

计算机高级编程技术



第6章 数据访问组件





概况

- ❑ Delphi组件板上的数据访问组件，提供了应用程序与数据库之间的联系。
- ❑ Delphi通过ADO数据集访问组件，可以通过微软的OLE来访问更为广泛的数据库中的数据。





目录

- ❑ 6.1 数据访问组件简介
- ❑ 6.2 TTable组件介绍
- ❑ 6.3 TQuery组件介绍
- ❑ 6.4 TStoredProc组件介绍
- ❑ 6.5 TDataSource组件介绍
- ❑ 6.6 TDatabase组件介绍
- ❑ 6.7 TSession组件介绍
- ❑ 6.8 其他组件介绍
- ❑ 6.9 数据访问组件的应用实例
- ❑ 6.10 ADO组件页介绍





6.1 数据访问组件简介

在数据访问组件中，**TTable**、**TQuery**和**TStoreProc**三个组件是用来联系应用程序和数据库信息的，而**TDataSource**组件是用来联系数据库信息和数据控制组件的。





6.2 TTable组件介绍

- TTable组件是Delphi开发数据库应用程序中最常用的组件。它在应用程序访问数据库时起着极其重要的作用。
 - 6.2.1 数据集组件的基本知识
 - 6.2.2 TTable组件的重要属性
 - 6.2.3 TTable组件的重要方法
 - 6.2.4 TTable组件的重要事件
 - 6.2.5 举例说明





6.2.1 数据集组件的基本知识

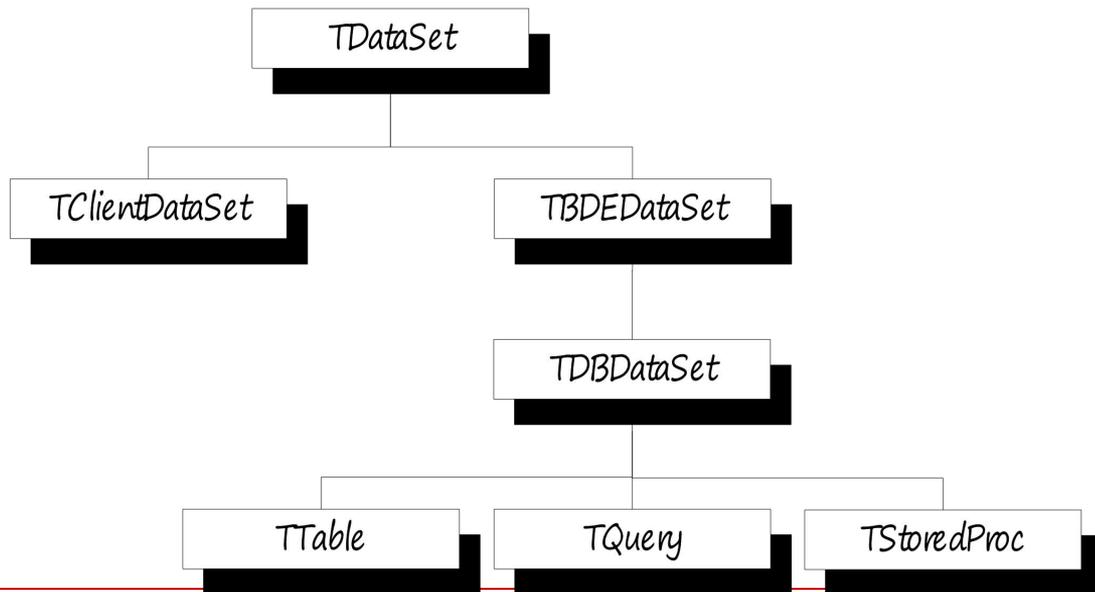
在Delphi中，访问数据的基本单元是数据集对象。应用程序正是提供数据集组件来访问数据库的。一个数据集对象就代表了数据库的一张表格，或者是访问数据库的一个查询或存储的过程。





