MONASH University

MONASH
INFORMATION
TECHNOLOGY

FIT5192 Module 2
Internet Applications Development
Lecture 10 – Part II

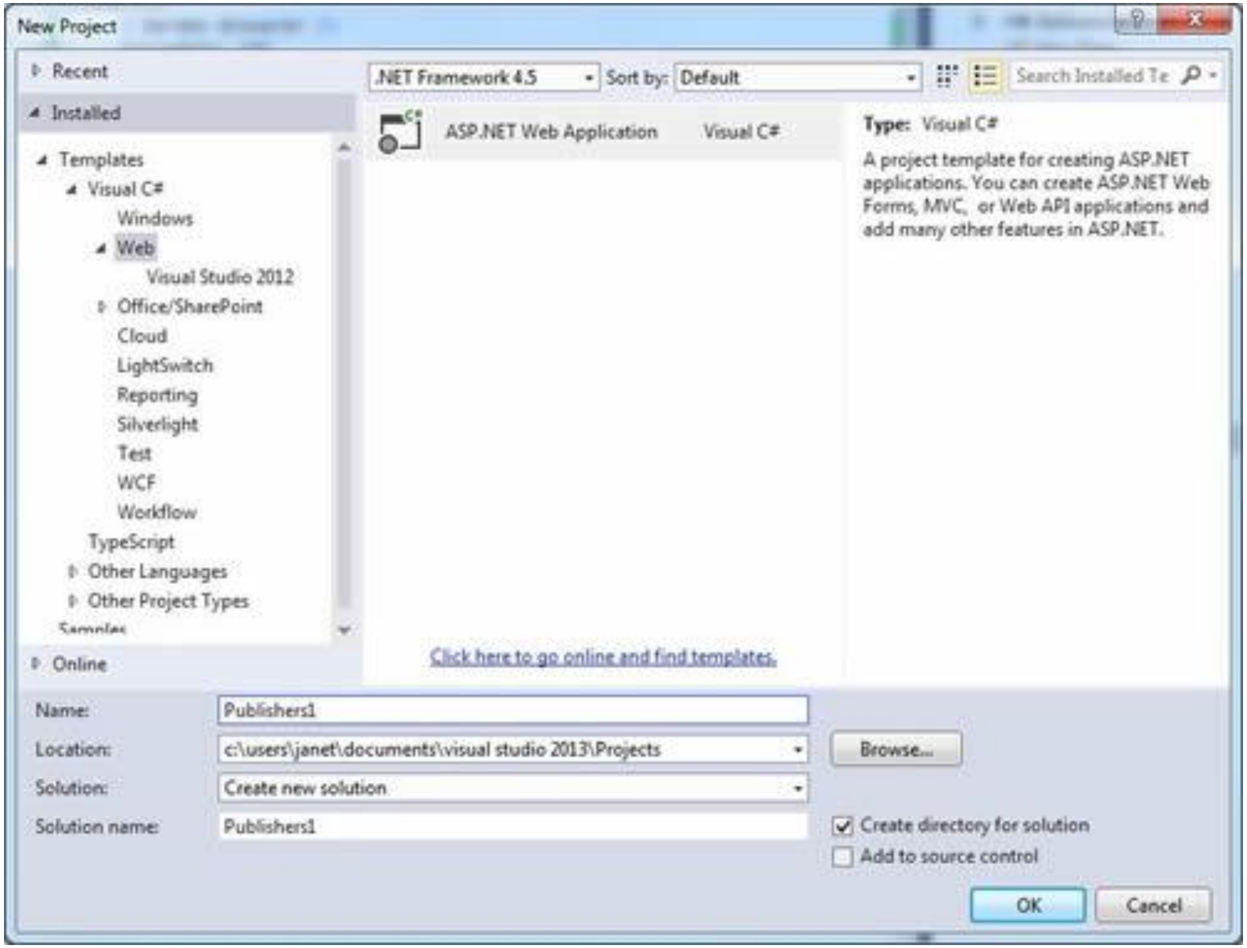
GROUP
OF EIGHT
AUSTRALIA

Lecture Overview

1. Introduction
2. Creating an SQL Server Database
3. Models- Using the Database
4. Generating Controllers
5. Generating Views
6. Changing the Look and Feel

Introduction

Creating an MVC Application



New Project

.NET Framework 4.5 · Sort by: Default

ASP.NET Web Application · Visual C#

Type: Visual C#

A project template for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Name: Publishers1

Location: c:\users\janet\documents\visual studio 2013\Projects

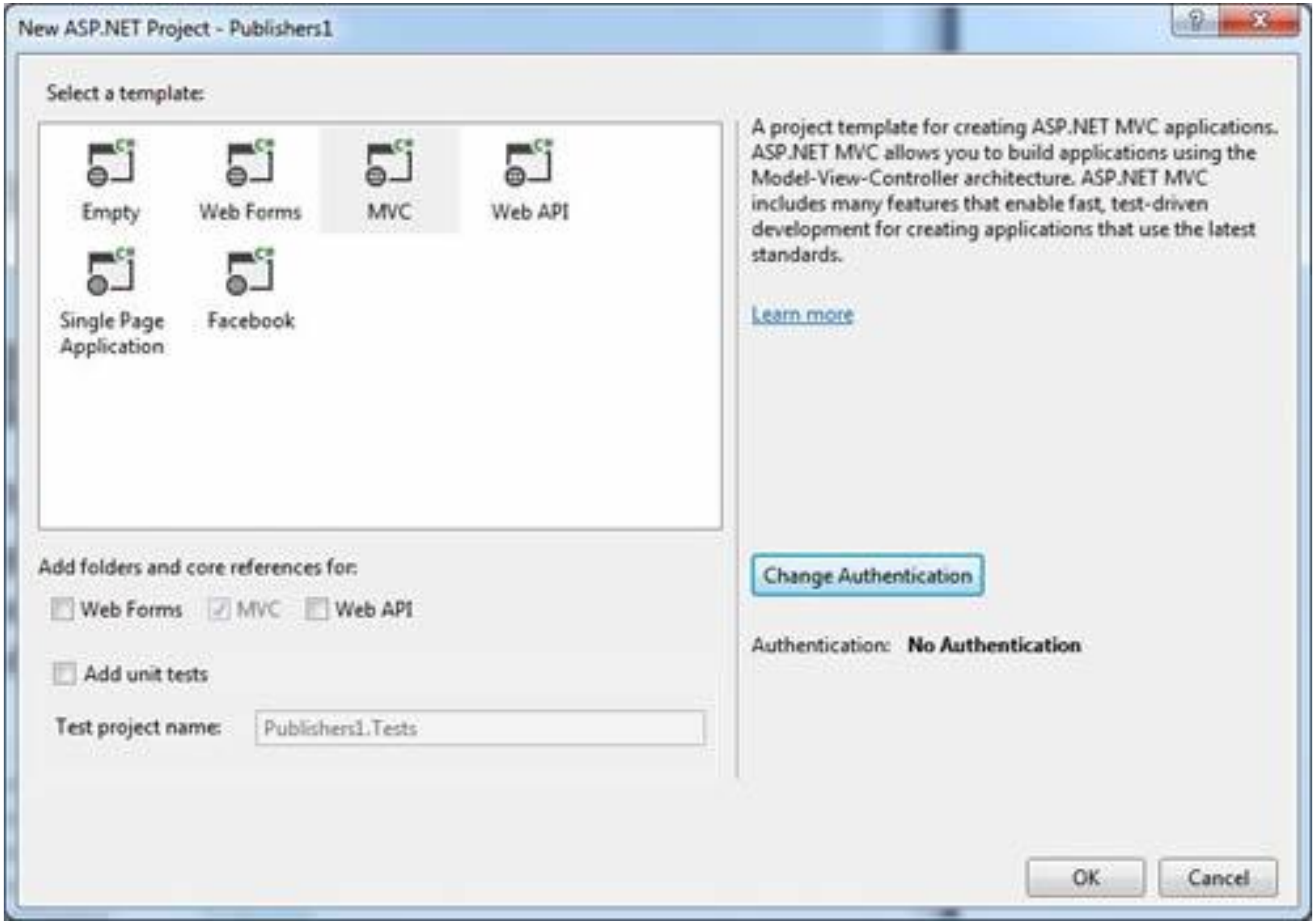
Solution: Create new solution

Solution name: Publishers1

☑ Create directory for solution

□ Add to source control

MVC Razor's View Application



New ASP.NET Project - Publishers1

Select a template:

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Add folders and core references for:

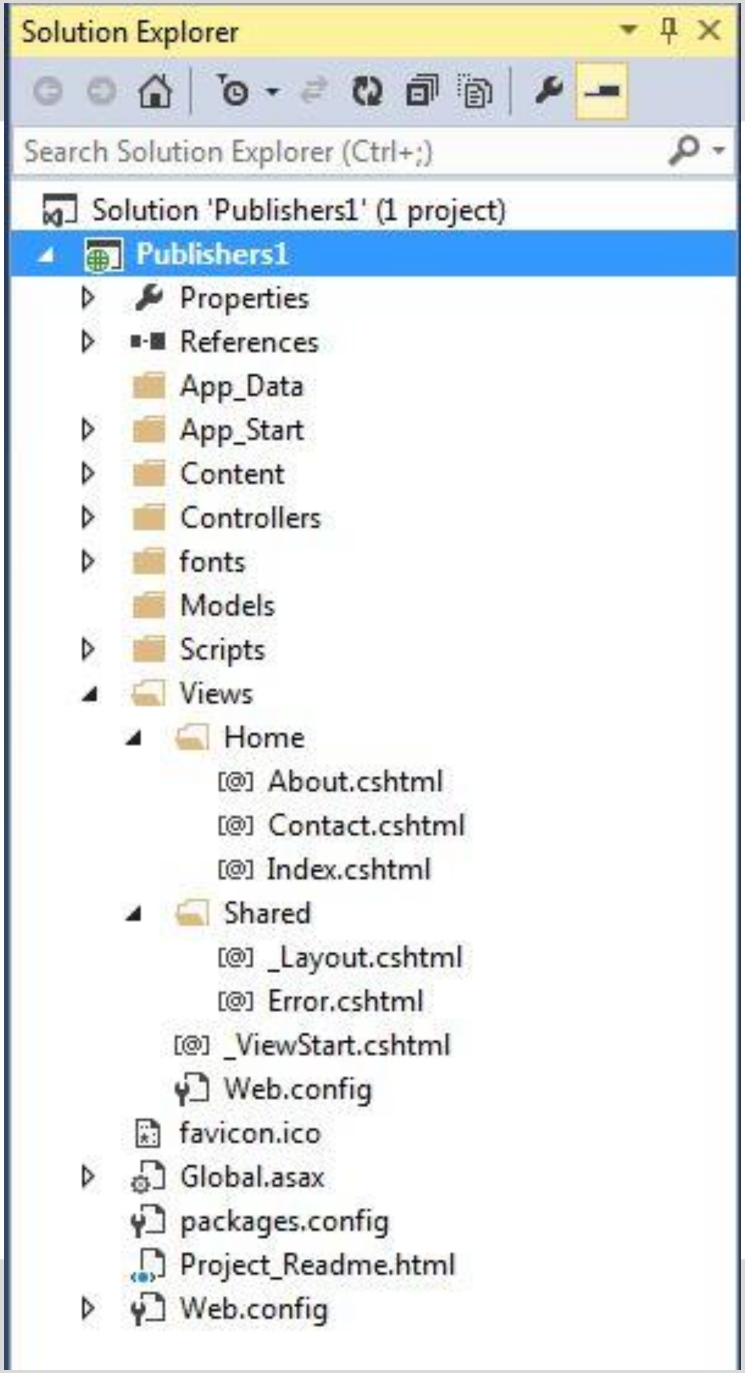
□ Web Forms ☑ MVC □ Web API

□ Add unit tests

Test project name: Publishers1.Tests

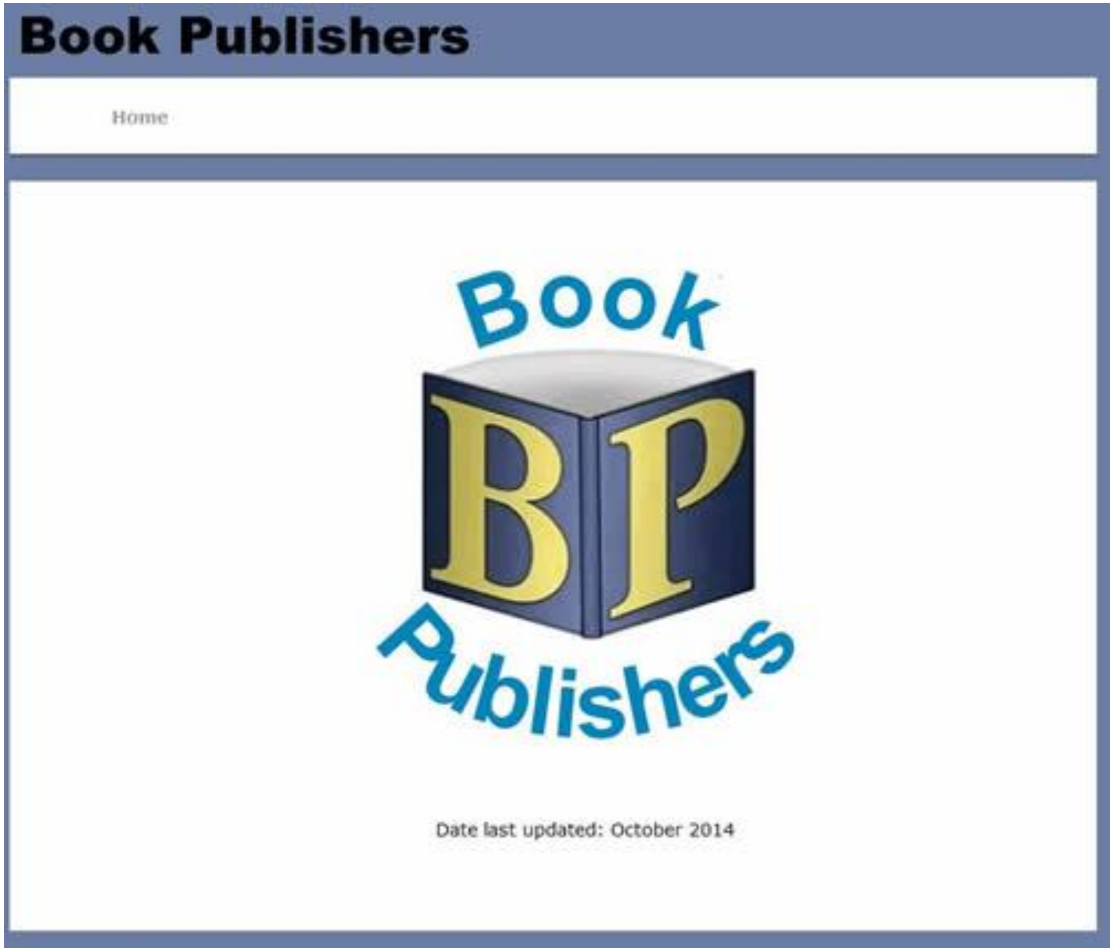
Authentication: **No Authentication**

Project File List



```
Solution 'Publishers1' (1 project)
  Publishers1
    Properties
    References
    App_Data
    App_Start
    Content
    Controllers
    fonts
    Models
    Scripts
    Views
      Home
        About.cshtml
        Contact.cshtml
        Index.cshtml
      Shared
        _Layout.cshtml
        Error.cshtml
      _ViewStart.cshtml
      Web.config
    favicon.ico
    Global.asax
    packages.config
    Project_Readme.html
    Web.config
```

