

VC++ 中应用 ODBC 实现异构数据库的访问

宋玲 李陶深 陈乐 (广西大学计算机与信息工程学院 530004)

摘要:介绍在 Visual C++ 6.0 环境下,应用 ODBC 开发基于 MFC 数据库类进行异构数据库访问的编程方法及关键技术。

关键词:ODBC MFC Visual C++ 异构数据库

一、引言

随着计算机技术的迅猛发展,数据库的应用也越来越广泛。很多部门都在逐步建立自己的数据库系统,而数据库产品之多,为各用户在性能、价格和应用等方面提供了广泛选择的同时,也引入了一个棘手问题:由于这些数据库管理系统具有诸多互不兼容的特性,因而严重阻碍了企业内部以及企业间数据库资源的共享。同时,由于开发环境与工具的不断更新,用户在实施新的管理系统的同时,都毫无例外地面对旧系统的升级、新旧系统间数据的转换与相互调用以及单机用户数据上网等问题,而用户绝不愿意将原有的数据抛弃而重新进行大量原始信息的录入。再者,在传统的数据库连接方式下,开发人员必须熟悉多个 DBMS(数据库管理系统)及其 API(应用程序接口),一旦 DBMS 端出现变动,则往往导致用户端系统的重建或源程序的修改,给开发与维护工作带来了很大困难。

为此,Microsoft 于 1993 年推出了 ODBC(开放数据库互连),它为不同的数据库产品的访问和操作提供了一个统一的解决方案,实现了对异构数据库资源的透明使用。

二、ODBC 与 MFC

与传统的数据库应用程序的实现方法相比,ODBC 使用相同的应用程序代码适用于不同的 DBMS,实现应用程序和 DBMS 逻辑上的分离。典型的 ODBC 体系结构如图 1 所示。

ODBC 管理器安装在 Windows 的控制面板中,在 ODBC 运行机制中起配置环境、登记信息的作用。应用程序由程序员编制,可以在程序中调用 ODBC API 函数,提交 SQL 语句并返回结果数据集合,对数据集合进行处

理。API 是应用程序与 ODBC 的接口,定义了一个供应用程序调用的函数调用集。ODBC 驱动程序管理器根据应用程序的要求,调用不同的 ODBC 驱动程序。ODBC 驱动程序根据应用程序的 ODBC 函数调用,对指定的数据源执行相应的 SQL 语句,并把结果数据集返回给应用程序。数据源包括数据库文件和与之相关的 DBMS、操作系统、网络平台等。

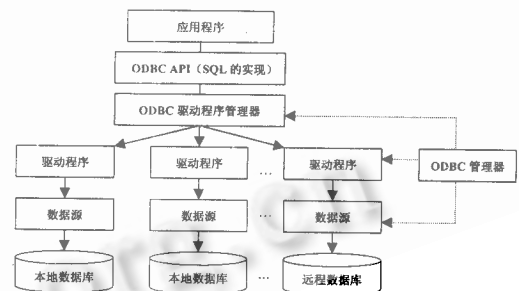


图 1 ODBC 的典型体系结构

在 Visual C++ 的开发环境下,有两种使用 ODBC 技术进行编程的方法。一种是直接调用 ODBC API 函数,实现对数据库的访问;另一种是利用 MFC(Microsoft 基本类库)中的数据库类进行数据库编程。直接调用 ODBC API 编程比较复杂,而 MFC 类库为访问 ODBC 数据源文件提供了一种简单直接的方法。Visual C++ 对 ODBC API 函数进行了封装并作为 MFC 的一部分,借助于 MFC 和 AppWizard,用户只要选择数据库就能够在生成应用程序的过程中,自动建立应用程序与数据库的连接关系。MFC 中主要包括三个数据库类 CDatabase(数据库类),CRecordSet(记录类),CRecordView(记录视图

