开发技术整合应用丛书

ASP+网络与数据库整合应用

精锐创作组　　编著

人民邮电出版社

需要全本请访问 www.ertongbook.com

5.3 表之间的连接

5.3.1 UNION 运算

UNION 运算简介

UNION 运算用于创建一个联合查询，它组合了两个或更多的独立查询或表的结果。

它的语法形式如下所示：

```
[TABLE] query1 UNION [ALL]
[TABLE] query2 [UNION[ALL]
[TABLE] queryn [ . ]]
```

- query1-n：SELECT 语句、已存储的查询的名称、或冠以 TABLE 关键字的表的名称。
- ALL：按照默认规定，使用 UNION 运算时不返回重复的记录，然而，可以包含 ALL 来确保返回所有的记录，这样，运行查询的速度也会快些。

所有在一个联合运算中的查询，要求有相同数目的字段；但是，字段不必大小相同或数据类型相同。

只能在第一个 SELECT 语句中使用别名，因为它们在其他语句中已被省略。在 ORDER BY 从句中，可引用在第一个 SELECT 语句中被调用的字段名。

要将返回的数据分组，可以在每一个 query 参数中使用 GROUP BY 从句或 HAVING 从句。

要想以指定的顺序来显示返回的数据，可以在最后一个 query 参数的末尾使用 ORDER BY 从句。

5.3.2 LEFT JOIN 和 RIGHT JOIN 运算

LEFT JOIN 和 RIGHT JOIN 运算简介

LEFT JOIN 和 RIGHT JOIN 运算可用在任何的 FROM 从句中，用以组合来源表的记录。

它的语法形式如下所示：

```
FROM tabl1   [ LEFT | RIGHT ] JOIN tabl2
ON tabl1.fid1 cpopr tabl2.fid2
```



- table1 和 table2：记录被组合的表的名称。
- fid1 和 fid2 ：被连接的字段的名称，这些字段必须有相同的数据类型及包含相同类型的数据，但它们不需要有相同的名称。
- cpopr ：任何的关系比较运算符？—— =、<、>、<=、>=、<>。

使用 LEFT JOIN 运算来创建一个左边外部连接。左边外部连接将包含了从第一个（左边）表中开始的两个表中的全部记录，即使在第二个（右边）表中并没有匹配值的记录。

使用一个 RIGHT JOIN 运算创建一个右边外部连接。右边外部连接将包含了从第二个（右边）表中开始的两个表中的全部记录，即使在第一个（左边）表中并没有匹配值的记录。

可以连接多个 ON 从句。

5.3.3 INNER JOIN 运算

INNER JOIN 运算简介

INNER JOIN 运算：组合两个表中的记录，只要在公共字段中有相符的值即可。

它的语法形式如下所示：

```
FROM tabl1 INNER JOIN tabl2
ON tabl1.fid1 cpopr tabl2.fid2
```

- table1 和 table2：要组合记录的表的名称。
- fid1 和 fid2 ：被连接字段的名称，如果这些字段不是数值的，被连接的字段必须是相同的数据类型，并且包含相同类型的数据，但是它们不必名称相同。
- cpopr ：任何关系比较运算符—— =、<、>、<=、>=、<>。

在任何 FROM 从句中都可以使用 INNER JOIN 运算，这是最普通的连接类型。只要在这两个表的公共字段中有相等值，内部连接将组合两个表中的记录。

在一个 JOIN 语句中可以连接多个 On 从句，JOIN 语句本身也可以嵌套连接。在一个 INNER JOIN 之中，还可以嵌套 LEFT JOIN 或 RIGHT JOIN，但是 LEFT JOIN 或 RIGHT JOIN 中不能嵌套 INNER JOIN。

5.4 数据定义语言

在前面的各节中，我们介绍了使用 SELECT 语句及其从句有选择地返回数据库中的数据、与查询相关的函数、创建表查询语句 SELECT...INTO、追加查询语句 INSERT...INTO、

更新查询语句 UPDATE、删除查询语句 DELETE 以及 UNION、LEFT JOIN、RIGHT JOIN、INNER JOIN 运算，这四种运算使查询不仅仅局限于一个表中。

在这一节中介绍的数据定义语言，Mirosoft Jet 数据库引擎不支持使用，而使用 ADO 或 DAO 的创建方法。不过，为了让读者对 SQL 语言有个全面的了解，我们还是简单介绍一下。

