

脚本之家

www.jb51.net

买了！买一送一千万购物卡

更有1折秒杀，100%中奖惊喜！

12核 / 16G / 2T

百兆独享 · 猛戳我

网页制作

网络编程

脚本专栏

脚本下载

数据库

CMS教程

电子书籍

平面设计

媒体动画

操作系统

网站运营

网络安全

在线手册

基础应用

实用技巧

自学过程

您的位置： 首页 → 网络编程 → ASP.NET → 自学过程 → 正文内容 ASP.NET 2.0 处理Computed Columns列

请输入关键词

搜索

低价出售流量10000IP只需8元

阿里云免费虚拟机+日志分析仪6元

16核16G独服百兆带宽899元/月|创梦网络

美国云20元/香港云29元/韩国云55元

装软件赚钱，每台电脑秒赚6.3元

百兆 12核24线程 16G内存 2T 999/月

微子网络 湛江双线高防 抗15G 700元起

中原地区最大IDC数据中心，首月托管免费

众生网络 基于云计算的互联网基础服务运

服务器租用/托管-域名空间/认准腾佑科技

BGP四线 亿恩1U服务器托管3999元/年

九九数据 — 工信部认可正规资质IDC接入商

枫信科技-江苏双线10M保证-399/元

港湾云主机，稳定、安全才是硬道理

免备vps20/百独799/双线350/45互联

韩国香港美国站群服务器 巨牛网络

鼎点网络百兆独享服务器仅需999元

[香港双高防]无视CC★DDOS/堪比广东！

畅游网络 百独服务器 包跑满 998元

服务器租用 199元起

8核服务器298元/月 45/月云主机

群英云服务器送10M带宽30G防御,49元起

免费空间

【1000UV流量只要8毛】还有点点击流量

西部数码

中国云主机领导品牌

云主机

仅需48元

DDoS高防IP 300G无限防

453 Gbps真实防御经检测，支持云外

高防云

158月/月起！

320G 立体防御

免备案云主机

阿里云

30+云计算产品 免费6个月

云计算应用更轻松

免费上云

在ASP.NET 2.0中操作数据之六十九：处理Computed Columns列

作者：heker2007 字体：[增加 减小] 类型：转载 时间：2016-05-19 我要评论

SQL Server创建表的时候允许我们创建computed columns列，这种列的值源与表达式，并且是只读的，所以我们不能通过Insert和Update为其赋值，本文具体讲解创建和调用显示的方法和步骤。

Ps

学习ps基础

4567890

新手学电脑

ASP.NET 2.0

自学网络编程

it培训机构排名

学电脑先学什么

oa系统

酒店管理系统

ps自学网

微型摄像机

u盘定制

液晶拼接屏

预付费水表

导言：

Microsoft SQL Server里有一种computed columns列.这种列的值是通过一个表达式来计算，而表达式引用的是同一张表的其它列的值.打个比方，有一张ServiceLog表，其包含了ServicePerformed, EmployeeID, Rate, Duration等列.虽然我们可以在一个web页面或其它什么界面里计算每笔服务的费用(也就是 比率 rate乘以时间段duration)，不过我们也可以手动向ServiceLog表添加一个 AmountDue列以反映该信息.我们可以将该列创建为一个普通列，只是任何时候当Rate 或 Duration列的值发生改变时需要更新AmountDue列的值.一个比较好的办法是将AmountDue创建成一个computed column 列，其使用的表达式为 Rate * Duration. 这样，当在一个查询里引用该列时SQL Server就可以自动的计算AmountDue列的值。

由于computed column列的值是由表达式决定的，所以这种列是只读的，并且不能在INSERT 或 UPDATE statements里对其赋值.然而，对使用ad-hoc SQL statements的TableAdapter来说，如果主查询里引用了computed column列，那么自动生成的INSERT 和 UPDATE statements也会自动的引用computed column列.所以，我们必须更新TableAdapter的 INSERT 和 UPDATE 查询，以及InsertCommand 和 UpdateCommand属性，以删除对任何computed column列的引用。

如果在使用 ad-hoc SQL statements的TableAdapter里使用computed columns的话，我们要面临的挑战之一便是，每当完成TableAdapte设置向导时，TableAdapter的 INSERT 和 UPDATE查询都会自动的生成，又一次的自动引用computed column列.不过如果TableAdapters使用存储过程的话，就不会出现这个问题。

在本文，我们将向Northwind数据库的Suppliers表添加一个computed column列，然后相应地创建一个TableAdapter来处理该表以及该computed column列.我们将在TableAdapter里使用存储过程而不是ad-hoc SQL statements.

第一步：向Suppliers表添加一个Computed Column

大家感兴趣的内容

1 .dll 文件反编译的工具软件集合

2 找不到类型或命名空间名称“Serv

3 AlternatingItemTemplate类似于

4 .NET 2.0获取配置文件AppSetting

5 无法在Web服务器上启动调试。未将

6 asp,asp.net学习教程下载

7 决定何时使用 DataGrid、DataLis

8 c#中过滤html的正则表达式

9 ExecuteReader(),ExecuteNonQuer

10 如何在DataGrid控件中实现自定义

在本文，我们将向Suppliers表添加一个名为FullContactName的computed column列，它以“ContactName (ContactTitle, CompanyName)”的格式返回contact的name, title，以及所在的公司。

打开服务器资源管理器，在Suppliers表上单击右键，选“Open Table Definition”，这将会显示出表所包含的列以及列的属性，比如数据类型、是否允许为NULL值等等。要添加一个computed column列，只需在表定义里键入表的名称，接下来在Column属性窗口的Computed Column Specification部分的(Formula)文本框里输入表达式（如图1所示）。将该computed column列命名为FullContactName，并使用下面的表达式：

```
1 ContactName + ' (' + CASE WHEN ContactTitle IS NOT NULL THEN
2 ContactTitle + ', ' ELSE ' ' END + CompanyName + ')'
```