类)。三个数据库类的关系如图2所示。

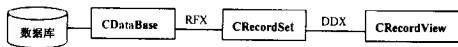


图2 数据库类之间相互作用关系图

CDataBase类的每个对象代表一个数据源的连接,用于对数据源的表格进行整体操作。CRecordSet类的每个对象代表了一个记录集合,每个CRecordSet类对象与一个CDataBase类对象相联系,并按SQL查询条件从数据源中取得记录集合。CRecordView类的每个对象都要与CRecordSet对象相联系,它是CRecordSet对象的外部表现形式,其作用是将记录集中的数据显示在用户界面上。应用程序通过CRecordSet的一个对象以RFX(记录字段交换)方式在其字段数据成员与数据源表的对应列间传送数据,通过DDX(对话框数据交换)方式把记录集当前记录中的字段送到格式控制中,并把更新过的信息回送到记录集中。

下面介绍在Visual C++ 6.0中如何利用MFC类库来实现异构数据库的管理和访问。

三、建立ODBC异构数据库 应用程序主框架

假设在系统中存在两个异构数据库,一个是用Access创建的学生信息库student.mdb,有id(学号)、name(姓名)、sex(性别)、birth(出生年月)等字段,另一个是用Visual Foxpro创建的学生成绩库soore.dbf,其字段为id(学号)、name(姓名)、chinese(语文)、math(数学)、english(英语)等。通过控制面板中的32位ODBC将这两个数据库登录到ODBC中,建立的数据源名分别为Access和VfpDsn。

1. 用AppWizard创建窗体应用程序

启动Visual C++ 6.0,在AppWizard中,创建一个基于SDI(单文档界面)或MDI(多文档界面)的项目(设项目名为odbctest)。其中,SDI应用程序在运行期间仅打开一个视图(View),在新的视图打开之前,当前的视图被关闭;而MDI可同时打开多个视图。然后是选择数据库支持的级别,可选择Database View With File Support或Database View Without File Support,前者在程序运行

中可通过File菜单的New选项在多个视图间切换,后者则不行,而且若选择后者,则必须选SDI。接下来选择数据源AccessDsn及其中的student.mdb作为第一个数据源,并选择记录集类型为Snapshot(快照),它使得对数据库的修改在该记录集被重新打开或查寻时才有效。完成后,主要产生出CRecordSet的派生类COdbctestSet和CRecordView的派生类COdbctestView,并在COdbctestView类中创建了一个COdbctestSet类的指针m_pSet。

为了在项目odbctest中加入异构数据源VfpDsn,可点击ClassView,然后右击odbctest classes,通过弹出菜单的New Class项派生一个基类为CRecordSet的MFC类,取名为CVfpSet。接下来可进行记录视图的设计了。

2. 记录视图的设计

(1)一视图对应多个数据源数据方式。实现应用中,有时需要将不同数据源的数据显示在同一视图上,具体方法是:

①在OdbctestDoc.h文件中加入头文件:#include "VfpSet.h"并输入生成CVfpSet类对象public: CVfpSet m_scoresset;②在OdbctestView.h文件中加入CVfpSet类的指针: CVfpSet * m_pSet2;③在OdbctestView.cpp文件的COdbctestView()中设置:m_pSet2 = NULL;并在OnInitialUpdate()初始化函数中加入代码:m_pSet2 = &GetDocument() -> m_scoresset; m_pSet2 -> Open();④连接第一个数据源后,产生一个对话框表格(IDD_ODBCTEST_FORM),置于ResourceView的Dialog中,双击它可进行记录视图的设计。在ClassWizard中我们会发现,只能将控件与第一个指针(m_pSet)所指的成员变量绑定,要实现在IDD_ODBCTEST_FORM中将控件与m_pSet2所指向的记录集类CVfpSet相连,则应在OdbctestView.cpp的DoDataExchange函数中手工添加一些代码。例如要将IDC_EDIT5控件与CVfpSet的math字段数据成员相连,添加的代码为(其余字段类推):

```
DDX_FieldText(pDX, IDC_EDIT5, m_pSet2 -> m_math, m_pSet2);
```

至此,两个不同类型的数据源都加入了应用程序,可以在COdbctestView中通过m_pSet指针和m_pSet2指针分别控制两个数据集对象,实现对两个异构数据库数据的操作了。

