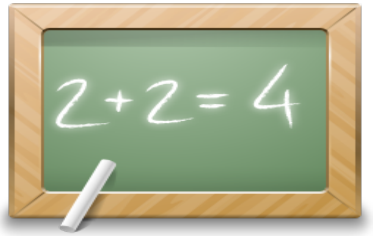
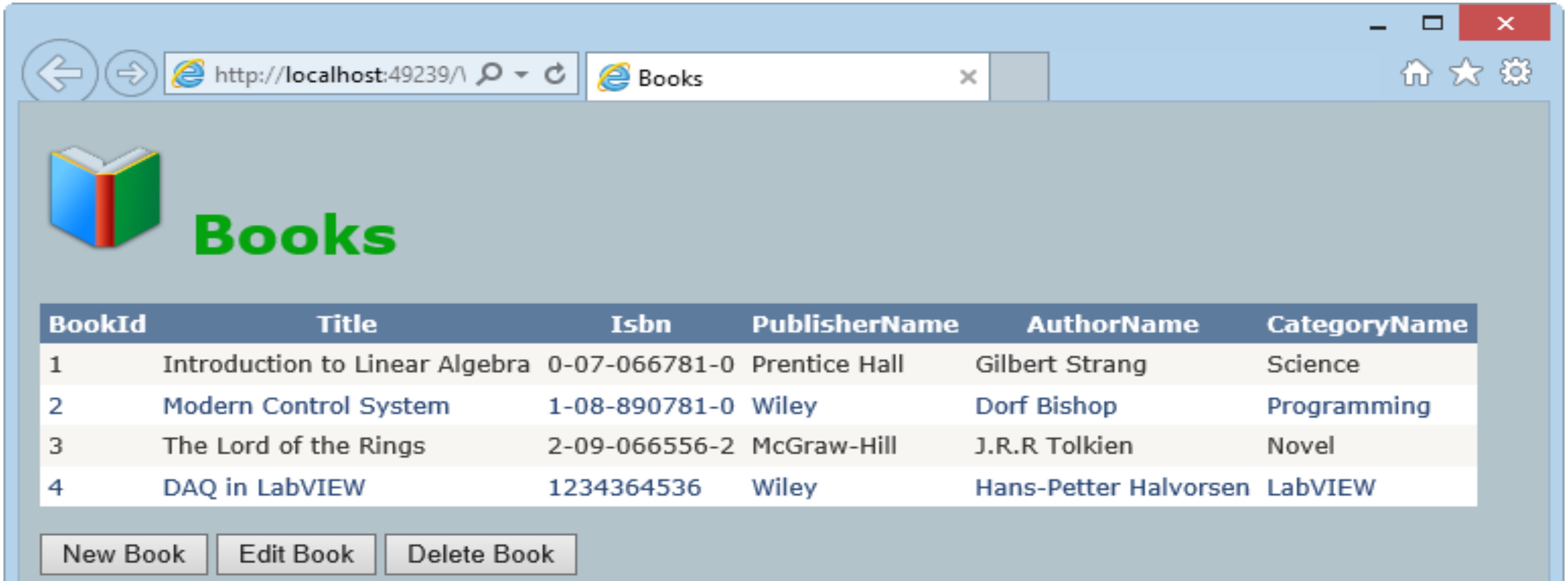


ASP.NET

Create a Database Web Form App



Books

BookId	Title	Isbn	PublisherName	AuthorName	CategoryName
1	Introduction to Linear Algebra	0-07-066781-0	Prentice Hall	Gilbert Strang	Science
2	Modern Control System	1-08-890781-0	Wiley	Dorf Bishop	Programming
3	The Lord of the Rings	2-09-066556-2	McGraw-Hill	J.R.R Tolkien	Novel
4	DAQ in LabVIEW	1234364536	Wiley	Hans-Petter Halvorsen	LabVIEW

Step by step Exercises

Hans-Petter Halvorsen, M.Sc.

Software Needed in the Exercises

1. ERwin (Database Design Tool)
2. SQL Server (Database System)
3. Visual Studio/C# (Programming)

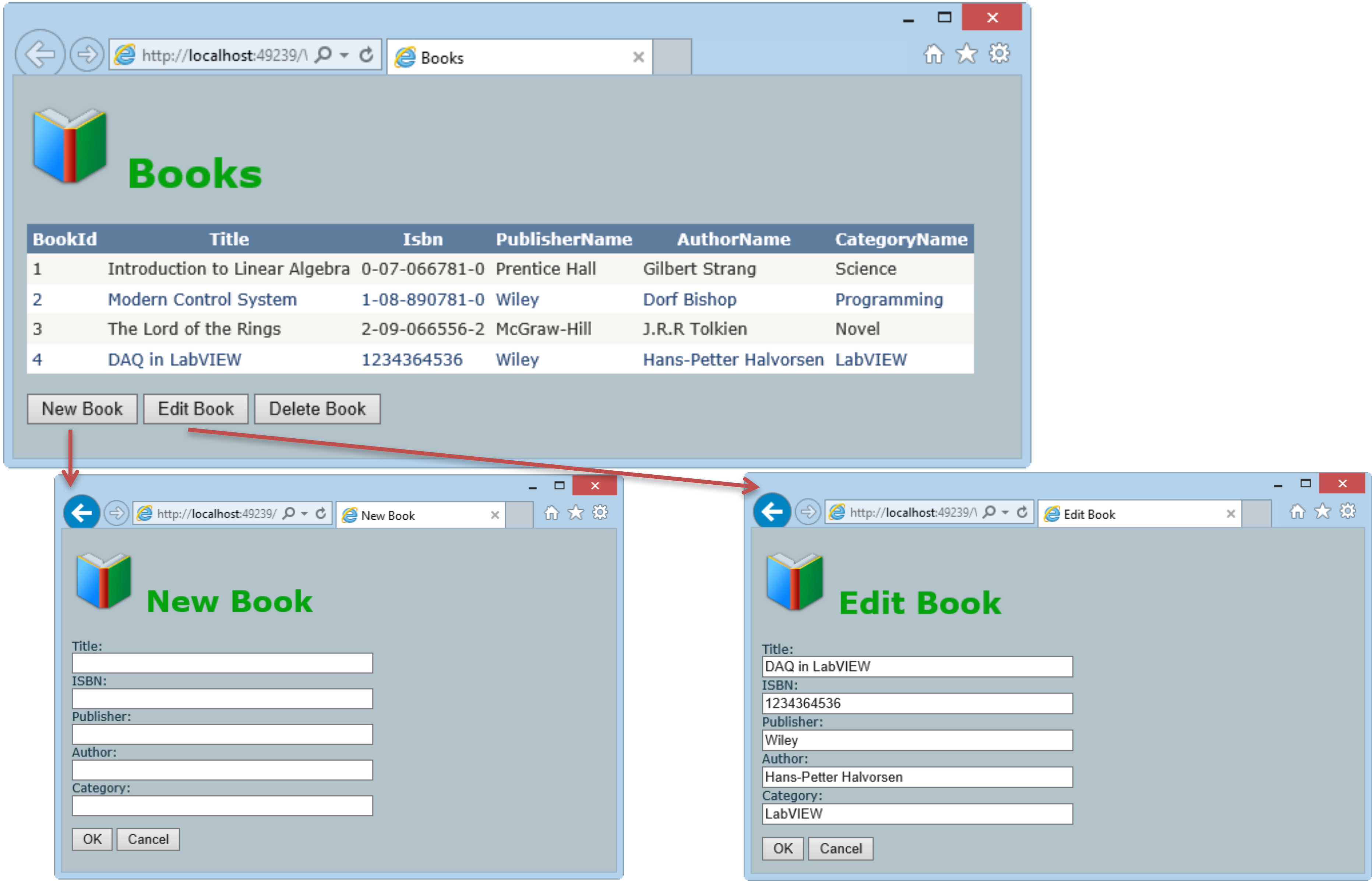
ERwin®

Microsoft® SQL Server®

Visual Studio®

ASP.NET WebForm App

This is the WebForm App we are going to create in this Tutorial:



Books

BookId	Title	Isbn	PublisherName	AuthorName	CategoryName
1	Introduction to Linear Algebra	0-07-066781-0	Prentice Hall	Gilbert Strang	Science
2	Modern Control System	1-08-890781-0	Wiley	Dorf Bishop	Programming
3	The Lord of the Rings	2-09-066556-2	McGraw-Hill	J.R.R Tolkien	Novel
4	DAQ in LabVIEW	1234364536	Wiley	Hans-Petter Halvorsen	LabVIEW

New Book

Title:

ISBN:

Publisher:

Author:

Category:

Edit Book

Title: DAQ in LabVIEW

ISBN: 1234364536

Publisher: Wiley

Author: Hans-Petter Halvorsen

Category: LabVIEW

Database Design in ERwin

Hans-Petter Halvorsen, M.Sc.

Database Design – ER Diagram

ER Diagram (Entity-Relationship Diagram)

- Used for Design and Modeling of Databases.
- Specify Tables and **relationship** between them (**Primary Keys** and **Foreign Keys**)

Example:

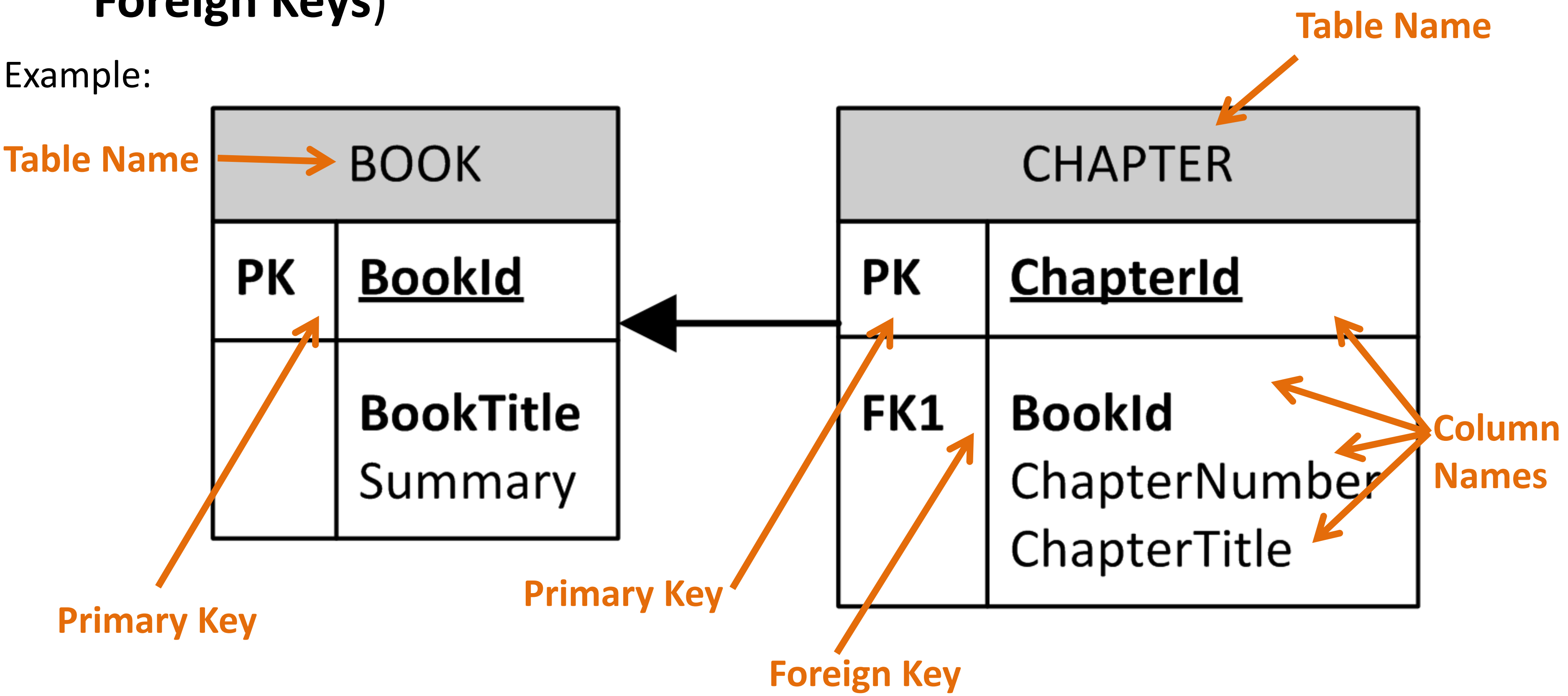


Table Name

BOOK

PK **BookId**

BookTitle

Summary

Primary Key

Table Name

CHAPTER

PK **ChapterId**

FK1 **BookId**

ChapterNumber

ChapterTitle

Primary Key

Foreign Key

Column Names

Relational Database. In a relational database all the tables have one or more relation with each other using Primary Keys (PK) and Foreign Keys (FK). Note! You can only have one PK in a table, but you may have several FK's.

Database Design Tools

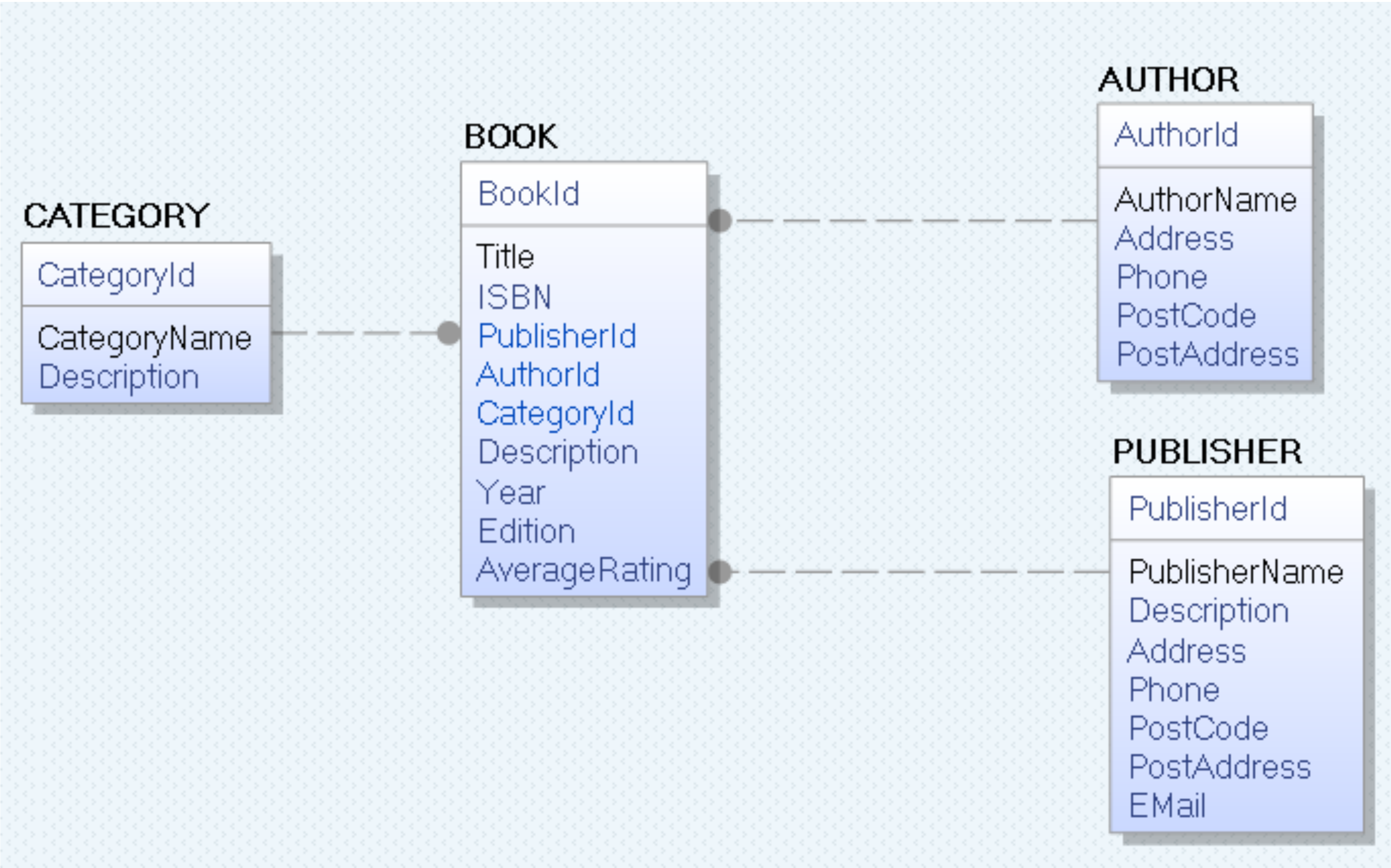
- Visio
- PowerDesigner
- CA ERwin
  - CA ERwin Data Modeler Community Edition
  - Community Edition is Free, 25 objects limit
  - Support for Oracle, SQL Server, MySQL, ODBC, Sybase
- Toad Data Modeler
- A Simple designer is also included with SQL Server (physical model, not logical model)
- etc.

Database - “Best Practice”

- **Tables**: Use upper case and singular form in table names – not plural, e.g., “STUDENT” (not students)
- **Columns**: Use Pascal notation, e.g., “StudentId”
- **Primary Key**:
  - If the table name is “COURSE”, name the Primary Key column “CourseId”, etc.
  - “Always” use Integer and Identity(1,1) for Primary Keys. Use UNIQUE constraint for other columns that needs to be unique, e.g. RoomNumber
- Specify **Required** Columns (NOT NULL) – i.e., which columns that need to have data or not
- Standardize on few/these **Data Types**: *int*, *float*, *varchar(x)*, *datetime*, *bit*
- Use English for table and column names
- Avoid abbreviations! (Use RoomNumber – not RoomNo, RoomNr, ...)

