，在做完修改之后，单击Post按钮以便将作的修改写入实际的数据库表中（更新实际的数据库表中的记录在数据库术语中叫作“投寄”记录即PostT）。如果想取消所做的修改，单击Cancel按钮。Cancel按钮只取消自从上一次往数据库表中投寄记录以来对记录所做的修改。例如，如果用户曾修改了CustNo字段并单击了Post按钮投寄了修改，然后再修改Company字段并按Cancel，那么只有对Company所做的修改将会被取消。也就是说，一旦修改被写入了数据库表中，再按Cancle按钮是无法取消对记录的修改的，要想恢复到以前的状态，用户必须要重新编辑修改记录。值得注意的是，当用户修改了当前的记录，并移动到其他记录时，TDBNavigaator部件会自动地投寄用户对记录的修改。例如：如果我们修改了记录的Company字段，并没有按Post按钮以更新表中的记录，而是移动到下一条记录，这时用户对记录的修改也会自动地被写入数据库表中。

14.2.3 定制TDBNavigator部件

TDBNavigator部件中的按钮对我们开发人员来说是很方便的，但对于程序的最终用户来说不一定那么一目了然。为了帮助最终用户或初级用户更方便有效地使用TDBNavigator部件，我们可以设置TDBNavigator部件的ShowHint属性为True，这样当鼠标光标停留在TDBNavigator部件上的某一个按钮上超过大约1秒钟，在屏幕上便会出现该按钮的提示信息。如果我们不想使用TDBNavigator部件本身嵌入的提示信息，我们还可以设置TDBNavigtor部件的Hints属性，为每个按钮指定特定的提示信息，以帮助用户使用TDBNavigator部件。 TDBNavigator部

件中有多个功能按钮，但并不是所有的按钮对每一个数据库应用程序都是需要的，特别是那些不允许修改表中的数据，或修改只是在很严格的控制下进行的数据库应用程序。我们可以通过设置TDBNavigator部件的 VisibleButtons 属性来确定要在TDBNavigator中显示哪些按钮步显示哪些按钮。例如，如果我们不允许用户修改表中的记录，我们就不需要Add、Delete、Post、Cancel 或 Refresh 按钮，我们设置这些按钮的VisibleButtons属性为False，这样在TDBNavigator部件中将不会出现这些按钮。 TDBNavigator部件的ConfirmDelete属性和Delete 按钮配合使用对用户删除数据库表中的记录是非常有用的，当ConfirmDelete属性设置为 True（缺省设置），当用户单击Delete按钮试图删除当前记录时，Delphi会弹出一个确认框，要用户确认是否真的想删除当前记录。这样，在用户进行删除记录的操作时，会更安全一些。如果用户不希望按下Delete按钮时出现确认框，只要把ConfirmDelete设置为False就可以了。还有一些属性可以用来定制TDBNavigator部件的外观和性能，有关这方面的详细信息请参看联机帮助。

14.3 创建主要 明细数据库应用

我们前面在介绍的基于单个数据库表的数据库应用程序只能对数据库表进行简单的管理，大多数只用来浏览单个数据库表中的记录信息，如果我们想浏览多个相关的数据库表中的记录信息，就必须创建主要 明细型数据库应用程序。在主要 明细型数据库应用程序中，一个数据库表作为主要表，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，它们当中存放着更详细的信息。例如，当数据库表Customer.DB作为主表，它包含着客户的综合信息如编号、姓名、所在公司的名称

等等。而数据库表Orders.DB中包含着每个客户的订货单的详细信息，如订单号、发货日期、起运日期、发货目的地等信息，这样当在Customer.DB表中查看某一位客户时，利用其中的字段CustNo与Orders.DB表发生联系，自动地从Orders.DB表中检索出这位客户曾经发来的所有订货单的详细信息。主要

明细型数据库体现了关系数据库的特点，即独立的数据库表之间基于它们共同的字段而发生联系。在这里Customer.DB和Orders.DB拥有一个共同的字段CustNo。

14.3.1 一对多关系的主要 明细型数据库应用程序 主要和
明细数据库表之间存在一对多的关系，意思是说对于主表中的一条记录，在明细表中有多条记录与之对应。例如，创建一个主要
明细型数据库应用程序，其包括两个
表Customer.DB和Orders.DB，它们分别作为主表和明细表，
创建好的应用如图14.5所示，窗体中各部件的属性设置 表14.2
主要 明细型数据库应用中各部件的属性

部件属性属性值注释

	Table1 Active
True (主表)	DatabaseName DBDEMOS TableName
CUSTOMER.DB	
	DataSource1 DataSet Table1
AutoEdit False	
	Table2 Active True (明细表)
DatabaseName DBDEMOS	TableName ORDERS.DB
IndexFieldNames CUSTNO	指定字段CUSTNO作为 Table2中的
索引字段	MasterField CUSTNO 指定与主表发生联系 的字段

MasterSource DataSource1 说明与主表相连接的数据源
即DataSource

DataSource2 DataSet Table2

AutoEdit False

DBGrid1 DataSource

DataSource1 (对应主表)

DBGrid2 DataSource

DataSource2 (对应明细表) TableName ORDERS.DB

DBNavigator1 DataSource DataSource1

一对多关系是非常普遍的关系。即便是简单的名字/地址数据库都有一对多的关系，因为一个人可能不止一个地址：家庭地址、工作地址、还可能有别墅地址。在本例中，公司的一个客户常常有多个订货单，当我们单击DBNavigator1中的向前、向后按钮时，会移动DBGrid1中的记录指针，而在DBGrid2中会自动显示与DBGrid1中当前记录相关的多条记录，即显示一个客户的信息时，同时会显示该客户的所有订货单的详细信息。

14.3.2 一对多 多关系的数据库应用

前面我们介绍了基于两个表的一对多关系的应用，下面我们介绍怎样创建一个从三个表中浏览数据记录的一对多关系的应用。例如：一个客户也许有多张订货单，而每一张订货单中有多个订货项目，这样我们在Customer.DB表和Orders.DB表之间建立一个主要 明细型关系，同时在orders.DB表和Items.DB表之间建立一个主要 明细型关系。窗体中各部件的属性如表14.3所示

表14.3 一对多 多关系的应用中

各部件的属性

部件属性属性值注释

Active True Table1 DatabaseName DBDEMOS
TableName CUSTOMER.DB
DataSource1
DataSet Table1 AutoEdit False
Active True
DatabaseName DBDEMOS Table2 TableName ORDERS.DB
IndexFieldNames CUSTNO MasterField CUSTNO MasterSource
DataSource1
DataSource2 DataSet Table2
AutoEdit False
Active True DatabaseName
DBDEMOS Table3 TableName ORDERS.DB IndexFieldNames
ORDERNO MasterField ORDERNO MasterSource DataSource2
DataSource3 DataSet Table3 AutoEdit False
DBGrid1 DataSource DataSource3
DBNavigator1 DataSource DataSource1

窗体中其余的部件都是TDBEdit和TLabel部件，它们用于显示Customer.DB中的字段值和Order.DB中的字段值。在该例子中，总共连接了三个表，Customer.DB表是主要表

，Orders.DB表在窗体中起到了双重作用，它既是Customer.Db表的明细表，同时又是Items.DB表的主要表，Items.DB表是Orders.DB表的明细表。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com