

Jet 数据库中多参数存储过程的动态管理

Dynamic Management of Multiparameter Stored Procedure in Jet DB

亓莱滨 陈檀 李国 冯希叶 尹晓燕

(山东教育学院 计算机科学与技术系 山东 济南 250013)

摘要:利用 Jet SQL、ADO 和 ADOX 可以对 MS Jet 数据库中的存储过程进行动态管理。讨论和比较了管理存储过程的各种方法。分析并修正了微软公布的“CREATE PROCEDURE”语句的语法错误。通过应用实例着重介绍了 Jet 数据库中含有默认值的多参数存储过程的创建和执行。

关键词:Jet 数据库 多参数存储过程 Jet SQL ADO ADOX

1 引言

在大型数据库管理系统中,支持存储过程是必备功能,其优点毋庸赘述。MS Jet 引擎自 4.0 版开始增加了存储过程功能,使小型数据库也可以通过多种方法对存储过程的创建、执行、删除和修改等进行动态管理。现有文献中,探讨在 Jet 数据库(主要是 Access)中动态管理多参数存储过程者尚不多见,本文对此作概要讨论。

2 存储过程的创建

MS Access 开发环境内部未提供创建存储过程的功能,因此需要借助于 VBA 或其他编程语言(如 VB、VC、Delphi 等),利用 MS Jet SQL 或 ADOX 实现。

2.1 用 MS Jet SQL 创建存储过程

2.1.1 基本语法及其正误

阐释 Jet SQL 语言的帮助文件 JETSQL40.CHM 随同 Office 2000 及其后续版本一起发布,其中介绍了“CREATE PROCEDURE”语句的语法,在微软官方网站中亦可检索到相关信息^[1,2]。令人遗憾的是,上述资源中对该语句语法的说明存在明显的错误,并且从发布至今一直未作修改,致使许多程序设计者认为无法用该语句创建带有参数的存储过程。

微软公布的“CREATE PROCEDURE”语句的语法格式如下:

```
CREATE PROCEDURE procedure [ param1 datatype
[ , param2 datatype[ , ... ] ] AS sqlstatement
```

其中,procedure 为存储过程名称,param1、param2 为参数,datatype 为参数的数据类型,sqlstatement 为 SQL 语句(如 SELECT、UPDATE、DELETE、INSERT、CREATE TABLE、DROP TABLE 等)。

按照上述语法创建带有参数的存储过程时将会引发实时错误,导致程序崩溃。作者经过对 Jet SQL 语法结构的分析和探索,纠正了其中的错误,修订后的语法格式如下:

```
CREATE PROCEDURE procedure AS [ PARAMETERS
param1 datatype [ = default ][ , param2 datatype [ =
default ] ] [ , ... ] ; ] sqlstatement
```

比较正误两种格式可以看出,原语法存在三处错误:① AS 保留字的位置有误;② 遗漏了用于参数声明的保留字 PARAMETERS;③ 遗漏了参数声明结尾处的分号。此外,修订后的语法补充了原语法中未公布的参数默认值(default)功能。

2.1.2 创建存储过程

调用 ADO 连接对象或命令对象的 Execute 方法均可执行 Jet SQL 语句创建存储过程。以下是在 VB 中利用 ADO 连接对象创建带有参数 pName 的存储过程的简单示例。

```
Dim cnn As New ADODB.Connection, sProc As
String
cnn.ConnectionString = " Provider = Microsoft.
Jet.OLEDB.4.0;" _
& " Data Source = " & App.Path & " \Student.
```

mdb" 设置连接字符串

```
cnn.CursorLocation = adUseClient '若需用 Data-Grid 显示数据则应设置此属性
```

```
cnn.Open '打开连接
```

```
'用于创建存储过程的 Jet SQL 语句:
```

```
sProc = " CREATE PROCEDURE ProcTest AS PA-RAMETERS pName Char = %';" _
```

```
& " Select * From 学籍 Where 姓名 Like pName"
```

```
cnn.Execute sProc '创建存储过程
```

上述存储过程参数 pName 的默认值为 "%", 当调用该存储过程时, 若省略参数 (取默认值), 则相当于对姓名字段不作筛选, 返回学籍表中的全部记录。

2.2 用 ADOX 创建存储过程

ADOX 最初随 ADO 2.1 一起发布, 目前最高版本为 2.8。有关 ADOX 的功能及其应用、ADOX 类库的引用等在现有文献中已有较多的介绍^[3-5], 在此仅就如何用 ADOX 创建存储过程作简要说明。示例如下。

```
Dim cat As New ADOX.Catalog, cmm As New ADOX.Command
```

```
Set cat.ActiveConnection = cnn 'cnn 为已打开的连接对象
```

```
Set cmm.ActiveConnection = cnn
```

```
'创建参数化命令 (遵循 Jet SQL 语法)
```

```
cmm.CommandText = " PARAMETERS pName Char = %';" _
```

```
& " Select * From 学籍 Where 姓名 Like pName"
```

```
'用 Catalog 对象的 Procedures 集合创建存储过程  
cat.Procedures.Append " ProcTest", cmm
```

