

### VB 数据库编程一

Visual Basic 作为应用程序的开发“利器”也表现在数据库应用程序的开发上，它良好的界面和强大的控件功能使数据库编程变得简单多了。但即便如此，数据库应用程序的开发仍然算得上是 VB 编程中的难点，这是因为你不仅要熟悉 VB 中关于数据库编程方面的知识（当然这是十分简单的）还要了解数据库的知识。所以我们先介绍一下数据库的基本知识，算是学习数据库编程前的热身运动吧！

#### 一、热身运动

首先需要声明是，我们这里介绍的数据库知识都是指的关系数据库。所谓关系数据库就是将数据表示为表的集合，通过建立简单表之间的关系来定义结构的一种数据库。

不管表在数据库文件中的物理存储方式如何，它都可以看作一组行和列，与电子表格的行和列类似。在关系数据库中，行被称为记录，而列则被称为字段。下面是一个客户表的例子。

表 1 客户表

客户号	姓名	地址	城市	街道	邮编
1723	Doe John	1234	Ffth Avenue	New York NY	1004
3391	Smith Mary	9876	Myrtle Lavee	Bosten MA	6078
3765	Blasel Mortimer	2296j	River Road	peoria IL	7011

此表中每一行是一个记录，它包含了特定客户的所有信息，而每个记录则包含了相同类型和数量的字段：客户号、姓名等等。

表 是一种按行与列排列的相关信息的逻辑组，类似于工作单表。

字段 数据库表中的每一列称作一个字段。表是由其包含的各种字段定义的，每个字段描述了它所拥有的数据。创建一个数据库时，须为每个字段分配一个数据类型、最大长度和其它属性。字段可包含各种字符、数字甚至图形。

记录 各个客户有关的信息存放在表的行，被称为记录。一般来说，数据库表创建时任意两个记录都不能相同。

键 键就是表中的某个字段（或多个字段），它（们）为快速检索而被索引。键可以是唯一的，也可以是非唯一的，取决于它（们）是否允许重复。唯一键可以指定为主键，用来唯一标识表的每行。例如，在前面的例子中，客户标识号（客户号）是表的主键，因为客户号唯一地标识了一个客户。

关系 数据库可以由多个表组成，表与表之间可以以不同的方式相互关联。例如，客户数据库还可以有一个包含某个客户的所有定单的表。它只用“客户号”字段来引用该定单的客户，而不在定单表中的每项重复所有客户信息，如下表所示：

表 2 定货表

定货	客户号	日期	内容	数量
14764	3391	2/23/94	27	\$22.95
14932	3391	3/17/94	46	\$9.57
15108	8765	2/15/96	27	\$22.95

在这个表中，客户号字段引用了客户表中的

客户号字段，从而把定单和客户联系起来。可以看到，客户 3391 (Mary Smith) 在 94 年 2 月 23 日订购了 27 项，在 94 年 3 月 17 日订购了 46 项。用来建立关系的键叫做外部键，因为它与“外部”表（客户表）的主键关联。

一对多和多对多关系 上表中的关系类型叫做一对多关系，因为一个客户可以发出多个定单，而某个特定的定单只能是一个客户所发。也可以建立多对多的关系。例如，列出所有可以销售的项（存货）的盘存表：

表 3 盘存表

内容	描述	供应商	费用	盘存
27	Straw Hat	Garden Supply Co.	\$14.00	50
46	Garden gloves	Garden Supply Co.	\$4.50	75
102	hanging floral industries		\$6.00	137

从盘存表中，可以看到在客户和存货项之间存在多对多的关系。也就是说，一个客户可以订购多个存货项，而一个存货项又能够被多个客户订购。多对多关系是通过两个独立的一对多关系来定义的，公共的“多”表包含了两个其它表的外部键。在该例中，定货 s 表与 盘存 表（通过“内容”）与 Customer 表（通过 客户号）都相关联。通过这三个表，我们可以看到，Mary Smith (客户号 3391) 订购了 Straw Hat (“内容” 27) 和 Garden Gloves (“内容” 46)，而 Mary Smith (客户号 3391) 和 Mortimer Blaselfatz (客户号 8765) 都订购了 Straw Hat (“内容” 27)。如果把客户表和盘存表的相关字段与 定货表的“定货”字段联结起来，建立一个“关联”表，那么这个关系就更清楚了。

表 4 关联表:按客户号和内容排序

定货号	客户号	姓名	内容	描述
14764	33391	Smith Mary	27	Straw Hat
14932	33391	Smith Mary	46	Garden Gloves
15168	8765	Blaselfatz Mortimer	27	Straw Hat

规范化 数据库设计者的任务就是组织数据，而组织数据的方法，应能消除不必要的重复，并为所有必要信息提供快速查找路径。为了达到这种目标而把信息分离到各种独立的表中去的过程，叫作规范化。

规范化是用许多指定的规则 and 不同级别的范式来进行规范的复杂过程。该过程的研讨已超出了本文的范围。但是，大多数简单数据库的规范化可以用下面简单的经验规则来完成:包含重复信息的表必须分成独立的几个表来消除重复。

例如，使学生和课程对应的学生数据库，包含了下表所示的信息。

表 5

学生	课程	描述	教授
1 4	Introduction to Physiology	Dawson	
2 3	Applied Basketweaing	Carruth	
3 1	Physics for Short-定货	cooks Adms	
4 2	Introduction to Physiology	Dawsons	

如果有选学了十二门课程的 1000 个学生，每门课程的说明和教师将显示 100 多次—

## VB 数据库编程

对选了那门课程的每个学生都要重复一次。要避免这种低效率，应当把表分成两个独立的表来规范化，一个用来表示学生，另一个用来表示课程，如表 6，表 7 所示。

学生 课程

1 4

2 3

3 1

4 4

课程 描述 教授

1 Physics for Short-定货 cooks Adms

2 Counterculture Sociology Beckely

3 Applied Basketweaing Carruth

4 Introduction to Physiology Dawsons

### VB 数据库编程二

信息来源：黑客代码 编辑：admin 浏览次数：211 加入时间：2010-3-31 11:49:15 评论(0)

