DDIP Documentation

Website–, Databases, silverlight and File Handlers

Niall Ferguson

city of glagow college |

2013

Table of Contents

[Databases 3](#_Toc358121044)

[SHOP.mdf 3](#_Toc358121045)

[Diagram 3](#_Toc358121046)

[3](#_Toc358121047)

[Tables 3](#_Toc358121048)

[EngineerServices.mdf 5](#_Toc358121049)

[Diagram 5](#_Toc358121050)

[5](#_Toc358121051)

[Tables 5](#_Toc358121052)

[5](#_Toc358121053)

[6](#_Toc358121054)

[ASPNETDB.MDF 7](#_Toc358121055)

[Diagram 7](#_Toc358121056)

[LINQ-to-SQL 8](#_Toc358121057)

[Shop.dbml 8](#_Toc358121058)

[Diagram 8](#_Toc358121059)

[Shop.designer.cs 8](#_Toc358121060)

[Stored Procedures 26](#_Toc358121061)

[Engineer.dbml 28](#_Toc358121062)

[Diagram 28](#_Toc358121063)

[Engineer.designer.cs 28](#_Toc358121064)

[Web.config 41](#_Toc358121065)

# Databases

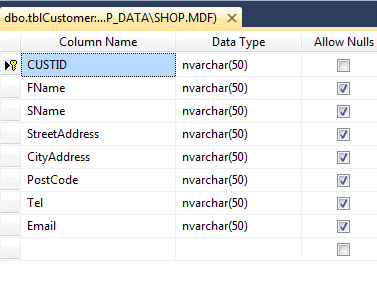
## SHOP.mdf

### Diagram

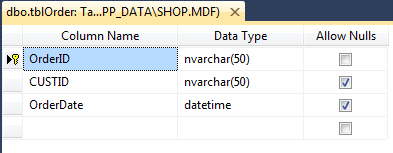
## 

### Tables

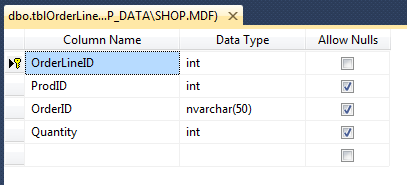
#### Customer



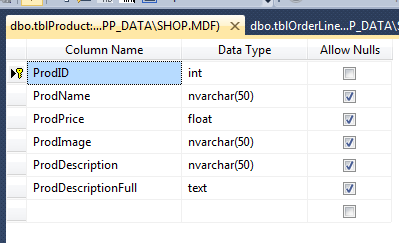
#### Order



#### Order Lines



#### Product



## EngineerServices.mdf

## Diagram

## 

## Tables

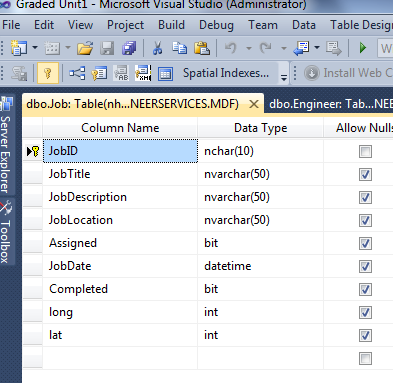
#### Completed Job

## 

#### Engineer

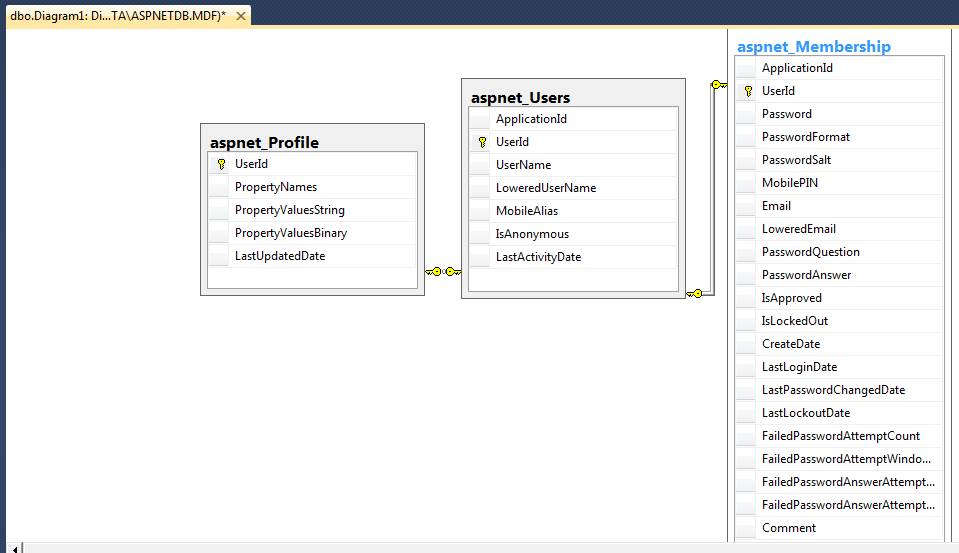
## 

#### Job



## ASPNETDB.MDF

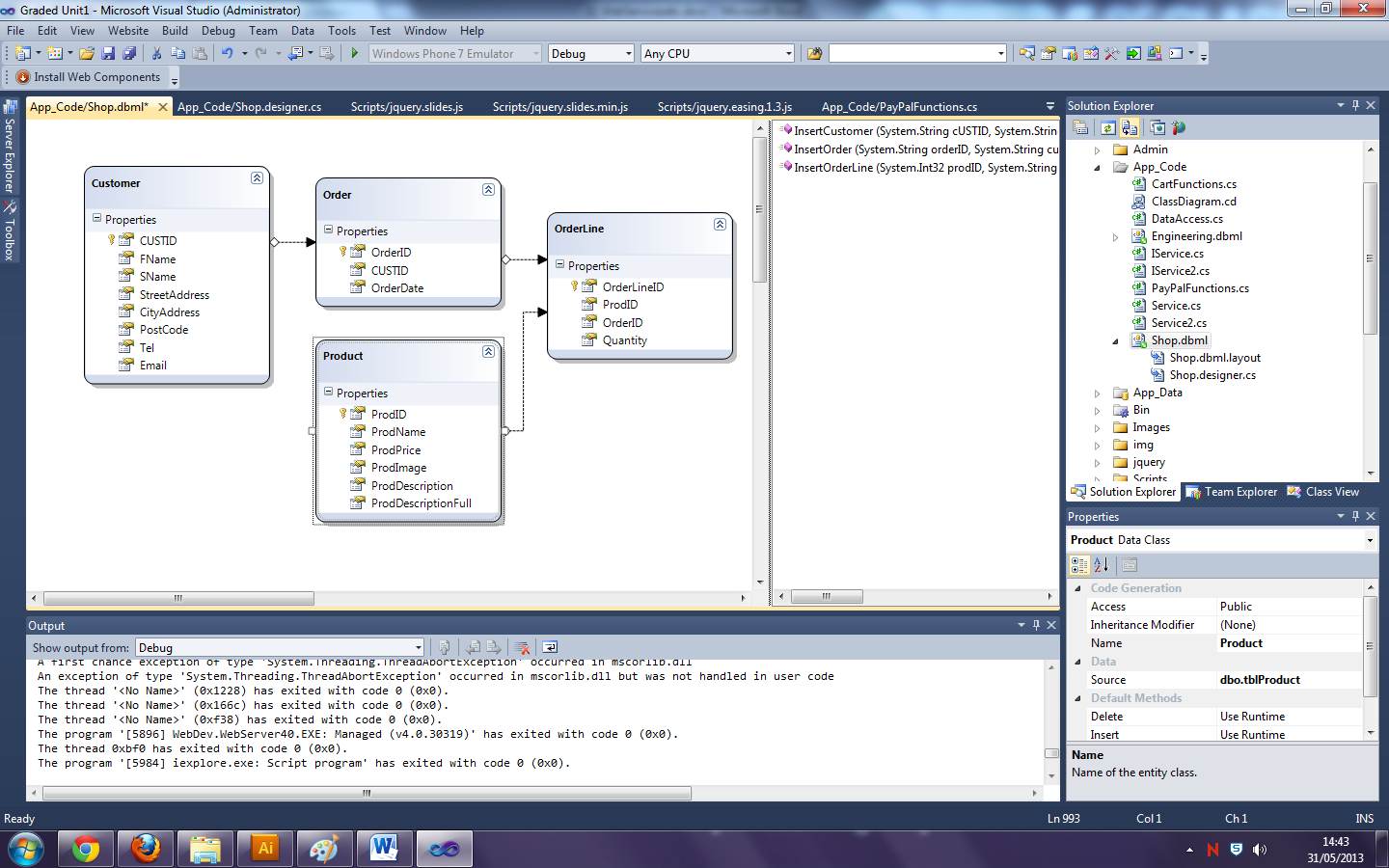
### Diagram



# LINQ-to-SQL

## Shop.dbml

### Diagram



### Shop.designer.cs

#pragma warning disable 1591

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated by a tool.

// Runtime Version:4.0.30319.18033

//

// Changes to this file may cause incorrect behavior and will be lost if

// the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.Linq;

using System.Data.Linq.Mapping;

using System.Linq;

using System.Linq.Expressions;

using System.Reflection;

[global::System.Data.Linq.Mapping.DatabaseAttribute(Name="SHOP")]

public partial class ShopDataContext : System.Data.Linq.DataContext

{

private static System.Data.Linq.Mapping.MappingSource mappingSource = new AttributeMappingSource();

#region Extensibility Method Definitions

partial void OnCreated();

partial void InsertCustomer(Customer instance);

partial void UpdateCustomer(Customer instance);

partial void DeleteCustomer(Customer instance);

partial void InsertOrder(Order instance);

partial void UpdateOrder(Order instance);

partial void DeleteOrder(Order instance);

partial void InsertProduct(Product instance);

partial void UpdateProduct(Product instance);

partial void DeleteProduct(Product instance);

partial void InsertOrderLine(OrderLine instance);

partial void UpdateOrderLine(OrderLine instance);

partial void DeleteOrderLine(OrderLine instance);

#endregion

public ShopDataContext() :

base(global::System.Configuration.ConfigurationManager.ConnectionStrings["SHOPConnectionString"].ConnectionString, mappingSource)

{

OnCreated();

}

public ShopDataContext(string connection) :

base(connection, mappingSource)

{

OnCreated();

}

