软件工程师培养丛书

工信部国家级计算机人才评定体系

ASP.NET网站开发

武汉厚溥教育科技有限公司　编著

“理论→总结→上机→习题”四阶段教学模式

- 理论结合实践，注重动手能力培养
- 任务驱动讲解，有效激发学习兴趣
- 典型项目案例，扎实培养专业素质
- 教学做一体化，极大提高教学效率

清华大学出版社

软件工程师培养丛书

工信部国家级计算机人才评定体系

ASP.NET 网站开发

武汉厚溥教育科技有限公司　编著

清华大学出版社

北　京

内 容 简 介

本书按照高等院校、高职高专计算机课程基本要求，以案例驱动的形式来组织内容，突出计算机课程的实践性特点。本书共包括 7 章：LINQ to SQL、用户控件与 HttpHandler、成员资格和角色管理、个性化用户配置、数据缓存、母版页与站点导航、项目整合和主题。

本书附赠 PPT 教学课件和相关教辅资料，这些教学资源可通过 http://www.tupwk.com.cn 或 http://www.hop-e.net下载。

本书内容安排合理，层次清楚，通俗易懂，实例丰富，突出理论和实践的结合，可作为各类高等院校、高职高专及培训机构的教材，也可供广大程序设计人员参考。

图书在版编目(CIP)数据

ASP.NET 网站开发/武汉厚溥教育科技有限公司编著.—北京：清华大学出版社，2016
(软件工程师培养丛书)
ISBN 978-7-302-42410-9

Ⅰ. ①A… Ⅱ. ①武… Ⅲ. ①网页制作工具－程序设计 Ⅳ. ①TP393.092

中国版本图书馆 CIP 数据核字(2015)第 298590 号

责任编辑：刘金喜
封面设计：崔东方
版式设计：妙思品位
责任校对：成凤进
责任印制：刘海龙

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn,010-62794504

印 装 者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：20.25　　**字　　数**：341 千字
版　　次：2016 年 1 月第 1 版　　**印　　次**：2016 年 1 月第1 次印刷
印　　数：1～2500
定　　价：39.80 元

产品编号：067415-01

目　录

第1章
LINQ to SQL

课程目标

- LINQ to SQL
- 动态数据支持

简 介

LINQ to SQL 是将对象关系映射到.NET 框架中的一种实现。它可以将关系数据库(如 SQL Server 数据库)映射为.NET Framework 中的一些类。然后，开发人员就可以使用 LINQ to SQL 对数据库中的数据进行查询、修改、插入、删除等操作。总之，通过使用 LINQ to SQL，开发人员可以使用 LINQ 技术访问基于关系的数据库，就如同访问内存中的集合一样。

1.1 LINQ to SQL 概述

LINQ to SQL 是 LINQ 中最重要的一个组件，为.NET Framework 3.5 所支持，它可以为关系数据库提供一个对象模型，并在该对象模型基础上实现对数据的查询、添加、修改、删除等功能，即 LINQ to SQL 提供了用于将关系数据作为对象管理的运行时基础结构。本章节介绍 LINQ to SQL 的基础知识，以及使用 LINQ to SQL 为 SQL Server 数据库创建对象模型的方法。

为了更加具体地说明 LINQ to SQL 的功能，以下使用 SQL Server 数据库来介绍 LINQ to SQL 的功能。

LINQ to SQL 最重要的一个功能就是为 SQL Server 数据库创建一个对象模型(由基于.NET 框架的类组成)，并将该对象模型映射到 SQL Server 数据库中相应的对象(如表、列、外键关系、存储过程、函数等)。其中，LINQ to SQL 类映射到 SQL Server 数据库中的表，这些 LINQ to SQL 类被称为“实体类”。实体类中的属性或字段映射到 SQL Server 数据库中表的列。实体类之间的关联映射到 SQL Server 数据库中的外键关系。LINQ to SQL 类中的方法映射为 SQL Server 数据库中的存储过程或函数。LINQ to SQL 对象模型和 SQL Server 数据库中的对象的映射关系如表 1-1 所示。

