Tutorial #1: Getting Started with ASP.NET

This is the first of a series of tutorials that will teach you how to build useful, real-world websites with dynamic content in a fun and easy way, using ASP.NET and C#. ASP.NET pages can display HTML, collect user input, and interact with databases.

Note that these tutorials are not meant to provide an in-depth training in ASP.NET programming. The goal is to walk you through the development steps of a simple project, which you may later follow in designing an interface to your own final database project for this class.

If you are interested in further study of web development with ASP.NET, I recommend the book by Scott Mitchell, *SAMS Teach Yourself ASP.Net in 24 Hours*, Pearson Education, 2010, ISBN 978-0-672-33305-7.

Step 1. Installing the Software Environment

Prior to using ASP.NET, you will need to install the following software:

- SQL Server 2008 R2 for the database engine.
- SQL Server Management Studio Express for the client (SSMS)
- Visual Web Developer Express 2010 for the IDE
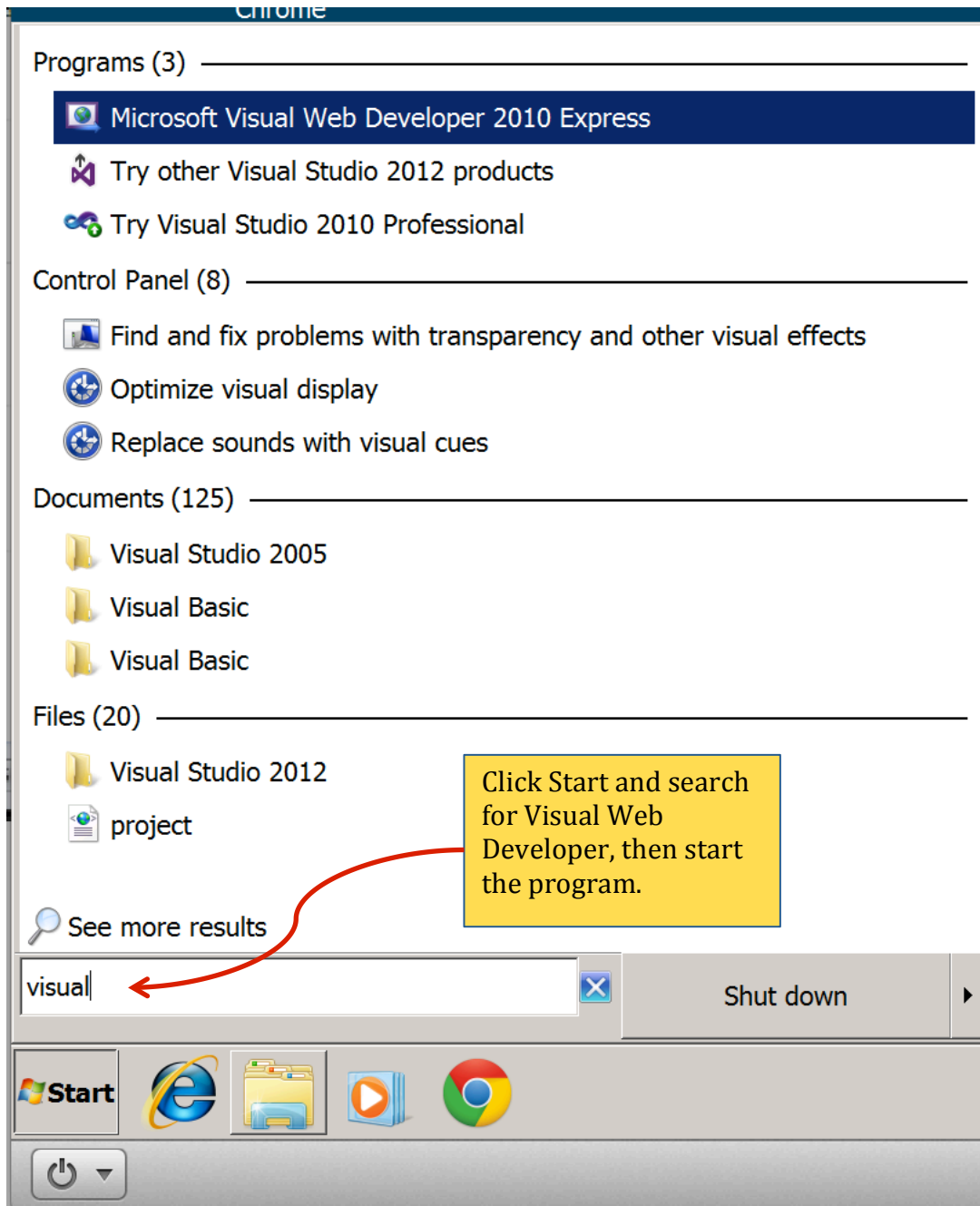- The .NET framework 4 environment

To install this software, visit the following website:
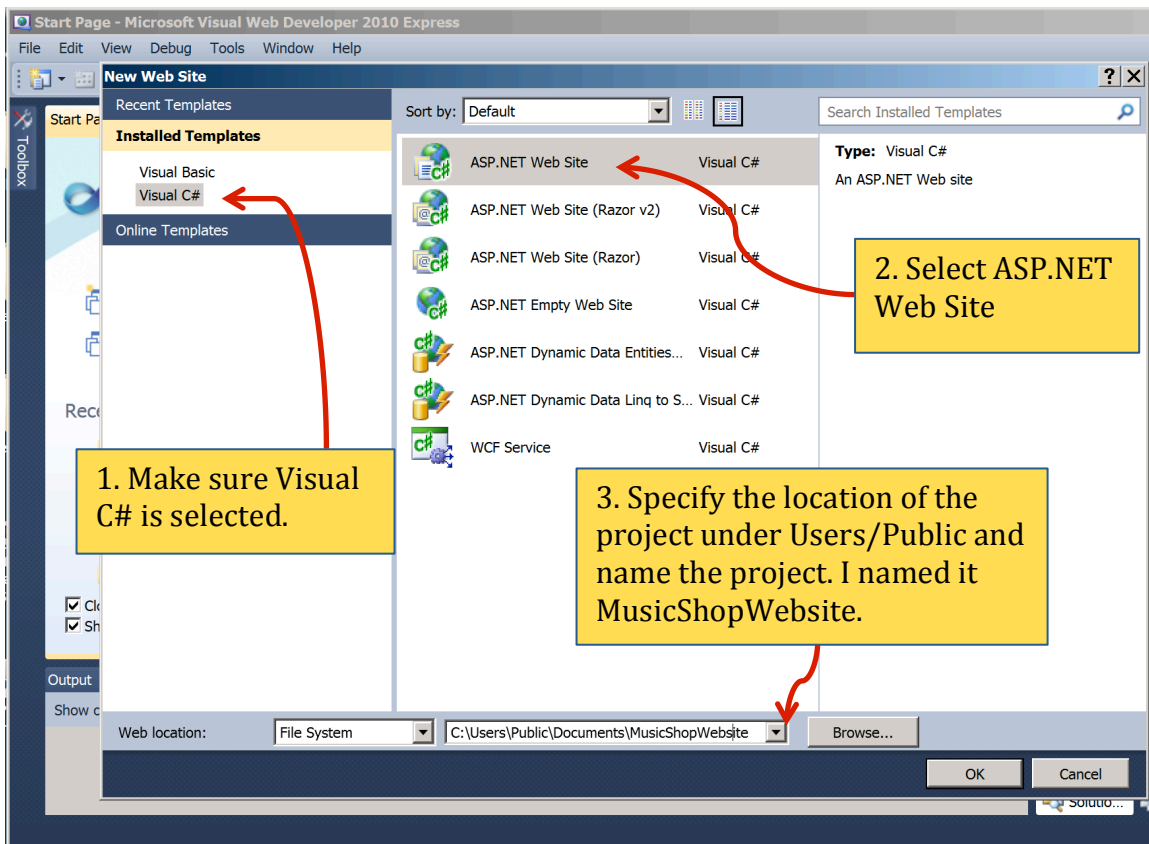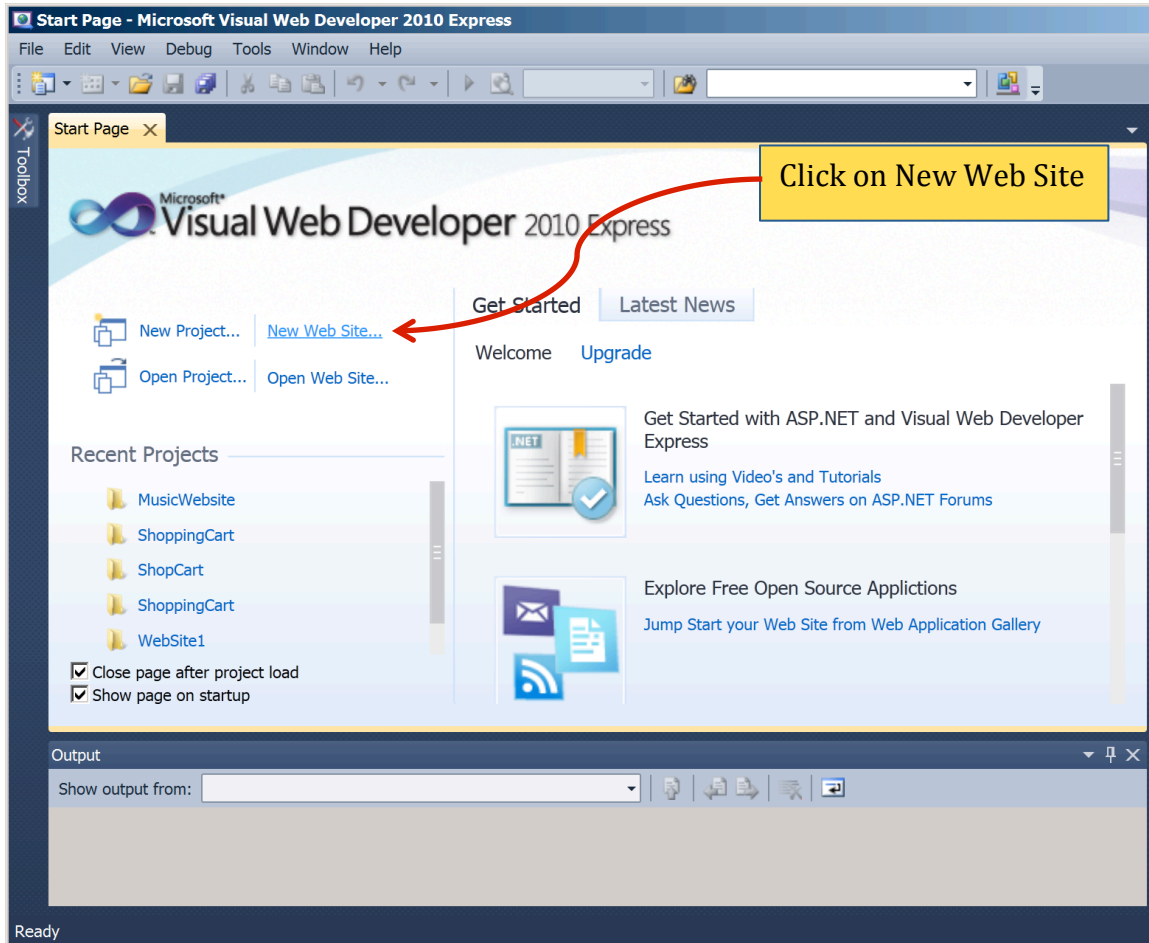
http://www.microsoft.com/express/Downloads/

Click on Download then scroll down to 2010 Express products and click on Visual Web Developer 2010 Express. Because Visual Web Developer is designed for developing ASP.NET websites, installing it automatically installs the .NET Framework and other required ASP.NET tools. You can also optionally install SQL Server 2008 (just checkmark "Microsoft SQL Server 2008 Express Service Pack 1" when prompted). If you forget to checkmark this option, you'll need to install Microsoft SQL Server 2008 Express separately.

Alternately, you may download and execute the Web Platform Installer 4.6, and checkmark the 4 products listed above to be installed.

Step 2. Getting started with Visual Web Developer

Screen 1 — Start Page:

Start Page - Microsoft Visual Web Developer 2010 Express

File Edit View Debug Tools Window Help

Start Page

Microsoft® **Visual Web Developer** 2010 Express

Click on New Web Site

New Project...   New Web Site...
Open Project...   Open Web Site...

Get Started | Latest News

Welcome Upgrade

Get Started with ASP.NET and Visual Web Developer Express
Learn using Video's and Tutorials
Ask Questions, Get Answers on ASP.NET Forums

Recent Projects
- MusicWebsite
- ShoppingCart
- ShopCart
- ShoppingCart
- WebSite1

Explore Free Open Source Appllctions
Jump Start your Web Site from Web Application Gallery

☑ Close page after project load
☑ Show page on startup

Output
Show output from:

Ready

Screen 2 — New Web Site dialog:

Start Page - Microsoft Visual Web Developer 2010 Express

File Edit View Debug Tools Window Help

New Web Site

Recent Templates

Installed Templates

Visual Basic
Visual C#

Online Templates

Sort by: Default

ASP.NET Web Site	Visual C#	
ASP.NET Web Site (Razor v2)	Visual C#	
ASP.NET Web Site (Razor)	Visual C#	
ASP.NET Empty Web Site	Visual C#	
ASP.NET Dynamic Data Entities...	Visual C#	
ASP.NET Dynamic Data Linq to S...	Visual C#	
WCF Service	Visual C#	

Search Installed Templates

Type: Visual C#
An ASP.NET Web site

2. Select ASP.NET Web Site

1. Make sure Visual C# is selected.

3. Specify the location of the project under Users/Public and name the project. I named it MusicShopWebsite.

Web location: File System   C:\Users\Public\Documents\MusicShopWebsite   Browse...

OK   Cancel

Pay attention to the directory where you store the project. The project should be in **C:\Users\Public\...** here I have named the project folder MusicShopWebsite. You might want to name it WebsiteExample, or any other name you like.

