8. Rework Example 3 of Section 5.1 (Weekly Pay) as a Web program. Require that data be entered into each input text box before the button's Click event is processed and validate that the number of hours worked is at most 168 (the number of hours in a week).
9. Rework Exercise 37 of Section 5.2 (Alphabetize Two Words) as a Web program with the additional provision that both words must begin with lower-case letters. Use a RequiredFieldValidator and a RangeValidator control with each text box.
10. Rework Example 3 of Section 7.3 (Display Countries by Continent and Area) as a Web program without using a structure.
11. Rework Exercise 29 of Section 7.3 (Display Justices from a Specified State) as a Web program without using a structure.
12. Rework Example 1 of Section 7.4 (Intercity Distances) as a Web program. Use a RequiredFieldValidator and a RangeValidator control with each input text box.

Solutions to Practice Problems 12.2

1. This was done to avoid reading a text file from a disk into an array more than once.
2. With a Web program, any array loses its values between postbacks.

12.3 Using Databases in Web Programs

Databases play a prominent role in Web applications. In this section we show how to use LINQ to manipulate information retrieved from databases. The information will be displayed both in bar charts and in grids. Our bar charts will be generated by the Chart control, which is new to Visual Web Developer 2010.

This section uses the same types of databases as Chapter 10. However, we will use them in a different format called a Microsoft SQL Server format. (SQL is pronounced *sequel.*) The databases will have the extension *mdf*. In order to open these databases when using VWD Express, you must have Microsoft SQL Server installed on your computer. (Microsoft SQL Server Express is contained on the DVD accompanying this book and is usually installed when you install Visual Basic Express or Visual Web Developer Express.)

Creating a Bar Chart from a Database

The following walkthrough uses the Megacities database discussed in Section 10.2 and displays a bar chart showing the cities and their 2010 populations. The cities will be displayed in descending order of their 2010 populations. **Note:** VWD refers to a *bar chart* as a *column chart*.

The walkthrough proceeds in four stages.

Stage 1: Design the Web page.

Stage 2: Add a database connection.

Stage 3: Create an *object model* for the database. (The object model is used to enable LINQ queries to be performed on data retrieved from relational databases.)

Stage 4: Use a LinqDataSource control to display data in a Chart control.

Stage 1: Design the web page

1. Start a VWD program with the name PopBarChart and delete the text in the Main Content region.
2. Place a button control on the form, set its ID property to btnDisplay, set its Width property to 300px, and set its Text property to "Display City Populations in Descending Order".

3. Place the cursor to the right of the button control, press the Enter key twice, and then double-click on the Chart control (in the *Data* group of the Toolbox) to place a chart control on the page. The chart control will display a generic bar chart as a placeholder. (See Fig. 12.22.)
4. Set the ID property of the chart control to *chtMegacities*.

