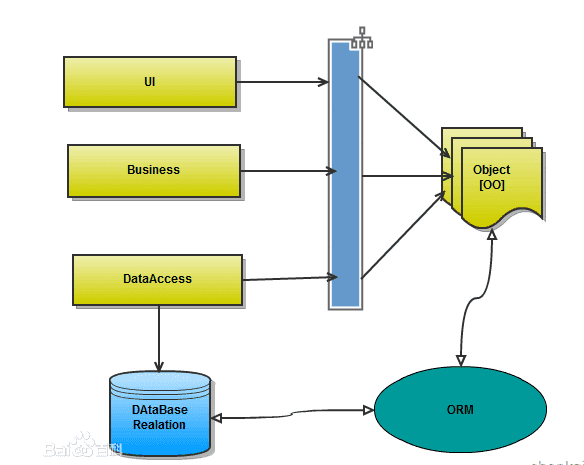
# 数据处理

## 数据访问层ORM框架之EntityFramework

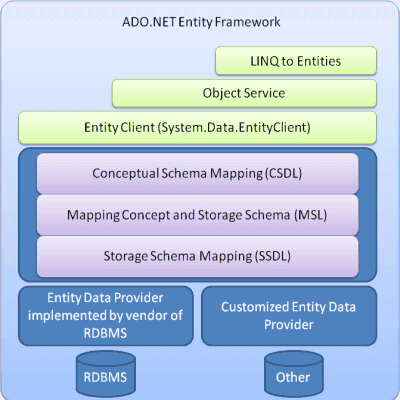
### 什么是 ORM ？



ORM => 对象关系映射（英语：Object Relational Mapping）对于O/R，即 Object（对象）和 Relational（关系型数据），表示必须同时使用面向对象和关系型数据进行开发。

ORM框架 用途 => 为了解决 软件项目 与 数据库 打交道的中间层。

### 什么是EntityFramework？



**ADO.NET Entity Framework微软用来替代ADO.NET的一套 ORM 框架.**

以 Entity Data Model (EDM) 为主，将数据逻辑层切分为三块，分别为 Conceptual Schema, Mapping Schema 与 Storage Schema 三层，其上还有 Entity Client，Object Context 以及 LINQ 可以使用。

ADO.NET： 一般使用SQLHelper直接来操作数据库。

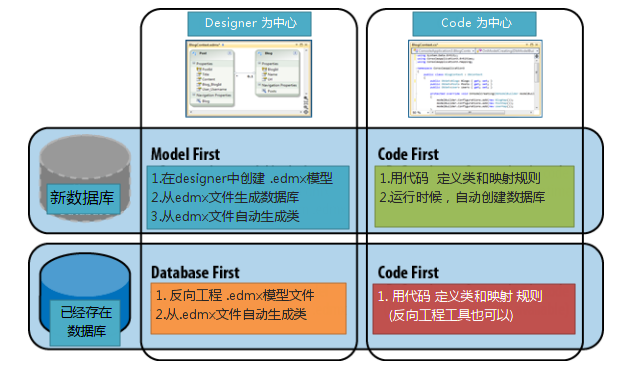
### EF 与 ADO.NET 的关系

EF首先生成sql,再调用ado.net访问数据库,最后使结果对象具体化.

### 有哪些 ORM 产品？

* NHibernate框架 来源于 Java 的 Hibernate 框架，采用XML文件配置的方式。
* Castle ActiveRecord 框架 是 Castle 中的一个子项目，底层封装了 NHibernate,改用 Attribute 来代替配置文件
* iBATIS.NET 框架 分为 DataMapper 和DataAccess两部分，DataMapper是这个框架的核心， DataMapper使用XML文件实现从实体到 SQL statements 的映射，学习起来非常简单，使用DataMapper 以后，我们可以自由的使用 SQL 语句或存储过程; DataMapper 允许我们通过一个简单的接口来操作数据，而不必了解底层实现的细节。
* ADO.NET Entity Framework框架 与Visual Studio SQLServer软件无缝集成

