数据库与控件程序设计实例丛书

Delphi 6 数据库实用程序设计 100 例

段兴　主编
唐一丁　李宝　于山山　编写

人民邮电出版社

需要全本请访问 www.ertongbook.com

内容提要

本书精心选取了利用 Delphi 6 开发的数据库方面具有代表性的 100 个编程实例。这些实例按照完成功能划分，包括数据库控件与界面设计、数据库基本操作、数据库高级操作、远程数据访问、第三方数据库控件及应用、XML 和典型数据库应用系统等几方面的内容，基本涵盖了 Delphi 6 数据库开发的主要方面。

实用是本书的最大特色。本书大部分实例模仿优秀流行软件的相关部分，或提供关键问题的解决方案，部分实例侧重于帮助读者理解 Delphi 6 数据库开发的基础概念、重点和难点。本书最后介绍的几个典型的数据库应用系统，如工资管理系统、人事管理系统、缴费系统、酒店客房管理系统等，实用性极强，界面美观，功能齐全。通过对这部分内容的阅读，读者可以对 Delphi 数据库软件开发的全过程有一个全面的掌握。

本书适合掌握了 Delphi 6 基本编程技术的读者阅读。

数据库与控作程序设计实例丛书

Delphi 6 数据库实用程序设计 100 例

◆ 主　　编　段　兴

编　　写　唐一丁　李　宝　于山山

责任编辑　邹文波

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号

邮编　100061　　电子函件　315@ptpress.com.cn

网址　http://www.ptpress.com.cn

读者热线：010-67180876

北京汉魂图文设计有限公司制作

北京　　印刷厂印刷

新华书店总店北京发行所经销

◆ 开本：787×1092　1/16

印张：

字数：　千字　　2002 年 9 月第 1 版

印数：1 –　000 册　　2002 年 9 月北京第 1 次印刷

ISBN 7-115-10597-9/TP・3062

定价：　.00（附光盘）

本书如有印装质量问题，请与本社联系　电话：（010）67129223

前　言

Delphi 6 是由软件界巨头、著名的 Borland 公司开发的可视化编程语言，由于其可视化程度高，代码简洁易读，因此是最容易入门的编程语言之一。

Delphi 使用了 Microsoft Windows 图形用户界面的许多先进特性和设计思想，采用了弹性可重复利用的完整的面向对象程序语言（Object-Oriented Language）、当今世界上最快的编辑器和最为领先的数据库技术。Delphi 数据库技术兼顾了 Visual FoxPro、PowerBuilder 等优秀数据库开发语言的可视化技术，提供了极其丰富的控件，而且融入了 Delphi 模块化结构等优秀设计思想，使得从桌面应用向多层、向 Web 应用程序、向 XML 方面扩展都非常容易。同时在跨平台使用和 WAP 手机无线应用方面都显示出巨大的潜力。所有上述优点，在本书中都有较详细的讲解。本书最后给出的 10 个典型应用系统，对前面实例进行了一定的集成，完整地讲解了常见数据库程序的开发思路及原理，使得读者可以尽快掌握实用型数据库的开发方法。

本书精心选取了 100 个编程实例，共分为 7 篇。

第一篇　数据库控件与界面设计，重点介绍 Delphi 数据库控件与界面设计技术，包括数据库表的连接、显示、动态编辑、分类查询、报表生成等技术及相关控件的使用，同时还为读者提供了 Delphi 通用界面设计与数据库系统设计的一些常用技巧。

第二篇　数据库基本操作，重点介绍 Delphi 数据库基本操作技术，包括与 Access、Paradox、Oracle、SQL Server 等不同格式数据库表的连接、转换、数据复制、数据指针定位、过滤、动态查询、统计等技术及相关控件的使用，同时还为读者提供了 Delphi 数据库基本操作的其他一些常用技巧。