Database

This is our Example Database (Designed with ERwin)



CATEGORY

CategoryId

CategoryName
Description

BOOK

BookId

Title
ISBN
PublisherId
AuthorId
CategoryId
Description
Year
Edition
AverageRating

AUTHOR

AuthorId

AuthorName
Address
Phone
PostCode
PostAddress

PUBLISHER

PublisherId

PublisherName
Description
Address
Phone
PostCode
PostAddress
EMail

You are finished with the Exercise

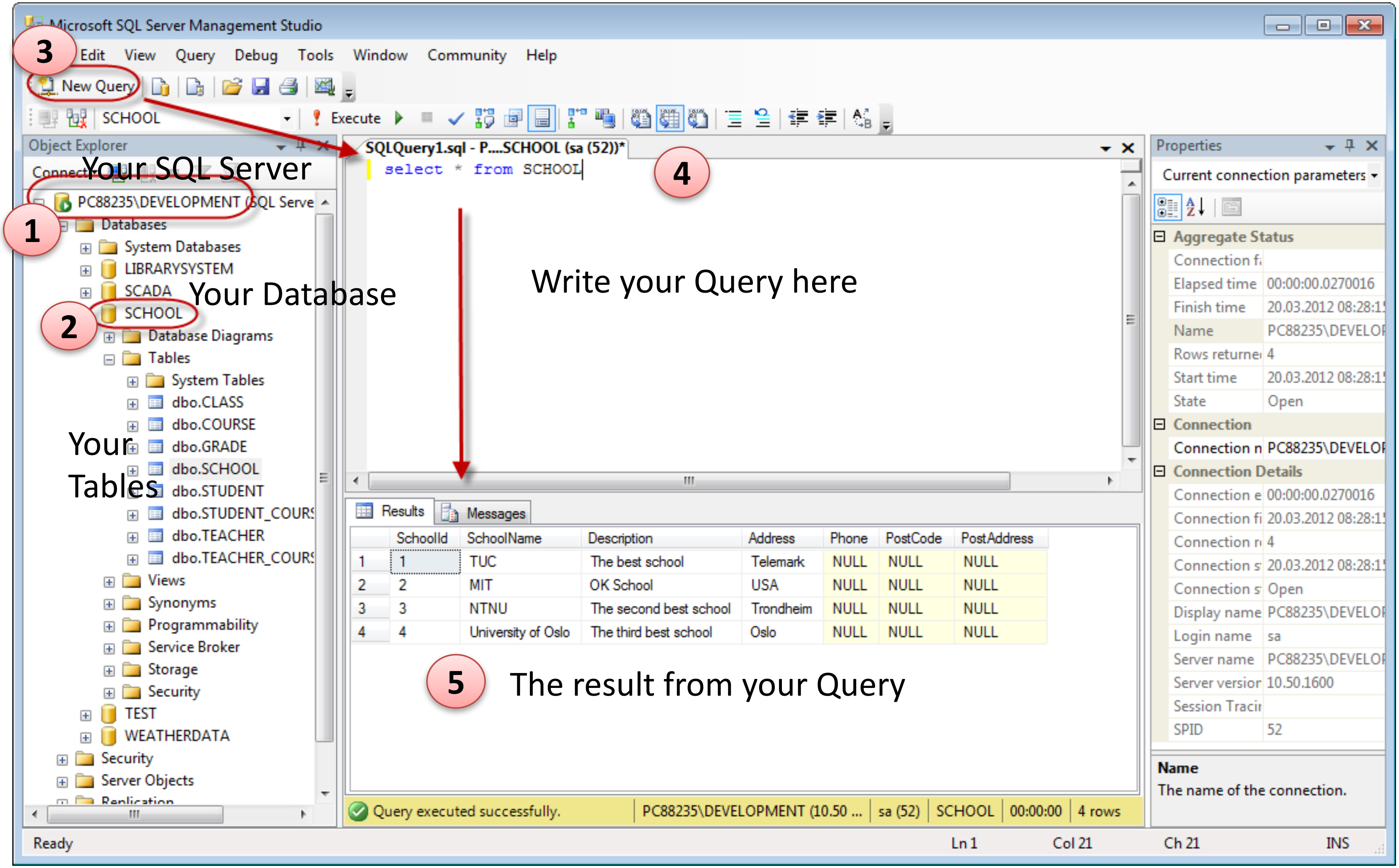
Microsoft®

SQL Server®

Create Database, Tables, Views and Stored Procedures

Hans-Petter Halvorsen, M.Sc.

Microsoft SQL Server – Management Studio



1 Your SQL Server

2 Your Database

3

4 Write your Query here

```
select * from SCHOOL
```

Your Tables

5 The result from your Query

	SchoolId	SchoolName	Description	Address	Phone	PostCode	PostAddress
1	1	TUC	The best school	Telemark	NULL	NULL	NULL
2	2	MIT	OK School	USA	NULL	NULL	NULL
3	3	NTNU	The second best school	Trondheim	NULL	NULL	NULL
4	4	University of Oslo	The third best school	Oslo	NULL	NULL	NULL

Query executed successfully. | PC88235\DEVELOPMENT (10.50 ... | sa (52) | SCHOOL | 00:00:00 | 4 rows

Aggregate Status	
Connection fi	
Elapsed time	00:00:00.0270016
Finish time	20.03.2012 08:28:1
Name	PC88235\DEVELOF
Rows returne	4
Start time	20.03.2012 08:28:1
State	Open
Connection	
Connection n	PC88235\DEVELOF
Connection Details	
Connection e	00:00:00.0270016
Connection fi	20.03.2012 08:28:1
Connection r	4
Connection s	20.03.2012 08:28:1
Connection s	Open
Display name	PC88235\DEVELOF
Login name	sa
Server name	PC88235\DEVELOF
Server versior	10.50.1600
Session Tracir	
SPID	52

Name
The name of the connection.

Microsoft SQL Server – Tips and Tricks

Do you get an error when trying to change your tables?

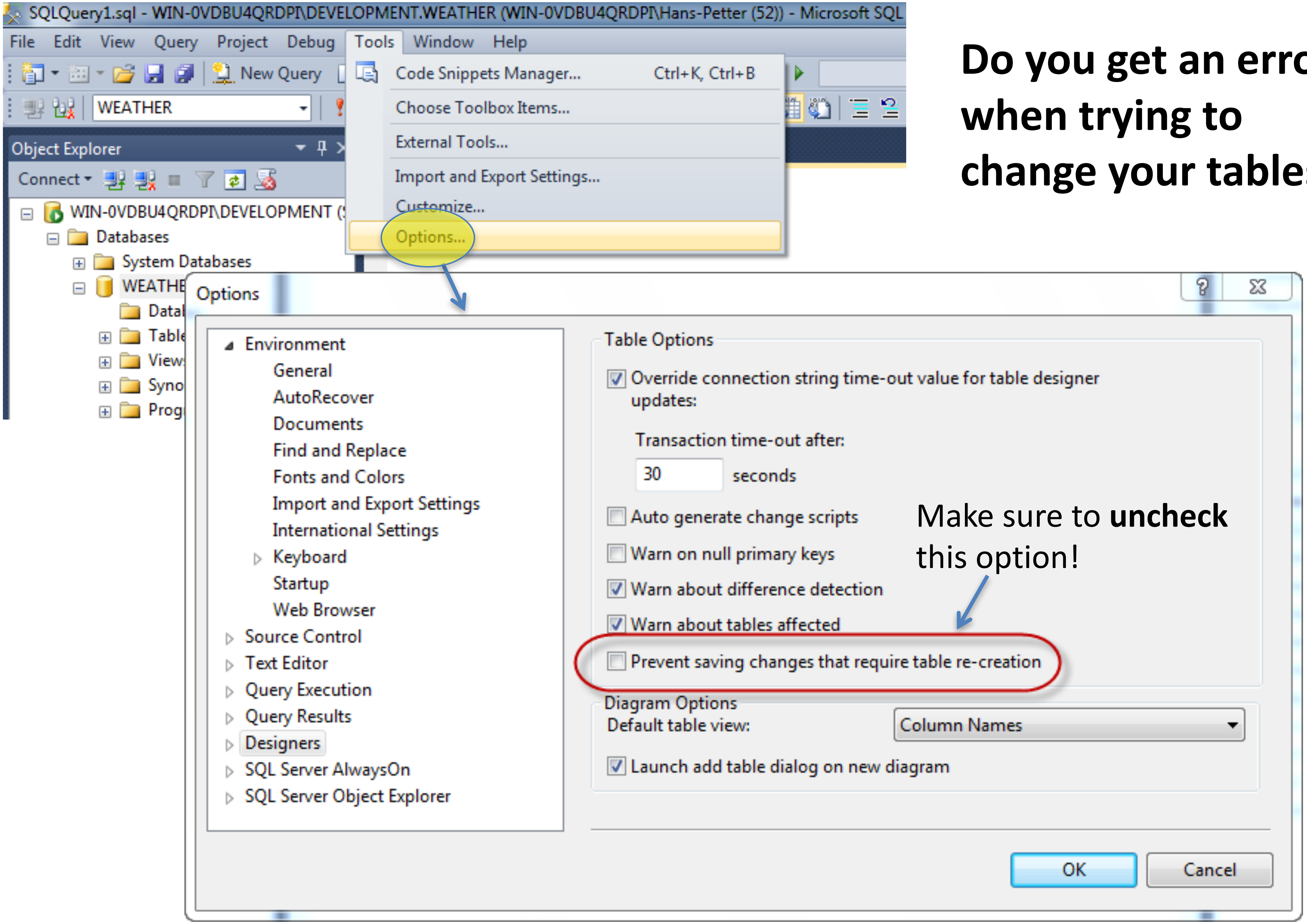


Table Options

☑ Override connection string time-out value for table designer updates:

Transaction time-out after: 30 seconds

☐ Auto generate change scripts

☐ Warn on null primary keys

☑ Warn about difference detection

☑ Warn about tables affected

☐ Prevent saving changes that require table re-creation

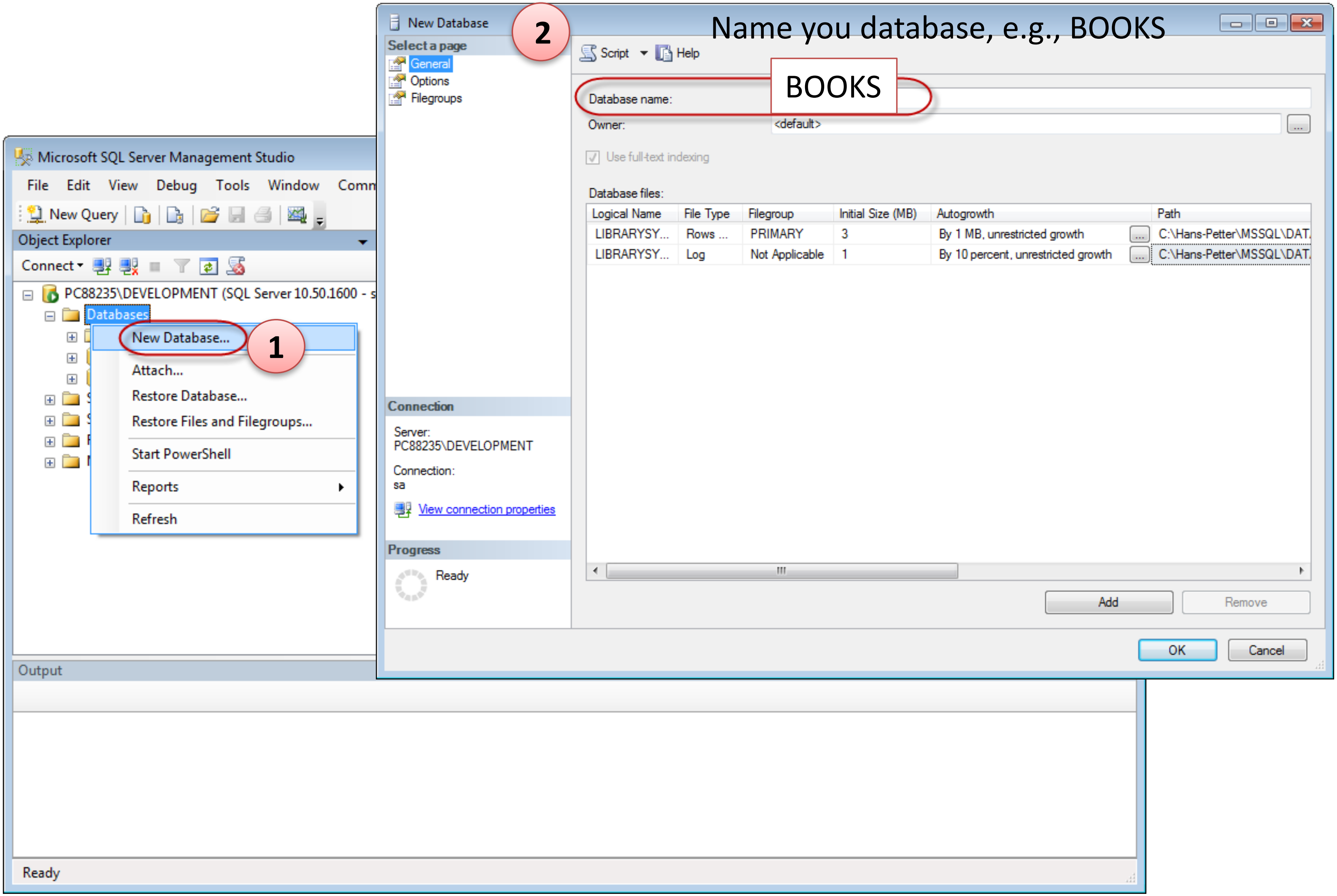
Diagram Options

Default table view: Column Names

☑ Launch add table dialog on new diagram

Make sure to **uncheck** this option!

Microsoft SQL Server – Create a New Database



1

2

Name you database, e.g., BOOKS

BOOKS

Database name:

Owner: <default>

☑ Use full-text indexing

Database files:

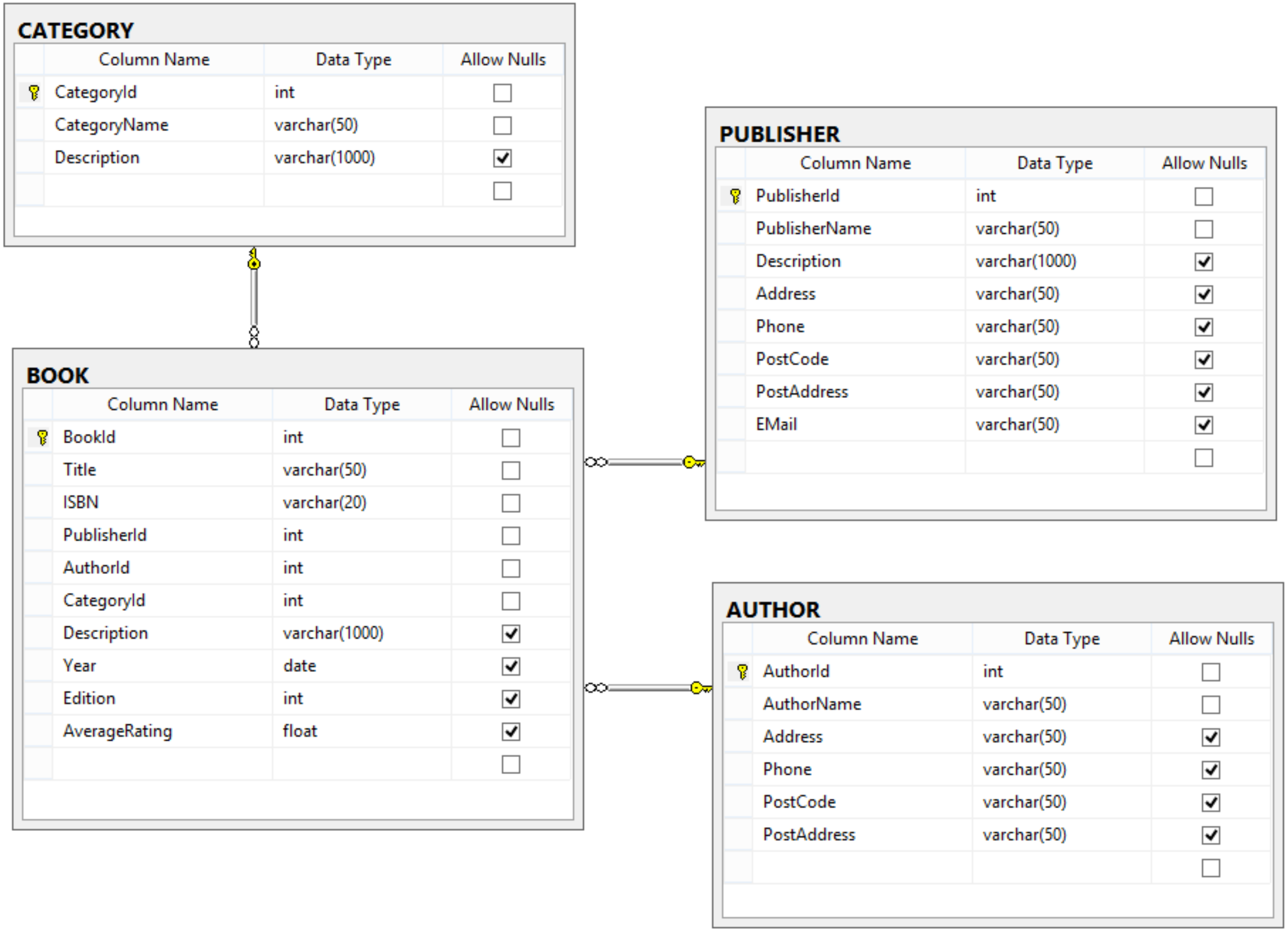
Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth	Path
LIBRARYSY...	Rows ...	PRIMARY	3	By 1 MB, unrestricted growth	C:\Hans-Petter\MSSQL\DAT
LIBRARYSY...	Log	Not Applicable	1	By 10 percent, unrestricted growth	C:\Hans-Petter\MSSQL\DAT

Server:
PC88235\DEVELOPMENT

Connection:
sa

Database

This is our Example Database (SQL Server Designer)



CATEGORY

	Column Name	Data Type	Allow Nulls
🔑	CategoryId	int	☐
	CategoryName	varchar(50)	☐
	Description	varchar(1000)	☑
			☐

PUBLISHER

	Column Name	Data Type	Allow Nulls
🔑	PublisherId	int	☐
	PublisherName	varchar(50)	☐
	Description	varchar(1000)	☑
	Address	varchar(50)	☑
	Phone	varchar(50)	☑
	PostCode	varchar(50)	☑
	PostAddress	varchar(50)	☑
	EMail	varchar(50)	☑
			☐

BOOK

	Column Name	Data Type	Allow Nulls
🔑	BookId	int	☐
	Title	varchar(50)	☐
	ISBN	varchar(20)	☐
	PublisherId	int	☐
	AuthorId	int	☐
	CategoryId	int	☐
	Description	varchar(1000)	☑
	Year	date	☑
	Edition	int	☑
	AverageRating	float	☑
			☐

AUTHOR

	Column Name	Data Type	Allow Nulls
🔑	AuthorId	int	☐
	AuthorName	varchar(50)	☐
	Address	varchar(50)	☑
	Phone	varchar(50)	☑
	PostCode	varchar(50)	☑
	PostAddress	varchar(50)	☑
			☐

SQL Script - Tables

```
if not exists (select * from dbo.sysobjects where id = object_id(N'[AUTHOR]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
CREATE TABLE [AUTHOR]
(
                [AuthorId] [int] IDENTITY(1, 1) NOT NULL PRIMARY KEY,
                [AuthorName] [varchar](50) NOT NULL UNIQUE,
                [Address] [varchar](50) NULL,
                [Phone] [varchar](50) NULL,
                [PostCode] [varchar](50) NULL,
                [PostAddress] [varchar](50) NULL,
)
GO

if not exists (select * from dbo.sysobjects where id = object_id(N'[PUBLISHER]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
CREATE TABLE [PUBLISHER]
(
                [PublisherId] [int] IDENTITY(1, 1) NOT NULL PRIMARY KEY,
                [PublisherName] [varchar](50) NOT NULL UNIQUE,
                [Description] [varchar](1000) NULL,
                [Address] [varchar](50) NULL,
                [Phone] [varchar](50) NULL,
                [PostCode] [varchar](50) NULL,
                [PostAddress] [varchar](50) NULL,
                [EMail] [varchar](50) NULL,
)
GO

if not exists (select * from dbo.sysobjects where id = object_id(N'[CATEGORY]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
CREATE TABLE [CATEGORY]
(
                [CategoryId] [int] IDENTITY(1, 1) NOT NULL PRIMARY KEY,
                [CategoryName] [varchar](50) NOT NULL UNIQUE,
                [Description] [varchar](1000) NULL,
)
GO

if not exists (select * from dbo.sysobjects where id = object_id(N'[BOOK]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
CREATE TABLE [BOOK]
(
                [BookId] [int] IDENTITY(1, 1) NOT NULL PRIMARY KEY,
                [Title] [varchar](50) NOT NULL UNIQUE,
                [ISBN] [varchar](20) NOT NULL,
                [PublisherId] [int] NOT NULL FOREIGN KEY REFERENCES [PUBLISHER] ([PublisherId]),
                [AuthorId] [int] NOT NULL FOREIGN KEY REFERENCES [AUTHOR] ([AuthorId]),
                [CategoryId] [int] NOT NULL FOREIGN KEY REFERENCES [CATEGORY] ([CategoryId]),
                [Description] [varchar](1000) NULL,
                [Year] [date] NULL,
                [Edition] [int] NULL,
                [AverageRating] [float] NULL,
)
GO
```