public ShopDataContext(System.Data.IDbConnection connection) :

base(connection, mappingSource)

{

OnCreated();

}

public ShopDataContext(string connection, System.Data.Linq.Mapping.MappingSource mappingSource) :

base(connection, mappingSource)

{

OnCreated();

}

public ShopDataContext(System.Data.IDbConnection connection, System.Data.Linq.Mapping.MappingSource mappingSource) :

base(connection, mappingSource)

{

OnCreated();

}

public System.Data.Linq.Table<Customer> Customers

{

get

{

return this.GetTable<Customer>();

}

}

public System.Data.Linq.Table<Order> Orders

{

get

{

return this.GetTable<Order>();

}

}

public System.Data.Linq.Table<Product> Products

{

get

{

return this.GetTable<Product>();

}

}

public System.Data.Linq.Table<OrderLine> OrderLines

{

get

{

return this.GetTable<OrderLine>();

}

}

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.InsertCustomer")]

public int InsertCustomer([global::System.Data.Linq.Mapping.ParameterAttribute(Name="CUSTID", DbType="NVarChar(50)")] string cUSTID, [global::System.Data.Linq.Mapping.ParameterAttribute(Name="FName", DbType="NVarChar(50)")] string fName, [global::System.Data.Linq.Mapping.ParameterAttribute(Name="SName", DbType="NVarChar(50)")] string sName, [global::System.Data.Linq.Mapping.ParameterAttribute(Name="StreetAddress", DbType="NVarChar(50)")] string streetAddress, [global::System.Data.Linq.Mapping.ParameterAttribute(Name="CityAddress", DbType="NVarChar(50)")] string cityAddress, [global::System.Data.Linq.Mapping.ParameterAttribute(Name="PostCode", DbType="NVarChar(50)")] string postCode, [global::System.Data.Linq.Mapping.ParameterAttribute(Name="Email", DbType="NVarChar(50)")] string email)

{

IExecuteResult result = this.ExecuteMethodCall(this, ((MethodInfo)(MethodInfo.GetCurrentMethod())), cUSTID, fName, sName, streetAddress, cityAddress, postCode, email);

return ((int)(result.ReturnValue));

}

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.InsertOrderLine")]

public int InsertOrderLine([global::System.Data.Linq.Mapping.ParameterAttribute(DbType="Int")] System.Nullable<int> prodID, [global::System.Data.Linq.Mapping.ParameterAttribute(DbType="NVarChar(50)")] string orderID, [global::System.Data.Linq.Mapping.ParameterAttribute(DbType="Int")] System.Nullable<int> quantity)

{

IExecuteResult result = this.ExecuteMethodCall(this, ((MethodInfo)(MethodInfo.GetCurrentMethod())), prodID, orderID, quantity);

return ((int)(result.ReturnValue));

}

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.InsertOrder")]

public int InsertOrder([global::System.Data.Linq.Mapping.ParameterAttribute(DbType="NVarChar(50)")] string orderID, [global::System.Data.Linq.Mapping.ParameterAttribute(DbType="NVarChar(50)")] string custID, [global::System.Data.Linq.Mapping.ParameterAttribute(DbType="DateTime")] System.Nullable<System.DateTime> orderDate)

{

IExecuteResult result = this.ExecuteMethodCall(this, ((MethodInfo)(MethodInfo.GetCurrentMethod())), orderID, custID, orderDate);

return ((int)(result.ReturnValue));