### VB 数据库编程二

现在表被规范化了，所以，要改变特定课程的课程描述或“数据”，只要改变一个记录就可以了。

以上是关于数据库的基本知识，这是学习数据库编程所必须的。虽然数据库技术作为一门学科，其深度和广度不是这点篇幅能描述的，但作为入门和简单数据库编程应该是足够了。

好了，下面我们就可以开始练练了。我们经常遇到数据库系统是登记系统，不管你是在单位，或是参加什么组织，登记是免不了的，而且它的结构比较简单，我们就以一个登记系统为例吧。分析一下该系统所涉及到的数据。

### 二、磨刀不误砍柴功

对于登记，要跟踪的信息包括：

- 姓名           ● 性别
- 籍贯           ● 年龄
- 出生年月       ● 单位
- 地址           ● 邮政编码
- 电话           ● 传真

当然，可以简单地创建一个表，使得上述的每个数据项对应一个字段。

现在需要给表指派主键,用以唯一标识每一条记录,在登记表中分别添加登记号作为唯一键,这样就保证数据库中的任两条记录都不同了。

对数据库作出以上分析后，我们就可以开始建立数据库了。

三、建营扎寨

在这里我们学习怎样建立数据库，首先需要确定要建立数据库的类型。在 Visual Basic 中通过数据访问控件或数据访问对象（DAO）可以访问下列数据库：

1. JET 数据库，即 Microsoft Access
2. ISAM 数据库，如：dBase, FoxPro 等
3. ODBC 数据库，凡是遵循 ODBC 标准的客户/服务器数据库。如：Microsoft SQL Server、Oracle

一般来说，如果要开发个人的小型数据库系统，用 Access 数据库比较合适，要开发大、中型的数据库系统用 ODBC 数据库更为适宜。而 dBase 和 FoxPro 数据库由于已经过时，除非特别的情况，否则不要使用。在我们的例子中，当然选用 Access 数据库了。建立 Access 数据库有两种方法：一是在 Microsoft Access 中建立数据库。点击“新建”按钮就可以建立新的表了（如图 1）。这里我们主要介绍第二种方法：使用可视化数据管理器，不需要编程就可创建数据库。可视化数据管理器是一个非常有用的应用程序，它是 VB 企业版和专业版附带的，在目录 \\.\DevStudio\vb\samples\Visdata 下，其界面如下图。

点击菜单“文件”项下“新建”子项“Microsoft ACCESS”子项的“版本 7.0 MDB”项。在弹出窗口中输入新建数据库的名称“登记”，出现下面图 3 所示窗口：

要生成新的表，右键单击数据库窗口弹出菜单，然后选择“新表”命令，在随后出现的“表结构”对话框中建立所要的字段。每次向表中加入新的字段，单击“增加字段”按钮，会出现图 4 的“增加字段”对话框。

“增加字段”对话框中的选项如表 10 所示，根据字段的类型，有些选项是无效的，无法读取。在我们建立的登记数据库中，各个字段的类型如表 11。

要注意的是，由于字段登记号用来唯一标志记录的，因此，它不能由用户输入。所以在定义该字段时需要定义为 Long 数据类型，“自动生成字段”项有效，并选中这一项。这样当用户每输入一条新记录时，系统就会在该字段上自动输入一个与其它记录不同的值。

在 ACCESS 数据库中，关键字是用索引实现的，作为编程人员在对表类型的记录集编程时，只需调用索引名。在查询时，Rushmore 技术自动用索引信息优化查询。完成表定义后，点击“增加索引”按钮，弹出如图 5 所示窗口。

在窗口中右边有三个选项，其意义如表 12。

添加索引对话框选项

完成之后如图 6。

当然，学会数据库的建立也并非一朝一夕的事，读者不妨多练习一下。下面你就可以运行 VB 开始我们的编程了。

#### 四、千里相会

Visual Basic 数据库应用程序有三个部分，如图 7 所示。

用户程序是程序员开发的，也是我们即将用 VB 来编写的部分。数据库引擎是数据库驱动程序，使用它程序员可以用统一的格式访问各种数据库，不管这个数据库是本地的 Visual Basic 数据库，还是所支持的其它任何格式的数据库格式，所使用的数据访问对象和编程技术都是相同的。数据库则是我们上面完成的部分。从这个结构可以看出用户与正在访问的特定数据库无关。那我们在用 VB 编写数据库程序时，就需要使程序能够访问指定的数据库。

如果是简单的数据库应用，可以使用 Data 控件来执行大部分数据访问操作，而根本不用编写代码。与 Data 控件相捆绑的控件自动显示来自当前记录的一个或多个字段的数据。

#### DATA 数据控件

属性

## VB 数据库编程

CONNECT 属性 指定打开的数据库类型，并且包括参数，如用户和口令等。

例如：

打开 Access 数据库（缺省）

```
CONNECT="ACCESS"
```

打开 ODBC 数据库

```
CONNECT="ODBC; DATABASE=? ? ; UID=? ? ; PWD=? ? ; DSN=? ? "
```

DATABASENAME 属性 确定数据控件访问哪一个数据库。

对于多表数据库它为具体的数据库文件名，例如：ACCESS 数据库

```
DATABASENAME="D:\..\DEMO.MDB"
```

对于单表数据库它为具体的数据库文件所在的目录，而具体文件名放在 RECORDSOURCE 属性中，例如：访问 FOXPRO 数据库文件 D:\FOX\DEMO.DBF

```
DATABASENAME="D:\FOX"
```

```
RECORDSOURCE="DEMO" 不带文件扩展名
```

