座號： 19 姓名： 柯昱廷

階段性作六

Code-First Model實作練習

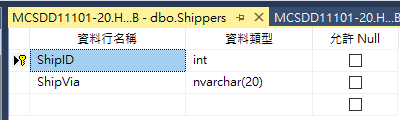
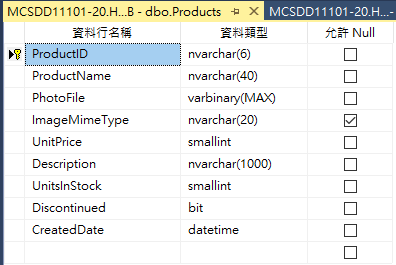
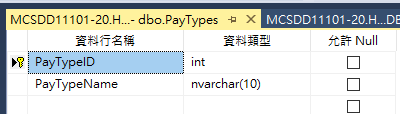
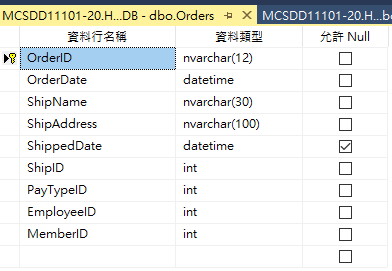
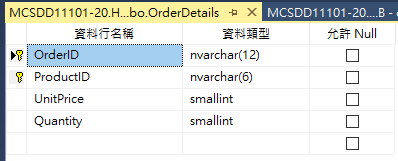
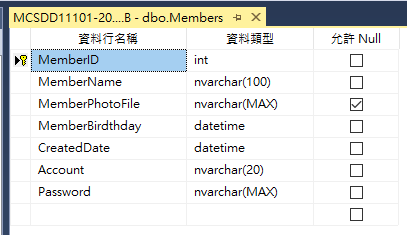
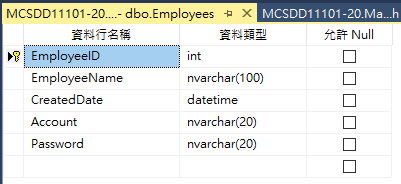
* 請參考附件當中的實體關聯圖(Entity-Relationship Diagram)，利用Entity Framework的Code-First方式建立適當的模型與資料庫。
  1. 本專案名稱請命名為「HW7Project」。
  2. 參考實體關聯圖建立模型中相對應的類別。
  3. 在類別中設計相對應的屬性及適當的屬性資料型態(Data Type)。
  4. 請建立適當的驗證器。
  5. 請建立類別之間的關聯(資料表的關聯)。
  6. 所有的設計內容必須是合理的資料存儲內容與貼近實務的設計。
  7. 請執行Code-First模型以建立資料庫。