}

}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.tblCustomer")]

public partial class Customer : INotifyPropertyChanging, INotifyPropertyChanged

{

private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);

private string \_CUSTID;

private string \_FName;

private string \_SName;

private string \_StreetAddress;

private string \_CityAddress;

private string \_PostCode;

private string \_Tel;

private string \_Email;

private EntitySet<Order> \_Orders;

#region Extensibility Method Definitions

partial void OnLoaded();

partial void OnValidate(System.Data.Linq.ChangeAction action);

partial void OnCreated();

partial void OnCUSTIDChanging(string value);

partial void OnCUSTIDChanged();

partial void OnFNameChanging(string value);

partial void OnFNameChanged();

partial void OnSNameChanging(string value);

partial void OnSNameChanged();

partial void OnStreetAddressChanging(string value);

partial void OnStreetAddressChanged();

partial void OnCityAddressChanging(string value);

partial void OnCityAddressChanged();

partial void OnPostCodeChanging(string value);

partial void OnPostCodeChanged();

partial void OnTelChanging(string value);

partial void OnTelChanged();

partial void OnEmailChanging(string value);

partial void OnEmailChanged();

#endregion

public Customer()

{

this.\_Orders = new EntitySet<Order>(new Action<Order>(this.attach\_Orders), new Action<Order>(this.detach\_Orders));

OnCreated();

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_CUSTID", DbType="NVarChar(50) NOT NULL", CanBeNull=false, IsPrimaryKey=true)]

public string CUSTID

{

get

{

return this.\_CUSTID;

}

set

{

if ((this.\_CUSTID != value))

{

this.OnCUSTIDChanging(value);

this.SendPropertyChanging();

this.\_CUSTID = value;

this.SendPropertyChanged("CUSTID");

this.OnCUSTIDChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_FName", DbType="NVarChar(50)")]

public string FName

{

get

{

return this.\_FName;

}

set

{

if ((this.\_FName != value))

{

this.OnFNameChanging(value);

this.SendPropertyChanging();

this.\_FName = value;

this.SendPropertyChanged("FName");

this.OnFNameChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_SName", DbType="NVarChar(50)")]

public string SName

{

get

{

return this.\_SName;

}

set

{

if ((this.\_SName != value))

{

this.OnSNameChanging(value);

this.SendPropertyChanging();

this.\_SName = value;

this.SendPropertyChanged("SName");

this.OnSNameChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_StreetAddress", DbType="NVarChar(50)")]

public string StreetAddress

{

get

{

return this.\_StreetAddress;

}

set

{

if ((this.\_StreetAddress != value))

{

this.OnStreetAddressChanging(value);

this.SendPropertyChanging();

this.\_StreetAddress = value;

this.SendPropertyChanged("StreetAddress");

this.OnStreetAddressChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_CityAddress", DbType="NVarChar(50)")]

public string CityAddress

{

get

{

return this.\_CityAddress;

}

set

{

if ((this.\_CityAddress != value))

{

this.OnCityAddressChanging(value);

this.SendPropertyChanging();

this.\_CityAddress = value;

this.SendPropertyChanged("CityAddress");

this.OnCityAddressChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_PostCode", DbType="NVarChar(50)")]

public string PostCode

{

get

{

return this.\_PostCode;

}

set

{

if ((this.\_PostCode != value))

{

this.OnPostCodeChanging(value);

this.SendPropertyChanging();

this.\_PostCode = value;

this.SendPropertyChanged("PostCode");

this.OnPostCodeChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_Tel", DbType="NVarChar(50)")]

public string Tel

{

get

{

return this.\_Tel;

}

set

{

if ((this.\_Tel != value))

{

this.OnTelChanging(value);

this.SendPropertyChanging();

this.\_Tel = value;

this.SendPropertyChanged("Tel");

this.OnTelChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_Email", DbType="NVarChar(50)")]

public string Email

{

get

{

return this.\_Email;

}

set

{

if ((this.\_Email != value))

{

this.OnEmailChanging(value);

this.SendPropertyChanging();

this.\_Email = value;

this.SendPropertyChanged("Email");

this.OnEmailChanged();

}

}

}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Customer\_Order", Storage="\_Orders", ThisKey="CUSTID", OtherKey="CUSTID")]

public EntitySet<Order> Orders

{

get

{

return this.\_Orders;

}

set

{

this.\_Orders.Assign(value);

}

}

public event PropertyChangingEventHandler PropertyChanging;

public event PropertyChangedEventHandler PropertyChanged;

protected virtual void SendPropertyChanging()

{

if ((this.PropertyChanging != null))

{

this.PropertyChanging(this, emptyChangingEventArgs);

}

}

protected virtual void SendPropertyChanged(String propertyName)

{

if ((this.PropertyChanged != null))

{

this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));

}

}

private void attach\_Orders(Order entity)

{

this.SendPropertyChanging();

entity.Customer = this;

}

private void detach\_Orders(Order entity)

{

this.SendPropertyChanging();

entity.Customer = null;

}

}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.tblOrder")]

public partial class Order : INotifyPropertyChanging, INotifyPropertyChanged

{

private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);

private string \_OrderID;

private string \_CUSTID;

private System.Nullable<System.DateTime> \_OrderDate;

private EntitySet<OrderLine> \_OrderLines;

private EntityRef<Customer> \_Customer;

#region Extensibility Method Definitions

partial void OnLoaded();

partial void OnValidate(System.Data.Linq.ChangeAction action);

partial void OnCreated();

partial void OnOrderIDChanging(string value);

partial void OnOrderIDChanged();

partial void OnCUSTIDChanging(string value);

partial void OnCUSTIDChanged();

partial void OnOrderDateChanging(System.Nullable<System.DateTime> value);

partial void OnOrderDateChanged();

#endregion

public Order()

{

this.\_OrderLines = new EntitySet<OrderLine>(new Action<OrderLine>(this.attach\_OrderLines), new Action<OrderLine>(this.detach\_OrderLines));

this.\_Customer = default(EntityRef<Customer>);

OnCreated();

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_OrderID", DbType="NVarChar(50) NOT NULL", CanBeNull=false, IsPrimaryKey=true)]

public string OrderID

{

get

{

return this.\_OrderID;

}

set

{

if ((this.\_OrderID != value))

{

this.OnOrderIDChanging(value);

this.SendPropertyChanging();

this.\_OrderID = value;

this.SendPropertyChanged("OrderID");

this.OnOrderIDChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_CUSTID", DbType="NVarChar(50)")]

public string CUSTID

{

get

{

return this.\_CUSTID;

}

set

{

if ((this.\_CUSTID != value))

{

if (this.\_Customer.HasLoadedOrAssignedValue)

{

throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();

}

this.OnCUSTIDChanging(value);

this.SendPropertyChanging();

this.\_CUSTID = value;

this.SendPropertyChanged("CUSTID");

this.OnCUSTIDChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_OrderDate", DbType="DateTime")]

public System.Nullable<System.DateTime> OrderDate

{

get

{

return this.\_OrderDate;

}

set

{

if ((this.\_OrderDate != value))