### EF的开发模式



## EF—CodeFirst模式实现CRUD

增加(Create)、读取查询(Retrieve)、更新(Update)和删除(Delete)

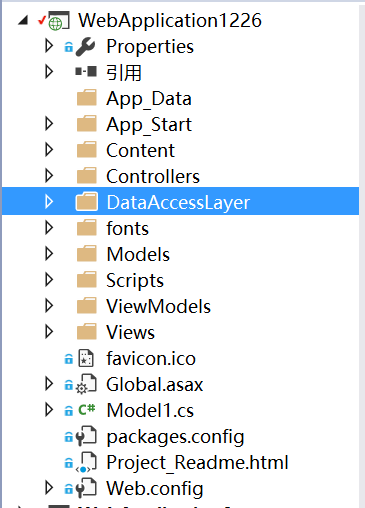
### 实验8 在项目中加入EF数据访问层

**步骤1 . 创建数据库**

连接SQL SERVER ,创建数据库 “SalesERPDB”

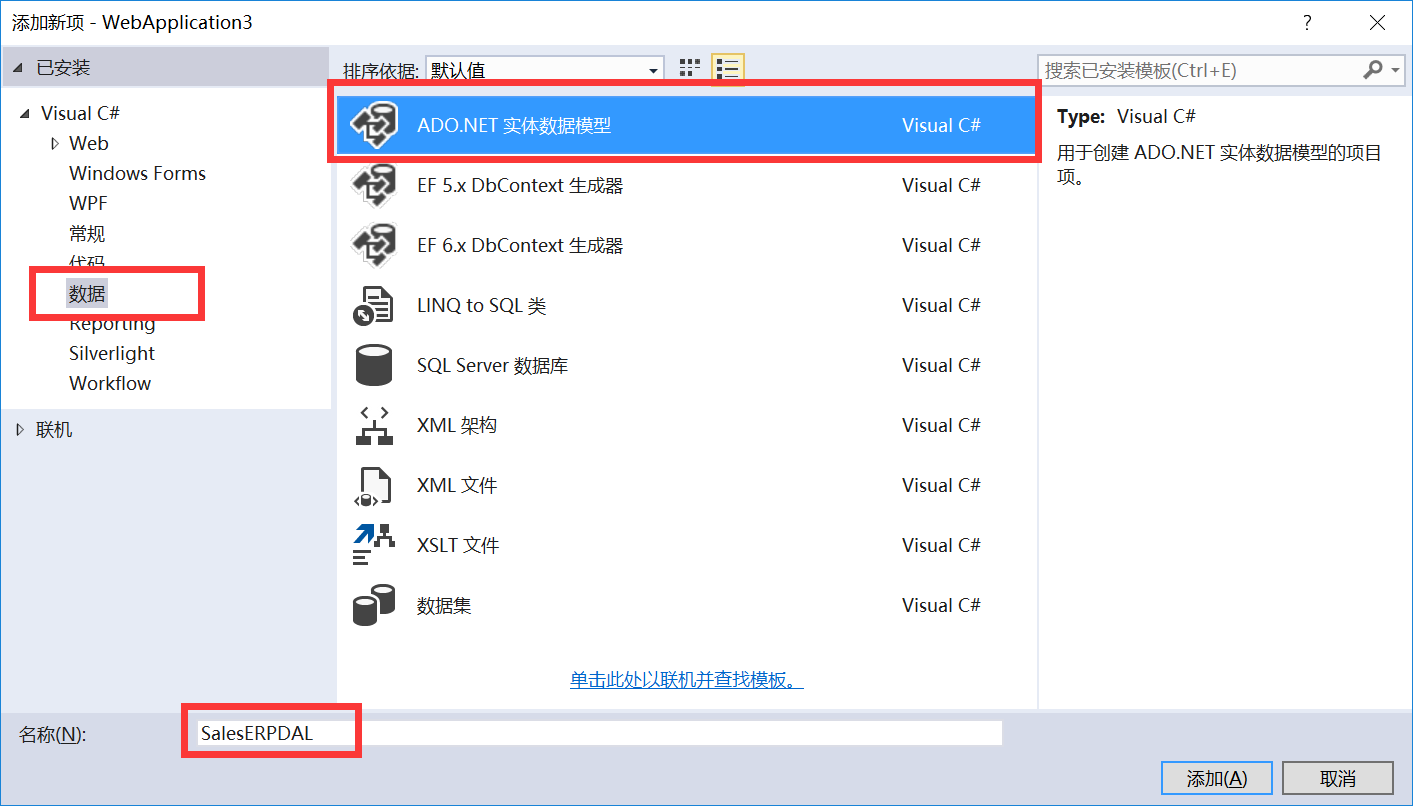
|  |
| --- |
|  |

**步骤2. 在项目的根部创建数据访问层文件夹**DataAccessLayer

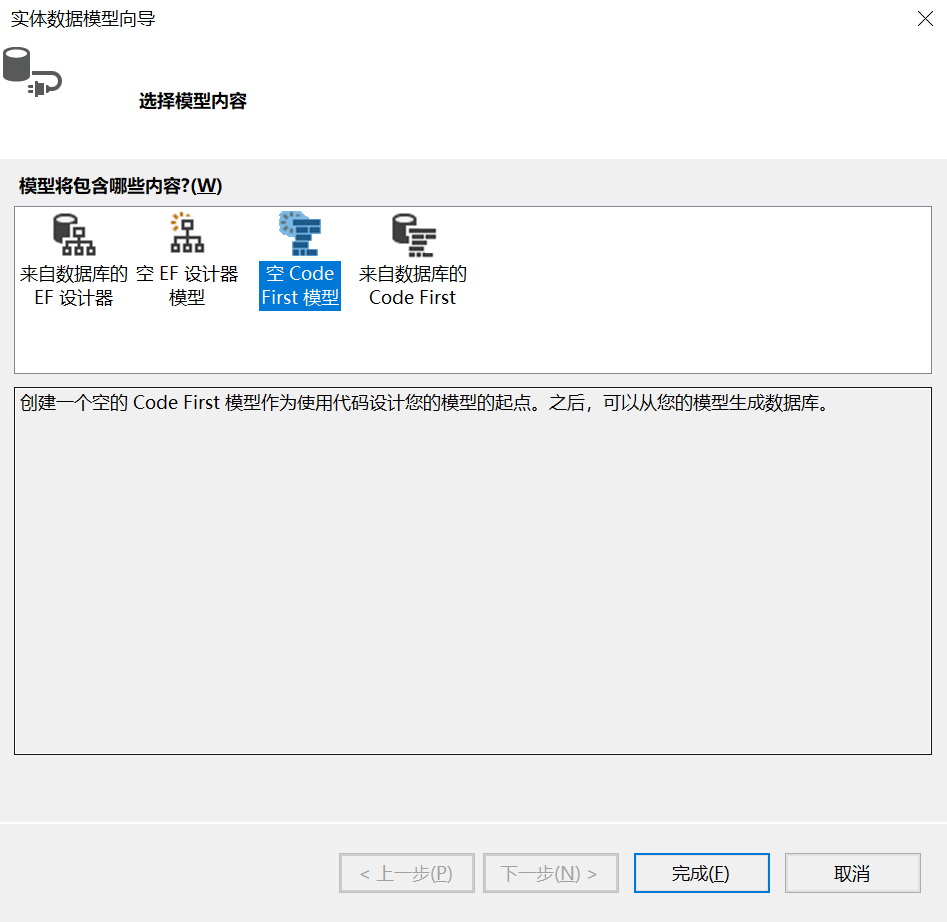


**步骤3 通过EF工具创建数据管理上下文SalesERPDAL**

（1）在DataAccessLayer文件夹上右键单击添加新建项，选择数据类别-》ADO.NET实体数据模型



（2）选择CodeFirst模型



经过（2）之后，项目文档中添加了以下内容

* EF引用

|  |
| --- |
|  |

* SalesERPDAL**类**

|  |
| --- |
|  |

* 默认连接字符串（web.config）文件中

|  |
| --- |
|  |

**步骤4 修改连接字符串**

|  |
| --- |
| <add name="SalesERPDAL" connectionString="server=.;uid=sa;pwd=123;database=TestDB;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework" providerName="System.Data.SqlClient" /> |