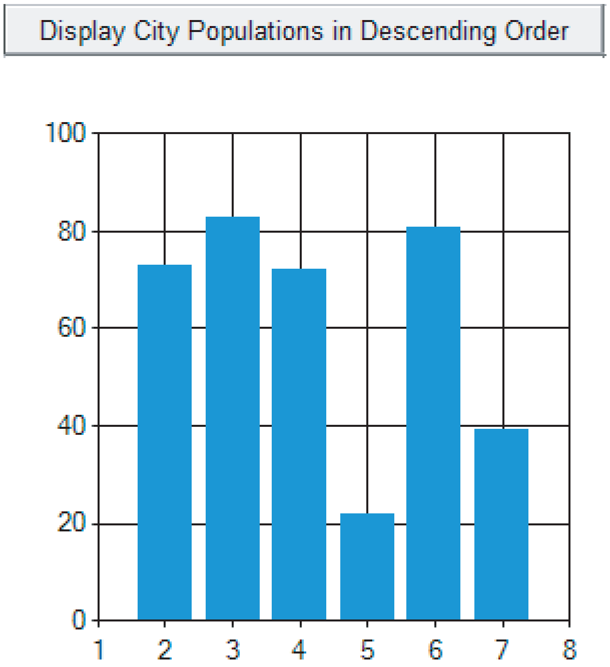


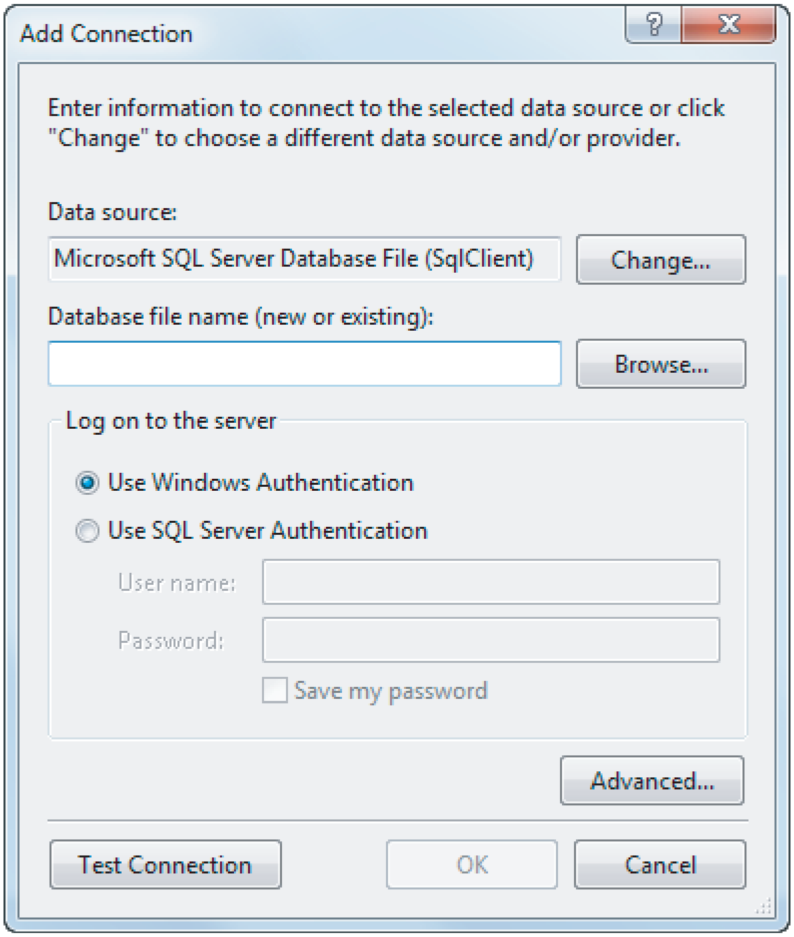
FIGURE 12.22 **Default.aspx page in Design view.**

Stage 2: Add a database connection

1. Open the Database Explorer window by clicking on its tab. Alternatively, select *Other Windows* from the *View* menu and click on *Database Explorer*. **Note:** In Visual Studio, *Database Explorer* is called *Server Explorer*.
2. Right-click on *Data Connections* in the Database Explorer window and then click on *Add Connection*. (The Add Connection dialog box in Fig. 12.23 on the next page will appear.)
3. If the Data source is not set to "Microsoft SQL Server Database File (SqlClient)", click on the *Change* button and select that data source.
4. Click on the *Browse* button, navigate to the folder Programs\Ch12\Databases in the materials you downloaded from the Pearson Web site for this book, and double-click on the file Megacities.mdf.
5. Click on the *Test Connection* button to verify that you are connected to the database.
6. Click on the *OK* button.

Stage 3: Create an object model for the database

1. Click on the Web page and then click on *Add New Item* in the *Website* menu. (An Add New Item dialog box will appear.)
2. Select *LINQ to SQL Classes*, change the name from the default name *DataClasses.dbml* to *Megacities.dbml*, and click on the *Add* button. (The window in Fig. 12.24 will appear.)
3. Click on the *Yes* button. (An Object Relational Designer consisting of two panes separated by a vertical scroll bar will fill the Document window.)



Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server Database File (SqlClient)

Database file name (new or existing):

Log on to the server

● Use Windows Authentication

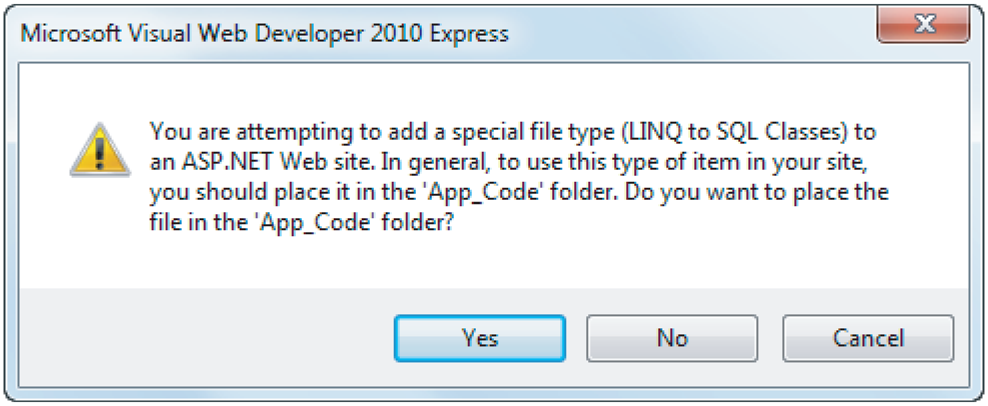
○ Use SQL Server Authentication

User name:

Password:

□ Save my password

FIGURE 12.23 The Add Connection dialog box.

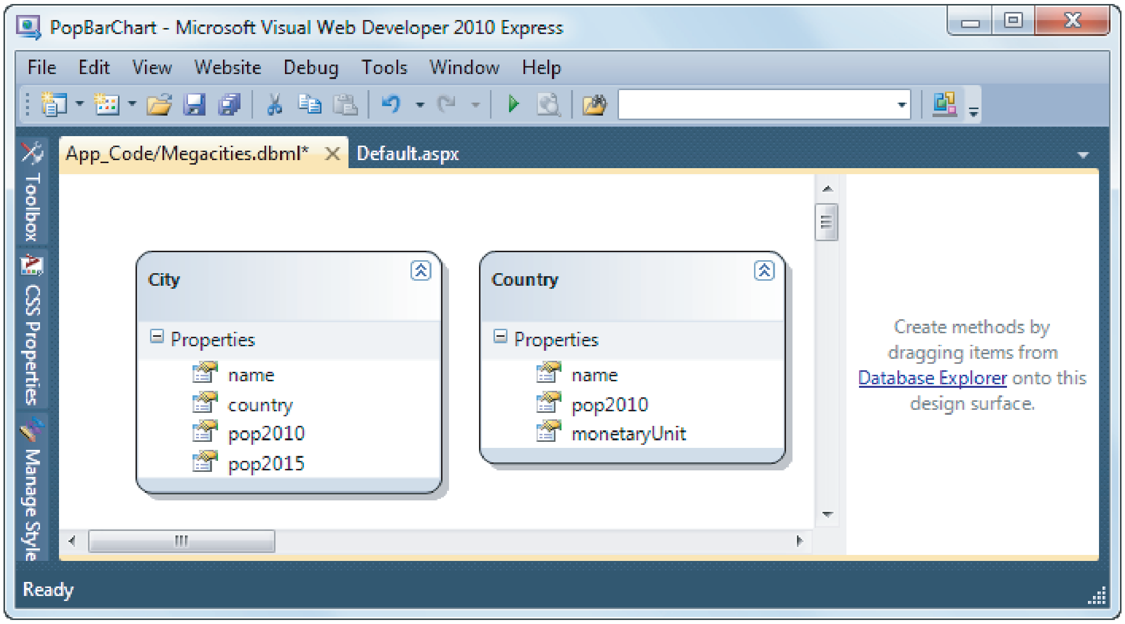


Microsoft Visual Web Developer 2010 Express

You are attempting to add a special file type (LINQ to SQL Classes) to an ASP.NET Web site. In general, to use this type of item in your site, you should place it in the 'App_Code' folder. Do you want to place the file in the 'App_Code' folder?

