利用数据访问对象(DAO)操作各种数据库

强建国,马 晓,杨东亚
(兰州理工大学 机电工程学院,甘肃 兰州 730050)

摘要:根据对VB数据库体系结构的分析,提出利用数据访问对象(DAO)操作各种非ACCESS数据库的方法,并给出一个具体实例的关键语句,以达到在Windows环境下用简单的编程而操作所有格式的数据库的目的。

关键词:数据访问对象;VB;非ACCESS数据库
中图分类号:TP311.13

1 引言

随着计算机和信息的发展,数据库应用越来越重要。目前较流行的数据库格式有Access、Foxpro、DBase、Paradox等,但这几种数据库格式之间一般不能完全兼容,给各种格式的数据库使用带来不便。作为当前最流行的开发平台,VB数据库功能强大,Access数据库格式是一种标准的内置格式。VB主要提供了有以下三种数据库方法:

(1)数据控件(DataControl)法;

(2)使用数据访问对象(Data Access Object,简称DAO)法;

(3)直接调用ODBC 2.0 API接口函数法。

其中DAO相对其它两种方法具有方便灵活、功能强大等突出优点。通过调用DAO的方法,可实现windows环境下直接对非Access格式数据库的建新库、拷贝数据库结构、动态调入等操作,达到在Windows环境下操作所有格式数据库的目的。

2 DAO的体系结构

VB数据库的核心结构以MicroSoft JET数据库引擎,它为VB与数据库的接口提供了基本的方法和手段。JET引擎被Microsoft产品所共享,VB中Access数据库格式是一种标准的内置格式,所有的非Access数据库都被称为外来数据库。JET引擎可以插入多种索引顺序存取方法(即ISAM)数据驱动程序,因此只要获得任何一种数据库的ISAM驱动接口程序,就可以利用DAO操作任何一中格式的数据库。数据访问对象的体系结构如图1所示(Datebases集合后的对象未全表示)。

位于DAO对象最高层的是DBEngine对象,它代表Jet数据引擎,用于为数据库引擎设置系统范围的参数,另外还可以设置默认的工作空间。DAO对象中的每个对象都包含了零个或多个集合,每个对象有其属性和方法,对象的属性描述了数据库部件的特征。DAO可以控制多种记录集类型,如Dynaset、Snapshot及Table记录集合对象,可以存储过程和查询动作,可以存取数据库集合对象,例如TableDefs、Fields、Indexes及QueryDefs,具有真正的事物处理能力。DAO对数据库处理的大多数情况都非常适用,记录集对象(RecordSet)同所使用的数据库格式及类型是相互独立的,即对如FoxPro等数据库仍然可以使用DAO对象变量,这为非Access数据库的访问提供了最重要的前提和方法。

利用VB中从一种数据库类型转化为另一种数据库类型几乎不需要或需要很少的代码修改,虽然dBASE、Paradox本身的DDL(Data Definition Language,即数据定义语言)和DML(Data Manipulation Language,即数据操纵语言)是非结构化查询语言,但它们仍然可以使用VB的SQL语句和JET引擎来操纵。

3 对非Access数据库参数设置及配置文件参数读取

在VB程序中用DAO对数据库进行操作并使程序在Windows环境下直接运行,则必须提供数据库配置(.INI)文件,并对不同类型的数据库进行设置,否则程序会显示一条错误信息"Not Found Installable ISAM",数据库操作不能进行。INI文件的文件名和应用程序的名称一般相同,程序会在Windows子目录中搜索和应用程序同名的INI文件。使用VBSetDataAccessOptions语句来设置INI文件,SetDataAccessOptions