第三篇　数据库高级操作，重点介绍 Delphi 数据库高级操作技术，包括 Access、Paradox、Oracle、SQL Server 等不同格式数据库表之间的任意转换、显示方式的优化与定制、数据表继承关系与动态建立、数据表的索引维护、图表动态生成、流（Tstream）技术在数据库中的应用等技术及相关控件使用，同时还为读者提供了 Delphi 数据库高级操作的其他一些常用技巧。

第四篇　远程数据访问，重点介绍 Delphi 远程数据访问技术，包括数据模块的使用、DataSnap 服务器的建立、GUID 的生成及使用、Web 服务及信息安全、WebSnap 数据库及 ActiveForm 数据浏览、DataSnap 服务端编辑、MTS 服务端及客户端建立等技术及相关控件的使用，同时还为读者提供了 Delphi 远程数据访问的一些常用技巧。

第五篇　数据库相关第三方控件，重点介绍 Delphi 数据库相关的第三方控件技术，包括通用报表工具、万能查询、用户定制、数据浏览等技术及相关控件的使用，同时还为读者提供了 Delphi 数据库相关第三方控件技术的一些常用技巧。

第六篇　其他相关技术，重点介绍 Delphi 数据库相关技术，包括数据库安全、版权保护、动画显示图像、XML 文档应用等技术及相关控件使用。

第七篇　典型应用，重点介绍 Delphi 数据库综合应用软件开发技术，使用到前面各篇的多种数据库相关技术。本篇介绍的典型数据库应用系统有工资管理系统、人事管理系统、缴

费系统、移动电话管理系统、学生成绩管理系统、列车票务系统、酒店客房管理系统等。这些系统实用性极强，界面美观，功能齐全。通过对这部分内容的阅读，读者可以对 Delphi 数据库软件开发的全过程有一个清楚的了解。部分系统是实用的商用数据库软件，读者可以直接引用或稍加修改，即可应用在自己的开发系统中。

每个实例的讲解分为三个步骤：

- 实例效果——讲解本例的功能，指出本例的设计目的与效果，使读者对本例有一个直观的认识。
- 程序说明——讲解本例的技术原理及设计思路，给出程序设计的基本流程，使读者能迅速理解程序代码；同时讲解本例涉及到的相关知识，使读者在学会本实例应用的同时，对相关概念、函数调用与参数配置等相关知识也有相应了解，对读者真正掌握实例的设计思路很有帮助。
- 制作步骤——给出具体的实现过程，包括窗体界面的建立、源代码与注释等，读者可以参照编写程序。

本书的配套光盘中包含本书的所有实例，全部经过精心调试，读者可以直接调用。

本书的编写凝聚了一批程序员、高校教师的辛勤工作。其中南京工业大学的唐一丁、青岛海洋大学的于山山、李宝都将很多优秀的原创代码无偿拿出并仔细修订、逐一调试，刘元副教授在百忙之中抽出时间对本书的重点章节及代码进行审阅、调试，并提出了很多宝贵的意见。此外参加本书编写的还有万修胜。正是他们出色的才华与良好的团队合作精神，才使得本书在质量上更进一步，在此谢谢他们。

本书由青岛三角洲数码技术开发中心策划，段兴主编。

由于水平有限，加之时间仓促，本书难免存在不足之处，还望广大读者批评指正。

本书的约定：书中提到的 Delphi 均指 Delphi 6。

本书光盘使用办法：

使用本书光盘实例源代码时，大部分实例需要重新给数据库所在的文件夹设置数据库别名，如“ mydatabase ”等；另外部分实例需要重新安装，有的需要注册第三方控件；远程访问一篇的实例需要设置更多的参数才能正常运行；各实例具体的数据库别名等参数设置请参阅该例程序说明及制作步骤。