RECORDSOURCE 属性

确定数据控件的记录集，即：所要访问的数据内容。它可以是一个表名、存储查询名或 SQL 语句。

例如：访问 Register 表所有数据：

```
RECORDSOURCE="Register" 访问 RC 表中 1973 年以前出生的数据：
```

```
RECORDSOURCE="SELECT *FROM Register WHERE [BIRTHDAY]<#1/1/1973#"
```

注意：当我们在运行时修改了该属性后，需要调用 REFRESH 方法刷新记录集。

### 方法

REFRESH 方法 当我们在运行时修改了 Record-

Source 属性后，需要调用该方法刷新记录集。

UPDATERECORD 方法 将绑定在数据控件上的控件的数据写入数据库中。即：当我们修改了数据后调用该方法确定修改。

CANCELUPDATE 方法 将数据库中的数据重新读到绑定在数据控件上的控件中。即：当我们修改了数据后调用该方法放弃修改。

## VB 数据库编程三

### 事件

VALIDATE 事件 当我们移动记录集记录指针时发生。例如：我们将记录集记录指针从 A 移动到记录 B 时当产生 VALIDATE 事件时，记录指针仍在记录 A 上。

```
Sub XXXX_Validate (Action As integer, Save As integer)
```

其中：

Action 指出如何产生了该事件，如：移动，增加，查询等。

Save 表示是否保存已修改的数据。当我们修改了绑定在数据控件的数据，又没有 UPDATERECORD，则移动指针时，Save=True。如果在事件中令 Save=False，则放弃修改。

例如：

```
Sub XXXX_Validate(Action As integer,Save As integer)
```

```
If Save then
```

```
I= MsgBox("Dada changed,Save?",vbYesNo)
```

```
If I = vbNo then
```

```
Save = False
End if
End if
End Sub
```

**Reposition 事件** 当我们移动记录集指针时发生。例如：我们将记录集记录指针从 A 移动到记录 B 时，当产生 Reposition 事件时，记录指针已移动到 B 上。

通常我们在该事件中显示当前的指针位置。例如：

```
Sub XXXX_Reposition()
XXXX.Caption=??
XXXX.RecordSet.AbsolutePosition + 1
End Sub
```

了解了 DATA 控件之后我们就可以连接数据库了。现在我们可以编写一个应用程序。因为虽然我们建立了 Register 数据库，但是数据库中却没有数据，我们程序的目的就是向数据库中输入数据。它的运行情况如图 8。

各个文本框正好对应着表 Register 的各个字段，在文本框中输入数据，点击“增加”按钮，就完成了了一条记录的输入。我们看一下，DATA 控件是怎样和数据库连接起来的，各个文本框又是怎样和 DATA 控件捆绑起来的。

在 DATA 控件的 CONNECT 属性中，选中“ACCESS”项，在 DatabaseName 属性中，输入“C:\TEMP\登记.mdb”，在 RecordSource 属性中，选中“Register”，这样就完成了数据库与 DATA 控件的连接，也就是完成了与应用程序的连接。

数据库中各个字段又是怎样和文本框连接起来的呢？在 VB 中，我们可以将普通控件绑定在数据控件上，来完成自动地显示、更新记录集的数据。常用的可绑定的控件有：Label,Text,checkBox,Image 等。通过设置这些控件的 DataSource 和 DataField 属性来完成绑定。

**DataSource 属性** 表示绑定到哪一个数据控件上，程序中我们可能使用多个数据控件。

**DataField 属性** 表示绑定到记录集的哪一个记录上。

现在我们需要把 Text1 与表“登记”中的姓名字段连接起来。完成 DATA 控件的连接之后，在 Text1 控件的 DataSource 属性中，选中“Data1”，在 DataField 属性中，选中“姓名”值。用同样的方法，将各个文本框分别绑定到对应的字段上，就完成了文本框的捆绑。

下面我们编写两个按钮命令，完成其相应的操作了。喂！别着急，还有一个重要的对象没讲呢！

当应用程序启动时，Data 控件被自动地初始化。如果 Connect、DatabaseName、Options、RecordSource、Exclusive、ReadOnly 和 RecordsetType 属性是合法的，Microsoft Jet 数据库引擎就会试图创建一个新的基于这些属性的 Recordset 记录集对象。Recordset 对象可以表示表中的记录或者作为查询结果的记录，使用 Recordset 对象可以在记录一级上对数据库中的数据进行处理。这在数据库编程中是一个十分重要的，也是比较复杂的对象。

Recordset 对象有三种类型：表、动态集、快照，它们之间存在明显的区别。

表类型的 Recordset 对象是指当前数据库中的表在创建表类型的记录集时，数据库引擎打开的表。后续的数据操作都是直接对表进行的。只能对单个的表打开表类型的记录集，而不能对联接或者联合查询打开表类型的记录集。与其它类型的 Recordset 对象相比，表类型的搜索与排序速度最快。

动态集类型的 Recordset

对象可以是本地的表，也可以是返回的行查询结果。它实际上是对一个或者几个表中的记录的一系列引用。可用动态集从多个表中提取和更新数据，其中包括链接的其它数据库中的表。动态集类型具有一种与众不同的特点：不同数据库的可更新联接。利用这种特性，可以对不同类型的数据库中的表进行可更新的联接查询。动态集和它的基本表可以互相更新。如果动态集中的记录发生改变，同样的变化也将在基本表中反映出来。在打开动态集的时候，如果其他的用户修改了基本表，那么动态集中也将反映出被修改过的记录。动态集类型是最灵活的 Recordset 类型，也是功能最强的。不过，它的搜索速度与其它操作的速度不及表类型的 Recordset。