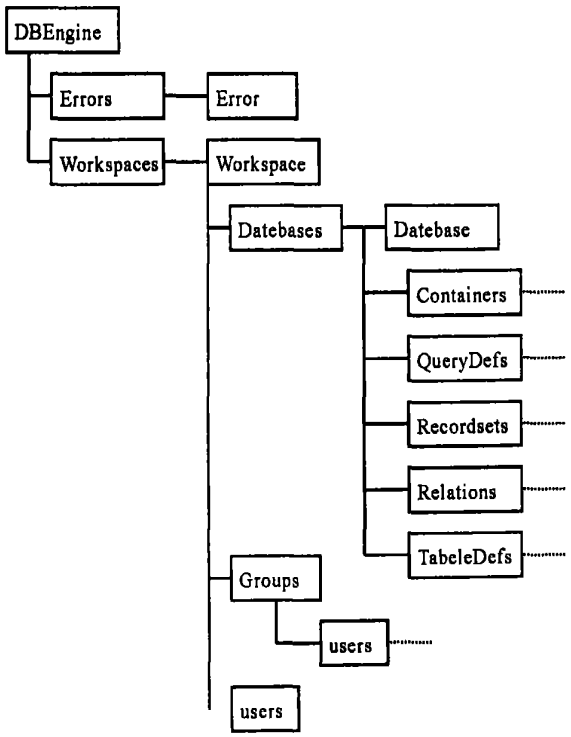

图 1　数据访问对象的体系结构简图

语句的用法如下:

SetDataAccessOptions 1, IniFileName

其中 IniFileName 参数指明的是 INI 文件的带路径的文件名。

Windows API 接口函数在动态链接库中提供了 OSWritePrivateProfileString 函数,此函数能按 Windows 下配置文件(.INI)的书写格式写入信息。

显然,程序要操作较流行的数据库,INI 文件中应有"[Options]"、"[ISAM]"、"[Installed ISAMs]"、"[Fox-Pro ISAM]"、"[dBASE ISAM]"、"[Paradox ISAM]"等设置段及一个属于应用程序的设置段如"[MyDB]"。可在其中设置 DataType、Server、DataBase、OpenOnStartup、DisplaySQL、QueryTimeOut 等较为重要的数据库参数,并以此限定应用程序一般的运行环境。

设应用程序的数据库配置文件为 MyDB.INI,则在操作数据库时可用自定义子函数实现获得 INI 文件设置段内的参数,实现对各种数据库格式的读取:

```
Funtion GetINIStr $ (Byval Fname $, Byval szItem $, Byval szDeFault $ )
    Dim Tmp As String, x As Integer
    Tmp = String( 2048,32 )
    x = OSGetPrivateProfileString(Fname $, szItem $, szDefault $, Tmp, Len(Tmp), "MyDB.INI")
    GetINIStr = Mid $ ( Tmp, 1, x )
End Function
```

4　用 DAO 对各种非 Access 数据库实现操作

(1) 非 Access 数据库的新建及库结构的修改

DAO 的变量可以分为两类,一类用于数据库结构的维护和管理,另一类用于数据的访问。其中表示数据库结构时可以使用 DataBase、TableDef 、Field、Index 对象和以及三个集合 TableDefs 、Fields、Indexes 三个集合。每个集合都由若干对象组成,数据对象的集合可看作一个数组,并按数组的方法来调用。一旦数据库对象建立,就可以用它对数据库的结构进行修改和数据处理。

对于非 Access 数据库,需先使用 VB 的 MkDir 语句先生成一个目录,亦即新建一个数据库,而把每个非 Access 数据库文件作为新数据库的一个数据表(Table),但实际上它们是互相独立的。新建一个数据库对象,非 Access 数据库的新建不用 CreateDatabase 函数,而用 OpenDatabase 函数。外来数据库的不同格式仅在 OpenDatabase 函数的最后一个参数 Connect 中有所体现,不同格式的外来数据库的 Connect 参数值也不同,除此以外对数据库的操作方法和步骤基本相同。

通过编程还可实现非 Access 数据库的库结构的拷贝来建立与原数据库结构相同的新库。拷贝库结构的方法是把一个已存在的数据库拷贝到一个新文件中,然后再删除新文件内的所有记录,保留其库结构,得到一个新建的结构完整的空库。

