

单元6-

数据访问及实体框架

1. ASP.NET数据访问技术

实作1-使用SQL Server创建数据库

实作2-使用EF生成实体类并运用

2. 使用LINQ实现数据访问

实作3-使用EF+LINQ的查询运用

实作4-EF+LINQ的增删改



1. ASP.NET数据访问技术

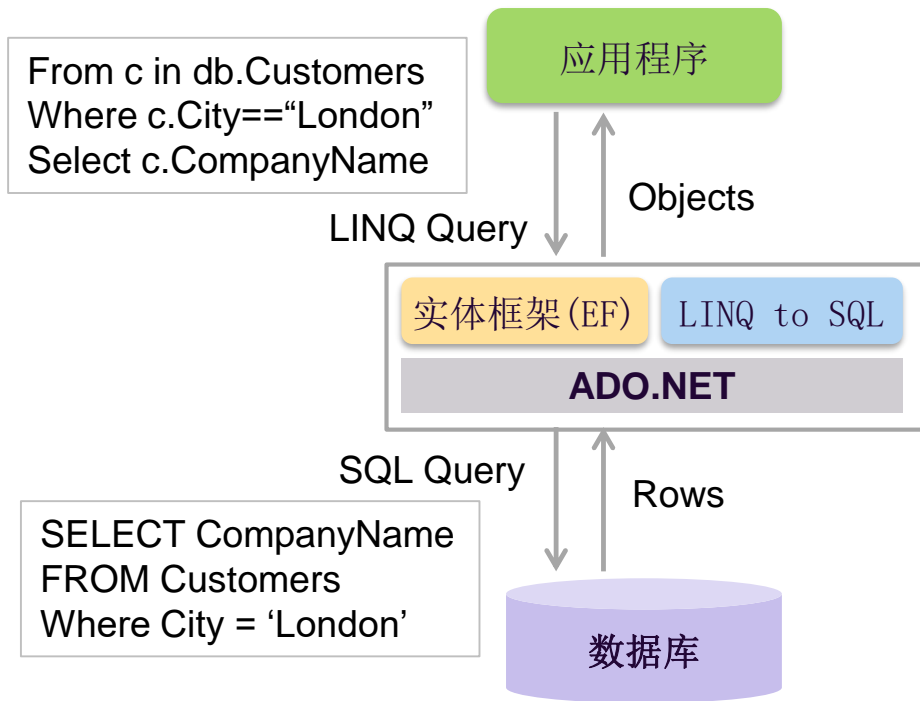


数据访问概述

- ASP.NET数据访问技术主要有两类：
 - 使用ADO.NET访问数据：传统技术。
 - 使用ORM（对象关系映射）技术：Entity Framework（实体框架）及LINQ to SQL。
- ADO.NET技术**：在C#代码中写SQL语句
- Entity Framework技术(本教程使用)**：ORM技术，使用LINQ查询表达式或使用LINQ扩充方法，对实体对象进行操作（CRUD）。
- LINT to SQL技术**：简化版ORM技术，使用方法类似EF技术，但在数据库方面只能接取SQL Server。



数据访问概述



说明:

1. EF及LINQtoSQL都是基于ADO.NET的ORM技术，并使用LINQ技术（查询表达式或扩充方法）对实体类进行操作。这些操作会自动转为SQL语句对后端数据库进行操作。
2. 传统的ASP.NET应用程序是直接透过ADO.NET访问数据库，需写SQL语句。



Entity Framework（实体框架）

- EF是微软公司的ORM技术，它使开发人员能够通过对象来操作数据库，不用再写SQL语句，使开发人员能更专注于代码的编写。
- 可结合LINQ语言的使用，使代码撰写更为简洁。
- EF有3种构建模式：
 - Database First（数据库优先）-本教程使用：由数据库生成模型以及实体类（Entity Class）。
 - Model First（模型优先）：先设计模型再生成实体类及数据库。
 - Code First（代码优先）：先以代码设计实体类（写C#代码），并透过这些实体类生成数据库及数据表。



SQL Server数据库管理系统

- ASP.NET可以使用各种常见的数据库管理系统。
- 在SQL Server有3类可选，用法均相同只有连接字符串不同（Web.config）。
 - VS工具自带的SQL Server Express LocalDB（如右图）-教程使用
 - Client-Server SQL Server
 - Azure SQL Server