快照类型的 Recordset 对象包含的数据是固定的，它反映了在产生快照的一瞬间数据库的状态。从 Microsoft Jet 数据源得到的快照是不可更新的，从开放数据库互连 (ODBC) 数据源得到的某些快照是可以更新的，这取决于数据库系统本身的能力。与动态集类型和表类型的 Recordset 对象相比，快照的处理开销较少。因此，它执行查询和返回数据的速度更快，特别是在使用 ODBC 数据源时。快照类型保存了表中所有记录的完整复本，因此，如果记录的个数很多，快照的性能将比动态集慢得多。为了确定快照与动态集哪一个更快，可以先以动态集方式打开记录集，然后再以快照方式打开它。

具体使用什么记录集，取决于需要完成的任务：是要更改数据呢，还是简单地查看数据。例如，如果必须对数据进行排序或者使用索引，可以使用表。因为表类型的 Recordset 对象是做了索引的，它定位数据的速度是最快的。如果希望能够对查询选定的一系列记录进行更新，可以使用动态集。如果在特殊的情况下不能使用表类型的记录集，或者只须对记录进行扫描，那么使用快照类型可能会快一些。

一般来说，尽可能地使用表类型的 Recordset 对象，它的性能通常总是最好的。

为选择特定的 Recordset 类型，把 Data 控件的 RecordsetType 属性设成：

### VB 数据库编程四

信息来源：黑客代码 编辑：admin 浏览次数：119 加入时间：2010-3-31 11:46:17 评论(0)

### VB 数据库编程四

#### RecordSet 记录集属性

BOF 属性 当记录集记录指针指向第一条记录时返回 True

EOF 属性 当记录集记录指针指向最后一条记录时返回 True

AbsolutePosition 属性 返回当前记录集记录指针，第一条记录为 0，是只读属性

Bookmark 属性 String 类型，返回或设置当前记录集记录指针的书签，是可读写属性。每一条记录都有自己唯一的书签，它与记录在记录集中的顺序无关。将 Bookmark 属性存放到变量中，后面可以通过将该变量赋值给 Bookmark 属性，并返回到这个记录。

注意：程序中使用 BookMark 属性重定位记录指针，而不能使用 Abslouteposition

NoMatch 属性 当我们使用 Find 方法查询时如果未找到则返回 True。常与 BookMark 属性同时使用。例如：查找[NAME]字段中第一个姓李的人

```
Dim S As String
```

```
With XXXX.RecordSet
```

```
S = .BookMark
```

```
.FindFirst "[NAME] Like '李*'"
```

```

if .NoMatch then
MsgBox "数据未找到 “
.BookMark = S
End if
End With

```

记录集方法

AddNew 方法 向记录集增加一条新记录

Delete 方法 从记录集中将当前记录删除。在删除后常使用 MoveNext 方法移动指针。

例如：

```

With XXXX.RecordSet
.Delete
.MoveNext
if .EOF then .MoveLast
End With

```

MoveXXXX 方法

MoveFirst 将记录集指针移动到第一条记录上

MoveLast 将记录集指针移动到最后一条记录上

MovePrevious 将记录集指针移动到前一条记录上

MoveNext 将记录集指针移动到下一条记录上

FindXXXX 方法

FindFirst 在记录集中查询符合条件的第一条记录

FindLast 在记录集中查询符合条件的最后一条记录

FindPrevious 在记录集中查询符合条件的前一条记录

FindNext 在记录集中查询符合条件的下一条记录

好了，有了这么充分的知识了，编写两个按钮命令简直是小菜一碟，先来试一下，添一个“增加”命令按钮吧。

```

Private Sub Command1_Click()
Data1.Recordset.AddNew
End Sub

```

哇！怎么这么简单，再看一下“删除”命令按钮

```

Private Sub Command2_Click()
Data1.Recordset.Delete
Data1.Recordset.AddNew
End Sub

```

就这样行了吗？运行程序吧，OK！一切正常，迫不及待地输入一条记录，点击“增加”按钮，怎么？出问题了！因为你只有在进行了 AddNew 方法后才可以输入数据，好吧，在窗口的初始化时就增加一条新记录吧。

```
Private Sub Form_Initialize()
Data1.Recordset.AddNew
End Sub
```

输入完了数据，我们打算退出程序，很自然的我们执行关闭窗口操作，就顺利地结束了输入工作。真的很顺利吗？打开数据库，看看数据库中的数据，我们发现刚才输入的最后一条记录没有存入数据库中。这个很好解释，每当我们调用 `AddNew` 方法时，它就将输入的记录存入数据库中，而当我们关闭窗口时，刚输入的记录并没有保存到数据库中，那么在关闭窗口之前对 `DATA` 控件进行一次刷新就可以将数据存入数据库中了。

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
Data1.Refresh
End Sub
```

到了这里，我们似乎可以稍稍轻松了一点，这个窗口的功能差不多完成了。但是我不得不给你提出一个忠告：在数据库系统中，应尽量将错误在应用级上处理。这句话看起来似乎有点抽象，实际上用在这个程序中就简单多了。在表 `Register` 中，我们将出生日期定义为 `Date/Time` 类型，如果在程序运行时，在该字段对应的文本框中输入的不是 `Date/Time` 格式，在向数据库提交数据时会出现什么情况呢？数据库会向用户报告错误信息。然而这样对应用程序并不好，这样的错误应该由用户程序处理，而不是交给数据库去处理，所以在数据提交之前就应该检查该字段的输入是否合法。