SQL Script – Views – “GetBookData”

```sql
IF EXISTS (SELECT name
            FROM   sysobjects
            WHERE  name = 'GetBookData'
            AND     type = 'V')
         DROP VIEW GetBookData
GO

CREATE VIEW GetBookData
AS

SELECT
BOOK.BookId,
BOOK.Title,
BOOK.ISBN,
PUBLISHER.PublisherName,
AUTHOR.AuthorName,
CATEGORY.CategoryName

FROM BOOK
INNER JOIN AUTHOR ON BOOK.AuthorId = AUTHOR.AuthorId
INNER JOIN PUBLISHER ON BOOK.PublisherId = PUBLISHER.PublisherId
INNER JOIN CATEGORY ON BOOK.CategoryId = CATEGORY.CategoryId

GO
```

SQL Script – Stored Procedures – “CreateBook”

```
IF EXISTS (SELECT name
                FROM   sysobjects
                WHERE  name = 'CreateBook'
                AND        type = 'P')
            DROP PROCEDURE CreateBook
GO

CREATE PROCEDURE CreateBook
@Title varchar(50),
@Isbn varchar(20),
@PublisherName varchar(50),
@AuthorName varchar(50),
@CategoryName varchar(50)
AS

if not exists (select * from CATEGORY where CategoryName = @CategoryName)
            INSERT INTO CATEGORY (CategoryName) VALUES (@CategoryName)

if not exists (select * from AUTHOR where AuthorName = @AuthorName)
            INSERT INTO AUTHOR (AuthorName) VALUES (@AuthorName)

if not exists (select * from PUBLISHER where PublisherName = @PublisherName)
            INSERT INTO PUBLISHER (PublisherName) VALUES (@PublisherName)

if not exists (select * from BOOK where Title = @Title)
            INSERT INTO BOOK (Title, ISBN, PublisherId, AuthorId, CategoryId)
            VALUES
            (
            @Title,
            @ISBN,
            (select PublisherId from PUBLISHER where PublisherName=@PublisherName),
            (select AuthorId from AUTHOR where AuthorName=@AuthorName),
            (select CategoryId from CATEGORY where CategoryName=@CategoryName)
            )

GO
```

SQL Script – Stored Procedures – “UpdateBook”

```
IF EXISTS (SELECT name
               FROM   sysobjects
               WHERE  name = 'UpdateBook'
               AND        type = 'P')
            DROP PROCEDURE UpdateBook
GO

CREATE PROCEDURE UpdateBook
@BookId int,
@Title varchar(50),
@ISBN varchar(20),
@PublisherName varchar(50),
@AuthorName varchar(50),
@CategoryName varchar(50)
AS

if not exists (select * from CATEGORY where CategoryName = @CategoryName)
            INSERT INTO CATEGORY (CategoryName) VALUES (@CategoryName)

if not exists (select * from AUTHOR where AuthorName = @AuthorName)
            INSERT INTO AUTHOR (AuthorName) VALUES (@AuthorName)

if not exists (select * from PUBLISHER where PublisherName = @PublisherName)
            INSERT INTO PUBLISHER (PublisherName) VALUES (@PublisherName)

UPDATE BOOK SET
Title = @Title,
ISBN = @ISBN,
PublisherId = (select PublisherId from PUBLISHER where
PublisherName=@PublisherName),
AuthorId = (select AuthorId from AUTHOR where AuthorName=@AuthorName),
CategoryId = (select CategoryId from CATEGORY where CategoryName=@CategoryName)

WHERE BookId = @BookId

GO
```

SQL Script – Stored Procedures – “DeleteBook”

```
IF EXISTS (SELECT name
           FROM   sysobjects
           WHERE  name = 'DeleteBook'
           AND    type = 'P')
        DROP PROCEDURE DeleteBook
GO

CREATE PROCEDURE DeleteBook
@BookId int
AS


delete from BOOK where BookId=@BookId


GO
```

SQL Script – Insert some Data into the Tables

```sql
--CATEGORY ----------------------------------
INSERT INTO CATEGORY (CategoryName) VALUES ('Science')
GO
INSERT INTO CATEGORY (CategoryName) VALUES ('Programming')
GO
INSERT INTO CATEGORY (CategoryName) VALUES ('Novel')
GO

--AUTHOR ----------------------------------
INSERT INTO AUTHOR (AuthorName) VALUES ('Knut Hamsun')
GO
INSERT INTO AUTHOR (AuthorName) VALUES ('Gilbert Strang')
GO
INSERT INTO AUTHOR (AuthorName) VALUES ('J.R.R Tolkien')
GO
INSERT INTO AUTHOR (AuthorName) VALUES ('Dorf Bishop')
GO

--PUBLISHER ----------------------------------
INSERT INTO PUBLISHER (PublisherName) VALUES ('Prentice Hall')
GO
INSERT INTO PUBLISHER (PublisherName) VALUES ('Wiley')
GO
INSERT INTO PUBLISHER (PublisherName) VALUES ('McGraw-Hill')
GO
```

SQL Script – Insert some Data into the Tables

```
--BOOK -----------------------------------
INSERT INTO BOOK (Title, ISBN, PublisherId, AuthorId, CategoryId)
VALUES
(
'Introduction to Linear Algebra',
'0-07-066781-0',
(select PublisherId from PUBLISHER where PublisherName='Prentice Hall'),
(select AuthorId from AUTHOR where AuthorName='Gilbert Strang'),
(select CategoryId from CATEGORY where CategoryName='Science')
)
GO

INSERT INTO BOOK (Title, ISBN, PublisherId, AuthorId, CategoryId)
VALUES
(
'Modern Control System',
'1-08-890781-0',
(select PublisherId from PUBLISHER where PublisherName='Wiley'),
(select AuthorId from AUTHOR where AuthorName='Dorf Bishop'),
(select CategoryId from CATEGORY where CategoryName='Programming')
)
GO

INSERT INTO BOOK (Title, ISBN, PublisherId, AuthorId, CategoryId)
VALUES
(
'The Lord of the Rings',
'2-09-066556-2',
(select PublisherId from PUBLISHER where PublisherName='McGraw-Hill'),
(select AuthorId from AUTHOR where AuthorName='J.R.R Tolkien'),
(select CategoryId from CATEGORY where CategoryName='Novel')
)
GO
```

You are finished with the Exercise

Visual Studio®

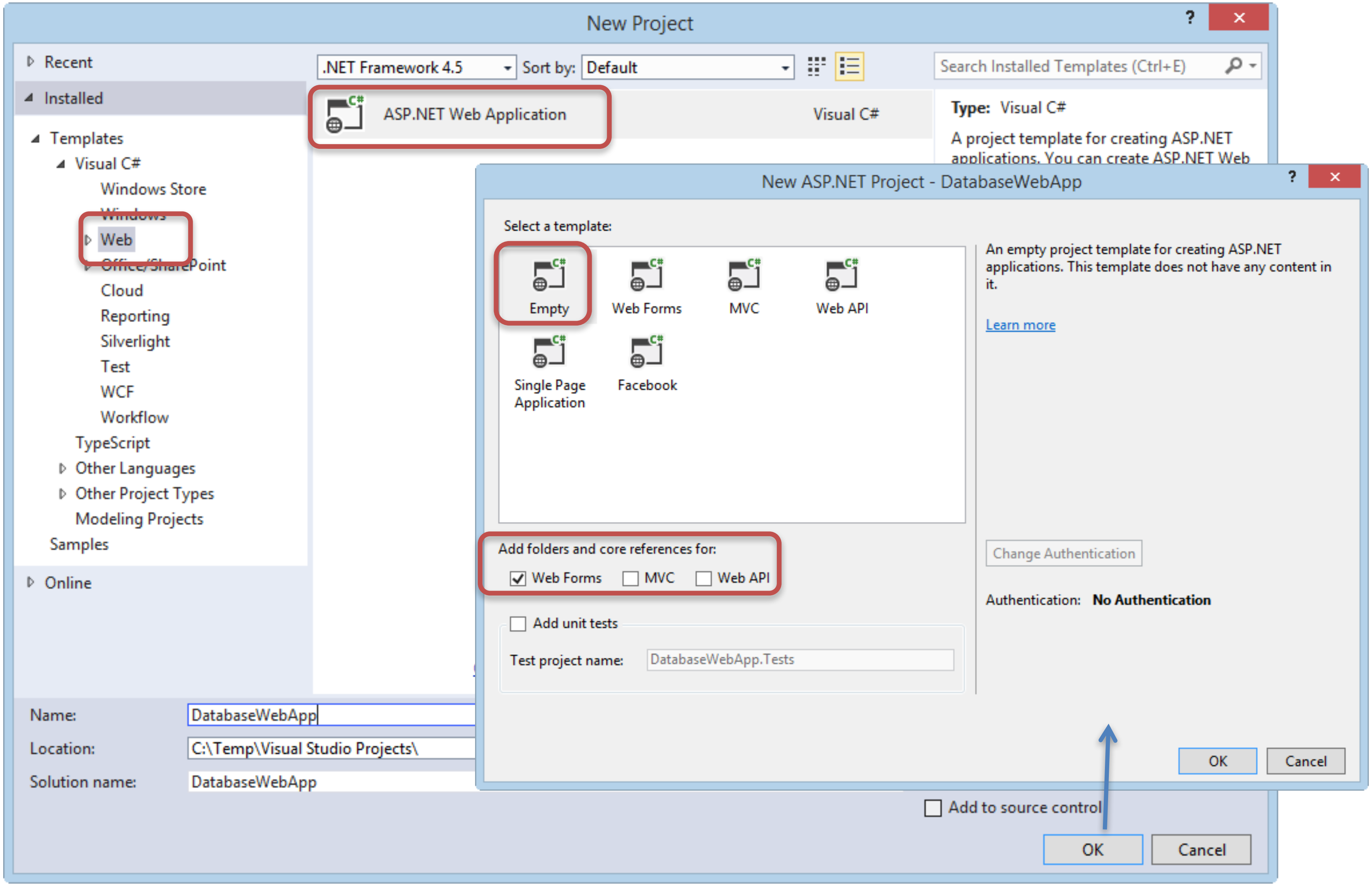
ASP.NET Web Forms

Hans-Petter Halvorsen, M.Sc.

Create New Project in Visual Studio

ASP.NET Web Form

Create a New Project in Visual Studio



.NET Framework 4.5 Sort by: Default

Type: Visual C#

A project template for creating ASP.NET applications. You can create ASP.NET Web

An empty project template for creating ASP.NET applications. This template does not have any content in it.

Add folders and core references for:

☑ Web Forms □ MVC □ Web API

□ Add unit tests

Test project name: DatabaseWebApp.Tests

Authentication: **No Authentication**

Name: DatabaseWebApp

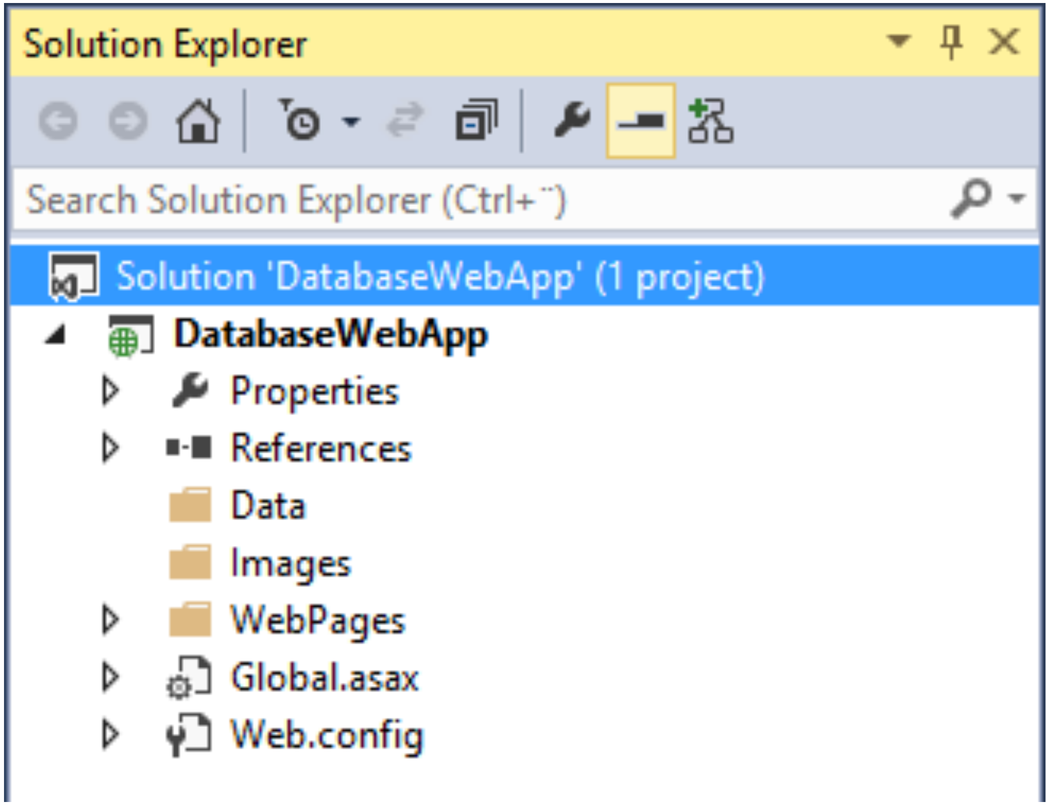
Location: C:\Temp\Visual Studio Projects\

Solution name: DatabaseWebApp

□ Add to source control

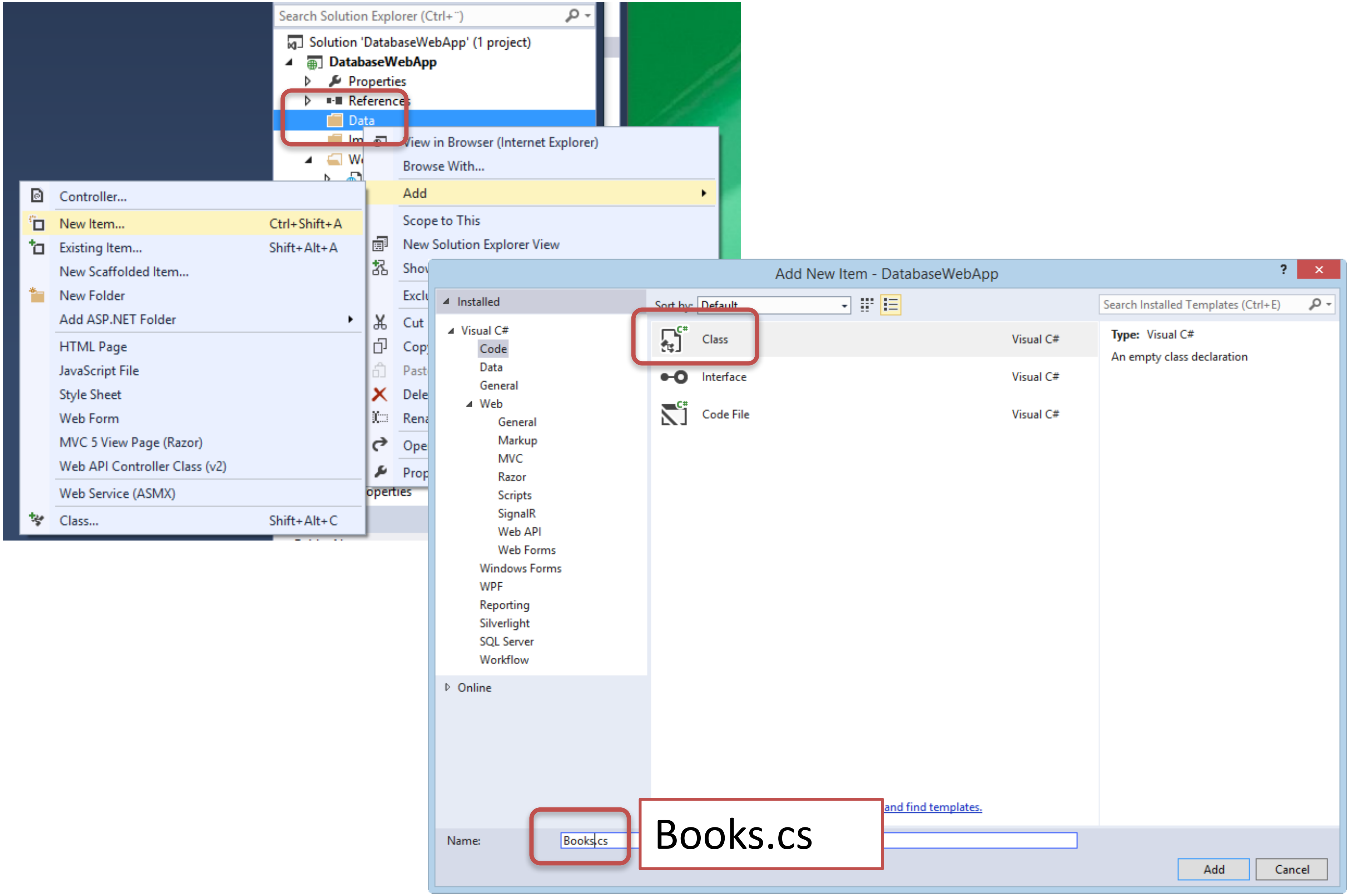
Clean Up Solution

1. Remove “App_Data” and “Models” folders
2. Create Folder called “WebPages”
3. Create Folder called “Data”
4. Create Folder called “Images”



Create Database Logic

Create Database Code – Create a new Class (“Books.cs”)