**步骤5创建Employee类的主键**

（1）添加数据验证命名空间System.ComponentModel.DataAnnotations

（2）为Employee类添加主键

|  |
| --- |
|  |

**步骤6 在SalesERPDAL类中加入员工数据集，以及相关的表格重命名**

|  |
| --- |
|  |

注意：上述代码中提到“TblEmployee”是表名称，是运行时自动生成的。

扩展阅读：OnModelCreating

https://docs.microsoft.com/zh-cn/previous-versions/visualstudio/gg679267(v=vs.113)

**步骤7 改变业务层代码，并从数据库中获取数据**

|  |
| --- |
|  |

运行并测试

|  |
| --- |
|  |

**步骤8向数据库中增加数据**

|  |
| --- |
|  |

**运行并测试**

|  |
| --- |
|  |

**步骤9 重构Employee控制器代码**

**通过重构方法的形式，增加index—行为的可读性**

|  |
| --- |
|  |

**关于实验8**

**什么是DbSet?**

DbSet数据集是数据库方面的概念 ，指数据库中可以查询的实体的集合。当执行Linq 查询时，Dbset对象能够将查询内部转换为针对数据库的操作。

在本实例中，DbSet是Employees，是所有Employee的实体集合。当每次需要访问Employees时，会获取“TblEmployee”的所有记录，并转换为Employee对象，返回Employee对象集。

**如何连接数据访问层和数据库？**

数据访问层和数据库之间的映射通过名称实现的，在实验8中，ConnectionString（连接字符串）的名称和数据访问层的类名称是相同的，都是SalesERPDAL，因此会自动实现映射。

连接字符串的名称可以改变吗？

可以改变，在实验8中，可以在数据访问层中定义构造函数，使用新名字

public SalesERPDAL():base("NewName")

{

}

### 实验9 新增数据实体

**步骤1 在Employee控制器中创建一个AddNew行为**

|  |
| --- |
|  |

步骤2 创建AddNew视图

|  |
| --- |
|  |

步骤3在Employee Index视图中创建一个新增超链接

|  |
| --- |
|  |

测试程序

从employee/index中点击超链接，进入新增页面

|  |
| --- |
|  |

**关于实验9**

**（1）使用Form 标签的作用是什么？**

Web编程模式不是事件驱动的编程模式，是请求响应模式。Form标签是HTML中产生请求的一种方式，Form标签内部的提交按钮只要一被点击，请求会被发送到相关的action 属性。

**（2）Form标签中方法属性是什么？**

方法属性决定了请求类型。有四种请求类型：get，post，put以及delete.

* Get： 当需要获取数据时使用。
* Post： 当需要新建一些事物时使用。
* Put: 当需要更新数据时使用。
* Delete：需要删除数据时使用。

**（3）使用Form 标签来生成请求，与通过浏览器地址栏或超链接来生成请求，有什么区别？**

使用Form标签生成请求时，所有有关输入的控件值会随着请求一起发送。

**（4）输入的值是怎样发送到服务器端的？**

当请求类型是Get，Put或Delete时，值会通过查询语句发送，当请求是Post类型，值会通过Post数据传送。

**（5）使用输入控件名的作用是什么？**

所有输入控件的值将随着请求一起发送。同一时间可能会接收到多个值，为了区分发送到所有值为每个值附加一个Key，这个Key在这里就是name属性。

**（6）name和 id的作用是否相同？**

不相同，名称属性是HTML内部使用的，当请求被发送时，然而 id属性是在JavaScript中开发人员为了实现一些动态功能而调用的。

**（7）“input type=submit” 和 “input type=button”的区别是什么？**

**submit**按钮在给服务器发送请求而专门使用的，而简单的**button**按钮是执行一些自定义的客户端行为而使用的。按钮不会自己做任何事情。

### 实验10 在服务器端获取Post数据

**步骤1在Employee控制器中新增一个SaveEmployee 方法获取Post数据**

|  |
| --- |
|  |

测试程序，新增之后获取结果

|  |
| --- |
|  |
|  |

**关于实验10**

**action 方法内部的Textbox 值是如何更新 Employee 对象的？**

在 Asp.Net MVC中有个 Model Binder的概念：

当发出的请求由带参的action方法执行，Model Binder将自动执行。Model Binder从HTTP请求中提取相应的数据，并以此来生成这些参数。这个过程就是Model的绑定技术