Visual web developer creates a skeleton aspx file named **Default.aspx**, which looks like the one below. This file contains only static code, but we will shortly see that we can *Web controls* to this page to support dynamic content.
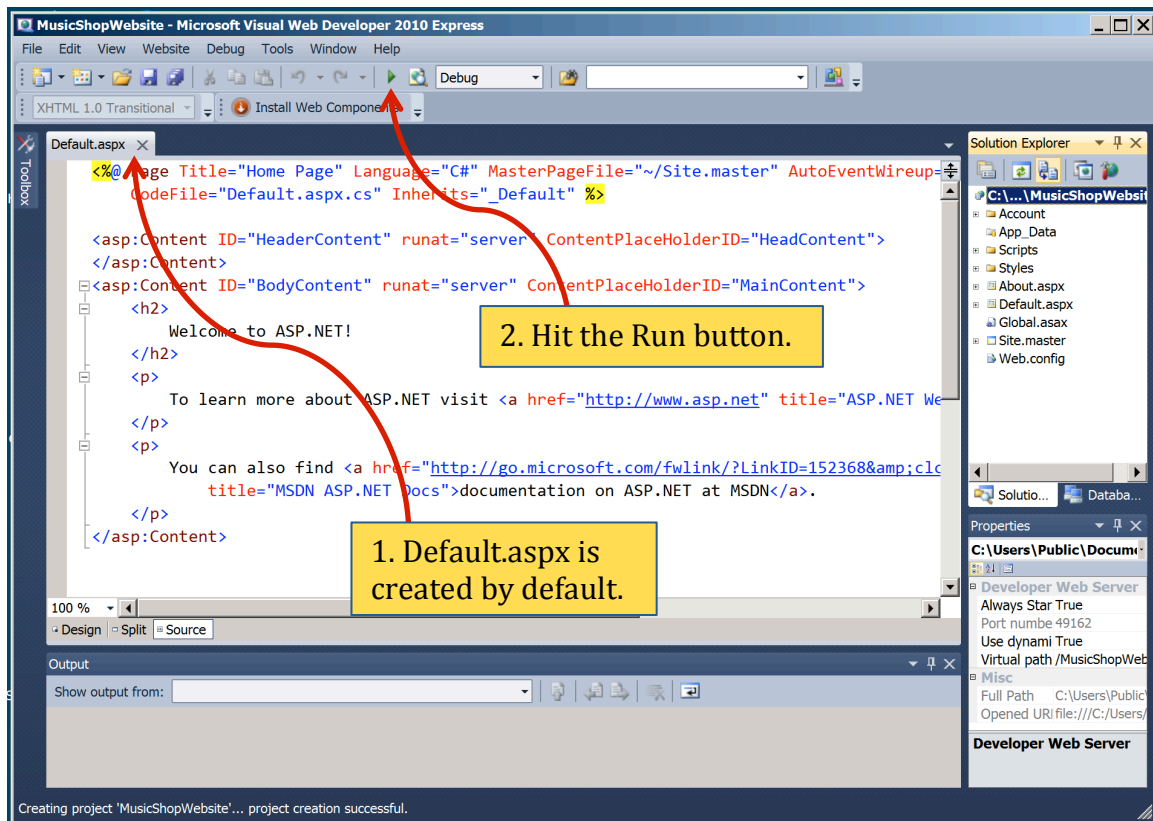

With Default.aspx open and your project folder in the Solution Explorer highlighted, hit the green triangular button at the top toolbar or F5. You may get a window like the one below. Go ahead and click OK.
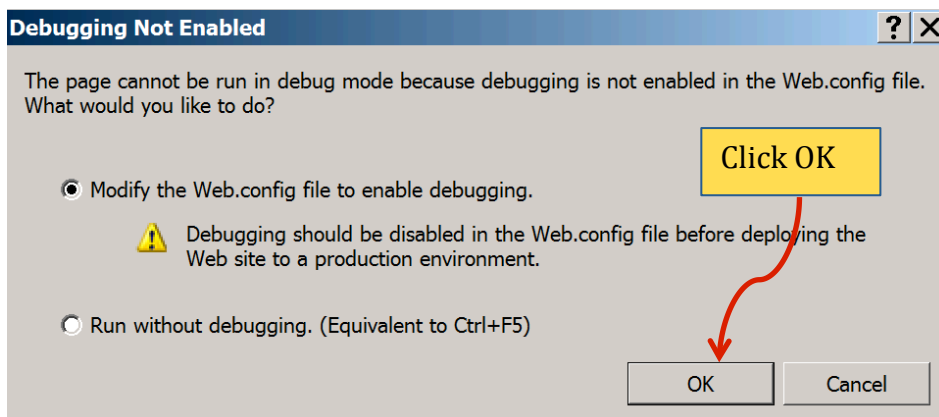
The default system browser opens and displays the application running.
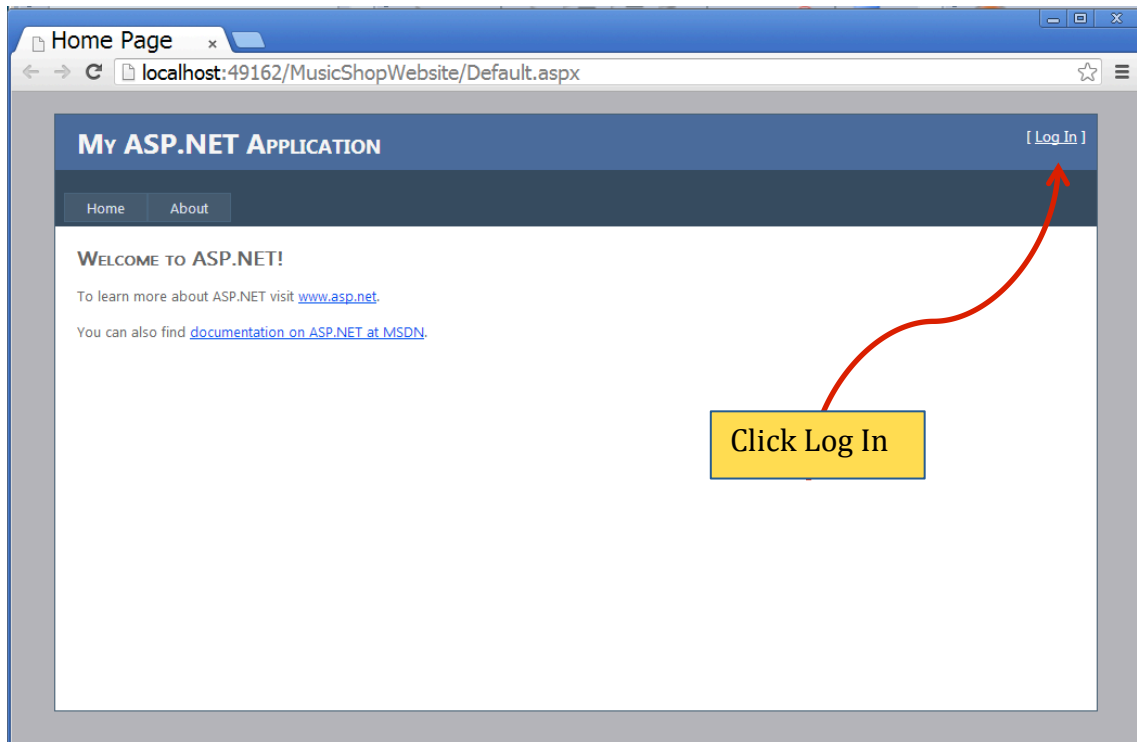
Many web applications require users to create an account before they can use certain features of the site. Visual Web Developer automatically includes Login and Register controls. Try logging in or create a new user.
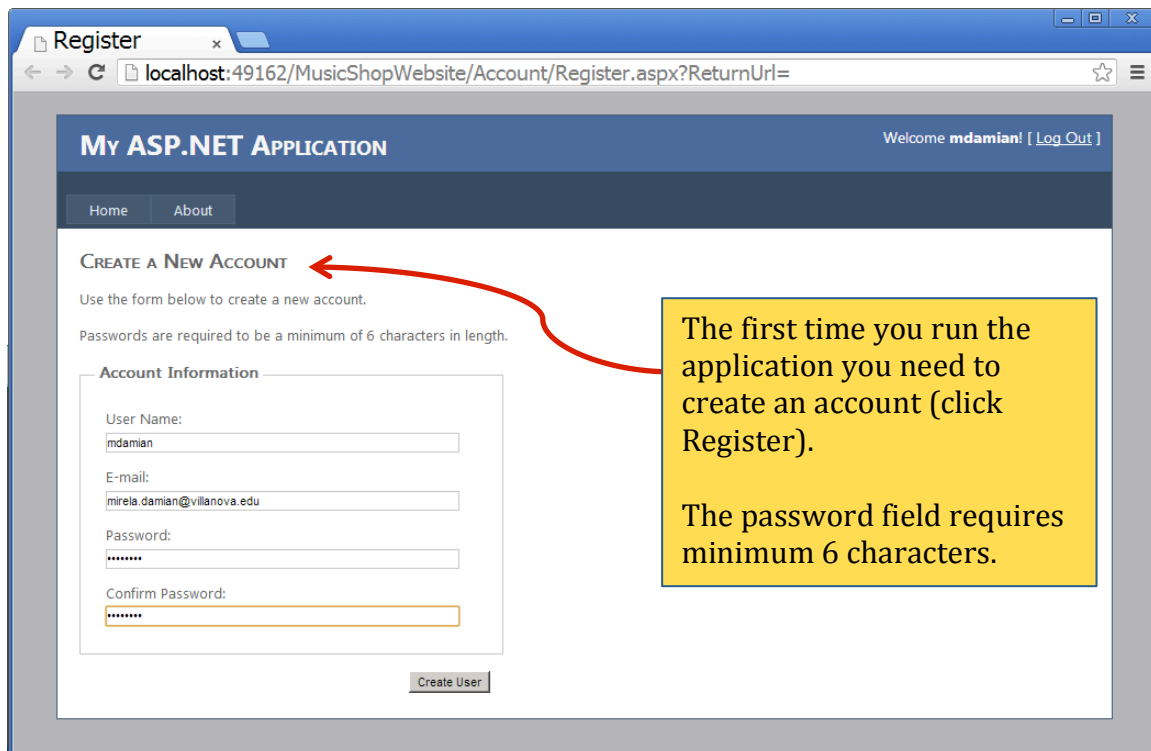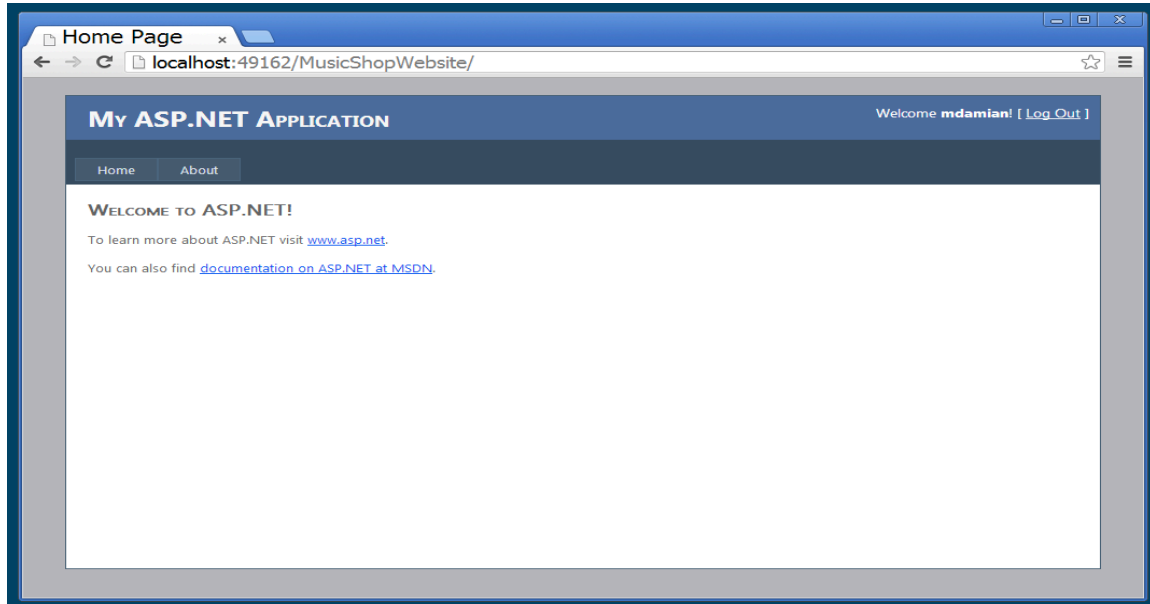
Close your browser and return to Visual Web Developer. For the first user registered with the website, Visual Web developer automatically creates a SQL Server 2008 Express Edition database named **ASPNETDB** in the **App_Data** folder with predesigned tables to support user accounts.
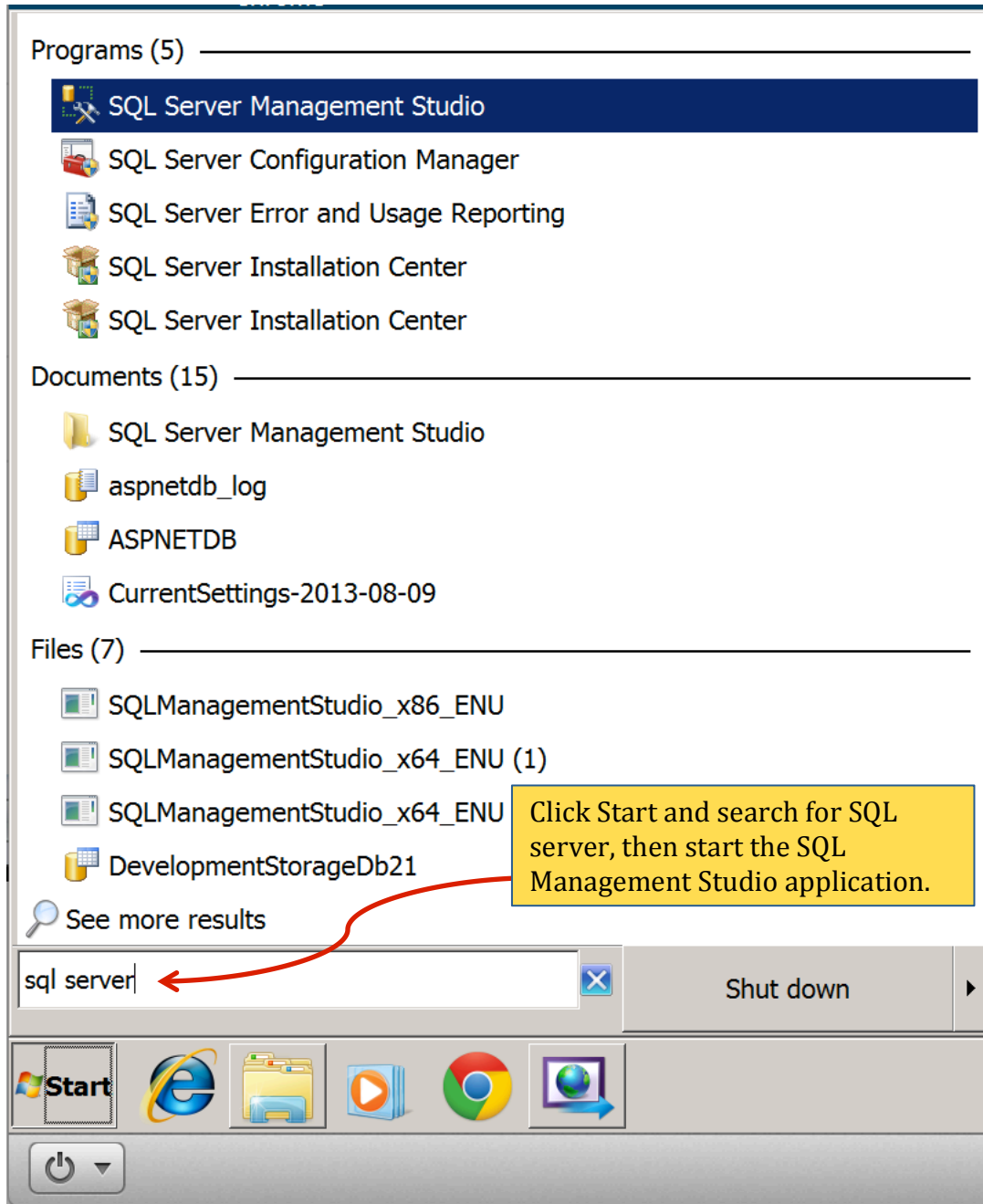
To verify this, right-click on the App_Data folder in the Solution Explorer window and select **Refresh Folder**. Expand App_Data by clicking on the plus sign. You will see the database file ASPNETDB.MDF used by your website. By default it includes 11 tables, providing functionality that extends beyond simple user accounts.
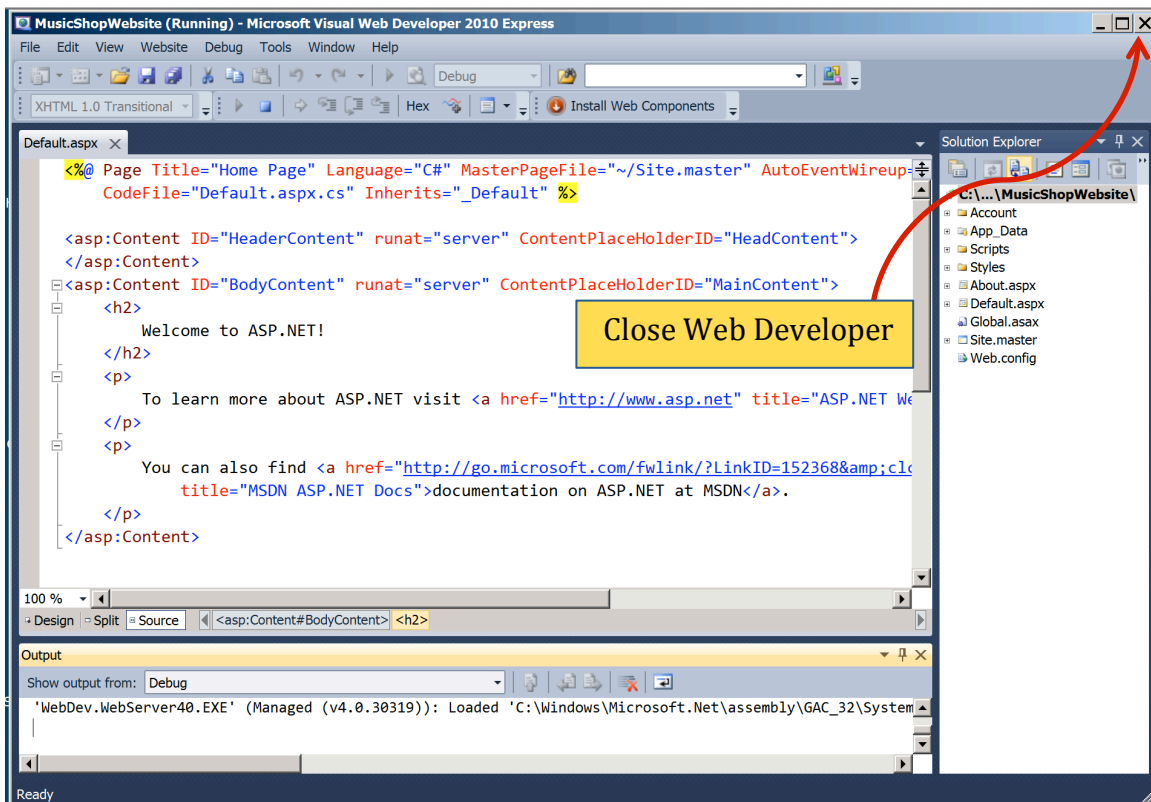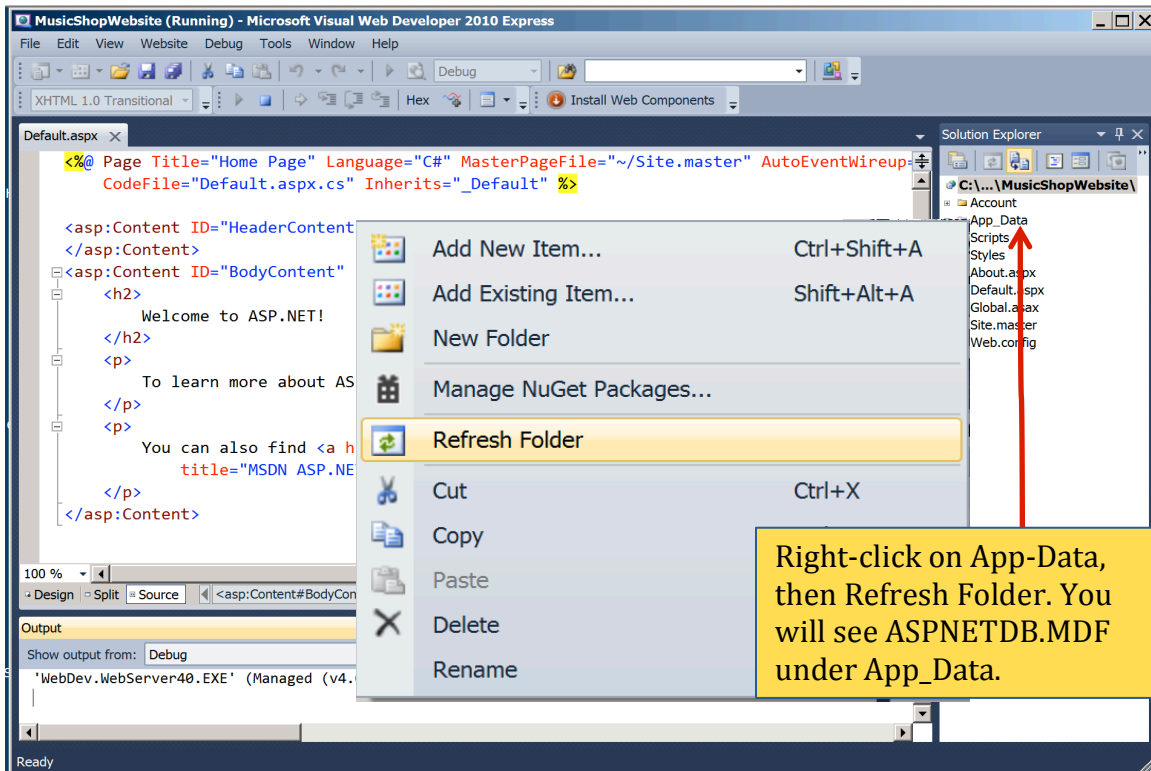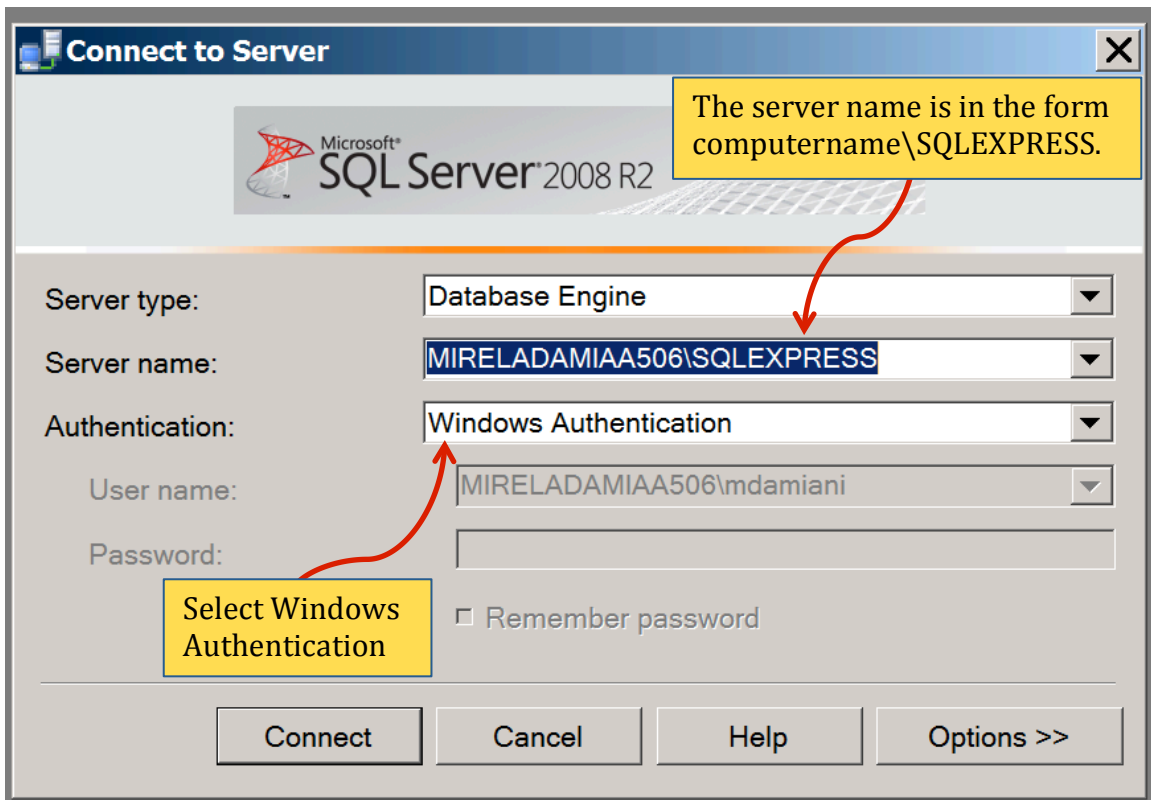
Now that you are a new user of this application you are ready to create the tables in the database. Save all changes and make sure you close Visual Web Developer.
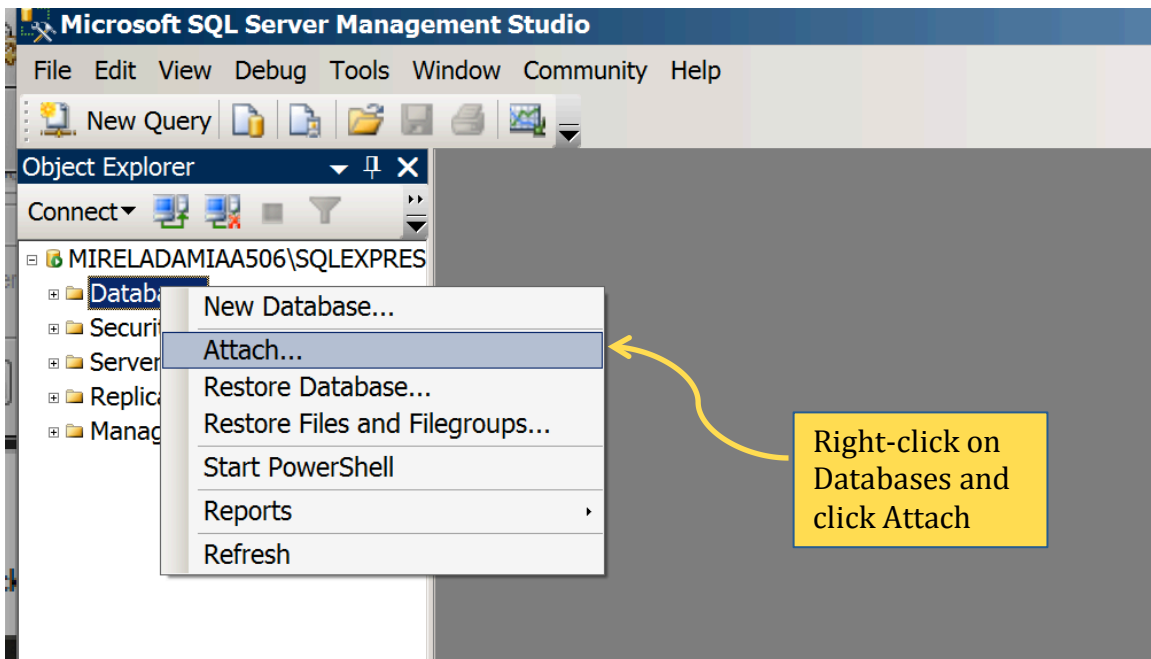
Step 3. Adding Tables to Your Database