{

this.OnOrderDateChanging(value);

this.SendPropertyChanging();

this.\_OrderDate = value;

this.SendPropertyChanged("OrderDate");

this.OnOrderDateChanged();

}

}

}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Order\_tblOrderLine", Storage="\_OrderLines", ThisKey="OrderID", OtherKey="OrderID")]

public EntitySet<OrderLine> OrderLines

{

get

{

return this.\_OrderLines;

}

set

{

this.\_OrderLines.Assign(value);

}

}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Customer\_Order", Storage="\_Customer", ThisKey="CUSTID", OtherKey="CUSTID", IsForeignKey=true)]

public Customer Customer

{

get

{

return this.\_Customer.Entity;

}

set

{

Customer previousValue = this.\_Customer.Entity;

if (((previousValue != value)

|| (this.\_Customer.HasLoadedOrAssignedValue == false)))

{

this.SendPropertyChanging();

if ((previousValue != null))

{

this.\_Customer.Entity = null;

previousValue.Orders.Remove(this);

}

this.\_Customer.Entity = value;

if ((value != null))

{

value.Orders.Add(this);

this.\_CUSTID = value.CUSTID;

}

else

{

this.\_CUSTID = default(string);

}

this.SendPropertyChanged("Customer");

}

}

}

public event PropertyChangingEventHandler PropertyChanging;

public event PropertyChangedEventHandler PropertyChanged;

protected virtual void SendPropertyChanging()

{

if ((this.PropertyChanging != null))

{

this.PropertyChanging(this, emptyChangingEventArgs);

}

}

protected virtual void SendPropertyChanged(String propertyName)

{

if ((this.PropertyChanged != null))

{

this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));

}

}

private void attach\_OrderLines(OrderLine entity)

{

this.SendPropertyChanging();

entity.Order = this;

}

private void detach\_OrderLines(OrderLine entity)

{

this.SendPropertyChanging();

entity.Order = null;

}

}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.tblProduct")]

public partial class Product : INotifyPropertyChanging, INotifyPropertyChanged

{

private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);

private int \_ProdID;

private string \_ProdName;

private System.Nullable<double> \_ProdPrice;

private string \_ProdImage;

private string \_ProdDescription;

private string \_ProdDescriptionFull;

private EntitySet<OrderLine> \_OrderLines;

#region Extensibility Method Definitions

partial void OnLoaded();

partial void OnValidate(System.Data.Linq.ChangeAction action);

partial void OnCreated();

partial void OnProdIDChanging(int value);

partial void OnProdIDChanged();

partial void OnProdNameChanging(string value);

partial void OnProdNameChanged();

partial void OnProdPriceChanging(System.Nullable<double> value);

partial void OnProdPriceChanged();

partial void OnProdImageChanging(string value);

partial void OnProdImageChanged();

partial void OnProdDescriptionChanging(string value);

partial void OnProdDescriptionChanged();

partial void OnProdDescriptionFullChanging(string value);

partial void OnProdDescriptionFullChanged();

#endregion

public Product()

{

this.\_OrderLines = new EntitySet<OrderLine>(new Action<OrderLine>(this.attach\_OrderLines), new Action<OrderLine>(this.detach\_OrderLines));

OnCreated();

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_ProdID", AutoSync=AutoSync.OnInsert, DbType="Int NOT NULL IDENTITY", IsPrimaryKey=true, IsDbGenerated=true)]

public int ProdID

{

get

{

return this.\_ProdID;

}

set

{

if ((this.\_ProdID != value))

{

this.OnProdIDChanging(value);

this.SendPropertyChanging();

this.\_ProdID = value;

this.SendPropertyChanged("ProdID");

this.OnProdIDChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_ProdName", DbType="NVarChar(50)")]

public string ProdName

{

get

{

return this.\_ProdName;

}

set

{

if ((this.\_ProdName != value))

{

this.OnProdNameChanging(value);

this.SendPropertyChanging();

this.\_ProdName = value;

this.SendPropertyChanged("ProdName");

this.OnProdNameChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_ProdPrice", DbType="Float")]

public System.Nullable<double> ProdPrice

{

get

{

return this.\_ProdPrice;

}

set

{

if ((this.\_ProdPrice != value))

{

this.OnProdPriceChanging(value);

this.SendPropertyChanging();

this.\_ProdPrice = value;

this.SendPropertyChanged("ProdPrice");

this.OnProdPriceChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_ProdImage", DbType="NVarChar(50)")]

public string ProdImage

{

get

{

return this.\_ProdImage;

}

set

{

if ((this.\_ProdImage != value))

{

this.OnProdImageChanging(value);

this.SendPropertyChanging();

this.\_ProdImage = value;

this.SendPropertyChanged("ProdImage");

this.OnProdImageChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_ProdDescription", DbType="NVarChar(50)")]

public string ProdDescription

{

get

{

return this.\_ProdDescription;

}

set

{

if ((this.\_ProdDescription != value))

{

this.OnProdDescriptionChanging(value);

this.SendPropertyChanging();

this.\_ProdDescription = value;

this.SendPropertyChanged("ProdDescription");

this.OnProdDescriptionChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_ProdDescriptionFull", DbType="Text", UpdateCheck=UpdateCheck.Never)]

public string ProdDescriptionFull

{

get

{

return this.\_ProdDescriptionFull;

}

set

{

if ((this.\_ProdDescriptionFull != value))

{

this.OnProdDescriptionFullChanging(value);

this.SendPropertyChanging();

this.\_ProdDescriptionFull = value;

this.SendPropertyChanged("ProdDescriptionFull");

this.OnProdDescriptionFullChanged();

}

}

}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Product\_tblOrderLine", Storage="\_OrderLines", ThisKey="ProdID", OtherKey="ProdID")]

public EntitySet<OrderLine> OrderLines

{

get

{

return this.\_OrderLines;

}

set

{

this.\_OrderLines.Assign(value);

}

}

public event PropertyChangingEventHandler PropertyChanging;

public event PropertyChangedEventHandler PropertyChanged;

protected virtual void SendPropertyChanging()

{

if ((this.PropertyChanging != null))

{

this.PropertyChanging(this, emptyChangingEventArgs);

}

}

protected virtual void SendPropertyChanged(String propertyName)

{

if ((this.PropertyChanged != null))

{

this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));

}

}

private void attach\_OrderLines(OrderLine entity)

{

this.SendPropertyChanging();

entity.Product = this;

}

private void detach\_OrderLines(OrderLine entity)

{

this.SendPropertyChanging();

entity.Product = null;

}

}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.tblOrderLine")]

public partial class OrderLine : INotifyPropertyChanging, INotifyPropertyChanged

{

private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);

private int \_OrderLineID;

private System.Nullable<int> \_ProdID;

private string \_OrderID;

private System.Nullable<int> \_Quantity;

private EntityRef<Order> \_Order;

private EntityRef<Product> \_Product;