6.2.1 数据集组件的基本知识

- 图中说明了所有数据集组件的继承关系。





6.2.2 TTable组件的重要属性

❑ 1. DatabaseName

DatabaseName属性是用来说明当前数据集的来源，即应用程序所利用的数据库的名字。

❑ 2. TableStyle

TableStyle属性是用来说明当前应用程序所操作的数据库表格的类型，





6.2.2 TTable组件的重要属性

□ 3. TableName

TableName是TTable最重要的属性之一。它是用来说明这个TTable组件所对应的是数据库中的哪一张数据库表格。

□ 4. Active

Active属性是用来说明数据库文件的打开状态。应用Active属性来决定及设定一个数据集组件与数据库数据之间的联系。





6.2.2 TTable组件的重要属性

□ 5. BOF和EOf

BOF（beginning of file）属性值为一个布尔值，表明当前记录指针所处的位置是否为数据集的第一个记录。

EOf（end of file）属性值为一个布尔值，表明当前记录指针所处的位置是否为数据集的最后一个记录。





6.2.2 TTable组件的重要属性

□ 6. DBHandle与Handle

DBHandle与Handle属性允许应用程序直接调用BDE的API函数。

只有在包含数据集和数据库的应用程序直接调用BDE API函数时才会用到DBHandle .

只有在应用TdataSet方法直接调用BDE API函数时才会用到Handle属性。





6.2.2 TTable组件的重要属性

□ 7. CanModify与Modified

CanModify属性用一个布尔值来表明应用程序是否可以更改数据集。

Modified属性用一个布尔值来表明数据集是否被修改。





6.2.2 TTable组件的重要属性

□ 8. Exclusive

Exclusive属性是用来标识当前应用程序是否拥有打开一张Paradox或dBASE这样的数据库表格的专有权。

□ 9. Fields

Fields属性是用来指出数据集的字段列表。应用**Fields**属性可以访问数据库表格中的字段组合。





6.2.2 TTable组件的重要属性

□ 10. Filter与Filtered

Filter属性允许用户定义一个数据集过滤器。当过滤被数据集应用时，只有那些满足过滤器条件的记录才会被显示。

Filtered属性是用来表明数据集的过滤是否被激活。通过检测**Filtered**属性值来确定一个数据集的过滤是否有效。





6.2.2 TTable组件的重要属性

□ 11. IndexFieldNames

IndexFieldNames属性显示了表格索引的字段名。

□ 12. IndexFields

IndexFields属性是数据库中的字段名列表。





6.2.2 TTable组件的重要属性

□ 13. IndexName

IndexFieldNames属性显示了表格索引的字段名。

□ 14. MasterFields和MasterSource

MasterFields用来在主表中指定一个或多个对应当前表格中的字段。

MasterSource属性指在主从关系的表格中，主表的datasource组件的名称。





6.2.2 TTable组件的重要属性

□ 15. UpdateMode

UpdateMode属性是数据集组件都支持的SQL的特殊属性，UpdateMode决定了BDE如何在多用户环境中确认一个正在被更新的记录。





6.2.3 TTable组件的重要方法

□ 1.CreateTable方法

CreateTable方法是用来应用新的结构信息建立一个新表格的方法。在应用程序运行时调用**CreateTable**会产生一个利用数据集当前定义建立的表格。





6.2.3 TTable组件的重要方法

- 2.数据集打开和关闭的方法：Open/Close
 - 调用Open方法将Active属性设置为True。只有当Active属性为True时，可以从数据库中读取数据或向数据库中写数据。
 - 调用Close设置Active属性值为False。当Active属性值为False时，数据集将被关闭，无法对数据库进行读/写操作。





6.2.3 TTable组件的重要方法

- 3. 设定数据库检索范围的方法
 - ApplyRange方法
 - CancelRange方法
 - EditRangeEnd方法
 - EditRangeStart方法
 - SetRange方法
 - SetRangeEnd方法
 - SetRangeStart方法





6.2.3 TTable组件的重要方法

- 4. 浏览数据库中数据的方法
 - First方法
 - Last方法
 - MoveBy方法
 - Next方法
 - Prior方法





6.2.3 TTable组件的重要方法

- 5. 在数据库中查询记录的方法
 - EditKey方法
 - FieldByName方法
 - FindKey方法
 - GotoKey方法
 - GotoNearest方法
 - Locate方法
 - SetKey方法