After closing the Visual Web Developer, open SQL Server Management Studio.
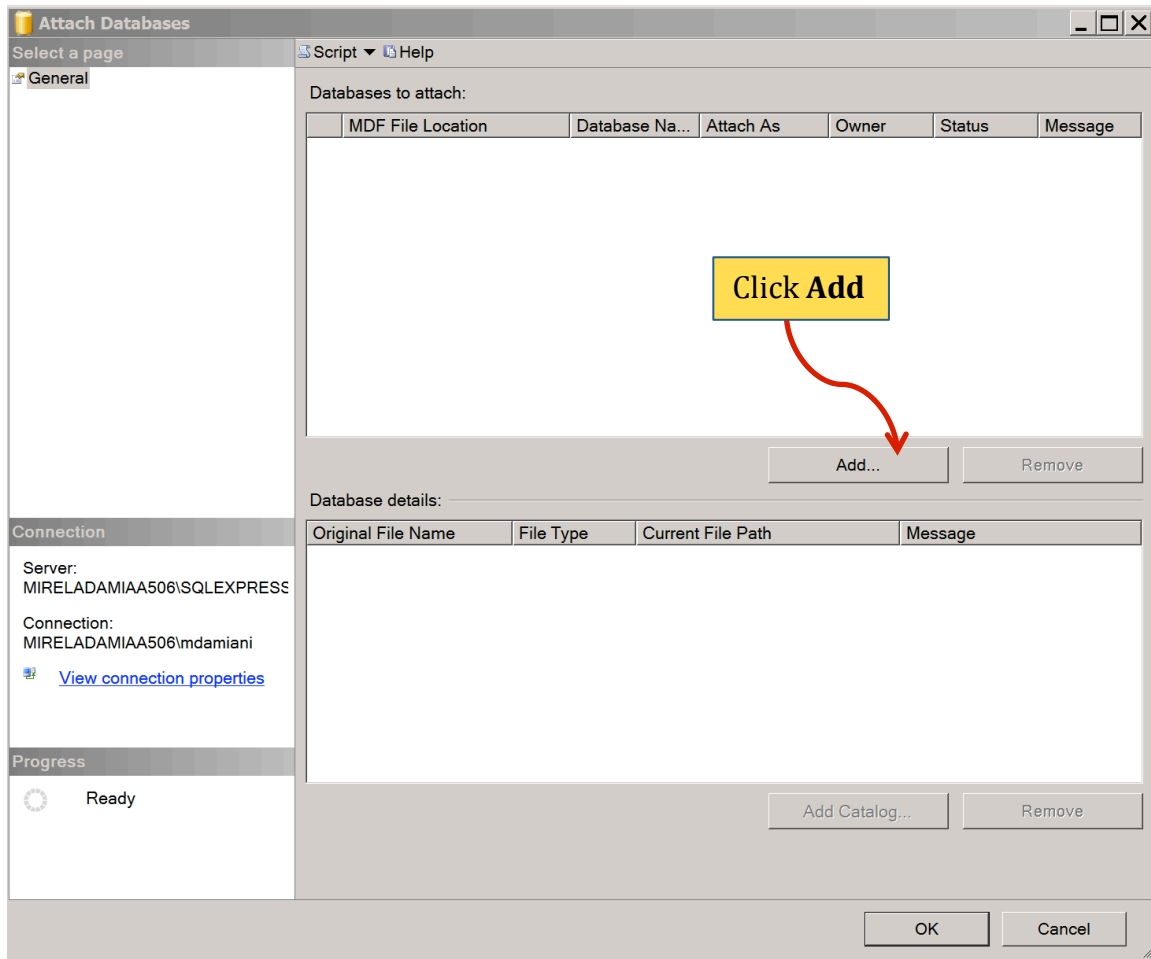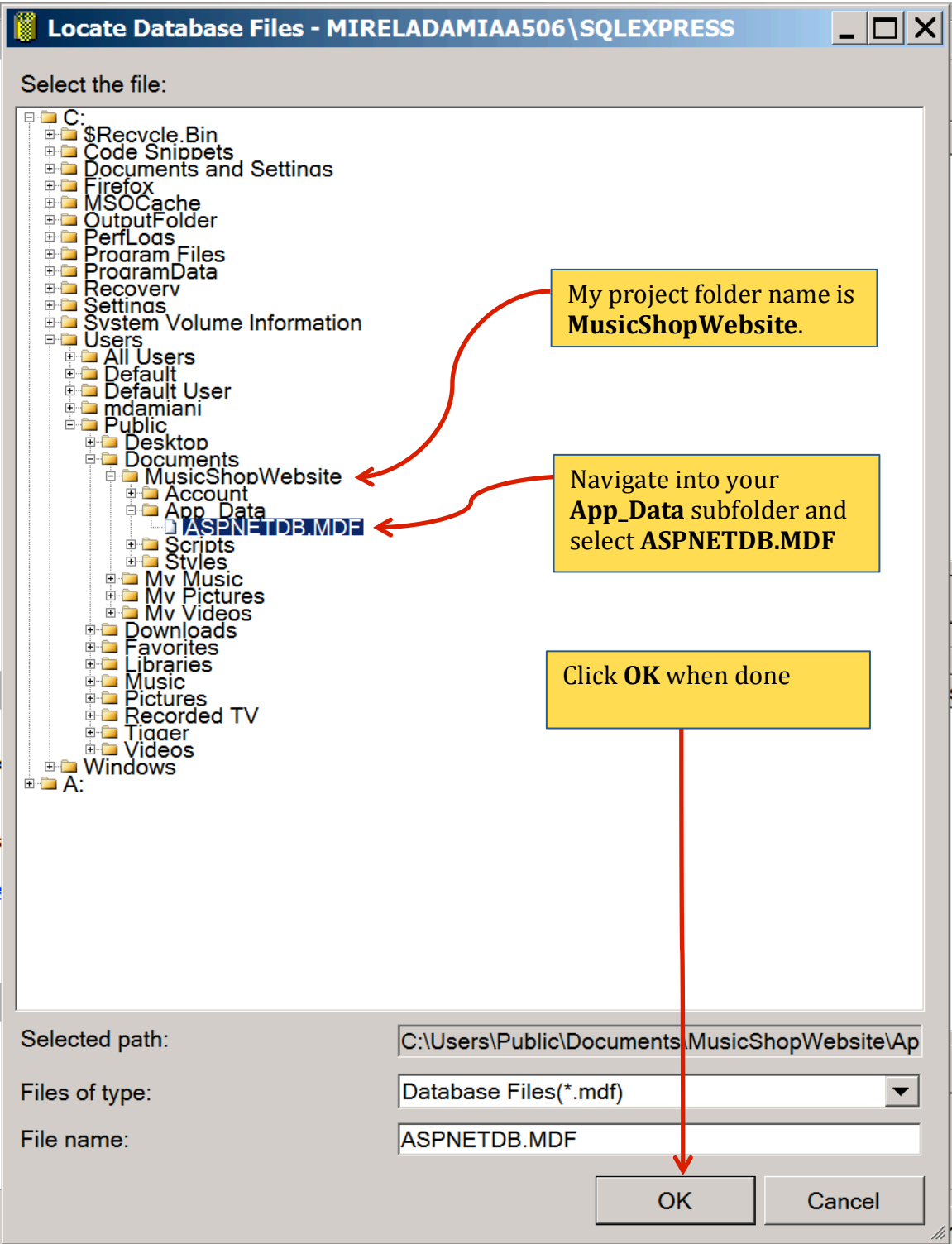