安装详细信息

▼ .NET 桌面开发

▼ 已包含

- ✓ .NET 桌面开发工具
- ✓ .NET Framework 4.7.2 开发工具
- ✓ C# 和 Visual Basic

▼ 可选

- ☒ .NET 开发工具
- ☒ .NET Framework 4 – 4.6 开发工具
- ☒ Blend for Visual Studio
- ☒ Entity Framework 6 工具
- ☒ .NET 分析工具
- ☐ .NET Framework 4.7.1 开发工具
- ☒ .NET Framework 4.8 开发工具
- ☐ .NET 可移植库目标包
- ☐ Windows Communication Foundation
- ☒ SQL Server Express 2016 LocalDB
- ☐ PreEmptive Protection - Dotfuscator



使用的数据库简介-Booksdb

Categories 分类

字段	型别	约束	备注
CategoryID	INT	IDENTITY (1, 1) NOT NULL, PK	分类号
CategoryName	NVARCHAR (30)	NOT NULL	分类名
Description	NVARCHAR (100)	NULL	说明

Books 书籍

字段	型别	约束	备注
BookID	INT	IDENTITY (1, 1) NOT NULL, PK	书号
BookName	NVARCHAR (100)	NOT NULL	书名
Author	NVARCHAR (20)	NOT NULL	作者
Publisher	NVARCHAR (20)	NOT NULL	出版社
UnitPrice	FLOAT (53)	NULL	价格
Quantity	INT	NOT NULL	数量
CategoryID	INT	NULL, FK(Category. CategoryID)	分类号