6.2.3 TTable组件的重要方法

- 6. 编辑修改数据库数据的方法
 - Append方法
 - Cancel方法
 - Delete方法
 - Edit方法
 - Insert方法
 - InsertRecord方法
 - Post方法





6.2.3 TTable组件的重要方法

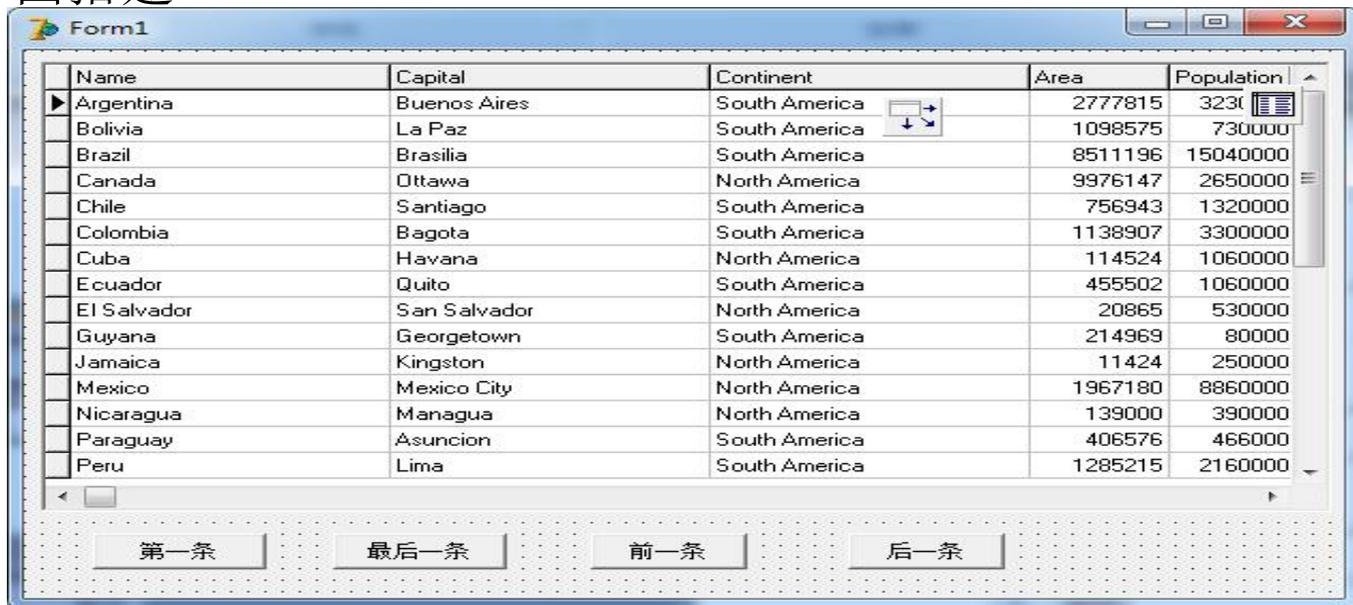
- 7. 链接数据控制组件的方法
 - DisableControls方法
 - EnableControls方法
 - Refresh方法





6.2.4 TTable组件的应用示例

1. 界面搭建





6.2.4 TTable组件的应用示例

2. 控件设置

控件名	设置	备注
Ttable1	DatabaseName: DBDEMOS TableName: country.db	
DataSource1	DataSet: Table1	
DBGrid1	DataSource: DataSource1	
Button		





6.2.4 TTable组件的应用示例

3. 主要代码

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    table1.First; //第一条  
end;  
  
procedure TForm1.Button2Click(Sender: TObject);  
begin  
    table1.Last; //最后一条  
end;
```





6.2.4 TTable组件的应用示例

□ 3. 主要代码