```
Private Sub Text3_LostFocus()
If IsDate(Text3.Text) Or Text3.Text = "" Then '检查是否输入合法数据
Exit Sub
End If
MsgBox ("输入错误，请输入你出生的年月日！")
,将选取不合法的数据，以便重新输入，并使控制焦点不动
Text3.SetFocus
Text3.SelStart = 0
Text3.SelLength = Len(Text3.Text)
End Sub
```

上面虽是应用程序处理错误的一个小例子，可是这种在应用级处理错误的思想是十分重要的。

## VB 数据库编程五

### 五、寻寻觅觅

在数据库管理系统中,输入和查询就象两个孪生姐妹不可或缺,下面将介绍如何创建查询窗口。查询窗口的设计分为两部分:查询结果和查询条件。查询结果是指用户所需要的数据,它包括根据查询条件查询出来的记录,但并非表中每个字段里的数据都需要提供给用户。比如在我们这个例子中,登记表中的登记号的值是用户不感兴趣的,所以在查询结果中,我们不希望显示 RegID 字段的值。查询条件是由用户提出的查询要求。比如在我们这个系统中,可以有姓名条件,当用户想知道某人的具体情况,他可以输入此人的姓名,就查询出此人各方面的情况;也可以有年龄条件,当用户输入某个年龄段,就会查询出处于这个年龄段的所有人的信息。到底采用哪些查询条件,这需要开发者根据用户和系统的要求进行设计,其具体实现过程大都大同小异。为了节省篇幅,我们就仅以年龄为条件进行查询。查询窗口运行情况如图 9。

在这个程序中我们使用了一个控件 DBGrid,这个控件用来显示查询结果,选中 VB 的“工程”菜单下的“部件……”项,在控件标签中,选中“Microsoft Data Bound Grid Control 5.0”即可,在工具箱中就会出现 DBGrid 控件的小图标。其使用和其它控件一样。

为了和数据库连接,DATA 控件是不可少的,回忆一下,该怎样设置它的属性,OK!同输入窗口一样,在 CONNECT 属性中,选中“ACCESS”项。在 DatabaseName 属性中,输入“C:\TEMP\登记.mdb”。在 RecordSource 属性中,选中……嘿,嘿,这里稍微有点不同,如果按输入窗口那样的设置,查询结果中就会包含登记号字段了。在此属性中我们应该输入 SQL 语句:

select name as 姓名,sex as 性别,hometown as 籍贯,age as 年龄,birthday as 生日,company as 单位,address as 地址,zip as 邮编,telephone as 电话,fax as 传真 from Register。别着急,尽管这条语句有点长,实际上却比较简单。这条语句的语法是:

SELECT 字段名, 字段名, ……FROM 表名 WHERE 条件;

对照语法,我们可以看出输入的 SQL 语句的含义:从表 Register 中查询姓名,性别,籍贯,……字段的值。只要在字段列表中不选中登记号字段,在查询结果中,就不会显示登记号的值了。如果你够细心的话就会注意到我们所写的 SQL 语句中在字段列表中并不仅仅输入字段名,在其后面还增加了 as……项,如“name as 姓名”,这是为 name 字段取一个别名“姓名”,以便在 DBGrid 控件中显示字段名时,就会显示“姓名”而不是“name”。

完成了 DATA 控件属性的设置就可以将 DBGrid 控件捆绑到 DATA 控件上,其方法同输入窗口。对了,将“DataSource”属性设置为“DATA1”即可。现在不妨运行一下程序,真令人兴奋,DBGrid 显示出表中所有的信息。可是怎样显示符合条件的数据呢?再看一看上面的 SQL 语法,WHERE 段后可以输入查询条件,比如:需要年龄在 20 到 30 岁之间的人员信息,其语句为: Select name,… from register where age>20 and age<30;

我们只要根据用户输入的条件构成新的 SQL 语句,并利用 DATA 控件的 Refresh 方法刷新数据库,就可以完成条件查询了。

整个程序十分简单,当用户在文本框中输入年龄段后,点按“查询”命令,就会显示符合条件的查询结果。程序代码如下:

Option Explicit

Const allinfo = "select name as 姓名,sex as 性别,hometown as 籍贯,age as 年龄,birthday as 生日,company as 单位,address as 地址,zip as 邮编,telephone as 电话,fax as 传真 from 登记" '定义不带条件的 SQL 查询语句

Private Sub Command1\_Click()

Dim t1, m, n As String

'构造 SQL 语句的 WHERE 条件段

If IsNumeric(Text1.Text) Then

```
m = " age >" + Text1.Text  
End If  
If IsNumeric(Text2.Text) Then  
n = " age <" + Text2.Text  
End If
```