请注意，在SQL里可以用操作符“+”来连接字符串。而CASE声明类似于传统编程语言里的条件语句。上面代码里的CASE 声明可以这样来理解：如果ContactTitle 不为NULL，那么输出ContactTitle值，再紧接一个逗号；如果为NULL，则无操作。关于CASE 声明的更多信息请参阅文章《The Power of SQL CASE Statements》（<http://www.4guysfromrolla.com/webtech/102704-1.shtml>）

注意：除了CASE声明外，我们还可以使用ISNULL(ContactTitle, "")。语法ISNULL(checkExpression, replacementValue) returns是这样工作的，如果checkExpression 不为NULL,则对其进行返回；如果为NULL则返回replacementValue。虽然本文这2种语法都可以使用，但是在一些稍微复杂点的情况下，使用ISNULL的情况要多一些。添加完computed column列后，你的屏幕看起来应该和图1差不多：



图1：向Suppliers表添加一个名为FullContactName的Computed Column列

添加完后点工具栏上的Save图标，或按Ctrl+S键,又或者在File菜单里选“保存Suppliers”。“保存”操作会自动地刷新服务器资源管理器，将刚刚添加的列展现在Suppliers表里。此外，键入到(Formula)文本框的表达式会自动的进行调整，剔除不必要的空白，将列名用[]括起来，并使用圆括号()来显示操作的先后顺序：

```
1 (((([ContactName]+' (')+case when [ContactTitle] IS NOT NULL
2 then [ContactTitle]+', ' else ' ' end)+[CompanyName])+')')
```

关于Microsoft SQL Server里computed columns列的更多信息请参考文章《technical documentation》（<http://msdn2.microsoft.com/en-us/library/ms191250.aspx>）；同时你也可以参考文章《How to: Specify Computed Columns》（<http://msdn2.microsoft.com/en-us/library/ms188300.aspx>），看如何一步步地创建computed columns列。

注意：默认情况下，数据库表并没有“实际”(physically)的包含computed columns列，而是每次在一个查询里引用它时重新计算其值。不过，我们可以选择“Is Persisted”选项来让SQL Server实实在在的在数据库表里创建computed columns列。这样的话我们可以为computed column列创建一个索引，当在一个查询的WHERE字句里使用computed column列的值时就可以提高执行效率。更多的信息请参阅文章《Creating Indexes on Computed Columns》（<http://msdn2.microsoft.com/en-us/library/ms189292.aspx>）

第二步：查看Computed Column列的值

在处理数据访问层前，让我们花点时间查看FullContactName列的值。在服务器资源管理器里，在Suppliers表上右键单击，选择“New Query”，这将启动一个查询窗口提示我们在查询里包含哪个表。添加Suppliers表，再点“Close”。接下来从Suppliers表里选择CompanyName, ContactName, ContactTitle,以及FullContactName列。最后，点击工具栏上的红色感叹号图标执行查询，查看结果。如图2所示，结果里包含了FullContactName列，它以ContactName (ContactTitle, CompanyName)”的格式使用了CompanyName, ContactName, ContactTitle这3列。

查看标识获取更多信息

图片相关推广



最近更新的内容

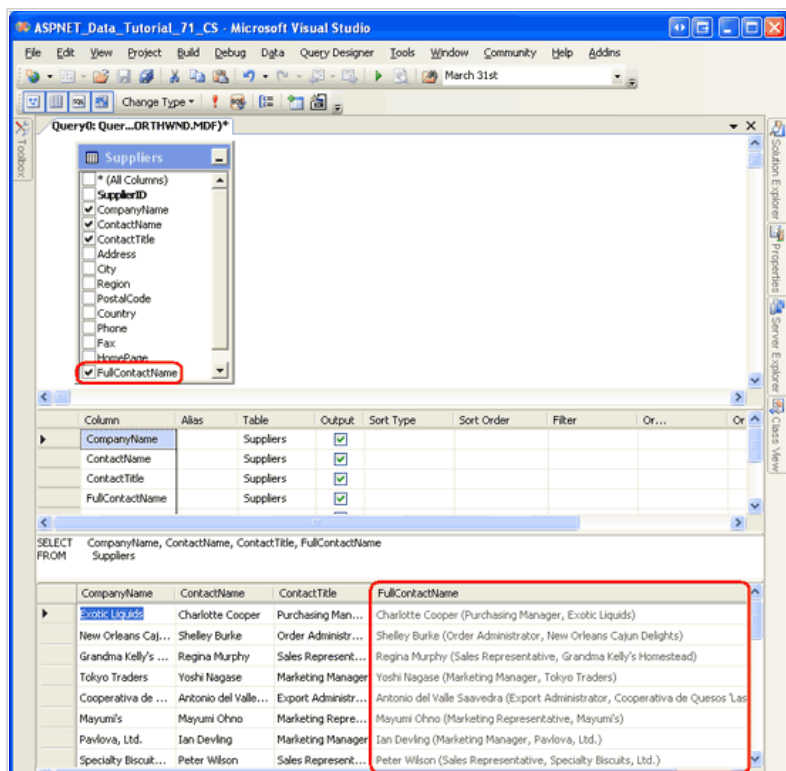
如何用Response.Redirect方法传递汉字
解读ASP.NET 5 & MVC6系列教程（4）：
细说SQL Server中的视图
在ASP.NET 2.0中操作数据之五十九：使用
在ASP.NET 2.0中操作数据之三十四：基于
在ASP.NET 2.0中操作数据之五十五：编辑
《解剖PetShop》之二：PetShop数据访问
解读ASP.NET 5 & MVC6系列教程（13）
ExecuteReader(),ExecuteNonQuery(),ExecuteScalar()
在ASP.NET 2.0中操作数据之十四：使用Fc

众生网络 品牌服务器租用	集思网络 VPS主机	枫信科技 IDC服务商
-----------------	---------------	----------------