Name: Books.cs

Books.cs

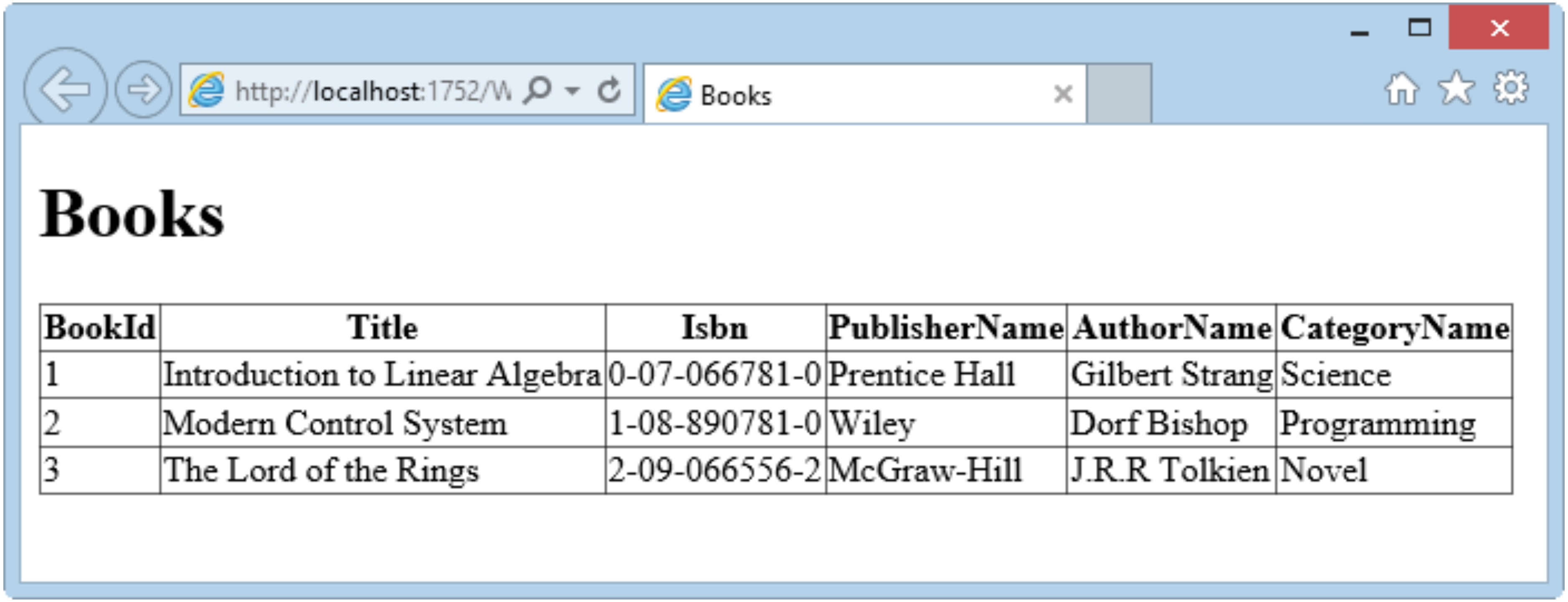
Create the Following Class in “Books.cs”

```
using System.Data.SqlClient;
using System.Data.SqlTypes;
using System.Data;

public class Book
 {
        public int BookId { get; set; }
        public string Title { get; set; }
        public string Isbn { get; set; }
        public string PublisherName { get; set; }
        public string AuthorName { get; set; }
        public string CategoryName { get; set; }

        public List<Book> GetBooks(string connectionString)
        {

            List<Book> bookList = new List<Book>();

            SqlConnection con = new SqlConnection(connectionString);

            string selectSQL = "select BookId, Title, Isbn, PublisherName, AuthorName, CategoryName  from GetBookData";

            con.Open();

            SqlCommand cmd = new SqlCommand(selectSQL, con);

            SqlDataReader dr = cmd.ExecuteReader();

            if (dr != null)
            {
                while (dr.Read())
                {
                    Book book = new Book();

                    book.BookId = Convert.ToInt32(dr["BookId"]);
                    book.Title = dr["Title"].ToString();
                    book.Isbn = dr["ISBN"].ToString();
                    book.PublisherName = dr["PublisherName"].ToString();
                    book.AuthorName = dr["AuthorName"].ToString();
                    book.CategoryName = dr["CategoryName"].ToString();

                    bookList.Add(book);
                }
            }

            return bookList;
        }
}
```

Create GUI

“Books.aspx” Web Form

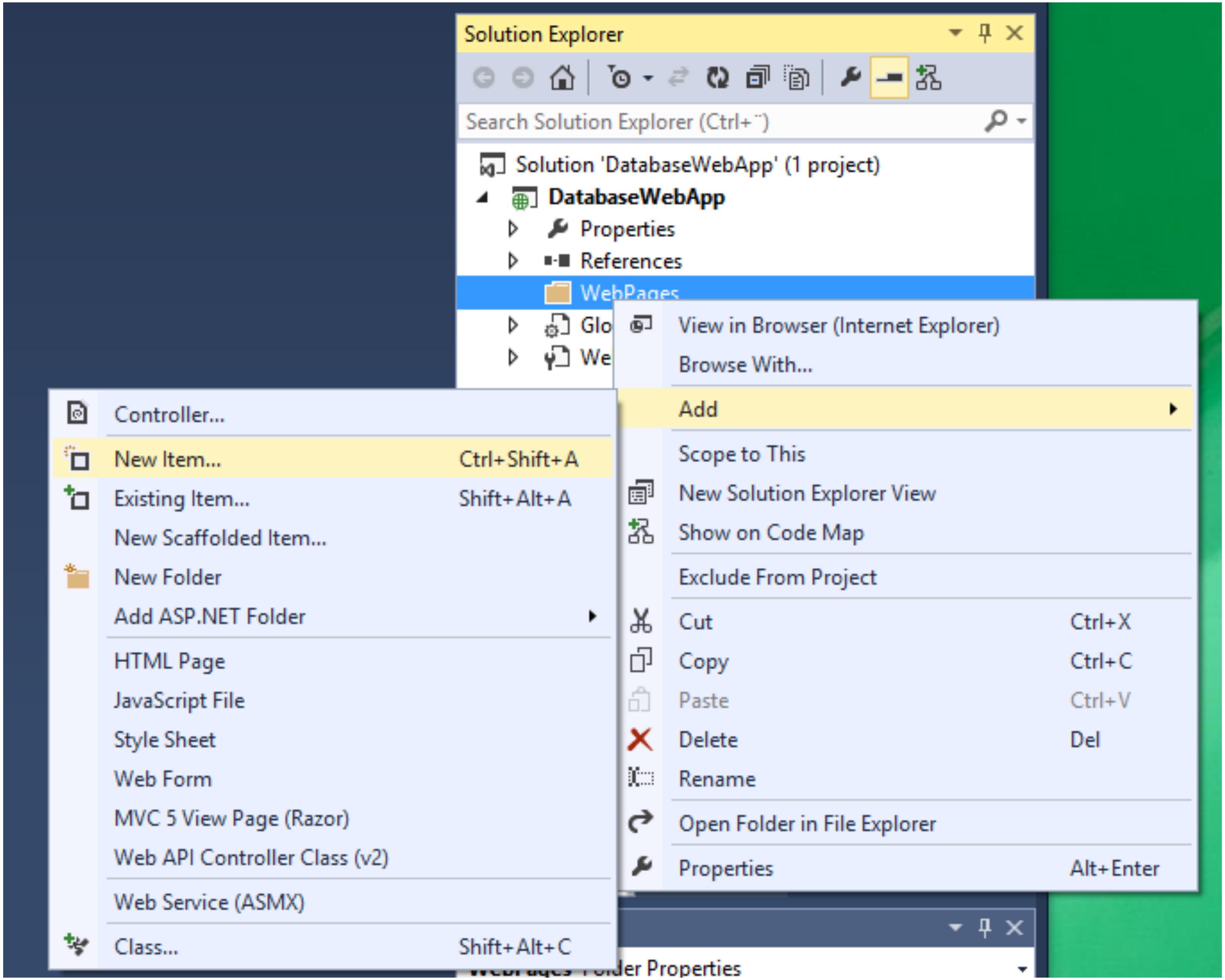
Next, We will start creating this Web Form Page, Step by Step



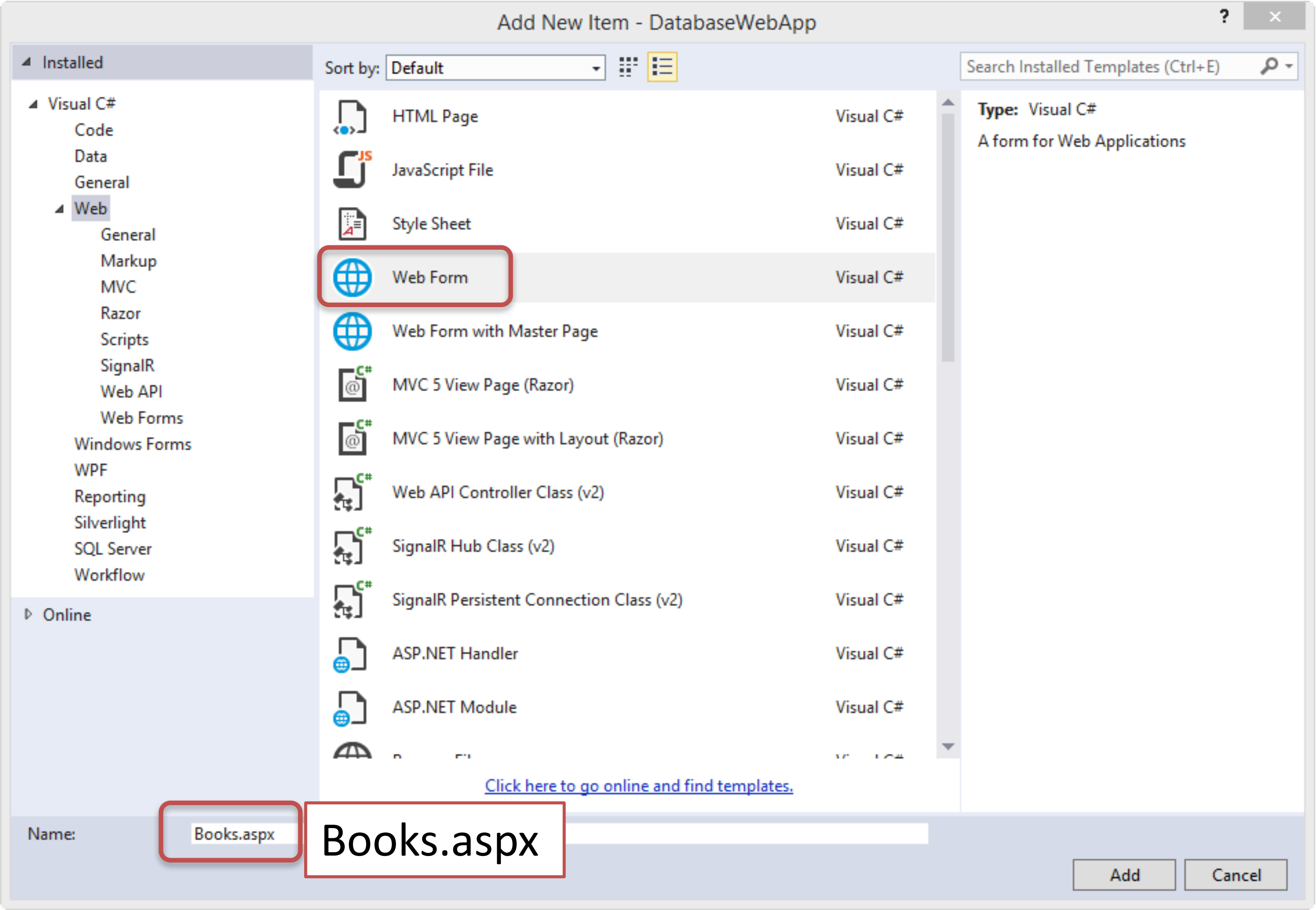
Books

BookId	Title	Isbn	PublisherName	AuthorName	CategoryName
1	Introduction to Linear Algebra	0-07-066781-0	Prentice Hall	Gilbert Strang	Science
2	Modern Control System	1-08-890781-0	Wiley	Dorf Bishop	Programming
3	The Lord of the Rings	2-09-066556-2	McGraw-Hill	J.R.R Tolkien	Novel

Add New Item in the “WebPages” Folder



Add New Web Form ("Books.aspx")



Add New Item - DatabaseWebApp

Sort by: Default

Search Installed Templates (Ctrl+E)

Type: Visual C#

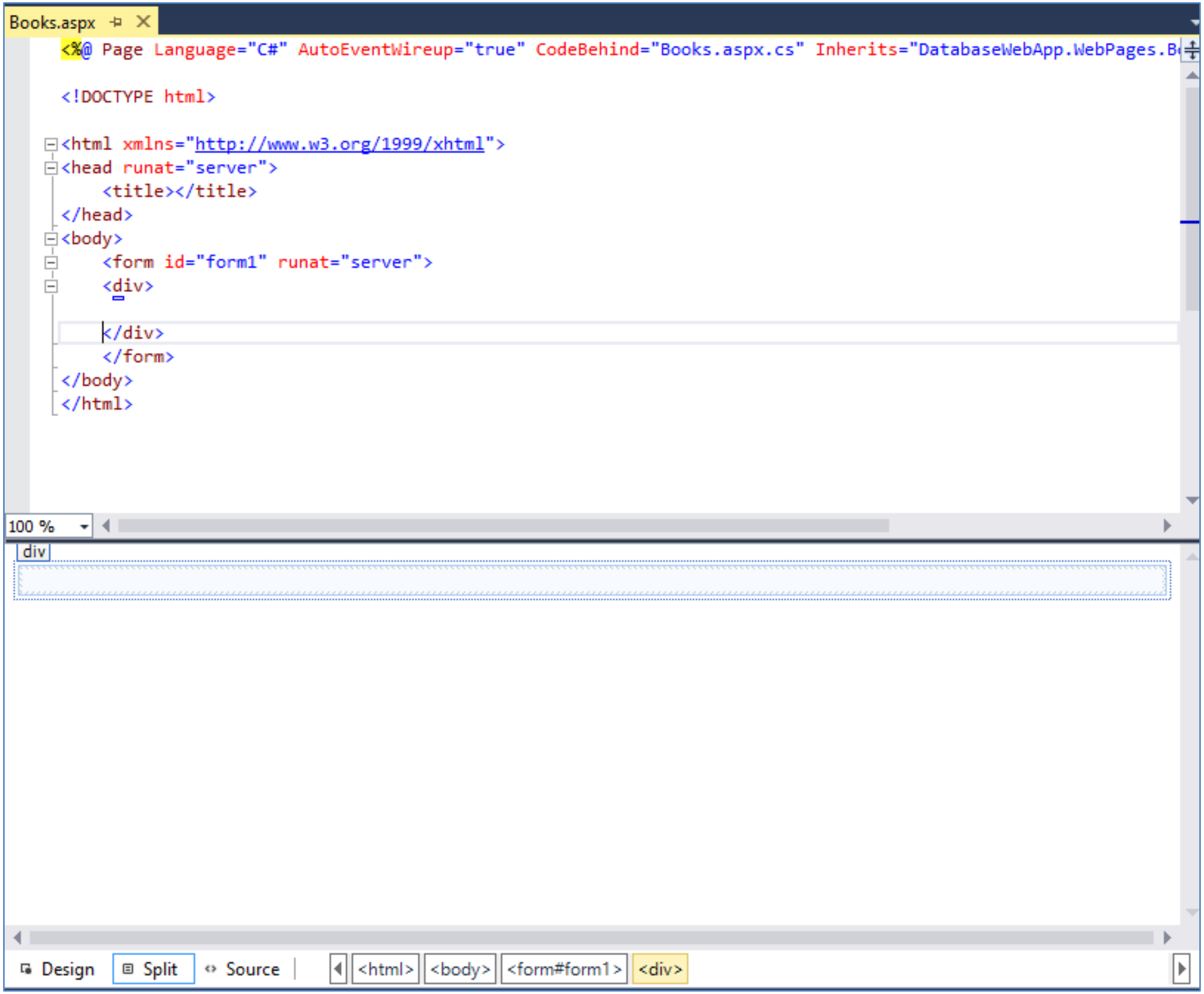
A form for Web Applications

Name: Books.aspx

Books.aspx

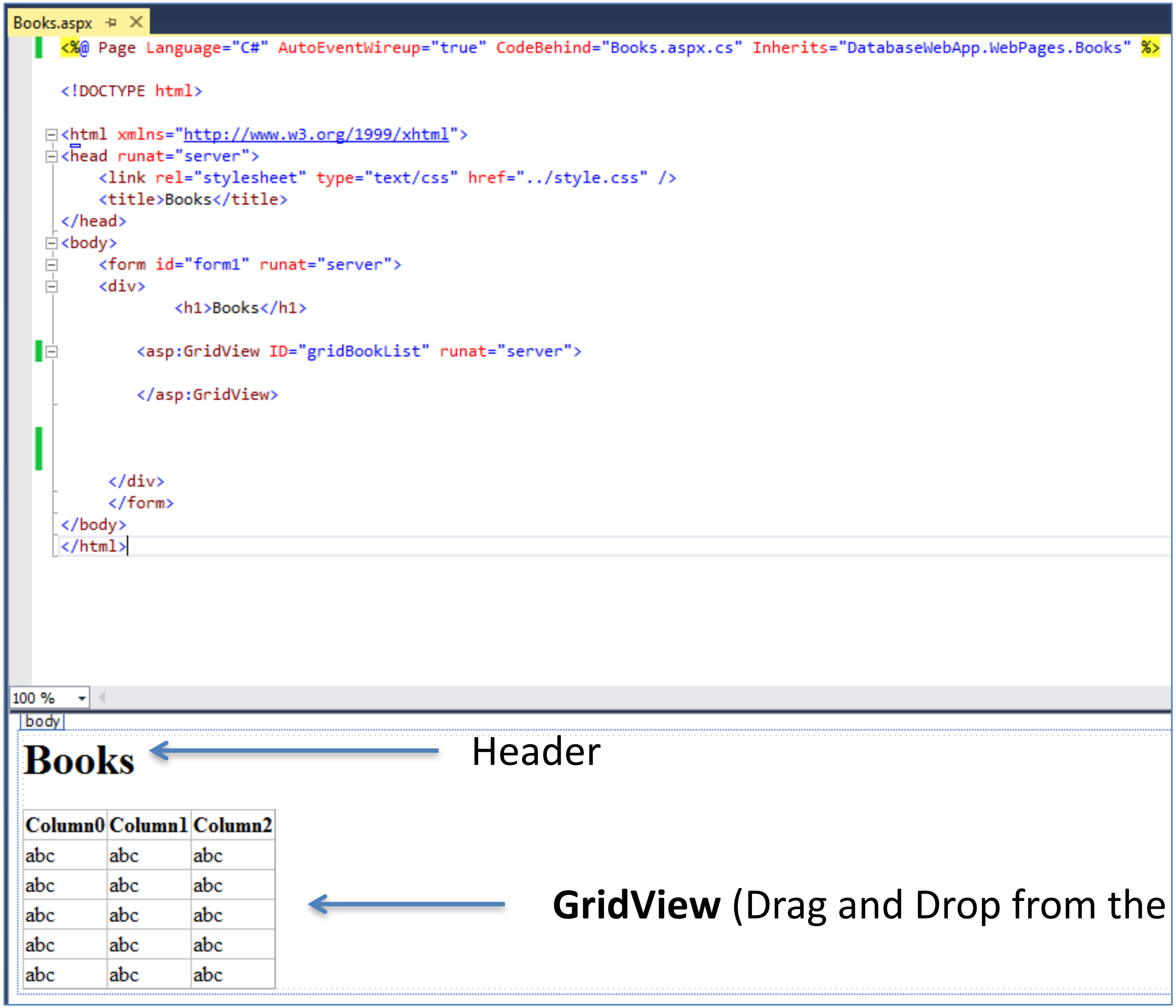
?
Installed
Visual C#
Code
Data
General
Web
General
Markup
MVC
Razor
Scripts
SignalR
Web API
Web Forms
Windows Forms
WPF
Reporting
Silverlight
SQL Server
Workflow
Online
HTML Page Visual C#
JavaScript File Visual C#
Style Sheet Visual C#
Web Form Visual C#
Web Form with Master Page Visual C#
MVC 5 View Page (Razor) Visual C#
MVC 5 View Page with Layout (Razor) Visual C#
Web API Controller Class (v2) Visual C#
SignalR Hub Class (v2) Visual C#
SignalR Persistent Connection Class (v2) Visual C#
ASP.NET Handler Visual C#
ASP.NET Module Visual C#
Click here to go online and find templates.
Add
Cancel