下面是新建一个 dbase 7 格式数据库程序实例的关键语句:

```
Sub CNew()
    ……
    MkDir Path $ '新建一个子目录
    Set Dbase1 = OpenDatabase(Path $, True, False, "dbase 7;")
    Set Td1 = Dbase1.TableDefs
    Tab1.Name = "MyDB" '新建一个数据表,数据表名为 MyDB
    ……
    Tab1.Fields.Append F1 '向数据表中添加字段
    ……
    Index1. Name = "Name": Index1. Fields = "Name": Index1.Primary = True '新建索引
    Tab1.Indexes.Append Index1 '向 Indexes 集合中添加新的索引
    Td1.Append Tab1 '向 TableDefs 集合中添加新表
```

```
Dbase1.Close '关闭数据库对象
End Sub
```

(2)非 Access 数据库的动态调入

在实际应用中,往往需要对未知库结构的数据库进行调入、显示及打印其记录。因为 VB 中网格控件非常适用于浏览数据库中的数据,因此只需利用 DAO 把数据放入网格即可。

为读取并显示记录(Colume)内容和字段(Row)内容(包括字段的名称、类型、值等),需要生成一个可以对应于一个或多个数据表中的全部或部分记录的 Dynaset 对象,Dynaset 对象还可以是一个动态查询的结果,能进行记录的增加、删除和修改等操作。下面是用网格显示 dbase 7 数据库实例程序的关键语句:

```
Sub DBaseLoad( )
……
Set Dbase1 = OpenDatabase( Path $, True, False, "dbase 7;" )
Set Ds1 = Dbase1.CreateDynaset( Fn $ )
……
'在网格中填入相应的数据
Dbase1.MoveFirst
I = 1
Do While Not Dbase1.EOF
……
For J = 1 To Fld.Count
Grid1.Col = J: MydbNum = Dbase1.Fields(J - 1).Value
'对记录的数据类型进行判断后做相应的处理
If IsNumeric( MydbNum ) Or IsDate( MydbNum ) Then
……
End If
Next J
Ds1.MoveNext
I = I + 1
Loop
Ds1.Close:Db1.Close
Exit Sub
```

参考文献:

[1] (美)Que CorPoration《开放数据库互连—ODBC 2.0 使用大全》清华大学出版社 1995;

[2] 林立军等《Visual Basic 6.0 数据库开发指南》西安:西安电子科技大学出版社 2000。

[3] 刘柄文等《Visual Basic win32 API》北京:清华大学出版社 2001,1;

(上接第 59 页)大炮,方案如图 4 所示。

3.3 算法效率分析

3.3.1 算法的时间复杂性分析

假设搜索在 M×N 的地图中进行,单次搜索的主要依赖因素是 M 和 N 的值,在最坏的情况下,几乎每一个方格都要被尝试一遍,所以,单次搜索算法的时间复杂度为 $O(M^2N^2)$。在全局搜索的情况下,从对角线方格出发的搜索可以得到最优解,这个特点可以用图论的相关知识进行证明。所以全局搜索的情况下算法的时间复杂度为,虽然这个复杂度较高,但是一般在解决小规模问题的情况下,完全可以接受,比如在搜索 25×25 的地图时,算法耗时大约 200ms 左右。

3.3.2 算法的空间复杂度分析

分支限界法在实现的时候可以用队列和迭代实现,也可以用堆栈和递归实现。因为递归的实现比较清晰和易懂,所以我们采用第二种方式实现,这种要占用递归堆栈,因而可以解决的问题的规模收到系统堆栈大小的制约,经过实际测试,这种实现方式完全可以解决 75×75 的炮兵布阵问题。

4 结论

经过测试,用分治法和分支限界法相结合,在一般的 PC 机上,完全能够在可以容忍的时间范围内解决一般规模炮兵布阵问题,因此,本方法具有实践意义。

参考文献:

[1] 顾启泰. 系统设计与仿真. 北京:清华大学出版社.

[2] Richard A.Brualdi .《组合数学》.北京:机械工业出版社.