#region Extensibility Method Definitions

partial void OnLoaded();

partial void OnValidate(System.Data.Linq.ChangeAction action);

partial void OnCreated();

partial void OnOrderLineIDChanging(int value);

partial void OnOrderLineIDChanged();

partial void OnProdIDChanging(System.Nullable<int> value);

partial void OnProdIDChanged();

partial void OnOrderIDChanging(string value);

partial void OnOrderIDChanged();

partial void OnQuantityChanging(System.Nullable<int> value);

partial void OnQuantityChanged();

#endregion

public OrderLine()

{

this.\_Order = default(EntityRef<Order>);

this.\_Product = default(EntityRef<Product>);

OnCreated();

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_OrderLineID", AutoSync=AutoSync.OnInsert, DbType="Int NOT NULL IDENTITY", IsPrimaryKey=true, IsDbGenerated=true)]

public int OrderLineID

{

get

{

return this.\_OrderLineID;

}

set

{

if ((this.\_OrderLineID != value))

{

this.OnOrderLineIDChanging(value);

this.SendPropertyChanging();

this.\_OrderLineID = value;

this.SendPropertyChanged("OrderLineID");

this.OnOrderLineIDChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_ProdID", DbType="Int")]

public System.Nullable<int> ProdID

{

get

{

return this.\_ProdID;

}

set

{

if ((this.\_ProdID != value))

{

if (this.\_Product.HasLoadedOrAssignedValue)

{

throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();

}

this.OnProdIDChanging(value);

this.SendPropertyChanging();

this.\_ProdID = value;

this.SendPropertyChanged("ProdID");

this.OnProdIDChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_OrderID", DbType="NVarChar(50)")]

public string OrderID

{

get

{

return this.\_OrderID;

}

set

{

if ((this.\_OrderID != value))

{

//if (this.\_Order.HasLoadedOrAssignedValue)

//{

// throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();

//}

this.OnOrderIDChanging(value);

this.SendPropertyChanging();

this.\_OrderID = value;

this.SendPropertyChanged("OrderID");

this.OnOrderIDChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_Quantity", DbType="Int")]

public System.Nullable<int> Quantity

{

get

{

return this.\_Quantity;

}

set

{

if ((this.\_Quantity != value))

{

this.OnQuantityChanging(value);

this.SendPropertyChanging();

this.\_Quantity = value;

this.SendPropertyChanged("Quantity");

this.OnQuantityChanged();

}

}

}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Order\_tblOrderLine", Storage="\_Order", ThisKey="OrderID", OtherKey="OrderID", IsForeignKey=true)]

public Order Order

{

get

{

return this.\_Order.Entity;

}

set

{

Order previousValue = this.\_Order.Entity;

if (((previousValue != value)

|| (this.\_Order.HasLoadedOrAssignedValue == false)))

{

this.SendPropertyChanging();

if ((previousValue != null))

{

this.\_Order.Entity = null;

previousValue.OrderLines.Remove(this);

}

this.\_Order.Entity = value;

if ((value != null))

{

value.OrderLines.Add(this);

this.\_OrderID = value.OrderID;

}

else

{

this.\_OrderID = default(string);

}

this.SendPropertyChanged("Order");

}

}

}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Product\_tblOrderLine", Storage="\_Product", ThisKey="ProdID", OtherKey="ProdID", IsForeignKey=true)]

public Product Product

{

get

{

return this.\_Product.Entity;

}

set

{

Product previousValue = this.\_Product.Entity;

if (((previousValue != value)

|| (this.\_Product.HasLoadedOrAssignedValue == false)))

{

this.SendPropertyChanging();

if ((previousValue != null))

{

this.\_Product.Entity = null;

previousValue.OrderLines.Remove(this);

}

this.\_Product.Entity = value;

if ((value != null))

{

value.OrderLines.Add(this);

this.\_ProdID = value.ProdID;

}

else

{

this.\_ProdID = default(Nullable<int>);

}

this.SendPropertyChanged("Product");

}

}

}

public event PropertyChangingEventHandler PropertyChanging;

public event PropertyChangedEventHandler PropertyChanged;

protected virtual void SendPropertyChanging()

{

if ((this.PropertyChanging != null))

{

this.PropertyChanging(this, emptyChangingEventArgs);

}

}

protected virtual void SendPropertyChanged(String propertyName)

{

if ((this.PropertyChanged != null))

{

this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));

}

}

}

#pragma warning restore 1591

### Stored Procedures

#### InsertCustomer

ALTER PROCEDURE dbo.InsertCustomer

(

@CUSTID NVARCHAR(50),

@FName NVARCHAR(50),

@SName NVARCHAR(50),

@StreetAddress NVARCHAR(50),

@CityAddress NVARCHAR(50),

@PostCode NVARCHAR(50),

@Email NVARCHAR(50)

)

AS

/\*

INSERT INTO [dbo].[tblCustomer]([CUSTID],[FName],[SName],[StreetAddress],[CityAddress],[PostCode],[Email])

VALUES(@CUSTID,@FName,@SName,@StreetAddress,@CityAddress,@PostCode,@Email)

\*/

IF EXISTS

(SELECT [CUSTID]

FROM [dbo].[tblCustomer]

WHERE [CUSTID] = @CUSTID)

BEGIN

UPDATE [dbo].[tblCustomer]

SET [FName]=@FName,[SName]=@SName,[StreetAddress]=@StreetAddress,[CityAddress]=@CityAddress,[PostCode]=@PostCode,[Email]=@Email

WHERE [CUSTID] = @CUSTID

END

ELSE

BEGIN

INSERT INTO [dbo].[tblCustomer]

([CUSTID],[FName],[SName],[StreetAddress],[CityAddress],[PostCode],[Email])

VALUES(@CUSTID,@FName,@SName,@StreetAddress,@CityAddress,@PostCode,@Email)

END

#### InsertOrder

ALTER PROCEDURE dbo.InsertOrder

(

@orderID nvarchar(50),

@custID nvarchar(50),

@orderDate datetime

)

AS

BEGIN

INSERT INTO [dbo].[tblOrder]

([orderID],[CUSTID],[OrderDate])

VALUES(@orderID,@custID,@orderDate)

END

#### InsertOrderLine

ALTER PROCEDURE dbo.InsertOrderLine

(

@prodID int,

@orderID nvarchar(50),

@quantity int

)

AS

BEGIN

INSERT INTO [dbo].[tblOrderLine]

([prodID],[orderID],[Quantity])