(2)视图与数据源一一对应方式。要采用此方式来访问各数据源数据,则在用AppWizard创建项目时,应选

择 MDI 或 SDI 方式和 Database View With File Support 数据库支持, 这样, 在开始运行时, 屏幕上出现由视图名称组成的菜单供选择, 如图 3 所示, 菜单中的视图名称可通过修改 ResourceView 中的 String Table 来使其显示界面更友好。并且在运行过程中可通过 File(文件) 菜单的 New(新建) 选项来随时选择和切换视图。当采用 MDI 方式时, 由于屏幕上同时保存多个视窗, 因此也可用鼠标点击窗口来进行切换。

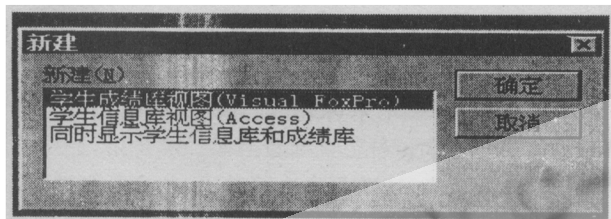


图 3 多种视图选择

添加一个视图的实现方法是: 在 ClassView 中右击 Odbctest Classes 后选择 New Form, 建立一个基类为 CRecordView 的对话框表格; 或选 New Class, 建立一个基类为 CRecordView 的 Form 类, 接下来是为建立的数据视图指定与其相连的记录集类(本例中是选 COdbctestSet 或 CVfpSet)。

四、异构数据库访问操作的 ODBC 编程

1. 同步移动多个数据库的记录

在一个视图对应多个数据源数据这种显示方式中, 特别要求当一个数据库(如: student.mdb)的记录前后移动时, 另一数据库(如: score.dbf)的记录也要根据同一关键字(id)进行同步移动, 程序实现为:

(1) 在 OdbctestView.cpp 的 OnInitialUpdate() 初始化函数中添加以下代码:

```
m_pSet2->Close(); m_pSet2->m_strFilter="
id=" + m_pSet->m_id + "'"; //程序开始执行时记录就同步
```

```
m_pSet2->Open(); UpdateData(FALSE);
```

(2) 在 OdbctestView.cpp 中重载虚函数 OnMove(), 代码如下:

```
BOOL COdbctestView:: OnMove ( UINT nID-
MoveCommand)
```

```
{ BOOL T= CRecordView:: OnMove(nIDMoveCom-
mand); //继承父类, 否则不能同步
m_pSet2->Close(); m_pSet2->m_strFilter="
id=" + m_pSet->m_id + "'"; //按 id 同步移动记录
m_pSet2->Open(); int rcount = m_pSet2->Ge-
tRecordCount();
if(rcount == 0) { MessageBox("成绩库中没有匹配
的记录!");
m_pSet2->Close(); m_pSet2->m_strFilter
=""; m_pSet2->Open();
}
UpdateData(FALSE); //将结果显示在屏幕上
return t;
}
```

2. 增加、删除和修改记录

增加、删除和修改记录分别用到 CRecordSet 的成员函数 AddNew(), delete() 和 Edit(), 此时应注意同时更新两个数据源的数据。例如在采用一视图对应多数据源数据方式中, 为增加一个记录所编写的函数如下(在 OdbctestView.cpp 中):

```
void COdbctestView:: OnRecordAdd() //增加一个记
录
{ m_pSet->AddNew(); m_pSet2->AddNew();
UpdateData(FALSE); }
void COdbctestView:: OnChangok(- //保存所做的修
改
{ UpdateData(TRUE); //将数据回写入 m_pset 所
指的数据库
CString id = m_pSet->m_id, name = m_pSet->
m_name;
if(m_pSet->Canupdate()) m_pSet->Update
(); //更新记录
m_pSet->m_id = id; m_pSet->m_name =
name;
//填写当前记录中第二个库与第一库相同的字段
m_pSet2->m_id = m_pSet->m_id; m_pSet2
->m_name = m_pSet->m_name;
if(m_pSet2->CanUpdate()) m_pSet2->Update
();
m_pSet->Requery(); m_pSet2->Requery();
```