常用在线小工具

在线JSON代码检验/检验/美化/格式化
JavaScript代码在线加密工具
RGB颜色查询对照表_配色工具
CSS代码工具
JavaScript压缩/格式化/加密工具
在线XML/JSON互相转换工具
php代码在线格式化美化工具
在线生成二维码工具(加强版)
Unix时间戳(timestamp)转换工具
在线JS脚本校验器错误





搜索
推荐

windows8系统下载 百度一下

图2: FullContactName列的格式为"ContactName (ContactTitle, CompanyName)"

第三步: 在数据访问层添加一个SuppliersTableAdapter

为了在我们的应用程序里处理supplier信息, 我们首先需要在DAL层创建一个TableAdapter 和 DataTable. 我们可以用前面的教程探讨的方法来进行创建, 稍微不同的是我们将要与computed columns列打交道.

如果你用ad-hoc SQL statements来构造一个TableAdapter的话, 你可以很简单的通过TableAdapter设置向导在TableAdapter的主查询里引用computed column列, 这样, 在自动生成的INSERT 和 UPDATE statements就会引用computed column列. 如果你执行这2个方法的话, 将会抛出这样的一个SqlException: "The column 'ColumnName' cannot be modified because it is either a computed column or is the result of a UNION operator".虽然我们可以在InsertCommand和UpdateCommand属性里手工改动INSERT 和 UPDATE statements, 但是一旦重新运行TableAdapter设置向导后, 我们所做的用户定制就会丢失掉.

查看标识获取更多信息

图片相关推广

在TableAdapters中使用现有存储过程的话, 很重要的一点是最开始, 你不要在主查询里引用computed columns列, 如果你在主查询里引用了computed columns列的话, 你刚完成设置, 向导就会提示你不能创建相应的存储过程.简而言之, 在设置TableAdapter时, 最开始不要在主查询里引用computed column列, 接下来再对相应的存储过程和TableAdapter的SelectCommand属性进行更改以引用computed column列.这种方法我们在第67章《在TableAdapters中使用JOINS》里探讨过.

本文我们将新添加一个TableAdapter并自动创建存储过程.当然我们要在主查询里忽略这个名为FullContactName的computed column列.打开~/App_Code/DAL文件夹里的NorthwindWithSprocs DataSet数据集, 在设计器里右键单击, 选"add a new TableAdapter", 这将开启TableAdapter设置向导, 指定数据库连接信息 (也就Web.config文件里的NORTHWNDConnectionString), 点Next. 选"Create new stored procedures"项, 再点Next.