所有使用到的数据库文件都存放在光盘根目录下，或光盘根目录下“ 数据库 ”文件夹内，请根据自己计算机的有关设置重新定位源代码的位置。

编者

bookwood@sina.com

目　录

第一篇　数据库控件与界面设计 …… 1

实例 1　将控件植入 DBGrid 的 Cell 中 …… 3
实例 2　让 DBGrid 自动调整宽度 …… 7
实例 3　单击 DBGrid 标题进行排序 …… 12
实例 4　在 DBGrid 中使用 ENTER 替代 TAB 操作 …… 15
实例 5　在 DBComboBox 中显示记录列表 …… 18
实例 6　改变当前 DBGrid 编辑框的背景颜色 …… 21
实例 7　制作登录密码对话框 …… 25
实例 8　ADO 数据集进度条的显示 …… 28
实例 9　ADO 连接 SQL Server …… 31
实例 10　给活动单元格/行加上特殊背景色 …… 33
实例 11　如何动态地增加字段 …… 38
实例 12　利用 Tbatch 组件完成数据的批处理 …… 41
实例 13　用 TreeView 控件实现数据库的分类查询 …… 45
实例 14　在 DBGrid 中插入下拉选框 …… 48
实例 15　处理数据栅格的编辑框 …… 51
实例 16　用 Delphi 的 QReport 部件生成报表 …… 54
实例 17　利用 API 实现网格内组件的嵌入 …… 57
实例 18　TSession 组件的应用 …… 61

第二篇　数据库基本操作 …… 65

实例 19　在同一数据表中复制记录 …… 67
实例 20　统计数据集的记录数 …… 71
实例 21　动态获取 BDE 别名的相关信息 …… 75
实例 22　建立 BDE-Paradox 数据库 …… 79
实例 23　在 DBGrid 中复制和粘贴数据 …… 83
实例 24　使用 ADO 打开带有密码的 Access 数据库 …… 87
实例 25　判断某一列数据是否为空 …… 92
实例 26　如何动态建立 Access 数据库 …… 96
实例 27　如何动态建立 SQL Server ODBC …… 100
实例 28　将 ADO 数据集保存成文件 …… 106
实例 29　使用 Locate 定位数据指针 …… 109
实例 30　利用 Table 的过滤机制实现动态查询和统计 …… 112
实例 31　在 DBGrid 中实现 Copy、Paste 功能 …… 115
实例 32　在 TDBGrid 控件中处理多个记录 …… 117

需要全本请访问 www.ertongbook.com

实例 33　数据库资料到文本格式的转换 …… 120
实例 34　字段的使用 …… 123
实例 35　数据库格式的转换 …… 127
实例 36　利用拼音首字符检索 …… 129
实例 37　对用户输入数据的控制 …… 133
实例 38　实现 Pack 功能 …… 137
实例 39　动态创建数据库别名 …… 140

第三篇　数据库高级操作 …… 143

实例 40　重整 DBGrid 中的栏目次序 …… 145
实例 41　快捷方式的建立与删除 …… 148
实例 42　在一个 DBGrid 中显示多数据库 …… 151
实例 43　保存 DBGrid 中的设置 …… 154
实例 44　在数据库内批量转换表格式 …… 158
实例 45　数据表继承关系的建立 …… 161
实例 46　动态建立数据表 …… 166
实例 47　使用计算出的字段 …… 170
实例 48　维护 Parodox 数据表的索引 …… 173
实例 49　动态生成图表 …… 178
实例 50　利用 TStream 读写数据 …… 181
实例 51　书签的应用 …… 185
实例 52　主从数据表的链接 …… 187
实例 53　下拉列表和查找字段在 DBGrid 中的应用 …… 189