```
procedure TForm1.Button3Click(Sender: TObject);  
begin  
    table1.Prior; //前一条  
end;  
  
procedure TForm1.Button4Click(Sender: TObject);  
begin  
    table1.Next; //后一条  
end;
```





6.2.4 TTable组件的应用示例

□ 4. 运行结果

The screenshot shows a Java Swing window titled "Form1" containing a TTable component. The table displays the following data:

Name	Capital	Continent	Area	Population
Argentina	Buenos Aires	South America	2777815	3230000
Bolivia	La Paz	South America	1098575	730000
Brazil	Brasilia	South America	8511196	15040000
Canada	Ottawa	North America	9976147	2650000
Chile	Santiago	South America	756943	1320000
Colombia	Bagota	South America	1138907	3300000
Cuba	Havana	North America	114524	1060000
Ecuador	Quito	South America	455502	1060000
El Salvador	San Salvador	North America	20865	530000
Guyana	Georgetown	South America	214969	80000
Jamaica	Kingston	North America	11424	250000
Mexico	Mexico City	North America	1967180	8860000
Nicaragua	Managua	North America	139000	390000
Paraguay	Asuncion	South America	406576	466000
Peru	Lima	South America	1285215	2160000

Below the table, there are four navigation buttons: "第一条" (First), "最后一条" (Last), "前一条" (Previous), and "后一条" (Next).





6.3 TQuery组件介绍

- TQuery是一个建立在SQL基础上，专门用于对数据库中的数据进行查询的组件。
 - 6.3.1 TQuery组件的基本知识
 - 6.3.2 TQuery组件的重要属性
 - 6.3.3 TQuery组件的重要方法





6.3.1 TQuery组件的基本知识

- TQuery组件和TTable组件同属于数据集组件，所以数据集组件具有的属性、方法和事件，TQuery组件都具有。





6.3.2 TQuery组件的基本属性

□ 1. SQL

SQL属性是用来设置在TQuery组件调用ExecSQL或Open方法执行查询时所需的选择条件的。

□ 2. DataSource

设置DataSource属性可以自动地将另一个数据集的字段值赋给当前查询的参数，但这些参数必须具有与该数据集相同的字段名。





6.3.2 TQuery组件的基本属性

□ 3. Params和ParamCount

Params来动态地观察和设置参数名、参数值和数据类型，在设计阶段可以通过编辑Params属性来设置参数的形式。

ParamCount的值表示Params属性中有多少参数。





6.3.2 TQuery组件的基本属性

□ 4.Prepared

通过检查Prepared的属性值来确定是否准备好查询。

□ 5. RequestLive

设置RequestLive属性来说明执行查询时，是否允许BDE即时向应用程序返回对数据结果的修改。





6.3.3 TQuery组件的基本方法

□ 1. Create

调用Create方法在运行时开始一个查询，TQuery组件会自动地放置在窗体或数据模块中。

□ 2. Destroy

调用Destroy方法来取消与服务器之间的联系，撤销SQL语句列表、参数列表、数据联系以及SQL的存储区域。





6.3.3 TQuery组件的基本方法

□ 3. ExecSQL

调用ExecSQL方法来执行当前SQL属性中的SQL查询语句。
ExecSQL用于不需要返回记录指针的数据访问。

□ 4. ParamByName

调用ParamByName方法根据参数名来设置和使用参数信息。





6.3.3 TQuery组件的基本方法

□ 5. Prepare

调用Prepare方法让BDE和远程的数据库服务器为查询分配资源并执行一些优化操作，称为执行查询前的准备工作。

□ 6. UnPrepare

执行查询的准备工作会消耗一定的资源，所以在完成查询以后应该调用UnPrepare方法来释放分配给查询的资源。





6.4 TStoredProc组件介绍

- TStoredProc组件与TQuery组件、TTable组件一样，属于数据集组件。TStoredProc组件用来完成数据库服务器上的存储过程。
 - 6.4.1 TStoredProc组件的基本知识
 - 6.4.2 TStoredProc组件的重要属性
 - 6.4.3 TStoredProc组件的重要方法





6.4.1 TStoredProc组件的基本知识