“Books.aspx”



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Books.aspx.cs" Inherits="DatabaseWebApp.WebPages.B

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div>

    </div>
    </form>
</body>
</html>
```

Create the following GUI (“Books.aspx”)



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Books.aspx.cs" Inherits="DatabaseWebApp.WebPages.Books" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <link rel="stylesheet" type="text/css" href="../style.css" />
    <title>Books</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
            <h1>Books</h1>

        <asp:GridView ID="gridBookList" runat="server">

        </asp:GridView>



      </div>
      </form>
</body>
</html>
```

Books

Header

Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

GridView (Drag and Drop from the Toolbox)

Create the following Code ("Books.aspx.cs")

```
using System.Web.Configuration;
using DatabaseWebApp.Data;

...

public partial class BookList : System.Web.UI.Page
{
 private string connectionString = WebConfigurationManager.ConnectionStrings["LibraryDBConnectionString"].ConnectionString;

        void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                FillBookGrid();
            }
        }

        private void FillBookGrid()
        {
            List<Book> bookList = new List<Book>();
            Book book = new Book();

            bookList = book.GetBooks(connectionString);

            gridBookList.DataSource = bookList;
            gridBookList.DataBind();
        }
}
```

Reference to our Class that communicates with the Database

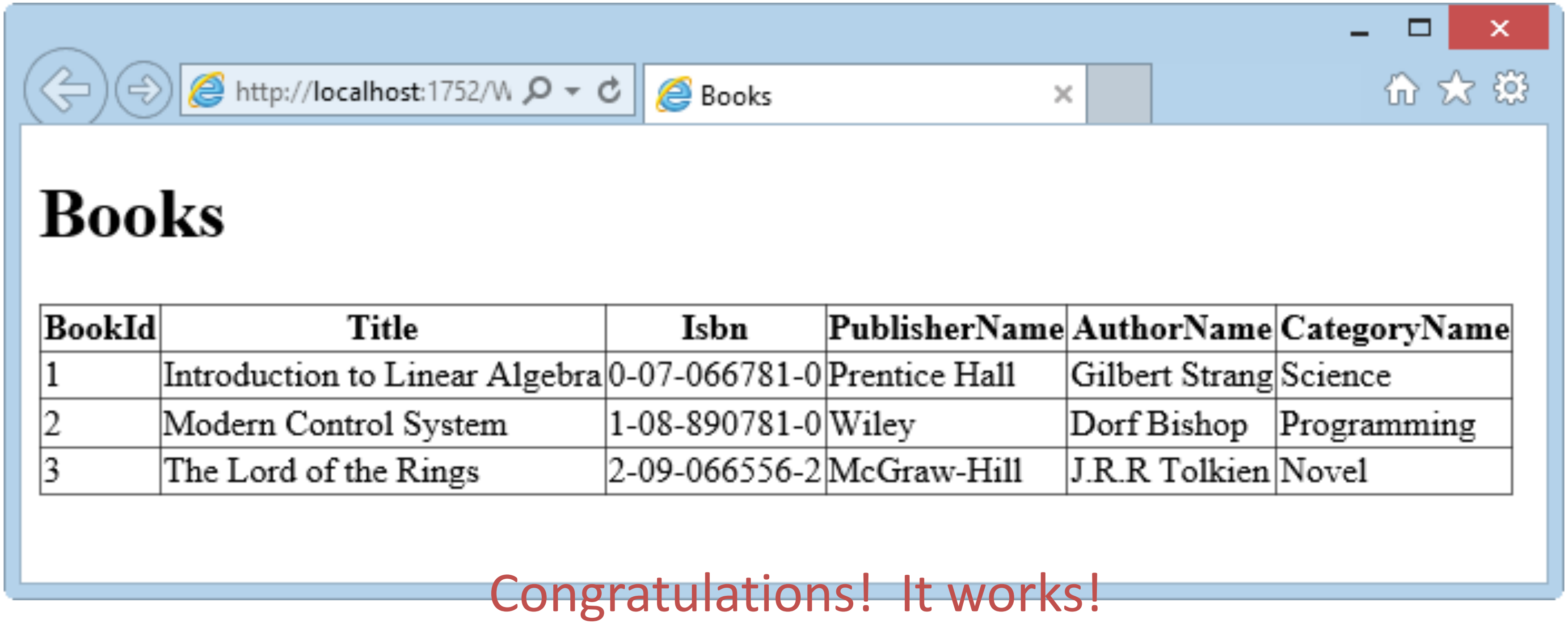
We shall create the Connection String to the Database in the "Web.config" File

Create Database Connection String in “Web.config”

```
 <connectionStrings>
    <add name="LibraryDBConnectionString"
            connectionString="Data Source=macwin8;Initial Catalog=BOOKS;Persist Security Info=True;User
ID=sa;Password=xxx"
            providerName="System.Data.SqlClient" />
  </connectionStrings>
```

Where “xxx” is your SQL Server Database Password

Finally, Run your application:

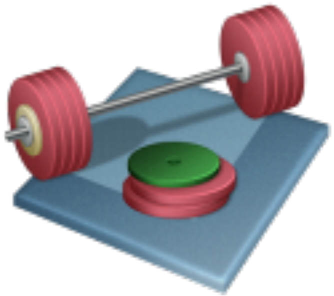


Books

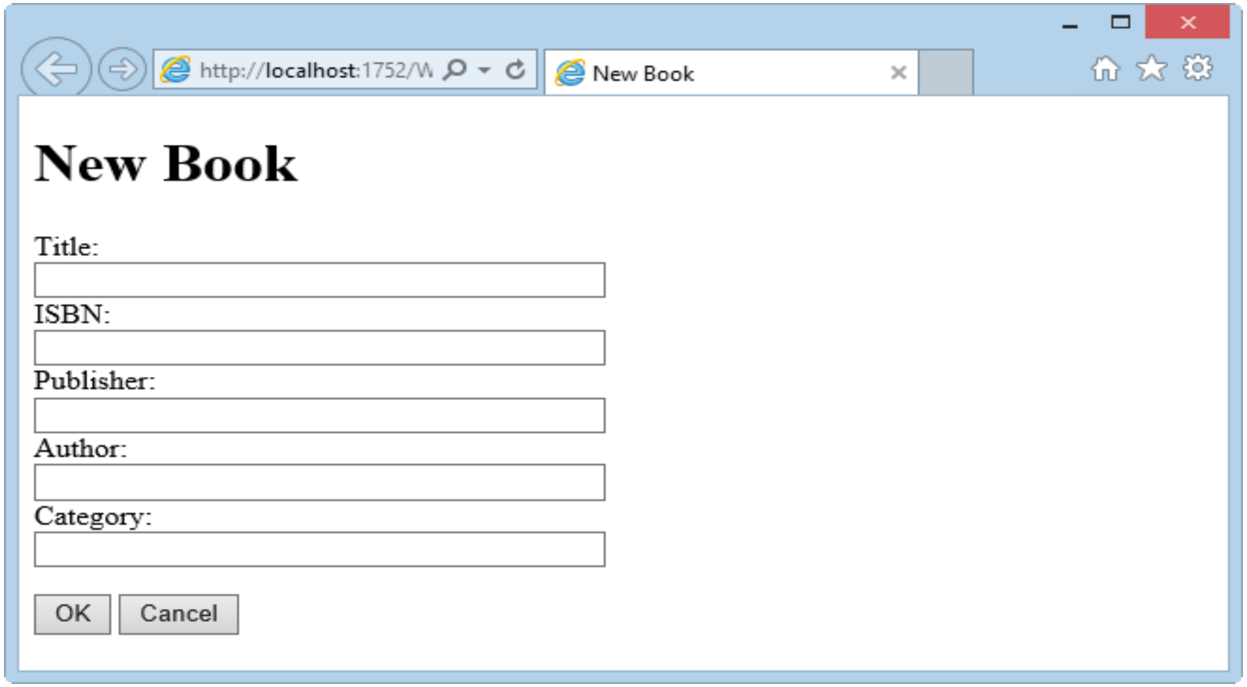
BookId	Title	Isbn	PublisherName	AuthorName	CategoryName
1	Introduction to Linear Algebra	0-07-066781-0	Prentice Hall	Gilbert Strang	Science
2	Modern Control System	1-08-890781-0	Wiley	Dorf Bishop	Programming
3	The Lord of the Rings	2-09-066556-2	McGraw-Hill	J.R.R Tolkien	Novel

Congratulations! It works!

You are finished with the Exercise



Create New Books



New Book

Title:

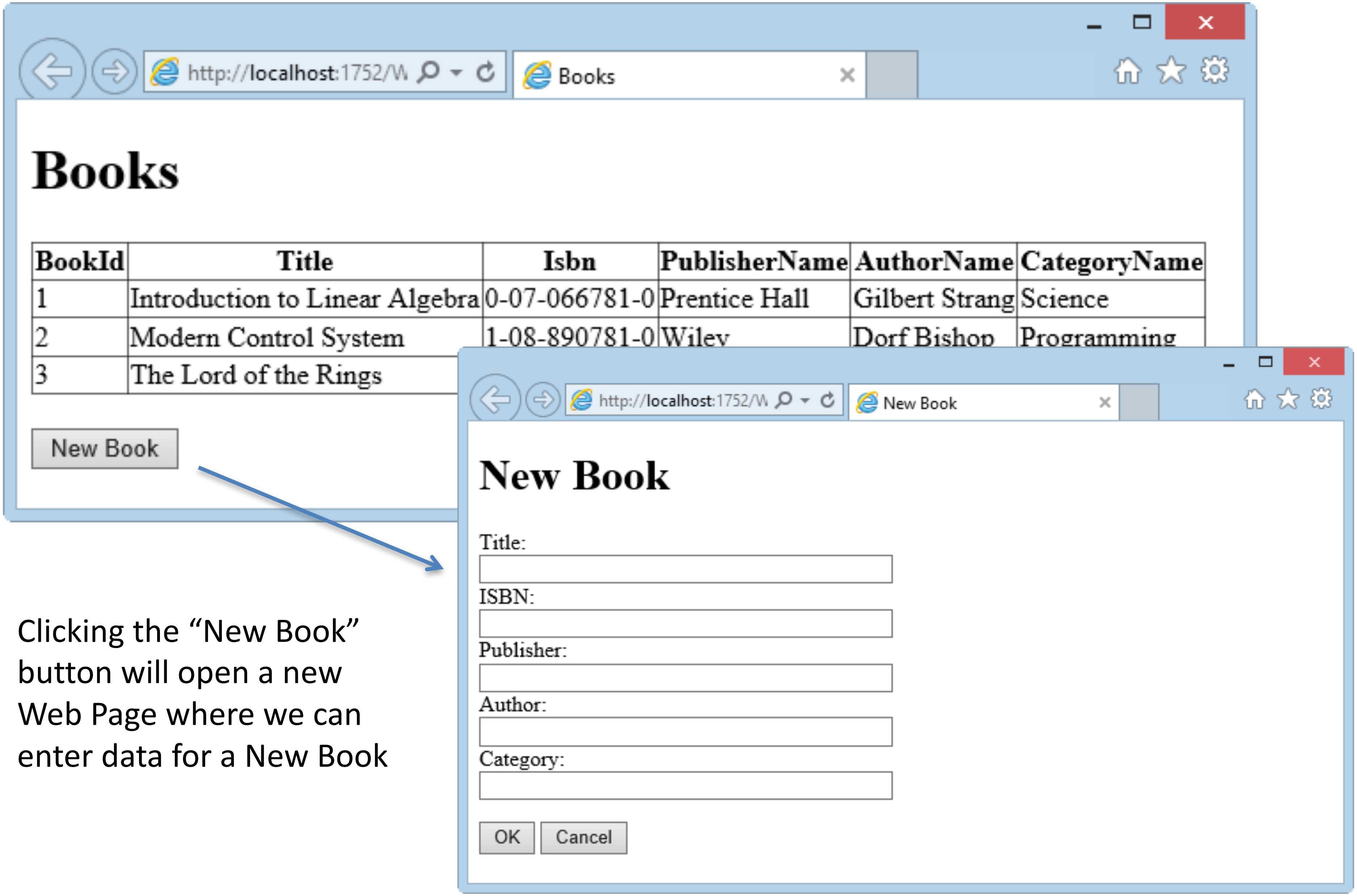
ISBN:

Publisher:

Author:

Category:

We will now add functionality for Creating New Books



Books

BookId	Title	Isbn	PublisherName	AuthorName	CategoryName
1	Introduction to Linear Algebra	0-07-066781-0	Prentice Hall	Gilbert Strang	Science
2	Modern Control System	1-08-890781-0	Wiley	Dorf Bishop	Programming
3	The Lord of the Rings				

New Book

Title:

ISBN:

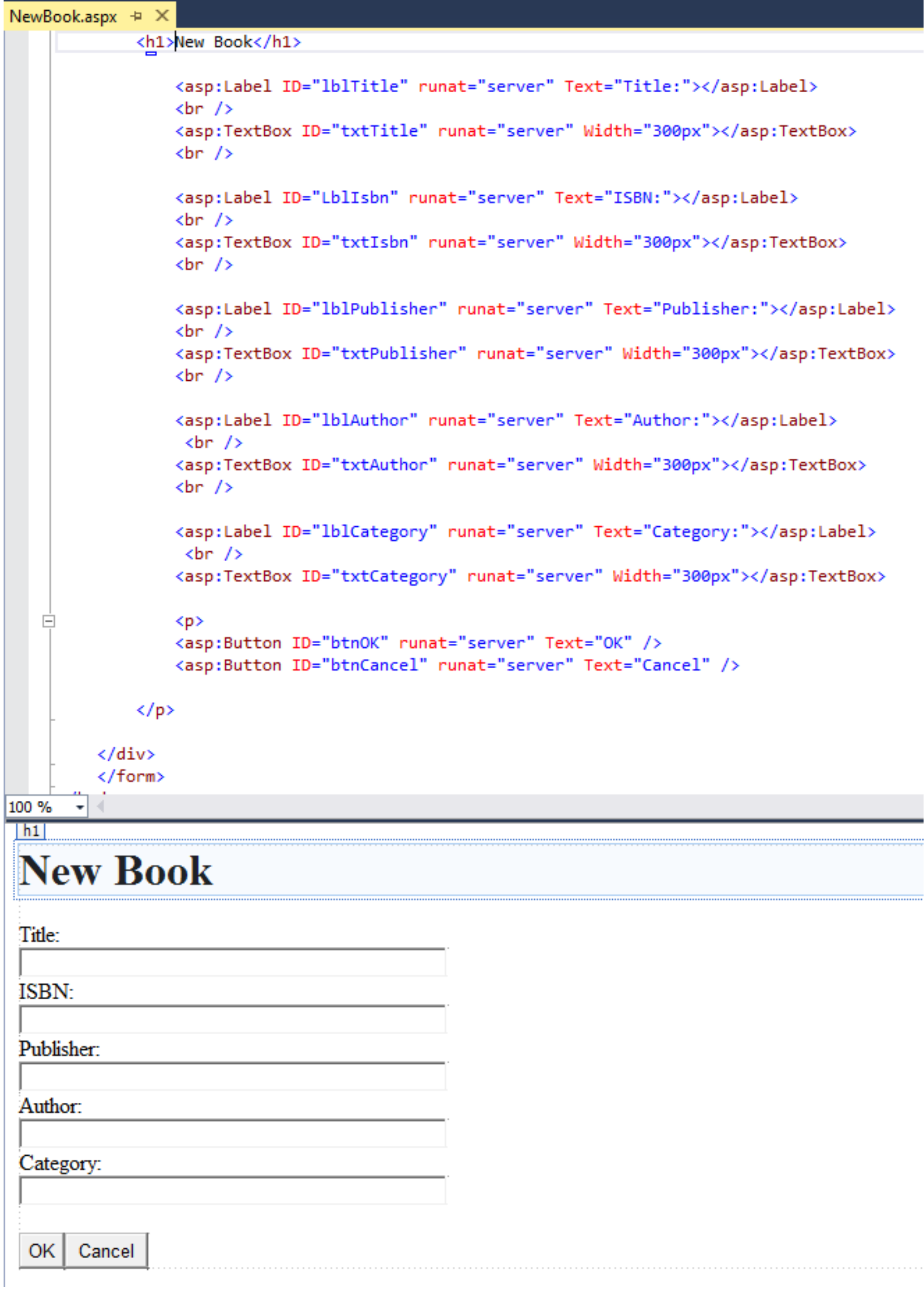
Publisher:

Author:

Category:

Clicking the “New Book” button will open a new Web Page where we can enter data for a New Book

Create New Web Form (NewBook.aspx)