实作1-使用SQL Server创建数据库

App_Data\ Booksdb.mdf



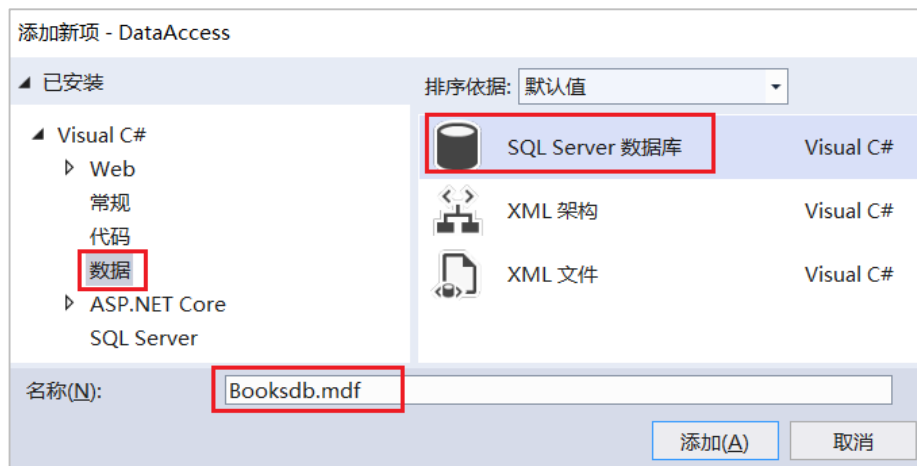
实作1-使用SQL Server创建数据库

- 实作说明：以VS内置的SQL Server创建数据库及数据表
- 实作步骤：

1. 使用方案 / 项目：LabASPNET，鼠标右击App_Data文件夹

->添加->新建项：

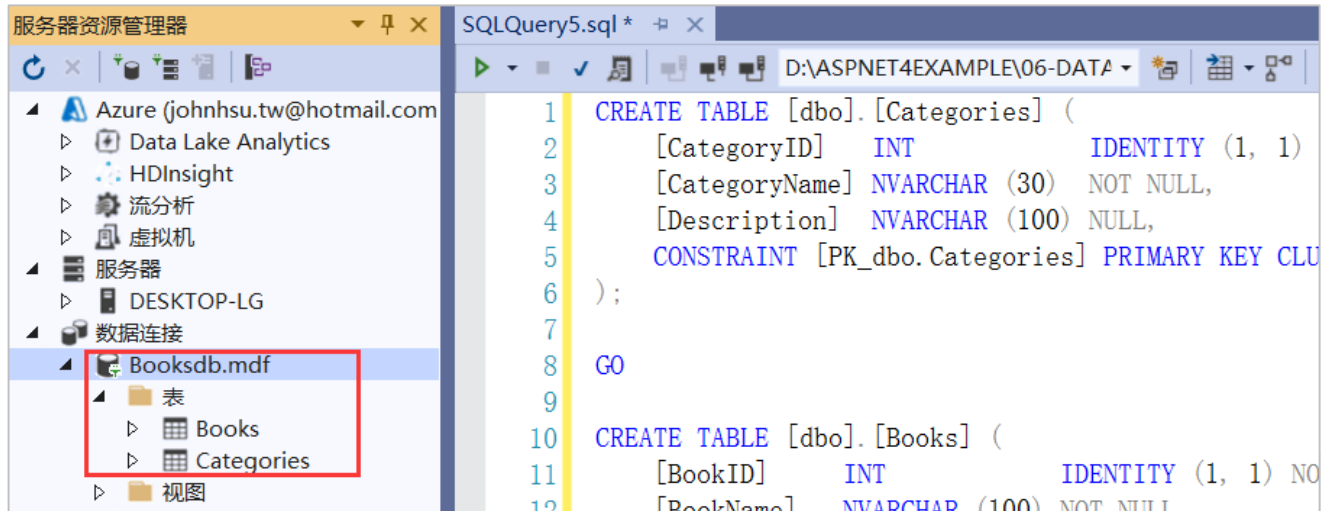
输入：Booksdb.mdf





实作1-使用SQL Server创建数据库

2. 创建Categories及Books数据表：使用教程提供的Booksdb.sql脚本创建并添加测试数据，步骤：右击Booksdb.mdf（在服务资源管理器）->新建查询，将Booksdb.sql内容贴入视窗并运行，如果如下：





实作1-使用SQL Server创建数据库

dbo.Books [数据]							
最大行数(Q): 1000							
	BookID	BookName	Author	Publisher	UnitPrice	Quantity	CategoryID
▶	1	分布式应用系...	谢文辉	人民邮电出版社	67.1	10	1
	2	从零开始构建...	张光华	机械工业出版社	61.4	10	1
	3	程序员的38堂...	宋伯远	人民邮电出版社	54.9	10	1
	4	大象如何站在...	环球科学	外语教学与研...	39.8	10	1
	5	亚马逊跨境电...	纵雨果	电子工业出版社	39.3	10	2
	6	引爆社群: 移...	唐兴通	机械工业出版社	52.6	10	2
	7	赢在B端: B2B...	刘宇航	中国纺织出版社	55.1		
	8	哲学简史	伯特兰·罗素	台海出版社	32.1		

dbo.Categories [数据]			
最大行数(Q): 1000			
	CategoryID	CategoryName	Description
▶	1	科技类	科技类图书分类
	2	经管类	经管类图书分类
	3	文学类	文学类图书分类
	4	教育类	教育类图书分类
	5	艺术类	艺术类图书分类

检视Categories及Books数据表内容:



实作2-使用EF生成实体类并运用

Source: Labs/Unit06/BookQuery.aspx



实作2-使用EF生成实体类并运用

- 实作说明：以Database First方式，在实作1基础上，使用EF从Booksdb数据库生成实体类
- 实作步骤：
 - 在项目LabASP.NET上，右击Models文件夹->添加->新建项：





实作2-使用EF生成实体类并运用

3 您的应用程序应使用哪个数据连接与数据库进行连接?(W)

Booksdb.mdf

连接字符串:

metadata=res://*/Models.BooksModel.csdl|res://*/Models.BooksModel.res://*/Models.BooksModel.msl;provider=System.Data.SqlClient;providerstring="data source=(LocalDB)\MSSQLLocalDB;attachdbfilename=|D:\Booksdb.mdf;integrated security=True;MultipleActiveResultSets=True"

☒ 将 Web.Config 中的连接设置另存为(S):

BooksdbEntities

这名代表数据库实体

< 上一步(B) 下一步(N) >

4 您要使用实体框架的哪个版本(W)?

