

实验名称：数据访问

1. 实验要求：

- (1) 掌握在 VSC 2019 中建立、连接和管理数据库的方法。
- (2) 了解数据源控件的使用
- (3) 熟练掌握 LINQ 表达式的使用。
- (4) 熟练掌握利用 LINQ to SQL 和 LINQ to XML 进行数据访问管理的方法。

2. 实验内容：

1. 基于 Entity Framework 利用 LINQ to SQL 进行数据管理

- (1) 参考主教材 15.2 节中购物车、商品分类、用户、订单、订单详情信息、商品、供应商等数据表的设计及各字段含义，建立 MyPetShop 数据库。
- (2) 建立数据管理的导航页面。
- (3) 单击“显示全部”按钮时，显示 MyPetShop 数据库中 Category 表的所有内容。
- (4) 单击“模糊查找”按钮，输入要查找的内容，单击“搜索”按钮，当未找到满足条件的数据时显示“没有满足条件的数据!”，否则以 GridView 形式显示数据。
- (5) 单击“插入”按钮，输入“分类名”和“描述”，再单击“插入”按钮后，将向 Category 表添加一条记录，其中新增记录中的 Category 值会自动递增，单击“返回”将返回到数据管理的导航页面。
- (6) 输入“分类 Id”，再单击“修改”按钮，可修改“分类名”和“描述”，但不能修改“分类 Id”，修改完成后单击“修改”按钮将修改 Category 表中对应的数据，单击“返回”按钮将返回到数据管理的导航界面。
- (7) 输入“分类 Id”，再单击“删除”按钮将删除“分类 Id”值指定的记录。

2. 利用 LINQ 技术将 Category 表转换成 XML 文档 Category.xml。

3. 利用 LINQ to XML 管理 XML 文档。

- (1) 建立 Category.xml 数据管理的导航界面。

(2) 单击“显示全部”按钮，将新开一个浏览器窗口或选项卡显示 Category.xml 文件内容。

(3) 输入“分类名”，单击“查询”按钮，当未找到满足条件的数据时显示“没有满足条件的数据！”，否则以 Label 控件形式显示满足条件的数据。

(4) 单击“插入按钮”，输入“分类名 Id”“分类名”和“描述”，再单击“插入”按钮，向 Category.xml 中添加相应的元素。

3. 实验步骤：

2. 利用 LINQ to SQL 进行数据处理

- (1) 新建网站
- (2) 建立 MyPetShop 数据库
- (3) 建立 MyPetShop.dbml 文件
- (4) 设计 DataManager.aspx
- (5) 编写 DataManager.aspx.cs 中的方法代码
- (6) 设计 FuzzyQuery.aspx
- (7) 编写 FuzzyQuery.aspx.cs 中的方法代码
- (8) 设计 Insert.aspx
- (9) 编写 Insert.aspx.cs 中的方法代码
- (10) 设计 Update.aspx
- (11) 编写 Update.aspx.cs 中的方法代码
- (12) 测试