Book Publisher’s Example



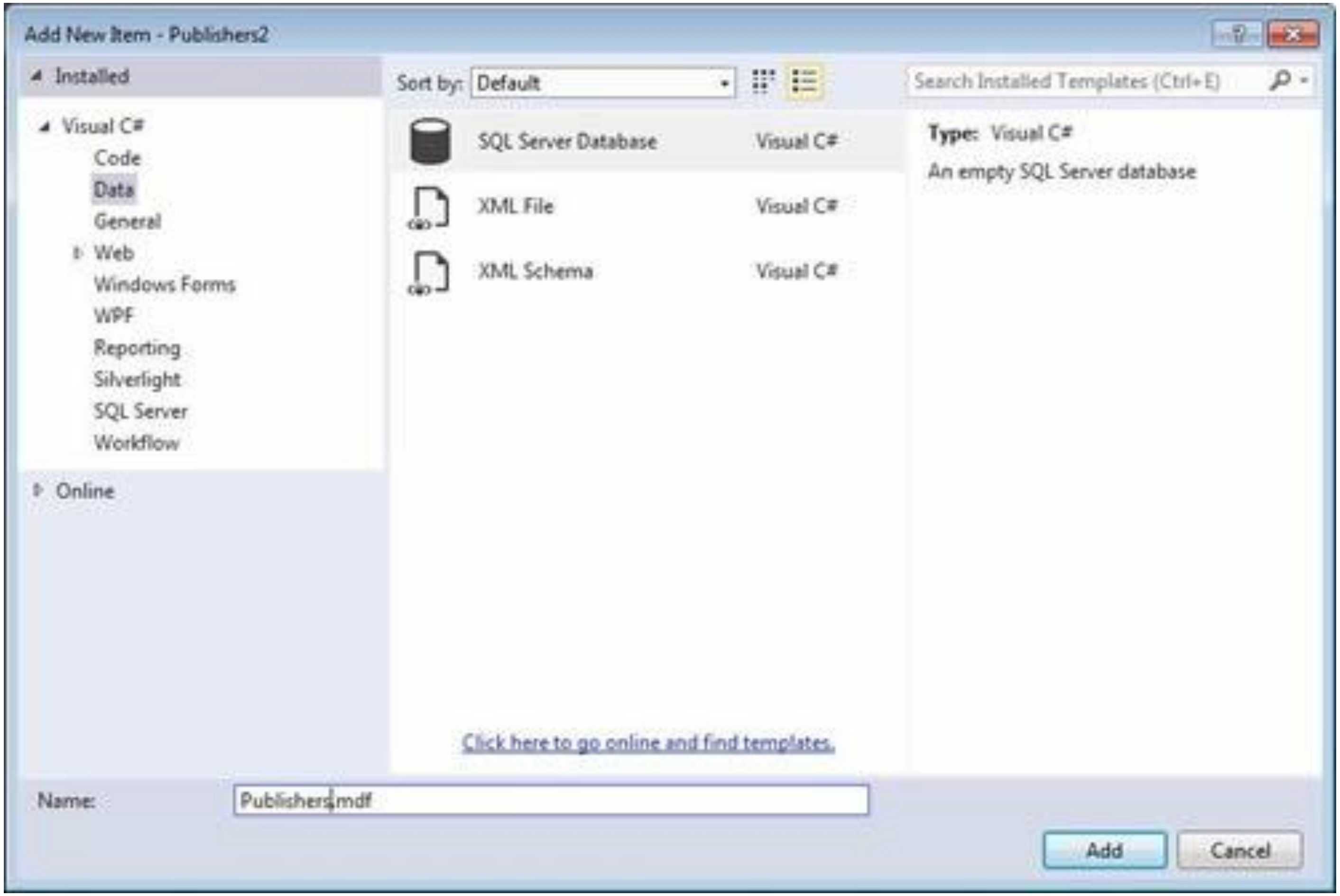
Book Publishers

Date last updated: October 2014

MONASH University

Creating an SQL Server database

Embedded SQL Server



Add New Item - Publishers2

SQL Server Database	Visual C#
XML File	Visual C#
XML Schema	Visual C#

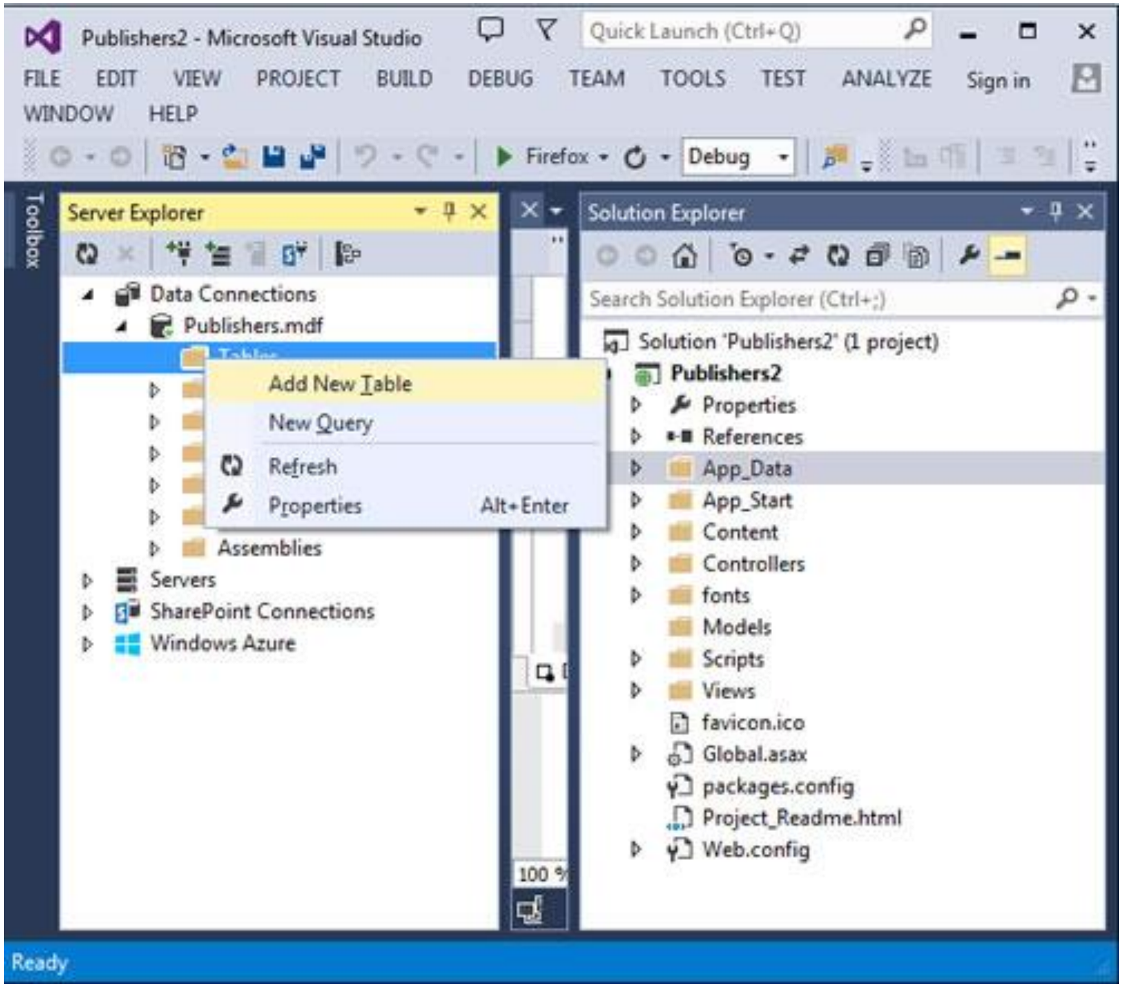
Type: Visual C#

An empty SQL Server database

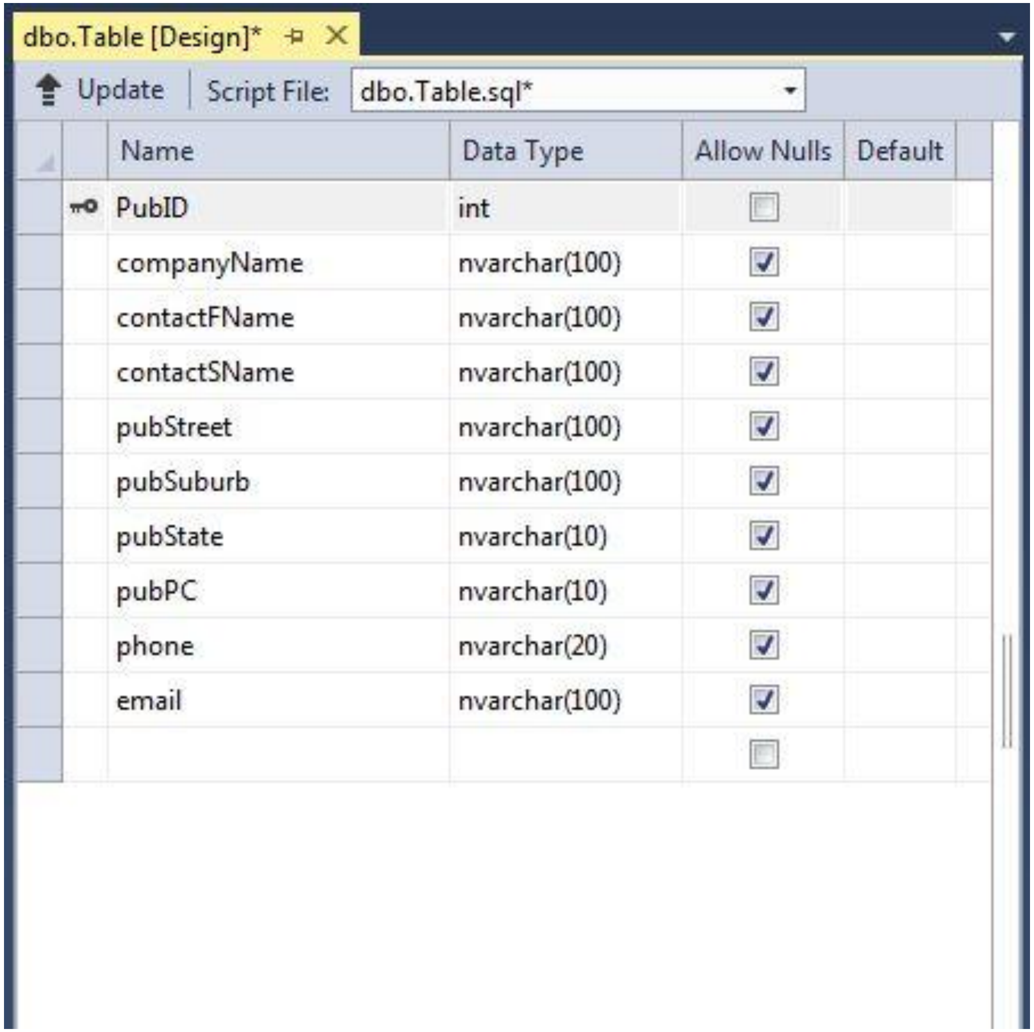
Click here to go online and find templates.