从以上程序段可以看出, 用 ADOX 创建存储过程仍然是依靠 Jet SQL, 但是避开了 CREATE PROCEDURE 主句和 AS 保留字, 不会因 AS 保留字位置有误而导致程序出错。

3 存储过程的执行

存储过程一旦创建, 只要不删除它, 就会永久保存在数据库中, 随时可以调用。

3.1 用连接对象执行存储过程

执行存储过程最简单的做法是直接通过 ADO 连

接对象调用, 可以省略具有默认值的参数。例如:

```
Dim rs As New ADOX.Recordset
```

```
cnn.ProcTest Trim $ (txtName.Text), rs '调用
```

```
Set DataGrid1.DataSource = rs '显示结果
```

3.2 用命令对象执行存储过程

使用 ADO 命令对象执行存储过程的优点是可以事先对参数集合 (Parameters) 进行设置, 从而实现更加灵活的调用。当执行多参数存储过程时, 这种做法的优势尤为突出。根据是否使用 ADOX, 可以将存储过程的执行分为两种情况。

3.2.1 借助 ADOX 执行存储过程

利用 ADOX 可以通过 OLE DB 提供者检索 Jet 数据库中存储过程的全部信息, 灵活地调用含有默认值的多参数存储过程。例如, 在数据库中已创建了名称为 "MyProc" 的存储过程, 其中含有 pID、pName、pSex 三个参数, 默认值均为 "%", 分别作为按学号、姓名和性别查询的筛选条件。以下示例仅对姓名参数 pName 赋值, 忽略学号和性别参数。

```
Dim cat As New ADOX.Catalog
```

```
Dim rs As New ADOX.Recordset, cmm As New ADOX.Command
```

```
Set cat.ActiveConnection = cnn
```

```
'利用存储过程的 Command 属性获取命令对象
```

```
Set cmm = cat.Procedures (" MyProc"). Command
```

'仅对姓名参数 pName 赋值, 忽略学号和性别参数。亦可用索引号指定参数

```
cmm.Parameters (" pName"). Value = Trim $ (txtName.Text)
```

```
Set rs = cmm.Execute '执行存储过程并返回结果集
```

使用 ADOX 执行存储过程时, 需要从提供者端检索并获取存储过程的全部信息, 然后向数据库发送执行命令。这种做法有如下缺点: ① 向数据库发送的命令中包含了存储过程的全部命令文本, 而不仅仅是存储过程的名称和参数, 如果存储过程的内容较多, 显然会影响效率和性能; ② 增加了对提供者端的调用次数。克服缺点①的方法是获取命令对象后改变其命令类型和文本。例如:

```
cmm.CommandType = adCmdStoredProc '设置命
```

令类型为存储过程

```
cmm.CommandText = " MyProc" `命令文本为存储过程名称
```

解决第二个矛盾的方法是不使用 ADOX。

3.2.2 不使用 ADOX 执行存储过程

为了在使用存储过程时提高性能,应当尽量减少对提供者的调用。在数据消费者端创建命令对象的参数集合,可以避开对提供者端参数信息的检索,即不必调用参数信息的提供者。创建参数集合的主要环节包括调用 ADO 命令对象的 CreateParameter 方法创建 Parameter 对象,然后用 Parameters 集合的 Append 方法将其添加到参数集合中。具体做法请参考本文“5 应用实例”。

3.3 使用 Jet SQL 语句执行存储过程

Jet SQL 中的 EXECUTE 语句可用于执行存储过程,语法格式为:

```
EXECUTE 存储过程名称[参数列表]
```

当参数具有默认值时,若省略了多个参数中的某一参数,则其后续参数必须全部省略。

在实际应用中具体采用上述何种方法应视情而定。若存储过程无参数或参数较少,推荐直接通过 ADO 连接对象调用;若参数较多,可使用 ADO 命令对象执行。Jet SQL 语句的限制较多,通常少用。

4 存储过程的其他管理

4.1 提取数据库中存储过程的信息

提取存储过程信息需要借助于 ADOX 和 ADO 命令对象。遍历 Catalog 对象的 Procedures 集合可以获取数据库中所有存储过程的名称。利用 Procedure 对象的 Command 属性可以获取存储过程的文本以及参数集合中各参数的名称、类型、值和大小。限于篇幅,具体做法请参考文献^[3-5]。

4.2 存储过程的删除和修改

删除存储过程可以通过执行 Jet SQL 语句实现。例如:

```
cnn.Execute " DROP PROCEDURE MyProc"
```

亦可借助于 ADOX 中的 Catalog 对象和 Procedures 集合。例如:

```
cat.Procedures.Delete " MyProc"
```