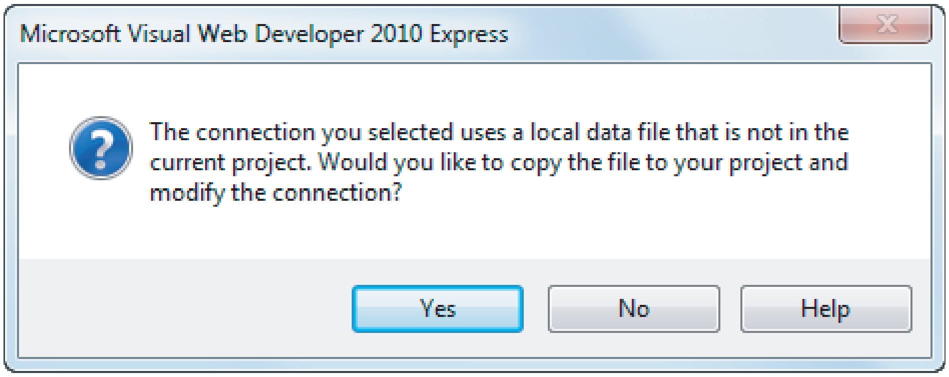
FIGURE 12.24 Placement dialog box.

4. Open Database Explorer and click on the right-pointing triangle (or plus box) to the left of Megacities.mdf to display a list of folders.
5. Click on the right-pointing triangle (or plus box) to the left of the *Tables* folder to display a list of the two tables in the database.
6. Drag each table onto the left pane of the Object Relational Designer. (The screen should look something like Fig. 12.25. Also, the dialog box in Fig. 12.26 will appear after the first table is dragged. Click on the *Yes* button in the dialog box.)



Create methods by dragging items from Database Explorer onto this design surface.

FIGURE 12.25 Object Relational Designer.



The connection you selected uses a local data file that is not in the current project. Would you like to copy the file to your project and modify the connection?

FIGURE 12.26 Copy File to Project dialog box.

7. Click on the *Save All* button in the Toolbar.

Stage 4: Use a LinqDataSource control to display data in a chart control

1. Click on the Default.aspx tab to return to the Designer and place a LinqDataSource control (from the Toolbox's *Data* group) at the bottom of the page. **Note:** This control will not be visible at run time.
2. Click on the LinqDataSource control's *Tasks* button and click on *Configure Data Source* to bring up the Configure Data Source dialog box.
3. Select MegacitiesDataContext in the "Choose your context object" combo box and click on the *Next* button. (You will now be asked to configure the data selection.)
4. The default selections made in the Configure Data Source dialog box are fine as is. Click on the *Finish* button.
5. Double-click on the *Display* button to bring up the code-behind editor, and then enter the following code:

```
Protected Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim mcDC As New MegacitiesDataContext
  Dim query = From city In mcDC.Cities
              Order By city.pop2010 Descending
              Select city.name, city.pop2010
  chtMegacities.DataBindTable(query, "name")
  chtMegacities.ChartAreas(0).AxisX.Interval = 1
  chtMegacities.ChartAreas(0).AxisX.Title = "City"
  chtMegacities.ChartAreas(0).AxisY.Title = "2010 Population in Millions"
End Sub
```

Note: The line following the query statement binds the chart to the query and uses the *name* field to populate the *x*-values. The next line guarantees that all *x*-values will be displayed.