第四篇　远程数据访问 …… 193

实例 54　如何使用数据模块 …… 195
实例 55　如何建立 ClientDataSet Flat 数据库 …… 198
实例 56　转换成 CDS 数据库 …… 202
实例 57　如何生成一个 GUID …… 205
实例 58　建立基本 DataSnap 服务器 …… 209
实例 59　建立 DCOM 远程数据访问连接 …… 214
实例 60　建立 SocketConnection 远程数据访问连接 …… 218
实例 61　建立 WebConnection 远程数据访问连接 …… 222
实例 62　从 WSDL Import Wizard 生成 Web 服务 …… 224
实例 63　为生成的 Web 服务制作客户端 …… 229
实例 64　建立 Web 服务的数据提供端 …… 234
实例 65　建立 Web 服务的数据访问端 …… 240
实例 66　开发 WebSnap 数据库程序 …… 244
实例 67　建立 ActiveForm 数据浏览 …… 250

实例 68　建立数据查询 WebServices 服务器端 …… 254
实例 69　数据查询 Web 服务客户端开发 …… 258
实例 70　基于 WAP 的手机无线应用 …… 261
实例 71　使用类型库编辑 DataSnap 服务端 …… 269
实例 72　建立自动化控制器获取数据 …… 273
实例 73　建立基本 Web 数据库服务器应用 …… 276
实例 74　建立查询条件 Web 数据库服务器应用 …… 281
实例 75　建立基本 MTS 服务端 …… 285
实例 76　建立 MTS 数据访问客户端 …… 291

第五篇　数据库相关第三方控件 …… 295

实例 77　通用报表工具 SReport …… 297
实例 78　万能查询控件 WNQuery …… 300
实例 79　查询控件 EsLib …… 303
实例 80　报表控件 FastReport …… 306
实例 81　Exporter 控件组的使用 …… 309

第六篇　其他相关技术 …… 313

实例 82　在数据表中使用弹出菜单形式的下拉列表 …… 315
实例 83　自动删除前 30 天的记录 …… 319
实例 84　用户登录时最多可输入 3 次密码 …… 323
实例 85　动画显示图像字段 …… 326
实例 86　用 TXmlDocument 创建一个 XML 文档 …… 331
实例 87　数据库生成 TreeView 列表 …… 337
实例 88　软件版权保护 …… 340
实例 89　E-mail 的发送（利用 NMSTMP） …… 345
实例 90　实现远程抓图 …… 349

第七篇　典型数据库应用系统 …… 355

实例 91　学生宿舍管理系统 …… 357
实例 92　工资管理系统 …… 365
实例 93　人事管理系统 …… 373
实例 94　学生缴费系统 …… 382
实例 95　移动电话管理系统 …… 392
实例 96　学生成绩管理系统 …… 400
实例 97　列车票务系统 …… 410
实例 98　酒店客房管理系统 …… 416
实例 99　个人上网记费系统 …… 425
实例 100　医院住院部管理系统 …… 434

第一篇　数据库控件与界面设计

在 Delphi 出现之前，注重于数据库引擎技术的相关语言（如 Dbase、Foxbase）,与注重于界面设计的算法语言（如 C 语言、C++、Visual C++等）是平行发展的。经过近 20 年的发展，两者不但没有相互融合，其间距离反而越来越大，导致了用数据库编程语言（如 Visual FoxPro、Power Builder 等）编写的软件界面单调、可用控件很少，大大制约了数据库程序的界面效果，也与当今人性化、个性化的程序界面设计潮流不相符合。

在这种情况下，Delphi 提供了将数据库技术与界面设计技术完美结合的完整解决方案。特别是 Delphi 6 的推出，使得用于算法语言的各种控件与界面设计技术，几乎可以全部用于数据库程序。流行的 Windows XP 菜单、换肤技术、异形窗口等，都可以使用在 Delphi 编写的数据库程序中，其强大的功能令人叹为观止。

本篇将重点介绍 Delphi 数据库控件与界面设计技术，关系到数据库表的连接、显示、动态编辑、分类查询、报表生成等技术及相关控件的使用，同时还为读者提供了 Delphi 通用界面设计与数据库系统设计的一些常用技巧，使读者对 Delphi 数据库控件与界面设计技术有较深刻的了解。