Name: Publishers.mdf

Add a New Table- Data Connections View

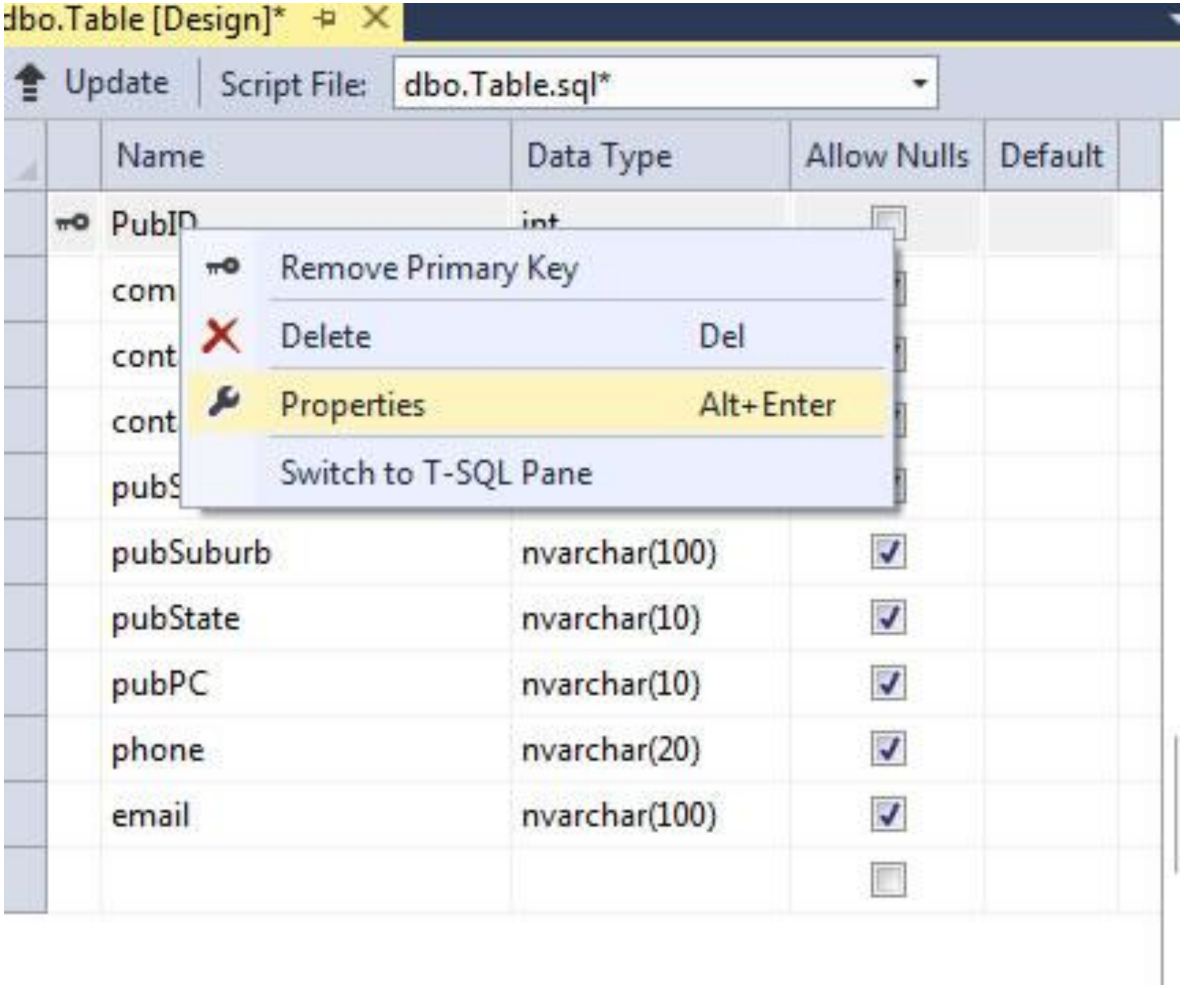


Input Database Design



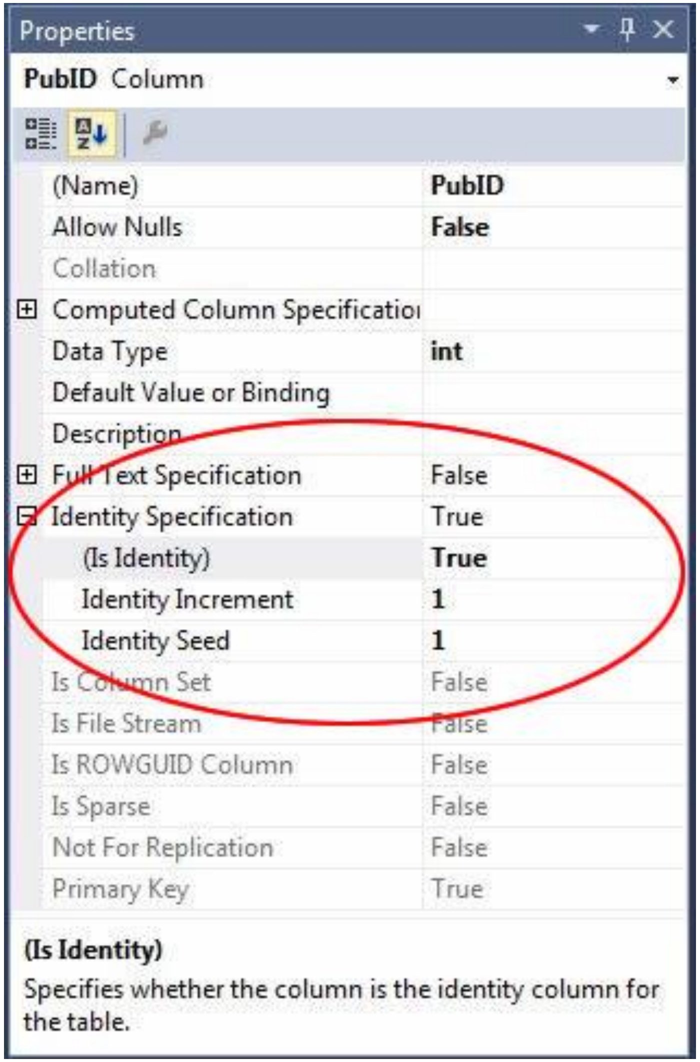
	Name	Data Type	Allow Nulls	Default
🔑	PubID	int	□	
	companyName	nvarchar(100)	☑	
	contactFName	nvarchar(100)	☑	
	contactSName	nvarchar(100)	☑	
	pubStreet	nvarchar(100)	☑	
	pubSuburb	nvarchar(100)	☑	
	pubState	nvarchar(10)	☑	
	pubPC	nvarchar(10)	☑	
	phone	nvarchar(20)	☑	
	email	nvarchar(100)	☑	
			□	

Set auto increment- right click properties of PubID



Name	Data Type	Allow Nulls	Default
PubID	int	☐	
com			
cont			
cont			
pubS			
pubSuburb	nvarchar(100)	☑	
pubState	nvarchar(10)	☑	
pubPC	nvarchar(10)	☑	
phone	nvarchar(20)	☑	
email	nvarchar(100)	☑	
		☐	

Set Identity as True, Increment and Seed



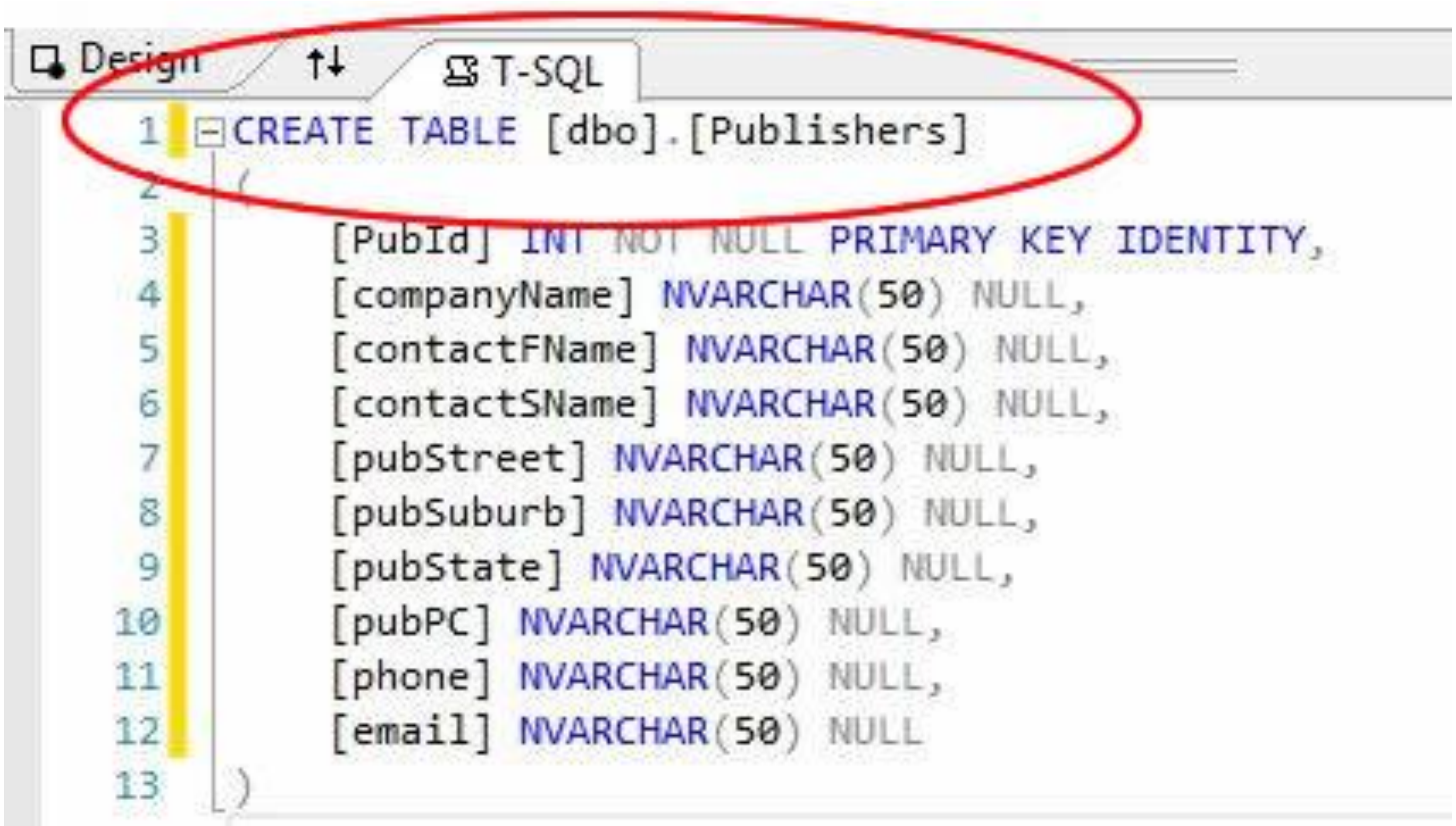
PubID Column

(Name)	**PubID**
Allow Nulls	**False**
Collation	
⊞ Computed Column Specificatio	
Data Type	**int**
Default Value or Binding	
Description	
⊞ Full Text Specification	False
⊟ Identity Specification	True
(Is Identity)	**True**
Identity Increment	**1**
Identity Seed	**1**
Is Column Set	False
Is File Stream	False
Is ROWGUID Column	False
Is Sparse	False
Not For Replication	False
Primary Key	True

(Is Identity)

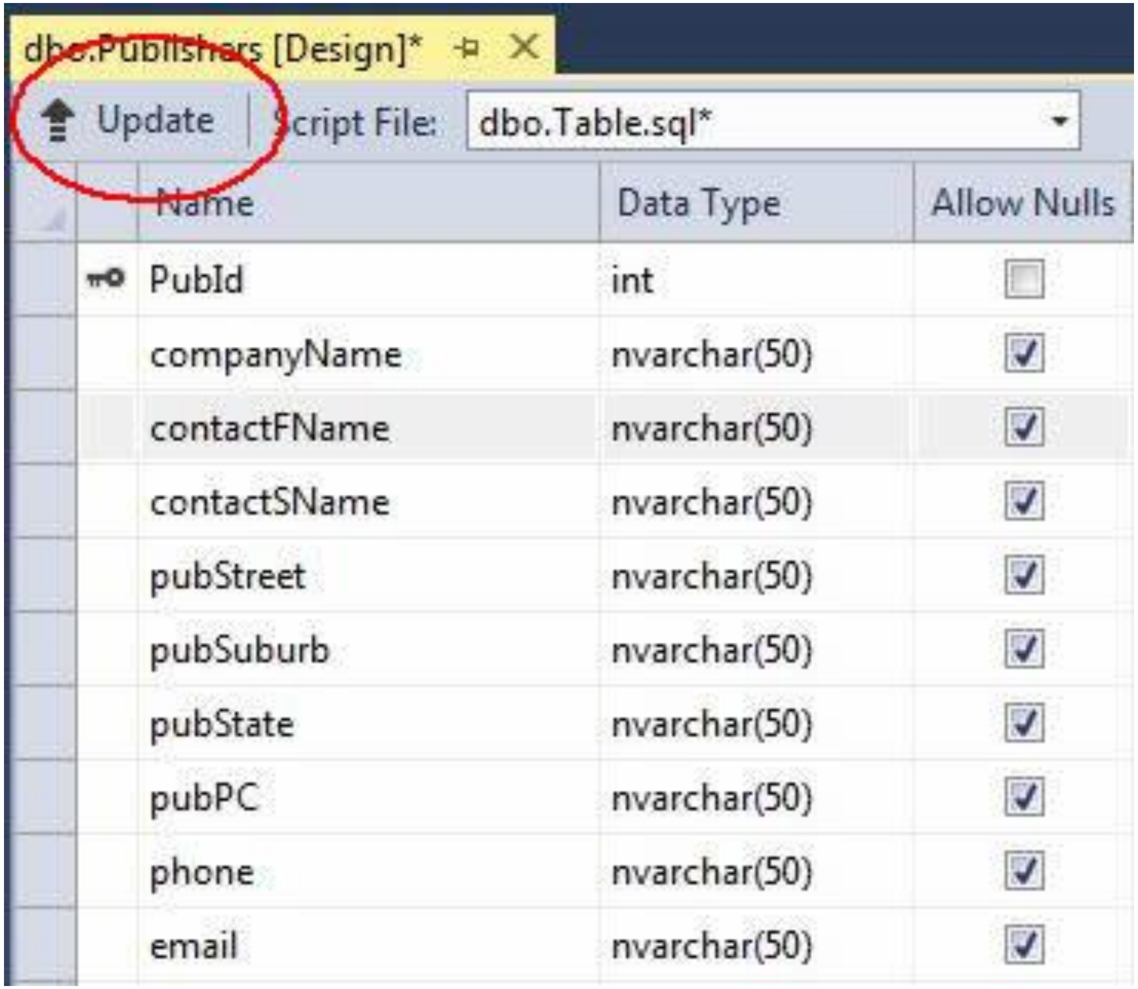
Specifies whether the column is the identity column for the table.

Change Table Name- in T-SQL pane



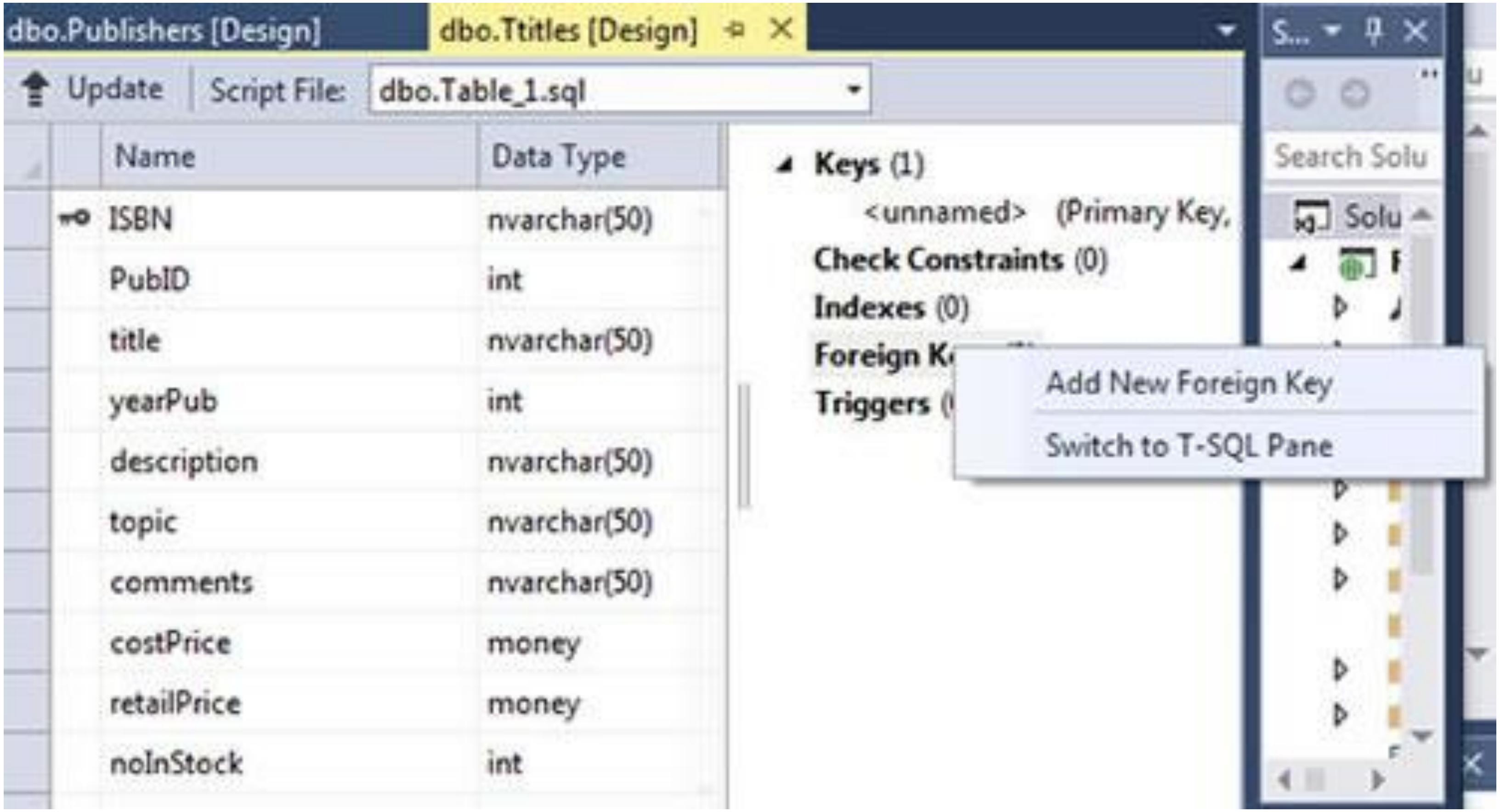
```
CREATE TABLE [dbo].[Publishers]
(
    [PubId] INT NOT NULL PRIMARY KEY IDENTITY,
    [companyName] NVARCHAR(50) NULL,
    [contactFName] NVARCHAR(50) NULL,
    [contactSName] NVARCHAR(50) NULL,
    [pubStreet] NVARCHAR(50) NULL,
    [pubSuburb] NVARCHAR(50) NULL,
    [pubState] NVARCHAR(50) NULL,
    [pubPC] NVARCHAR(50) NULL,
    [phone] NVARCHAR(50) NULL,
    [email] NVARCHAR(50) NULL
)
```