关闭

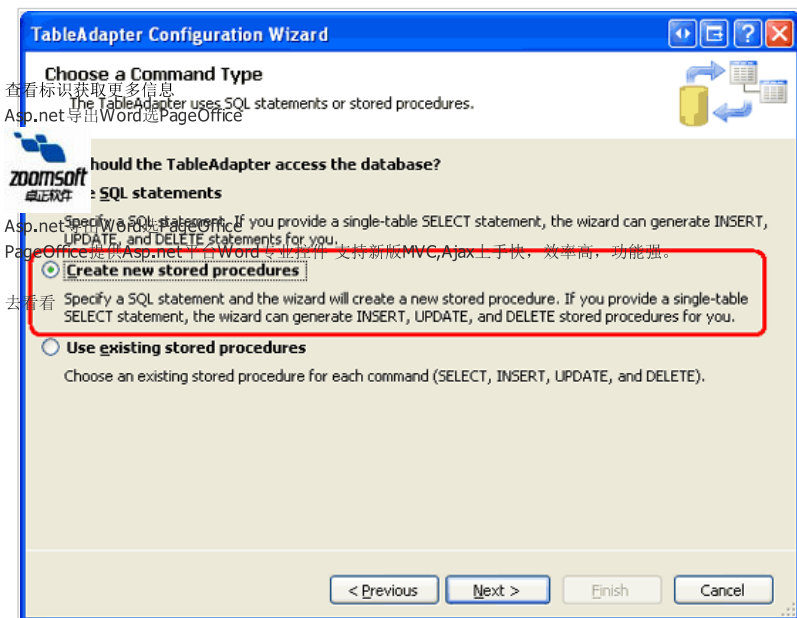


图3：选择“Create new stored procedures”项

接下来我们要指定主查询，键入如下的查询，其返回每个supplier的SupplierID, CompanyName, Contact

sql computed columns asp.net自学视屏

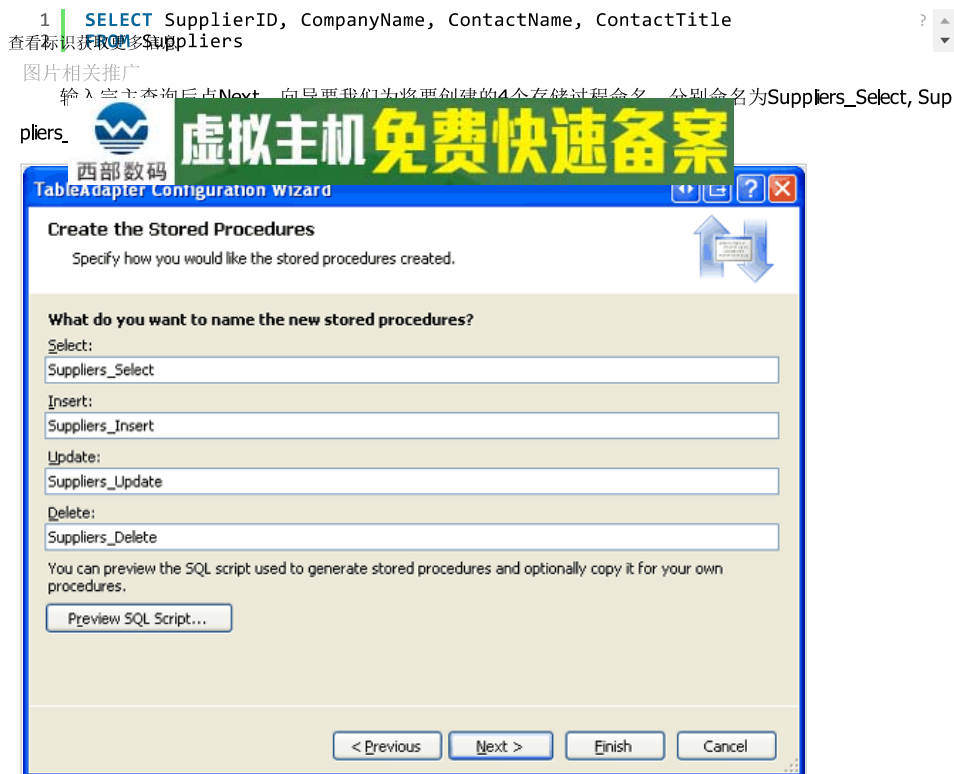


图4：对自动生成的存储过程命名

接下来我们要为TableAdapter的方法命名并指定用于访问和更新数据的模式.我们全部选中这3项，不过将GetData方法重命名为GetSuppliers.点击Finish完成配置。

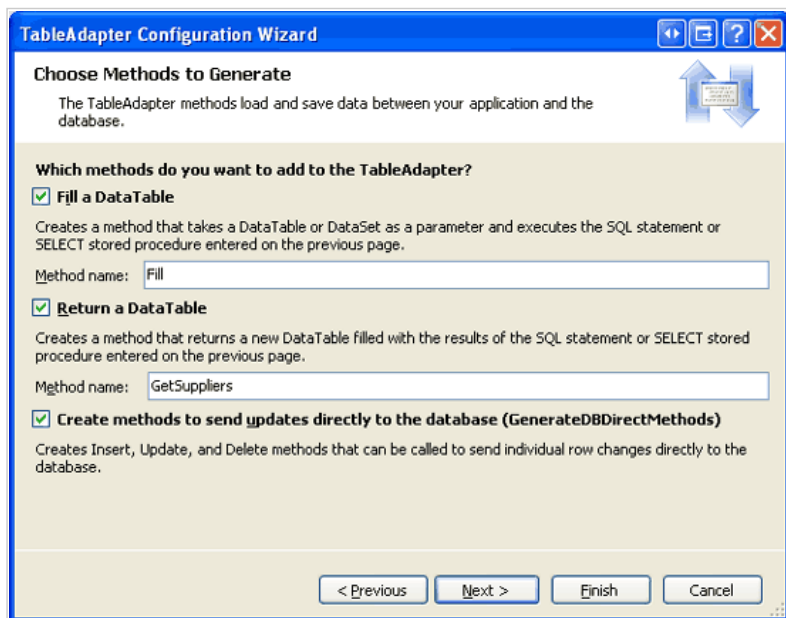


图5：将GetData方法重命名为GetSuppliers

完成后向导将创建这4个存储过程，并向类型化的DataSet添加ableAdapter以及对应的DataTable。

第四步：在TableAdapter的主查询里引用Computed Column列

接下来我们将对第三步创建的TableAdapter 和 DataTable进行更新以引用FullContactName列，这要经过2个步骤：

- 1.更新名为Suppliers_Select的存储过程以返回FullContactName列
- 2.更新DataTable以包含相应的FullContactName列

首先在服务器资源管理器里打开存储过程文件夹，打开Suppliers_Select存储过程，更新其SELECT查询以引用FullContactName列：

```

1  SELECT SupplierID, CompanyName, ContactName, ContactTitle, FullContactName
2  FROM Suppliers

```

保存所做的修改.接下来返回到DataSet Designer，在SuppliersTableAdapter上右键单击，选“Configure”。我们可以注意到Suppliers_Select里的Data Columns集里已经包含了FullContactName列。

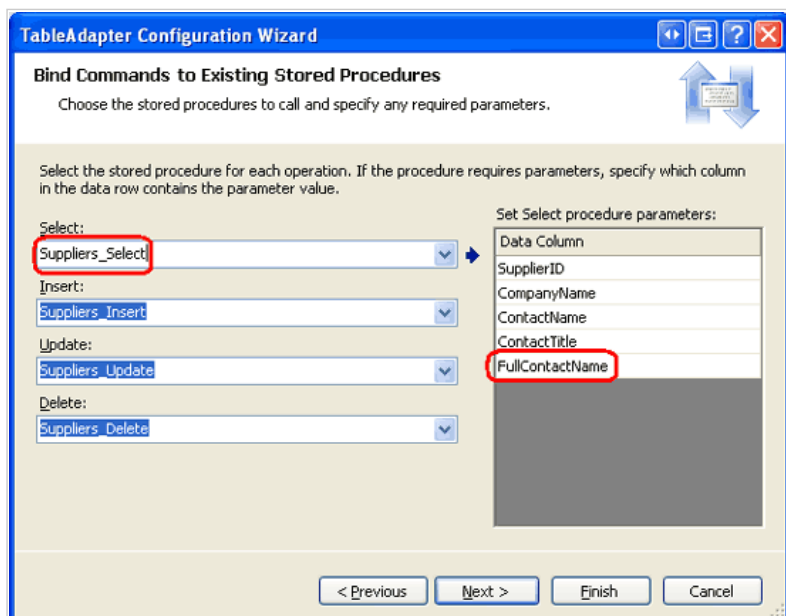


图6：返回到TableAdapter的设置向导更新DataTable的列

点击Finish完成设置，这将自动地为SuppliersDataTable添加相应的列.TableAdapter发觉FullContactName列是一个computed column列，且是只读的.因此将设置该列的ReadOnly属性为true.我们可以进行验证：在SuppliersDataTable里选择该列，打开其属性窗口(如图7)，我们注意到FullContactName列的DataType 和 Max Length属性都作了相应的设置。