3. 利用 LINQ 技术将 Category 表转换成 XML 文档 Category.xml

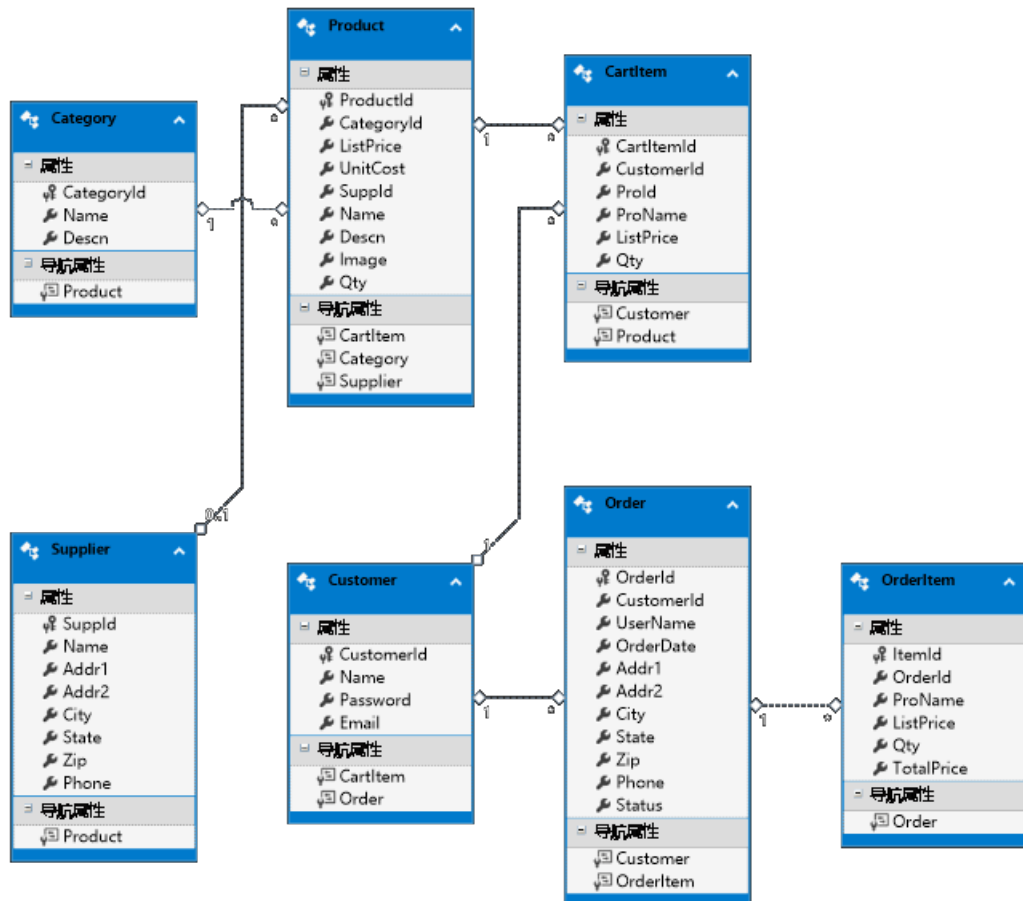
4. 利用 LINQ to XML 管理 XML 文档

- (1) 新建网站
- (2) 设计 LinqXml.aspx
- (3) 编写 LinqXml.aspx.cs 中的方法代码
- (4) 设计 LinqXmlInsert.aspx
- (5) 编写 LinqXmlInsert.aspx.cs 中的方法代码

(6) 测试

4. 实验代码:

1. 利用 LINQ to SQL 进行数据管理



```
<%@ Page Language="C#" AutoEventWireup="true"
    CodeFile="DataManage.aspx.cs" Inherits="DataManage" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Linq to SQL 管理数据</title>
</head>
<body>
    <form id="form1" runat="server">
```

```

<div>
    分类 Id: <asp:TextBox ID="txtCategoryId" runat="server"
        Width="250px">输入分类 Id, 只用于 “修改” 和 “删除” </asp:TextBox>
    <br />
    <asp:Button ID="btnQueryAll" runat="server"
        OnClick="BtnQueryAll_Click" Text="显示全部" Width="80px" />
    <asp:Button ID="btnFuzzy" runat="server"
        OnClick="BtnFuzzy_Click" Text="模糊查找" Width="80px" />
    <asp:Button ID="btnInsert" runat="server"
        OnClick="BtnInsert_Click" Text="插入" Width="50px" />
    <asp:Button ID="btnUpdate" runat="server"
        OnClick="BtnUpdate_Click" Text="修改" Width="50px" />
    <asp:Button ID="btnDelete" runat="server" Text="删除"
        OnClick="BtnDelete_Click" Width="50px" />
    <br />
    <br />
    <asp:GridView ID="gvCategory" runat="server">
    </asp:GridView>
</div>
</form>
</body>
</html>

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class DataManage : System.Web.UI.Page
{
    MyPetShopEntities db = new MyPetShopEntities();
    protected void Bind()
    {
        var results = db.Category.ToList();
        gvCategory.DataSource = results;
        gvCategory.DataBind();
    }
}