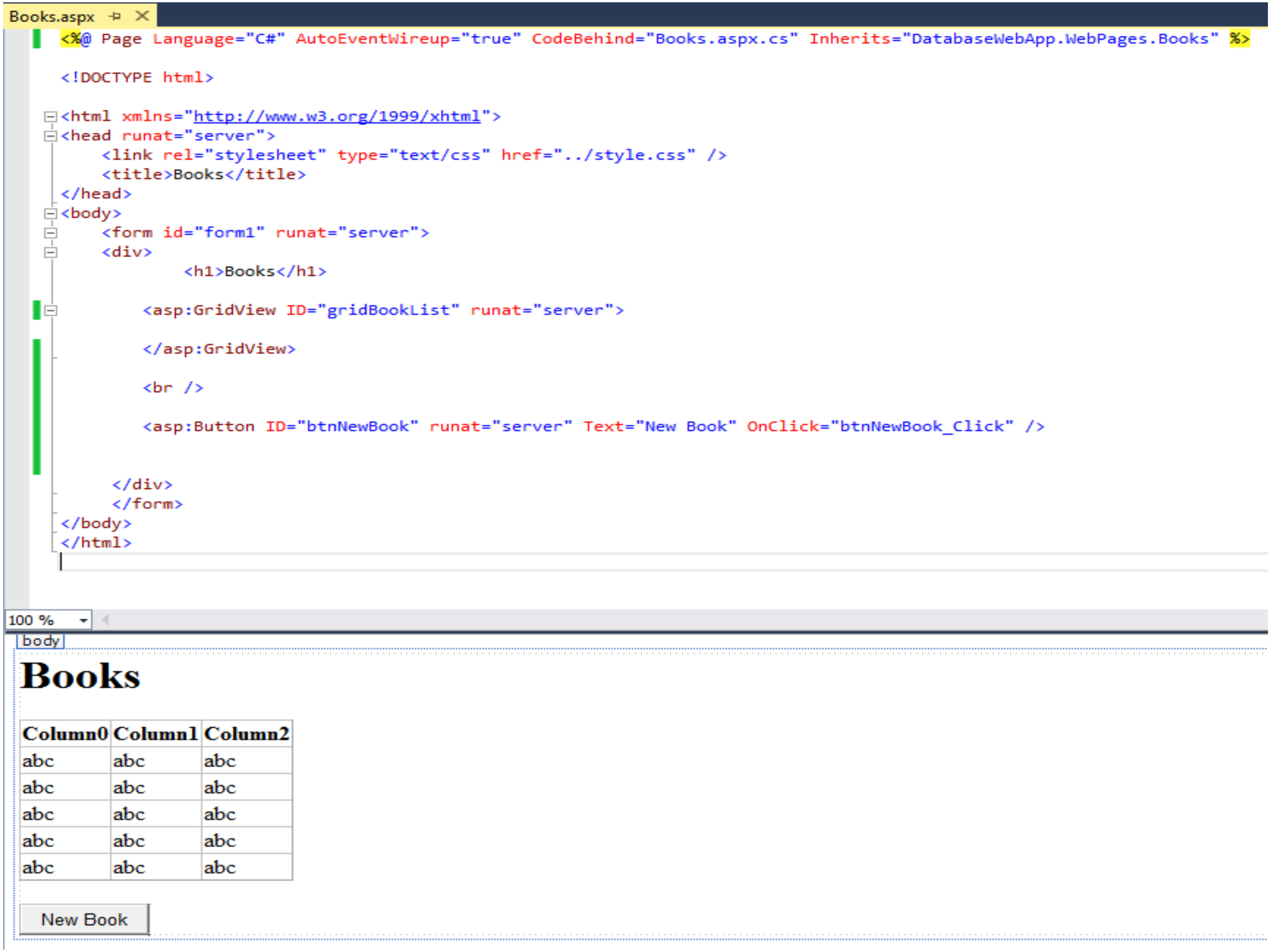
```
    <h1>New Book</h1>

        <asp:Label ID="lblTitle" runat="server" Text="Title:"></asp:Label>
        <br />
        <asp:TextBox ID="txtTitle" runat="server" Width="300px"></asp:TextBox>
        <br />

        <asp:Label ID="LblIsbn" runat="server" Text="ISBN:"></asp:Label>
        <br />
        <asp:TextBox ID="txtIsbn" runat="server" Width="300px"></asp:TextBox>
        <br />

        <asp:Label ID="lblPublisher" runat="server" Text="Publisher:"></asp:Label>
        <br />
        <asp:TextBox ID="txtPublisher" runat="server" Width="300px"></asp:TextBox>
        <br />

        <asp:Label ID="lblAuthor" runat="server" Text="Author:"></asp:Label>
         <br />
        <asp:TextBox ID="txtAuthor" runat="server" Width="300px"></asp:TextBox>
        <br />

        <asp:Label ID="lblCategory" runat="server" Text="Category:"></asp:Label>
         <br />
        <asp:TextBox ID="txtCategory" runat="server" Width="300px"></asp:TextBox>

        <p>
        <asp:Button ID="btnOK" runat="server" Text="OK" />
        <asp:Button ID="btnCancel" runat="server" Text="Cancel" />

    </p>

</div>
</form>
```

Add “TexBox”es and “Label”s

“NewBook.aspx”

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="NewBook.aspx.cs" Inherits="DatabaseWebApp.WebPages.NewBook" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>New Book</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>

        <h1>New Book</h1>

            <asp:Label ID="lblTitle" runat="server" Text="Title:"></asp:Label>
            <br />
            <asp:TextBox ID="txtTitle" runat="server" Width="300px"></asp:TextBox>
            <br />

            <asp:Label ID="LblIsbn" runat="server" Text="ISBN:"></asp:Label>
            <br />
            <asp:TextBox ID="txtIsbn" runat="server" Width="300px"></asp:TextBox>
            <br />

            <asp:Label ID="lblPublisher" runat="server" Text="Publisher:"></asp:Label>
            <br />
            <asp:TextBox ID="txtPublisher" runat="server" Width="300px"></asp:TextBox>
            <br />

            <asp:Label ID="lblAuthor" runat="server" Text="Author:"></asp:Label>
             <br />
            <asp:TextBox ID="txtAuthor" runat="server" Width="300px"></asp:TextBox>
            <br />

            <asp:Label ID="lblCategory" runat="server" Text="Category:"></asp:Label>
             <br />
            <asp:TextBox ID="txtCategory" runat="server" Width="300px"></asp:TextBox>

            <p />
            <asp:Button ID="btnOK" runat="server" Text="OK" />
            <asp:Button ID="btnCancel" runat="server" Text="Cancel" />

    </div>
    </form>
</body>
</html>
```

Add a Button “New Book” in “Books.aspx”



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Books.aspx.cs" Inherits="DatabaseWebApp.WebPages.Books" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <link rel="stylesheet" type="text/css" href="../style.css" />
    <title>Books</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
            <h1>Books</h1>

        <asp:GridView ID="gridBookList" runat="server">

        </asp:GridView>

        <br />

        <asp:Button ID="btnNewBook" runat="server" Text="New Book" OnClick="btnNewBook_Click" />


     </div>
     </form>
</body>
</html>
```

Books

Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

Create a Button Event Handler in “Books.aspx”

Double-Click on the “New Book” Button in order to create an Event Handler for the Button.

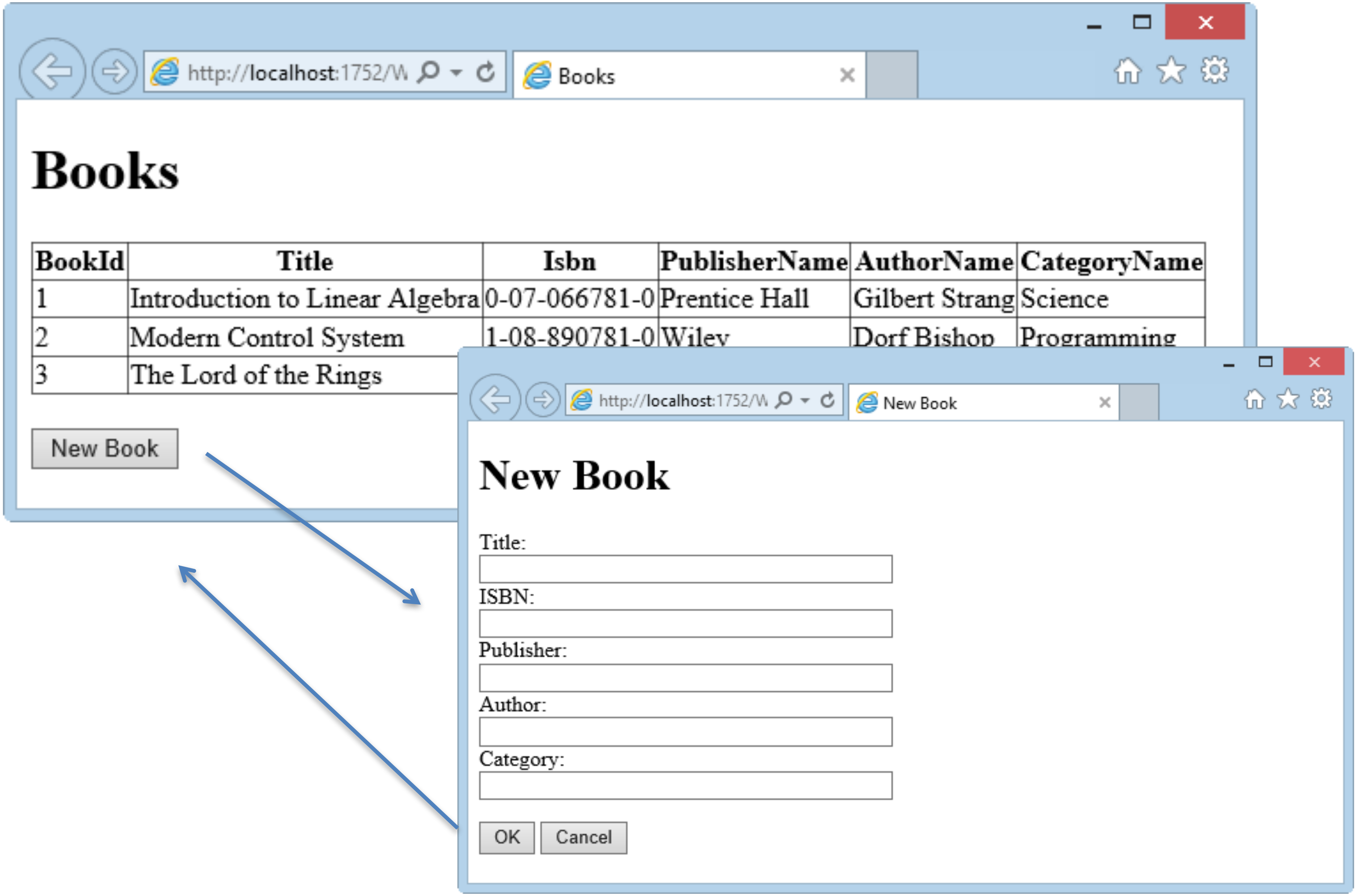
Create the following Code inside the Event Handler:

```
protected void btnNewBook_Click(object sender, EventArgs e)
{
   Response.Redirect("NewBook.aspx");
}
```

Create a Button Event Handler in “NewBooks.aspx”

```
protected void btnOK_Click(object sender, EventArgs e)
{
    Response.Redirect("Books.aspx");
}

protected void btnCancel_Click(object sender, EventArgs e)
{
    Response.Redirect("Books.aspx");
}
```

Run your Application (F5) to check that it works



Books

BookId	Title	Isbn	PublisherName	AuthorName	CategoryName
1	Introduction to Linear Algebra	0-07-066781-0	Prentice Hall	Gilbert Strang	Science
2	Modern Control System	1-08-890781-0	Wiley	Dorf Bishop	Programming
3	The Lord of the Rings				

New Book

Title:

ISBN:

Publisher:

Author:

Category:

Next, Create Database Logic for Creating New Books

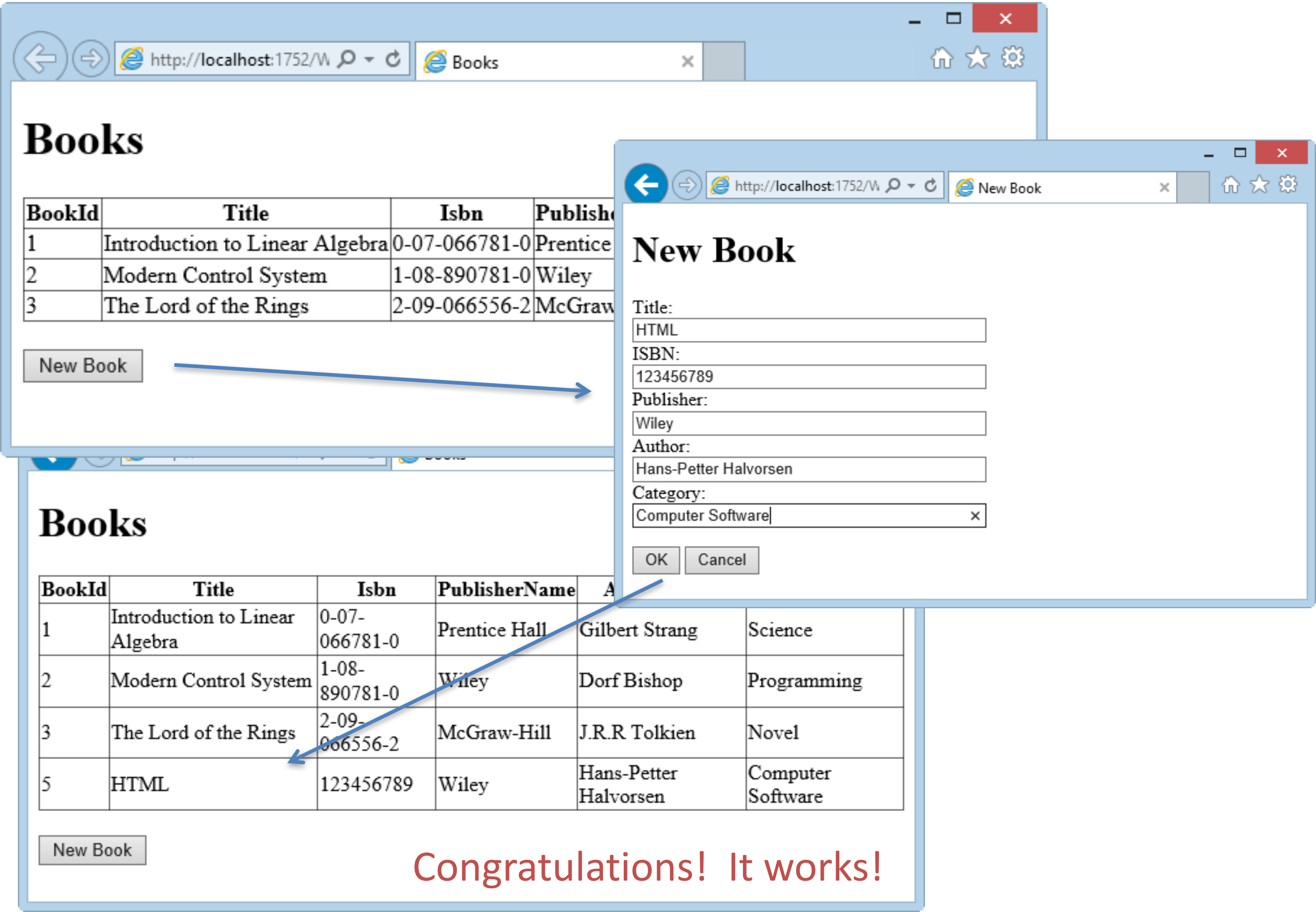
Create a New Method in “Books.cs”

```
public void CreateBook(string connectionString, Book book)
{
           try
           {
                using (SqlConnection con = new SqlConnection(connectionString))
                {
                    SqlCommand cmd = new SqlCommand("CreateBook", con);
                    cmd.CommandType = CommandType.StoredProcedure;

                    cmd.Parameters.Add(new SqlParameter("@Title", book.Title));
                    cmd.Parameters.Add(new SqlParameter("@Isbn", book.Isbn));
                    cmd.Parameters.Add(new SqlParameter("@PublisherName", book.PublisherName));
                    cmd.Parameters.Add(new SqlParameter("@AuthorName", book.AuthorName));
                    cmd.Parameters.Add(new SqlParameter("@CategoryName", book.CategoryName));

                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                }
            }
            catch (Exception ex)
            {
                throw ex;
            }
}
```

Create/Update Code in NewBook.aspx.cs

```
...
using System.Web.Configuration;
using DatabaseWebApp.Data;

namespace DatabaseWebApp.WebPages
{
    public partial class NewBook : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        private string connectionString =
             WebConfigurationManager.ConnectionStrings["LibraryDBConnectionString"].ConnectionString;

        protected void btnOK_Click(object sender, EventArgs e)
        {
            Book book = new Book();

            book.Title = txtTitle.Text;
            book.Isbn = txtIsbn.Text;
            book.PublisherName = txtPublisher.Text;
            book.AuthorName = txtAuthor.Text;
            book.CategoryName = txtCategory.Text;

            book.CreateBook(connectionString, book);

            Response.Redirect("Books.aspx");
        }

        protected void btnCancel_Click(object sender, EventArgs e)
        {
            Response.Redirect("Books.aspx");
        }
    }
}
```

Test your Application (F5)



Books

BookId	Title	Isbn	Publish
1	Introduction to Linear Algebra	0-07-066781-0	Prentice
2	Modern Control System	1-08-890781-0	Wiley
3	The Lord of the Rings	2-09-066556-2	McGraw

New Book

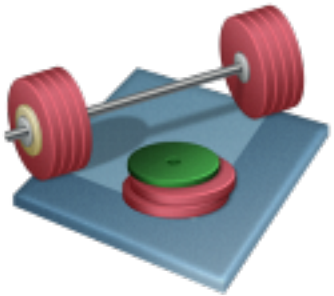
Title: HTML
ISBN: 123456789
Publisher: Wiley
Author: Hans-Petter Halvorsen
Category: Computer Software

Books

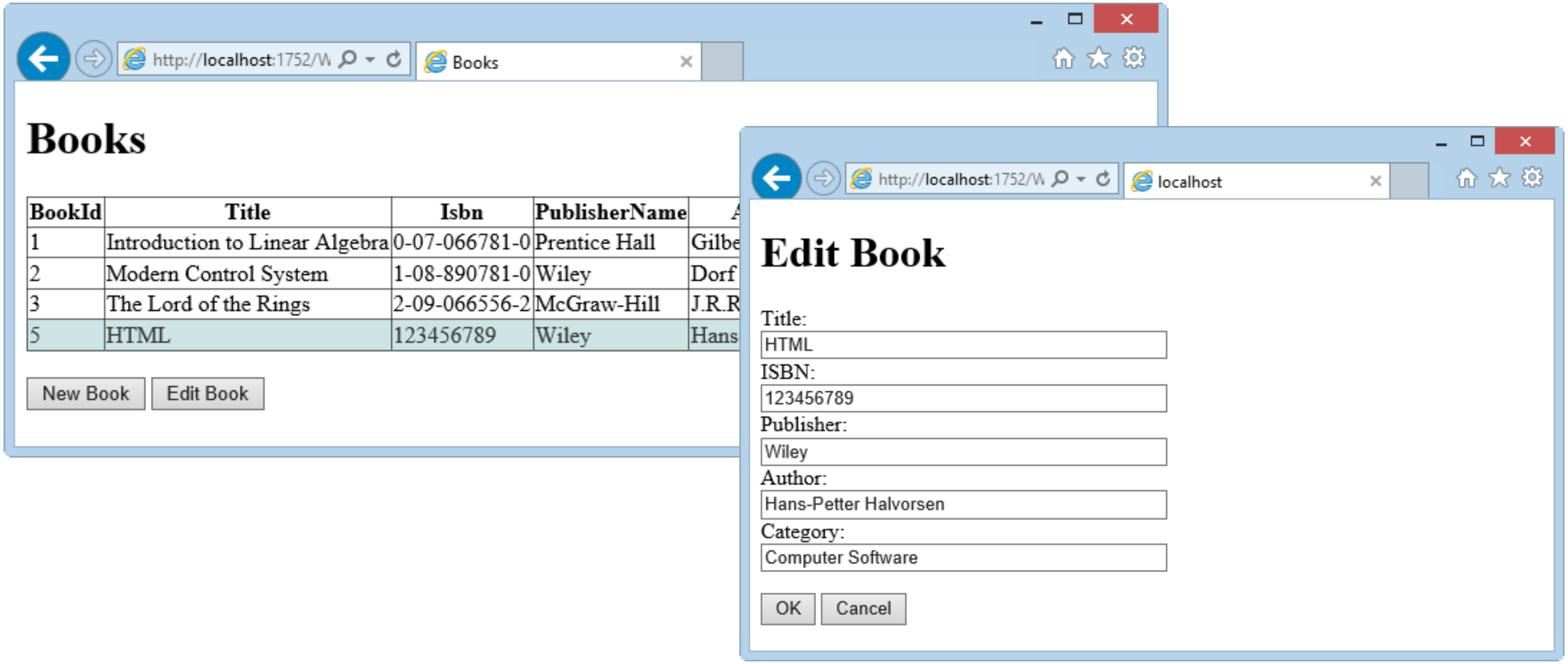
BookId	Title	Isbn	PublisherName	A	
1	Introduction to Linear Algebra	0-07-066781-0	Prentice Hall	Gilbert Strang	Science
2	Modern Control System	1-08-890781-0	Wiley	Dorf Bishop	Programming
3	The Lord of the Rings	2-09-066556-2	McGraw-Hill	J.R.R Tolkien	Novel
5	HTML	123456789	Wiley	Hans-Petter Halvorsen	Computer Software

Congratulations! It works!