☒ 实体框架6.x

☐ 实体框架5.0

6 右击项目名, 选“生成”才能在程序中使用模型的实体类。

5 您要在模型中包括哪些数据库对象?(W)

☒ 表

☒ dbo

☒ Books

☒ Categories

☐ 视图

☐ 存储过程和函数

☒ 确定所生成对象名称的单复数形式(S) 因我们的数据表Books及Categories都是复数,要选才能在建实体类时恢复单数

☒ 在模型中包括外键列(K)

☐ 将所选存储过程和函数导入到实体模型中(I)

模型命名空间(M):

BooksdbModel

< 上一步(B) 下一步(N) > 完成(E)

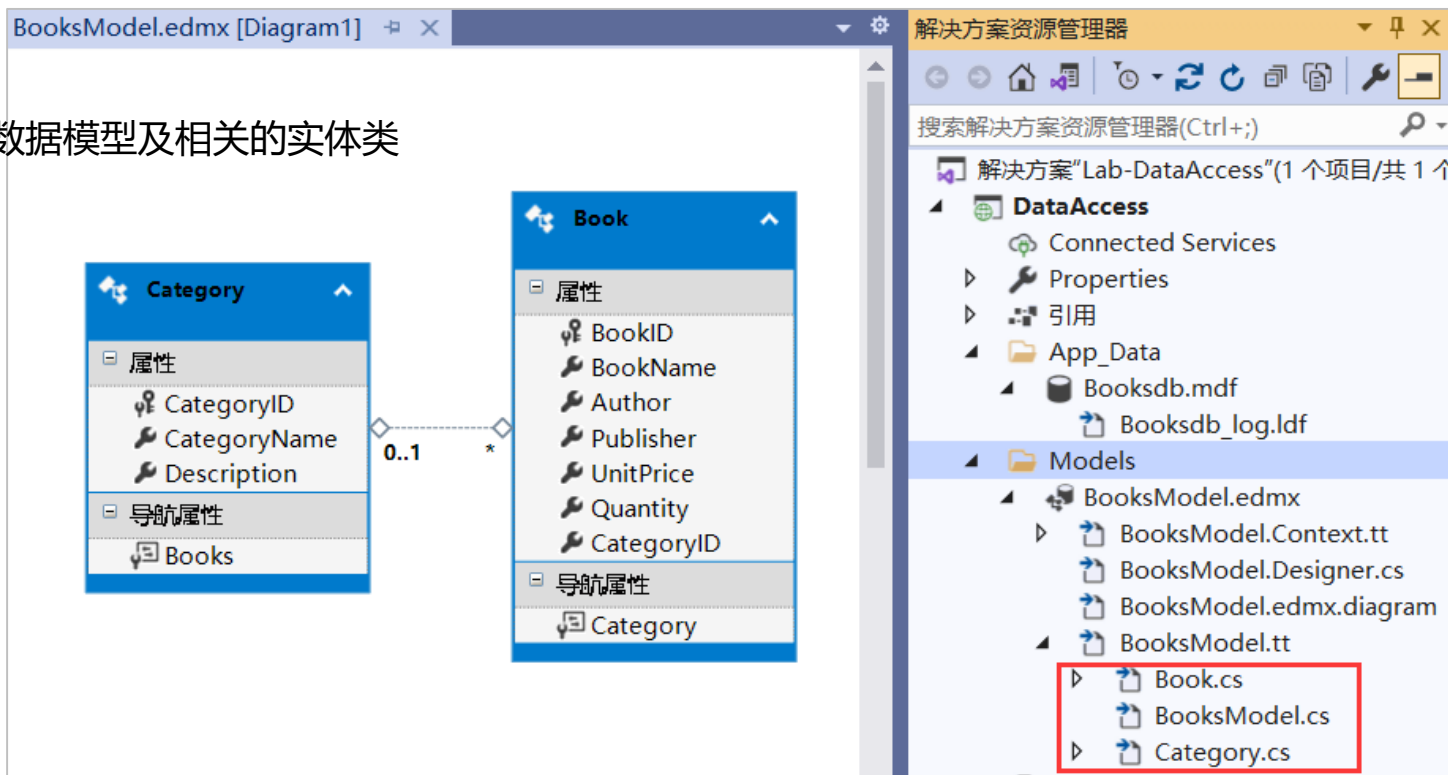
过程中需再按下3次“确定”才能完成。



实作2-使用EF生成实体类并运用

6

生成数据模型及相关的实体类





实作2-使用EF生成实体类并运用

- 使用LINQ操作实体类
进行查询图书：在
Labs下添加Unit06文
件夹，在其中添加
BookQuery.aspx

请输入书名或关键词：（不输入可列出全部书籍）

查询图书

Label

Source: Labs/Unit06/BookQuery.aspx

```
protected void btnQuery_Click() {  
    string keyword = txtBookName.Text;  
    StringBuilder sb = new StringBuilder();  
    //声明并创建数据库实体对象db  
    BooksdbEntities db = new BooksdbEntities();  
    //使用LINQ扩充方法  
    var result = db.Books.Where(b =>  
        b.BookName.Contains(keyword));  
    if (result.Count() == 0) {  
        sb.Append("查无此书!");  
    } else {  
        foreach (var b in result) {  
            sb.Append(b.BookID + " " + b.BookName + " "  
                + b.Author + "<br/>");  
        }  
    }  
    lblMessage.Text = sb.ToString();  
}
```




实作2-使用EF生成实体类并运用

localhost:52671/BookQuery.as x +

localhost:52671/BookQuery.aspx

应用 書籤 百度 QQ邮箱 JD 京东 学者网

请输入书名或关键词：（不输入可列出全部书籍）

3 经典图书：程序员的38堂成长课 宋伯远
8 经典图书：哲学简史 伯特兰·罗素