表 1-1　LINQ to SQL 对象模型和 SQL Server 数据库中的对象映射关系

LINQ to SQL 对象模型的基本元素	SQL Server 数据库中的对象
实体类	表
属性或字段	列
关联	外键关系
方法	存储过程或函数

下面使用 SQL Server 数据库 LinqDB 介绍 LINQ to SQL 对象模型和 SQL Server 数据库中的对象的映射关系。其中，LinqDB 数据库包含多个表，如 UserInfo、UserRole、Role 等。LinqDB 数据库中部分表(只列出了 UserInfo、UserRole 和 Role 表，其他表已经省略)的关系图如图 1-1 所示。

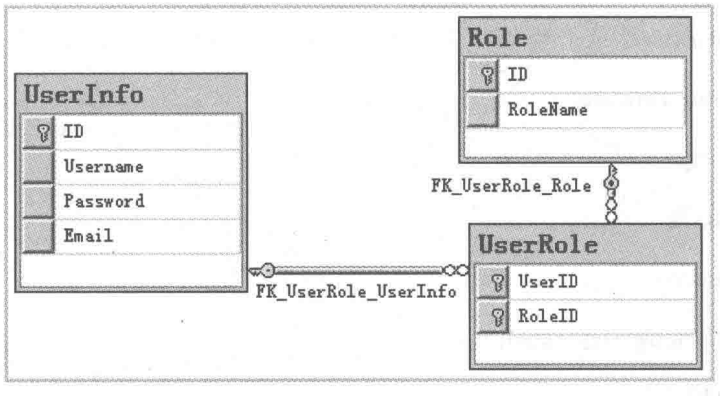


图 1-1

1. 实体类和数据库表

实体类和.NET Framework 中的其他类比较相似。但是，实体类使用 TableAttribute 属性来描述其与数据库中表的关联关系。下面的实例代码创建了一个名称为“UserInfo”的实体类，该实体类映射到 LinqDB 数据库中的 UserInfo 表。

```
[Table(Name="UserInfo")]
public class UserInfo { ... }
```

2. 属性或(字段)和数据库表中的列

如果一个实体类映射到数据库中的表，那么它的属性或字段可以被映射到数据库表中的列。其中，它们之间的映射关系使用 ColumnAttribute 属性表示。下面的实例代码创建了一个名称为 UserInfo 的实体类，该实体类映射到 LinqDB 数据库中的 UserInfo 表。ID 属性映射到 UserInfo 表的 ID 列，并且 ID 列为 UserInfo 表的主键。Username 属性映射到 UserInfo 表中的 Username 列。

```
[Table(Name="UserInfo")]
public class UserInfo
{
    private int iD;
    private string username;
    /// <summary>
    /// 映射 ID 列
    /// </summary>
    [Column(IsPrimaryKey = true)]
    public int ID
    {
        get { return iD; }
        set { iD = value; }
    }
    /// <summary>
    /// 映射 Username 列
    /// </summary>
    [Column]
    public string Username
    {
```

```
        get { return username; }
        set { username = value; }
    }
    ...
}
```

3. 关联和数据库外键关系

在 LINQ to SQL 中，LinqDB 数据库中的外键关联通过 AssociationAttribute 属性表示。LinqDB 数据库中的 UserInfo 和 UserRole 表之间就存在外键关系(UserRole 表的 UserID 列应用 UserInfo 表的 ID 列作为外键)。下面的实例代码在 UserInfo 类中创建了一个名称为 UserRole 的关联，该关联映射到 UserInfo 和 UserRole 表的外键关系(UserInfo_UserRole)。

```
[Table(Name="UserInfo")]
public class UserInfo
{
    ...
    private EntitySet<UserRole> _UserRole;
    [Association(Name="UserInfo_UserRole",
        Storage="_UserRole", OtherKey="UserID")]
    private EntitySet<UserRole> UserRole
    {
        get { return _UserRole; }
        set { _UserRole.Assign(value); }
    }
    ...
}
```