```

If m = "" And n = "" Then
t1 = ""
Else
If m = "" Then
t1 = n
Else
If n = "" Then
t1 = m
Else
t1 = m + " and " + n
End If
End If
End If
If t1 <> "" Then
t1 = " where " + t1
End If
Data1.RecordSource = allinfo + t1
'将 RecordSource 属性的值设置为新的 SQL 语句
Data1.Refresh
'刷新数据库，获得符合当前条件的查询结果
End Sub

```

在这个程序中，你还可以加上姓名查询、籍贯查询等，其设计过程都是通过构造 SQL 语句来实现的。另外，还需要注意的一点是既然是查询窗口，当然允许修改，将 DBGrid 控件的 AllowUpdate 属性值设为 False。OK！运行程序吧，啊！万事大吉。看一看应用程序，再看一看你编写的代码，不可思议，这么短的代码居然完成了这么强大的功能。事实就是这样，数据库编程中最重要的是对概念的清楚，与之相比，技巧好象摆在了次要位置。让我们闭上眼睛，想一想我们需要了解那些东西吧：DATA 控件，RECORDSET 对象、控件的捆绑、简单的 SQL 查询语句。熟悉了这些东西，你就可以进行数据库编程了，如果你对开发数据库系统比较感兴趣，可以再加强一下在数据库设计和 SQL 语句方面的知识。但数据库编程真的总是这样简单吗？如果已经熟悉了上面的内容，建议你看一下下面这一节。

## VB 数据库入门 6

### 六、拨云见雾

还记得我们在前面说过 Visual Basic 提供了两种与 Jet 数据库引擎接口的方法吗？Data 控件和数据访问对象。我们已经见识了 DATA 控件，确实给人意想不到的方便，但 Data 控件只给出有限的访问现存数据库的功能。而 DAO 模型则可以全面控制数据库的完整编程接口。这两种方法事实上并不是互斥的，实际上，它们常同时使用。

DAO 模型是设计关系数据库系统结构的对象类集合。它们提供了完成管理这样一个系统所需的全部操作属性和方法，包括创建数据库，定义表、字段和索引，建立表间的关系，定位和查询数据库等工具。DAO 结构的主要对象如图 10 所示。

Visual Basic

中的数据库编程就是创建 DAO（数据访问对象）。这些对象对应于被访问物理数据库的不同部分，如 Database、TableDef、Field 和 Index 对象，用这些对象的属性和方法来实现对数据库的操作，能够在 Visual Basic 窗体中使用绑定和非绑定控件来显示操作结果并接收用户输入。这样就简化了代码，它赋予程序员很大的灵活性。因为可以使用同样的对象、属性和方法来处理各种不同的数据库格式。同时，如果从一种数据库格式变到另一种格式（例如，将本地的 Microsoft Access 数据库转换为网络上的 SQL Server 数据库），只需变动少量的代码就可以适应这种改变。甚至可以创建这样的应用程序，在单一的查询或报表中连接来自多个不同数据库的表。

我们先介绍 DAO 对象，从上图中我们可以看到，DAO 对象中最关键的是 DBEngine 对象，所有的数据库操作都要通过它来完成。

### DBEngine 对象

DBEngine 对象相当于 jet 数据库引擎，不需要创建该对象。

CreateWorkspace 方法:创建一个工作区对象

例如：

```
Dim ws As Workspace
Set ws = DbEngine.CreateWorkspace (SpName,UserID,password",SpType)
```

其中

SpName 工作区名称 String

UserID 用户名 String

Password 口令 String

SpType 工作区类型 = dbUsejet jet 工作区

= dbUseODBCODBC 工作区

### Workspace 对象

Workspace 对象为用户定义一个会话，通过与之关联的用户名和口令建立一个安全级别。当不需要安全级别时可使用缺省的工作区 DBEngine.Workspace。

方法:

OpenDatabase 打开一个数据库

Close 关闭当前工作区

BeginTrans 启动一个事务

CommitTrans 提交当前事务

RollBack 回滚当前事务

例如我们需要打开一个数据库。

我们调用 Workspace 的 OpenDatabase 方法打开一个数据库。

例如:

```
Dim db As DataBase
Set db=OpenDatabase(dbname,exc,readonly,source)
```

其中:

dbname 数据库文件名（对于单表数据库为路径）

exc = TRUE 表示打开数据库后，该数据库不能被其它应用程序访问。

FALSE 表示共享地打开数据库

Readonly=TRUE 表示打开数据库后，只能对数据库进行读操作

=FALSE 表示打开数据库后，可对数据库进行读写操作

source 数据源名，用于指定打开地数据库类型。

### 创建记录集

我们使用 DataBase 对象的 OpenRecordSet 方法打开一个记录集。

例如:

```
Dim rs As RecordSet
```

```
Set rs=db.OpenRecordSet(source, type)
```

其中:

source 指定记录集的内容,它可以是一个表名,查询名或 SQL 语句

type 创建记录集的类型

=dbOpenTable 表型记录集

=dbOpenDynaset 动态型记录集

=dbOpenSnapshot 快照型记录集

### 访问 ODBC 数据库

Visual Basic 通过 DAO 和 Jet 引擎可以识别三类数据库: Visual Basic 数据库, 外部数据库, ODBC 数据库。在开发大中型数据库系统中一般都采用 ODBC 数据库, 如 Microsoft SQL Server 等。DAO (数据访问对象) 的 ODBCdirect 模式允许直接访问 ODBC 数据。另外, 远程数据对象 (RDO) 库和远程数据控件 (RDC) 也 允许直接访问 ODBC 数据。

要访问 ODBC 数据库需要两步, 首先需要在 WINDOWS 95 或 WINDOWS NT 的控制面板中设置 ODBC 数据源名 (DSN)。然后在 OpenDatabase 方法的 Source 参数中使用该数据源名。

例如: 访问 Microsoft SQL Server

```
Source="ODBC;DSN=MyDSN;UID=sa;PWD=xx;DataBase=pubs"
```

在程序中使用 DAO

DAO 的强大功能确实让人兴奋, 只是较之 DATA 控件要复杂得多。在复杂的数据库处理中, 这两种方法常常同时使用。要使用 DAO, 必须选中 VB 的“工程”菜单下的“部件……”项, 在控件标签中选中"Microsoft DAO 3.5 Object Library"之后就可以在程序中定义 DAO 变量了。在这篇文章中我们所举的数据库例子比较简单, 有兴趣的朋友不妨采用数据访问对象 (DAO) 来实现这个程序。

### 七、高屋建瓴

到了这里, 似乎 VB 数据库编程的内容也差不多完整了, 但是你完全可以再往前走一步。在目前数据库开发过程中, 客户/服务器体系结构占这个领域的主体地位, 利用 VB 进行客户/服务器应用程序的开发是完全可行的。同时, 在程序开发过程中, 还可以利用面向对象技术, 提高你的程序的可重用性和可维护性。总之, 数据库应用程序的开发始终是技术和理论上的热点。使用最高级的方法, 最先进的技术, 你会变得越来越 COOL

### VB 控制光驱弹出关闭

#### Option Explicit

```
Private Declare Function CDdoor Lib "winmm.dll" Alias "mciSendStringA" _  
    (ByVal lpstrCommand As String, ByVal lpstrReturnString As String, _  
    ByVal uReturnLength As Long, ByVal hwndCallback As Long) As Long
```