- CREATE TABLE 语句：创建表。
- CREATE INDEX 语句：创建索引。
- ALTER TABLE 语句：修改表、创建索引、删除索引。
- DROP 语句：删除已存在的表，或从表中删除已存在的索引。

5.4.1 CREATE TABLE 语句

CREATE TABLE 语句用于创建一个新表。

它的语法形式如下：

```
CREATE TABLE tabl_nam
(fid1 type [(sze)] [NOT NULL] [idex1]
[, fid2 type [(sze)] [NOT NULL] [idex2] [, .]]
[, CONSTRAINT multifididex [    , .]])
```

- table_nam ：欲创建的表的名称。
- fid1 和 fid2 ：在新表中欲创建的字段的名称，至少创建一个字段。
- type：在新表中的字段的数据类型。
- sze ：字段的字符长度(文本及二进制字段)。
- index1 和 index2：定义单一字段索引的 CONSTRAINT 从句。
- multifidindex ：定义多重字段索引的 CONSTRAINT 从句。
- NOT NULL：如果将一个字段指定为 NOT NULL，则新记录的该字段值必须是有效的数据。

例如：

```
CREATE TABLE 档案 (姓名 Text(10) NOT NULL, 年龄 Shrt)
```

这条语句的意思是：创建一个新表，表的名称是“档案”，表中有两个字段“姓名”和“年龄”。其中，“姓名”字段的数据类型为 Text，大小设为 10 字节，并且字段值不能为空；“年龄”字段的数据类型为 Short。



5.4.2 CREATE INDEX 语句

CREATE INDEX 语句用于在现有的表上创建新的索引。

它的语法形式如下所示：

```
CREATE [ UNIQUE ] INDEX idex_nam
ON tabl_nam (fid [ASC|DESC][, fid [ASC|DESC]        , .])
[WITH { PRIMARY | DISALLOW NULL | IGNORE NULL }]
```

- index_nam ：欲创建的索引的名称。
- table_nam ：包含该索引的现存表的名称。
- fid ：欲被索引的字段的名称。

要创建单一字段索引，在表名称后面的括号中列出字段名；要创建多重字段索引，列出包括在索引中的每一个字段的名称。如果索引为递减排序，使用 DESC 保留字；否则，索引总是递增的。

在可选的 WITH 从句中，可以使用强制数据有效性规则：

- 用 PRIMARY 保留字将索引字段设为主关键字。
- 通过使用 DISALLOW NULL 选项来禁止在新记录的索引字段中使用 Null 项。
- 通过使用 IGNORE NULL 选项，避免在索引中包含索引字段为 Null 值的记录。

也可以使用 ALTER TABLE 语句，在表中添加单一或多重字段索引，使用 DROP 语句删除用 ALTER TABLE 或 CREATE INDEX 创建的索引。

例如：

```
CREATE INDEX 新索引
ON 档案(姓名 A,  年龄 De)
With Dial Nul
```

以上语句为表“档案”创建名称为“新索引”的索引，该索引先按照“姓名”字段的升序排列，再按照“年龄”字段的降序排列，使用 Dialow Null 选项来禁止在新记录的索引字段中使用 Null 项。

5.4.3 ALTER TABLE 语句

ALTER TABLE 语句修改已用 CREATE TABLE 语句创建好的表。

它的语法形式如下所示：

```
ALTER TABLE tabl_nam
```


需要全本请访问 www.ertongbook.com

```
{ADD {COLUMN fid    type[(sze)] [NOT NULL] [CONSTRAINT idex] |
CONSTRAINT multifididex} |
DROP {COLUMN fid I CONSTRAINT idexnam      } }
```

- table_nam ：欲修改的表的名称。
- fid ：要在表内增加或删除的字段的名称。
- type：字段的数据类型。
- sze ：字段的字符长度(仅文本字段及二进制字段)。
- Index：字段索引。
- indexnam ：欲删除的多重字段索引的名称。

当使用 Add Column 在表中添加新的字段时，需要指定字段名、数据类型、还可以 (对文本和二进制字段)指定长度。

当使用 DROP COLUMN 删除字段时，只要指定欲删除的字段名即可。

使用 Add COLUMN 添加多重字段索引。

要使用 DROP CONSTRAINT 删除多重字段索引时，只要在 CONSTRAINT 保留字后面指定索引名即可。

不能同时添加或删除一个以上的字段或索引。

下面的语句实现在表“ 档案 ”中添加一个字段，删除“ 年龄 ”字段并删除“ 新索引 ”：

```
ALTER TABLE 档案
Add COLUMN 性别
DROP COLUMN 年龄
DROP CONSTRAINT 新索引
```