1.2 使用 Visual Studio 2008 创建 DBML 文件

本节介绍使用 Visual Studio 2008 为 SQL Server 数据库 LinqDB 创建 DBML 文件的方法，并为该数据库中的表(如 UserInfo、Role、UserRole、Category 等)创建实体类，为每一个表的列创建相应的属性。如 LinqDB 数据库中的 UserInfo 表将和 LinqDB.dbml 文件中的 UserInfo 类进行映射，UserInfo 表的列和 UserInfo 表的属性进行映射，它们之间的映射关系如图 1-2 所示。

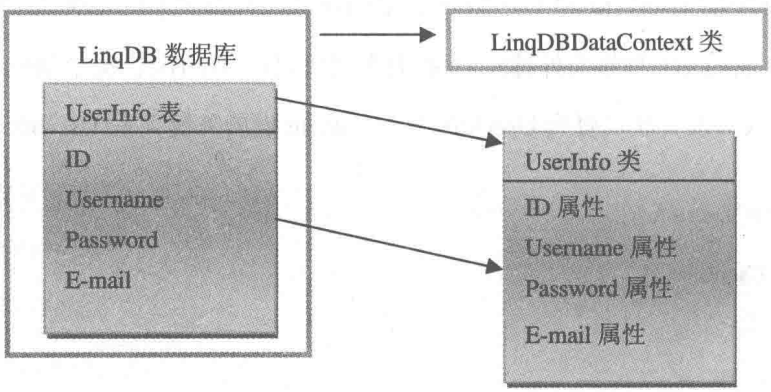


图 1-2

下面介绍在使用 Visual Studio 2008 为 SQL Server 数据库 LinqDB 创建 DBML 文件的方法，具体步骤如下。

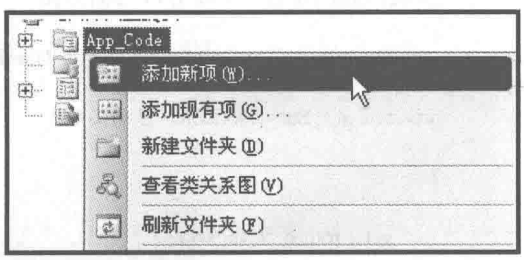


图 1-3

(1) 在 Visual Studio 2008 集成开发环境中，右键单击“解决方案资源管理器”面板中的“App_Code”节点，并选择“添加新项”命令，操作如图 1-3 所示。

(2) 单击“添加新项”命令，弹出“添加新项”对话框。选择“LINQ to SQL 类”图标，并在“名称”输入框中输入“LinqDB.dbml”，如图 1-4 所示。

