##  LabVIEW中访问数据库的几种不同方法

 虚拟仪器VI(Virtual Instruments)是National Instruments公司在其产品LabVIEW中首先提出的创新概念[1]。虚拟仪器系统的概念是测控系统的抽象。不管是传统的还是虚拟的仪器，它们的功能都是相同的：采集数据并进行分析处理，然后显示处理的结果。它们之间的不同主要体现在灵活性方面。虚拟仪器由用户自己定义功能，可以自由地组合计算机平台、硬件、软件以及完成应用系统所需要的各种功能。另外，虚拟仪器开发周期短、成本低、维护方便，易于应用新理论和新技术实现仪器的换代升级[2]。

现代的测试测量系统大多数需要对被测对象进行全方位检测，这必然会使获取的数据量急剧增长。面对大量的数据信息，采用数据库技术，可准确反映各类数据之间的密切联系，能够有效地管理和组织数据，是现代测试测量系统的发展趋势。但是现有的LabVIEW版本本身并不具备数据库访问功能，不能像VB、VC++、Delphi、PowerBuilder那样非常方便的进行数据库程序的开发。因此以LabVIEW编制的虚拟仪器系统需要其它辅助的方法来进行数据库访问。

**1、在LabVIEW中访问数据库的方式**

(1)利用NI公司的附加工具包中的数据库接口工具包LabVIEW SQL Toolkit进行数据库访问。该工具包集成了一系列的高级功能模块，这些模块封装了大多数的数据库操作和一些高级的数据库访问功能。它的优点是易于理解，操作简单，用户可以不学习SQL语法。缺点是需要另外购买且价格昂贵，无疑会增加系统成本。

(2)利用LabVIEW的ActiveX功能，调用Microsoft ADO对象，利用SQL语言实现数据库的访问。使用这种方法需要用户对Microsoft ADO以及SQL语言有较深的了解。

(3)通过第三方开发的免费工具包LabSQL访问。LabSQL利用Microsoft ADO以及SQL语言来完成数据库访问，将复杂的底层ADO及SQL操作封装成一系列的LabSQL VIs，简单易用。

(4)通过调用动态链接库DLL(Dynamic Link Library)访问。先利用其它语言如Visual C++编写DLL，再利用LabVIEW的调用库函数节点CLFN(Calling Library Function Node)调用此DLL访问数据库。但这种方法需要从底层进行复杂的编程才能实现，对非专业编程人员来讲是不现实的[3]。

(5)利用中间文件存取数据。先将数据存入文件之中，待测量结束后再用专门的工具将数据导入到数据库之中。这种方法需要磁盘文件作为中介，而且不具有实时性，不方便使用[4]。

实际应用中，前三种方法使用最为广泛，下面将重点讨论这三种方法。

**2、利用ADO技术访问数据库**

**2.1 ADO技术简介**

ADO(ActiveX Data Objects)是Microsoft为OLE DB设计的应用层接口，它为一致的数据访问接口提供了良好的扩展性而不局限于单一的数据源。

ADO接口编程模型中包括连接(Connection)、命令(Command)、参数(Parameter)、记录集(Recordset)、字段(Field)、错误(Error)、属性(Property)、集合(Collection)、事件(Event)等元素。其中Connection、Command、Recordset是三个主体对象，Parameters、Fields、Errors、Properties是四个集合对象[5]。

ADO的一般操作方式为：1连接到数据源；2指定访问数据源的命令；3执行命令；4将命令执行后产生的结果保存在缓存中；5更新数据库。ADO的主要优点是易于使用、高速度、低内存支出和占用磁盘空间较少[6]。

**2.2 LabVIEW中对ADO的调用**

ADO对象在LabVIEW中是以ActiveX对象的形式提供的。LabVIEW自4.1版本就引入了支持ActiveX自动控制的功能模块，在5.1版本之后支持客户和服务器双方，即虽然程序是在双方各自独立存在，但它们的信息是共享的。这种信息共享是通过客户端使用由服务器端发布的ActiveX控件来实现的。

