《ASP.NET动态网页设计项目教程（第三版）》

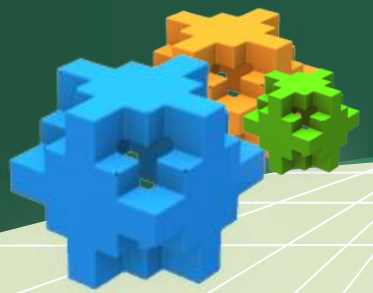
科学出版社　　主编 ： 郭建东

项目9 存储过程及事务处理

教学目标

- 知识教学目标
  - 创建存储过程
  - 了解输入参数和输出参数及获取返回结果
  - 图表显示查询结果
  - 创建事务处理
- 技能培养目标
  - 掌握存储过程的应用
  - 掌握事务处理的应用

目 录



任务9.1 客户订单查询

任务描述：

在电子商务网站中，会有大量的用户高频率地进行产品的查询、订单查看等操作，采用存储过程实现信息的查找有助于提高查找效率。本任务采用存储过程查看所选客户的所有订单。

任务9.1 客户订单查询

关键知识：创建存储过程

创建存储过程的两种方式：

（1）启动 MMSM 数据库管理器，单击“新建查询”命令，编写存储过程代码，单击“执行”命令，完成存储过程的创建。

（2）使用 Visual Studio.NET 创建网站，打开网页，创建存储过程步骤如下：

1） 打开“视图“菜单下的“服务器资源管理器”面板，选择“数据连接->添加连接”命令，或单击“工具”菜单，选择“连接到数据库”命令，在弹出的对话框中的服务器名输入.\sqlexpress，数据库选择 Northwind，点击下一步完成后可在服务器资源管理器面板中看到 Northwind 数据库。

2） 在数据库目录树中展开“存储过程”（Stored Procedures）节点，右击节点，在弹出的对话框中选择“添加新存储过程”（Add New Stored Procedure）命令。

3） 在弹出的存储过程编辑器中输入需要定义的存储过程并执行，完成存储过程的编译。

任务9.1 客户订单查询

关键知识：应用存储过程

通过 Command 类应用存储过程，常用如下两种方式：

（1）存储过程名称作为 Command 类对象参数进行调用，代码如下：

```
SqlCommand cmd = new SqlCommand("存储过程名", conn);
cmd.CommandType = CommandType.StoredProcedure;
```

（2）另一种是建立一个方法，返回 Command 命令类，与 SqlDataAdapter 配合使用，产生表格形式的输出结果，代码如下：

```
private static SqlCommand GenerateSelectCommand(SqlConnection conn)
{
   SqlCommand cmd = new SqlCommand("存储过程名", conn);
   cmd.CommandType = CommandType.StoredProcedure;
   return cmd;
}
```

任务9.1 客户订单查询

关键知识：存储过程中的输入输出参数

输入参数的转换并赋值方法 1 代码如下：

```
cmd.Parameters.Add(new SqlParameter("@CustomerID", SqlDbType.NChar,5));
cmd.Parameters["@CustomerID"].Value = customerlist.SelectedValue;
```

输入参数转换并赋值方法 2 代码如下：

```
SqlParameter pname = new SqlParameter("@CustomerID ", customerlist.
SelectedValue);
cmd.Parameters.Add(pname);
```

（2）输出参数的转换及获取结果

以@lastname NVarChar（20）输出参数为例，存储过程的输出参数转换代码如下：

```
cmd.Parameters.Add(new SqlParameter("@lastname",
SqlDbType.NVarChar, 20));
cmd.Parameters["@lastname"].Direction = ParameterDirection.Output;
```

获取输出参数值代码如下：

```
cmd.Parameters["@lastname"].Value.ToString()
```

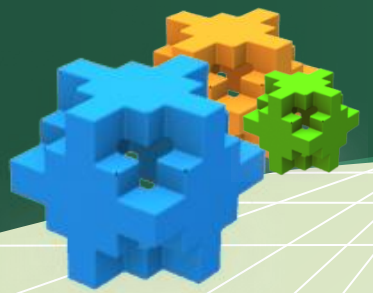
任务9.1 客户订单查询

任务实施：

请选择客户： ANATR

OrderID	OrderDate	RequiredDate	ShippedDate
10308	1996-9-18 0:00:00	1996-10-16 0:00:00	1996-9-24 0:00:00
10625	1997-8-8 0:00:00	1997-9-5 0:00:00	1997-8-14 0:00:00
10759	1997-11-28 0:00:00	1997-12-26 0:00:00	1997-12-12 0:00:00
10926	1998-3-4 0:00:00	1998-4-1 0:00:00	1998-3-11 0:00:00

图 9-1　客户订单查询



任务9.2 产品查询

任务描述：

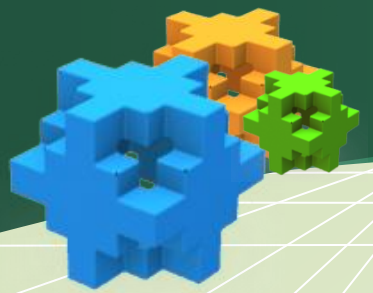
在电子商务网站中，产品信息的查找属于高频事件。为了提高产品的查找效率，可采用存储过程实现产品信息的查找。用户在搜索框输入要查找产品的关键字，通过编程进行模糊查找，使得用户可搜索出包括关键字的所有产品信息。

任务9.2 产品查询

关键知识：数据库的模糊查询

在产品表中实现对产品名称的模糊查找语句如下：

```
select * from Products where productname like '%'+@productname+'%'
```