```



上海工程技术大学

实验报告

姓名 张浩宇 班级 0211213 学号 028121308 实验日期 2024.03.29
课程名称 Web 信息管理系统 指导教师 杨孜苗 成绩

```
protected void BtnQueryAll_Click(object sender, EventArgs e)
{
    Bind();
}

protected void BtnFuzzy_Click(object sender, EventArgs e)
{
    Response.Redirect("FuzzyQuery.aspx");
}

protected void BtnInsert_Click(object sender, EventArgs e)
{
    Response.Redirect("Insert.aspx");
}

protected void BtnUpdate_Click(object sender, EventArgs e)
{
    Response.Redirect("Update.aspx?CategoryId=" + txtCategoryId.Text);
}

protected void BtnDelete_Click(object sender, EventArgs e)
{
    var category = db.Category.Find(int.Parse(txtCategoryId.Text));

    db.Category.Remove(category);
    db.SaveChanges();
    Bind();
}
}

<%@ Page Language="C#" AutoEventWireup="true"
    CodeFile="FuzzyQuery.aspx.cs" Inherits="FuzzyQuery" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>模糊查找</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            分类名: <asp:TextBox ID="txtSearch" runat="server"></asp:TextBox>
```

```

        <asp:Button ID="btnSearch" runat="server" Text="搜索"
            OnClick="BtnSearch_Click" />
        <asp:Button ID="btnReturn" runat="server"
            OnClick="BtnReturn_Click" Text="返回" />
        <asp:GridView ID="gvCategory" runat="server">
        </asp:GridView>
        <asp:Label ID="lblMsg" runat="server"></asp:Label>
    </div>
</form>
</body>
</html>

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class FuzzyQuery : System.Web.UI.Page
{
    protected void BtnSearch_Click(object sender, EventArgs e)
    {
        MyPetShopEntities db = new MyPetShopEntities();
        var results = from c in db.Category.Where(c =>
c.Name.Contains(txtSearch.Text)).ToList() select c;
        if (results.Count() != 0)
        {
            gvCategory.DataSource = results;
            gvCategory.DataBind();
        }
        else
        {
            lblMsg.Text = "没有满足条件的数据! ";
        }
    }

    protected void BtnReturn_Click(object sender, EventArgs e)
    {
        Response.Redirect("DataManage.aspx");
    }
}

```

```
}  
}
```

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Insert.aspx.cs"  
    Inherits="Insert" %>
```

```
<!DOCTYPE html>  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
    <title>插入数据</title>  
    <style type="text/css">  
        .auto-style1 {  
            width: 100%;  
        }  
  
        .auto-style2 {  
            width: 71px;  
        }  
    </style>  
</head>  
<body>  
    <form id="form1" runat="server">  
        <div>  
            <table class="auto-style1">  
                <tr>  
                    <td class="auto-style2" style="text-align: right">分类名:  
                </td>  
                <td>  
                    <asp:TextBox ID="txtName"  
                        runat="server"></asp:TextBox></td>  
                </tr>  
                <tr>  
                    <td class="auto-style2" style="text-align: right; vertical-align:  
top">描述: </td>  
                    <td>  
                        <asp:TextBox ID="txtDescn" runat="server"  
                            TextMode="MultiLine" Height="50px"  
                            Width="148px"></asp:TextBox></td>
```

```

        </tr>
    </table>
    <asp:Button ID="btnInsert" runat="server" Text="插入"
        OnClick="BtnInsert_Click" />
    <asp:Button ID="btnReturn" runat="server" Text="返回"
        OnClick="BtnReturn_Click" />
</div>
</form>
</body>
</html>

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

```

```

public partial class Insert : System.Web.UI.Page
{
    protected void BtnInsert_Click(object sender, EventArgs e)
    {
        MyPetShopEntities db = new MyPetShopEntities();
        Category category = new Category();
        category.Name = txtName.Text;
        category.Descn = txtDescn.Text;
        db.Category.Add(category);
        db.SaveChanges();
    }
    protected void BtnReturn_Click(object sender, EventArgs e)
    {
        Response.Redirect("DataManage.aspx");
    }
}

```

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Update.aspx.cs"
    Inherits="Update" %>

```


[illegible]

```

Width="148px"></asp:TextBox></td>
        </tr>
    </table>
    <asp:Button ID="btnUpdate" runat="server"
        OnClick="BtnUpdate_Click" Text="修改" />
    <asp:Button ID="btnReturn" runat="server"
        OnClick="BtnReturn_Click" Text="返回" />
</div>
</form>
</body>
</html>

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Update : System.Web.UI.Page
{
    MyPetShopEntities db = new MyPetShopEntities();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            string categoryId = Request.QueryString["CategoryId"];
            var category = db.Category.Find(int.Parse(txtCategoryId.Text));
            txtCategoryId.Text = categoryId;
            txtCategoryId.ReadOnly = true;
            txtName.Text = category.Name;
            txtDescn.Text = category.Descn;
        }
    }
    protected void BtnUpdate_Click(object sender, EventArgs e)
    {
        var category = db.Category.Find(int.Parse(txtCategoryId.Text));
        category.Name = txtName.Text;
    }
}

