

在 Visual C#. Net 下利用 DataGrid 控件

实现主 - 从式查询^①

Master - detail query by DataGrid Control in Visual C#. Net

韩毅 (长春光机物理所、中科院研究生院 130024)
(东北师范大学计算机学院 130024)

陈建 (北京联合大学 生物化学工程学院 100081)

张靖波 肖永鹏 (长春 东北师范大学 计算机学院 130024)

摘要:本文给出一种在 C#中利用单一 DataGrid 控件实现主 - 从式查询的方法。采用该方法生成的主 - 从式数据网格在根据主表信息查询明细时具有视觉直观、导航灵活的特点。

关键词:数据库 C# DataGrid 控件 主 - 从式查询

1 引言

在数据库应用中经常会遇到主 - 从式查询的情况,即两个数据表是属于一对多的关系,若表 A 中的每行可以在表 B 中有若干行对应,但是表 B 中的每行只能在表 A 中有一个对应行,则称表 A 为主表,而称表 B 为从表。在查询时往往需要根据表 A 中的某一行去查找表 B 中对应的多个行,这种查询就是主 - 从式查询。例如在 SQL Server 2000 自带的示例数据库 Northwind 中,orders 表和 order details 表就是一对多的关系:orders 表中的每一个订单包含各自的信息,如订单的客户号、订单号、订单日期等;而针对某订单的具体订货信息(如商品号、数量等)则对应到 order details 表中的多行数据,即 order details 表中多个订货商品可能同属于 orders 表中的某一个订单。在实际应用中,我们可能会根据查询出的订单信息来查询该订单下所有货品的详细信息。

2 常用的主 - 从式查询的实现方法

以往实现主 - 从式查询的方法有很多,如主表的关键字段(主键)以组合列表框或列表框的形式列出,

从表数据则用 DataGrid 控件列出。当在列表框中选择主表中某一行时将表中与之关联的那些行列出到 DataGrid 控件。而在 Visual Studio . Net 的 MSDN 文档中给出的方法是使用两个 DataGrid 控件以主控与明细格式显示数据。一个 DataGrid 被指定为主网格,而另一个被指定为详细信息网格。当在主列表中选择某项时,所有相关的子项都显示在详细信息列表中。上述方法虽然实现简单,但实际效果却不是很好,经常有用户反映这些方式浏览数据不直观,前者看不到主表中的详细信息,后者则不易将两个 DataGrid 中的数据一一对应起来,给人的感觉象是两个无关的数据表。


3 利用单一 DataGrid 控件实现

主 - 从式查询

经过多次尝试与实践,笔者摸索出一种简单可行的方法,利用单一 DataGrid 控件,可以直观地表示出主表中的某一行与从表中多行数据的对应关系,且完全不存在主表数据翻页时从表数据不同步的问题。图 1 为主表中的数据信息。

在图 1 中 DataGrid 控件显示父表中各行数据的信

^① 本文受东北师范大学自然科学基金资助(项目编号:111494023)

息。父表和子表之间的关系使用具有导航功能的父子结构进行显示。在每行的标头位置都有一个展开器,单击展开器会生成指向子表的、类似于 Web 链接的列表(根据实际应用的情况,可以设置多个这样的链接。这些链接实际是程序中设定的表间关联的名称。),这些链接一经选择就会显示子表中相关的数据,并且在 DataGrid 控件的题头处会显示当前父表中与子表相关的那行数据(如图 2)。单击“显示/隐藏父行”图标()将隐藏有关父表的信息(如果以前已经隐藏,则会使之重新显示)。单击后退按钮则返回到浏览父表。



订单号	雇员号	客户号	订单日期
10248	5	VINET	1996-7-4
10249	6	TOMSP	1996-7-5
10250	4	HANAR	1996-7-8
10251	3	VICTE	1996-7-8
10252	4	SUPRD	1996-7-9
10253	3	HANAR	1996-7-10
10254	5	CHOPS	1996-7-11
10255	9	RICSU	1996-7-12