**如果两个参数是相关联的会发生什么状况，如参数”Employee e“和 “string Name”？**

Name变量和 e.FirstName 属性会同时获得值。

**Model Binder 是否可以用于类的组合关系？**

可以

（1）在Models文件夹中新增地址类Address

|  |
| --- |
|  |

（2）修改顾客类的地址属性类型

|  |
| --- |
|  |

(3) 如果希望修改顾客的信息，则相关表单项可以这么写

|  |
| --- |
|  |

### 实验11重置和取消按钮

**步骤1在新增视图中添加消按钮**

|  |
| --- |
|  |

注意，此时取消按钮和保存按钮是都是submit类型，且name属性都是BtnSubmit

**步骤2重写Employee控制器类SaveEmployee方法**

|  |
| --- |
|  |

**测试—保存**

|  |
| --- |
|  |

**测试重置**

|  |
| --- |
|  |

**测试取消**

|  |
| --- |
|  |

**关于实验11**

**在实验11中为什么将保存和取消按钮设置为同名？**

在日常使用中，保存/取消按钮之后，请求会被发送到服务器端，所有输入控件的值都将被发送。保存/取消也是输入按钮的一种。因此保存按钮和取消按钮的值也会被发送。

当保存按钮被点击时，保存按钮的值也会随着请求被发送到服务器端，当点击取消按钮时，取消按钮的值”取消“会随着请求发送。

在Action 方法中，Model Binder 将维护这些工作。会根据接收到的值更新参数值。

实现多重提交按钮有没有其他可用的方法？

可以通过前端技术实现多重提交按钮，如单击事件

|  |
| --- |
|  |

如果控件名称与类属性名称不匹配会发生什么情况？

（1）可以通过Request.Form["名称"]获得值

（2）可以通过设计同名参数获得值

（3）可以创建自定义 Model Binder ，代替默认的Model Binder

**RedirectToAction 函数的功能？**

RedirectToAction 生成 RedirectToRouteResult 如ViewResult 和 ContentResult，RedirectToRouteResult是 ActionResult的孩子节点，表示间接响应，当浏览器接收到RedirectToRouteResult，新Action 方法产生新的请求。

EmptyResult是什么？

是ActionResult的一个孩子节点，当浏览器接收到 EmptyResult，作为响应，它会显示空白屏幕，表示无结果。在本实验中不会发生EmptyResult。

### 实验12在数据库中保存记录和更新表格

**步骤1在**EmployeeBusinessLayer  **中创建 SaveEmployee方法**

|  |
| --- |
|  |

**步骤2 在Employee控制器saveEmployee行为中修改内容**

|  |
| --- |
|  |

**并测试增加过程查看结果**



## 实验13 添加服务器端校验

**步骤1修改Employee类设置服务器端校验参数**

名字不能为空

字符串长度必须大于2

整数可以为空

|  |
| --- |
|  |

步骤2 修改**SaveEmployee Action 方法**

如果输入有效加入数据库，否则回到新增视图

|  |
| --- |
|  |

步骤3在View中显示错误，修改 Views/Index/CreateEmployee.cshtml 代码

|  |
| --- |
|  |

步骤4 支持修改MODEL

在Global.asax文件中在 Application\_Start后添加以下语句

|  |
| --- |
| Database.SetInitializer(new DropCreateDatabaseIfModelChanges<SalesERPDAL>()); |

## 实验14 自定义服务器端校验

新增一个校验类，并设置给具体的属性

|  |
| --- |
|  |

测试看效果