Programs (5)

- SQL Server Management Studio
- SQL Server Configuration Manager
- SQL Server Error and Usage Reporting
- SQL Server Installation Center
- SQL Server Installation Center

Documents (15)

- SQL Server Management Studio
- aspnetdb_log
- ASPNETDB
- CurrentSettings-2013-08-09

Files (7)

- SQLManagementStudio_x86_ENU
- SQLManagementStudio_x64_ENU (1)
- SQLManagementStudio_x64_ENU
- DevelopmentStorageDb21

See more results

sql server   Shut down

Start

Click Start and search for SQL server, then start the SQL Management Studio application.

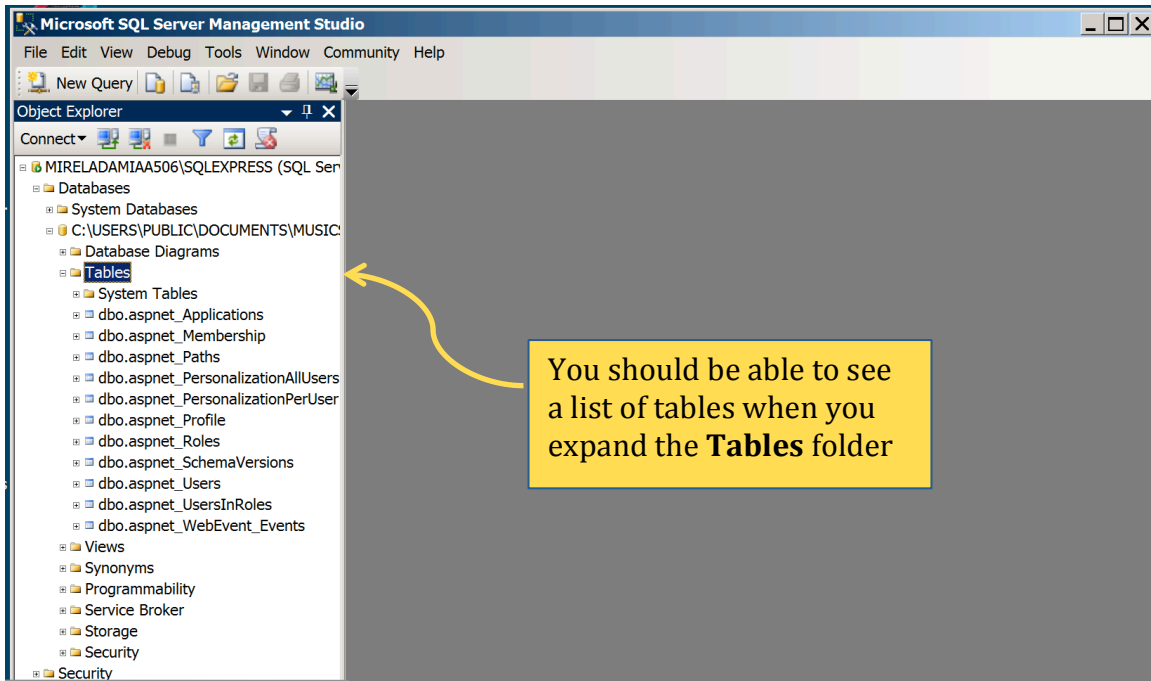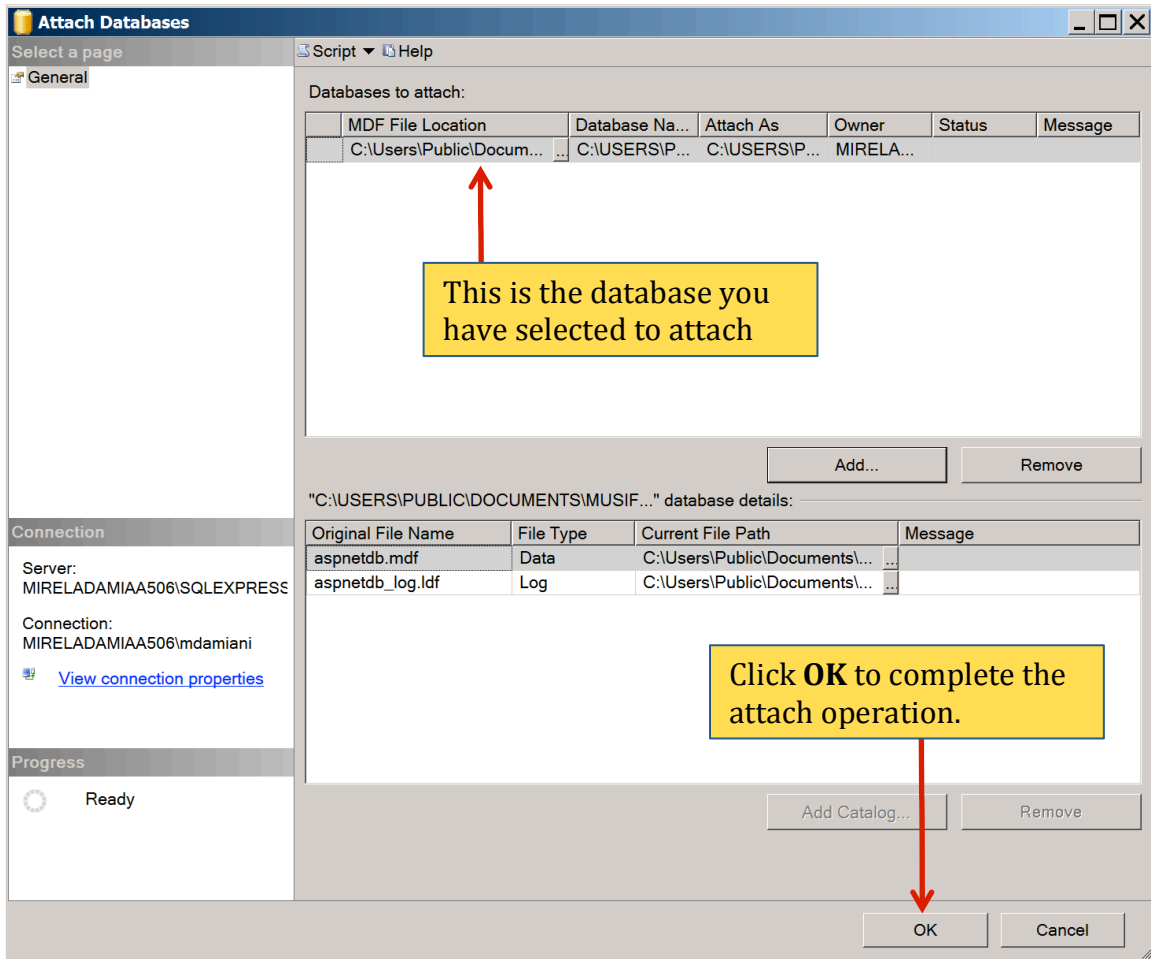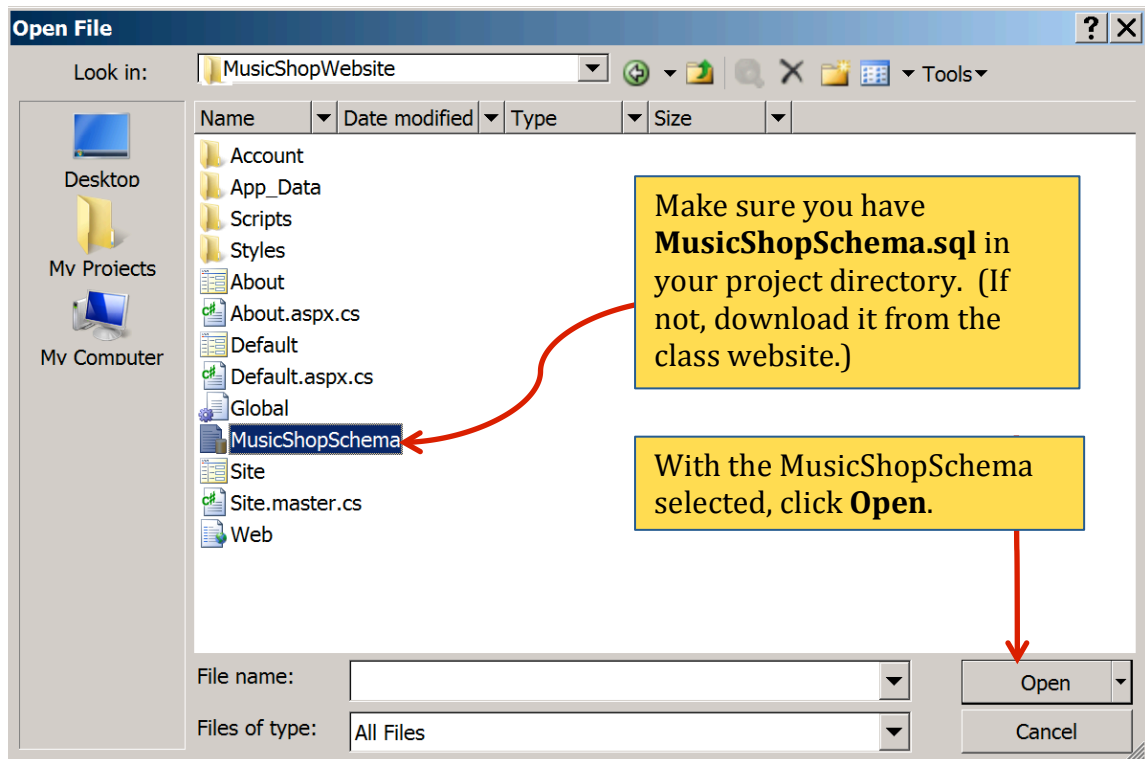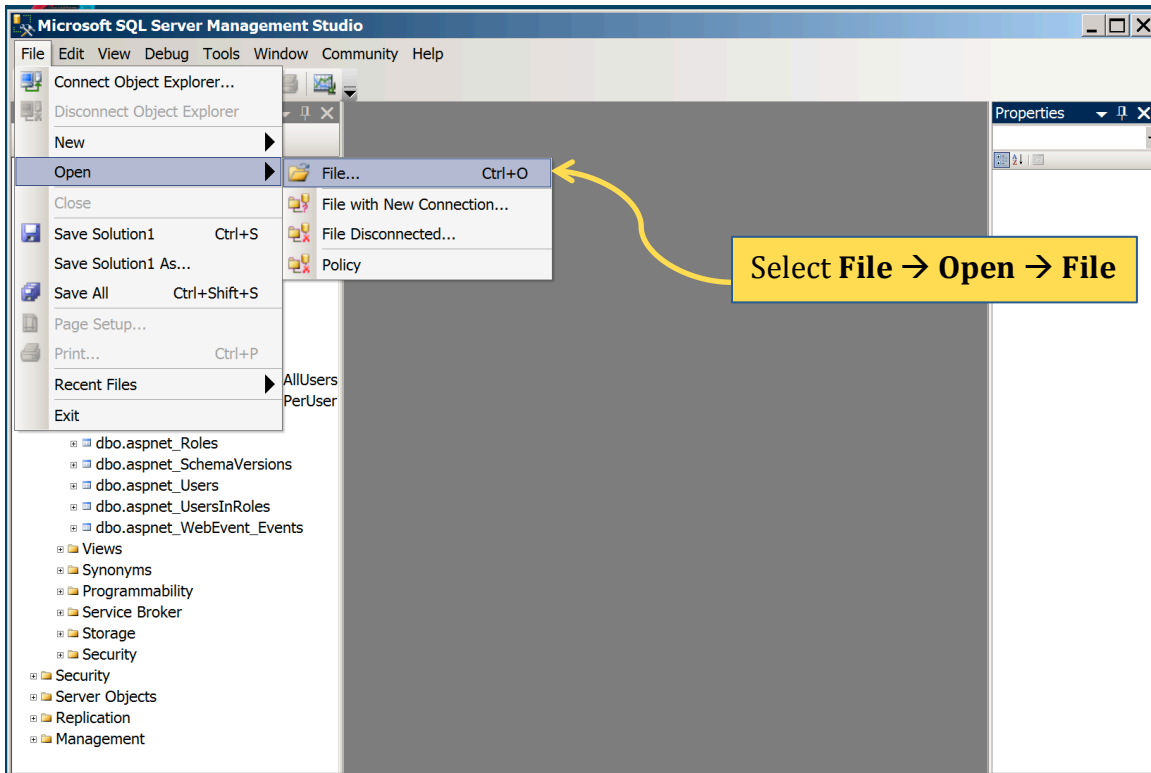Once you are connected right click on **Databases** and click **Attach**.