5.4.4 DROP 语句

DROP 语句从数据库中删除已存在的表，或从表中删除已存在的索引。

它的语法形式如下所示：

```
DROP {TABLE tabl_nam | INDEX idex_nam ON tabl}
```

- table_nam ：欲删除的表的名称，或欲删除的索引所在的表的名称。
- index_nam ：欲从表中删除的索引的名称。

必须先关闭表，然后才能删除此表或此表中的索引，例如：

```
DROP INDEX 新索引 ON 档案
DROP TABLE 档案
```



上面的语句删除了索引“ 新索引 ”（我们在以前举例时已经删除了“ 新索引 ”，但是，为了说明 DROP，不妨再删除一遍）和表“ 档案 ”。

5.5 小结

在这一章里，我们主要介绍了以下内容：

- 使用 SELECT 语句及其从句有选择地返回数据库的数据，及查询相关函数。
- 创建表查询 SELECT...INTO 语句，追加查询语句 INSERT...INTO。
- 更新查询语句 UPDATE，删除查询语句 DELETE。
- 数据库记录的连接操作，如 UNION、LEFT JOIN、RIGHT JOIN、INNER JOIN 运算，这四种运算使查询不仅仅局限于一个表中。

还介绍了数据定义语言：

- CREATE TABLE 语句创建表。
- CREATE INDEX 语句创建索引。
- ALTER TABLE 语句修改表、创建索引、删除索引。
- DROP 语句删除已存在的表，或从表中删除已存在的索引。



第 6 章
DO 技术



6.1 ODBC 简介

现在最流行的连接数据源的方法是 ODBCPI (Open Databasonnetivity API) 。

ODBC 是一种 C/C++应用程序编程接口(API)。无论对于任何一种客户/服务器关系数据库管理系统(RDBMS)，还是最流行的索引顺序访问方法(ISAM)数据库(Jet、FoxPro)以及电子表格，都能找到 32 位 ODBC 驱动程序，微软公司的 Offi 中大多数和数据库连接的特性都依赖于 ODBC 驱动程序。

6.1.1 ODBCPI 的结构

Windows 动态链接库(.DLL)组成了 ODBCPI ，对于 ODBC 驱动程序可用到的所有数据库类型，动态链接包括两组函数，可以提供基本的数据库服务。

数据库服务

数据库服务的主要内容包括：

- 添加、配置和删除 ODBC 数据源。



一个数据源是对一个数据库的有名连接，通常称为数据源名称(DSN)。连接不同类型的数据库需要不同的驱动程序，ODBC 使用 Odbct.dl 库来配置和删除数据源；而 Odbc32.xe 是一个独立的、可执行的 32 位应用程序（ODBC 数据管理器），用于配置 ODBC 数据源，Odbc32.xe 的图标位于“控制面板”窗口中。

- 管理从客户端到数据库服务器的查询语句和其他的 SQL 语句的通信，并且返回查询结果或执行动作要求的反向确认。

数据源的 ODBC 驱动程序将 SQL 语句传送到驱动程序，当 Jet 数据库引擎或客户/服务器的 RDBMS（关系型数据库管理系统）处理 SQL 语句后，数据源的 ODBC 驱动程序将结果返回到应用程序。

ODBC 驱动程序

下面我们了解一下 ODBC 驱动程序，它可以分为单层 ODBC 和多层 ODBC：

- 单层的 ODBC 被设计成不能直接处理 ANSI SQL 语句，而是将 ANSI SQL 语句转换为一系列长的低级指令，直接操作数据库文件。

当前，Mirosoft 的单层 ODBC 驱动程序连接 dBas 、FoxPro、Viual FoxPro 、Paradox、Exce 和文本文件。

- 多层 ODBC 驱动程序依靠客户/服务器 RDBMS 来处理 ANSI SQL 语句。

由于服务器只返回符合 SQL 语句条件的特定记录，所以网络上的数据流量被最小化。VB6 中包括 Mirosoft SQL Server 4.2x 、6.x 和 7.0 的 32 位 ODBC 驱动程序，即 Sqlsvr32.dll；VB6 同样也包括 Orac 7.3 和 8.x 的 32 位 ODBC 驱动程序。为了使用 Orac 的驱动程序，必须在计算机上安装 Orac 客户端动态库。