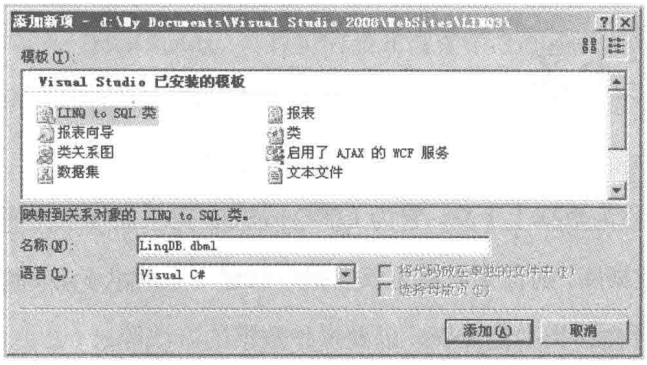


图 1-4

(3) 在“服务器资源管理器”面板中选择 LinqDB 数据库的各个表，并直接拖放到 LimlDB.dbml 文件的视图面板中。

(4) 将LinqDB 数据库中所有的表和存储过程都直接拖放到LinqDB.dbml文件的视图面板中。

(5) 单击按钮保存上述操作的修改，即可创建 LinqDB.dbml 文件。

(6) 在“解决方案资源管理器”面板中查看 LinqDB.dbml 文件。

(7) 其中，LinqDB.dbml.1ayout 文件保存 LinqDB.dbml 文件的布局，该文件为一个 XML 文件，如图 1-5 所示。

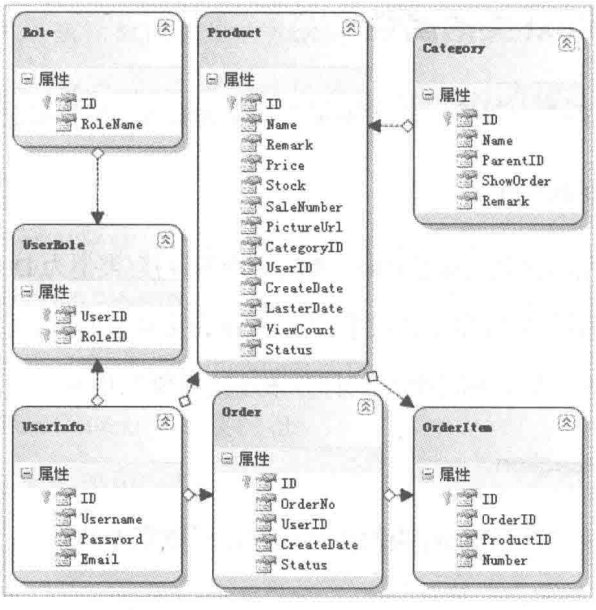


图 1-5

1.3 数据上下文

DataContext 又称为数据上下文，它为 LINQ to SQL 提供操作数据库的入口。如果使用 LINQ to SQL 操作数据库，则首先需要为该数据库创建一个继承于 DataContext 类的自定义的数据上下文类，并在该类中定义表，以及操作数据的方法等。

1.3.1 DataContext 概述

DataContext 类是一个 LINQ to SQL 类，它充当 SQL Server 数据库与映射到该数据库的 LINQ to SQL 实体类之间的管道，它包含用于连接数据库以及操作数据库数据的连接字符串信息和方法。DataContext 类能够通过数据库连接或连接字符串来映射数据库中的所有实体的源，并跟踪和标识用户对数据库的更改。用户可以调用其 SubmitChanges()方法将所有更改提交到数据库。

当 DataContext 类(或从其派生的类)的实例操作数据库时，实例会自动打开和关闭数据库的连接。如果用户显式打开了实例的连接，那么也必须显式关闭实例的连接。

1.3.2 DataContext 类的属性

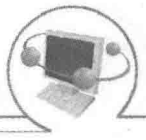
DataContext 类包括多个属性，如连接属性 Connection、事务属性 Transaction 等。

1. 连接属性 Connection

Connection 属性可以获取 DataContext 类的实例的连接(类型为 DbConnection)。值得注意的是，用户获取该属性的值(即连接对象)之后，该连接对象的默认状态是关闭的。因此，用户如果要使用该连接对象，则需要显式打开该连接对象的状态。

2. 事务属性 Transaction

Transaction 属性为 DataContext 类的实例设置访问数据库的事务。其中，LINQ to SQL 支持以下 3 种事务。



(1) 显式事务。如果 DataContext 类的实例显式设置了 Transaction 属性的值，那么，DataContext 类的实例调用 SubmitChanges()方法时将在同一个事务上进行。

(2) 隐式事务。当 DataContext 类的实例调用 SubmitChanges()方法时，LINQ to SQL 会检查该方法是否在已经指定的事务内，如果不是，则 LINQ to SQL 将启动本地事务，并使用此事务执行所生成的 SQL 命令。

(3) 显式可分发事务。DataContext 类的实例可以在 Transaction 属性指定的事务内调用 LINQ to SQL API，而不会创建新的事务。