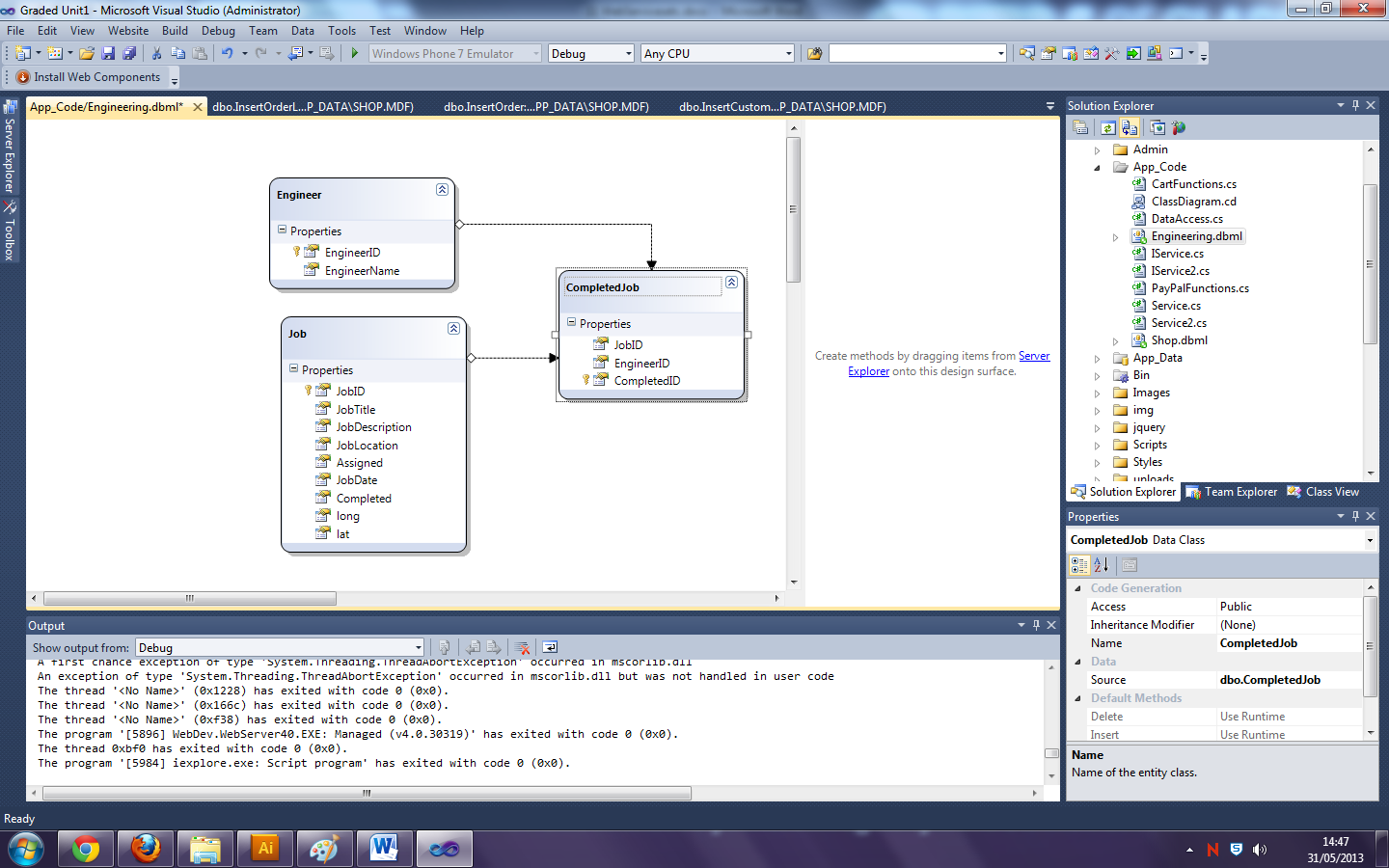
VALUES(@prodID,@orderID,@quantity)

END

RETURN

## Engineer.dbml

### Diagram



### Engineer.designer.cs

#pragma warning disable 1591

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated by a tool.

// Runtime Version:4.0.30319.18033

//

// Changes to this file may cause incorrect behavior and will be lost if

// the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.Linq;

using System.Data.Linq.Mapping;

using System.Linq;

using System.Linq.Expressions;

using System.Reflection;

[global::System.Data.Linq.Mapping.DatabaseAttribute(Name="EngineerServices")]

public partial class EngineeringDataContext : System.Data.Linq.DataContext

{

private static System.Data.Linq.Mapping.MappingSource mappingSource = new AttributeMappingSource();

#region Extensibility Method Definitions

partial void OnCreated();

partial void InsertCompletedJob(CompletedJob instance);

partial void UpdateCompletedJob(CompletedJob instance);

partial void DeleteCompletedJob(CompletedJob instance);

partial void InsertJob(Job instance);

partial void UpdateJob(Job instance);

partial void DeleteJob(Job instance);

partial void InsertEngineer(Engineer instance);

partial void UpdateEngineer(Engineer instance);

partial void DeleteEngineer(Engineer instance);

#endregion

public EngineeringDataContext() :

base(global::System.Configuration.ConfigurationManager.ConnectionStrings["EngineerServicesConnectionString"].ConnectionString, mappingSource)

{

OnCreated();

}

public EngineeringDataContext(string connection) :

base(connection, mappingSource)

{

OnCreated();

}

public EngineeringDataContext(System.Data.IDbConnection connection) :

base(connection, mappingSource)

{

OnCreated();

}

public EngineeringDataContext(string connection, System.Data.Linq.Mapping.MappingSource mappingSource) :

base(connection, mappingSource)

{

OnCreated();

}

public EngineeringDataContext(System.Data.IDbConnection connection, System.Data.Linq.Mapping.MappingSource mappingSource) :

base(connection, mappingSource)

{

OnCreated();

}

public System.Data.Linq.Table<CompletedJob> CompletedJobs

{

get

{

return this.GetTable<CompletedJob>();

}

}

public System.Data.Linq.Table<Job> Jobs

{

get

{

return this.GetTable<Job>();

}

}

public System.Data.Linq.Table<Engineer> Engineers

{

get

{

return this.GetTable<Engineer>();

}

}

}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.CompletedJob")]

public partial class CompletedJob : INotifyPropertyChanging, INotifyPropertyChanged

{

private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);

private string \_JobID;

private string \_EngineerID;

private int \_CompletedID;

private EntityRef<Job> \_Job;

private EntityRef<Engineer> \_Engineer;

#region Extensibility Method Definitions

partial void OnLoaded();

partial void OnValidate(System.Data.Linq.ChangeAction action);

partial void OnCreated();

partial void OnJobIDChanging(string value);

partial void OnJobIDChanged();

partial void OnEngineerIDChanging(string value);

partial void OnEngineerIDChanged();

partial void OnCompletedIDChanging(int value);

partial void OnCompletedIDChanged();

#endregion

public CompletedJob()

{

this.\_Job = default(EntityRef<Job>);

this.\_Engineer = default(EntityRef<Engineer>);

OnCreated();

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_JobID", DbType="NChar(10)")]

public string JobID

{

get

{

return this.\_JobID;

}

set

{

if ((this.\_JobID != value))

{

if (this.\_Job.HasLoadedOrAssignedValue)

{

throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();

}

this.OnJobIDChanging(value);

this.SendPropertyChanging();

this.\_JobID = value;

this.SendPropertyChanged("JobID");

this.OnJobIDChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_EngineerID", DbType="NChar(10)")]

public string EngineerID

{

get

{

return this.\_EngineerID;