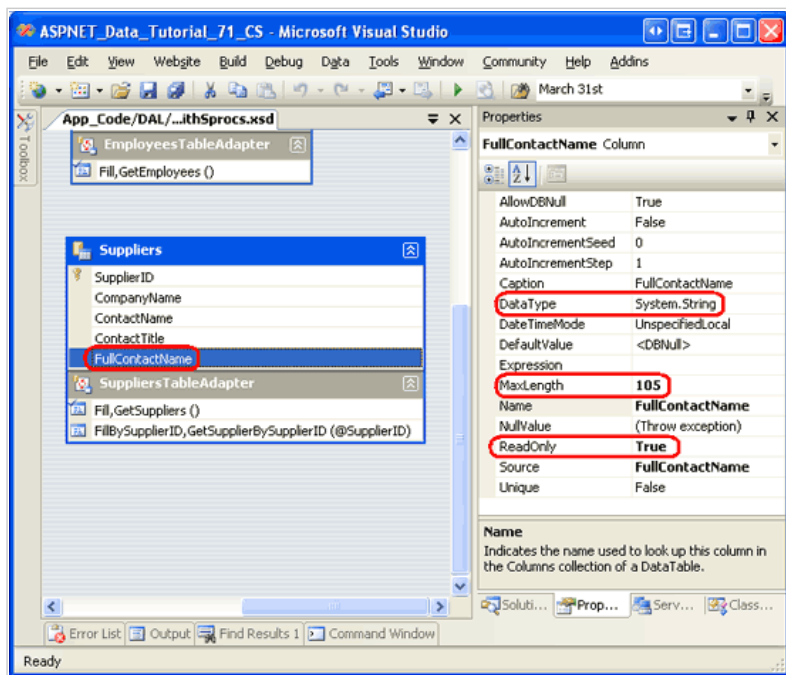


图7：FullContactName列标记为Read-Only

第五步：向TableAdapter添加一个GetSupplierBySupplierID方法

在本文我们将在一个具有更新功能的ASP.NET页面里展示suppliers信息.在前面的文章里，我们从DAL获取指定的记录并将其作为一个强类型的DataTable返回给BLL以做更新，然后将更新后的DataTable再传递给DAL，对数据库做相应的改动.为此，第一步——从DAL返回要更新的记录——我们需要向DAL层添加一个名为GetSupplierBySupplierID(supplierID)的方法。

在DataSet Design设计器里右键单击SuppliersTableAdapter，选“Add Query”，再选“Create new stored procedure”(可参考3图)。再选“SELECT which returns rows”再点Next。

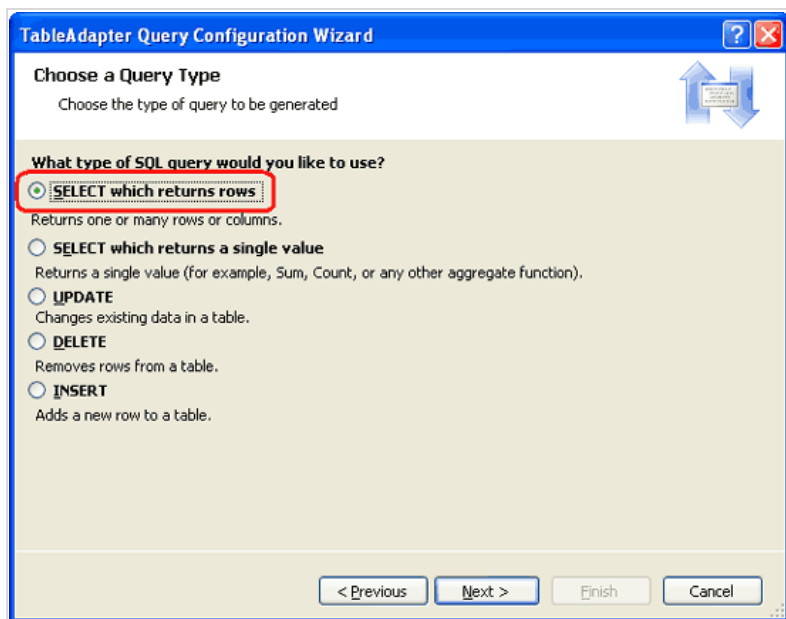


图8：选“SELECT which returns rows”项

接下来为该方法指定查询，键入如下的代码，它将检索某个具体的supplier返回的列与主查询一样。

```
1 SELECT SupplierID, CompanyName, ContactName, ContactTitle, FullContactName
2 FROM Suppliers
3 WHERE SupplierID = @SupplierID
```

接下来我们将该存储过程命名为Suppliers_SelectBySupplierID，点Next。

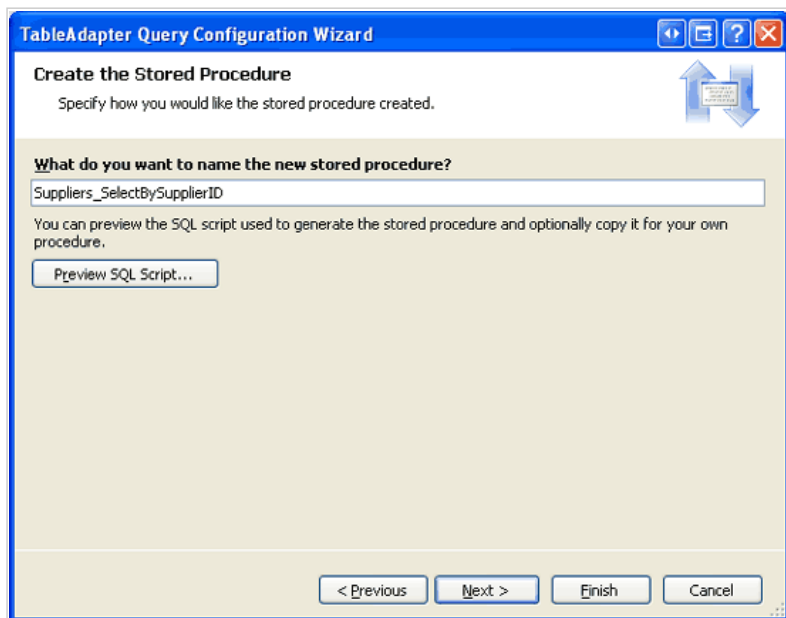


图9：将存储过程命名为**Suppliers_SelectBySupplierID**

在接下来的界面，全部选中图里的2项，并将**FillBy** 和 **GetDataBy**方法分别命名为**FillBySupplierID** 和 **GetSupplierBySupplierID**。