Update the Table Definition



Name	Data Type	Allow Nulls
PubId	int	☐
companyName	nvarchar(50)	☑
contactFName	nvarchar(50)	☑
contactSName	nvarchar(50)	☑
pubStreet	nvarchar(50)	☑
pubSuburb	nvarchar(50)	☑
pubState	nvarchar(50)	☑
pubPC	nvarchar(50)	☑
phone	nvarchar(50)	☑
email	nvarchar(50)	☑

Same for Book Table, then Add Foreign Key



Name	Data Type
ISBN	nvarchar(50)
PubID	int
title	nvarchar(50)
yearPub	int
description	nvarchar(50)
topic	nvarchar(50)
comments	nvarchar(50)
costPrice	money
retailPrice	money
noInStock	int

Edit code with correct table and column

```
CONSTRAINT [FK_Ttitles_Publishers] FOREIGN KEY ([PubID])
    REFERENCES [Publishers]([PubID])
```

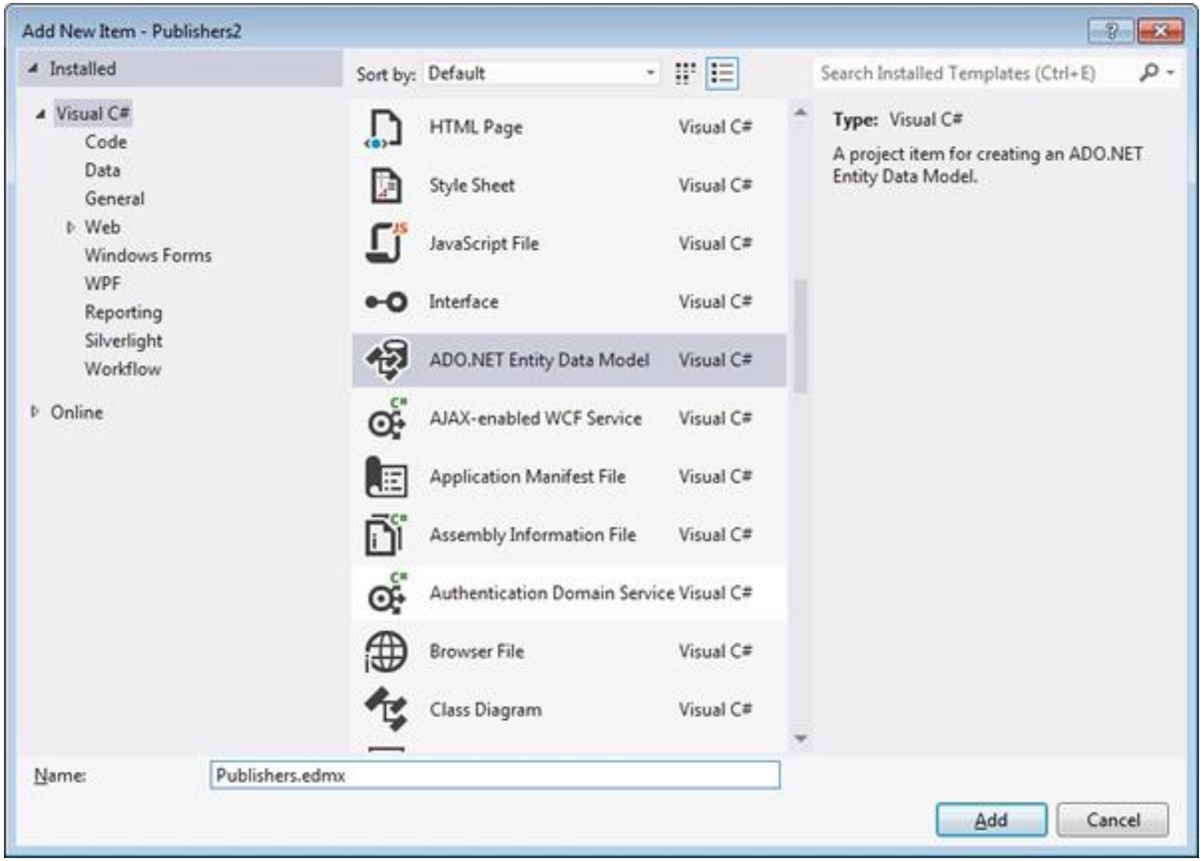
Add the update and delete cascade constraint

```
CONSTRAINT [FK_Ttitles_Publishers] FOREIGN KEY ([PubID])
    REFERENCES [Publishers]([PubID]) on DELETE CASCADE on UPDATE
    CASCADE
```

PRESS the UPDATE button to ensure the tables and constraints are updated

Models- Using the database

Create a Model from the Database- use Entity



Add New Item - Publishers2

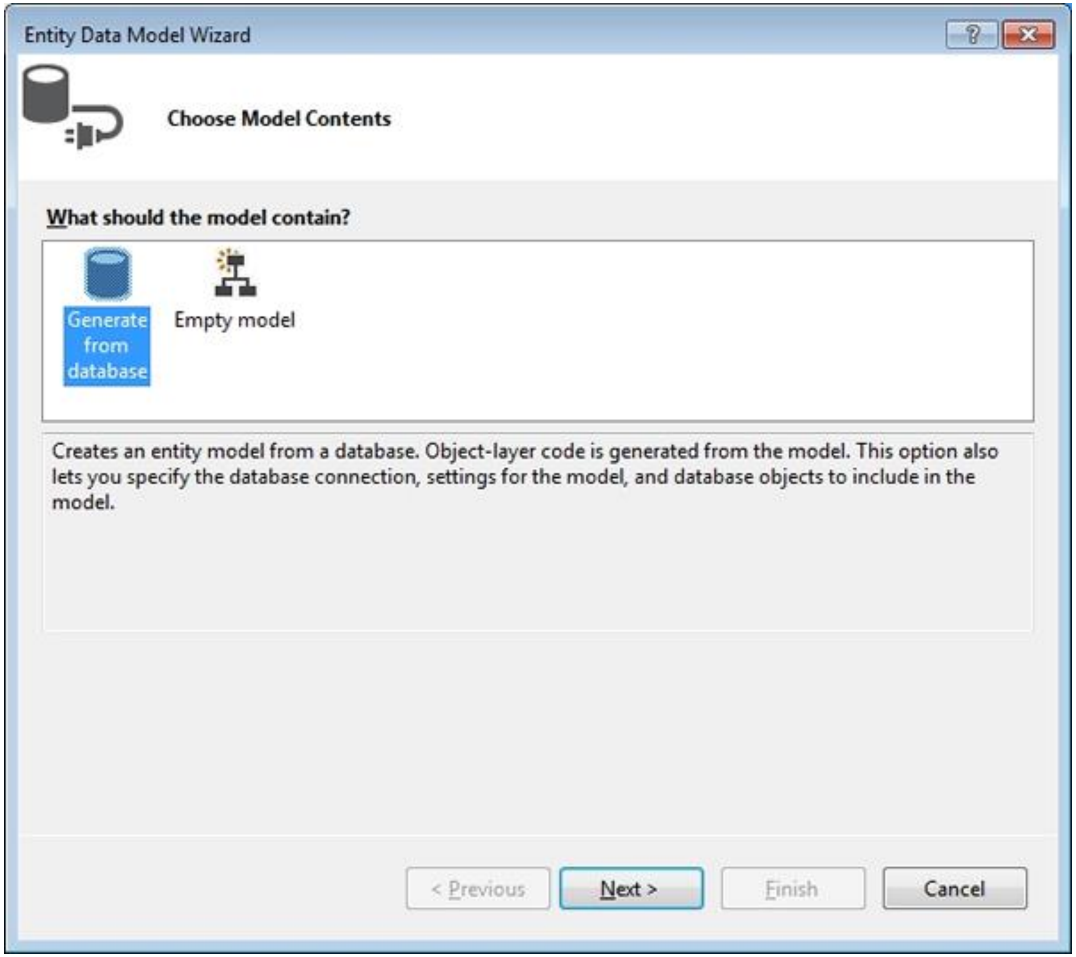
HTML Page	Visual C#
Style Sheet	Visual C#
JavaScript File	Visual C#
Interface	Visual C#
ADO.NET Entity Data Model	Visual C#
AJAX-enabled WCF Service	Visual C#
Application Manifest File	Visual C#
Assembly Information File	Visual C#
Authentication Domain Service	Visual C#
Browser File	Visual C#
Class Diagram	Visual C#

Type: Visual C#

A project item for creating an ADO.NET Entity Data Model.

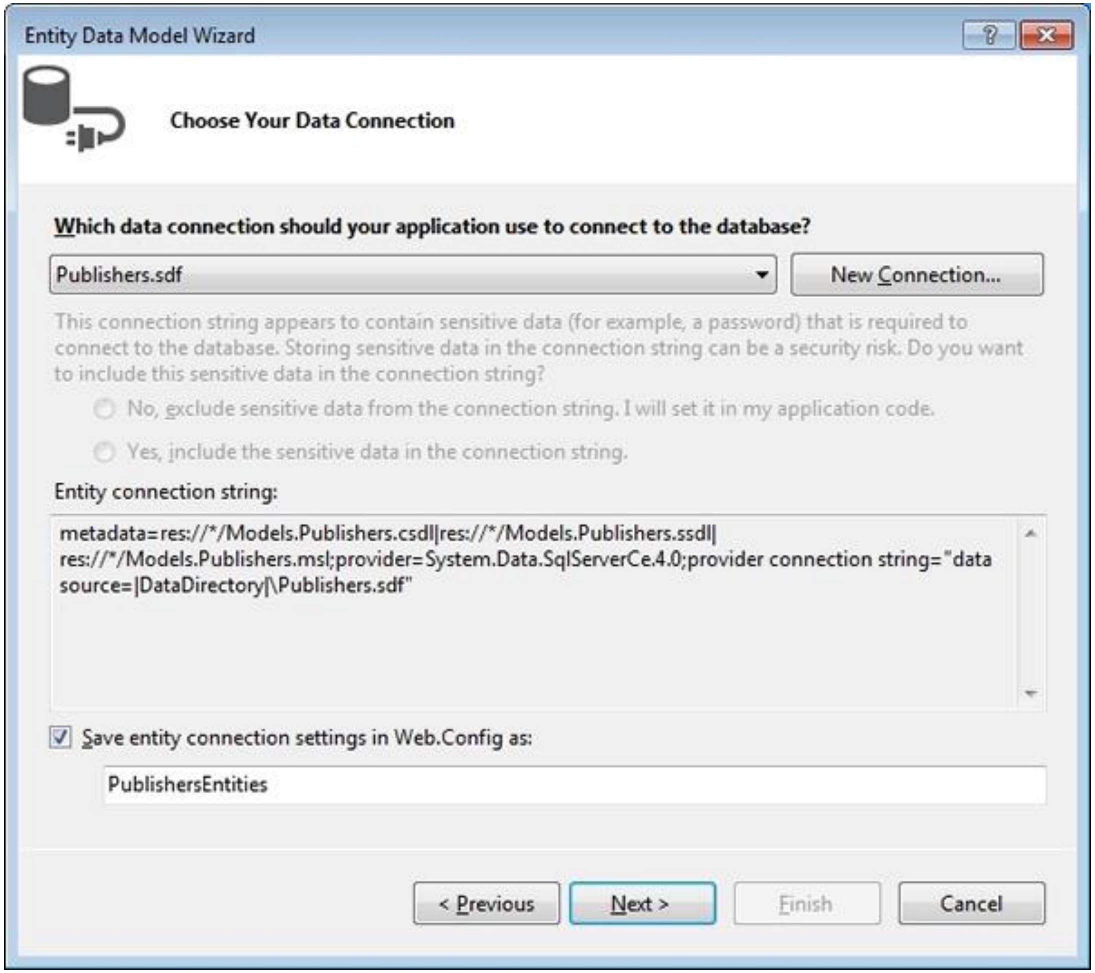
Name: Publishers.edmx

Generate from the Database



Creates an entity model from a database. Object-layer code is generated from the model. This option also lets you specify the database connection, settings for the model, and database objects to include in the model.

Choose the Connection



Choose Your Data Connection

Which data connection should your application use to connect to the database?

Publishers.sdf ▾

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

○ No, exclude sensitive data from the connection string. I will set it in my application code.

○ Yes, include the sensitive data in the connection string.