3. 执行命令的最大时间属性 CommandTimeout

CommandTimeout 属性可以设置或获取 DataContext 类的实例的查询数据库操作的超时期限。该时间的单位为秒，默认值为 30 秒。有时，查询数据库操作可能需要很长的时间。此时，则需要增大该属性的值，以保证查询数据库的操作能够完成。

1.3.3 DataContext 类的方法

DataContext类包括多个方法，如创建数据库的 CreateDatabase()方法、检测数据库是否存在的 DatabaseExists()方法、删除数据库的 DeleteDatabase()方法、执行 SQL 命令的 ExecuteCommand()方法、执行 SQL 查询的 ExecuteQuery()方法等。

本节将重点讲解 ExecuteCommand()方法与 ExecuteQuery()方法。

1. 执行 SQL 命令的 ExecuteCommand()方法

ExecuteCommand()方法能够执行指定的 SQL 语句，并通过该 SQL 语句来操作数据库。ExecuteCommand()方法返回一个整数值，即该 SQL 语句影响记录的数量。

下面的实例代码调用 ExecuteCommand()方法执行指定的 SQL 语句。具体步骤如下：

(1) 创建 LinqDBDataContext 类的实例 db。

(2) 创建被执行的 SQL 语句。

(3) 调用 ExecuteCommand()方法执行上述指定的 SQL 语句。

(4) 显示 ExecuteCommand()方法执行的结果，即被修改记录的数量。

```
/// <summary>
/// 执行 SQL 命令的 ExecuteCommand()方法
/// </summary>
private void ExecuteCommandWithParam()
{
    //创建 LinqDB 数据库上下文的实例
    LinqDBDataContext db = new LinqDBDataContext(
                                LinqSystem.LinqDBConnectionString);
    //创建 SQL 语句
    string sql = "UPDATE UserInfo SET Username = {0} WHERE ID > 90";
    //执行 SQL 语句，并返回 SQL 语句影响的行数
    int result = db.ExecuteCommand(sql, "New Name");
    //显示执行的结果
    Response.Write(result.ToString() + "条数据被修改！ ");
}
```

执行结果如图 1-6 所示。

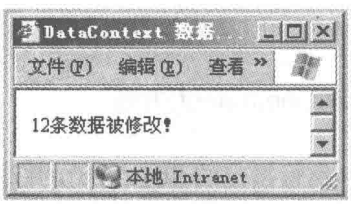
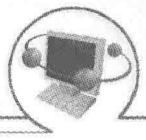


图 1-6

2. 执行 SQL 查询的 ExecuteQuery()方法

ExecuteQuery()方法可以执行指定的 SQL 查询语句，并通过 SQL 查询语句检索数据，查询结果保存数据类型为 IEnumerable 或 IEnumerable<TResult>的对象。

下面的实例代码调用 ExecuteQuery()方法执行指定的 SQL 查询语句。具体步骤如下：



(1) 创建 LinqDBDataContext 类的实例 db。

(2) 创建被执行的 SQL 查询语句。

(3) 调用 ExecuteQuery()方法执行上述指定的 SQL查询语句，查询结果保存在 result 变量中。其中，查询结果中的元素的数据类型为 UserInfo。

(4) 使用 foreach 语句显示 result 变量中的信息。

```
/// <summary>
/// 执行 SQL 查询的 ExecuteQuery()方法
/// </summary>
private void ExecuteSqlQuery()
{
    //创建 LinqDB 数据库上下文的实例
    LinqDBDataContext db = new LinqDBDataContext(
                                LinqSystem.LinqDBConnectionString);
    //创建 SQL 语句
    string sql = "SELECT TOP 5 * FROM UserInfo";

    IEnumerable<UserInfo> users
        = db.ExecuteQuery<UserInfo>(sql);
    //显示查询的结果
    foreach (UserInfo u in users)
    {
        Response.Write("用户名称：" + u.Username + "<br />");
    }
}
```

执行结果如图 1-7 所示。