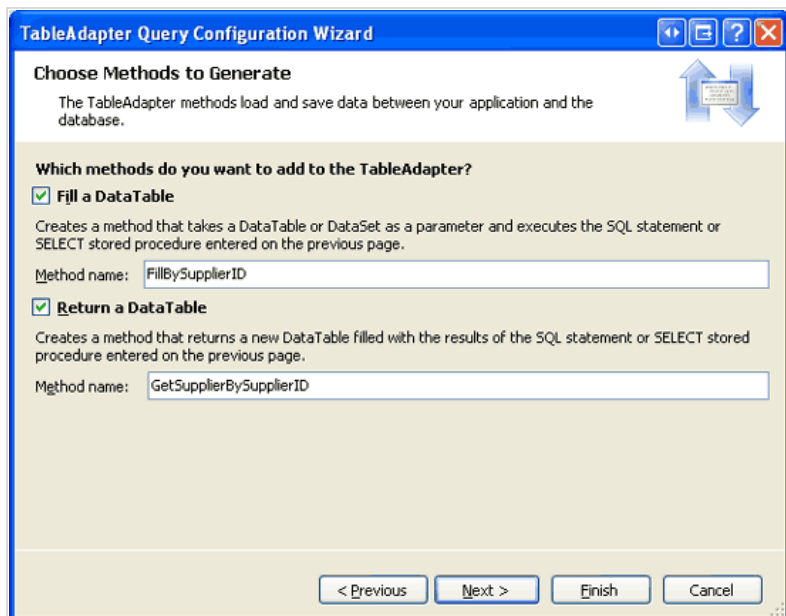


图10：将**TableAdapter**的方法命名为**FillBySupplierID** 和 **GetSupplierBySupplierID**

点**Finish**完成向导

第六步：创建业务逻辑层 **Business Logic Layer**

在创建ASP.NET页面前，我们首先要在BLL添加相应的方法.我们将在第7步创建页面，其允许我们查看并编辑suppliers.因此我们在BLL至少要包含2个方法，一个获取所有的suppliers，一个用于更新某个具体的supplier.

在~/App_Code/BLL文件夹里创建一个名为SuppliersBLLWithSprocs的新类，添加代码如下：

```

1  using System;
2  using System.Data;
3  using System.Configuration;
4  using System.Web;
5  using System.Web.Security;
6  using System.Web.UI;
7  using System.Web.UI.WebControls;
8  using System.Web.UI.WebControls.WebParts;
9  using System.Web.UI.HtmlControls;
10 using NorthwindWithSprocsTableAdapters;
11
12 [System.ComponentModel.DataObject]
13 public class SuppliersBLLWithSprocs
14 {
15     private SuppliersTableAdapter _suppliersAdapter = null;
16     protected SuppliersTableAdapter Adapter
17     {
18         get

```

```

19 {
20     if (_suppliersAdapter == null)
21         _suppliersAdapter = new SuppliersTableAdapter();
22     return _suppliersAdapter;
23 }
24 }
25 }
26
27 [System.ComponentModel.DataObjectMethodAttribute
28 (System.ComponentModel.DataObjectMethodType.Select, true)]
29 public NorthwindWithSprocs.SuppliersDataTable GetSuppliers()
30 {
31     return Adapter.GetSuppliers();
32 }
33
34 [System.ComponentModel.DataObjectMethodAttribute
35 (System.ComponentModel.DataObjectMethodType.Update, true)]
36 public bool UpdateSupplier(string companyName, string contactName,
37 string contactTitle, int supplierID)
38 {
39     NorthwindWithSprocs.SuppliersDataTable suppliers =
40     Adapter.GetSupplierBySupplierID(supplierID);
41     if (suppliers.Count == 0)
42         // no matching record found, return false
43         return false;
44
45     NorthwindWithSprocs.SuppliersRow supplier = suppliers[0];
46
47     supplier.CompanyName = companyName;
48     if (contactName == null)
49         supplier.SetContactNameNull();
50     else
51         supplier.ContactName = contactName;
52     if (contactTitle == null)
53         supplier.SetContactTitleNull();
54     else
55         supplier.ContactTitle = contactTitle;
56
57     // Update the product record
58     int rowsAffected = Adapter.Update(supplier);
59
60     // Return true if precisely one row was updated, otherwise false
61     return rowsAffected == 1;
62 }
63 }

```

和它的BLL class类一样，SuppliersBLLWithSprocs有一个protected Adapter属性，2个public方法：GetSuppliers 和 UpdateSupplier.其中，GetSuppliers方法调用Data Access Layer层对应的GetSupplier方法，该方法将SuppliersDataTable返回给BLL层；而UpdateSupplier方法通过调用DAL层的GetSupplierBySupplierID(supplierID)方法来获取某人具体supplier的信息，然后更新其CompanyName, ContactName,ContactTitle属性，再将修改后的SuppliersRow对象传递给Data Access Layer层的 Update方法，以对数据库做相应的更新。

注意：除了SupplierID 和 CompanyName外，Suppliers表的所有列都允许为NULL值，所以如果传递的contactName 或 contactTitle参数为null的话，我们将分别调用SetContactNameNull 和 SetContactTitleNull方法来将ContactName 和 ContactTitle 属性设置为NULL。

第七步：在表现层处理Computed Column列

做完了所有的必要工作后，我们将创建一个ASP.NET页面来处理FullContactName列，打开AdvancedDAL文件夹里的ComputedColumns.aspx 页面，拖一个GridView控件到页面，设其ID为Suppliers，在其智能标签里绑定到一个名为SuppliersDataSource的ObjectDataSource控件，设置其调用SuppliersBLLWithSprocs类，点Next.