6. Run the program and click on the *Display* button. Figure 12.27 shows the output.

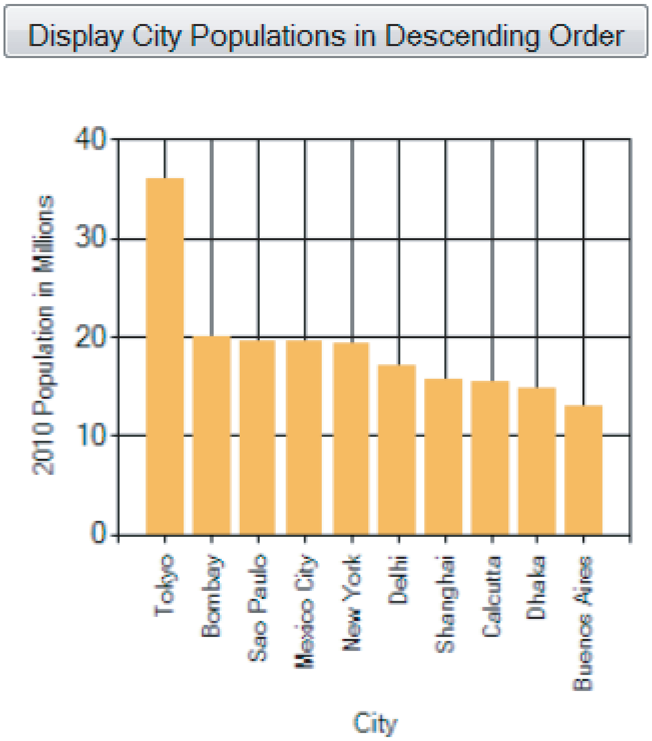


FIGURE 12.27 **Output of walkthrough.**

VideoNote
Displaying information in a grid

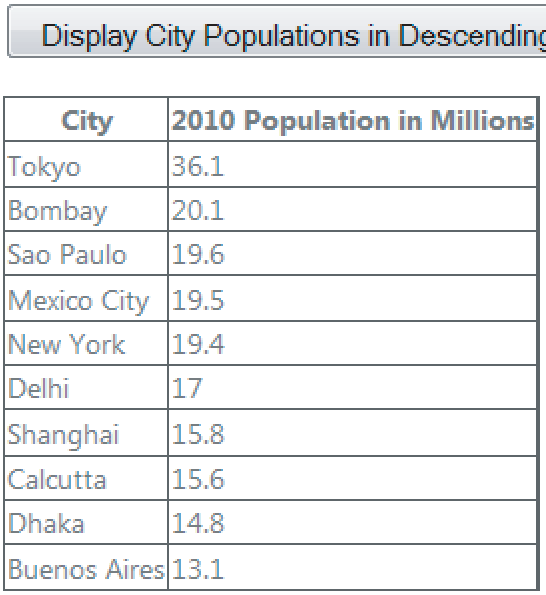
Displaying Database Information in a Grid

A slight variation of the program developed in the walkthrough above will fill a GridView control with the same information. The changes required are as follows:

1. In stage 1 place a GridView control on the page instead of a chart control. Name the control grvMegacities.
2. In the fifth step of stage 4, replace the chtMegacities statements with the four statements

```
grvMegacities.DataSource = query
grvMegacities.DataBind()
grvMegacities.HeaderRow.Cells(0).Text = "City"
grvMegacities.HeaderRow.Cells(1).Text = "2010 Population in Millions"
```

3. Figure 12.28 shows the output when the program is run and the button is clicked on.



City	2010 Population in Millions
Tokyo	36.1
Bombay	20.1
Sao Paulo	19.6
Mexico City	19.5
New York	19.4
Delhi	17
Shanghai	15.8
Calcutta	15.6
Dhaka	14.8
Buenos Aires	13.1

FIGURE 12.28 Output of GridView program.

Comments

1. The chart control can produce clustered bar charts. For instance, if the LINQ clause

```
Select city.name, city.pop2010
```

from the program PopBarChart is changed to

```
Select city.name, city.pop2010, city.pop2015
```

and the string “2010 Population in Millions” is changed to “2010 & 2015 Population in Millions”, then the clustered bar chart in Fig. 12.29 will be produced.

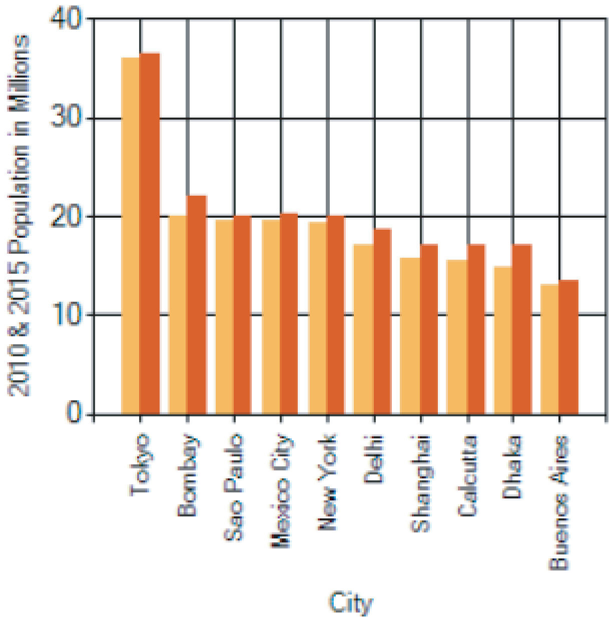


FIGURE 12.29 Clustered bar chart.