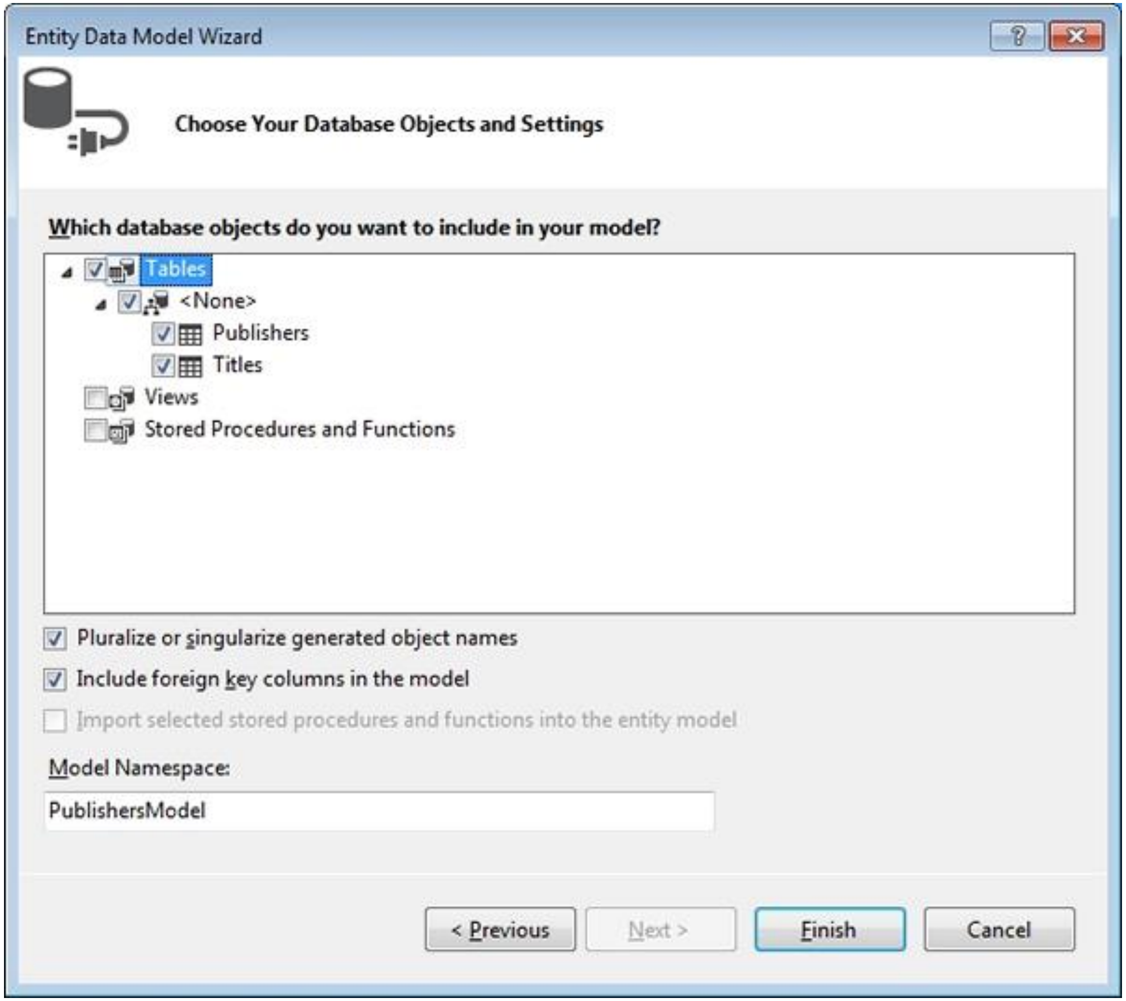
Entity connection string:

```
metadata=res://*/Models.Publishers.csdl|res://*/Models.Publishers.ssdl|
res://*/Models.Publishers.msl;provider=System.Data.SqlServerCe.4.0;provider connection string="data
source=|DataDirectory|\Publishers.sdf"
```

☑ Save entity connection settings in Web.Config as:

PublishersEntities

Choose the Tables



Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

- ☑ Tables
  - ☑ <None>
    - ☑ Publishers
    - ☑ Titles
- □ Views
- □ Stored Procedures and Functions

☑ Pluralize or singularize generated object names

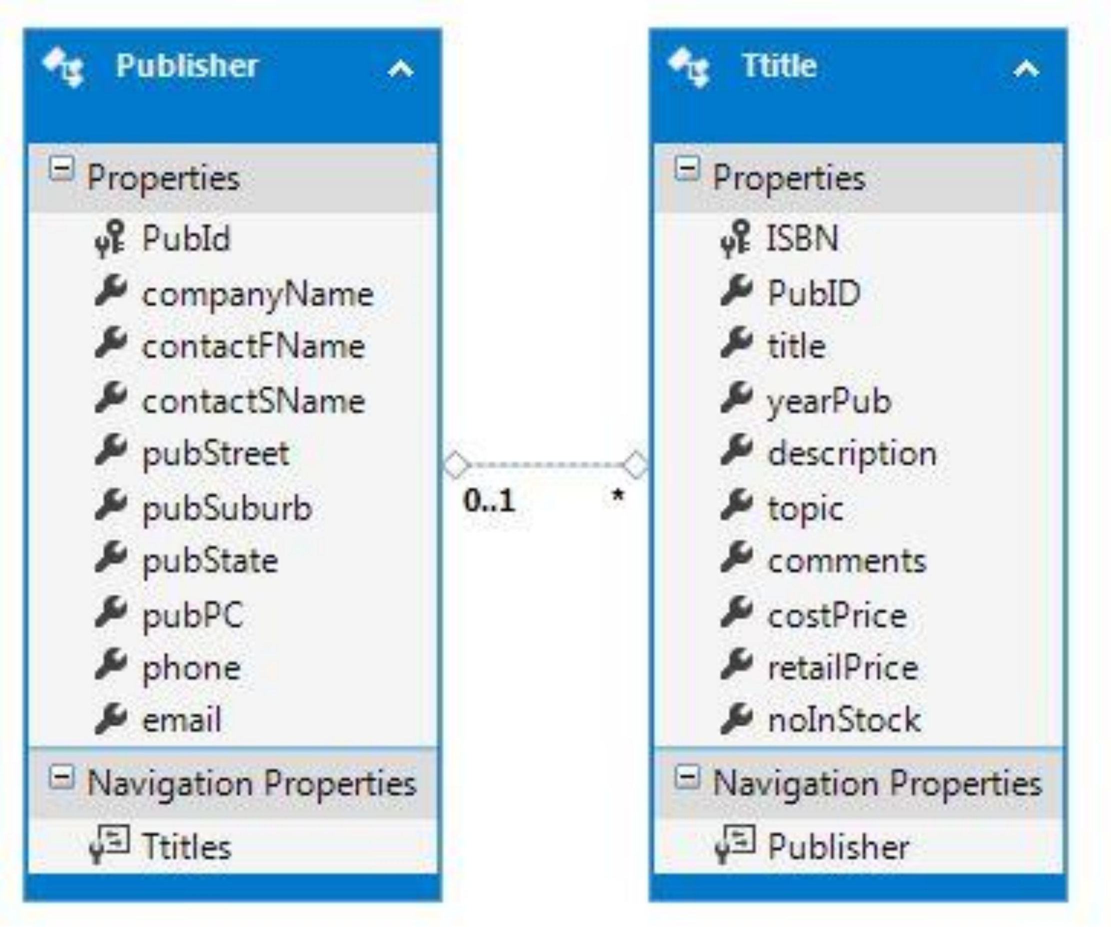
☑ Include foreign key columns in the model

□ Import selected stored procedures and functions into the entity model

Model Namespace:

PublishersModel

Model Classes Generated

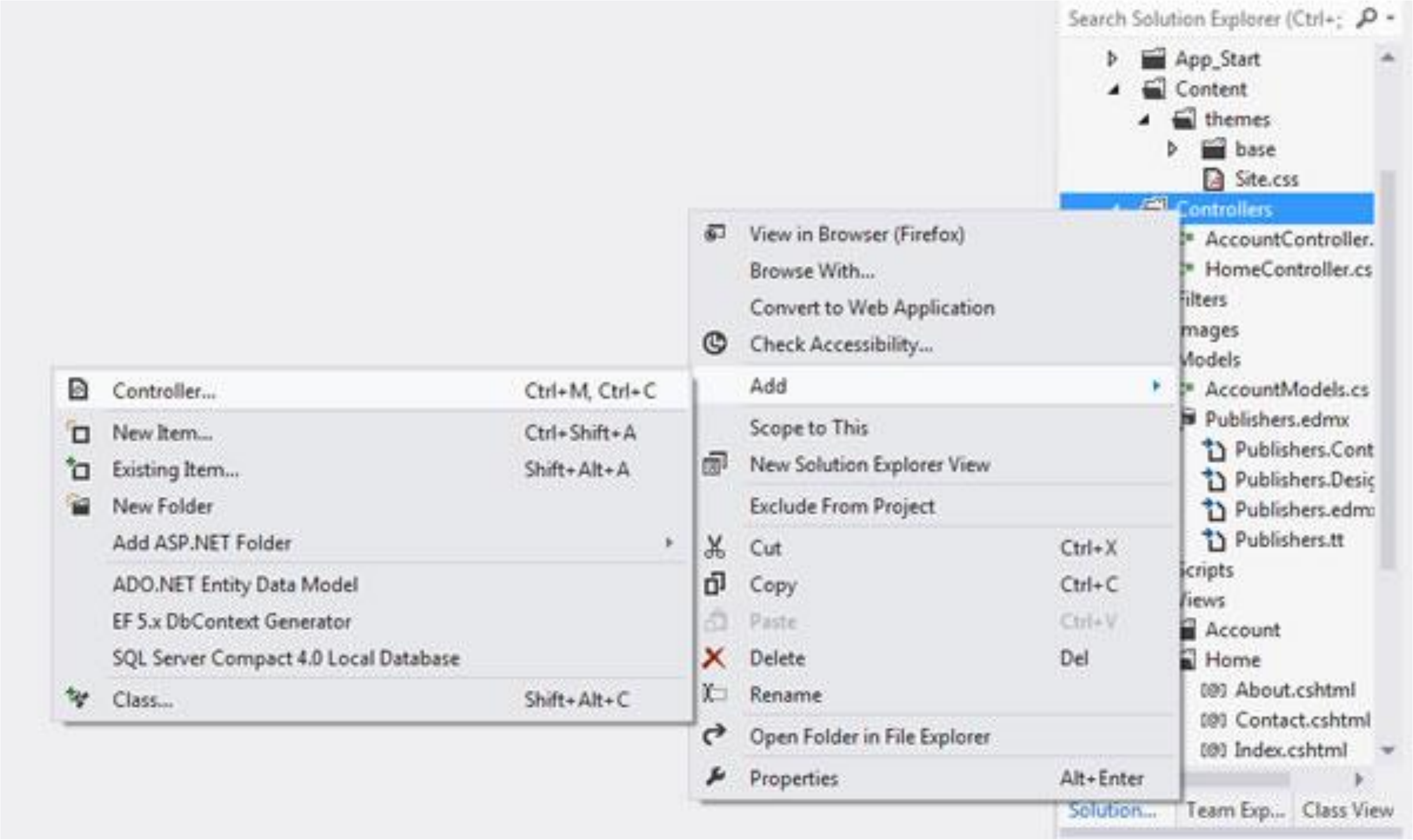


Build the Application

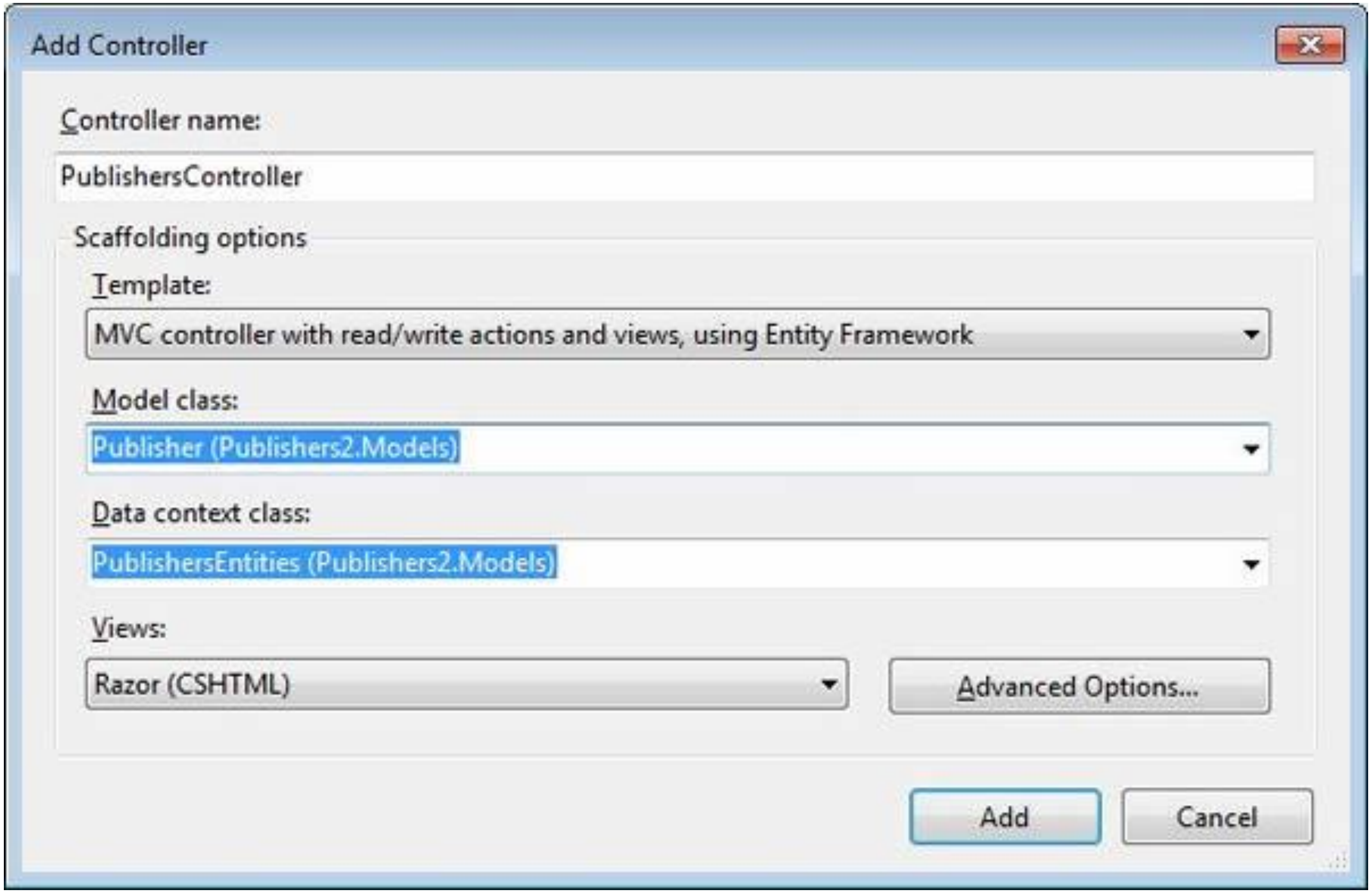
Before other items can be added the application should be built so that the relevant code is generated