You are finished with the Exercise



Edit/Update Books



Books

BookId	Title	Isbn	PublisherName	A
1	Introduction to Linear Algebra	0-07-066781-0	Prentice Hall	Gilbe
2	Modern Control System	1-08-890781-0	Wiley	Dorf
3	The Lord of the Rings	2-09-066556-2	McGraw-Hill	J.R.R
5	HTML	123456789	Wiley	Hans

Edit Book

Title:
HTML

ISBN:
123456789

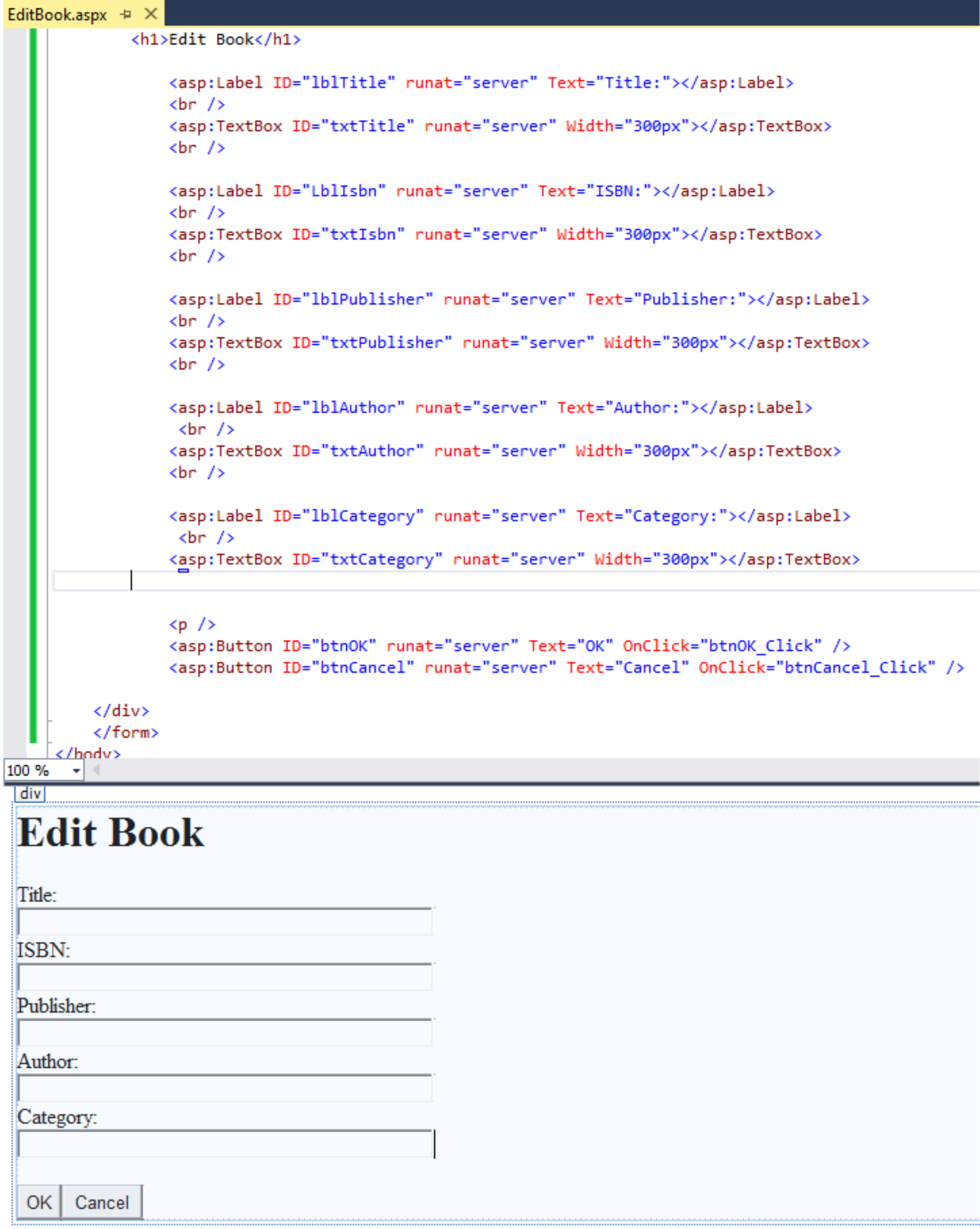
Publisher:
Wiley

Author:
Hans-Petter Halvorsen

Category:
Computer Software

Create New Web Form (EditBook.aspx)

(Similiar as NewBook.aspx)



```
        <h1>Edit Book</h1>

            <asp:Label ID="lblTitle" runat="server" Text="Title:"></asp:Label>
            <br />
            <asp:TextBox ID="txtTitle" runat="server" Width="300px"></asp:TextBox>
            <br />

            <asp:Label ID="LblIsbn" runat="server" Text="ISBN:"></asp:Label>
            <br />
            <asp:TextBox ID="txtIsbn" runat="server" Width="300px"></asp:TextBox>
            <br />

            <asp:Label ID="lblPublisher" runat="server" Text="Publisher:"></asp:Label>
            <br />
            <asp:TextBox ID="txtPublisher" runat="server" Width="300px"></asp:TextBox>
            <br />

            <asp:Label ID="lblAuthor" runat="server" Text="Author:"></asp:Label>
             <br />
            <asp:TextBox ID="txtAuthor" runat="server" Width="300px"></asp:TextBox>
            <br />

            <asp:Label ID="lblCategory" runat="server" Text="Category:"></asp:Label>
             <br />
            <asp:TextBox ID="txtCategory" runat="server" Width="300px"></asp:TextBox>


            <p />
            <asp:Button ID="btnOK" runat="server" Text="OK" OnClick="btnOK_Click" />
            <asp:Button ID="btnCancel" runat="server" Text="Cancel" OnClick="btnCancel_Click" />

    </div>
    </form>
</body>
```

Edit Book

Title:

ISBN:

Publisher:

Author:

Category:

Add “Edit Book”Button in “Books.aspx”
Add Event Handler for the Button, etc.

```
... //Existing Code

namespace DatabaseWebApp.WebPages
{
    public partial class Books : System.Web.UI.Page
    {

        .... //Existing Code

        protected void btnEditBook_Click(object sender, EventArgs e)
        {
            string bookId = null;

            if (gridBookList.SelectedIndex != -1)
                bookId = gridBookList.SelectedRow.Cells[0].Text;

            Response.Redirect("EditBook.aspx?bookid=" + bookId);

        }

        protected override void Render(HtmlTextWriter writer)
        {
            foreach (GridViewRow row in gridBookList.Rows)
            {
                row.Attributes.Add("onclick", Page.ClientScript.GetPostBackEventReference
                    (gridBookList, "Select$" + row.RowIndex.ToString(), true));
            }

            base.Render(writer);
        }

        protected void gridBookList_RowDataBound(object sender, GridViewRowEventArgs e)
        {
            if (e.Row.RowType == DataControlRowType.DataRow)
            {
                e.Row.Attributes.Add("OnMouseOver", "this.style.cursor='pointer';");
                e.Row.ToolTip = "Click on select row";
            }
        }

    }
}
```

Next, Create Database Logic for Edit Books

Create a “**GetBookData**” Method in “Books.cs” Class

```
public Book GetBookData(string connectionString, int bookId)
{

       SqlConnection con = new SqlConnection(connectionString);

       string selectSQL = "select BookId, Title, Isbn, PublisherName, AuthorName, CategoryName  from GetBookData where BookId = " + bookId;
       con.Open();

       SqlCommand cmd = new SqlCommand(selectSQL, con);

       SqlDataReader dr = cmd.ExecuteReader();

       Book book = new Book();

       if (dr != null)
       {
           while (dr.Read())
           {
               book.BookId = Convert.ToInt32(dr["BookId"]);
               book.Title = dr["Title"].ToString();
               book.Isbn = dr["ISBN"].ToString();
               book.PublisherName = dr["PublisherName"].ToString();
               book.AuthorName = dr["AuthorName"].ToString();
               book.CategoryName = dr["CategoryName"].ToString();
            }
       }
       return book;
}
```

Next, Create Database Logic for Edit Books

Create a “**EditBook**”Method in “Books.cs” Class

```
public void EditBook(string connectionString, Book book)
{
          try
          {
              using (SqlConnection con = new SqlConnection(connectionString))
              {
                  SqlCommand cmd = new SqlCommand("UpdateBook", con);
                  cmd.CommandType = CommandType.StoredProcedure;

                  cmd.Parameters.Add(new SqlParameter("@BookId", book.BookId));
                  cmd.Parameters.Add(new SqlParameter("@Title", book.Title));
                  cmd.Parameters.Add(new SqlParameter("@Isbn", book.Isbn));
                  cmd.Parameters.Add(new SqlParameter("@PublisherName", book.PublisherName));
                  cmd.Parameters.Add(new SqlParameter("@AuthorName", book.AuthorName));
                  cmd.Parameters.Add(new SqlParameter("@CategoryName", book.CategoryName));

                  con.Open();
                  cmd.ExecuteNonQuery();
                  con.Close();
              }

          }
          catch (Exception ex)
          {
              throw ex;
          }

}
```

Create Code in EditBook.aspx.cs

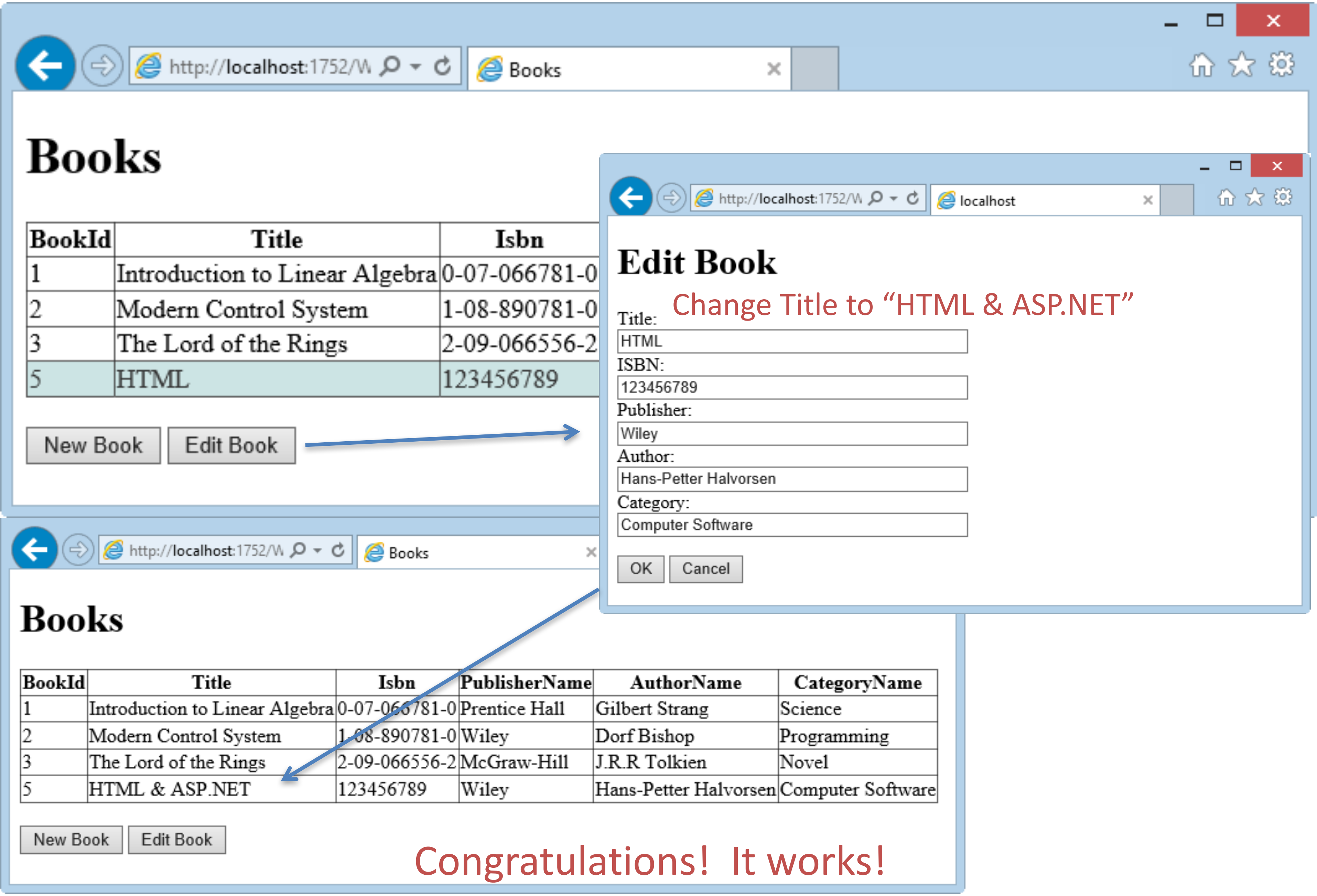
```
...

using System.Web.Configuration;
using DatabaseWebApp.Data;

namespace DatabaseWebApp.WebPages
{
    public partial class EditBook : System.Web.UI.Page
    {
        private string connectionString = WebConfigurationManager.ConnectionStrings["LibraryDBConnectionString"].ConnectionString;

        int bookId = 0;

        protected void Page_Load(object sender, EventArgs e)
        {

            if (Request.QueryString["bookid"] != "")
                bookId = Convert.ToInt16(Request.QueryString["bookid"]);
            else
                Response.Redirect("Books.aspx");

            if (!IsPostBack)
            {
                FillBookData();
            }
        }

        private void FillBookData()
        {

            Book book = new Book();

            book = book.GetBookData(connectionString, bookId);

            txtTitle.Text = book.Title;
            txtIsbn.Text = book.Isbn;
            txtPublisher.Text = book.PublisherName;
            txtAuthor.Text = book.AuthorName;
            txtCategory.Text = book.CategoryName;
        }

        // Continues next slide
    }
}
```

See Next Slide for rest of the code

Create Code in EditBook.aspx.cs

```
... //See Previous Slide


protected void btnOK_Click(object sender, EventArgs e)
{

     Book book = new Book();

     book.BookId = bookId;

     book.Title = txtTitle.Text;
     book.Isbn = txtIsbn.Text;
     book.PublisherName = txtPublisher.Text;
     book.AuthorName = txtAuthor.Text;
     book.CategoryName = txtCategory.Text;

     book.EditBook(connectionString, book);

    Response.Redirect("Books.aspx");

}


protected void btnCancel_Click(object sender, EventArgs e)
{
     Response.Redirect("Books.aspx");
}
```

Run your Application (F5)



Books

BookId	Title	Isbn
1	Introduction to Linear Algebra	0-07-066781-0
2	Modern Control System	1-08-890781-0
3	The Lord of the Rings	2-09-066556-2
5	HTML	123456789

Edit Book

Change Title to "HTML & ASP.NET"

Title: HTML

ISBN: 123456789

Publisher: Wiley

Author: Hans-Petter Halvorsen

Category: Computer Software

Books

BookId	Title	Isbn	PublisherName	AuthorName	CategoryName
1	Introduction to Linear Algebra	0-07-066781-0	Prentice Hall	Gilbert Strang	Science
2	Modern Control System	1-08-890781-0	Wiley	Dorf Bishop	Programming
3	The Lord of the Rings	2-09-066556-2	McGraw-Hill	J.R.R Tolkien	Novel
5	HTML & ASP.NET	123456789	Wiley	Hans-Petter Halvorsen	Computer Software

Congratulations! It works!

You are finished with the Exercise

Delete Books

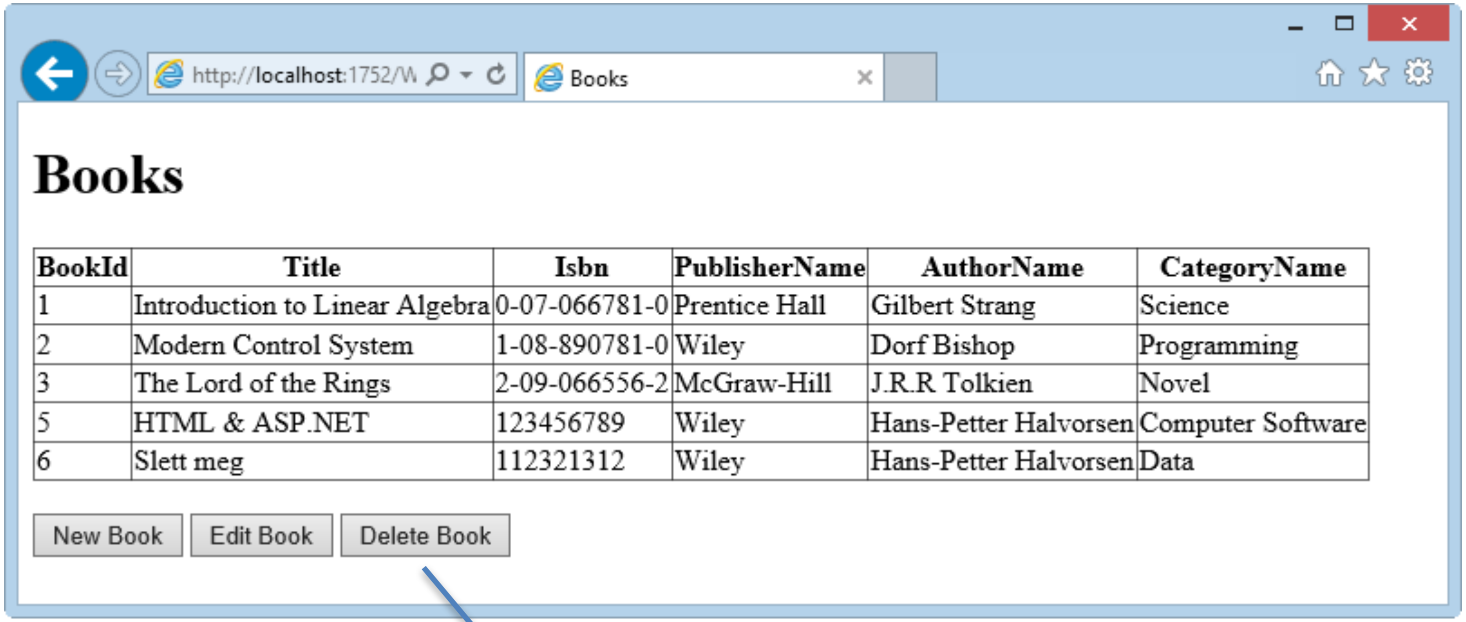
“Books.cs” Class: Add “DeleteBook” Method

```
public void DeleteBook(string connectionString, int bookId)
{
       try
       {
           using (SqlConnection con = new SqlConnection(connectionString))
           {
               SqlCommand cmd = new SqlCommand("DeleteBook", con);
               cmd.CommandType = CommandType.StoredProcedure;

               cmd.Parameters.Add(new SqlParameter("@BookId", bookId));

               con.Open();
               cmd.ExecuteNonQuery();
               con.Close();
           }

       }
       catch (Exception ex)
       {
           throw ex;
       }
}
```

Books.aspx: Add “Delete Book” Button, Event Handler, etc.