After clicking **Attach**, you will be presented with the following window:
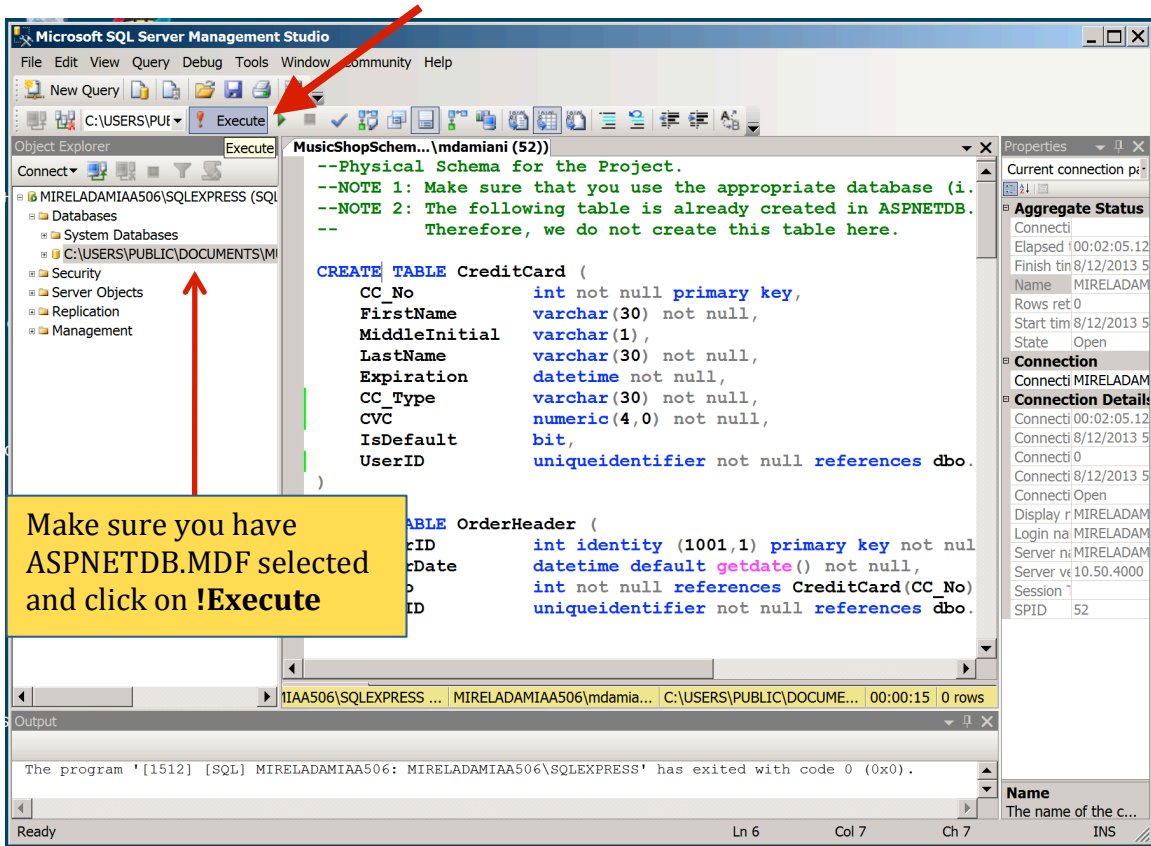
Navigate to your project folder under C:\Users\Public... and select ASPNETDB.MDF. (This file was created when you added yourself as a new user of the Visual Web Development Application).
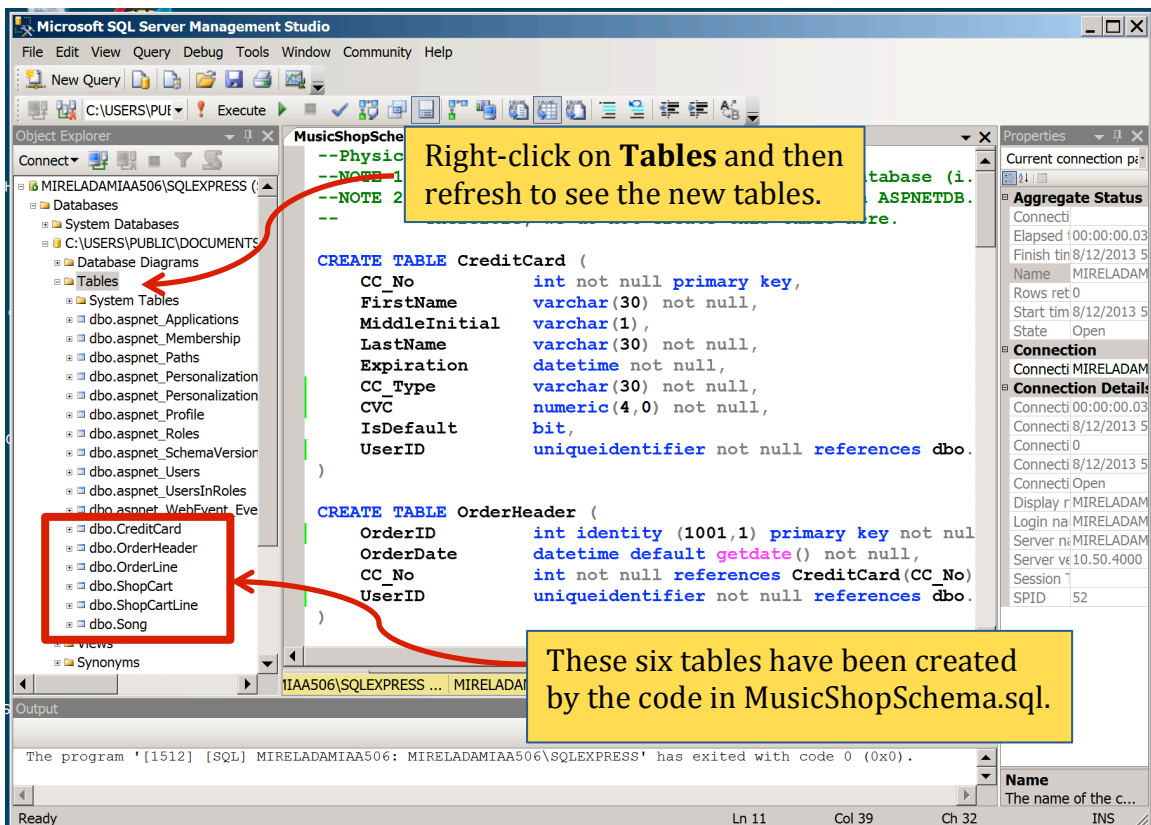
Locate Database Files - MIRELADAMIAA506\SQLEXPRESS

Select the file:

```
C:
  $Recycle.Bin
  Code Snippets
  Documents and Settings
  Firefox
  MSOCache
  OutputFolder
  PerfLogs
  Program Files
  ProgramData
  Recovery
  Settings
  System Volume Information
  Users
    All Users
    Default
    Default User
    mdamiani
    Public
      Desktop
      Documents
        MusicShopWebsite
          Account
          App_Data
            ASPNETDB.MDF
          Scripts
          Styles
        My Music
        My Pictures
        My Videos
      Downloads
      Favorites
      Libraries
      Music
      Pictures
      Recorded TV
      Tigger
      Videos
  Windows
A:
```

My project folder name is **MusicShopWebsite**.

Navigate into your **App_Data** subfolder and select **ASPNETDB.MDF**

Click **OK** when done

Selected path: C:\Users\Public\Documents\MusicShopWebsite\Ap

Files of type: Database Files(*.mdf)

File name: ASPNETDB.MDF

OK   Cancel

Attach Databases

Script ▼ Help

Databases to attach:

	MDF File Location	Database Na...	Attach As	Owner	Status	Message
	C:\Users\Public\Docum...	...	C:\USERS\P...	C:\USERS\P...	MIRELA...	

This is the database you have selected to attach

Add...   Remove

"C:\USERS\PUBLIC\DOCUMENTS\MUSIF..." database details:

Original File Name	File Type	Current File Path	Message
aspnetdb.mdf	Data	C:\Users\Public\Documents\...	
aspnetdb_log.ldf	Log	C:\Users\Public\Documents\...	

Connection

Server:
MIRELADAMIAA506\SQLEXPRESS

Connection:
MIRELADAMIAA506\mdamiani

View connection properties

Progress

Ready

Click **OK** to complete the attach operation.

Add Catalog...   Remove

OK   Cancel

Microsoft SQL Server Management Studio

File Edit View Debug Tools Window Community Help

New Query

Object Explorer

Connect

- MIRELADAMIAA506\SQLEXPRESS (SQL Ser
  - Databases
    - System Databases
    - C:\USERS\PUBLIC\DOCUMENTS\MUSIC
      - Database Diagrams
      - Tables
        - System Tables
        - dbo.aspnet_Applications
        - dbo.aspnet_Membership
        - dbo.aspnet_Paths
        - dbo.aspnet_PersonalizationAllUsers
        - dbo.aspnet_PersonalizationPerUser
        - dbo.aspnet_Profile
        - dbo.aspnet_Roles
        - dbo.aspnet_SchemaVersions
        - dbo.aspnet_Users
        - dbo.aspnet_UsersInRoles
        - dbo.aspnet_WebEvent_Events
      - Views
      - Synonyms
      - Programmability
      - Service Broker
      - Storage
      - Security
  - Security

You should be able to see a list of tables when you expand the **Tables** folder

Select **File → Open → File**

Make sure you have **MusicShopSchema.sql** in your project directory. (If not, download it from the class website.)

With the MusicShopSchema selected, click **Open**.

The figure above contains the annotation: "Make sure you have ASPNETDB.MDF selected and click on **!Execute**"

The SQL code from MusicShopSchema.sql adds 6 tables to the database: Song, ShopCart, ShopCartLine, OrderHeader, OrderLine and CreditCard. It also adds some sample data to the Song table. After the SQL code gets executed, you should be able to see the new tables in the Object Explorer when you refresh.

The figure above contains the annotations: "Right-click on **Tables** and then refresh to see the new tables." and "These six tables have been created by the code in MusicShopSchema.sql."

Now that we have created the tables we needed to work with, the job of the SQL
server is done. We need to detach the database, so that it can be used by other
applications (our Web Development server in particular).

Right-click on **Tables**, then select Tasks → Detach
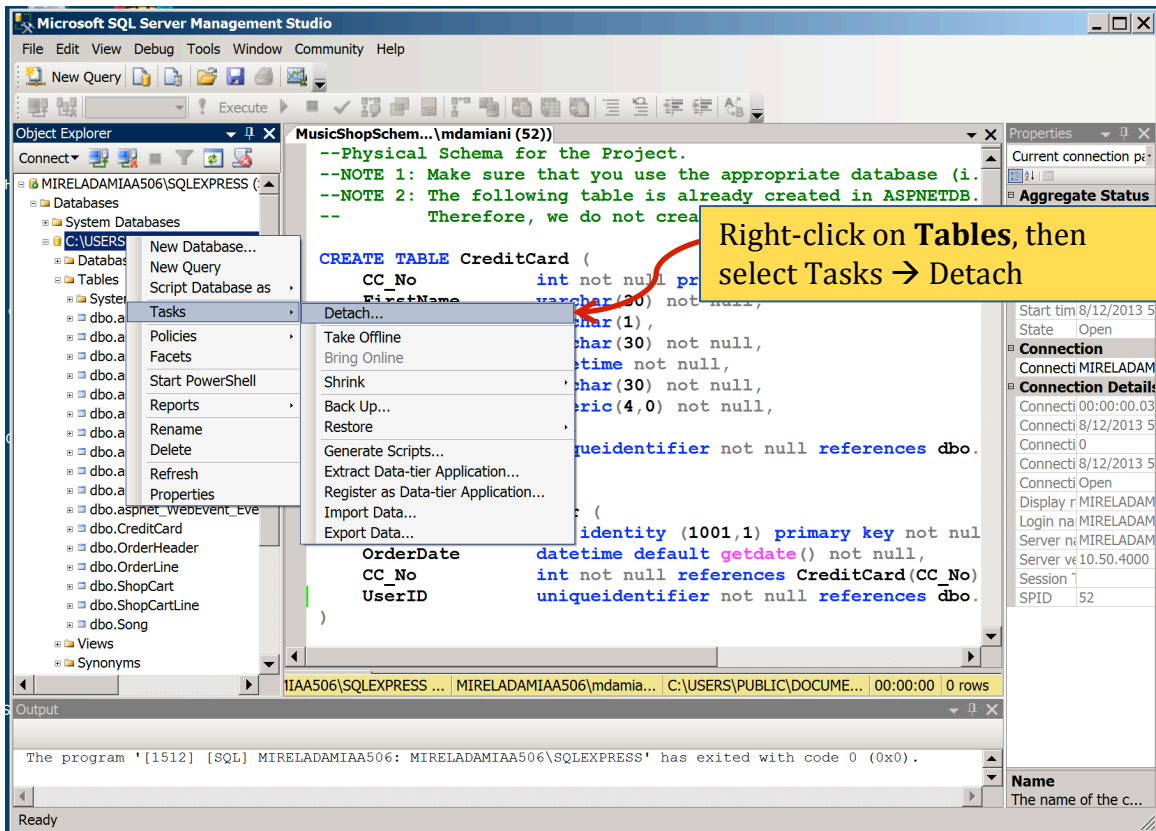
You will be prompted with a window for selecting the database to detach.
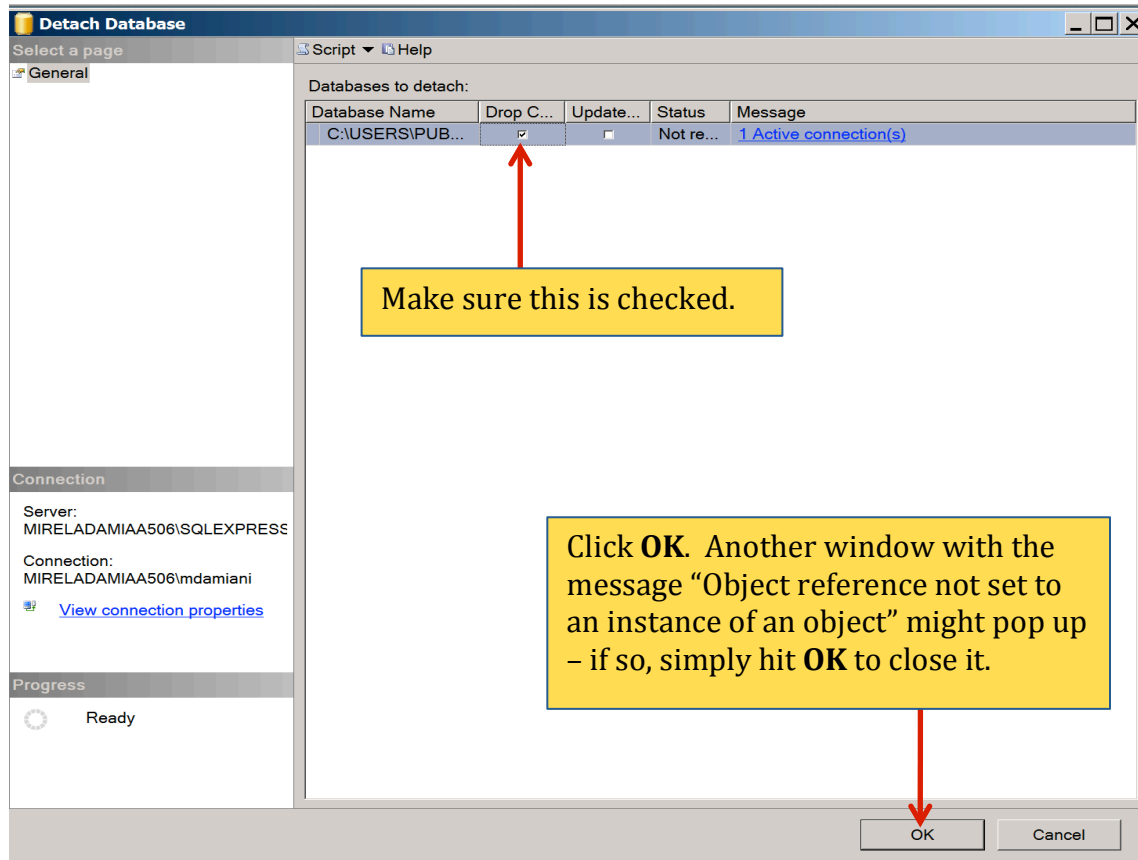
Close SQL Server Management Studio.
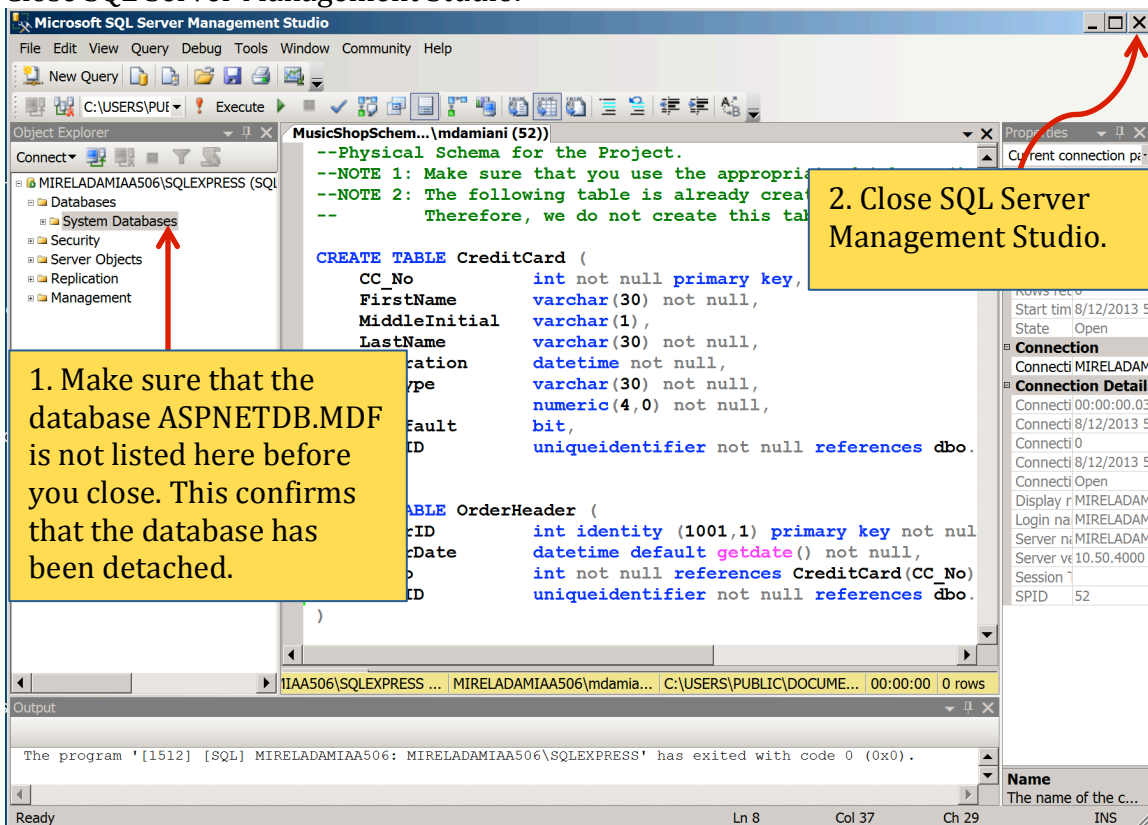
Step 4. Retrieving Data from the Database