2. When LINQ is used to produce a bar chart, one of the items in the Select clause must have type String and the remaining items must have a numeric type. However, when a LINQ query is used to produce a grid, the items in the Select clause can have any data types.

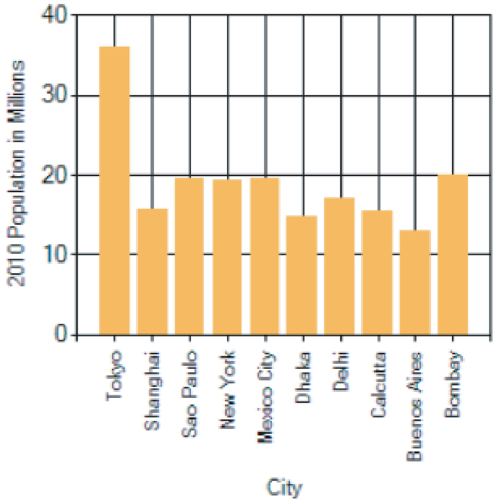
Practice Problem 12.3

1. Write a VWD program that uses the database Megacities.mdf to generate a bar chart displaying the 2010 populations of the countries whose currency is the peso.

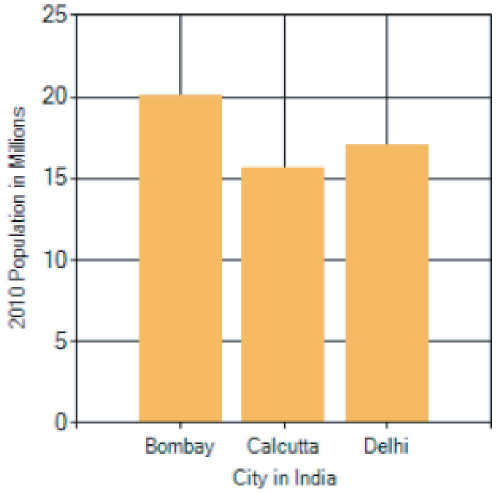
EXERCISES 12.3

In Exercises 1 through 10, write a Web program that uses the database Megacities.mdf to generate the bar chart.

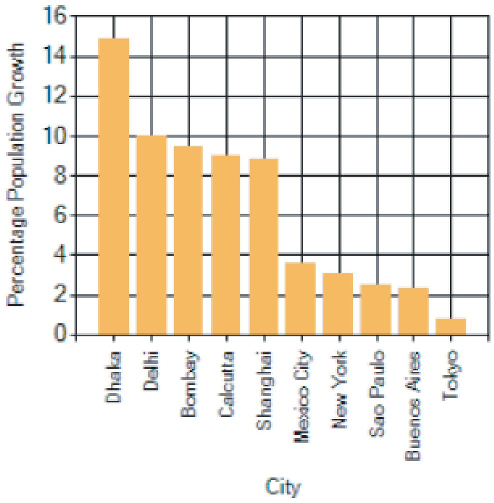
1.



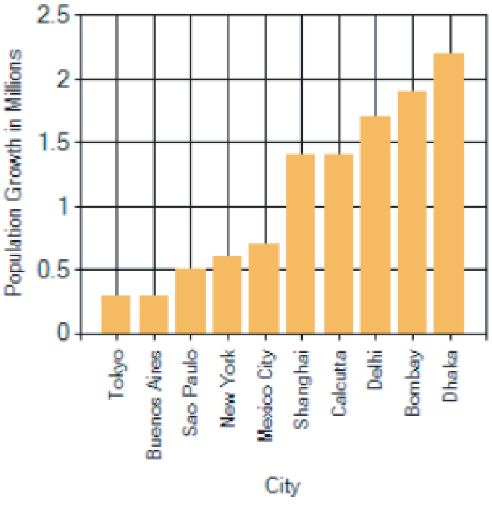
2.