9 经典图书：津巴多普通心理学 邹智敏
13 每天读点英文经典短篇小说全集 马钟元

书名中有“经典”关键词的书籍

localhost:52671/BookQuery.as x +

localhost:52671/BookQuery.aspx

应用 書籤 百度 QQ邮箱 JD 京东 学者网

请输入书名或关键词：（不输入可列出全部书籍）

1 分布式应用系统架构设计与实践 谢文辉
2 从零开始构建深度前馈神经网络 张光华
3 经典图书：程序员的38堂成长课 宋伯远
4 大象如何站在铅笔上-超乎想象的科学解读 环球科学
5 亚马逊跨境电商运营从入门到精通 纵雨果
6 引爆社群：移动互联网时代的新4C法则（第2版） 唐兴通
7 赢在B端：B2B品牌营销增长手册 刘宇航
8 经典图书：哲学简史 伯特兰·罗素

文本框不输入则列出全部书籍



2. 使用LINQ实现数据访问



LINQ技术概述

- 微软在2007年发表了LINQ（Language-Integrated Query）数据查询技术，让开发人员用一致性的语法来查询不同的数据源。
- LINQ集成于.NET Framework中，以一致的语法实现多种数据源的查询和操作。
 - LINQ to Objects：用于操作Array和List等集合类型数据。
 - LINQ to XML：用于操作XML类型数据。
 - LINQ to DataSet：用于操作DataSet类型数据。
 - LINQ to SQL：用于操作SQL Server数据库类型数据。
 - **LINQ to Entities**用于操作实体数据模型。



LINQ的两种使用方式

◆ LINQ扩充方法

- ◇ 使用说明：比LINQ查询表达式简洁，适用于较简单的查询需求，如在不需要Join合并运算的时机。（建议：尽量使用）

- ◇ 语法：

Var 变项 = 集合.LINQ扩充方法 (Lambda表达式)

- ◇ 实例：

```
BooksdbEntities db = new BooksdbEntities();  
var result = db.Books.Where(b => b.BookName.Contains(keyword));
```

扩充方法

Lambda表达式



LINQ的两种使用方式

◆ LINQ查询表达式

- ◇ 使用说明：只要对集合对象查询都可使用，尤其是LINQ扩充方法无法完成任务的时机。语法较扩充方法复杂些，相似于SQL语法。
- ◇ 语法：

```
Var 变项 = from [数据类型] 范围变项 in 集合
            where <条件>
            orderby 属性1 [ascending][descending] [属性2[...]]
            select new {
                        [别名1=] 属性1 [ [别名2=]属性2 [...]]
                    };
```