Generating Controllers

Add a Controller- Right Click



Choose the Model and Entities



Add Controller

Controller name: PublishersController

Scaffolding options

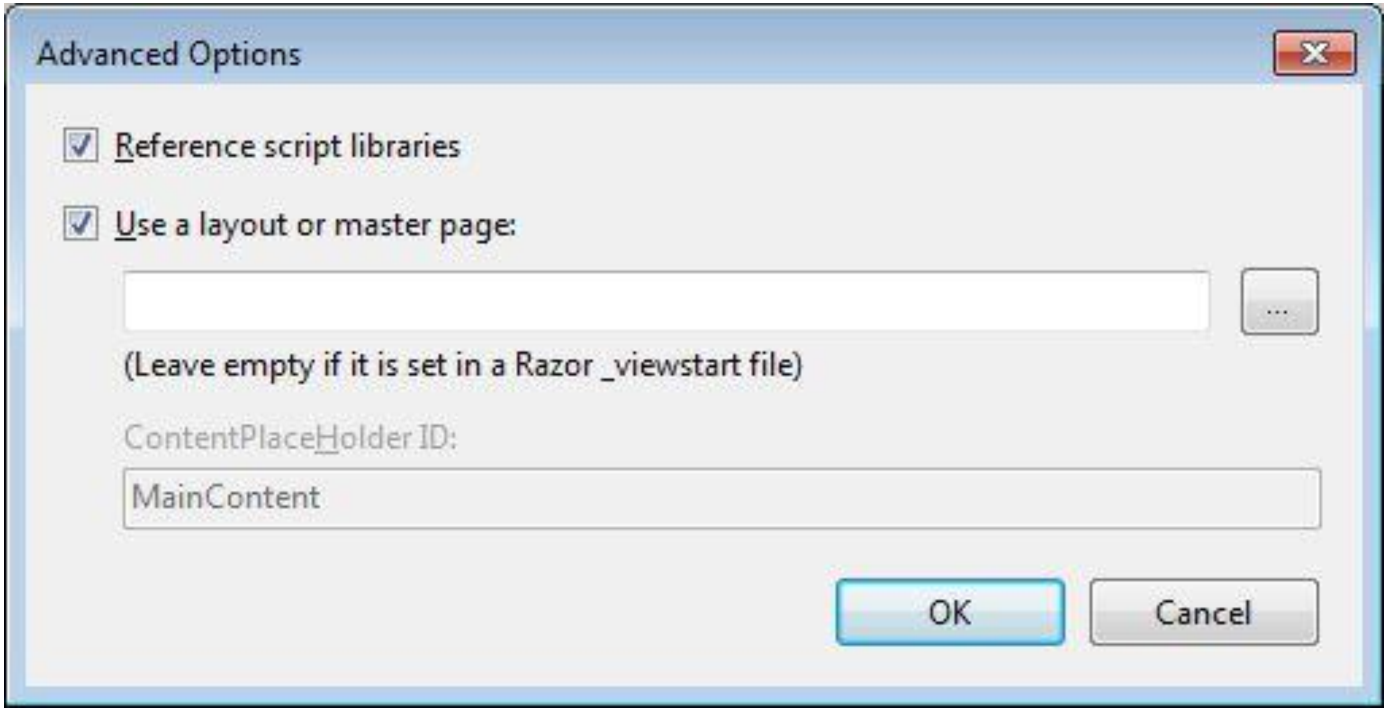
Template: MVC controller with read/write actions and views, using Entity Framework

Model class: Publisher (Publishers2.Models)

Data context class: PublishersEntities (Publishers2.Models)

Views: Razor (CSHTML)

Link in the Layout



Advanced Options

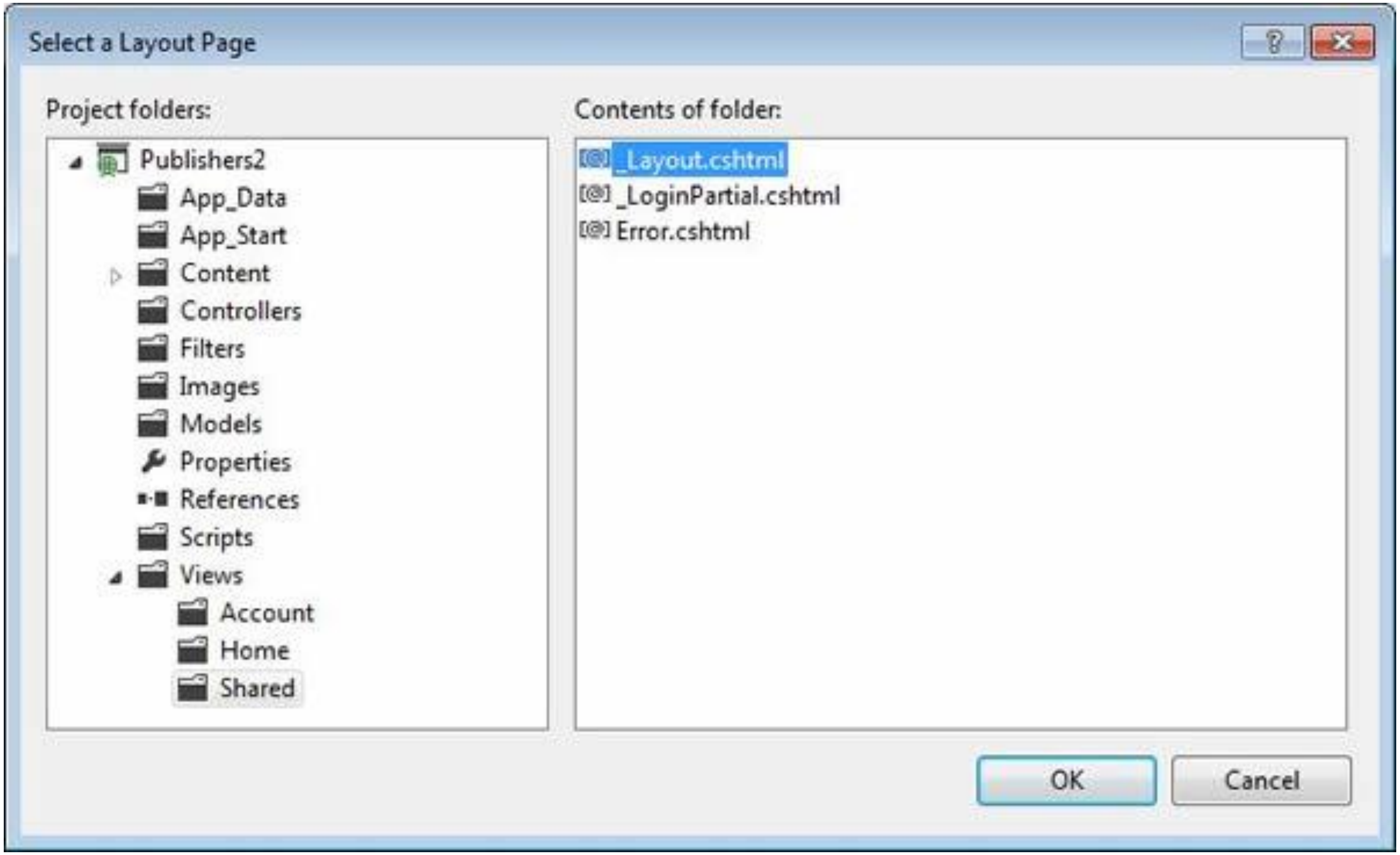
☑ Reference script libraries

☑ Use a layout or master page:

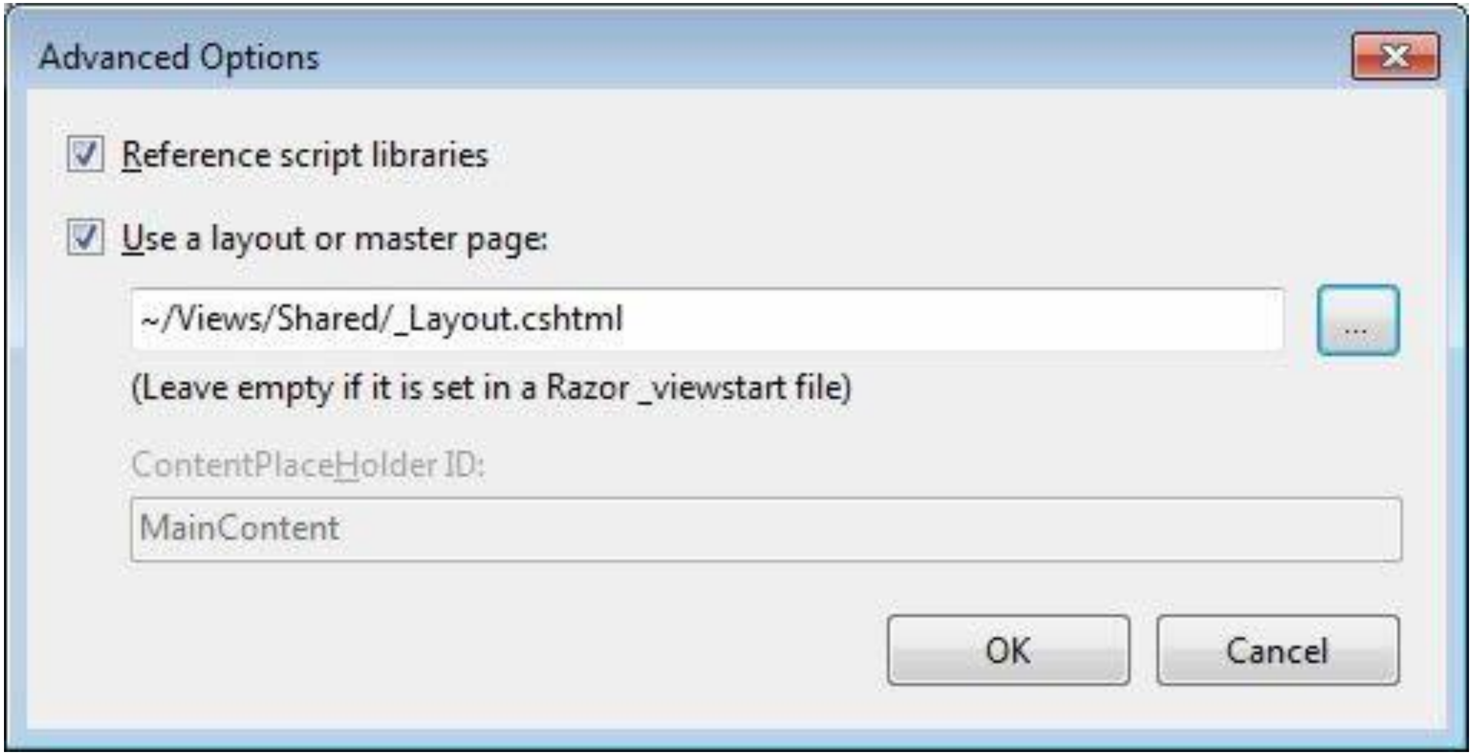
(Leave empty if it is set in a Razor _viewstart file)

ContentPlaceHolder ID: MainContent

Choose layout page



Complete Controller Creation



Advanced Options

☑ Reference script libraries

☑ Use a layout or master page:

~/Views/Shared/_Layout.cshtml

(Leave empty if it is set in a Razor _viewstart file)

ContentPlaceHolder ID:

MainContent

Created Controller

Autogenerated Controllers

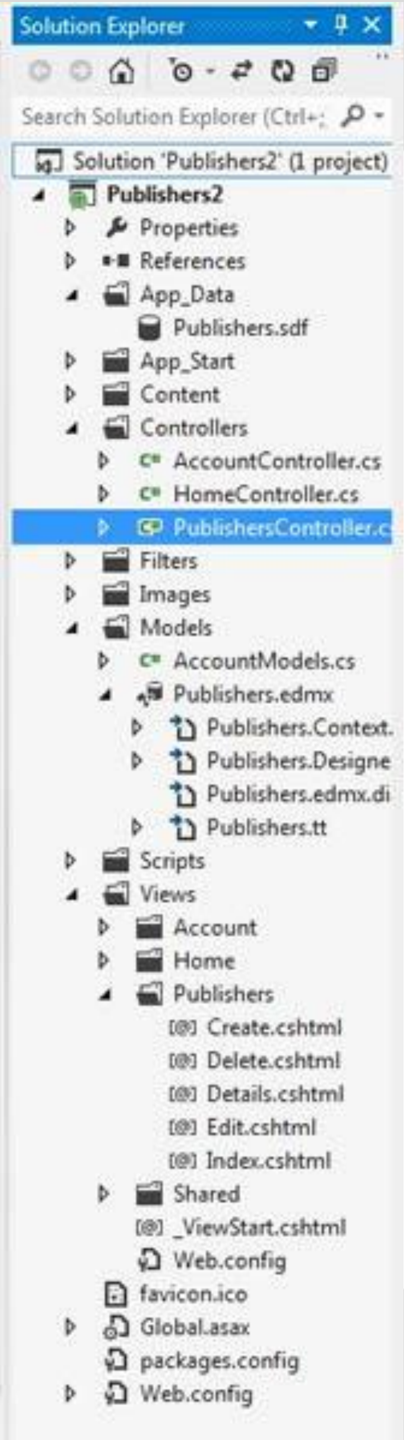
PublishersController class contains methods such as:

- *Index*
- *Details*
- *Create*
- *Edit*
- *Delete*

Plus the associated *Views*

Autogenerated Views

Views are Also Autogenerated



```
Solution 'Publishers2' (1 project)
  Publishers2
    Properties
    References
    App_Data
      Publishers.sdf
    App_Start
    Content
    Controllers
      AccountController.cs
      HomeController.cs
      PublishersController.c
    Filters
    Images
    Models
      AccountModels.cs
      Publishers.edmx
        Publishers.Context.
        Publishers.Designe
        Publishers.edmx.di
        Publishers.tt
    Scripts
    Views
      Account
      Home
      Publishers
        Create.cshtml
        Delete.cshtml
        Details.cshtml
        Edit.cshtml
        Index.cshtml
      Shared
      _ViewStart.cshtml
      Web.config
    favicon.ico
    Global.asax
    packages.config
    Web.config
```

Views files can be edited

```
<td>
    @Html.ActionLink("Edit", "Edit", new { /* id=item.PrimaryKey */ }) |
    @Html.ActionLink("Details", "Details", new { /* id=item.PrimaryKey */ }) |
    @Html.ActionLink("Delete", "Delete", new { /* id=item.PrimaryKey */ })
</td>
```

Edit so code is active

```
<td>
   @Html.ActionLink("Edit", "Edit", new { id=item.PubID }) |
    @Html.ActionLink("Details", "Details", new { id=item.PubID }) |
    @Html.ActionLink("Delete", "Delete", new { id=item.PubID })
</td>
```

