

数据库应用程序需注意的问题 PDF转换可能丢失图片或格式
，建议阅读原文

https://www.100test.com/kao_ti2020/273/2021_2022__E6_95_B0_E6_8D_AE_E5_BA_93_E5_c97_273095.htm

总的来说，提高应用程序性能的最好的方法是发现应用的瓶颈之所在，和数据库进行交互的性能无疑是决定应用程序性能的重要环节之一。

因为ADO是当前最新的基于组件的数据库编程的接口，这里我们主要讨论用ADO编程所需要注意的问题，因为ADO是一个和编程语言无关的COM组件系统，所以这里讨论的要点适用于所有的编程语言和编程环境，比如：VB、VBScript、VC、Java等等。显式的定义对象变量的类型：实际上，这条准则不仅适用于ADO编程，也适用于其他的COM对象相关的编程，因为如果一开始就定义变量类型的话，编译器在编译的时候就可以知道变量的类型，编译器实际上就采用vtable偏移的方式来得到具体的COM对象包含的方法的地址（这一点和C中的虚函数的地址的获取类似），但如果一开始不指定变量类型的话，比如简单的采用如下的语句：DIM myCon as Object 或者是 DIM myCon那么编译器在编译的时候就不能得到变量的类型，而只能在运行的时候动态的得到方法的信息（通过使用接口IDispatch的方法Invoke来实现的），这样为了得到方法的地址和相关的变量情况就需要在内部进行两次调用，无疑就使速度降低。当浏览记录的时候，绑定列到具体的字段对象上去这个意思就是说在一开始的时候我们就建立对字段对象的引用，避免在每次得到记录的时候需要

在Rcordset::Fields中进行查找而增加系统的开销。比如可以采用如下的示例代码形式：Private Sub TblBrowse_Click() Dim

```
fld1 As ADODB.Field Dim fld2 As ADODB.Field Dim rs As  
ADODB.Recordset set rs=g_cn.execute(...) g_cn为全  
局adodb.connection对象 Set fld1 = rs.Fields("id") 数据表的字段  
Set fld2 = rs.Fields("name") 数据表的字段 If rs.BOF = False Then  
While rs.BOF = False Debug.Print fld1.Value Debug.Print  
fld2.Value rs.MoveNext Wend End If rs.Close End Sub
```

尽量采用SQL语句和存储过程进行数据更新 尽管采用Recordset对象来更新数据是非常方便的，但是它的开销也更大，所以如果可能的话，就要采用SQL语句来更新数据。使用存储过程而不是单一的SQL语句来获取信息。因为存储过程是在服务器端执行的，只把结果返回到客户端，这样一方面可以降低网络进行数据交互的开销，另一方面使系统更加容易维护，并且保持数据的一致性。而如果使用recordset来得到结果的话，通过数据源对象返回的查询集不仅包含了数据，而且也包含了元数据（metadata），在有些时候元数据可能比数据本身还要大，这样系统的开销无疑也增加了不少。如果必须要使用游标的话，最好使用集合的方法对单条的SELECT语句进行操作Recordset::get_Collect和Recordset::put_Collect方法是Recordset对象的快捷方式，可以使你快速的得到一个字段的值而不需要获得关于一个字段的引用。可以参考如下的示例代码：

```
Sub Collect() Dim rs As New Recordset  
rs.ActiveConnection = "..."  
rs.Source = "一条SQL查询语句"  
rs.Open  
Debug.Print rs.Collect(0), rs.Collect(1), rs.Collect(2)  
Debug.Print rs!au_id, rs!au_fname, rs!au_lname
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com