本部分主要有以下精彩内容：

- 将控件植入 DBGrid 的网格中；
- 用 TreeView 控件实现数据库的分类查询；
- 用 Delphi 的 QReport 部件生成报表；
- TSession 组件的应用。

实例 1　将控件植入 DBGrid 的 Cell 中

实例效果

本例演示如何将一些用于辅助数据显示的控件（如 ComboBox DBMemo 控件）插入到 DBGrid 网格中。由于在默认 DBGrid 控件中不显示 MEMO 字段的内容，应用本方法可以显示 MEMO 字段。当改变数据指针时，相应控件即出现在当前所在数据指针的位置。运行界面如图 1-1、图 1-2 所示。

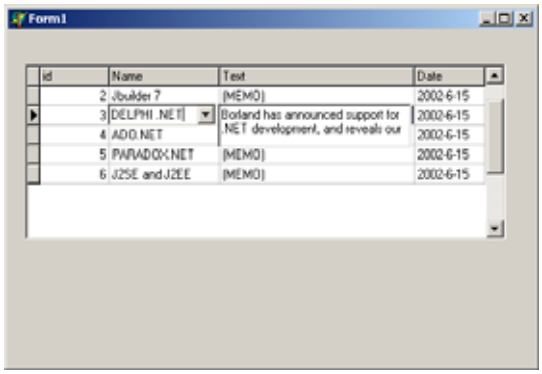

图 1-1　文本框控件在 DBGrid 单元格中

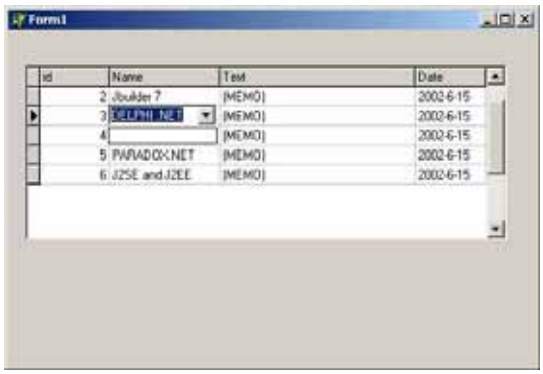

图 1-2　ComboBox 控件在 DBGrid 单元格中

程序说明

在本例中，笔者使用的是 MicroSoft Access 2002 建立的数据库，命名为 Sample.mdb（位于本书光盘根目录下），通过 Delphi 中的 ADO 控件连接数据源，表 MainTable 的字段设计如图 1-3 所示。

字段名称	数据类型
id	数字
Name	文本
Text	备注
Date	日期/时间
Money	文本
NewField	文本
Photo	OLE 对象

图 1-3　字段设计

本例中使用到一些通常的感知数据控件，如 Data Source 控件（Data Controls 组件面板），可以直观快捷地显示指定字段范围内的数据。Data Controls 组件面板如图 1-4 所示。



图 1-4　Data Controls 组件面板

在后面的实例中还会介绍使用 API 函数在 DBGrid 中插入可视控件及弹出菜单的方法，读者可以比较一下它们的异同。

制作步骤

一、窗体的建立

1.单击 File /New Application 菜单项新建一个工程，增加 Label、Edit、Button 控件（Standard 组件页）若干，分别设置各控件的 Caption 属性和布局，增加 ADOConnection1 控件（ADO 组件页）、ADOTable1 控件（ADO 组件页）、DataSource1 控件（DataAccess 组件页）和 DBGrid1 控件（DataControls 组件页）。增加 DBMemo1 和 DBComboBox1（DataControls 组件页）控件用于数据字段内容显示，稍后此二控件将被安排到网格中显示，完成后的界面如图 1-5 所示。