**繳交作業時，請以FTP上傳程式原始檔與本文件檔各一份，原始程式碼請將整個方案壓縮成一個檔，壓縮檔案名稱為「MyProject○○」，其中○○為自己的座號，並放置於「完成結果檔」資料夾內。**

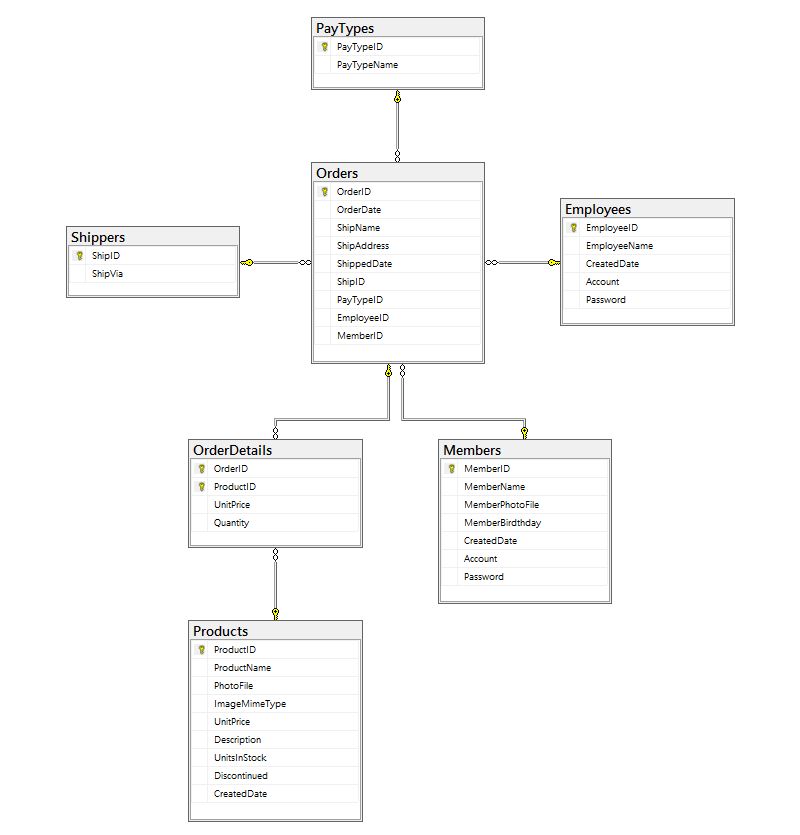
* ※以上各項均須完成始得通過※

資料庫部份：

所有資料表的設計畫面(截圖)



資料表關聯圖(截圖)



程式碼部份：

所有的資料模型類別：

Employees:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel;

using System.ComponentModel.DataAnnotations;

namespace HW7Project.Models

{

public class Employees

{

//這個動作叫做封裝

[Key] //資料庫就會把它設為PK

public int EmployeeID { get; set; }

[Required(ErrorMessage = "必填欄位")]

[StringLength(100, ErrorMessage = "長度不超過100字元")]

[DisplayName("員工姓名")]

public string EmployeeName { get; set; }

[DisplayName("建立日期")]

[DataType(DataType.Date)] //這是給View看的

[DisplayFormat(DataFormatString ="{0:yyyy-MM-dd}",ApplyFormatInEditMode =true)] //設定日期格式預設為2022/7/18 09:12:00，ApplyFormatInEditMode一併套用至所有區域

public DateTime CreatedDate { get; set; }

[Required(ErrorMessage = "必填欄位")]

[StringLength(20, ErrorMessage = "長度不超過20字元")]

[RegularExpression("[A-Za-z][A-Za-z0-9]{4,19}")]//最少4次，最多19次

[DisplayName("帳號")]

public string Account { get; set; }

[Required(ErrorMessage = "必填欄位")]

[MaxLength(20,ErrorMessage ="最大長度為20字元")]

[MinLength(8, ErrorMessage = "最小長度為8字元")]

[DataType(DataType.Password)]

[DisplayName("密碼")]

public string Password { get; set; }

}

}

Members:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel;

using System.ComponentModel.DataAnnotations;

using System.Security.Cryptography;

namespace HW7Project.Models

{

public class Members

{

//若有ViewModel，就只要寫跟資料庫有關的驗證就好了。

[Key]

public int MemberID { get; set; }

[DisplayName("姓名")]

[StringLength(100)]

[Required]

public string MemberName { get; set; }

[DisplayName("照片")]

[MaxLength]

public string MemberPhotoFile { get; set; }

[DisplayName("生日")]

[Required]

public DateTime MemberBirdthday { get; set; }

[DisplayName("建立日期")]

[Required]

public DateTime CreatedDate { get; set; }

[DisplayName("帳號")]

[Required]

[StringLength(20)]

public string Account { get; set; }

//\*\*\*封裝:{get;set;}

string password; //定義一個password的field，欄位

[Required]

[DataType(DataType.Password)] //view上就會把密碼蓋起來

[DisplayName("密碼")]

public string Password //屬性

{

//讀取

get

{

return password; //這個是string password;

}

//寫入

set

{

//\*\*\*密碼的雜湊，用某種運算方式轉成暗碼

//為甚麼要做雜湊

//密碼是string，進到資料庫後可以看到所有人密碼

byte[] hashValue;

string result = "";