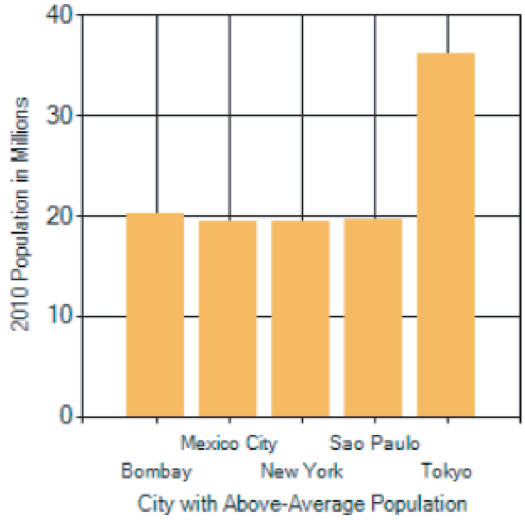
3.



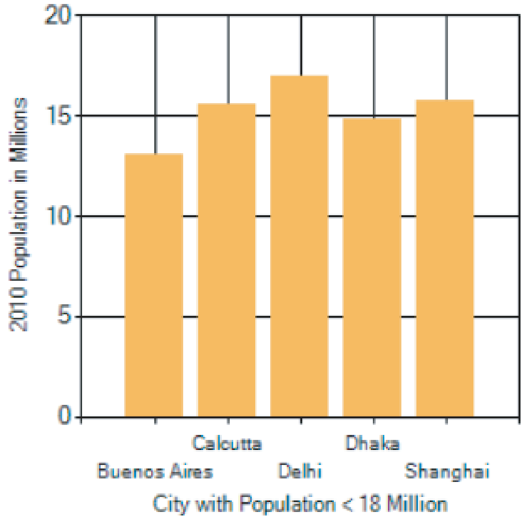
4.



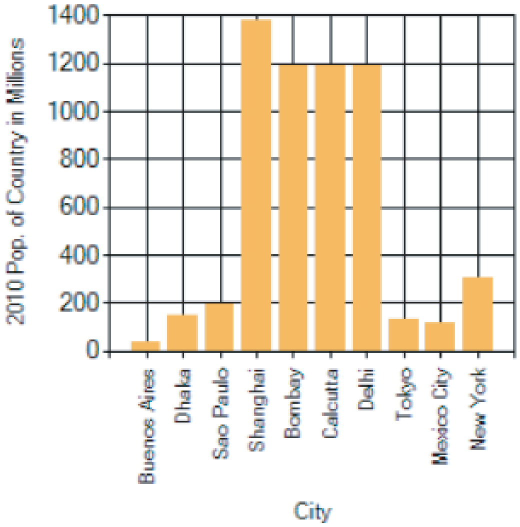
5.


2010 Population in Millions
40
30
20
10
0
Mexico City
Sao Paulo
Bombay
New York
Tokyo
City with Above-Average Population

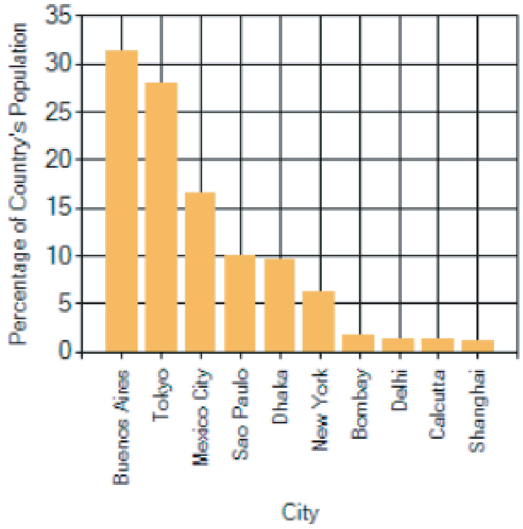
6.