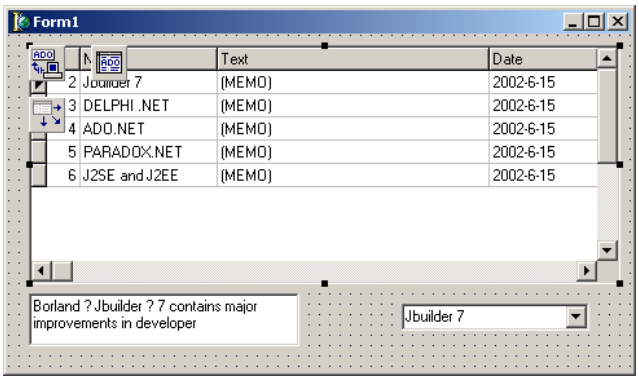

图 1-5　基本窗体设计期界面图

2．设置 ADOConnection1 的数据库连接（实现方法略，参照实例 19 制作步骤中的 ADO 连接数据库），设置 ADOTable1 的 Connection 属性为 ADOConnection1，设置 DataSource1 的 DataSet 属性为 ADOTable1。

3．为 DBGrid 控件添加 DrawDataCell 事件。

```
procedure TForm1.DBGrid1DrawDataCell(Sender: TObject; const Rect: TRect;
  Field: TField; State: TGridDrawState);
begin
  if (gdFocused in State) then
  begin
    if (Field.FieldName = DBMemo1.DataField) then
    begin
      {显示 DBMemo1}
      DBMemo1.Left := Rect.Left + DBGrid1.Left;
      DBMemo1.Top := Rect.Top + DBGrid1.top;
      DBMemo1.Width := Rect.Right - Rect.Left;
      DBMemo1.Visible := True;
    end;
    if (Field.FieldName = DBComboBox1.DataField) then
```

```
        begin
          {显示 DBComboBox1}
          DBComboBox1.Left := Rect.Left + DBGrid1.Left;
          DBComboBox1.Top := Rect.Top + DBGrid1.top;
          DBComboBox1.Width := Rect.Right - Rect.Left;
          DBComboBox1.Visible := True;
        end;
    end;
end;
```

4．编译完成。

二、源代码与注释

1．主文件代码

```
unit Unit1;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, Grids, DBGrids, DB, ADODB, StdCtrls, DBCtrls;
type
    TForm1 = class(TForm)
        ADOConnection1: TADOConnection;
        ADOTable1: TADOTable;
        DataSource1: TDataSource;
        DBGrid1: TDBGrid;
        DBMemo1: TDBMemo;
        DBComboBox1: TDBComboBox;
        procedure DBGrid1DrawDataCell(Sender: TObject; const Rect: TRect;
            Field: TField; State: TGridDrawState);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form1: TForm1;

implementation
```


需要全本请访问 www.ertongbook.com

```
{$R *.dfm}

procedure TForm1.DBGrid1DrawDataCell(Sender: TObject; const Rect: TRect;
  Field: TField; State: TGridDrawState);
begin
  if (gdFocused in State) then
  begin
    if (Field.FieldName = DBMemo1.DataField) then
    begin
      {显示 DBMemo1}
      DBMemo1.Left := Rect.Left + DBGrid1.Left;
      DBMemo1.Top := Rect.Top + DBGrid1.top;
      DBMemo1.Width := Rect.Right - Rect.Left;
      DBMemo1.Visible := True;
    end;
    if (Field.FieldName = DBComboBox1.DataField) then
    begin
      {显示 DBComboBox1}
      DBComboBox1.Left := Rect.Left + DBGrid1.Left;
      DBComboBox1.Top := Rect.Top + DBGrid1.top;
      DBComboBox1.Width := Rect.Right - Rect.Left;
      DBComboBox1.Visible := True;
    end;
  end;
end;

end.
```