任务9.2 产品查询

关键知识：服务器端表格控件

表格控件由行和单元格组成，使用表格控件显示内容需要先建立表格行对象并实例化，用Add方法将行添加到表格对象中；然后再创建单元格对象并实例化，往单元格中添加内容后，再通过row.Cells.Add方法将单元格添加到相应的行中。

```
while (dr.Read())
  {    TableRow row = new TableRow();
       tabresult.Rows.Add(row);
       for (int i = 0; i < 2; i++)   {
            TableCell cell = new TableCell();
            cell.Text = Convert.ToString(dr[i]);
            row.Cells.Add(cell);   }
  }
```

任务9.2 产品查询

任务实施：

请输入要查找的产品名 ch 查找

1	Chai
2	Chang
4	Chef Anton's Cajun Seasoning
5	Chef Anton's Gumbo Mix
12	Queso Manchego La Pastora
19	Teatime Chocolate Biscuits
26	Gumbär Gummibärchen
27	Schoggi Schokolade
34	Sasquatch Ale
39	Chartreuse verte
41	Jack's New England Clam Chowder
48	Chocolade
55	Pâté chinois
56	Gnocchi di nonna Alice

图 9-2　产品的模糊查询

任务9.3 种类产品数量查询及图表显示

任务描述：

用图表进行数据的显示，往往更直观方便。本任务要获得数据库中不同种类产品的数量，涉及两张表：种类表和产品表，对同一种类的产品进行数量统计，同时以图表的形式进行显示。

任务9.3 种类产品数量查询及图表显示

关键知识：图表控件

- **图表控件的事件**

编写Chart图表控件的DataBound事件，在DataBound事件中利用Chart控件的DataManipulator属性的Sort方法可对图表任意轴上的系列进行升序或降序排序。

- **图表控件序列的配置**

在图表控件中，需要设定以图表元素Series["Series1"]方式显示，这里Series1代表图表元素，选定了图表类型后，需要对系列值进行设定。

任务9.3 种类产品数量查询及图表显示

任务实施：

查找

种类产品数量统计

CategoryName	Count
aab	2
Beverages	12
Condiments	12
Confections	13
Dairy Product	10
Grains/Cereals	7
Meat/Poultry	6
Produce	5
Seafood	12

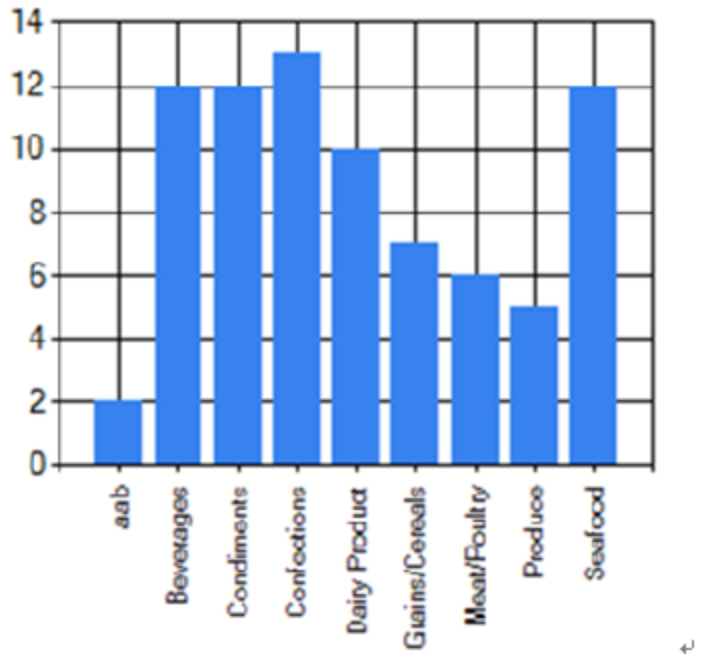
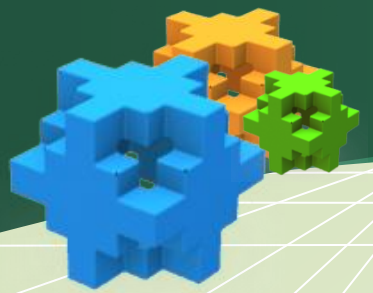


图 9-3　种类产品数量统计



任务9.4 ADO.NET事务处理

任务描述：

- 本任务以添加客户表信息为例进行事务处理演示。
- 以Customers表为例，在程序中事务开始后批量插入多条信息，提交事务后，当其中一条信息因各种原因无法被插入到数据库时，根据事务处理的原则，需要调用Rollback()方法对事务进行回滚处理，回滚操作后，在事务开始到事务结束期间对数据库的操作将全部取消，回到事务开始前状态。

任务9.4 ADO.NET事务处理

关键知识：事务处理

- 事务是一组相关的任务，一个数据库的事务通常包含多个操作，多个操作绑定在一起，这批操作中任意一个任务失败，都会使得前面的任务回滚到开始时的状态。
- 在ADO.NET中，用System.Data.Common.DbTransaction类表示数据库的事务，对于SQL Server数据库来说，SqlTransaction表示数据库的事务。
- SqlTransaction常用方法：

  - BeginTransaction()：启动本地事务，事务启动后，可以用 Command 对象的 Transaction 属性在该事务中登记命令，再根据事务组件的成功或失败决定提交或回滚数据源上进行的修改。
  - Commit()：提交数据库事务。
  - Rollback()：事务执行不成功，回滚事务。

任务9.4 ADO.NET事务处理

任务实施：

这里所插入的第二条记录的第一个字段，因数据库限定该字段长度为 5 个字符，但插入的数据为6 个字符，因此不能正确插入数据库。因编写了事务处理程序，所以连带第一条记录也不能被正确插入。

图 9-4　事务处理案例