

利用 JET 引擎存取数据库的分析

焦永军 王涛 (清华大学精密仪器系 100084)

摘要:本文分析了在 VB4.0 中,利用 JET 引擎存取数据库的一般机制,重点对 ODBC 数据源的存取进行了举例说明。

关键词:DAO Jet 数据库引擎 SQL

一、DAO(Data Access Object)对象模型

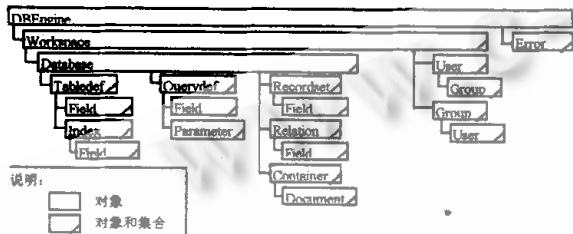


图 1 DAO 对象层次模型

微软的 Jet 数据库引擎(Jet database engine, 以下简称 Jet 引擎)由图 1 中的对象组成。这组对象模型成为 DAO。Visual Basic 的数据存取功能和 Access 都建立在此引擎之上。其实 Jet 引擎由一个内嵌类型库的动态连接库,如 DAO3032.DLL(32 位的 VB4 中)实现,是一个 OLE 自动化进程服务器,它能输出图 1 所示的对象。利用 VB 的 Object Browser 能更详细地观察 Jet 引擎所输出的对象及其属性和方法(说明:在 32 位 VB4.0 中使用的是 Jet 3.0;在 16 位 VB4.0 中使用的是 Jet 2.5)。

DAO 中主要对象的功能如下:

DBEngine 代表 Jet 引擎,是预定义的对象,不能再创建。主要用于控制 Jet 引擎,创建 Workspace 对象等。

Workspace 对象定义用户会话(Session)。会话打开数据库,进行事务处理以及建立基于用户名和口令的工作组安全性机制。当 Jet 引擎启动时,作为缺省系统创建 DBEngine.Workspaces(0)。当不指定 Workspace 而使用 OpenDatabase 方法时,缺省对象 DBEngine.Workspaces(0)被使用。Workspace 的事务处理影响它的所有 Database 对象,因此必须对独立使用事务的 Database 创

建独立的 Workspace。

Database 管理一个打开的数据库,可以建立或修改库中的表(TableDef),查询(QueryDef),关系(Relation),记录集(Recordset)等。可以执行动作查询(Action Query)。所谓动作查询是添加、删除、更新及建表等的查询,也包括 SQL 直通式(Pass Through)查询。TableDef 管理基表和连接表(attached table)。

QueryDef 管理存储的查询(Stored Query),查询可以是参数化的。可以把存储的查询看作已经编译过的 SQL 语句,其执行速度比一般查询要快。

Recordset 管理基表中的和查询返回的行记录。往往通过执行一个选择查询(Select Query)建立 Recordset 对象。选择查询由 Select 语句组成,和动作查询不同,它返回数据集。

二、利用 DAO 存取数据的一般过程

建立会话环境指建立必要的独立的 Workspace,创建用户及其口令。如无以上必要,该步可以省略。

DAO 可以打开三种数据库:

1. MDB 格式的 Jet 本源数据库。对此种数据库的操作,实际是 Access 数据库操作的一个子集合,本文不再专门介绍。

2. 可安装的 ISAM 数据库,包括 Fox, dBase, Excel, Btrieve 等桌面数据库。

3. 可通过 ODBC 驱动器存取的数据库,主要指 SQL 数据库。如 SQL Server, Oracle 等。

后两种数据库对 Jet 引擎来说是外部数据库。第 2 种数据库其实是一些平面文件,打开此种类型的数据库,只需指定一个存放数据文件的目录名作为库名即可。

对外部数据库,有两种打开的方法:

(1)直接打开。简称直接法。

(2)先打开一个 Jet 数据库,再建立和外部数据文件或外部 ODBC 数据源有关联的连接表,通过连接表操纵外部数据。简称间接法。

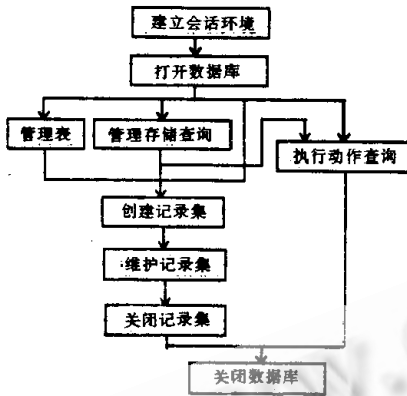


图2 DAO存取数据的一般过程

对 ISAM 数据库,两种打开方式差别不大;但对 ODBC 数据源,Jet 建议使用间接法。因为当直接打开 ODBC 数据库时,Jet 引擎要向外部数据库发送一系列的查询以确定它能获得的所有表和视图的名称和模式。在随后对表创建记录集时,Jet 引擎再发送一系列查询以确定该表的模式。所有这些查询都耗费时间,尤其对远程 SQL 数据库。因此间接法的效率更高。连接表保存的是外部数据表的结构,而非外部数据表本身。另外当直接打开外部数据库时,DAO 中的一些对象是不可用的,如 QueryDef、User 和 Group。

创建记录集即建立光标。DAO 提供了三种类型的记录集:

①表集。直接存取整个数据库表。只适用于 Jet 和 ISAM 数据库。如果对 ODBC 数据库实现表集,则要耗费很大的资源,这一般是不切实际的。

②瞬态集(相当与 ODBC 的静态光标)。不可更新,瞬态集打开后,集合的成员,顺序及值都保持固定不变。创建瞬态集时,DAO 取回整行的数据,因此瞬态集只适用于小记录集和不包含 Memo 或 OLE Object 字段的记录集。确保返回只需要的字段也可以提高检索速度。如不需要双向滚动的能力,可以创建只向前滚动的瞬态集。

③动态集(相当与 ODBC 的键集光标)。创建动态集时,DAO 只保存整个结果集的键值(keys),因此集合的成

员、顺序都保持固定不变,但值的改变在下次取数据时是可见的。其实这种类型的记录集称作动态集是不适合的,因为它并不能察觉被其他用户删除或插入的记录(因为键集的缘故)。键集的使用提高了检索速度,减少了网络流量,因此适用于比较大的记录集和包含 Memo 或 OLE Object 字段的记录集。但对太大的记录集,要保存其整个的键集,是不适用的。创建动态集耗费的机器资源最大。

维护记录集,是指对记录集进行编辑、删除、添加和更新(记录集和类型许可)。以行为单位进行。

对 SQL 数据库,经常采用动作查询。它适用于以下情况:执行存储过程;运行用外部语法(非 Jet 引擎语法)写成的 SQL 查询;在 SQL 服务器上创建数据库,表或索引;创建或管理触发器,缺省,规则或存储过程;执行系统管理和维护任务;在一个批中执行多个插入或更新语句等。

在关闭记录集之前,执行必要的更新,最后关闭数据库,否则可能导致数据丢失。

三、存取 ISAM 数据库

存取 ISAM 数据库,实际是存取平面磁盘文件,除了一般的数据库文件外,通常还包括索引或模式文件(对 text 数据)。ISAM 驱动器在存取数据时,须要后一种文件。对所有 ISAM 数据库的存取都是相似的,现以 dBase III 为例进行说明。

例:把 employees 表中值为 Sales Representative 的雇员的 title 改成 Account Executive。

间接法:

```

Dim dbEmp As Database, tdfLinkedEmp As TableDef
Dim rdsEmp as Recordset, qdfEmp as QueryDef
Set dbEmp = OpenDatabase("C:\Access\Data\Test.mdb") '打开 Jet 数据库
Set tdfLinkedEmp = dbEmp.CreateTableDef("Linked dBase Table")
tdfLinkedEmp.Connect = "dBase III; DATABASE = C:\Data\"
'数据库目录为 C:\Data\ tdfLinkedEmp.SourceTableName = "employees" '设置连接信息
dbEmp.TableDefs.Append tdfLinkedEmp '附加 tdfLinkedEmp, 创建连接
Set rdsEmp = dbEmp.OpenRecordset("Linked dBase Table", -dbOpenDynaset) '创建动态集
  