```
Dim state1 As Boolean
```

```
Private Sub Command1_Click()
```

```
'打开光驱
```

```
state1 = True
```

```
Call CDdoor("set CDAudio door open", 0, 0, 0)
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
'关闭光驱
```

```
state1 = False
```

```
Call CDdoor("set CDAudio door closed", 0, 0, 0)
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
End
```

```
End Sub
```

vb 实现小写转大写金额

信息来源：黑客代码 编辑：wanghujun 浏览次数：144 加入时间：2010-3-30 22:09:10 评论(0)

vb 实现小写转大写金额

```
Private Function setdata(num As Integer) As String '数字转换
```

```
Select Case num
```

```
Case 0
```

```
setdata = "零"
```

```
Case 1
```

```
setdata = "壹"
```

```
Case 2
```

```
setdata = "贰"
```

```
Case 3
```

```
setdata = "叁"
```

```
Case 4
```

```
setdata = "肆"
```

```
Case 5
```

```
setdata = "伍"
```

```
Case 6
```

```
setdata = "陆"
```

```
Case 7
```

```

        setdata = "柒"
    Case 8
        setdata = "捌"
    Case 9
        setdata = "玖"
    End Select
End Function
Private Function chang(aaa As Integer) As String '位数转换
    Select Case aaa
    Case 1
        chang = ""
    Case 2
        chang = "十"
    Case 3
        chang = "百"
    Case 4
        chang = "千"
    Case 5
        chang = "万"
    Case 6
        chang = "十"
    Case 7
        chang = "百"
    Case 8
        chang = "千"
    Case 9
        chang = "亿"
    Case 10
        chang = "十"
    End Select
End Function

Private Sub Form_Activate() '设定文本长度
    Text2.MaxLength = 10
    Text2.SetFocus
End Sub

Private Sub Text2_Change() '小写转大写
    Dim i As Integer
    Dim j As Integer
    Dim myint As Integer
    Dim myint1 As Integer
    Dim mydoub As Double
    Dim mystr As String

```

```

Dim mystr1 As String
Dim mystr2 As String
Dim mystr3 As String
Dim mystr4 As String
Dim money As Long
Dim money1 As Integer
Dim money2 As Long
mystr = Text2.Text
myint = InStr(mystr, ".")
If myint = 0 Then
    mystr = Text2.Text
Else
    mystr3 = Right(Text2.Text, Len(Text2.Text) - myint)
    If mystr3 <> "" Then      '转换小数位
        mystr4 = Left(mystr3, 1)
        mystr3 = Right(mystr3, Len(mystr3) - 1)
        If mystr4 <> "0" Then
            mystr2 = mystr2 + setdata(Val(mystr4)) + "角"
        End If
        If mystr3 <> "" Then
            mystr4 = Left(mystr3, 1)
            mystr2 = mystr2 + setdata(Val(mystr4)) + "分"
        End If
    End If
    mystr = Left(Text2.Text, myint - 1)
End If
j = Len(mystr)
For i = 1 To Len(mystr)      '转换整数位
    money2 = Left(mystr, i)
    money1 = Right(money2, 1)
    If money1 = 0 Then
        If j = 5 Then
            If Right(mystr1, 1) <> "万" Then mystr1 = mystr1 & "万"
        Else
            If Right(mystr1, 1) <> "零" And Right(money, j) > 0 Then mystr1 = mystr1 & "零"
        End If
    Else
        mystr1 = mystr1 & setdata(money1) + chang(j)
    End If
    j = j - 1
Next i
Text1.Text = mystr1 & "元" & mystr2 '显示大写
End Sub

```

```
Private Sub Command1_Click()
    End
End Sub
```

## VB 写数据库登录教程

```
Dim TIM As Integer
Dim mydb As Database
Dim myrs As Recordset
Dim sql As String
Private Sub Form_Activate()
    Set mydb = Workspaces(0).OpenDatabase(App.Path & "\kfgl.mdb") '打开数据库
    sql = "select 操作员 from qxsz"
    Set myrs = mydb.OpenRecordset(sql)
    If myrs.EOF = False Then myrs.MoveLast
    If myrs.BOF = False Then myrs.MoveFirst
    For i = 0 To myrs.RecordCount - 1
        master.AddItem (myrs.Fields(0))
        myrs.MoveNext
    Next i
    If master.ListCount > 0 Then master.ListIndex = 0
    myrs.Close
    mydb.Close
    master.SetFocus
End Sub
Private Sub Cmdok_Click() '确认操作员和密码
    Dim MESSAGE As String
    If master.Text <> "" Then
        Set mydb = Workspaces(0).OpenDatabase(App.Path & "\kfgl.mdb")
        Set myrs = mydb.OpenRecordset("qxsz", dbOpenDynaset)
        myrs.FindFirst "操作员 = " + Chr(34) + master.Text + Chr(34) + "" '查找操作员
        If myrs.NoMatch Then '没查到记录
            MsgBox ("无此操作员! ")
        Else
            If password.Text = myrs.Fields("密码") Then '确认密码
                MsgBox "系统登陆成功! " '系统登录成功
            Else
                If TIM = 3 Then
                    MESSAGE = MsgBox("密码输入错误,请向系统管理员查询!", 0, "")
                    If MESSAGE = vbOK Then End
                End If
                MsgBox ("密码错误,请重新输入密码!")
            End If
        End If
    End If
End Sub
```

## VB 数据库编程