完整编译代码略，详见本书光盘。

2．相关代码解释

函数 TGridDrawState = set of (gdSelected, gdFocused, gdFixed)的参数说明如表 1-1 所示。

表 1-1　　TGridDrawState 参数属性

值	说明
gdSelected	当前 Cell 被选择
gdFocused	当前 Cell 为输入焦点
gdFixed	决定 Cell 接不接收焦点

实例 2　让 DBGrid 自动调整宽度

实例效果

在 DBGrid 中的字段默认显示通常为横跨整个窗体，因此必然需要滚动条。本例演示如何让 DBGrid 和窗体同时根据字段的显示宽度而改变自身宽度，运行后界面如图 2-1 所示。

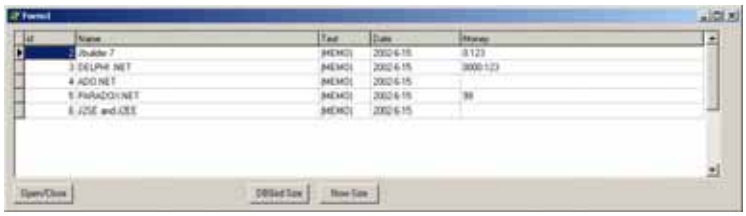

图 2-1　运行界面图

程序说明

本例中引用了两种不同方法统计宽度值：

1．Showmessage(IntToStr(iCalcGridWidth(DBGrid1)));//统计 DBGrid 所需宽度方法

通过 function TForm1.iCalcGridWidth(dbg: TDBGrid): integer;取得程序默认字体显示宽度；通过 function TForm1.NewTextWidth(fntFont: TFont; const sString: OpenString): integer;函数值作为计算基准，在函数中根据字段显示所需宽度加以统计，得出结果值。

2．在窗体建立之前（代码在 FormCreate 中给出），从显示 DBGrid 的 DBGrid1.Columns.Count 属性中遍历数据列，累加宽度。

此外，Showmessage(Inttostr(DBGrid1.Width))用来显示当前 DBGrid 宽度值。

制作步骤

一、窗体的建立

1.单击 File /New Application 菜单项新建一个工程，增加 Label、Edit、Button 控件（Standard 组件页）若干，分别设置各控件的 Caption 属性和布局，增加 ADOConnection1 控件（ADO 组件页）、ADOTable1 控件（ADO 组件页），增加 DataSource1 控件（DataAccess 组件页）和 DBGrid1 控件（DataControls 组件页），完成后的界面如图 2-2 所示。

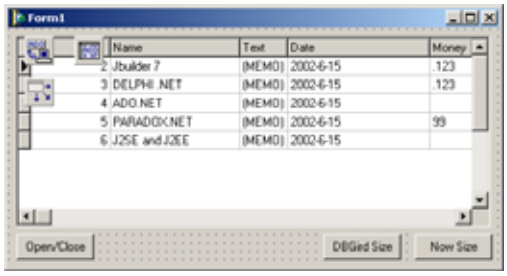

图 2-2　设计期界面图

2．设置 ADOConnection1 的数据库连接（略），设置 ADOTable1 的 Connection 属性为 ADOConnection1，设置 DataSource1 的 DataSet 属性为 ADOTable1。

3．为打开关闭数据库按钮添加代码：

```
procedure TForm1.Button3Click(Sender: TObject);
begin
  ADOTable1.Active := not ADOTable1.Active; //打开关闭数据库显示
end;
```

4．为 Form 添加 FormCreat 事件来调用函数。

```
procedure TForm1.FormCreate(Sender: TObject);
var
  W, i: Integer;
begin
  ADOTable1.Active := True;
  W := 0;
  for i := 0 to DBGrid1.Columns.Count - 1 do //遍历数据列累加宽度
    W := W + DBGrid1.Columns[i].Width + 4; //增加必要的值协调右边框余值
  DBGrid1.ClientWidth := W; //设置宽度
  Self.ClientWidth := (DBGrid1.Left * 2) + DBGrid1.Width; //调整窗体宽度
end;
```