- 一个存储过程是一系列作为服务器的一部分存储的表达式。存储过程执行与数据库相关的重复性任务，并将结果传递给用户。
- 使用存储过程，可以提高数据库应用程序的效率：
 - 充分利用服务器的处理能力和处理速度。
 - 因处理过程在数据存放的服务器上进行，减少了网络传输的工作量。





6.4.2 TStoredProc组件的基本属性

□ 1. Overload

利用Overload属性来说明在一个Oracle服务器上执行哪一种重载的存储过程。

□ 2. StoredProcName

通过设置StoredProcName属性来指明服务器调用的存储过程名。





6.4.2 TStoredProc组件的基本属性

□ 3. Params与ParamCount

Params属性是用来储存存储过程中输入和输出的参数的。

ParamCount的属性值可以知道当前Params属性中存储参数的数目





6.4.3 TStoredProc组件的基本方法

□ 1. Create

在应用程序中通过调用**Create**方法来初始化一个已声明的存储过程。

□ 2. ExecProc

ExecProc方法是用来执行服务器上的存储过程的。





6.4.3 TStoredProc组件的基本方法

□ 3. GetResults

调用GetResults方法使一个Sybase存储过程返回给客户结果。

□ 4. ParamByName

调用ParamByName方法会根据参数名获取相应的参数信息。





6.4.3 TStoredProc组件的基本方法

□ 5. Prepare

调用Prepare方法是用来在调用ExecProc方法开始存储过程之前建立参数间的联系的。





6.5 TDataSource组件介绍

- TDataSource组件提供了联系数据集组件（TTable组件、TQuery组件和TStoredProc组件）与数据控制组件（TDBGrid组件、TDBNavigator组件）的纽带。
 - 6.5.1 TStoredProc组件的基本属性
 - 6.5.2 TStoredProc组件的重要方法
 - 6.5.3 TStoredProc组件的重要事件





6.5.1 TDataSource组件的基本属性

□ 1. AutoEdit

AutoEdit属性决定了当与一个数据源组件相连的数据控制组件获得焦点时，此数据源组件是否自动地调用数据集组件的Edit方法

□ 2. DataSet

DataSet属性是用来指明一个与当前数据源组件相联系的已存数据集组件的名字。





6.5.1 TDataSource组件的基本属性

□ 3. Enabled

Enabled属性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。

□ 4. State

State属性读取与此数据源组件相连的数据集组件的当前状态。





6.5.2 TDataSource组件的基本方法

□ 1. Create

调用Create方法在运行时建立一个TDataSource组件。

□ 2. Destroy

调用Destroy方法将OnStateChange事件置为空，将与数据源组件相连的数据集置为空，同时撤销数据链接列表。





6.5.2 TDataSource组件的基本方法

□ 3. Edit

调用**Edit**方法允许用户对与该数据源组件相连的数据集进行编辑修改。

□ 4. IsLinkedTo

调用**IsLinkedTo**方法来证明数据源组件与一个特定的数据集相连。





6.5.3 TDataSource组件的基本事件

□ 1. onDataChange

onDataChange事件发生在与该数据源组件相连的数据集的当前记录被编辑后，应用程序转移到其他字段或者转移到其他记录时，也就是提交被编辑的当前记录的时候。

□ 2. onStateChange

onStateChange事件发生在与该数据源组件相连的数据集的状态改变时。





6.5.3 TDataSource组件的基本事件

□ 3. OnUpdateData

OnUpdateData事件发生在向数据库提交对数据的改变之前。

主要作用是执行附加的数据处理和确认操作。





6.6 TDatabase组件介绍

- TDatabase组件在应用程序中通过建立与数据库之间的联系对数据库进行控制。
 - 6.6.1 TDatabase组件的基本属性
 - 6.6.2 TDatabase组件的重要方法
 - 6.6.3 TDatabase组件的重要事件





6.6.1 TDatabase组件的基本属性

□ 1. AliasName

AliasName是用来指明连接中的数据库的**BDE**别名的。这个数据库的**BDE**别名必须是已经存在的。**BDE**别名里包含了连接需要的数据库的特征信息。

□ 2. DatabaseName

DatabaseName属性用来指明一个**TDatabase**组件使用的数据库名。





6.6.1 TDataSource组件的基本属性

□ 3. DataSets

应用程序可以通过使用DataSets属性来访问与该TDatabase组件相连的所有处于打开状态的数据集。

□ 4.DriverName

应用程序使用DriverName来指明联系中的BDE驱动器名。





6.6.1 TDatabase组件的基本属性

□ 5. KeepConnection

KeepConnection属性用来说明在当前没有数据集处于打开状态时，应用程序是否保持与数据库的联系。

□ 6. Connected

设置Connected属性值为True可以在不打开数据集的情况下建立一个数据库联系。





6.6.1 TDatabase组件的基本属性