图1展示的是LabVIEW中使用ActiveX控件的程序流程。ActiveX对象的打开和关闭是通过打开自动化节点(Automation Open)和关闭自动化节点(Close Reference)来实现的，属性的设置和获取则通过属性节点(Property Node)进行，而调用节点(Invoke Node)用于对象方法的调用。其中关键的是第二步，只有充分的利用“属性”和“方法”的设置才能成功地实现对数据库的访问[7]。



图1 LabVIEW中使用ActiveX控件的程序流程图

**2.3应用实例**

(1)建立ADO对象

在前面板控件模板的Refnum子模板中选择Automation Refnum项，把它拖放到前面板上，在其右键菜单中选择Select ActiveX Class>>Broswer…，弹出Select ActiveX Object对话框，在类型库Type Library下拉列表中选择Microsoft ActiveX Data Objects 2.7 Library Version 2.7，在下面的对象Objects列表栏中出现这个库对LabVIEW可用的对象，选中Connection对象，点击OK按钮即可。使用同样的方法可以建立Command、Recordset等对象。

(2)连接到数据源

在程序框图功能模板中选择Communication>>ActiveX子模板，选择其中的Automation Open与ADODB.\_Connection相连即可打开Connection对象。接着从ActiveX子模板中选择Invoke Node并与Automation Open相连，在其上单击右键选择Methods>>Open，即出现图2中所示的节点。其中“ConnectionSring”是连接到数据源的字符串，“UserID”和“PassWord”是连接到数据源的用户名和密码，正确设置这些参数后便可连接到数据源。

(3)生成SQL命令、执行命令

与上一步相同，用 Invoke Node调用Connection对象的Execute方法执行所要的操作。Execute方法所必需的参数为CommandText，这里为所要执行的SQL语句。例如可以使用Create命令创建表，用Drop命令删除表，用Insert命令向表中插入数据，用Select命令进行查询并返回操作结果等[8]。本例使用Create命令创建了一个名为Staff的表，包括编号、姓名和年龄三个字段，其中id字段为主键，自动编号。对数据表的删除和添加数据的操作过程与建表过程类似。如果要对执行后的记录进行显示或读取字段值则需要建立Recordset对象，并与Execute相连。

(4)关闭连接

对数据库访问操作完毕后要及时关闭连接对象以释放内存和所用的系统资源。首先使用Connection对象的Close方法关闭数据库连接，然后使用Close Reference关闭ActiveX自动化参数号。



图2 数据库中建表的程序框图

**3、利用LabSQL访问数据库**

**3.1 LabSQL简介**

LabSQL是一个免费的、多数据库、跨平台的LabVIEW数据库访问工具包，由Premkc Development公司的Chris Roth和Jeffrey Travis开发。目前的最新版本是LabSQL Release 1.1a，可以到网站 [http://jeffreytravis.com 免费下载。LabSQL支持Windows操作系统中任何基于OBDC的数据库，包括Access，SQL Server，Oracle，Pervasive，Sybase等。](http://jeffreytravis.com免费下载。labsql支持windows操作系统中任何基于obdc的数据库，包括access，sql/)

LabSQL利用Microsoft ADO对象和SQL语言来完成数据库访问，将复杂的底层ADO及SQL操作封装成一系列的LabSQL VIs，简单易用。

LabSQL VIs按照功能可以分为四类：Command VIs，Connection VIs，Recordset VIs和Top Level VIs。

Command VIs的功能是完成一系列的基本ADO操作，例如，创建或删除一个Command，对数据库的某一个参数进行读或写等。

Connection VIs的功能是管理LabVIEW与数据库之间的连接。

Recordset VIs用于对数据库中的记录进行各种操作，例如，创建或删除一条记录，对记录中的某一个条目进行读或写等。

Top Level VIs属于顶层的LabSQL应用，对前三类LabSQL VIs某些功能的封装，例如，SQL Execute.vi可用于直接执行命令[9]。

**3.2 LabSQL应用实例**

本应用实例是向数据表添加记录，所使用的数据源与上例相同，数据表为staff表。具体步骤如下：

第一步：建立与数据库的连接。首先通过ADO Connection Create.vi创建一个Connection对象，然后利用ADO Connection Open.vi建立与数据库的连接。数据库由ADO Connection Open.vi的ConnectionString指定，此参数由前面板的字符串控件提供，如图3所示。