LINQ的两种使用方式

- from: 指定查询操作的**数据源**和**范围变量**。
- where: 指定筛选操作的逻辑**条件**。
- orderby: 用来指定符合条件对象的**排序**方式。
- 以select子句指定比对后所要的查询结果；或用select new指定要取得的新对象及其字段，最后再以LINQ查询结果存到一个指定的变项中。

```
BooksdbEntities db = new BooksdbEntities();  
var result = from b in db.Books  
             where b.BookName.Contains(keyword)  
             select b;
```

最后result集合对象会包含0到多个符合条件的book对象



LINQ扩充方法

扩充方法	使 用 说 明
Average	回传结果的 平均值
Count	回传结果的 总笔数
Max	回传结果的 最大值
Min	回传结果的 最小值
Sum	回传结果的 加总值
Where	回传 指定条件 的记录
Take	回传 特定笔数 的记录
Skip	跳过 指定的笔数



LINQ扩充方法

扩充方法	使用 说 明
OrderBy	设定递增排序，须在 Take 和 Skip 前
OrderByDescending	设定递减排序，须在 Take 和 Skip 前
ThenBy	指定后续递增排序
ThenByDescending	指定后续递减排序
FirstOrDefault	回传第一笔记录，若无则回传默认值
SingleOrDefault	回传单一笔记录，若无则回传默认值
ToList	将回传结果转成 List 数据型别



扩充方法与查询表达式

- 查询score整数数组并递增排序

扩充方法

```
int [ ] score = new int [ ] {80,83,92,65,70};  
var result = score.OrderBy(s => s);
```



查询表达式

```
int [ ] score = new int [ ] {80,83,92,65,70};  
var result = from s in score  
              orderby s ascending  
              select s;
```

Source:Examples/Unit06/LinqQueryMethod.aspx



扩充方法与查询表达式

- 查询书名中有“经典”二字的图书，并以书号递增排序

扩充方法

```
var result = db.Books
    .Where(b => b.BookName.Contains("经典"))
    .OrderBy(b => b.BookID);
```



查询表达式

```
var result = from b in db.Books
    where b.BookName.Contains("经典")
    orderby b.BookID ascending
    select b;
```



实作3-使用EF+LINQ的查询运用

Source: Labs/Unit06/LinqMethodQuery.aspx



实作3-使用EF+LINQ的查询运用

- 实作说明：以LINQ扩充方法或查询表达式撰写数个查询图书的功能
- 实作步骤：
 - 在Labs\Unit06下添加LinqMethodQuery.aspx
 - 在表单上布置3个Button，每个执行一项功能并以Label显示结果。

//使用LINQ扩充方法-查单价大于50的书并排序

```
var result = db.Books
    .Where(b => b.UnitPrice > 50)
    .OrderBy(b => b.UnitPrice);
```

//使用LINQ扩充方法-图书信息统计

```
var result = db.Books;
//打印
result.Average(b => b.UnitPrice)
result.Count()
result.Max(b => b.UnitPrice)
result.Min(b => b.UnitPrice)
```

查询单价大于50元的图书并按价格

图书信息统计

图书分类统计

Label1

//使用LINQ查询表达式-图书分类统计

```
var result = from c in db.Categories
join b in db.Books
on c.CategoryID equals b.CategoryID
into num
select new {
    分类名称 = c.CategoryName,
    图书数量 = num.Count()
};
```



实作3-使用EF+LINQ的查询运用

运行结果

查询单价大于50元的图书并按价格排序			图书信息统计	图书分类
6	引爆社群：移动互联网时代的新4C法则（第2版）	52.6		
3	经典图书：程序员的38堂成长课	54.9		
7	赢在B端：B2B品牌营销增长手册	55.1		
2	从零开始构建深度前馈神经网络	61.4		
1	分布式应用系统架构设计与实践	67.1		
11	现代西班牙语2学生用书	68		
9	经典图书：津巴多普通心理学	89		
16	美国纽约摄影学院摄影教材	92.8		