}

set

{

if ((this.\_EngineerID != value))

{

if (this.\_Engineer.HasLoadedOrAssignedValue)

{

throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();

}

this.OnEngineerIDChanging(value);

this.SendPropertyChanging();

this.\_EngineerID = value;

this.SendPropertyChanged("EngineerID");

this.OnEngineerIDChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_CompletedID", DbType="Int NOT NULL", IsPrimaryKey=true)]

public int CompletedID

{

get

{

return this.\_CompletedID;

}

set

{

if ((this.\_CompletedID != value))

{

this.OnCompletedIDChanging(value);

this.SendPropertyChanging();

this.\_CompletedID = value;

this.SendPropertyChanged("CompletedID");

this.OnCompletedIDChanged();

}

}

}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Job\_CompletedJob", Storage="\_Job", ThisKey="JobID", OtherKey="JobID", IsForeignKey=true)]

public Job Job

{

get

{

return this.\_Job.Entity;

}

set

{

Job previousValue = this.\_Job.Entity;

if (((previousValue != value)

|| (this.\_Job.HasLoadedOrAssignedValue == false)))

{

this.SendPropertyChanging();

if ((previousValue != null))

{

this.\_Job.Entity = null;

previousValue.CompletedJobs.Remove(this);

}

this.\_Job.Entity = value;

if ((value != null))

{

value.CompletedJobs.Add(this);

this.\_JobID = value.JobID;

}

else

{

this.\_JobID = default(string);

}

this.SendPropertyChanged("Job");

}

}

}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Engineer\_CompletedJob", Storage="\_Engineer", ThisKey="EngineerID", OtherKey="EngineerID", IsForeignKey=true)]

public Engineer Engineer

{

get

{

return this.\_Engineer.Entity;

}

set

{

Engineer previousValue = this.\_Engineer.Entity;

if (((previousValue != value)

|| (this.\_Engineer.HasLoadedOrAssignedValue == false)))

{

this.SendPropertyChanging();

if ((previousValue != null))

{

this.\_Engineer.Entity = null;

previousValue.CompletedJobs.Remove(this);

}

this.\_Engineer.Entity = value;

if ((value != null))

{

value.CompletedJobs.Add(this);

this.\_EngineerID = value.EngineerID;

}

else

{

this.\_EngineerID = default(string);

}

this.SendPropertyChanged("Engineer");

}

}

}

public event PropertyChangingEventHandler PropertyChanging;

public event PropertyChangedEventHandler PropertyChanged;

protected virtual void SendPropertyChanging()

{

if ((this.PropertyChanging != null))

{

this.PropertyChanging(this, emptyChangingEventArgs);

}

}

protected virtual void SendPropertyChanged(String propertyName)

{

if ((this.PropertyChanged != null))

{

this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));

}

}

}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.Job")]

public partial class Job : INotifyPropertyChanging, INotifyPropertyChanged

{

private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);

private string \_JobID;

private string \_JobTitle;

private string \_JobDescription;

private string \_JobLocation;

private System.Nullable<bool> \_Assigned;

private System.Nullable<System.DateTime> \_JobDate;

private System.Nullable<bool> \_Completed;

private System.Nullable<int> \_long;

private System.Nullable<int> \_lat;

private EntitySet<CompletedJob> \_CompletedJobs;

#region Extensibility Method Definitions

partial void OnLoaded();

partial void OnValidate(System.Data.Linq.ChangeAction action);

partial void OnCreated();

partial void OnJobIDChanging(string value);

partial void OnJobIDChanged();

partial void OnJobTitleChanging(string value);

partial void OnJobTitleChanged();

partial void OnJobDescriptionChanging(string value);

partial void OnJobDescriptionChanged();

partial void OnJobLocationChanging(string value);

partial void OnJobLocationChanged();

partial void OnAssignedChanging(System.Nullable<bool> value);

partial void OnAssignedChanged();

partial void OnJobDateChanging(System.Nullable<System.DateTime> value);

partial void OnJobDateChanged();

partial void OnCompletedChanging(System.Nullable<bool> value);

partial void OnCompletedChanged();

partial void OnlongChanging(System.Nullable<int> value);

partial void OnlongChanged();

partial void OnlatChanging(System.Nullable<int> value);

partial void OnlatChanged();

#endregion

public Job()

{

this.\_CompletedJobs = new EntitySet<CompletedJob>(new Action<CompletedJob>(this.attach\_CompletedJobs), new Action<CompletedJob>(this.detach\_CompletedJobs));

OnCreated();

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_JobID", DbType="NChar(10) NOT NULL", CanBeNull=false, IsPrimaryKey=true)]

public string JobID

{

get

{

return this.\_JobID;

}

set

{

if ((this.\_JobID != value))

{

this.OnJobIDChanging(value);

this.SendPropertyChanging();

this.\_JobID = value;

this.SendPropertyChanged("JobID");

this.OnJobIDChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_JobTitle", DbType="NVarChar(50)")]

public string JobTitle

{

get

{

return this.\_JobTitle;

}

set

{

if ((this.\_JobTitle != value))

{

this.OnJobTitleChanging(value);

this.SendPropertyChanging();

this.\_JobTitle = value;

this.SendPropertyChanged("JobTitle");

this.OnJobTitleChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_JobDescription", DbType="NVarChar(50)")]

public string JobDescription

{

get

{

return this.\_JobDescription;

}

set

{

if ((this.\_JobDescription != value))

{

this.OnJobDescriptionChanging(value);

this.SendPropertyChanging();

this.\_JobDescription = value;

this.SendPropertyChanged("JobDescription");

this.OnJobDescriptionChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_JobLocation", DbType="NVarChar(50)")]

public string JobLocation

{

get

{

return this.\_JobLocation;

}

set

{

if ((this.\_JobLocation != value))

{

this.OnJobLocationChanging(value);

this.SendPropertyChanging();

this.\_JobLocation = value;

this.SendPropertyChanged("JobLocation");

this.OnJobLocationChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_Assigned", DbType="Bit")]

public System.Nullable<bool> Assigned

{

get

{

return this.\_Assigned;

}

set

{

if ((this.\_Assigned != value))

{

this.OnAssignedChanging(value);

this.SendPropertyChanging();

this.\_Assigned = value;

this.SendPropertyChanged("Assigned");

this.OnAssignedChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_JobDate", DbType="DateTime")]

public System.Nullable<System.DateTime> JobDate

{

get

{

return this.\_JobDate;

}

set

{

if ((this.\_JobDate != value))

{

this.OnJobDateChanging(value);

this.SendPropertyChanging();

this.\_JobDate = value;

this.SendPropertyChanged("JobDate");

this.OnJobDateChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_Completed", DbType="Bit")]

public System.Nullable<bool> Completed

{

get

{

return this.\_Completed;

}

set

{

if ((this.\_Completed != value))

{

this.OnCompletedChanging(value);

this.SendPropertyChanging();

this.\_Completed = value;

this.SendPropertyChanged("Completed");

this.OnCompletedChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Name="long", Storage="\_long", DbType="Int")]

public System.Nullable<int> @long

{

get

{

return this.\_long;