图 1 主表数据信息



订单号	货品号	单价	数量	折扣
10250	41	7.7	10	0
10250	51	42.4	35	0.15
10250	65	16.8	15	0.15
*				

图 2 从表数据信息(标题栏上方是主表对应信息)

现在我们就以 Northwind 数据库为例,将具体实现方法介绍如下:

3.1 窗体布局

为使示例程序简单起见,在本程序中窗体上仅放置一个 DataGrid 控件,除将其大小设置为充满窗体外(Dock = Fill),其余属性均按照默认值设置。而对 Dat-

aGrid 控件的初始化工作则在 Form1_Load 事件处理过程中完成。

3.2 使用 ADO.NET 连接到 SQL Server 数据源

在 Visual Studio.net 中要连接数据库,必须要使用 SqlConnection 对象。该对象属于 System.Data.SqlClient 命名空间,故在程序开始处应加上 using System.Data.SqlClient; 语句。

首先声明连接字符串 conn,并将其初始化为 @"uid = sa;pwd = ;server = localhost;database = Northwind"。(这里可能读者的数据库服务器名、密码与笔者不同,请自行更改。)

然后以上述连接字符串实例化一个 SqlConnection 对象 myConn,则 myConn 为指向 Northwind 数据库的一个连接。通过调用 myConn 对象的 Open 方法将 ADO.NET 连接到 Northwind 数据库。

3.3 使用数据适配器将数据填充到数据集

数据适配器(SqlDataAdapter)是 ADO.NET 托管提供程序(用于在数据源和数据集之间通信的一组对象)的组成部分,用于在数据源和数据集(DataSet)之间交换数据。实例化数据适配器和数据集后可以通过调用数据适配器的 Fill 方法将某个 SQL 语句或存储过程执行的结果填充到数据集。

在本例中,首先声明一个 DataSet 对象 ds,然后分别将 orders 表和 order Details 表中的数据通过数据适配器填充到数据集 ds。由于篇幅所限,这里仅选取了 orders 表中的部分列和 order Details 表中的所有列。相应代码如下:

```
DataSet ds = new DataSet();
```

```
//从 myConn 指定的数据源中执行 SQL 语句,从 orders 表中选择数据(部分列)
```

```
SqlDataAdapter myAdapter = new SqlDataAdapter("select orderID as 订单号, EmployeeID as 雇员号, CustomerID as 客户号, orderDate as 订单日期 from orders", myConn);
```

```
//将 SQL 语句的结果数据填充到数据集中的 or-
```

ders 表

```
myAdapter.Fill(ds,"orders");
```

//从 myConn 指定的数据源中执行 SQL 语句,从 order Details 表中选择数据(所有列)

```
SqlDataAdapter myAdapter1 = new SqlDataAdapter("select orderID as 订单号,ProductID as 货品号,Unit-Price as 单价,Quantity as 数量,Discount as 折扣 from [order Details]",myConn);
```

//将 SQL 语句的结果数据填充到数据集中的 order Details 表

```
myAdapter1.Fill(ds,"order Details");
```

填充好数据后就可以将数据库的连接断开了,即调用 myConn 的 Close 方法。

3.4 定义数据集中主表的主键

因 DataGrid 最初是要显示主表的内容,故应将其 DataSource 属性设置为本例中的主表——orders。

然后将主表中的主键指定为 orderID 列:

```
ds.Tables["orders"].PrimaryKey = new DataColumn[] { ds.Tables["orders"].Columns["orderID"]};
```

3.5 定义数据集中主表与从表的关联

如果数据集中有多个表,这些表中的信息可能是相关的。数据集没有这些关系的继承知识,因此为了使用相关表中的数据,可以创建 DataRelation 对象来描述数据集中表之间的关系。DataRelation 对象一经建立,就会自动跟踪表之间的关系并使父记录和子记录保持同步,也可以使与当前记录相关的记录可用。如果在父记录中,它提供子记录;如果正使用子记录,它提供父记录。