Now that we have created our database, our next step is accessing data from the database. Among many other Web controls, ASP.NET contains an SqlDataSource control designed to retrieve data from a database, and a GridView control designed to display the retrieved data on a page.

Open MS Visual Web Developer. Your website may be listed under Recent Projects, so you can click on it directly. If not listed, then select File → Open and navigate to your website in your Users/Public folder.

Double-click to open your website.

The Source view shows the asp code behind the website design. Alternate views are Design only and Split (both).
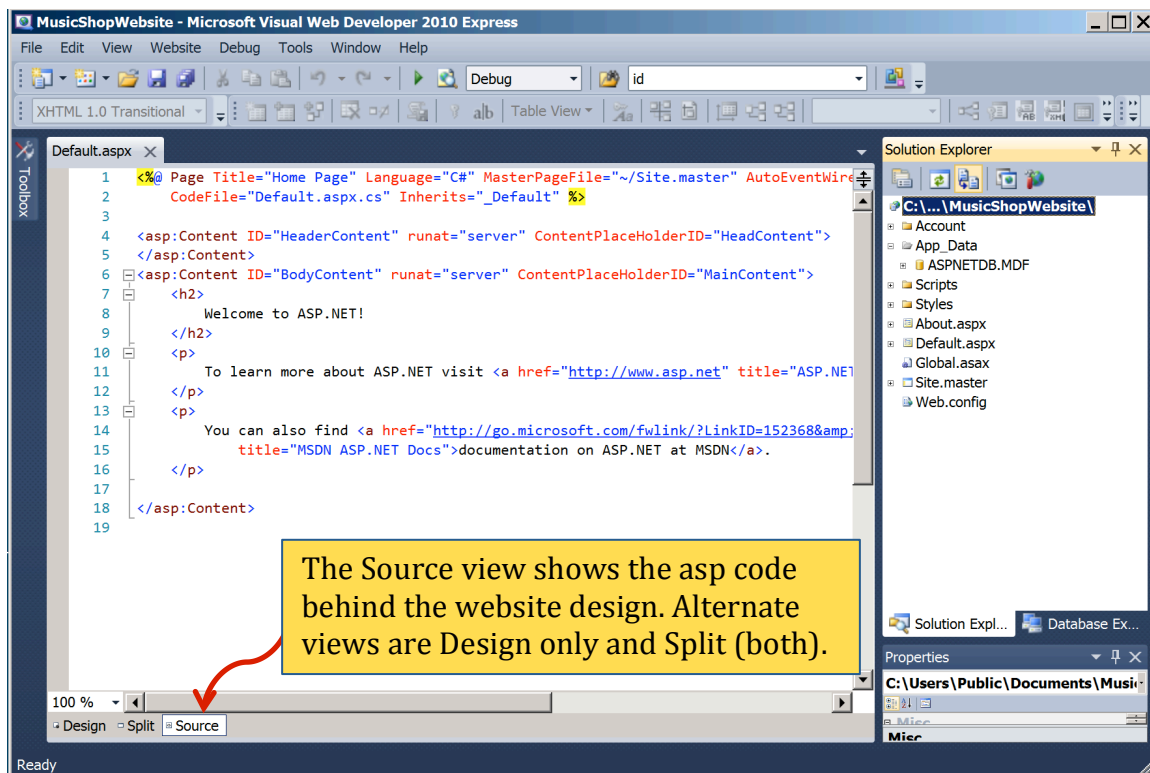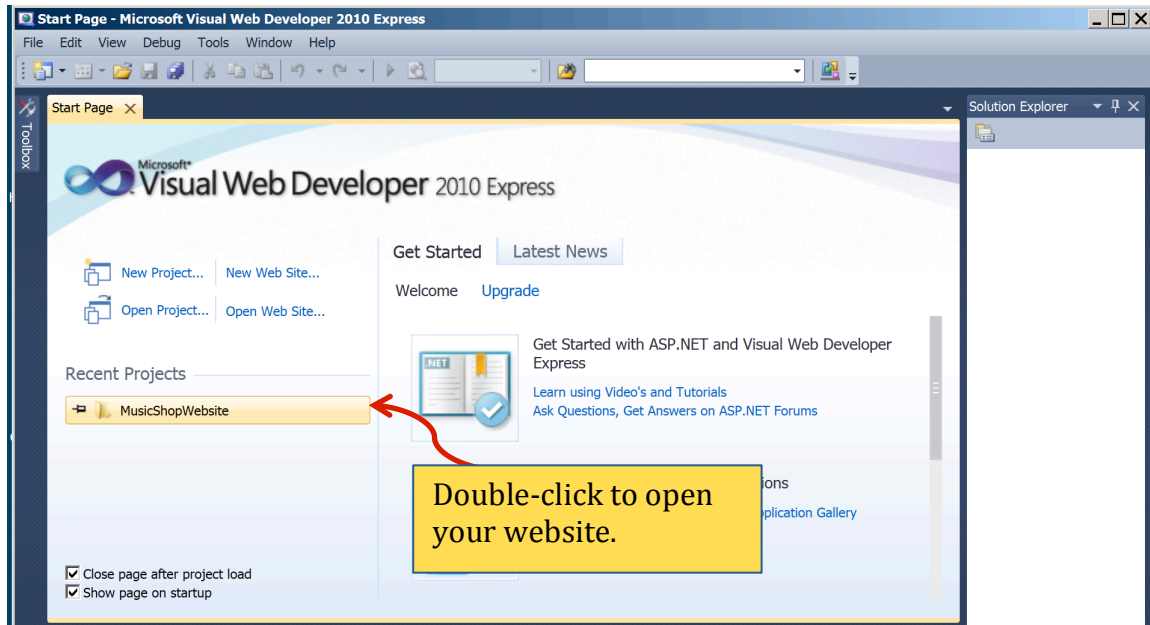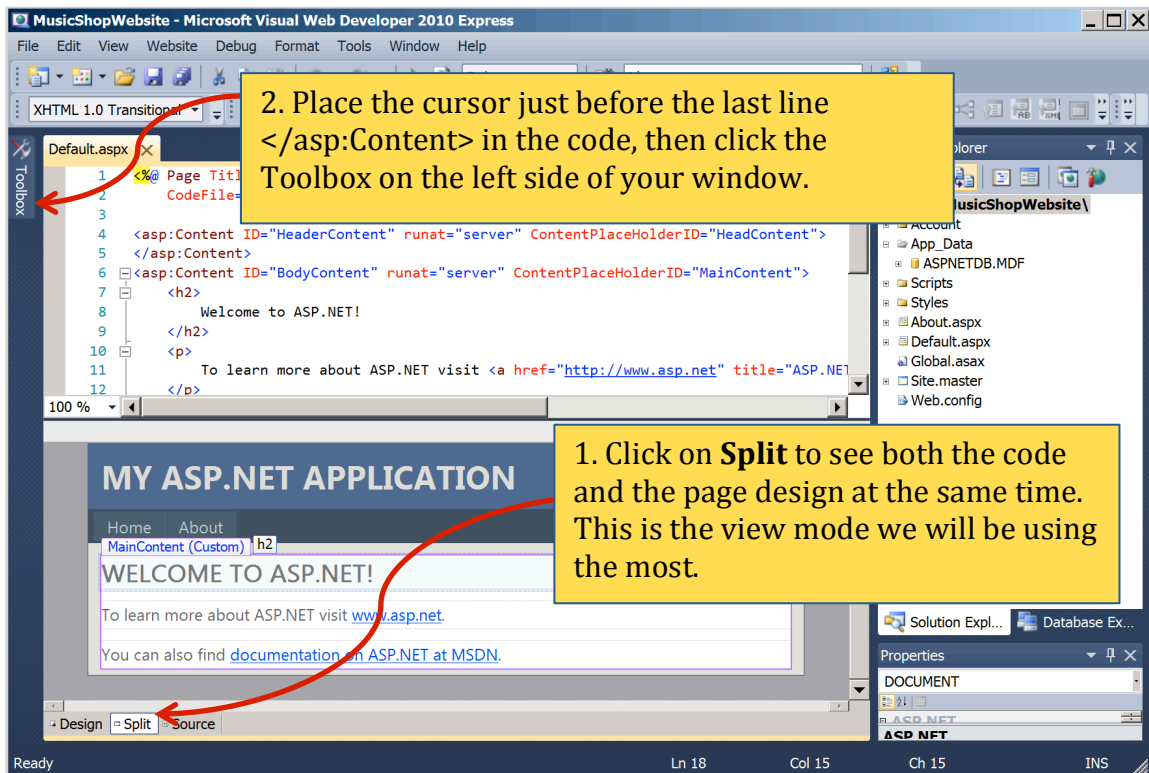
If you don't find the Toolbox, go to View → Other Windows → Toolbox, or Windows→ Reset Window Layout. From the drop-down menu of the Toolbox, double click SqlDataSource (you'll find it under the Data section of the Toolbox).
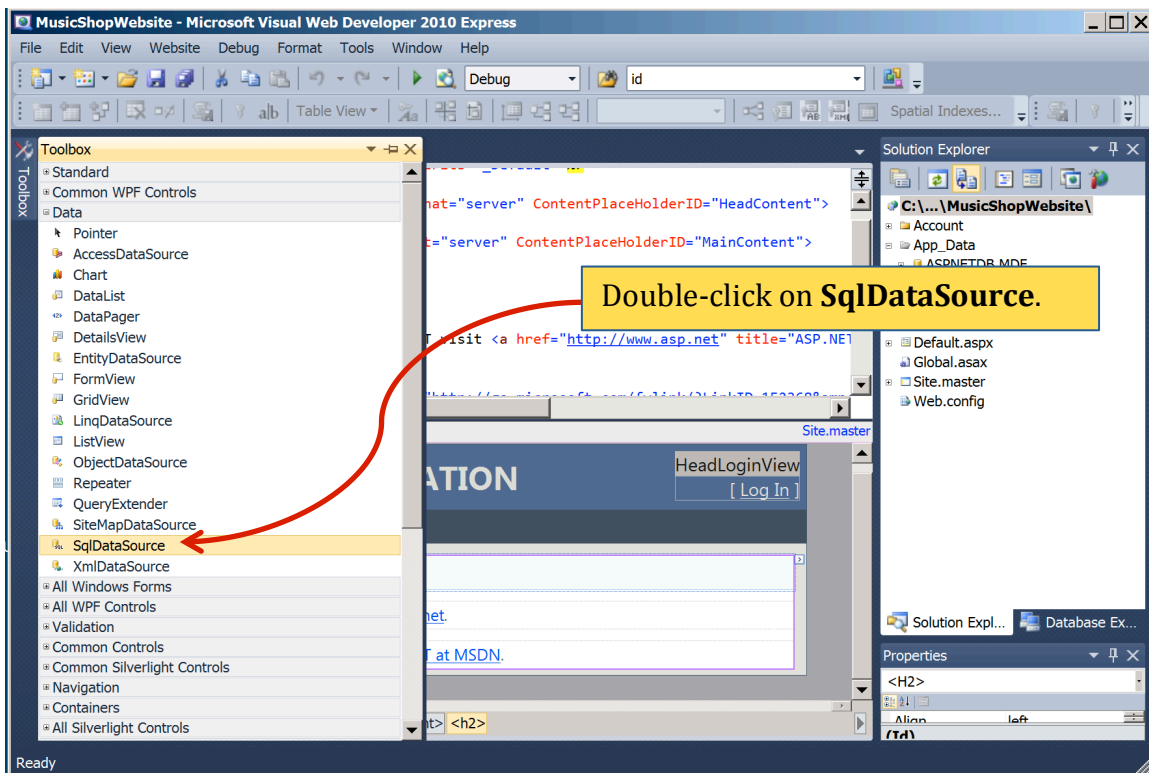
Once you have added the **SqlDataSource** you should be able to see the code generated in the Source view.
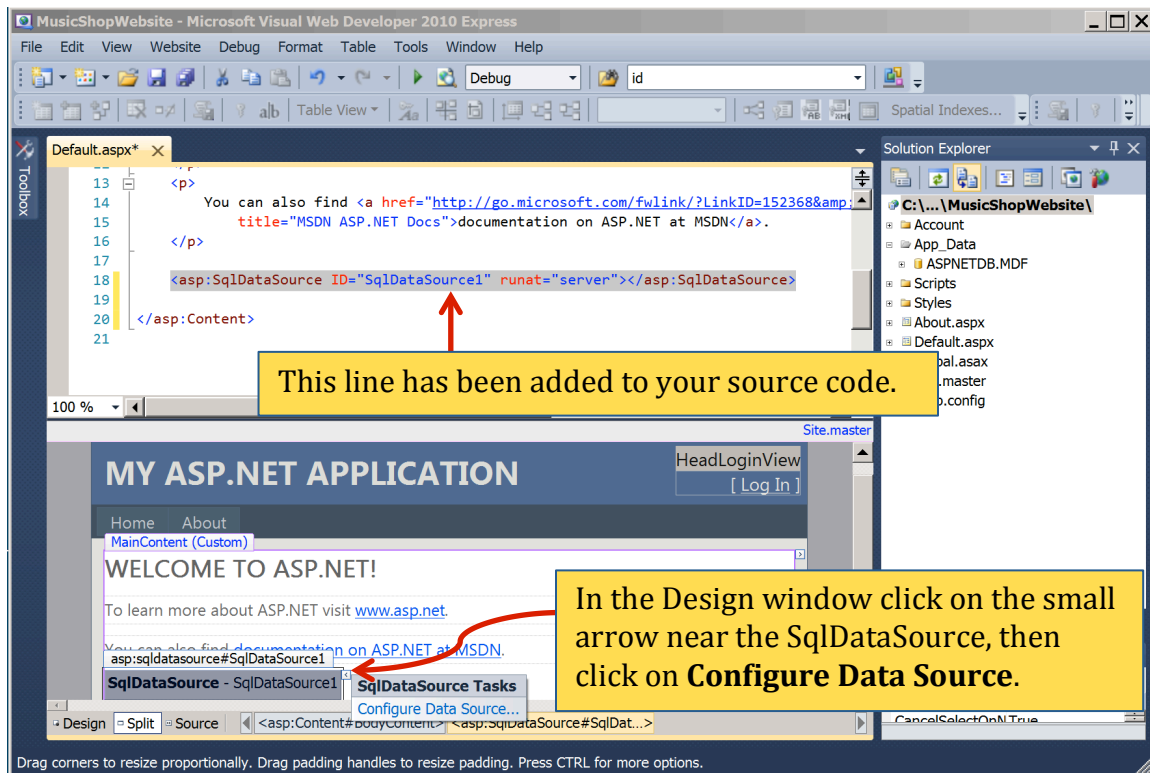
To be able to retrieve data, the SqlDataSource control needs to know two bits of information: (1) How to connect to the database, and (2) What data to retrieve from the database. The control wizard will next prompt you to provide this information.
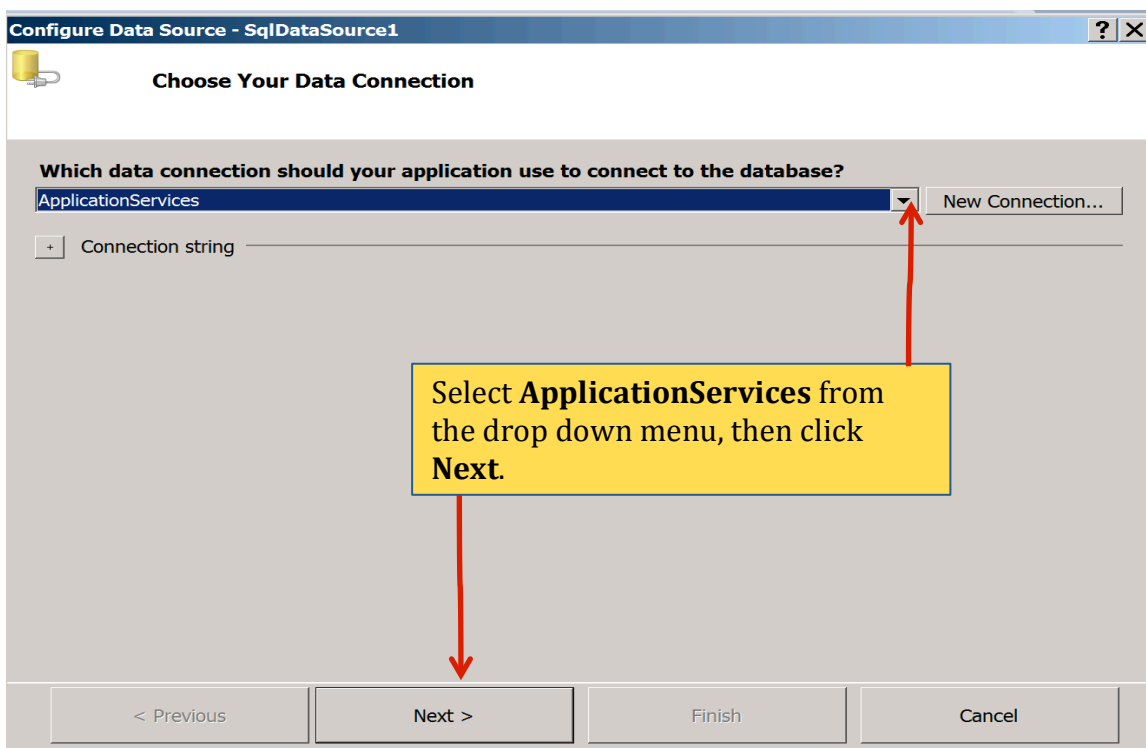
Configure Data Source - SqlDataSource1   ? X