System.Text.UnicodeEncoding ue = new System.Text.UnicodeEncoding();

byte[] pwBytes = ue.GetBytes(value);

SHA256 shHash = SHA256.Create();

hashValue = shHash.ComputeHash(pwBytes);

foreach (byte b in hashValue)

{

result += b.ToString();

}

password = result;

}

}

//新增會員的確認密碼怎麼做

//1.直接在View上增加一個欄位，在前端驗證

//2.建立ViweModel並新增一個確認密碼的屬性(實務上都這麼做)

}

}

OrderDetails:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace HW7Project.Models

{

public class OrderDetails

{

[Key]

[DisplayName("訂單編號")]

[Column(Order = 1)]

public string OrderID { get; set; }

[Key]

[DisplayName("商品編號")]

[Column(Order = 2)]

public string ProductID { get; set; }

[DisplayName("商品單價")]

[Required(ErrorMessage = "請輸入商品單價")]

[Range(0, short.MaxValue, ErrorMessage = "單價不可小於0")]

public short UnitPrice { get; set; }

[DisplayName("數量")]

[Required(ErrorMessage = "請輸入商品數量")]

[Range(0, short.MaxValue, ErrorMessage = "數量不可小於0")]

public short Quantity { get; set; }

//拉關聯

public virtual Orders Order { get; set; }

public virtual Products Product { get; set; }

}

}

Orders:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel;

using System.ComponentModel.DataAnnotations;

namespace HW7Project.Models

{

public class Orders

{

[Key]

[DisplayName("訂單編號")]

[StringLength(12)]

public string OrderID { get; set; }

[DisplayName("訂單成立時間")]

[DataType(DataType.DateTime)]

public DateTime OrderDate { get; set; }

[DisplayName("收貨人姓名")]

[Required(ErrorMessage = "請填寫收貨人姓名")]

[StringLength(30, ErrorMessage = "收貨人姓名最多30個字")]

public string ShipName { get; set; }

[DisplayName("收貨人地址")]

[Required(ErrorMessage = "請填寫收貨人地址")]

[StringLength(100, ErrorMessage = "收貨人地址最多100個字")]

public string ShipAddress { get; set; }

[DisplayName("出貨日")]

[DataType(DataType.Date)]

[DisplayFormat(DataFormatString = "{0:yyyy/MM/dd}", ApplyFormatInEditMode = true)]

public Nullable<DateTime> ShippedDate { get; set; }

//FK，已經再別的Model驗證過了

public int ShipID { get; set; }

public int PayTypeID { get; set; }

public int EmployeeID { get; set; }

public int MemberID { get; set; }

//拉關聯

public virtual Shippers Shipper { get; set; }

public virtual PayTypes PayType { get; set; }

public virtual Employees Employee { get; set; }

public virtual Members Member { get; set; }

}

}

PayTypes:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel;

using System.ComponentModel.DataAnnotations;

namespace HW7Project.Models

{

public class PayTypes

{

[Key]

public int PayTypeID { get; set; }

[DisplayName("付款方式")]

[Required(ErrorMessage = "必填欄位")]

[StringLength(10, ErrorMessage = "長度不超過10字元")]

public string PayTypeName { get; set; }

}

}

Products:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel;

using System.ComponentModel.DataAnnotations;

using System.Web.Mvc;

namespace HW7Project.Models

{

public class Products

{

[Key]

[DisplayName("商品編號")]

[StringLength(6)]

[RegularExpression("[A-F][0-9]{5}")]

public string ProductID { get; set; }

[DisplayName("商品名稱")]

[Required(ErrorMessage = "請輸入商品名稱")]

[StringLength(40, ErrorMessage = "商品名稱最多40個字")]

public string ProductName { get; set; }

//\*\*\*照片

[DisplayName("商品照片")]

[Required(ErrorMessage = "請上傳商品照片")]

[MaxLength]

public byte[] PhotoFile { get; set; }

//HiddenInput隱藏欄位

[HiddenInput(DisplayValue = false)]