//重新生成记录集

```
UpdateData(FALSE);
}
void COdbctestView::OnChangecancel() //放弃所做的修改
{
    m_pSet -> CancelUpdate(); m_pSet2 -> CancelUpdate();
    m_pSet -> MoveFirst(); m_pSet2 -> m_strFilter = "id='"+ m_pSet -> m_id + "'";
    UpdateData(FALSE);
}
```

3. 查询、浏览记录

在一视图对应多数据源数据方式中,对异构数据库记录的整体查询可以采用这样的实现方法:首先将用户输入的查询条件转化为对多个数据库的局部查询条件,分别进行查询后,得到多个记录集,然后按关键字求出这多个记录集的交集。例如在项目 Odbctest 中,先将查询条件按数据库分成两部分,分别查询 student 和 score 这两个库后得到两个记录集,将在两个记录集中均出现的 id 字段值记下,即求出了交集,接着就可以根据 id 值来显示各个记录了。

在实际应用中,除了要求每次显示一个记录外,还需要一次能浏览全部或部分记录。可用 CListBox 类的 AddString() 函数将各记录的字段值在列表框中逐一列出,若是浏览多个数据库,则按第一个库的关键字过滤出其他库的对应记录,即可以回到单记录操作视图。做法是:取出所点击记录的关键字值,用作成员变量 m_strFilter 的值进行过滤,再用 UpdateData(FALSE) 把记录在对话框中显示出来。

五、使用 ActiveX 控件访问数据库

在 VC++ 6.0 中使用 Microsoft ActiveX 控件 ADO Data Control 和 DataGrid Control 相结合,可以很方便、快速地操作数据库。具体主要分以下几个步骤来实现:

1. 在项目中添加 ActiveX 控件

点击 Project,从菜单中选择 Add to Project 然后选择 Components And Controls。选定 Registered ActiveX Controls,这可获得目前注册在系统中的所有 ActiveX 控件,

从中选择 Microsoft ADO Data Control 和 Microsoft DataGrid Control 这两个控件添加到项目中。

2. 使 ADO Data Control 指向一个 OLE DB 数据源

在一个对话框中右击鼠标,选择 Insert ActiveX Control,选取控件 Microsoft ADO Data Control 添加到该对话框中,然后右击该控件,选 Properties 项,在 Control 标签下选 Use Connection String,按 Build 按钮,选择 ODBC 驱动程序提供者后进入 DataLink Properties 框,返回 ADO Data Control 属性框,选择 RecordSource 标签,在 Command Type 下拉列表中选 2 - adcmdTable,并选择数据源中的一个表(如 student)。

3. 在对话框中添加 DataGrid control 控件并与 ADO Data Control 相连

DataGrid control 是一个数据绑定控件,将该控件添加到对话框并右击它,选择 Properties,选 All 标签,在 DataSource 一栏的下拉列表中选择 ADO Data Control 的 ID 名,默认值为 IDC_ADODCI。

若要在同一视图中访问其他数据库,需重复以上步骤。这样,就将数据库的表与 DataGridcontrol 控件动态绑定起来,在运行时,通过该控件可以方便、直接地查看或修改表中的记录,实现所选的表与控件间的双向数据交换。

六、结束语

当前,ODBC 技术正被广泛地应用于数据库应用程序的开发工作中,研究采用 ODBC 技术进行异构数据库访问和操作的相关技术,在实现中具有重要的意义。本文结合具体的实践,介绍了在 VC++ 6.0 环境下应用 MFC 数据库类实现异构数据库访问和操作的基本方法,具有一定的参考价值和实用价值。

参考文献

- [1] David J. Druglinski 等, Visual C++ 6.0 技术内幕,北京希望电子出版社
- [2] 宛延, C++ 语言和面向对象程序设计,北京:清华大学出版社,1997.12

(来稿时间:1999年8月)