□ 5. KeepConnection

KeepConnection属性用来说明在当前没有数据集处于打开状态时，应用程序是否保持与数据库的联系。

□ 6. Connected

设置Connected属性值为True可以在不打开数据集的情况下建立一个数据库联系。





6.6.1 TDatabase组件的基本属性

□ 7.LoginPrompt

LoginPrompt属性是用来控制远程数据库联系的登录方法的。

□ 8. TransIsolation

TransIsolation用来说明Borland Database Engine (BDE) 控制database事务的独立等级。





6.6.2 TDatabase组件的基本方法

□ 1. ApplyUpdates

调用ApplyUpdates方法向数据库服务器提交当前数据集的缓存更新。

□ 2. Close

调用Close方法来中断与数据库服务器的联系，并释放系统分配给联系的资源。





6.6.2 TDatabase组件的基本方法

□ 3. Commit

永久地保存当前事务对数据的更新、插入或删除的操作，并结束当前的事务。

□ 4. Rollback

撤销当前事务对数据的更新、插入或删除的操作，并结束当前的事务。





6.6.2 TDatabase组件的基本方法

□ 5.StartTransaction

在数据库服务器上开始一个新的事务。





6.7 TSession组件介绍

- TSession用来实现在一个应用程序中控制与数据库的联系。我们无法显式地创建一个TSession组件，也无法观察到它，但我们在应用程序中却可以使用它。
 - 6.7.1 TDatabase组件的基本属性
 - 6.7.2 TDatabase组件的重要方法





6.7.1 TSession组件的基本属性

□ 1. DatabaseCount

通过检查DatabaseCount属性值来获悉与此TSession组件相连的活动数据库的数目。

□ 2. Database

通过使用Databases属性来访问与此TSession组件相连的处于活动状态的数据库组件，即当前与数据库服务器相连的数据库组件。





6.7.1 TSession组件的基本属性

□ 3. KeepConnections

说明在没有任何活动的数据集与数据库组件相连时，该数据库组件是否与数据库服务器保持联系。

□ 4. Database

通过使用Databases属性来访问与此TSession组件相连的处于活动状态的数据库组件。





6.7.2 TSession组件的基本方法

□ 1. GetAliasNames

用来获取一个已存在的**BDE**数据库别名的列表。

□ 2. GetAliasParams

用来获得与一个**BDE**数据库别名有关的参数信息。





6.7.2 TSession组件的基本方法

□ 3. GetDatabaseNames

用来获得与BDE数据库别名和所有TDatabases组件的名字。

□ 4. GetDriverNames

获得此TSession组件可以使用的所有BDE驱动器。





6.7.2 TSession组件的基本方法

- 5. GetDriverParams

用来获取一个指定的BDE驱动器的参数信息。

- 6. GetTableNames

获得与数据库组件相连的所有表格的名称。





6.8 其他组件介绍

- 1. TBatchMove组件
 - TBatchMove允许应用程序对数据库的一组记录或整个表格进行操作。
 - 通过设置TBatchMove组件的属性指定需要的操作，调用Execute方法来执行操作。





6.8 其他组件介绍

□ 2. TUpdateSQL组件

- TUpdateSQL组件在允许缓存更新时提供一个更新只读数据集的对象。利用TUpdateSQL对象可以在允许缓存更新时，提供SQL语句来更新由TQuery组件表现的只读数据集。





6.8 其他组件介绍

□ 3. TNestedTable组件

- 通过TNestedTable组件可以访问嵌套数据集中的数据。

TNestedTable组件从TBDEDataSet组件那里继承了BDE的功能，所以可以使用BDE访问嵌套表格中的数据。





6.9 数据访问组件的应用实例

- 【例6-1】下面结合一个实际的例子体会数据源组件的作用。
- 1. 建立新的应用程序项目
 - 在Delphi中建立一个默认的应用程序项目。
- 2. 设计窗体
 - 向窗体中加入组件，如图所示。
 - 为了清楚地显示运行结果，在
 - 窗体中加入了一个数据控制组件TDBGrid。





6.9 数据访问组件的应用实例

Form1

请选择数据库名:

请选择数据表格:

显示字段名:

开始查询

上一条记录

下一条记录

结束应用程序





6.9 数据访问组件的应用实例

□ 3. 通过编写程序代码完成以下功能：

① 在组合框中显示数据库别名的列表，使用的是 `TSession` 组件的 `GetAliasNames` 方法。在组合框中选取数据库别名，列表框1中显示该数据库中的数据表格名列表，使用的是 `TSession` 组件的 `GetTableNames` 方法。





6.9 数据访问组件的应用实例

- ② 在列表框1中选取数据表格名，列表框2中显示该数据表格使用的字段名列表，使用的是Ttable组件的FieldList属性。
- ③ 在文本框中输入要查找的字段值，单击查询按钮，若数据表格中存在该记录，则显示查询成功，并将记录指针移至该记录处；如果数据表格中不存在该记录，则显示查询失败的提示信息。
- ④ 通过控制按钮可以实现记录指针的上下移动，查询结束时按结束按钮退出应用程序。





6.9 数据访问组件的应用实例