修改存储过程需要借助于 ADOX 和 ADO 命令对

象。例如:

```
Set cmm = cat.Procedures(" MyProc").Command `获取命令对象
```

```
cmm.CommandText = "... " `在双引号内加入新的存储过程命令
```

```
Set cat.Procedures(" MyProc").Command = cmm `更新存储过程
```

5 应用实例

存储过程最重要的应用是它的创建和执行。下面通过简单的实例进一步说明创建和调用含有默认值的多参数存储过程(不使用 ADOX)。程序运行界面如图 1 所示。



图 1 执行含有默认值的多参数存储过程

事先用在数据库中创建存储过程 ProcQuery, 含有 5 个参数, 分别与图 1 中的 5 个查询项目(工号、姓名、性别、职称和部门)对应, 各参数均具有默认值。

```
Dim sProc As String
```

```
sProc = " CREATE PROCEDURE ProcQuery AS PARAMETERS " _
```

```
&" pID Char = %`, pName Char = %`, pSex Char = %`,` _
```

```
&" pProf Char = %`, pDept Char = %`;" _
```

```
&" Select * From 基本信息 Where 工号 Like pID And 姓名 Like pName" _
```

```
&" And 性别 Like pSex And 职称 Like pProf And 部门 Like pDept" _
```

```
&" Order By 工号"
```

cnn. Execute sProc

在图 1 所示窗体中建立用于设置存储过程参数的自定义 Sub 过程 SetParam。该过程的 4 个参数分别为命令对象、存储过程参数索引、参数值以及是否模糊查询的标志。

```
Private Sub SetParam ( cmmX As ADODB. Command, pIndex As Integer, _
    sParam As String, bFuzzy As Boolean)
    If bFuzzy Then 模糊
        cmmX. Parameters ( pIndex ). Value = "% " &
Trim $ ( sParam ) & "% "
    Else 精确
        cmmX. Parameters ( pIndex ). Value = Trim
$ ( sParam )
    End If
End Sub
```

在“查询”按钮的单击事件中创建参数集合,根据所选查询项目和方式实现灵活的多参数模糊或精确查询。

```
Dim cmm As New ADODB. Command, rs As New
ADODB. Recordset
```

```
Dim prm As New ADODB. Parameter, i As Integer,
blnF As Boolean
```

```
Set cmm. ActiveConnection = cnn
```

```
cmm. CommandType = adCmdStoredProc 命令类
型为存储过程
```

```
cmm. CommandText = " ProcQuery" 命令文本为
存储过程名称
```

创建参数对象及参数集合,指定各参数的类型(变长字符串)和大小

```
For i = 0 To 4
```

```
Set prm = cmm. CreateParameter ( , adVar-
Char, , 20)
```

```
cmm. Parameters. Append prm
```

```
Next
```

```
blnF = optFuzzy. Value 根据单选按钮状态判断是
否为模糊查询
```

根据各复选框的选定状态调用 Sub 过程 SetParam 为选定的参数赋值

```
If chkID. Value = vbChecked Then SetParam cmm,
0, txtID. Text, blnF
```

```
If chkName. Value = vbChecked Then SetParam
cmm, 1, txtName. Text, blnF
```

```
If chkSex. Value = vbChecked Then SetParam
cmm, 2, cboSex. Text, blnF
```

```
If chkProf. Value = vbChecked Then SetParam
cmm, 3, cboProf. Text, blnF
```

```
If chkDept. Value = vbChecked Then SetParam
cmm, 4, cboDept. Text, blnF
```

```
Set rs = cmm. Execute 执行存储过程并返回结果
集
```

```
Set DataGrid1. DataSource = rs 显示结果
```

在“全部”按钮的单击事件中用连接对象执行省略全部参数的存储过程。

```
Set rs = cnn. Execute ( " ProcQuery" , , adCmd-
StoredProc)
```

6 结束语

Jet 数据库 Access 小巧易用,便于维护,在桌面数据库应用系统中被广泛采用。在数据吞吐量不太大的环境中,许多中小企业已成功地将用于 C/S 或 B/S 架构。存储过程的引入无疑会显著提升 Jet 数据库的效率,进而优化应用系统的整体性能,因此具有较强的实用性。

参考文献

- 1 Microsoft Corp. CREATE PROCEDURE Statement [CP/CD]. JETSQL40. CHM, 1999.
- 2 Microsoft Corp. CREATE PROCEDURE 语句 [EB/OL]. <http://office.microsoft.com/zh-cn/assistance/HP010322192052.aspx>.
- 3 Microsoft Corp. ADOX 程序员参考 [CP/CD]. ADO210. CHM, 2000.
- 4 Microsoft Corp. ADOX 2.8 Programmer's Reference [DB/OL]. <http://msdn.microsoft.com/library>.
- 5 杨伦、苗君, ADOX 基本原理及应用实例 [J], 计算机与现代化, 2002, (4): 61-65.