6.1.2 ODBC 添加新数据源

下面我们演示一下向 ODBC 添加新数据源。

进入 ODBC

单击 Windows 系统的菜单“ 开始 ”⇨“ 设置 ”⇨“ 控制面板 ”，进入“ 控制面板 ”窗口；双击控制面板的“ ODBC 数据源(32 位) ”，打开“ ODBC 数据管理器 ”对话框，如图 6-1 所示。

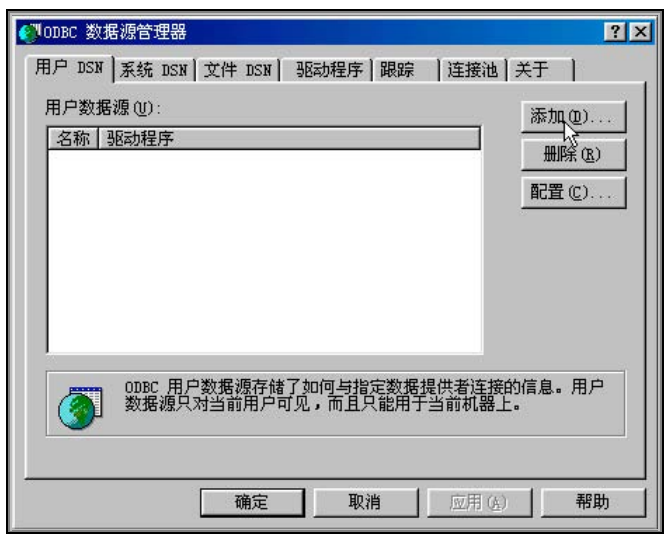


图 6-1 ODBC 数据管理器

单击“ 添加 ”按钮，进入“ 创建新数据源 ”对话框。

添加数据源

在“ 创建新数据源 ”对话框中，选择第一项 Mirosoft A Driver(*.db) 以创建安装 A 数据库所需的驱动程序，如图 6-2 所示。



图 6-2 “ 创建新数据源 ” 对话框

单击“ 完成 ” 按钮，进入“ ODBC Mirosoft A 安装 ” 对话框，如图 6-3 所示。



图 6-3 “ ODBC Miroft A 安装 ” 对话框

在“ ODBC Mirosoft A 安装 ” 对话框中，单击“ 选取 ” 按钮，进入“ 选定数据库 ” 对话框，如图 6-4 所示。

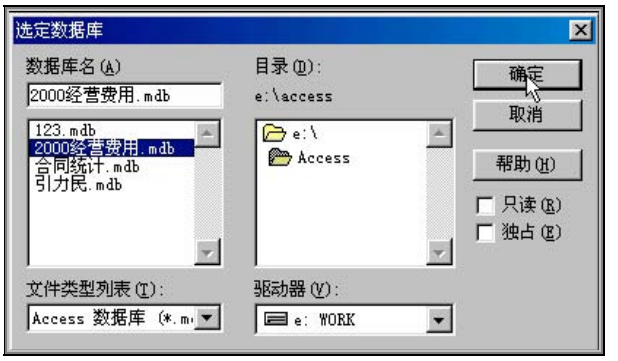


图 6-4 “ 选定数据库 ” 对话框

需要全本请访问 www.ertongbook.com

在“选定数据库”对话框中，选择“E:\A\2000　　经营费用.mdb”，单击“确定”按钮。

这时返回图 6-3 的“ODBC Mirosoft A　　安装”对话框，为这一数据源填上“数据源名”和“描述”后，单击“确定”按钮，返回图 6-1 的“ODBC 数据管理器”对话框。这时，对话框的空白栏中出现了已添加的一个数据源，如图 6-5 所示。

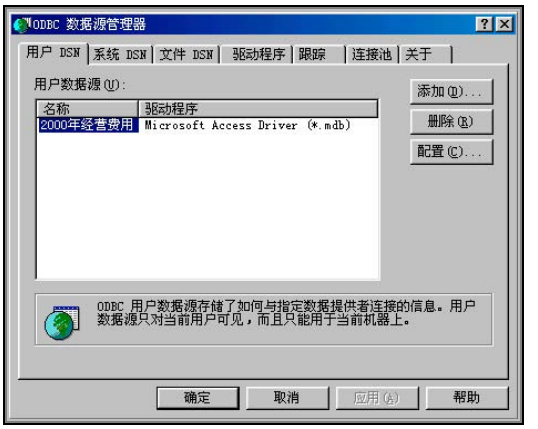


图 6-5　添加了新数据源后的“ODBC 数据管理器”