第二步：生成SQL命令，执行命令。在前面板table控件中输入表名，name和age控件中输入字段值。在程序框图中使用Format Into String生成SQL命令，将它连接到ADO Connection Execute.vi即可执行，同时可在前面板Command Text控件中显示此命令。

第三步：断开与数据库之间的连接。利用ADO Connection Close.vi关闭Connection对象，并使用ADO Connection Destroy.vi删除Connection对象。



图3 向数据表添加记录的前面板



图4向数据表添加记录的程序框图

**4、利用LabVIEW SQL Toolkit访问数据库**

**4.1 LabVIEW SQL Toolkit简介**

LabVIEW SQL Toolkit(又被称作为LabVIEW Database Connectivity Toolset)是用于数据库访问的附加工具包，它集成了一系列的高级功能模块，这些模块封装了大多数的数据库操作和一些高级的数据库访问功能[10]。其主要的功能如下：

(1)支持ADO所支持的所有的数据库引擎；

(2)支持所有与ODBC或OLE DB兼容的数据库驱动程序；

(3)具有高度的可移植性，在任何情况下，用户通过改变DB Tools Open Connection VI的输入参数Connection String就可以更换数据库；

(4)可以将数据库中Column Values的数据类型转换为标准LabVIEW Database Connectivity Toolset的数据类型，这进一步增强了它的可移植性；

(5)与SQL兼容；

(6)不使用SQL语句就可以实现数据库记录的查询、添加、修改以及删除等操作；

(7)用户可以完全不需要学习SQL语法。

用户可以使用LabVIEW SQL Toolkit在LabVIEW中支持ODBC的本地或远程数据库，例如Microsoft Access，Microsoft SQL Sever，Sybase SQL Server以及Oracle等[11]。

**4.2应用实例**

本应用实例是显示上面所建staff表中的所有记录，所使用的数据源与上面两例相同。具体步骤如下：

(1)使用DB Tools Open Connection打开数据库连接[12]。

(2)显示指定数据表中所有数据。

使用DB Tools List Tables以一维数组形式列出指定数据库中的所有表，同时显示在前面板数组控件tables中。用户指定Table Index后，通过索引数组函数Index Array索引其中一个表。然后将表名连接到DB Tools List Columns和DB Tools Select All Data。

DB Tools List Columns的Columns参数用一维字符串数组返回指定表的字段名，通过属性节点将字段名显示在表格控件Table Data的首行，同时使用数组大小函数Array Size取得字段数，即表的列数，显示在Columns中。

DB Tools Select All Data的Data参数以二维字符串数组形式返回表中所有数据，显示在Table Data中，同时使用Array Size和Index Array函数取得记录数，也就是行数，显示在Rows中。

(3)使用DB Tools Close Connection断开连接。

此程序的前面板和程序框图如图5、图6所示。



图5 显示表中所有数据的前面板



图6 显示一个数据表中所有数据的程序框图

**5、结论**

综上所述，使用ADO，LabSQL或LabVIEW SQL Toolkit访问数据库的基本步骤都是相同的，都要经过连接数据库、执行操作、断开连接这三个步骤。

使用ADO主要优点是易于使用、高速度、低内存支出和占用磁盘空间较少,但需要用户对Microsoft ADO以及SQL语言有较深的了解。

使用LabSQL不用精通ActiveX技术和SQL语言，操作简介，容易理解，但它没有对ADO的所有功能进行封装，功能上有局限性。

LabVIEW SQL Toolkit支持ADO所支持的所有的数据库引擎，不使用SQL语句就可以实现数据库记录的查询、添加、修改以及删除等操作，使用SQL语句能够实现复杂的数据库操作，功能强大，但其昂贵的价格对于很多用户来讲是不能承受的，这也限制了它的推广应用。

在实际应用中，用户究竟采用哪种方法要从项目难易程度、自身编程水平和技术经济等方面综合考虑