```
TIM = TIM + 1                                '密码输入错误次数
password.SetFocus
End If
End If
End If
End Sub
Private Sub master_KeyDown(KeyCode As Integer, Shift As Integer) '回车换行
    If KeyCode = vbKeyReturn Then
        password.SetFocus
    End If
End Sub
Private Sub password_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        Cmdok.Visible = True
        Cmdok.SetFocus
    End If
    If KeyCode = vbKeyDown Then
        Cmdok.Visible = True
        Cmdok.SetFocus
    End If
    If KeyCode = vbKeyUp Then
        master.SetFocus
    End If
End Sub
Private Sub Cmdexit_Click()
    End
End Sub
```

### VB 读取加密数据库技术

```
Private Sub Form_Load()
    Data1.DatabaseName = App.Path & "\sj.mdb" '设置连接数据库
    Data1.Connect = ";pwd=" & 111           '设置连接密码
    Data1.RecordSource = "sy"              '设置数据表
    Data1.Refresh
End Sub
Private Sub Command1_Click() '移动记录到第一行
    If Not Data1.Recordset.EOF Then Data1.Recordset.MoveFirst
End Sub
Private Sub Command2_Click() '移动记录到前一行
    If Not Data1.Recordset.EOF Then Data1.Recordset.MovePrevious
End Sub
Private Sub Command3_Click() '移动记录到后一行
```

## VB 数据库编程

```
If Not Data1.Recordset.EOF Then Data1.Recordset.MoveNext
End Sub
Private Sub Command4_Click() '移动记录到最后一行
    If Not Data1.Recordset.EOF Then Data1.Recordset.MoveLast
End Sub
Private Sub Command11_Click()
    End
End Sub
VB 破解 access 数据库密码
```

信息来源：黑客代码 编辑：wanghujun 浏览次数：83 加入时间：2010-3-30 22:01:02 评论(0)

### VB 破解 access 数据库密码

```
Dim j As Integer
Private Sub Form_Load()
    Dir1.Path = App.Path
    j = 0
End Sub
Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
End Sub
Private Sub Dir1_Change()
    = Dir1.Path
End Sub
Private Sub ()
    Dim chart(30) As Byte
    Dim data String
    text1 = ""
    Dim strbytes As Long
    If Len() > 3 Then
        datafile =  + "\" +
    Else
        datafile =  &
    End If
    Open Trim(datafile) For Binary Access Read As #1
        Get #1, 67, chart
    Close #1
    text1 = text1 & Chr$(chart(0) Xor &HA7)
    text1 = text1 & Chr$(chart(2) Xor &HEC)
    text1 = text1 & Chr$(chart(4) Xor &H7C)
```

```

text1 = text1 & Chr$(chart(6) Xor &H9C)
text1 = text1 & Chr$(chart(8) Xor &HE7)
text1 = text1 & Chr$(chart(10) Xor &H28)
text1 = text1 & Chr$(chart(12) Xor &H32)
text1 = text1 & Chr$(chart(14) Xor &H8A)
text1 = text1 & Chr$(chart(16) Xor &H75)
text1 = text1 & Chr$(chart(18) Xor &H7B)
text1 = text1 & Chr$(chart(20) Xor &HD4)
text1 = text1 & Chr$(chart(22) Xor &HDF)
text1 = text1 & Chr$(chart(24) Xor &H56)
text1 = text1 & Chr$(chart(26) Xor &H13)
text1 = text1 & Chr$(chart(28) Xor &HEE)
text1 = text1 & Chr$(chart(30) Xor &HB1)
End Sub
Private Sub Command1_Click()
    End
End Sub

```

#### VB 把设置内同保存到数据库

```

Private Sub Form_Load()
    Data1.DatabaseName = App.Path + "\data.mdb"
    Data1.RecordSource = "data"
End Sub
Private Sub Form_Activate()
    Data1.Recordset.FindFirst "name = "" & "Image1" & """"
    Image1.Top = Val(Text2(1).Text)
    Image1.Left = Val(Text2(2).Text)
    Image1.Height = Val(Text2(3).Text)
    Image1.Width = Val(Text2(4).Text)
    Data1.Recordset.FindFirst "name = "" & "Label1" & """"
    Label1.Top = Val(Text2(1).Text)
    Label1.Left = Val(Text2(2).Text)
    Label1.Height = Val(Text2(3).Text)
    Label1.Width = Val(Text2(4).Text)
End Sub
Private Sub Image1_Click()
    Data1.Recordset.FindFirst "name = "" & "Image1" & """"
End Sub
Private Sub Label1_Click()
    Data1.Recordset.FindFirst "name = "" & "Label1" & """"
End Sub

```

```
Private Sub Command2_Click()  
    End  
End Sub
```

VB 将图片存入数据库(方法一)

```
Private Sub Form_Load()  
    Data1.DatabaseName = App.Path & "\data.mdb"  
    Data1.RecordSource = "data"  
End Sub  
Private Sub Form_Activate()  
    Image2.Picture = LoadPicture(Text1(2).Text)  
End Sub  
Private Sub Command1_Click()  
    Data1.Recordset.AddNew  
    Data1.Enabled = False  
    command3.Enabled = True  
    Command2.Enabled = True  
    Command1.Enabled = False  
End Sub  
Private Sub Command2_Click()  
    Data1.Recordset.Update  
    Data1.Enabled = True  
    command3.Enabled = False  
    Command2.Enabled = False  
    Command1.Enabled = True  
End Sub  
Private Sub Command3_Click()  
    CmD1.DialogTitle = "选择图片"  
    CmD1.Filter = "bmp|*.bmp;*.ico"  
    CmD1.ShowOpen  
    If Len(CmD1.) Then  
        Text1(2).Text = CmD1.  
        Image1.Picture = LoadPicture(Text1(2).Text)  
        Image2.Picture = LoadPicture(Text1(2).Text)  
    End If  
End Sub  
Private Sub Data1_Reposition()  
    Image2.Picture = LoadPicture(Text1(2).Text)  
End Sub  
Private Sub Command4_Click()  
    End
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要  
下载或阅读全文，请访问：

<https://d.book118.com/638130027045006122>