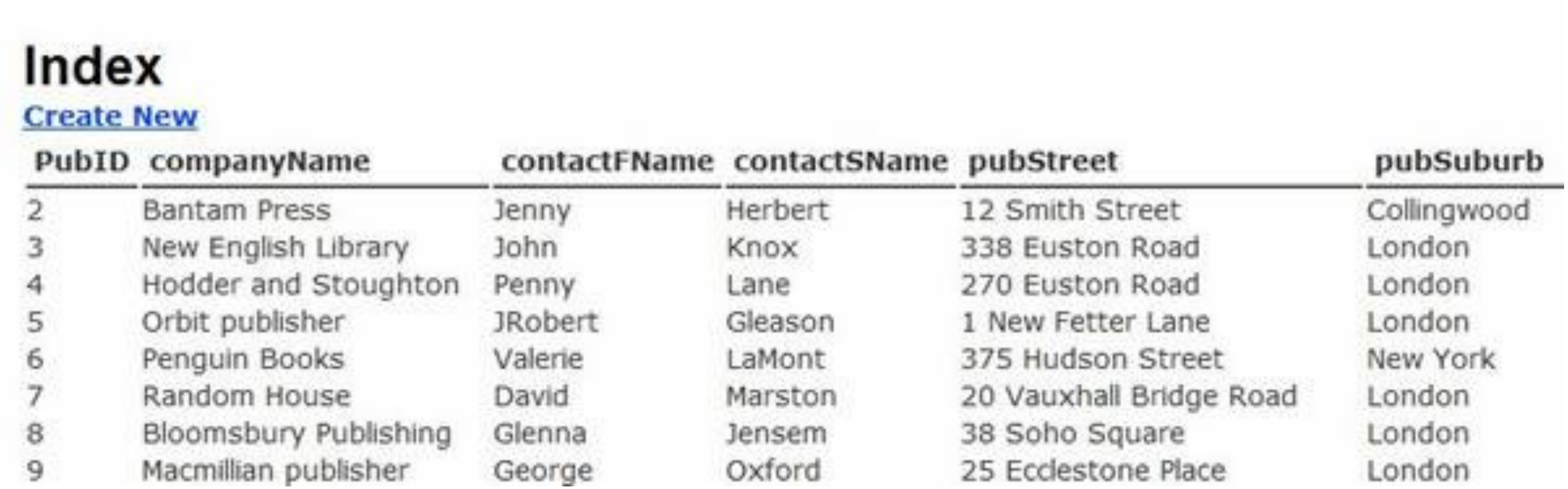
Editing Layout File

To see this View, edit *layout* file:

@Html.ActionLink("Publishers", "Index", "Publishers")

Autogenerated Application

Default Listing



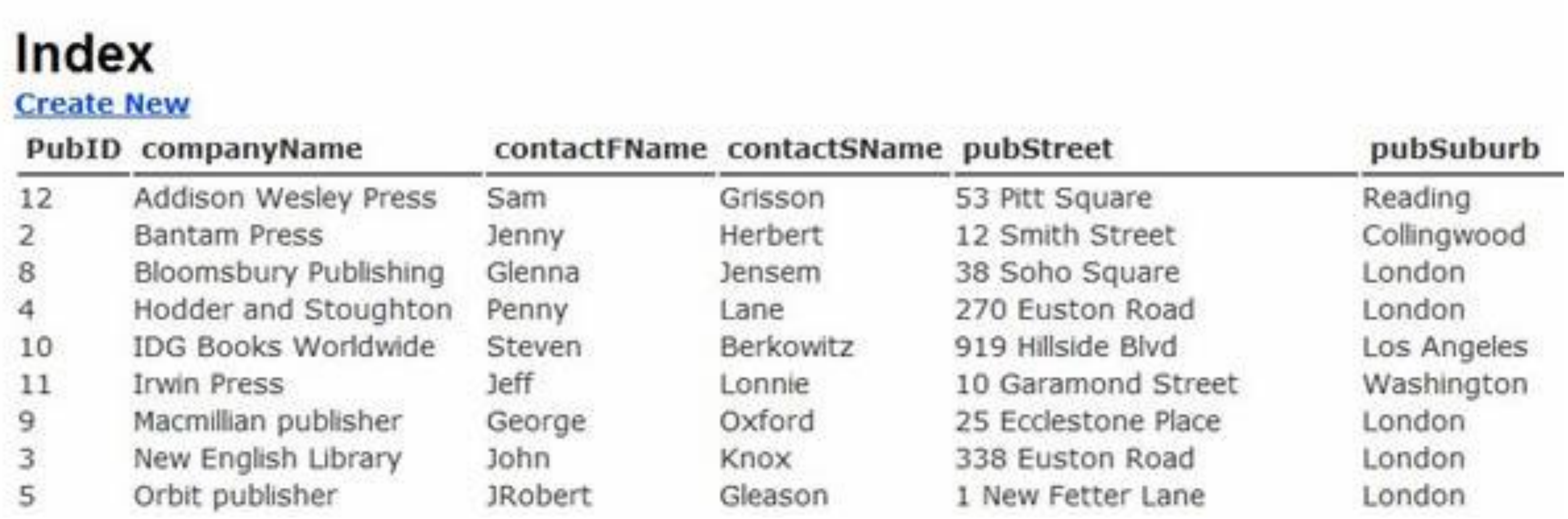
Index

Create New

PubID	companyName	contactFName	contactSName	pubStreet	pubSuburb
2	Bantam Press	Jenny	Herbert	12 Smith Street	Collingwood
3	New English Library	John	Knox	338 Euston Road	London
4	Hodder and Stoughton	Penny	Lane	270 Euston Road	London
5	Orbit publisher	JRobert	Gleason	1 New Fetter Lane	London
6	Penguin Books	Valerie	LaMont	375 Hudson Street	New York
7	Random House	David	Marston	20 Vauxhall Bridge Road	London
8	Bloomsbury Publishing	Glenna	Jensem	38 Soho Square	London
9	Macmillian publisher	George	Oxford	25 Ecclestone Place	London

Sorted Listings

```
public ActionResult Index() {
return
  View(db.Publishers.OrderBy(p=>p.companyName).ToList());
}
```



Index

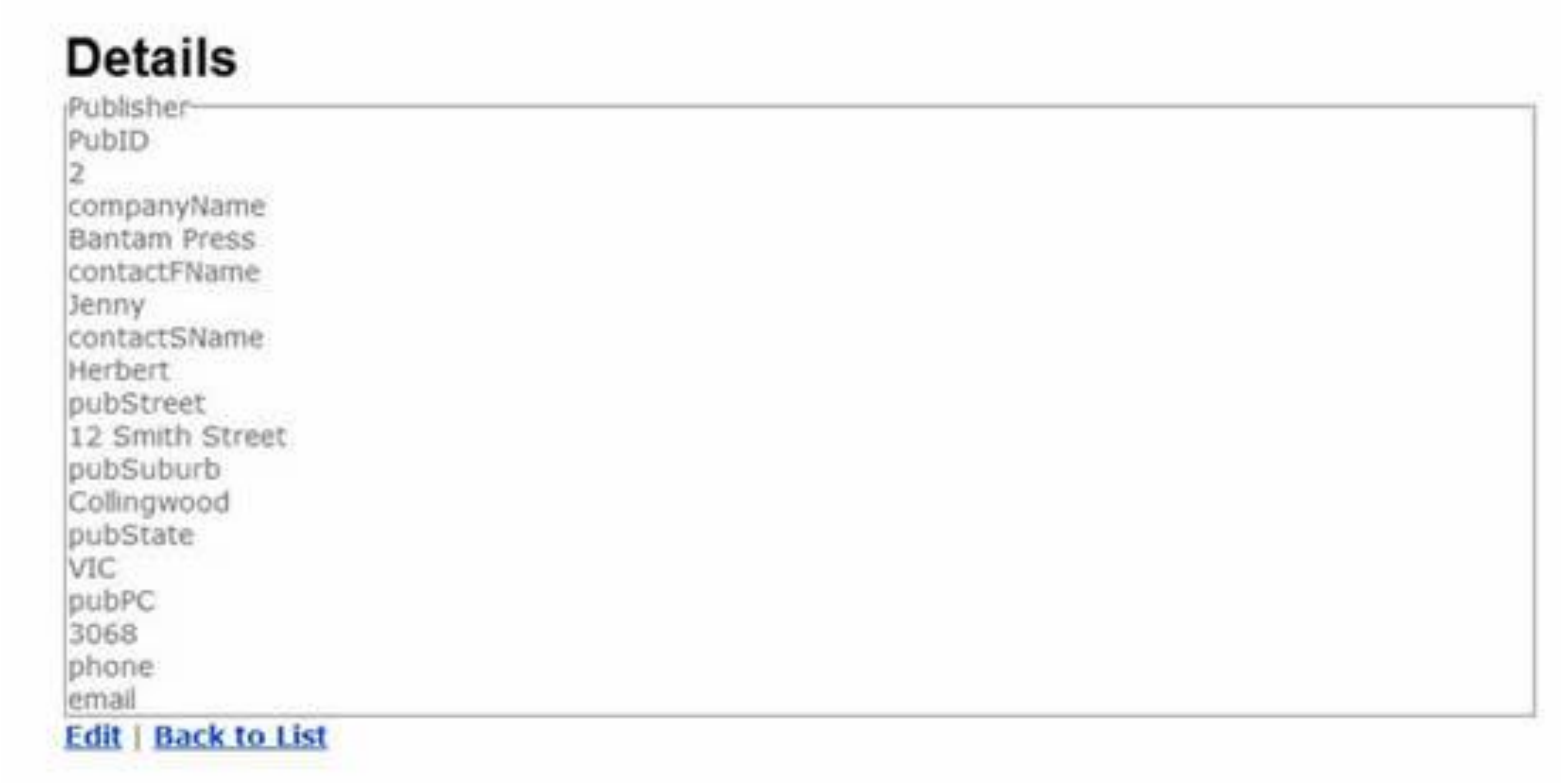
Create New

PubID	companyName	contactFName	contactSName	pubStreet	pubSuburb
12	Addison Wesley Press	Sam	Grisson	53 Pitt Square	Reading
2	Bantam Press	Jenny	Herbert	12 Smith Street	Collingwood
8	Bloomsbury Publishing	Glenna	Jensem	38 Soho Square	London
4	Hodder and Stoughton	Penny	Lane	270 Euston Road	London
10	IDG Books Worldwide	Steven	Berkowitz	919 Hillside Blvd	Los Angeles
11	Irwin Press	Jeff	Lonnie	10 Garamond Street	Washington
9	Macmillian publisher	George	Oxford	25 Ecclestone Place	London
3	New English Library	John	Knox	338 Euston Road	London
5	Orbit publisher	JRobert	Gleason	1 New Fetter Lane	London

MONASH University

Changing Look and Feel

Details View



Details

Publisher

PubID

2

companyName

Bantam Press

contactFName

Jenny

contactSName

Herbert

pubStreet

12 Smith Street

pubSuburb

Collingwood

pubState

VIC

pubPC

3068

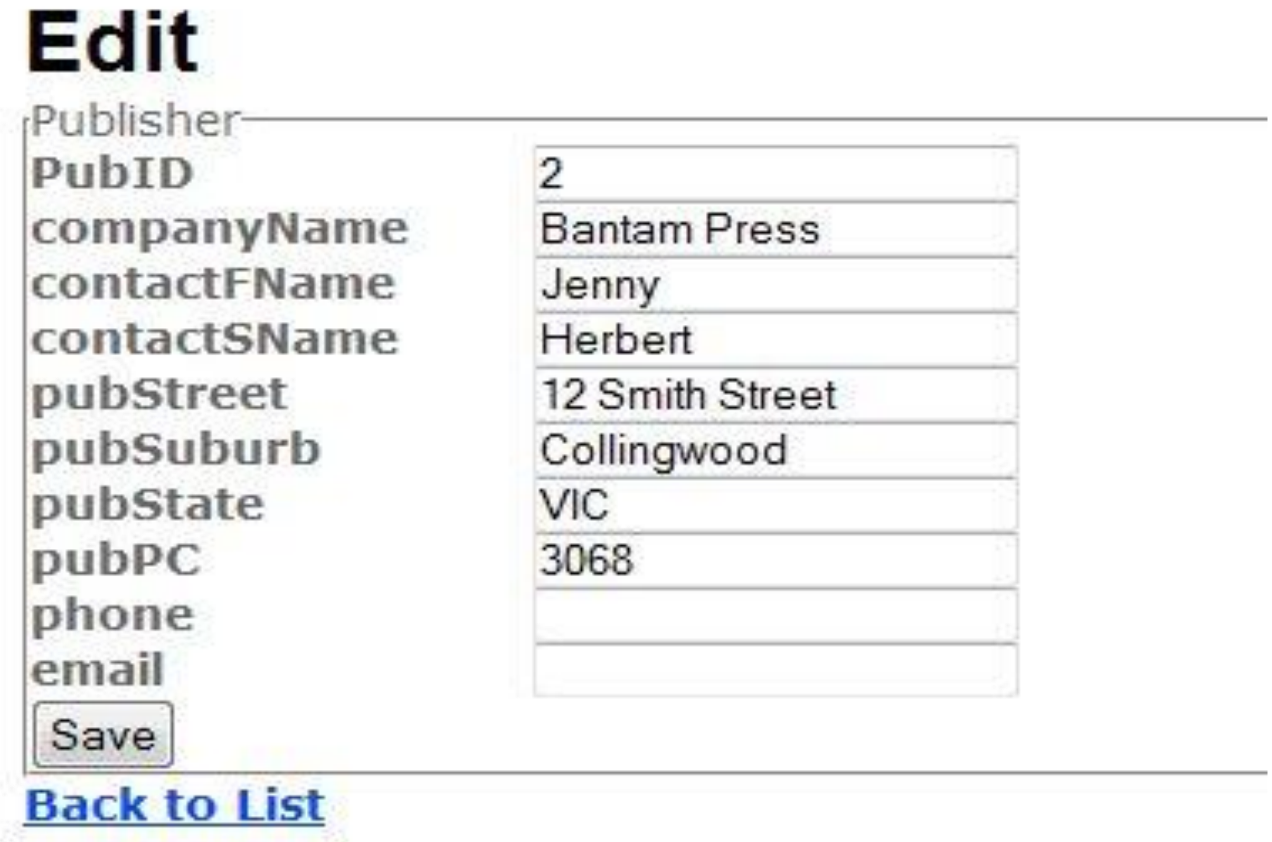
phone

email

Adding Buttons rather than Links

```
<a href="@Url.Action("Edit","Publishers",
        new {id =Model.PubID})"><button>Edit</button></a> </td>
  <td class="button">
<a href="@Url.Action("Delete","Publishers",
 new {id =Model.PubID})"><button>Delete</button></a> </td>
  <td class="button">
 <a href="@Url.Action("Index","Publishers")"> <button>Back to
  List</button></a>
```

Editing a Record



Edit

Publisher

PubID	2
companyName	Bantam Press
contactFName	Jenny
contactSName	Herbert
pubStreet	12 Smith Street
pubSuburb	Collingwood
pubState	VIC
pubPC	3068
phone	
email	