在本例中,将两个表通过 orderID 列建立关联,并将该关联命名为“单击查看明细”。关联可以是多个,其各自的名称会在 DataGrid 中以树的形式显示出来。

定义主表与从表的关联的代码如下:

```
DataRelation relationOrderID;
```

```
relationOrderID = ds.Relations.Add("单击查看明细",ds.Tables["orders"].Columns["订单号"],ds.Tables["order Details"].Columns["订单号"]);//注意此处列名应为 ds 中表的列名(中文)而非数据库中的表的列名
```

定义了关联后刷新 DataGrid 控件,任何时刻只要单击主表行首的展开器,就会出现指向子表的链接,单

击链接就会自动导航到子表,为用户进行主-从式查询提供了极大的方便和一目了然的视觉效果。

上述各对象的属性也可以在设计阶段直接指定。

本程序在 Windows 2000,SQL Server 2000 环境下调试通过。

本文相关源代码如下:(略去部分无关代码)

```
using System;
```

```
using System.Drawing;
```

```
using System.Collections;
```

```
using System.ComponentModel;
```

```
using System.Windows.Forms;
```

```
using System.Data;
```

```
using System.Data.SqlClient;
```

```
namespace DetailGrid
```

```
{
```

```
public class Form1 : System.Windows.Forms.Form
```

```
{
```

```
... ..
```

```
private void Form1_Load(object sender, System.
```

```
EventArgs e)
```

```
{
```

```
//初始化连接字符串。读者的数据库服务器、密码可能会不同
```

```
string conn = @"uid=sa;pwd=;server=localhost;database=Northwind";
```

```
//按指定的连接字符串打开数据库连接
```

```
SqlConnection myConn = new SqlConnection(conn);
```

```
myConn.Open();
```

```
//声明数据集 ds
```

```
DataSet ds = new DataSet();
```

```
//从 myConn 指定的数据源中执行 SQL 语句,从 orders 表中选择数据(部分列)
```

```
SqlDataAdapter myAdapter = new SqlDataAdapter("select orderID as 订单号,EmployeeID as 雇员号,CustomerID as 客户号,orderDate as 订单日期 from orders",myConn);
```

```
//将 SQL 语句的结果数据填充到数据集中的 orders 表
```

```
myAdapter.Fill(ds,"orders");
```

```
//从 myConn 指定的数据源中执行 SQL 语句,
从 order Details 表中选择数据(所有列)
SqlDataAdapter myAdapter1 = new SqlDataAdapter(
    "select orderID as 订单号,ProductID as 货
    品号,UnitPrice as 单价,Quantity as 数量,Dis-
    count as 折扣 from [ order Details ]", my-
    Conn);
//将 SQL 语句的结果数据填充到数据集中的
order Details 表
myAdapter1.Fill(ds,"order Details");
//释放数据库连接
myConn.Close();
//设置 dataGrid1 对象的数据源
dataGrid1.DataSource = ds.Tables["orders"];
//为 ds 中的 orders 表设置主建
ds.Tables["orders"].PrimaryKey = new Data-
Column[] { ds.Tables["orders"].Columns["
订单号"] };
DataRelation relationOrderID;
//为 orders 表和 orders details 表建立关联
relationOrderID = ds.Relations.Add("单击查看明
细",ds.Tables["orders"].Columns["订单号"],ds.
Tables["order Details"].Columns["订单号"]);//注
意此处列名应为 ds 中表的列名(中文)而非数据库中
```

```
的表的列名
//更新 dataGrid1 的显示
dataGrid1.Refresh();
}
}
}
```

4 结束语

主-从式查询的实际应用特别广泛,实现的方式也多种多样。如何选择一种合适的方式来实现这种功能,直接关系到用户对软件的认可程度和软件应用性的好坏。本文介绍的方法不仅实现起来特别简单,而且具有界面友好、视觉直观、导航灵活的特点。经过多个 MIS 项目的验证,用户一致反映较好。本文介绍的方法只需做略微的改动就可以同样适用于 Visual Basic. Net。

参考文献

- 1 温谦、郑小平、刘涛, C#语言程序设计[M], 人民邮电出版社, 2001. 7。
- 2 余安萍等编, C# 程序设计教程[M], 电子工业出版社, 2002. 8。
- 3 Petkovic. SQL Server7 循序渐进教程[M], 希望电子出版社, 1999. 6。