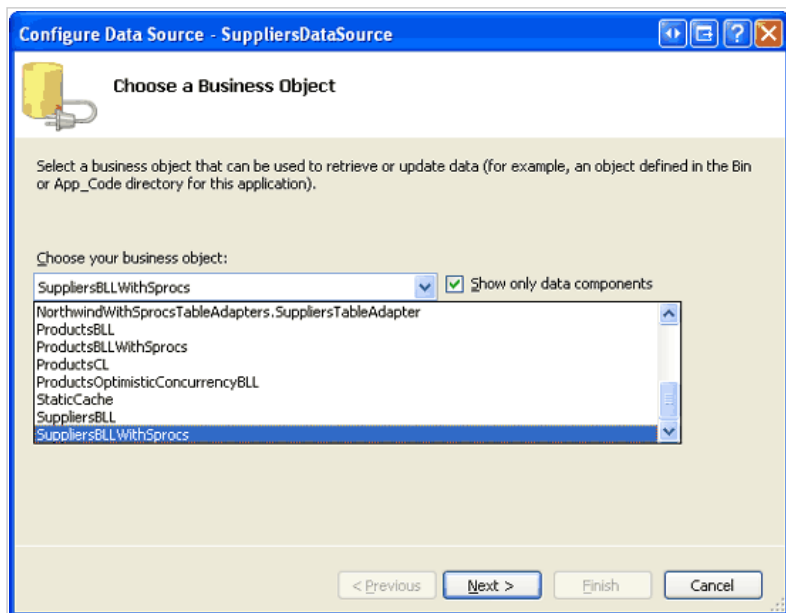


图 11: 设置 **ObjectDataSource** 调用 **SuppliersBLLWithSprocs Class** 类

在 **SuppliersBLLWithSprocs** 类里只有 2 个方法 **GetSuppliers** 和 **UpdateSupplier**. 确保在 **SELECT** 和 **UPDATE** 标签里分别选中这 2 个方法, 点 **Finish** 完成设置. 完成设置后, **Visual Studio** 将添加相应的 **BoundField**, 移除 **SupplierID** 列, 并将 **CompanyName**, **ContactName**, **ContactTitle** 和 **FullContactName** 列的 **HeaderText** 属性分别设置为 **"Company"**, **"Contact Name"**, **"Title"**, **"Full Contact Name"**, 再启用 **GridView** 的编辑功能.

Visual Studio 将 **ObjectDataSource** 控件的 **OldValuesParameterFormatString** 属性设置为 **"original_{0}"**. 我们要将其改为默认值 **"{0}"**. 如此这般, **GridView** 和 **ObjectDataSource** 控件的声明代码看起来和下面的差不多:

```

1  <asp:GridView ID="Suppliers" runat="server" AutoGenerateColumns="False" ?
2  DataKeyNames="SupplierID" DataSourceID="SuppliersDataSource">
3  <Columns>
4  <asp:CommandField ShowEditButton="True" />
5  <asp:BoundField DataField="CompanyName"
6  HeaderText="Company"
7  SortExpression="CompanyName" />
8  <asp:BoundField DataField="ContactName"
9  HeaderText="Contact Name"
10 SortExpression="ContactName" />
11 <asp:BoundField DataField="ContactTitle"
12 HeaderText="Title"
13 SortExpression="ContactTitle" />
14 <asp:BoundField DataField="FullContactName"
15 HeaderText="Full Contact Name"
16 SortExpression="FullContactName"
17 ReadOnly="True" />
18 </Columns>
19 </asp:GridView>
20
21 <asp:ObjectDataSource ID="SuppliersDataSource" runat="server"
22 SelectMethod="GetSuppliers" TypeName="SuppliersBLLWithSprocs"
23 UpdateMethod="UpdateSupplier">
24 <UpdateParameters>
25 <asp:Parameter Name="companyName" Type="String" />
26 <asp:Parameter Name="contactName" Type="String" />
27 <asp:Parameter Name="contactTitle" Type="String" />
28 <asp:Parameter Name="supplierID" Type="Int32" />
29 </UpdateParameters>
30 </asp:ObjectDataSource>

```

接下来我们在浏览器里登录该页面, 如图 12 所示. 每行都有一个 **FullContactName** 列, 格式为 **"ContactName (ContactTitle, CompanyName)"**.