5．编译完成。

二、源代码与注释

1．主文件代码

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, DBGrids, DB, ADODB, StdCtrls, DBCtrls;
type
  TForm1 = class(TForm)
    ADOConnection1: TADOConnection;
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    procedure FormCreate(Sender: TObject);
```

```
    function NewTextWidth(fntFont: TFont; const sString: OpenString): integer;
    function iCalcGridWidth(dbg: TDBGrid): integer;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
function TForm1.NewTextWidth(fntFont: TFont; const sString: OpenString): integer;
{根据程序默认系统取得文字所需宽度值}
var
  fntSave: TFont;
begin
  result := 0;
  fntSave := Application.MainForm.Font;
  Application.MainForm.Font := fntFont;
  try
    result := Application.MainForm.Canvas.TextWidth(sString);
  finally
    Application.MainForm.Font := fntSave;
  end;
end;

function TForm1.iCalcGridWidth(dbg: TDBGrid): integer;
{指定 TDBGrid 名称计算所需显示宽度}
const
  cMEASURE_CHAR = '0';
  iEXTRA_COL_PIX = 4;
  iINDICATOR_WIDE = 11;
var
  i, iColumns, iColWidth, iTitleWidth, iCharWidth: integer;
  //分别代表字段数,列数,列宽,列标题宽,字宽
begin
```

```
    iColumns := 0;
    result := GetSystemMetrics(SM_CXVSCROLL);
    //得到系统滚动条的大小
    iCharWidth := NewTextWidth(dbg.Font, cMEASURE_CHAR);
    with dbg.dataSource.dataSet do
      for i := 0 to FieldCount - 1 do
        with Fields[i] do
          if visible then{字段是否可视}
          begin
            iColWidth := iCharWidth * DisplayWidth;
            if dgTitles in dbg.Options then
            begin
              iTitleWidth := NewTextWidth(dbg.TitleFont, DisplayLabel);
              if iColWidth < iTitleWidth then
                iColWidth := iTitleWidth;
            end;
            inc(iColumns, 1);
            inc(result, iColWidth + iEXTRA_COL_PIX);
          end;
          if dgIndicator in dbg.Options then
          begin
            inc(iColumns, 1);
            inc(result, iINDICATOR_WIDE);
          end;
          if dgColLines in dbg.Options then
            inc(result, iColumns)
          else
            inc(result, 1);
  end;

  procedure TForm1.FormCreate(Sender: TObject);
  var
    W, i: Integer;
  begin
    ADOTable1.Active := True;
    W := 0;
    for i := 0 to DBGrid1.Columns.Count - 1 do //遍历数据列累加宽度
      W := W + DBGrid1.Columns[i].Width + 4; //增加必要的值协调右边框余值
    DBGrid1.ClientWidth := W; //设置宽度
```

```
    Self.ClientWidth := (DBGrid1.Left * 2) + DBGrid1.Width; //调整窗体宽度
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
    Showmessage(IntToStr(iCalcGridWidth(DBGrid1)));//统计 DBGrid 所需宽度方法
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
    Showmessage(Inttostr(DBGrid1.Width));//显示当前 DBGrid 宽度
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    ADOTable1.Active := not ADOTable1.Active; //打开关闭数据库显示
end;
end.
```

完整编译代码略，详见本书光盘。

2．相关代码解释

函数 GetSystemMetrics：用来取得系统滚动条的大小。

调用格式：GetSystemMetrics(SM_CXVSCROLL)，如：

```
var
  ScrollSize :Integer;
Begin
  ScrollSize := GetSystemMetrics(SM_CXVSCROLL);
  ShowMessage(IntToStr(ScrollSize));
end;
```

这个值允许用户在注册表中改变键值：

HKEY_CURRENT_USER\Control Panel\Desktop\WindowMetrics，如图 2-3 所示。



图 2-3　在注册表中改变键值

需要全本请访问 www.ertongbook.com