Changing Header Fields

```
@Html.LabelFor(model => model.companyName)
```

Can be changed to a specific header:

```
@Html.Label("Company Name")
```

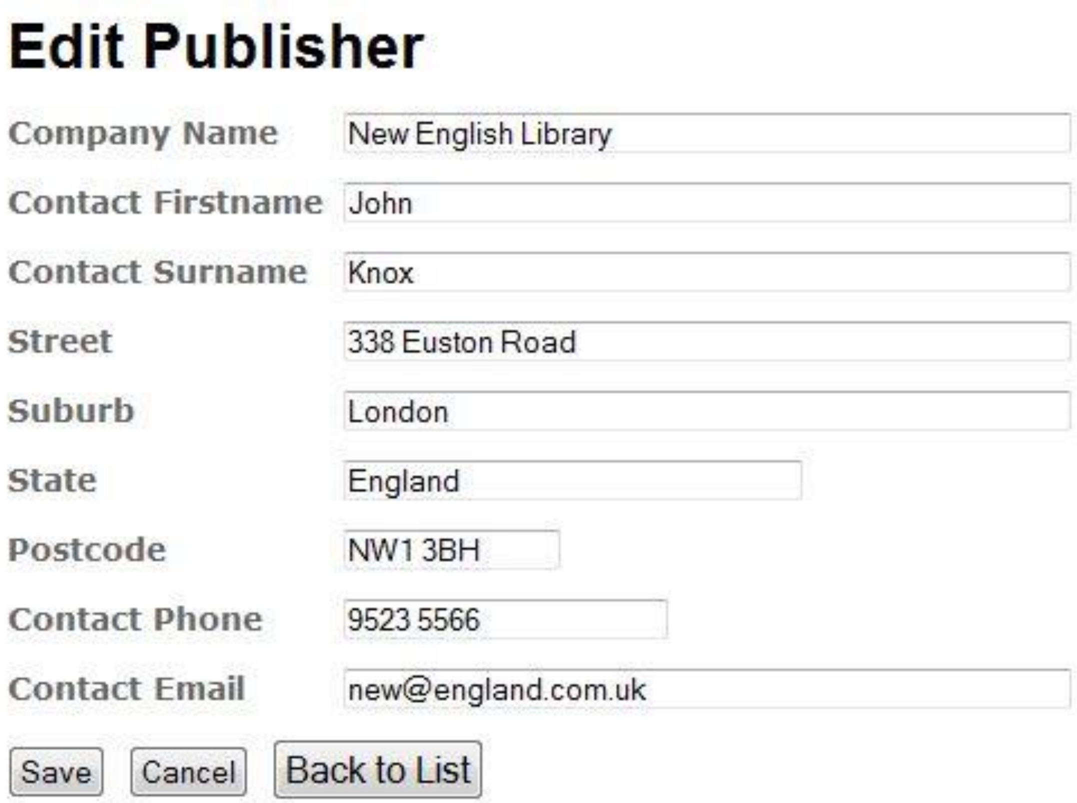
Textbox Sizes

```
@Html.EditorFor(model => model.companyName)
```

Can be changed to:

```
@Html.TextBoxFor(model => model.companyName, new {
  size=50 })
```

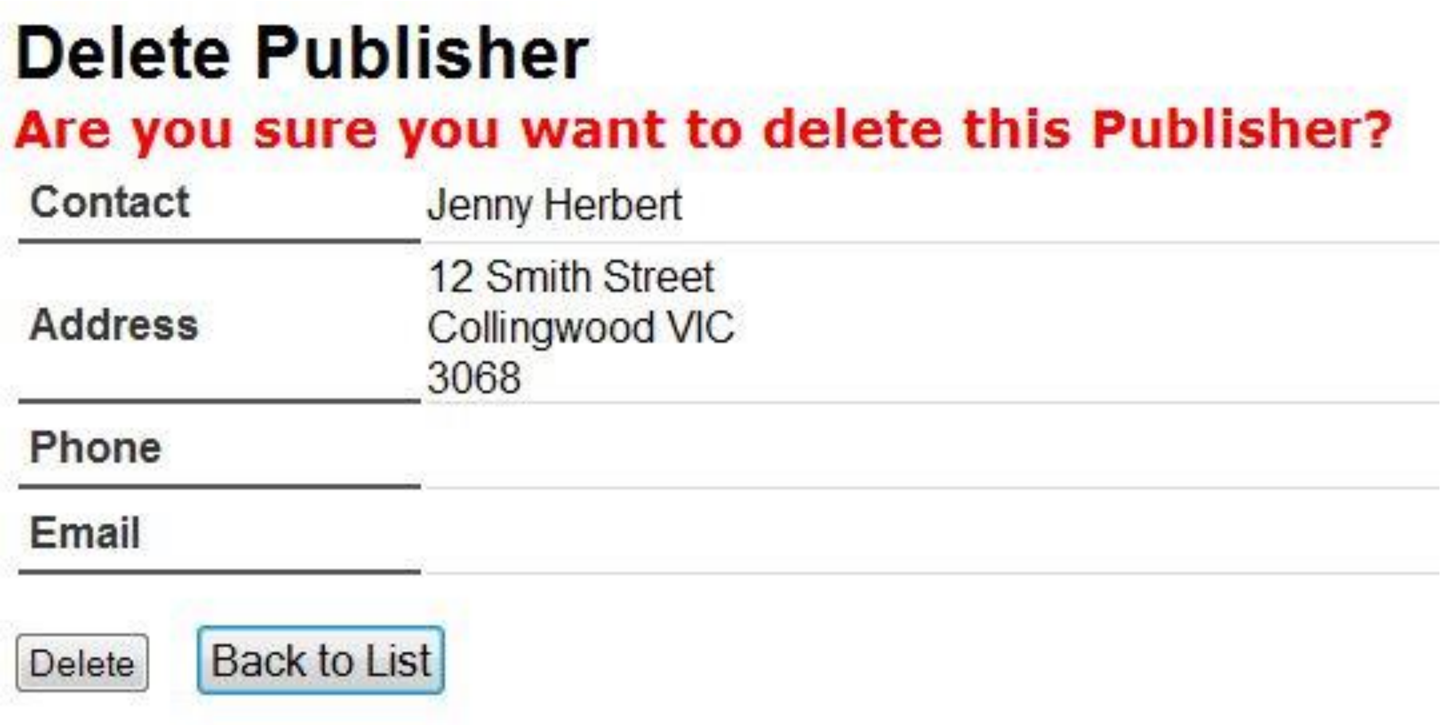
Changed Layout



Edit Publisher

Company Name	New English Library
Contact Firstname	John
Contact Surname	Knox
Street	338 Euston Road
Suburb	London
State	England
Postcode	NW1 3BH
Contact Phone	9523 5566
Contact Email	new@england.com.uk

Delete Publisher

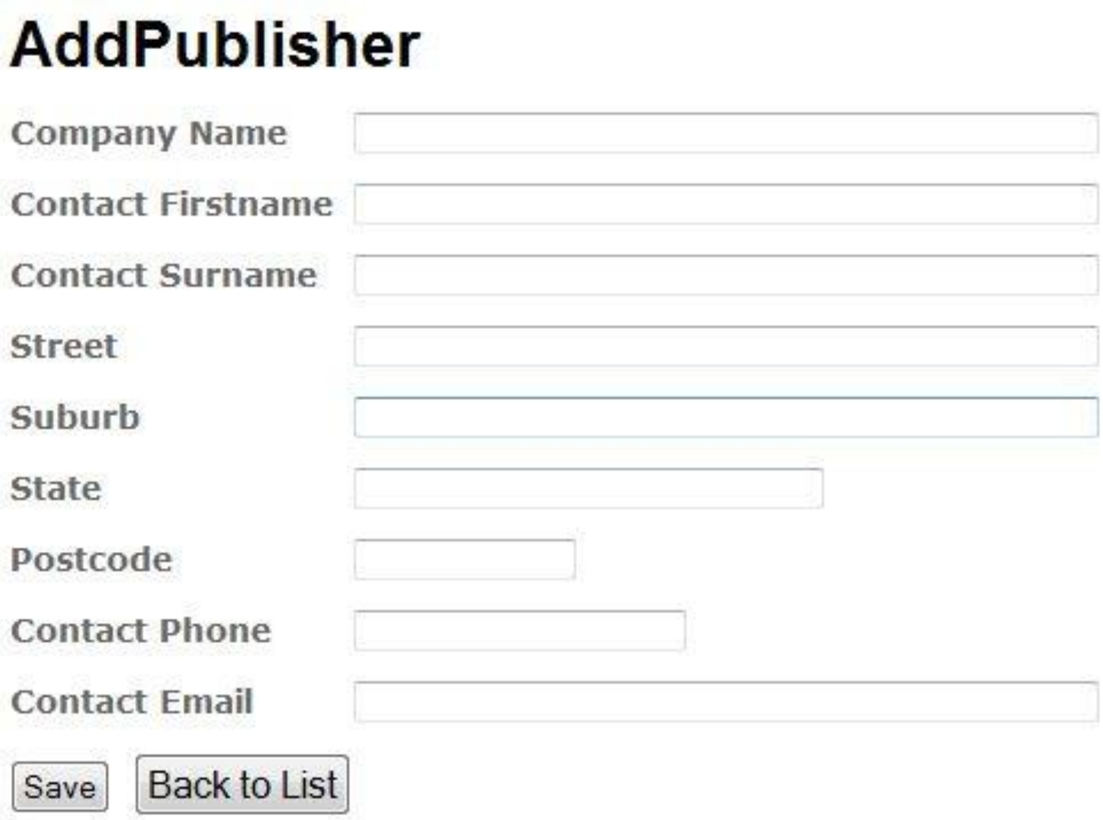


Delete Publisher

Are you sure you want to delete this Publisher?

Contact	Jenny Herbert
Address	12 Smith Street Collingwood VIC 3068
Phone	
Email	

Add Publisher



AddPublisher

Company Name	
Contact Firstname	
Contact Surname	
Street	
Suburb	
State	
Postcode	
Contact Phone	
Contact Email	

Linked tables- Book titles

Controllers for Linked tables Autogenerated

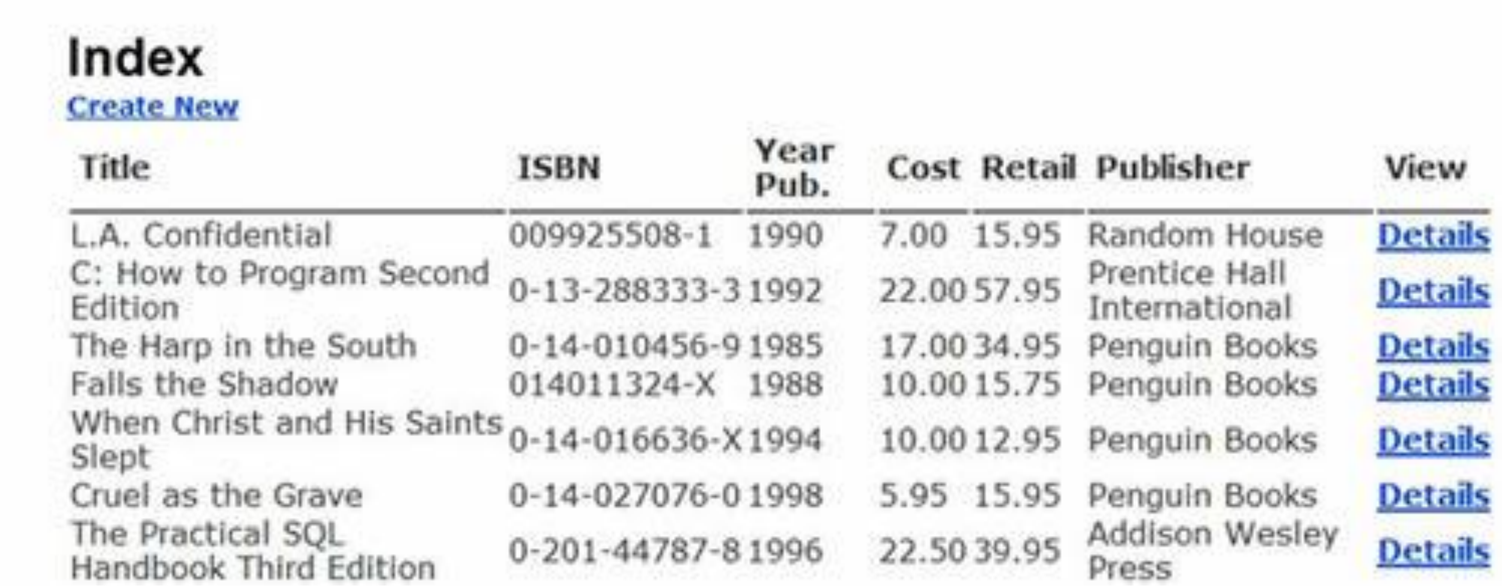
As before controllers for linked tables can be autogenerated.

The associated Entity classes (Publishers) can be included, so that they are linked in correctly

Book Titles

Because of the Foreign Key relationship the Publisher is automatically inserted

@Html.DisplayFor(modelItem => item.Publisher.companyName)



Index

Create New

Title	ISBN	Year Pub.	Cost	Retail	Publisher	View
L.A. Confidential	009925508-1	1990	7.00	15.95	Random House	Details
C: How to Program Second Edition	0-13-288333-3	1992	22.00	57.95	Prentice Hall International	Details
The Harp in the South	0-14-010456-9	1985	17.00	34.95	Penguin Books	Details
Falls the Shadow	014011324-X	1988	10.00	15.75	Penguin Books	Details
When Christ and His Saints Slept	0-14-016636-X	1994	10.00	12.95	Penguin Books	Details
Cruel as the Grave	0-14-027076-0	1998	5.95	15.95	Penguin Books	Details
The Practical SQL Handbook Third Edition	0-201-44787-8	1996	22.50	39.95	Addison Wesley Press	Details

Foreign Key Selection

The Publishers can be selected from a dropdown list.

Edit Action of the Title Controller:

```
ViewBag.PubID = new SelectList(db.Publishers, "PubID",
"companyName", title.PubID);
```

In the View:

```
@Html.DropDownList("PubID", String.Empty)
```

Sorting

Book Titles can be sorted:

in Index action of title controller:

```
return View(titles.OrderBy(t=>t.title1).ToList());
```

Publisher dropdown list sorted:

in create and edit method of Title Controller:

```
ViewBag.PubID = new SelectList(db.Publishers.OrderBy(p =>
   p.companyName), "PubID", "companyName");
```

Summary

1. Introduction
2. Creating an SQL Server Database
3. Models- Using the Database
4. Generating Controllers
5. Generating Views
6. Changing the Look and Feel

What you will do in the Studio

Try topic 12 examples

Read ahead ASP examples from topic 13

Run using Visual Studio 2013

Thanks and See you in the Studio!