[StringLength(20)]

public string ImageMimeType { get; set; }

[DisplayName("商品單價")]

[Required(ErrorMessage = "請輸入商品單價")]

[Range(0, short.MaxValue, ErrorMessage = "單價不可小於0")]

public short UnitPrice { get; set; }

[DisplayName("商品說明")]

[Required(ErrorMessage = "請輸入商品說明")]

[StringLength(1000, ErrorMessage = "商品介紹不得超過1000字")]

public string Description { get; set; }

[DisplayName("庫存量")]

[Required(ErrorMessage = "請輸入庫存量")]

[Range(0, short.MaxValue, ErrorMessage = "庫存量不可小於0")]

public short UnitsInStock { get; set; }

[DisplayName("是否在架上")]

[DefaultValue(false)]

public bool Discontinued { get; set; }

[DisplayName("建立日期")]

[DataType(DataType.Date)]

[DisplayFormat(DataFormatString = "{0:yyyy/MM/dd}", ApplyFormatInEditMode = true)]

public DateTime CreatedDate { get; set; }

}

}

Shippers:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel;

using System.ComponentModel.DataAnnotations;

namespace HW7Project.Models

{

public class Shippers

{

[Key]

public int ShipID { get; set; }

[DisplayName("運送方式")]

[Required(ErrorMessage = "請輸入運送方式")]

[StringLength(20, ErrorMessage = "運送方式最多20個字")]

public string ShipVia { get; set; }

}

}

有關Data Access Layer的所有類別：

HW7ProjectContext:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Data.Entity;

namespace HW7Project.Models

{

//1.使用NUGET裝EntityFramework

public class HW7ProjectContext:DbContext

{

//3.建立建構子，及修改WebConfig

public HW7ProjectContext() : base("name=HW7ProjectConnection")

{ }

//2.填資料表，若有ViewMode不用加進來

public DbSet<Employees> Employees { get; set; } //一個資料表

public DbSet<Members> Members { get; set; }

public DbSet<Orders> Orders { get; set; }

public DbSet<OrderDetails> OrderDetails { get; set; }

public DbSet<PayTypes> PayTypes { get; set; }

public DbSet<Products> Products { get; set; }

public DbSet<Shippers> Shippers { get; set; }

public System.Data.Entity.DbSet<HW7Project.Models.VMMember> VMMembers { get; set; }

}

}

DBInitializer:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Data.Entity;

namespace HW7Project.Models

{

public class DBInitializer : DropCreateDatabaseAlways<HW7ProjectContext>

{

protected override void Seed(HW7ProjectContext context)

{

base.Seed(context);

////////////////////////////////////////////

///

List<Members> members = new List<Members>

{

new Members

{

MemberName="莊孝為",

MemberBirdthday=Convert.ToDateTime("1981/7/9"),

CreatedDate=DateTime.Today,

Account="shaio",

Password= "12345678"

}

};

members.ForEach(s => context.Members.Add(s));

context.SaveChanges();

List<Shippers> shippers = new List<Shippers>

{

new Shippers

{

ShipVia="到店取貨"

},

new Shippers

{

ShipVia="宅配到府"

},

new Shippers

{

ShipVia="郵寄"

}

};

shippers.ForEach(s => context.Shippers.Add(s));

context.SaveChanges();

List<PayTypes> payType = new List<PayTypes>

{

new PayTypes

{

PayTypeName="信用卡"

},

new PayTypes

{

PayTypeName="Line Pay"

},

new PayTypes

{

PayTypeName="貨到付款"

},

new PayTypes

{

PayTypeName="到店取貨付款"

}

};

payType.ForEach(s => context.PayTypes.Add(s));

context.SaveChanges();

List<Employees> employees = new List<Employees>

{

new Employees

{

EmployeeName="王立弘",

CreatedDate=DateTime.Today,

Account="admin",

Password="P@ssw0rd"

}

};

employees.ForEach(s => context.Employees.Add(s));

context.SaveChanges();

}

}

}

其他(若有才需貼，並請標示檔名)

附件 Entity-Relationship Diagram