LinqMethodQuery.aspx

查询单价大于50元的图书并按价格排序	图书信息统计	图书分类统计
平均单价：51.725 图书数量：16 单价最高：92.8 单价最低：25.2		

查询单价大于50元的图书并按价格排序	图书信息统计	图书分类统计
科技类 类别共有 4 种图书 经管类 类别共有 3 种图书 文学类 类别共有 3 种图书 教育类 类别共有 3 种图书 艺术类 类别共有 3 种图书		



实作4-EF+LINQ的增删改

Source: Labs/Unit06/LinqManageData.aspx



实作4-EF+LINQ的增删改

- 实作说明：以LINQ扩充方法或查询表达式撰写图书信息的-添加/删除/修改-的功能
- 实作步骤：
 - 在Labs\Unit06下添加**LinqManageData.aspx**
 - 在表单上布置3个TextBox, 3个Button，每个执行一项功能并以Label显示结果。

书籍分类信息的CRUD处理

分类编号:

分类名称:

分类说明:

目前分类信息如下:

Label



实作4-EF+LINQ的增删改

主要代码如下：

添加分类

```
//添加1个分类信息
Category c = new Category();
c.CategoryName = txtCategoryName.Text;
c.Description = txtDescription.Text;
db.Categories.Add(c);
db.SaveChanges();
```

```
//使用LINQ扩充方法找到那笔数据并删除
var result = db.Categories.Where(c => c.CategoryID
                                ==cid).FirstOrDefault();
if (result != null) {
    db.Categories.Remove(result);
    db.SaveChanges();
}
```

删除分类

```
//使用LINQ扩充方法找到那笔数据并修改
var result = db.Categories.Where(c => c.CategoryID==cid).FirstOrDefault();
if (result != null) {
    result.CategoryName = txtCategoryName.Text;
    result.Description = txtDescription.Text;
    db.SaveChanges();
}
```

修改分类

Source: Labs/Unit06/LinqManageData.aspx



实作4-EF+LINQ的增删改

添加

localhost:52671/LinqManageD x +

← → ↻ 🏠 localhost:52671/Linq

应用 書籤 百度 QQ邮箱 JD

书籍分类信息的CRUD处理

分类编号:

分类名称:

分类说明:

添加

修改

删除

目前分类信息如下:

1 科技类 科技类图书分类
2 经管类 经管类图书分类
3 文学类 文学类图书分类
4 教育类 教育类图书分类
5 艺术类 艺术类图书分类
6 测试类 测试类测试类测试类

修改

书籍分类信息的CRUD处理

分类编号:

分类名称:

分类说明:

添加

修改

删除

目前分类信息如下:

1 科技类 科技类图书分类
2 经管类 经管类图书分类
3 文学类 文学类图书分类
4 教育类 教育类图书分类
5 艺术类 艺术类图书分类
6 其它类 其它类图书分类项

删除

书籍分类信息的CRUD处理

分类编号:

分类名称:

分类说明:

添加

修改

删除

目前分类信息如下:

1 科技类 科技类图书分类
2 经管类 经管类图书分类
3 文学类 文学类图书分类
4 教育类 教育类图书分类
5 艺术类 艺术类图书分类
6 其它类 其它类图书分类项

Source: Labs/Unit06/LinqManageData.aspx



实作4-EF+LINQ的增删改

书籍分类信息的CRUD处理

分类编号:

分类名称:

分类说明:

删除分类

目前分类信息如下:

1 科技类 科技类图书分类
2 经管类 经管类图书分类
3 文学类 文学类图书分类
4 教育类 教育类图书分类
5 艺术类 艺术类图书分类
6 其它类 其它类图书分类项

书籍分类信息的CRUD处理

分类编号:

分类名称:

分类说明:

目前分类信息如下:

1 科技类 科技类图书分类
2 经管类 经管类图书分类
3 文学类 文学类图书分类
4 教育类 教育类图书分类
5 艺术类 艺术类图书分类



单元小结

- 本单元对ASP.NET的几种数据访问技术，包含传统的ADO.NET方法、LINQ to Entity方法及LINQ to SQL三种方法作一些简介。
- 在LINQ to Entity方法中我们采用了Database First模式，将图书数据库透过Entity Framework工具转为模型及实体类，以支持后续的CRUD实作。
- LINQ技术在配合Entity Framework时，可使用LINQ扩充方法或LINQ查询表达式去编写代码。