}

set

{

if ((this.\_long != value))

{

this.OnlongChanging(value);

this.SendPropertyChanging();

this.\_long = value;

this.SendPropertyChanged("@long");

this.OnlongChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_lat", DbType="Int")]

public System.Nullable<int> lat

{

get

{

return this.\_lat;

}

set

{

if ((this.\_lat != value))

{

this.OnlatChanging(value);

this.SendPropertyChanging();

this.\_lat = value;

this.SendPropertyChanged("lat");

this.OnlatChanged();

}

}

}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Job\_CompletedJob", Storage="\_CompletedJobs", ThisKey="JobID", OtherKey="JobID")]

public EntitySet<CompletedJob> CompletedJobs

{

get

{

return this.\_CompletedJobs;

}

set

{

this.\_CompletedJobs.Assign(value);

}

}

public event PropertyChangingEventHandler PropertyChanging;

public event PropertyChangedEventHandler PropertyChanged;

protected virtual void SendPropertyChanging()

{

if ((this.PropertyChanging != null))

{

this.PropertyChanging(this, emptyChangingEventArgs);

}

}

protected virtual void SendPropertyChanged(String propertyName)

{

if ((this.PropertyChanged != null))

{

this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));

}

}

private void attach\_CompletedJobs(CompletedJob entity)

{

this.SendPropertyChanging();

entity.Job = this;

}

private void detach\_CompletedJobs(CompletedJob entity)

{

this.SendPropertyChanging();

entity.Job = null;

}

}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.Engineer")]

public partial class Engineer : INotifyPropertyChanging, INotifyPropertyChanged

{

private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);

private string \_EngineerID;

private string \_EngineerName;

private EntitySet<CompletedJob> \_CompletedJobs;

#region Extensibility Method Definitions

partial void OnLoaded();

partial void OnValidate(System.Data.Linq.ChangeAction action);

partial void OnCreated();

partial void OnEngineerIDChanging(string value);

partial void OnEngineerIDChanged();

partial void OnEngineerNameChanging(string value);

partial void OnEngineerNameChanged();

#endregion

public Engineer()

{

this.\_CompletedJobs = new EntitySet<CompletedJob>(new Action<CompletedJob>(this.attach\_CompletedJobs), new Action<CompletedJob>(this.detach\_CompletedJobs));

OnCreated();

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_EngineerID", DbType="NChar(10) NOT NULL", CanBeNull=false, IsPrimaryKey=true)]

public string EngineerID

{

get

{

return this.\_EngineerID;

}

set

{

if ((this.\_EngineerID != value))

{

this.OnEngineerIDChanging(value);

this.SendPropertyChanging();

this.\_EngineerID = value;

this.SendPropertyChanged("EngineerID");

this.OnEngineerIDChanged();

}

}

}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="\_EngineerName", DbType="NChar(10)")]

public string EngineerName

{

get

{

return this.\_EngineerName;

}

set

{

if ((this.\_EngineerName != value))

{

this.OnEngineerNameChanging(value);

this.SendPropertyChanging();

this.\_EngineerName = value;

this.SendPropertyChanged("EngineerName");

this.OnEngineerNameChanged();

}

}

}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Engineer\_CompletedJob", Storage="\_CompletedJobs", ThisKey="EngineerID", OtherKey="EngineerID")]

public EntitySet<CompletedJob> CompletedJobs

{

get

{

return this.\_CompletedJobs;

}

set

{

this.\_CompletedJobs.Assign(value);

}

}

public event PropertyChangingEventHandler PropertyChanging;

public event PropertyChangedEventHandler PropertyChanged;

protected virtual void SendPropertyChanging()

{

if ((this.PropertyChanging != null))

{

this.PropertyChanging(this, emptyChangingEventArgs);

}

}

protected virtual void SendPropertyChanged(String propertyName)

{

if ((this.PropertyChanged != null))

{

this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));

}

}

private void attach\_CompletedJobs(CompletedJob entity)

{

this.SendPropertyChanging();

entity.Engineer = this;

}

private void detach\_CompletedJobs(CompletedJob entity)

{

this.SendPropertyChanging();

entity.Engineer = null;

}

}

#pragma warning restore 1591

# Web.config

<?xml version="1.0"?>

<!--

For more information on how to configure your ASP.NET application, please visit

http://go.microsoft.com/fwlink/?LinkId=169433

-->

<configuration>

<connectionStrings>

<add name="SHOPConnectionString" connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\SHOP.mdf;Integrated Security=True;User Instance=True" providerName="System.Data.SqlClient"/>

<add name="EngineerServicesConnectionString" connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\EngineerServices.mdf;Integrated Security=True;User Instance=True" providerName="System.Data.SqlClient"/>

</connectionStrings>

<system.web>

<compilation debug="true" targetFramework="4.0">

<assemblies>

<add assembly="System.Data.Linq, Version=4.0.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>

<add assembly="mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>

<add assembly="System.Design, Version=4.0.0.0, Culture=neutral, PublicKeyToken=B03F5F7F11D50A3A"/>

<add assembly="System.Web.Extensions.Design, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

<add assembly="System.Windows.Forms, Version=4.0.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>

</assemblies>

</compilation>

<anonymousIdentification enabled="true"/>

<authentication mode="Forms">

<forms loginUrl="~/Account/Login.aspx" timeout="2880"/>

</authentication>

<membership>

<providers>

<clear/>

<add name="AspNetSqlMembershipProvider" type="System.Web.Security.SqlMembershipProvider" connectionStringName="LocalSqlServer" enablePasswordRetrieval="false" enablePasswordReset="true" requiresQuestionAndAnswer="false" requiresUniqueEmail="true" maxInvalidPasswordAttempts="5" minRequiredPasswordLength="6" minRequiredNonalphanumericCharacters="0" passwordAttemptWindow="10" applicationName="/"/>

</providers>

</membership>

<profile>

<providers>

<clear/>

<add name="AspNetSqlProfileProvider" type="System.Web.Profile.SqlProfileProvider" connectionStringName="LocalSqlServer"/>

</providers>

</profile>

<roleManager enabled="true">

<providers>

<clear/>

<add connectionStringName="LocalSqlServer" applicationName="/" name="AspNetSqlRoleProvider" type="System.Web.Security.SqlRoleProvider"/>

<add applicationName="/" name="AspNetWindowsTokenRoleProvider" type="System.Web.Security.WindowsTokenRoleProvider"/>

</providers>

</roleManager>

</system.web>

<system.serviceModel>

<behaviors>

<serviceBehaviors>

<behavior name="">

<serviceMetadata httpGetEnabled="true"/>

<serviceDebug includeExceptionDetailInFaults="false"/>

</behavior>

</serviceBehaviors>

</behaviors>

<serviceHostingEnvironment multipleSiteBindingsEnabled="true"/>

</system.serviceModel>

</configuration>