```
procedure TForm1.FormCreate(Sender:TObject);
begin
  with combobox1 do
  begin
    Items.clear;
    {调用Tsession组件的GetAliasNames方法获取数据库别名的列表}
    Session.GetAliasNames(Items);
    Text:=Items[0];
  end;
```





6.9 数据访问组件的应用实例

```
procedure TForm1.ComboBox1Click(Sender:TObject);
begin
  str1:=combobox1.Items.Strings[combobox1.itemindex];
  with listbox1 do
  begin
    Items.Clear;
    {调用Tsession组件的GetTableNames方法获取数据表格名的列表}
    Session.GetTableNames(str1,",",true,true,listbox1.Items);
  end;
end;
```





6.9 数据访问组件的应用实例

```
procedure TForm1.Button3Click(Sender:TObject);  
begin  
    {关闭应用程序}  
    application.Terminate;  
end;  
procedure TForm1.ListBox1Click(Sender:TObject);  
begin
```





6.9 数据访问组件的应用实例

```
with table1 do
begin
close;
{设置Table1组件的DatabaseName属性和TableName属性}
DatabaseName:=str1;
TableName:=listbox1.items.Strings[listbox1.itemindex];
open;
end;
```





6.9 数据访问组件的应用实例

```
with listbox2 do
begin
  Items.Clear;
  {使用Table1组件的FieldList属性获得字段列表}
  items:=table1.FieldList;
  str2:=Items.Strings[0];
  label4.Caption:='请输入要查询的'+str2+'字段值:'
end;
end;
```





6.9 数据访问组件的应用实例

```
procedure TForm1.Button1Click(Sender:TObject);
begin
  {调用Table1组件的Prior方法移动记录指针}
  table1.Prior;
end;

procedure TForm1.Button2Click(Sender:TObject);
begin
  {调用Table1组件的Next方法移动记录指针}
  table1.Next;
end;
```





6.9 数据访问组件的应用实例

```
procedure TForm1.Button4Click(Sender:TObject);
begin
  with Table1 do
  begin
    close;
    IndexFieldNames:=str2; {设置索引字段}
    open;
    {调用Table1组件的Findkey方法在数据集中查询记录}
    if Findkey([edit1.text]) then
    begin
      label5.Caption:='查找成功!';
    end
  end
end
```





6.9 数据访问组件的应用实例