Configure the Select Statement

How would you like to retrieve data from your database?

○ Specify a custom SQL statement or stored procedure
⦿ Specify columns from a table or view

Name:

| Song | ▼ |

1. Select **Song** from the drop down menu.

Columns:

☑ *
☐ SongID
☐ Title
☐ Album
☐ Artist
☐ Genre
☐ ReleaseDate
☐ ListPrice
☐ ClipName

2. Make sure * (which means everything) is checked, then click **Next**.

☐ Return only unique rows

| WHERE... |
| ORDER BY... |
| Advanced... |

SELECT statement:

SELECT * FROM [Song]

| < Previous | Next > | Finish | Cancel |

Configure Data Source - SqlDataSource1   ? X

Test Query

To preview the data returned by this data source, click Test Query. To complete this wizard, click Finish.

SongID	Title	Album	Artist	Genre	ReleaseDate	ListPrice	ClipName
101	Disorganized Fun	Studio Album	Ronald Jenkees	Electronic	4/5/2009	0.9900	Fun.mp3
102	Harry Potter	Very Potter	StarKid	Showtunes	7/31/2010	0.9900	HarryPotter.mp3
103	Cannon in D	Ronald Jenkees	Ronald Jenkees	Classic	11/11/2007	0.9900	CannonD.mp3
104	Bad Day	Chipwrecked	Alvin & Chipmunks	Pop	11/15/2011	0.9900	Chipmunks
105	Beat Box	Album1	Joe Publique	Rock	11/11/2007	0.9900	BeatBox.mp3
106	Coyote Kisses	Acid Wolfpack	Joe and Bryce	Electronic	3/26/2012	0.9900	Coyote.mp3
107	Part of Me	Teenage Dream	Katie Perry	Pop-Rock	2/5/2012	0.9900	PartOfMe.mp3

1. You may test your query. Make sure that the above table is displayed upon clicking **Test Query**.

| Test Query |

2. Click **Finish**.

SELECT statement:

SELECT * FROM [Song]

| < Previous | Next > | Finish | Cancel |

You should be able to see the table retrieved and the Select Statement from the database when you hit Test Query.
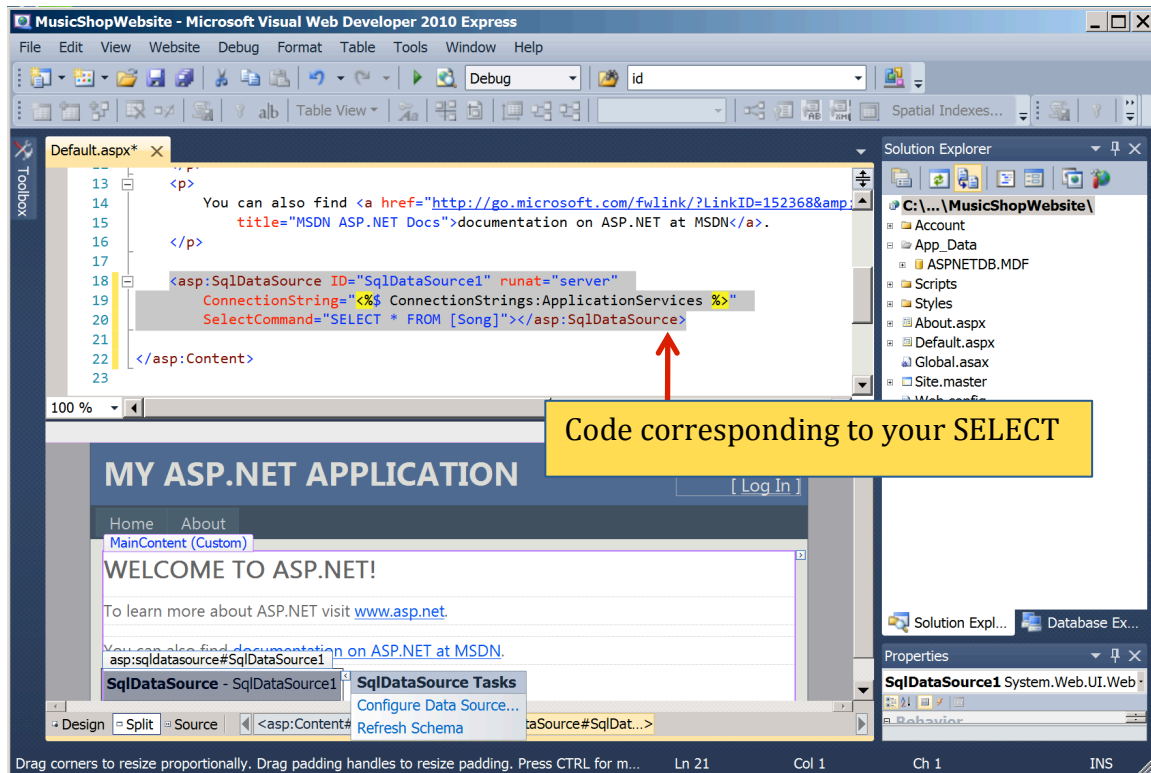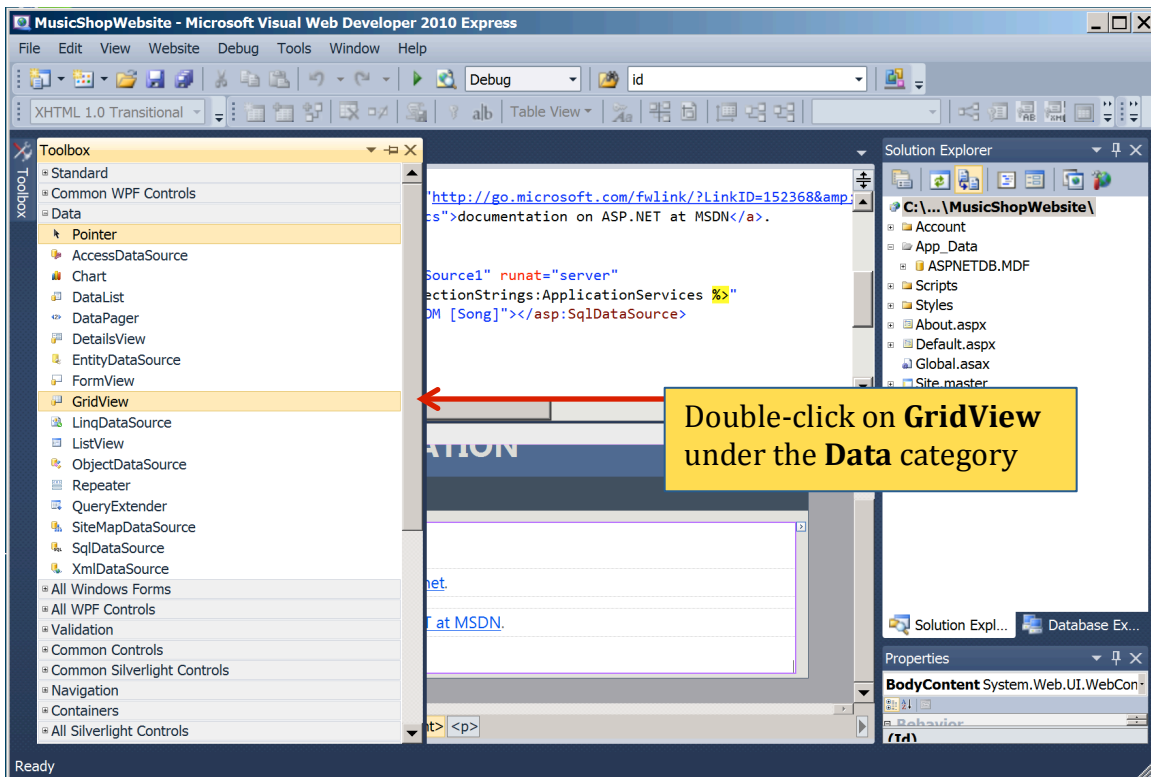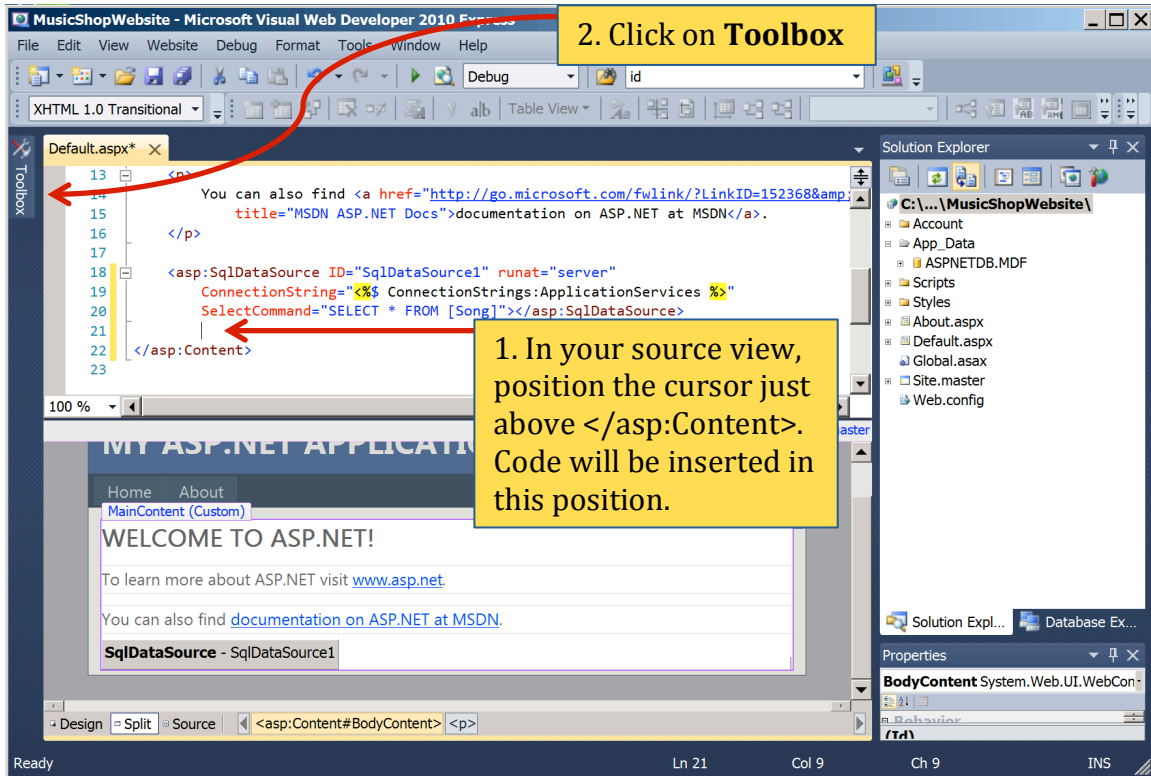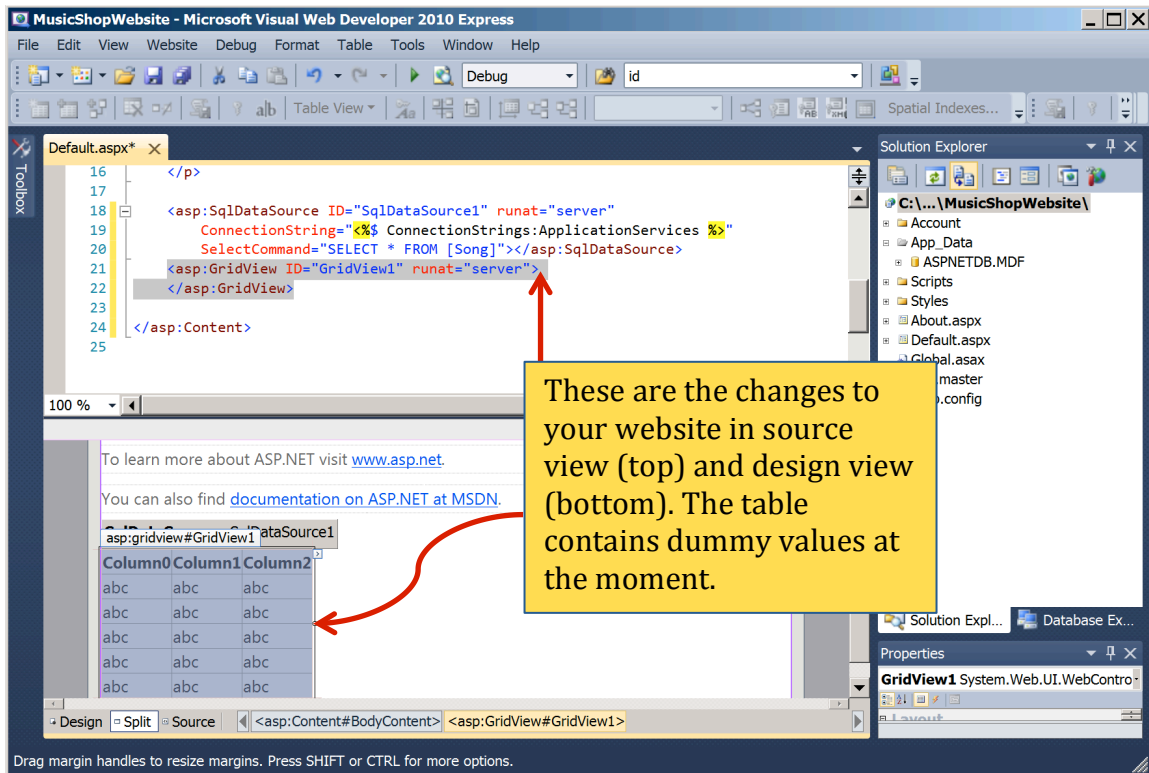