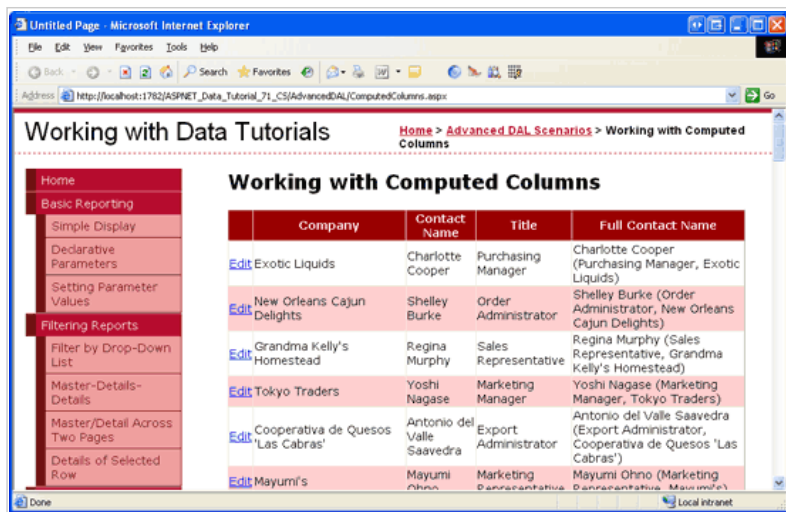


图12：每行展示一个Supplier

点击某行的Edit按钮将导致页面回传，且该行显示为一个编辑界面(如图13)，头3行呈现为默认的编辑界面——一个TextBox控件，且其Text属性为该数据域(data field)的值。不过FullContactName列仍然呈现为一个文本框，在Data Source设置向导完成并向GridView控件添加完相应的BoundFields时，FullContactName Bound Field的ReadOnly属性为true。我们在第四步注意到，FullContactName列的ReadOnly属性为true，因为TableAdapter意识到该列为一个computed column列。

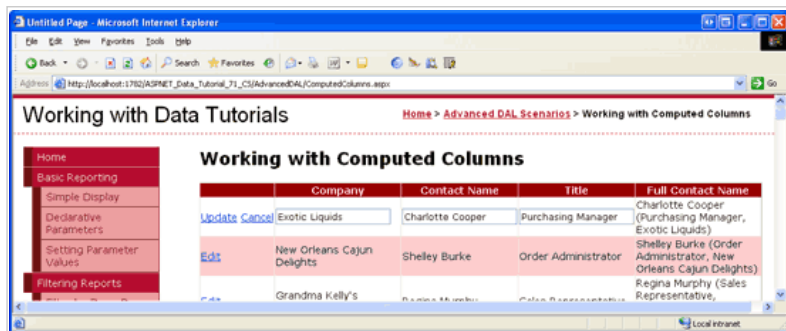


图13：FullContactName列为只读

我们改动这3个列中至少一个列的值，点Update按钮。我们发现FullContactName列的值跟着发生改变。

注意：由于GridView当前用的是BoundFields，导致编辑时用的是默认的界面。又由于CompanyName列是必需的，我们应将其转化成一个TemplateField以包含一个RequiredFieldValidator控件。我将此作为一个练习留给读者，你可以参考第19章《给编辑和新增界面增加验证控件》，看如何一步步的将BoundField转换成TemplateField，再添加一个确认控件。

结语：

当创建一个表时，Microsoft SQL Server允许我们创建一个computed columns列。这些computed columns列引用该条记录的其它列，再通过一个表达式对其赋值。由于其值来源于一个表达式，因此这种列是只读的，且不能通过INSERT或UPDATE statement对其赋值。正是如此，当在一个TableAdapter的主查询里引用computed column的话，要想自动的生成对应的INSERT, UPDATE,和DELETE statements有点麻烦。

在本文，我们探讨了使用computed columns列面临的挑战。具体来说，由于使用ad-hoc SQL statements的TableAdapters自身固有的不稳定性，我们探讨了使用存储过程的情况。当使用TableAdapter向导创建一个新的存储过程的时候，很重要的一点是，最开始不要在主查询里引用任何的computed columns，不然就不能自动的生成对应的存储过程。完成向导后，我们要手动修改SelectCommand属性以引用computed columns列。

祝编程快乐！

作者简介

本系列教程作者 Scott Mitchell，著有六本ASP/ASP.NET方面的书，是4GuysFromRolla.com的创始人，自1998年以来一直应用 微软Web技术。大家可以点击查看全部教程《[翻译]Scott Mitchell 的ASP.NET 2.0数据教程》，希望对大家的学习ASP.NET有所帮助。

您可能感兴趣的文章：

[DevExpress GridControl实现根据RowIndex和VisibleColumnsIndex来获取单元格值](#)

[如何让easyui gridview 宽度自适应窗口改变及fitColumns应用](#)

[ERROR 1222 \(21000\): The used SELECT statements have a different number of columns](#)



设备管理系统



it培训机构排名



哈尔滨自由行



oa系统



自学asp.net

自学网络编程 立体车库 酒店管理系统

哈尔滨旅游 真正云主机 家居装修网

u盘定制 插卡水表价格 开发一个app多少钱

液晶拼接屏 激光切割机 透明松紧带

电视电脑一体机 编程培训 微型摄像机

注册会计师培训 近视眼恢复 打码机

在线进销存 小型服务器 一级建造师改革

水表 光纤激光切割机 空压机价格

Tags: ASP.NET Computed Columns

相关文章

在ASP.NET 2.0中操作数据之十一：基于数据的自定义格式化	2016-05-05
在ASP.NET 2.0中操作数据之七十四：用Managed Code创建存储过程和	2016-05-05
在ASP.NET 2.0中操作数据之三十一：使用DataList来一行显示多条记	2016-05-05
基于.NET平台常用的框架和开源程序整理	2016-03-03
解读ASP.NET 5 & MVC6系列教程（1）：ASP.NET 5简介	2016-06-06
解读ASP.NET 5 & MVC6系列教程（8）：Session与Caching	2016-06-06
在ASP.NET 2.0中操作数据之五十八：在程序启动阶段缓存数据	2016-05-05
在ASP.NET 2.0中操作数据之六十：创建一个自定义的Database-Driv	2016-05-05
在ASP.NET 2.0中操作数据之六十五：在TableAdapters中创建新的存	2016-05-05
AlternatingItemTemplate类似于 ItemTemplate 元素	2006-10-10



学习ps基础



it培训机构排名



ps基础学习



自学英语零基础



ps自学网



面膜排行榜



数据库学习



新手学电脑



crm客户管理系



室外无线网桥



ps入门学习



钻戒品牌排行

最新评论

评论(0人参与，0条评论)



来说两句吧...

微博登录

QQ登录

手机登录

还没有评论，快来抢沙发吧！

Powered by 畅言