ODBC 数据管理器不仅能添加数据源，还可以配置、删除数据源。

6.2 ADO 技术

6.2.1　ADO 技术概述

ADO 是 Mirosoft　处理关系数据库和非关系数据库中信息的最新技术（一般的关系数据库管理系统用表来操纵信息，但并非所有的数据源都遵从这一模式）。

ADO 基于微软最新的被称为 OLE　DB 的数据访问模式。它是专门为了给大型的商业数据源提供访问而设计的，包括传统的关系数据库表、电子邮件系统、图形格式、Internet 资源等等。

6.2.2　ADO 对象模型

ADO 对象模型如图 6-6 所示。

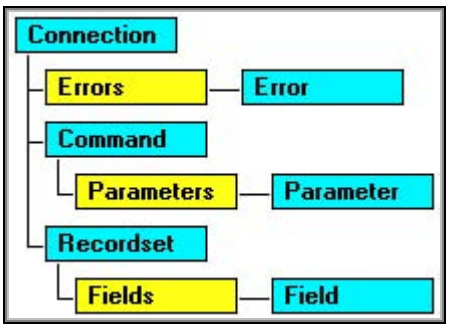


图 6-6　ADO 对象模型

ADO 顶层有三个对象 Conntion　、Cand　　、Reordset ，其中：

- Conntion　对象可以指定数据源，建立与数据源的连接以及隔离特定的连接。
- Cand　　对象对数据源执行指定的命令。它可以接受 SQL 语句、表的名称和存储过程的名称，执行 SQL 查询、更新数据、插入记录等。
- Reordset　对象表示的是来自表或命令执行结果的记录全集，操作来自提供者的所有数据。

模型也反映出了 ADO 的四个集合：

- 每个对象都有 Properti　（属性）集合。
- Errors（错误）集合包含连接数据库失败时产生的所有 Error　对象。
- Paramters （参数）集合包含 Cand　　对象的所有 Paramter　对象。
- Fids　（字段）集合包含 Reordset　对象的所有 Fid　对象。

6.3 ADO 的 Ccti　　对象

Conntion　对象是一个主要的顶层 ADO 组件，必须成功地打开一个连接到数据源的 Conntion　对象，才有可能使用相关的 Cand　　对象或 Reordset　对象。

6.3.1　Cti　　对象的属性

通常在连接数据库之前首先要对 Conntion　对象的属性进行设置，以便对连接进行配置，下面分别介绍 Conntion　对象的属性。

CursorLocati 属性

CursorLocation 属性设置游标类型，通常有以下两种选择：

- 将 CursorLocation 属性设置为 adUsnt ，则使用由本地游标库提供的客户端游标。

本地游标服务通常允许使用的许多功能可能是驱动程序提供的游标无法使用的，因此使用该设置对于那些将要启用的功能是有好处的。

- adUsServer 是 CursorLocation 属性的默认值，使用数据提供者的或驱动程序提供的游标。这些游标有时非常灵活，对于其他用户对数据源所作的更改具有额外的敏感性。

CursorLocation 属性仅对属性设置后建立的连接才有影响，更改 CursorLocation 属性不会影响现有的连接。

CursorLocation 属性允许在可用于数据提供者（数据库）的各种游标库中进行选择。通常，可以选择使用客户端游标库或位于服务器上的某个游标库。

Mode 属性

Mode 属性只能在关闭 Cnnetion 对象时方可设置，它用于设置对数据的可用权限，可返回当前连接正在使用的访问权限。通常是下列常量之一：

- adModeUnknown：默认值，表明权限尚未设置或无法确定。
- adModeRead：表明权限为只读。
- adModeWrite：表明权限为只写。
- adModeReadWrite：表明权限为读/写。
- adModeShareDenyRead：防止其他用户使用读权限打开链接。
- adModeShareDenyWrite：防止其他用户使用写权限打开链接。
- adModeShareExclusive：防止其他用户打开链接。
- adModeShareDenyNone：防止其他用户使用任何权限打开链接。

当用于客户端 Conntion 对象时，Mode 属性只能设置为 adModeUnknow 。

State 属性

State 属性是只读的，它返回对象的状态是打开或是关闭，返回值是下列常量之一：

- adStateosed ：默认值，指示对象是关闭的。
- adStateOpe ：指示对象是打开的。
- adStatenneting ：指示 Reordset 对象正在链接。