```

```

        category.Descn = txtDescn.Text;
        db.SaveChanges();
    }

    protected void BtnReturn_Click(object sender, EventArgs e)
    {
        Response.Redirect("DataManage.aspx");
    }
}

```

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="LinqXml.aspx.cs"
Inherits="LinqXml" %>

```

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>利用 LINQ 将 Category 表转换为 XML 文件</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            </div>
    </form>
</body>
</html>

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Xml.Linq;

public partial class LinqXml : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

```

```

        string xmlFilePath = Server.MapPath("~/Category.xml");
        XDocument doc = new XDocument
        (
            new XDeclaration("1.0", "utf-8", "yes"),
            new XComment("分类"),
            new XElement("Categories")
        );
        doc.Save(xmlFilePath);
        XElement els = XElement.Load(xmlFilePath);
        MyPetShopEntities db = new MyPetShopEntities();
        var results = from c in db.Category
                      select c;
        foreach (Category category in results)
        {
            XElement el = new XElement("Category",
                new XElement("CategoryId", category.CategoryId),
                new XElement("Name", category.Name),
                new XElement("Descn", category.Descn));
            els.Add(el);
        }
        els.Save(xmlFilePath);
        Response.Redirect("~/Category.xml");
    }
}

```

```

<%@ Page Language="C#" AutoEventWireup="true"
    CodeFile="LinqXmlInsert.aspx.cs" Inherits="LinqXmlInsert" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Linq to XML 插入数据</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <table class="style1">
                <tr>

```

```

        <td class="style2">分类 Id:
    </td>
    <td>
        <asp:TextBox ID="txtCategoryId" runat="server"
            Width="120px"></asp:TextBox>
    </td>
</tr>
<tr>
    <td class="style2">分类名:
    </td>
    <td>
        <asp:TextBox ID="txtName" runat="server"
            Width="120px"></asp:TextBox>
    </td>
</tr>
<tr>
    <td class="style2">描述:
    </td>
    <td>
        <asp:TextBox ID="txtDescn" runat="server"
            Width="120px"></asp:TextBox>
    </td>
</tr>
</table>
<asp:Button ID="btnInsert" runat="server" Text="插入"
    OnClick="btnInsert_Click" />
<asp:Button ID="btnReturn" runat="server" Text="返回"
    OnClick="btnReturn_Click" />
</div>
</form>
</body>
</html>

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

```

```

using System.Xml.Linq;

public partial class LinqXmlInsert : System.Web.UI.Page
{
    protected void btnInsert_Click(object sender, EventArgs e)
    {
        string xmlFilePath = Server.MapPath("~/Category.xml");
        XElement els = XElement.Load(xmlFilePath);
        XElement el = new XElement("Category",
            new XElement("CategoryId", txtCategoryId.Text),
            new XElement("Name", txtName.Text),
            new XElement("Descn", txtDescn.Text));
        els.Add(el);
        els.Save(xmlFilePath);
    }

    protected void btnReturn_Click(object sender, EventArgs e)
    {
        Response.Redirect("LinqXml.aspx");
    }
}

```

实验截图：

分类 Id:

CategoryId	Name	Descn
1	Fish	Fish
2	Bugs	Bugs
3	Backyard	Backyard
4	Birds	Birds
5	Endangered	Endangered

分类名:

CategoryId	Name	Descn
2	Bugs	Bugs
3	Backyard	Backyard
4	Birds	Birds

分类名:
描述:

分类 Id:

CategoryId	Name	Descn
1	Fish	Fish
2	Bugs	Bugs
3	Backyard	Backyard
4	Birds	Birds
5	Endangered	Endangered
11	test	123

分类 Id:
分类名:
描述:

5. 实验小结:

1. 实验中容易出错的地方:

- (1) 在数据输入的过程中, SQL 在 mdf 生成的过程中没有权限, 查阅之后进行了文件夹权限的修改。

2. 心得体会:

- (1) 本次实验将 WEB 与数据库进行联动, 并且引入了 LINQ 表达, 通过 LINQ to SQL 来进行数据管理。LINQ 的接口是对本地的实体数据模型的操作, 与烦琐的通过 host 的固定端口访问远程数据库不同。