Step 5. Displaying Data with GridView

After the data source control has been added to the page and configured, displaying its data through a data Web control is a cinch: Simply drag the GridView control onto the ASP.NET page and, from its smart tag, specify what data source control to use. That's it! Take note of the name **SqlDataSource1** of your data source control (see the source code in Default.aspx).

The GridView control will display data in a grid, with one row for each record in the data source control.

Position the cursor in the Source view similar to the screen shot below.

2. Click on Toolbox

1. In your source view, position the cursor just above </asp:Content>. Code will be inserted in this position.

Double-click on GridView under the Data category

These are the changes to your website in source view (top) and design view (bottom). The table contains dummy values at the moment.

Next we need to associate a data source with the GridView. Click the arrow at the right top corner of the Grid View control to get a GridView Tasks window and select **SqlDataSource1** from the drop down.

Click on the arrow at the right top corner of the GridView control to get a GridView Tasks window and select SqlDataSource1
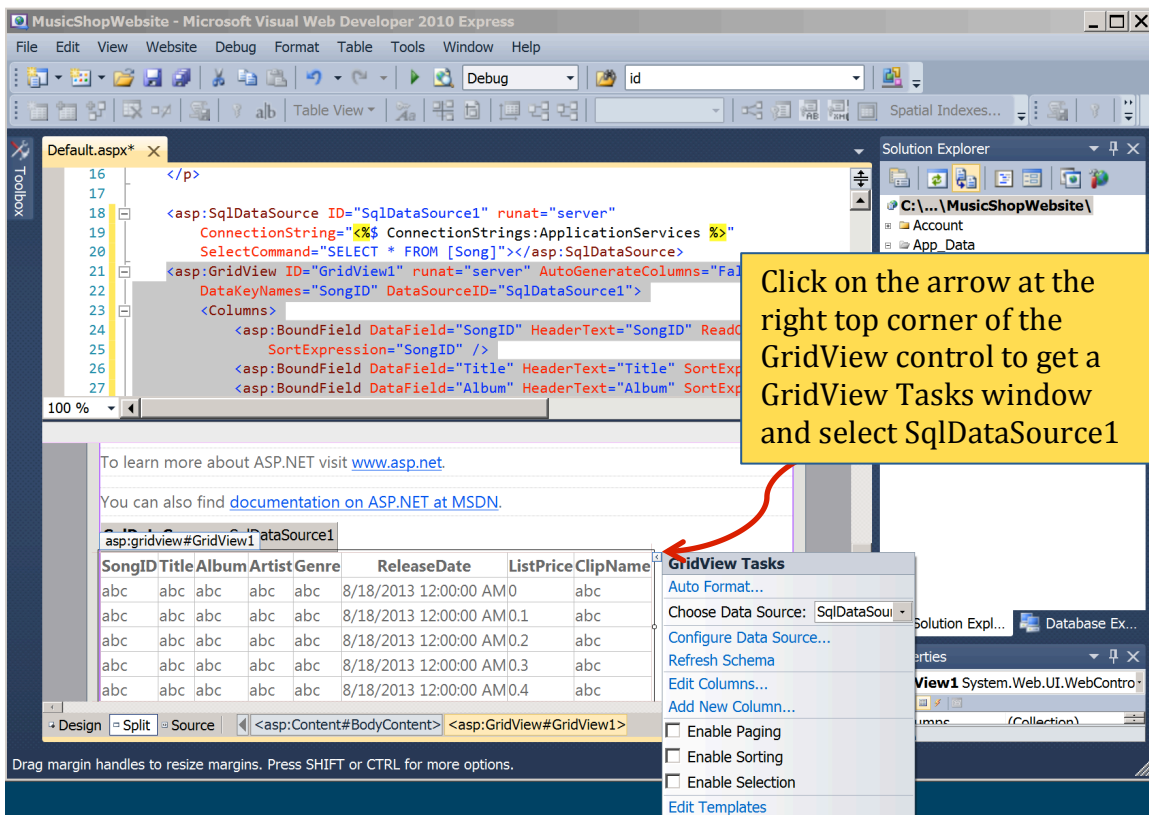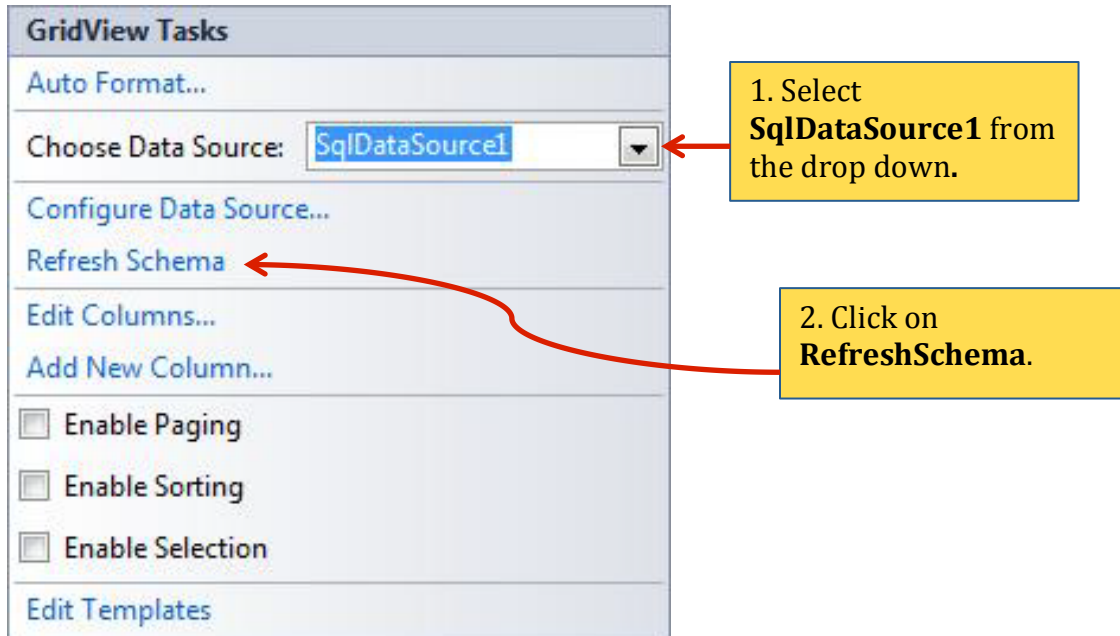
Once you have selected the Data Source, hit on Refresh Schema to view the column names of the Song table in grid view.
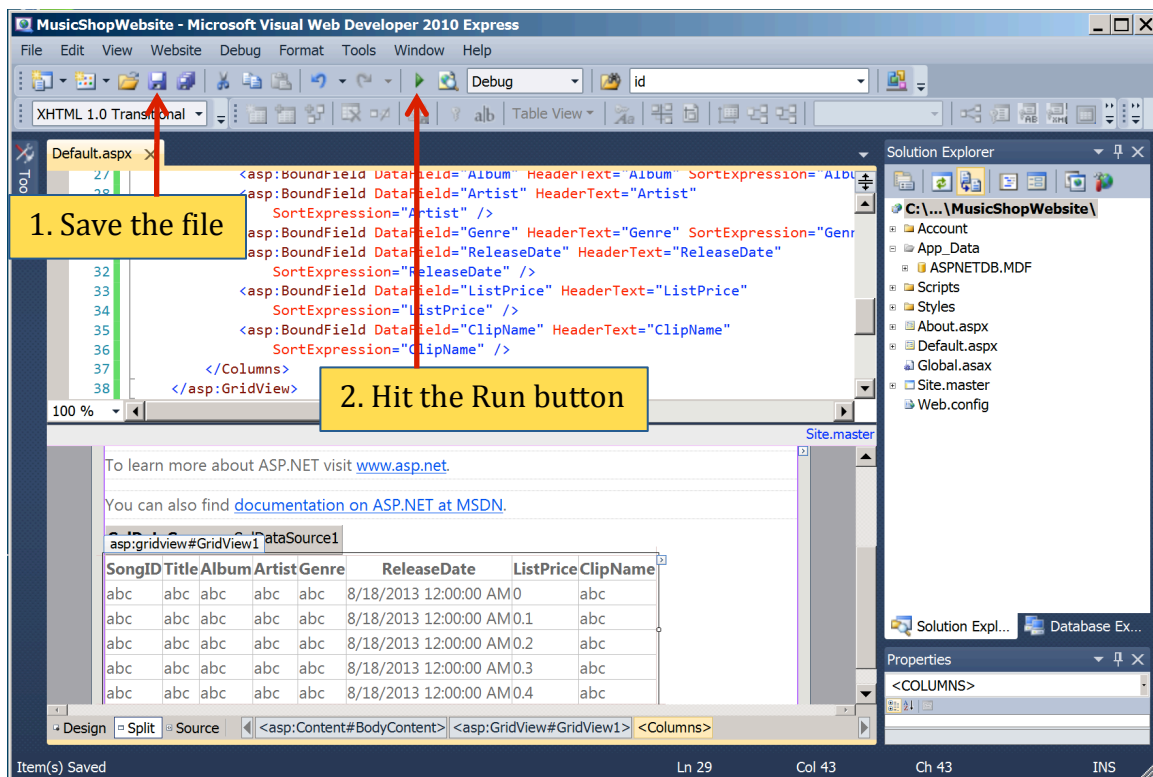
Save the file and you will be able to view the Columns from the database. Now hit the Run button, so that you can view the page in a browser.

You can now view the actual values from the database.

Make sure you stop debugging every time after you run the program, because only by doing so you can make changes to your program.

CONGRATULATIONS! You now have an ASP.NET page that displays the contents of the query executed by SqlDataSource1. Take a moment to review the process for displaying data in an ASP.NET page:

1. The first step is to add the data source control that retrieves the appropriate subset of data.

2. Next you add a data Web control (in our case GridView) and, through its smart tag, specify to use the data source added in Step 1.

Study the ASP.NET code in Default.aspx. You'll find it quite intuitive:

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
        ConnectionString="<%$ ConnectionStrings:ApplicationServices %>"
        SelectCommand="SELECT * FROM [Song]">
</asp:SqlDataSource>
```

```
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
            DataSourceID="SqlDataSource1">
<Columns>
    <asp:BoundField DataField="SongID" HeaderText="SongID" ReadOnly="True"
                    SortExpression="SongID" />
    <asp:BoundField DataField="Title" HeaderText="Title" SortExpression="Title" />
    <asp:BoundField DataField="Album" HeaderText="Album" SortExpression="Album" />
    <asp:BoundField DataField="Artist" HeaderText="Artist" SortExpression="Artist"/>
    <asp:BoundField DataField="Genre" HeaderText="Genre" SortExpression="Genre" />
    <asp:BoundField DataField="ReleaseDate" HeaderText="ReleaseDate"
                    SortExpression="ReleaseDate" />
    <asp:BoundField DataField="ListPrice" HeaderText="ListPrice"
                    SortExpression="ListPrice" />
    <asp:BoundField DataField="ClipName" HeaderText="ClipName"
                    SortExpression="ClipName" />
</Columns>
</asp:GridView>
```

Soon you might be able to type in the HTML and Web control syntax by hand, rather than using the drag-and-drop capabilities of Web Developer.