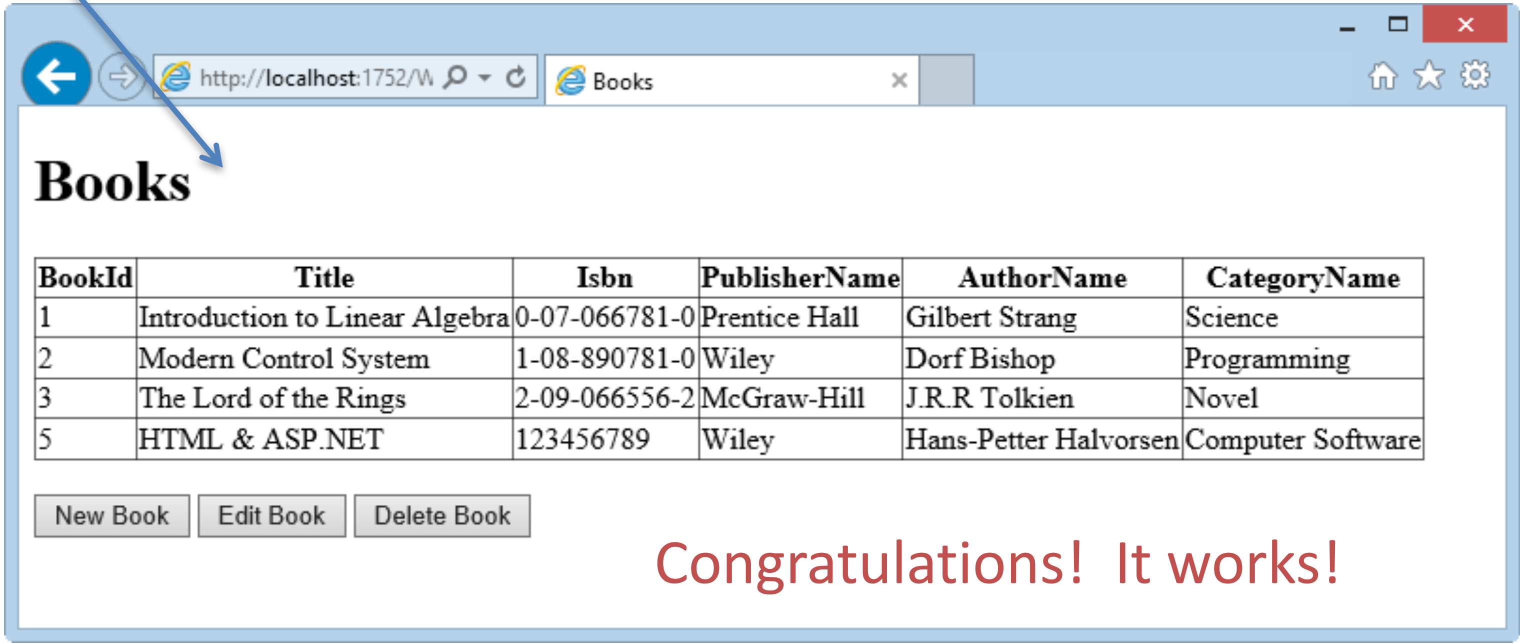
```
protected void btnDeleteBook_Click(object sender, EventArgs e)
{
       int bookId = 0;

       if (gridBookList.SelectedIndex != -1)
       {
            bookId = Convert.ToInt32(gridBookList.SelectedRow.Cells[0].Text);

            Book book = new Book();

            book.DeleteBook(connectionString, bookId);

            FillBookGrid();

        }

}
```

Run your Application (F5)



Books

BookId	Title	Isbn	PublisherName	AuthorName	CategoryName
1	Introduction to Linear Algebra	0-07-066781-0	Prentice Hall	Gilbert Strang	Science
2	Modern Control System	1-08-890781-0	Wiley	Dorf Bishop	Programming
3	The Lord of the Rings	2-09-066556-2	McGraw-Hill	J.R.R Tolkien	Novel
5	HTML & ASP.NET	123456789	Wiley	Hans-Petter Halvorsen	Computer Software
6	Slett meg	112321312	Wiley	Hans-Petter Halvorsen	Data

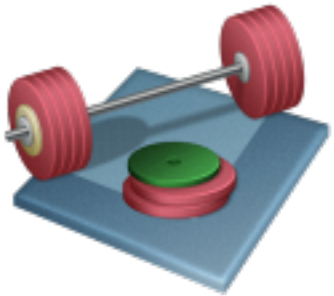


Books

BookId	Title	Isbn	PublisherName	AuthorName	CategoryName
1	Introduction to Linear Algebra	0-07-066781-0	Prentice Hall	Gilbert Strang	Science
2	Modern Control System	1-08-890781-0	Wiley	Dorf Bishop	Programming
3	The Lord of the Rings	2-09-066556-2	McGraw-Hill	J.R.R Tolkien	Novel
5	HTML & ASP.NET	123456789	Wiley	Hans-Petter Halvorsen	Computer Software

Congratulations! It works!

You are finished with the Exercise



Formatting

Format your Web Pages using CSS, etc.

Add a Style Sheet (“style.css”) to your Project

Add New Item, Then Select Style Sheet

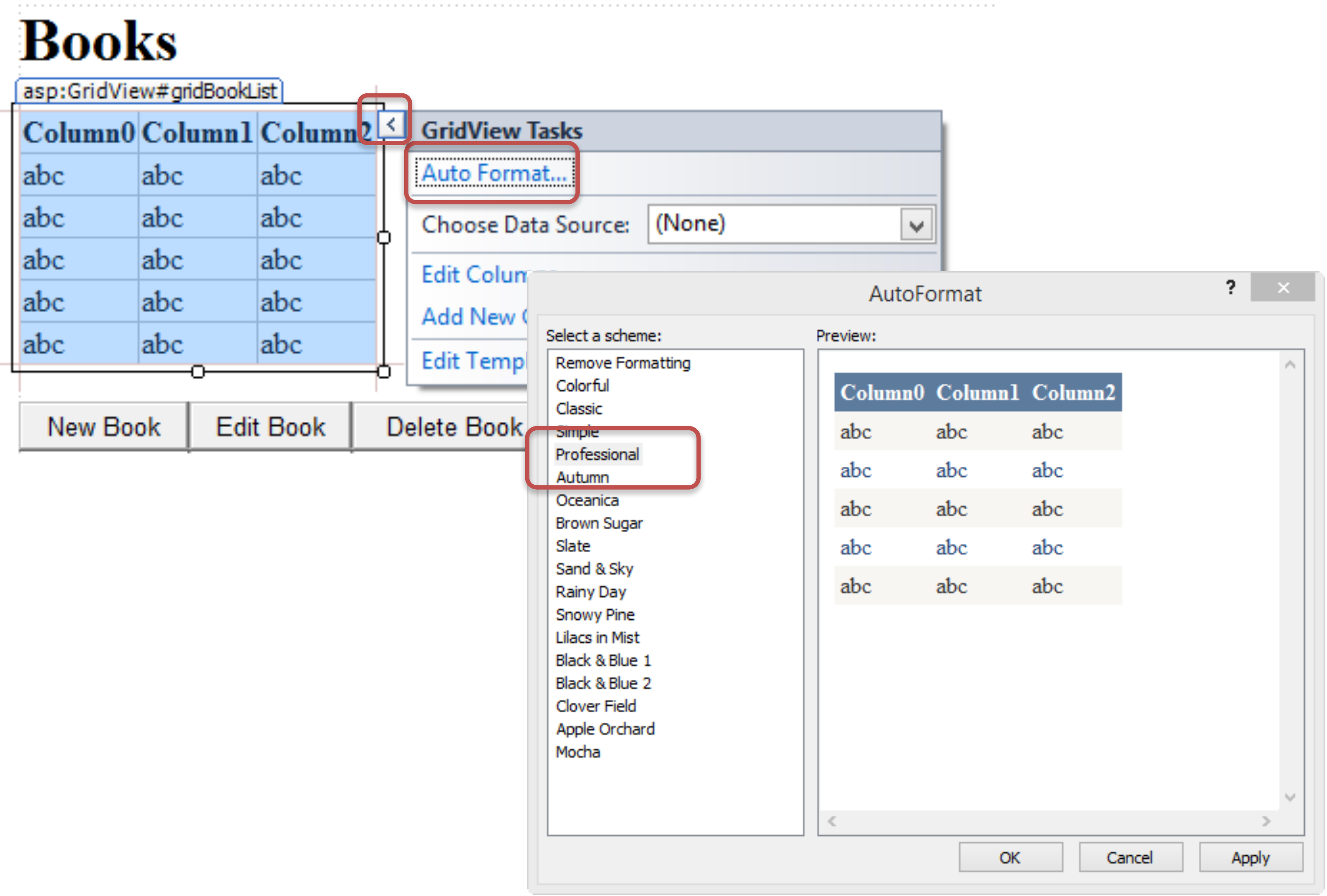
Enter the following code into style.css:

```
body
{
font-size:75%;
font-family:verdana,arial,'sans serif';
background-color:#B3C3CA;
color:#153744;
margin:10px;
}

h1 {font-size:250%; color:#05A30E;}
h2 {font-size:220%; color:#C3544E;}
h3 {font-size:200%; color:#76923C;}
```

Link the Style Sheet into every Web Page (“Books.aspx”, “NewBook.aspx”, “EditBook.aspx”)

```
...
<head runat="server">
...
    <link rel="stylesheet" type="text/css" href="../style.css" />
    ..
</head>
...
```

Format GridView



Books

asp:GridView#gridBookList

Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

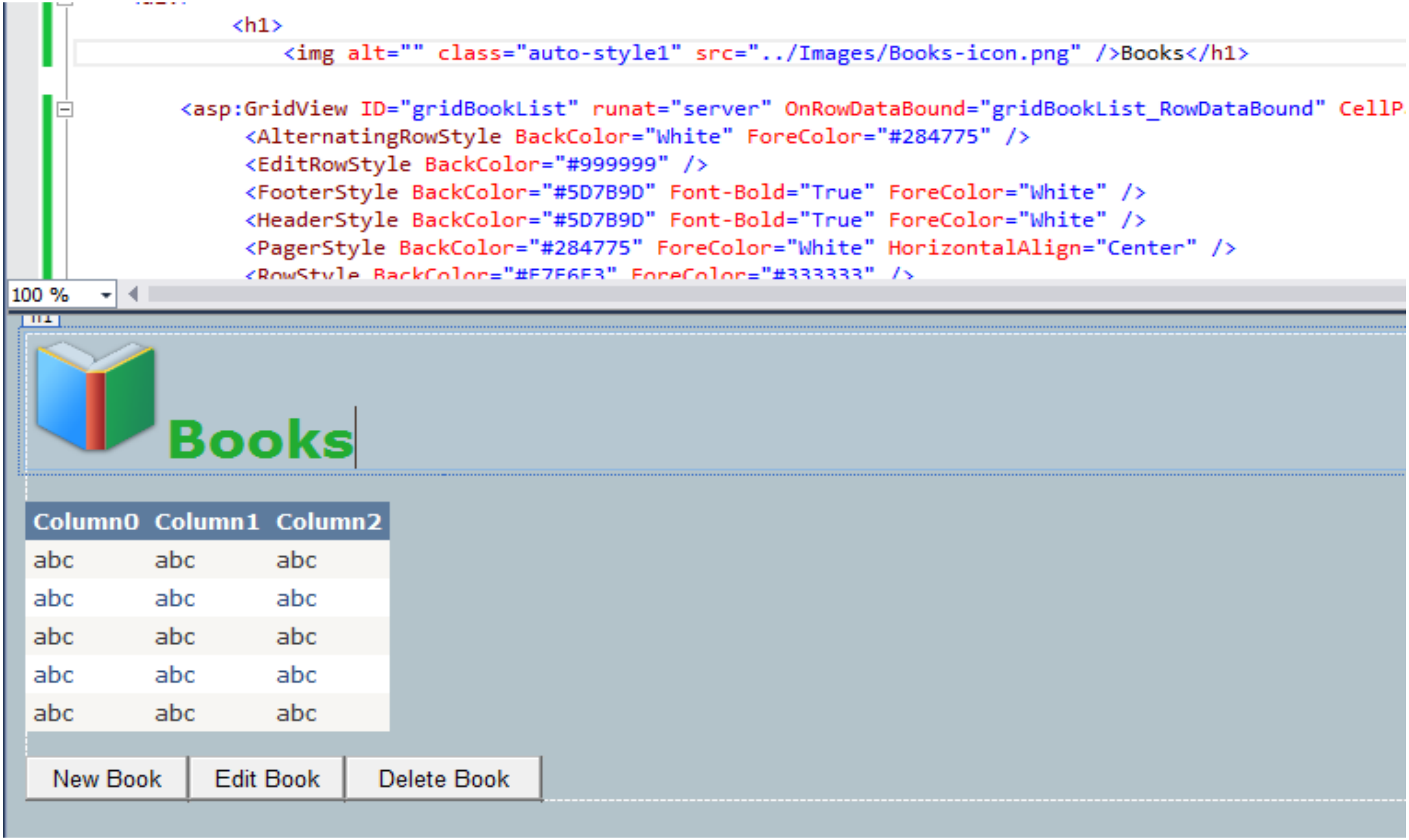
Choose Data Source: (None)

Preview:

Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

Add an Image on the Web Pages

Find an image on the Internet or something...

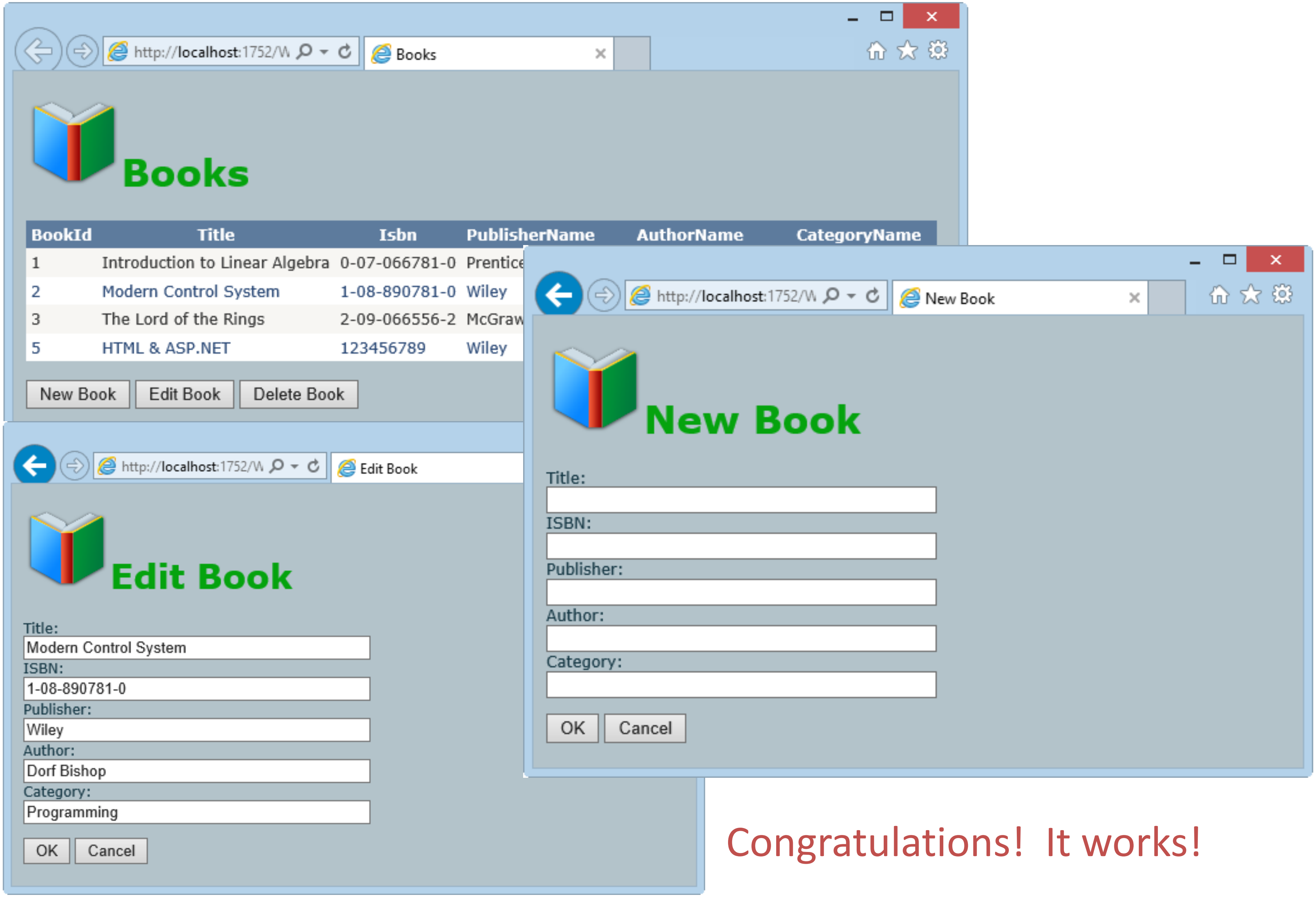


```
            <h1>
                <img alt="" class="auto-style1" src="../Images/Books-icon.png" />Books</h1>

        <asp:GridView ID="gridBookList" runat="server" OnRowDataBound="gridBookList_RowDataBound" CellP
            <AlternatingRowStyle BackColor="White" ForeColor="#284775" />
            <EditRowStyle BackColor="#999999" />
            <FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
            <HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
            <PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
            <RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
```

Books

Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

Run your Application (F5)



Books

BookId	Title	Isbn	PublisherName	AuthorName	CategoryName
1	Introduction to Linear Algebra	0-07-066781-0	Prentice		
2	Modern Control System	1-08-890781-0	Wiley		
3	The Lord of the Rings	2-09-066556-2	McGraw		
5	HTML & ASP.NET	123456789	Wiley		

Edit Book

Title:
Modern Control System
ISBN:
1-08-890781-0
Publisher:
Wiley
Author:
Dorf Bishop
Category:
Programming

New Book

Title:

ISBN:

Publisher:

Author:

Category:

Congratulations! It works!

You are finished with the Exercise

Hans-Petter Halvorsen, M.Sc.

University College of Southeast Norway

www.usn.no

E-mail: hans.p.halvorsen@hit.no

Blog: http://home.hit.no/~hansha/