```

```

rdsEmp.MoveFirst '定位到动态集中第一条记录
Do Until rdsEmp.EOF '开始循环
    If rdsEmp! title = "Sale Representative" Then '检查 title 字段值
        rdsEmp.Edit '允许编辑
        rdsEmp! title = "Account Executive" '改变 title 字段值
        rdsEmp.Update '保存变化
    End If
    rdsEmp.MoveNext '定位到下一条记录
Loop
rdsEmp.Close '关闭动态集
dbEmp.Close '关闭数据库
直接法:
...
Set dbEmp = OpenDatabase("C:\Data\ ", False, False, "dBase III;") '打开 dBase 库
Set rdsEmp = dbEmp.OpenRecordset("SELECT * FROM employees - WHERE title = 'Sales Representative'", dbOpenDynaset) '创建动态集
Do Until rdsEmp.EOF
    rdsEmp.Edit
    rdsEmp! title = "Account Executive"
    rdsEmp.Update
    rdsEmp.MoveNext
Loop
...
其实用动作查询更简明:
...
Set dbEmp = OpenDatabase("C:\Data\ ", False, False, "dBase III;")
dbEmp.Execute "UPDATE employees SET title = 'Account Executive' - Where title = 'Sales Representative'"
dbEmp.close
前两种方法在性能上差别不大,最后一种方法简明,速度快。但对 ODBC 数据库情况有所不同。

```

四、存取 ODBC 数据库

1. 配置。首先要安装好 ODBC 驱动器,并用 ODBC 管理器或 RegisterDatabase 方法创建数据源名称(DSN)。

ODBC 驱动器安装程序不会把初始化缺省设置写入系统注册表,用户必须手工创建 Jet \ 3.0 \ Engines \ ODBC 子键,然后在此子键下写入设置值。常见的设置项包括注册超时,查询超时,连接超时等等(如用户没有建立此子键,DAO 使用缺省值)。

2. 存取方法。对上节中的例子,直接法语法如下:

```

...
Set dbEmp = OpenDatabase("", False, False, "ODBC; DSN = SqlServer; - "UID = sa;PWD = sql;DATABASE = pubs;")
...

```

为了提高性能,一般采用以下三种方法:

(1)间接法。对上节中的例子,间接法语法如下:

```

...
tdfLinkedEmp.Attributes = dbAttachedODBC
tdfLinkedEmp.Connect = "ODBC;DSN = SqlServer; - "UID = sa;PWD = sql;DATABASE = pubs;"
tdfLinkedEmp.SourceTableName = "Employees"
...

```

间接法不支持 SQL 直通式查询。如果在间接法中,对附加的 ODBC 表创建动态集,然后进行操作(如上面的例子),会引起较大的网络流量,因此速度和效率也是很低的。一般采用(3)中的方法。但如果确实需要象上面那样编码,那么应该在 DAO 中使用“高速缓存”技术。

Recordset 对象的属性 CacheSize 和 CacheStart 允许在本地存放 ODBC 动态类型的记录集,这能显著加速操作的速度。CacheSize 说明缓存的行数,CacheStart 说明缓存的第一行记录的书签。用 FillCache 方法向缓存中添入服务器数据。上面的例子用高速缓存法如下:

```

...
const EmpCache % = 100 '对 employees 的缓存值
...
Set rdsEmp = dbEmp.OpenRecordset("Linked dBase Table", - dbOpenDynaset) '创建动态集
rdsEmp.CacheSize = EmpCache
I = 0
rdsEmp.MoveFirst
Do Until rdsEmp.EOF
    If I Mod EmpCache = 0 Then
        rdsEmp.CacheStart = rdsEmp.BookMark
    End If
    I = I + 1
    rdsEmp.MoveNext
Loop

```

```

rdsEmp.FillCache
End If
If rdsEmp! title = "Sale Representative" Then
    rdsEmp.Edit
    rdsEmp! title = "Account Executive"
    rdsEmp.Update
End If
rdsEmp.MoveNext
I = I + 1

```

```

Loop
...