- adStateExecuting：指示 Reordset 对象正在执行命令。
- adStateFetcng ：指示 Reordset 对象的行正在被读取。

可以随时使用 State 属性指定对象的当前状态，它不仅仅是 Cnnetion 对象的属性，也是 Cand 对象、Reordset 对象的属性，并返回它们的状态。

CtiTiut 属性

ConntionTiout 属性设置在终止尝试和产生错误前建立连接期间所等待的时间，默认值为 15 秒。

Versio 属性

Verson 属性返回 ADO 执行的版本号。

DefaultDatabas 属性

DefaultDatabas 属性设置连接的默认数据库，如果没有指定连接的数据库名称，就使用 DefaultDatabas 属性指定的数据库名称。对 SQL Server 而言，其值通常是 pubs。

Provider 属性

Provider 属性指示 Connetion 对象提供者的名称，如果没有指定提供者，该属性将默认为 MSDASQL(Mirosoft OLE DB Provider for ODBC) 。

关闭连接时 Provider 的属性为读/写，打开连接时该属性为只读。打开 Ction 对象或访问 Cnnetion 对象的 Properti 集合后该设置才生效；如果该设置无效，将产生错误。

也可以通过下面介绍的 ConntionString 属性或 Ope 方法的 CtionString 参数设置该属性。



最好不要在多处指定提供者，否则调用 Ope 方法连接数据库时可能会产生无法预料的后果。

CtiStrig 属性

ConntionString 属性设置建立连接数据源的信息，它包含以下几个参数：

- Provider：指定连接的提供者名称。
- Fi Nam ：指定特定提供者的文件名称。
- Reote Provider ：指定打开客户端连接时使用的提供者名称（仅限于远程数据服务）。

- Reote Server ：指定打开客户端连接时使用的服务器的路径名称（仅限于远程数据服务）。

设置 ConntionString 属性并打开 Cnnetion 对象后，提供者可以更改属性的内容。

下面的示例说明了使用 ConntionString 属性打开 Connetion 对象的不同方法，并使用 State 属性显示连接状态：

```
Imports Syste
Imports SystepotModeDegn
Imports SysteData
Imports SysteDrawg
Imports SysteWeb
Imports SysteWeb.SeState
Imports SysteWeb.UI
Imports SysteWeb.UI.WebCtro
Imports SysteWeb.UI.Htmtro
Imports Miroft.ViualBas
Imports SysteData.DO

Publas WebForm1
    Inrits SysteWeb.UI.Page
#Regi " Web Form Degnr Gerated Cde "
    Di WithEvets WebForm1 A SysteWeb.UI.Page
    Sub Ne()
        WebForm1 = Me
    End Sub
    'CODEGEN: Th produre required by th Web Form Degnr
    'Dot mdify it usg thdeditor.
    Private Sub Intialzepot()
    End Sub

#End Regi

    Proteted Sub WebForm1_ Load(ByVal Seder A SysteObjet, ByVal
SysteEvetArgs)
        If Not IsPotback Th  ' Eval true first ti bror hts th page
```

```
        End If
End Sub

Proteted Sub WebForm1_  Int(ByVal Seder A SysteObjet, ByVal SysteEvetArgs)
        'CODEGEN: Ththd cal required by th Web Form Degnr
        'Dot mdify it usg thdeditor.
        Intialzepot()
        CtiStrigX()
End Sub

'---- ObjetStateEnum Value ----
Ct adStated A Integer = &H0
Ct adStateOpe Integer = &H1
Ct adStatetig A Integer = &H2
Ct adStateExeutig A Integer = &H4
Ct adStateFetcg A Integer = &H8

Sub CtiStrigX()
        Di1, c2, c3 ADODB.ti

        c1.tiStrig = "DSN=ASPDe;"
        c1.Ope()
        'ODBC 标记打开连接

        c2 = NeDODB.ti()
        c2.tiStrig = "Data Sourc=ASPDe;"
        c2.Ope()
        '使用  DSN（数据源名）

        c3 = NeDODB.ti()
        c3.Ope("ASPDe")
        '使用 DSN 和单个参数而非连接
        RepoWrite("c3 state: " & GetState(c3.State) & "<br>")
        RepoWrite("c2 state: " & GetState(c2.State) & "<br>")
        RepoWrite("c1 state: " & GetState(c1.State) & "<br>")
        '显示连接的状态
```


需要全本请访问 www.ertongbook.com

1 = NeDODB.ti()