```
begin
    label5.Caption:='查找失败!';
end;
end;
end;

if Findkey([edit1.text]) then
begin
    label5.Caption:='查找成功!';
end
Else
begin
    label5.Caption:='查找失败!';
end;
end;
end;
```





6.9 数据访问组件的应用实例

Form1

请选择数据库名: DBDEMOS

请选择数据表格:

- animals.dbf
- biolife.db
- clients.dbf
- country.dbf
- custoly.db
- customer.db
- employee.db
- events.db
- holdings.dbf
- industry.dbf

显示字段名:

- CustNo
- Company
- Addr1
- Addr2
- City
- State

请输入要查询的 CustNo 字段值: 4684

查找成功!

CustNo	Company	Addr1
4531	On-Target SCUBA	7-73763 Nanakawa Road
4652	Jamaica Sun, Inc.	PO Box 643
4684	Underwater Fantasy	PO Box 842
5132	Princess Island SCUBA	PO Box 32 Waiyevo
5151	Central Underwater Supplies	PO Box 737
5163	Safari Under the Sea	PO Box 7456





6.10 ADO组件页介绍

- Delphi中的ADO数据访问组件页如图所示。ADO组件包含了ADO结构的各项功能。





6.10 ADO组件页介绍

- 使用ADO和ADO组件允许程序员使用ADO代替原来的Borland数据库引擎（BDE）来访问数据。
 - 6.10.1 ADO组件页概述
 - 6.10.2 TADOConnection对象
 - 6.10.3 TADOCommand组件
 - 6.10.4 TADODataSet组件





6.10.1 ADO组件页概述

□ 1. AliasName

AliasName是用来指明连接中的数据库的BDE别名的。这个数据库的BDE别名必须是已经存在的。BDE别名里包含了连接需要的数据库的特征信息。

□ 2. DatabaseName

DatabaseName属性用来指明一个TDatabase组件使用的数据库名。





6.10.2 TADOConnection的重要属性

□ 1. ConnectionString

ConnectionString是用来指明数据库的连接信息的属性。通过设置ConnectionString来指明将ADO联系组件与数据集相连的必要信息。

□ 2. Connected

Connected属性用来说明一个与数据库的联系是否被激活。





6.10.2 TADOConnection的重要属性

□ 3.ConnectionObject

提供对 A D O 联系对象的直接访问。通过设置 ConnectionObject 属性可以取得与其参照的 A D O 联系对象的直接联系。

□ 4.ConnectionTimeout

ConnectionTimeout 是用来声明连接可能需要的最长时间。如果一个联系超过了它规定的时间，它会终止联系请求，并产生一个异常。





6.10.2 TADOConnection的重要方法

□ 1.Cancel

Cancel方法是用来取消一个准备与ADO数据库建立的联系。

□ 2.Open

调用Open方法来开始建立与ConnectionString属性中指定的数据库之间的联系。它们可以在发出联系要求的同时登录用户的身份和密码信息。





6.10.3 TADOCCommand的重要属性

□ 1. CommandText

CommandText属性用来说明要使用ADO操作组件执行的命令内容。CommandText属性值是文本形式的命令，例如SQL语句、一个数据表格名或者是一个存储过程名。

□ 2. CommandType

CommandType的属性值应该与CommandText中所说明操作相一致，





6.10.3 TADOCCommand的重要属性

□ 1. CommandObject

通过设置ConnectionObject属性可以取得与其对应的ADO操作对象直接联系。通过这种访问，应用程序可以使用与其对应的ADO操作对象的方法和属性。





6.10.3 TADOCCommand的重要方法

□ 1 Execute

Execute函数是调用来立即执行CommandText中定义的操作的。

□ 2 Cancel

Cancel方法是用来取消一个操作组件准备执行的操作。





6.10.4 TADODataSet的重要属性

□ 1. ConnectionString

ConnectionString是用来指明数据库的连接信息的属性。

□ 2. CommandText属性

CommandText是一个可以用来表达语句、数据表格名或者执行的存储过程，要为该属性赋一个文本值。





6.10.4 TADODataset的重要属性

□ 3. CommandType属性

CommandType用于指明CommandText属性的类型，该属性的值应该和属性的值相一致。

□ 4. Connection属性

使用Connection属性来指明ADO组件与ADO数据库联系的方式。