```

(2) 创建 QueryDef 对象, 和 ODBC 数据库建立参考。见第(3)小节。

(3) SQL 直通(PassThrough)式查询。采用 SQL 直通式查询(以下简称 SPT)的理由和前面的动作查询相同。对 ODBC 数据源, 动作查询包括在 SPT 中。SPT 绕过 Jet 引擎的处理而直接把 SQL 语句发送到 SQL 数据库, 因此允许 SQL 中使用 SQL 数据库的“方言”语法。如果 SPT 返回结果集, Jet 引擎只能为此创建瞬态集。有两种 SPT 方法:

①手工编码。

```

...
Set dbEmp = OpenDatabase("C:\Access\Data\
Test.mdb") '打开 Jet 数据库
dbEmp.Connect = "ODBC; DSN = SqlServer; - "UID
= sa; PWD = sql; DATABASE = pubs;"
'带 dbSQLPassThrough 项引用 Execute, 不能有结果
集返回
dbEmp.Execute "UPDATE employees SET title = '
Account Executive' - Where title = 'Sales Representative'"
dbSQLPassThrough - dbEmp.Close

```

②(2)的方法。

```

...
set qdfEmp = dbEmp.CreateQueryDef("employees
Query")
'设置连接串以通知 Jet 引擎这是 SPT, 至此 Jet 引
擎不再分析 SQL 语法
qdfEmp.Connect = "ODBC; DSN = SqlServer; UID =
sa; PWD = sql; - DATABASE = pubs;"
'如果 SQL 中有参数, 则必须符合 Jet 引擎的 SQL
语法
qdfEmp.SQL = "UPDATE employees SET title = '

```

```

Account Executive' - WHERE title = 'Sales Representa-
tive'"

```

```

dbEmp.Execute dbSQLPassThrough

```

如果执行 SQL, 返回结果, 还必须设置属性 qdf.ReturnRecords = True, 代替 Execute, 而执行以下方法: qdf.OpenRecordset("Employees Query", dbOpenSnapshot)

②比①灵活, 通常被采用。但如果使用带参数的方言查询, 必须使用①。

3. 性能优化

存取 ODBC 数据源时, 往往采用 Client/Server 结构。为了优化性能, 应注意以下问题:

(1) 仅请求必需的数据。可以采用适当的 Where 子句, 限制返回的字段等。

(2) 局部存放数据很少变化的远程表或视图。

(3) 仅使用满足要求的功能。如果瞬态集能满足要求, 就不应使用动态集。

(4) 注意 ISAM 数据库(包括 Jet 数据库)和 ODBC 数据库编码方式的差别。DAO 的很多编码方式对 ISAM 数据库比较有效, 但对 ODBC 数据库则是不可接受的。例如第三大节中的直接法和间接法, 对 ODBC 数据库速度会很慢, 必需采用动作查询或 SPT。

(5) 操作尽量在服务器上完成。在服务器上执行复杂查询, 把查询和修改作为一个逻辑放在存储过程中。

(6) 锁定的选用。当存取 ODBC 数据库时, 为减少并发冲突, 一般使用乐观锁定。只有在必要时才使用悲观锁定, 此时把属性 LockEdits 设为 True 并不起作用, 必须采用 SQL“方言”。比如对 SQL Server 6.0 可以采用以下 SQL 语句在事务中生成一个表级排它锁:

```

SELECT * FROM employees WHERE empid > 100
(TABLOCKX HOLDLOCK)

```

应该指出在存取 ODBC 数据库时, DAO 在不少方面做得并不好。如光标种类少(如没有真正的动态集), 不能高效率地进行参数查询, 无法直接处理存储过程的输出参数和返回值等。VB4.0 企业版中包含一个远程数据对象 RDO, 能较好的处理以上问题。

参考文献

- [1] Visual Basic 4.0 编程。科学出版社。1997
- [2] 网络数据库实现—SQL Server 6.0。科学出版社。1997

(来稿时间: 1997年6月)