2010 Population in Millions
20
15
10
5
0
Calcutta
Dhaka
Buenos Aires
Delhi
Shanghai
City with Population < 18 Million

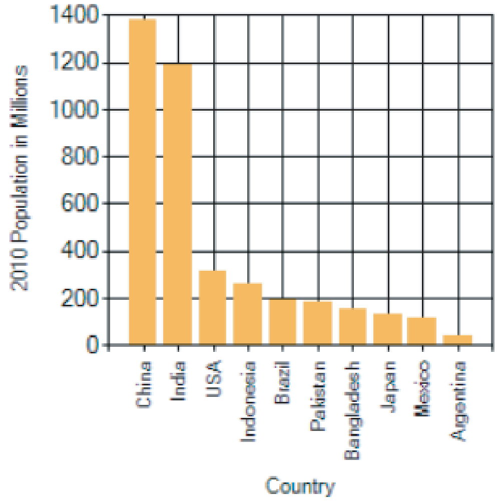
7.


2010 Pop. of Country in Millions
1400
1200
1000
800
600
400
200
0
Buenos Aires
Dhaka
Sao Paulo
Shanghai
Bombay
Calcutta
Delhi
Tokyo
Mexico City
New York
City

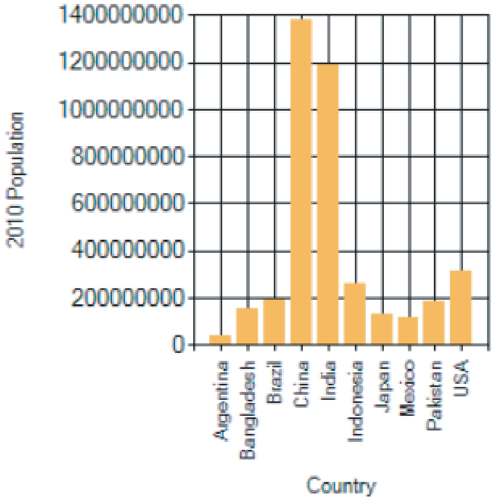
8.


Percentage of Country's Population
35
30
25
20
15
10
5
0
Buenos Aires
Tokyo
Mexico City
Sao Paulo
Dhaka
New York
Bombay
Delhi
Calcutta
Shanghai
City

9.

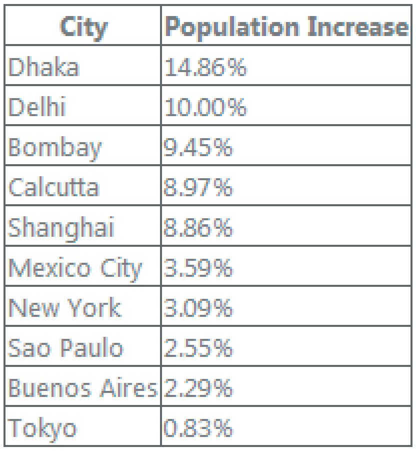

2010 Population in Millions
1400
1200
1000
800
600
400
200
0
China
India
USA
Indonesia
Brazil
Pakistan
Bangladesh
Japan
Mexico
Argentina
Country

10.


2010 Population
1400000000
1200000000
1000000000
800000000
600000000
400000000
200000000
0
Argentina
Bangladesh
Brazil
China
India
Indonesia
Japan
Mexico
Pakistan
USA
Country

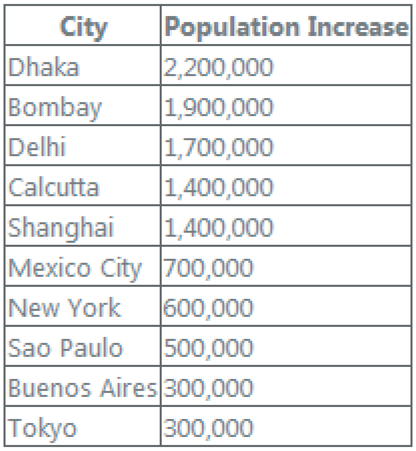
In Exercises 11 through 16, write a Web program that uses the database Megacities.mdf to generate the grid.

11.



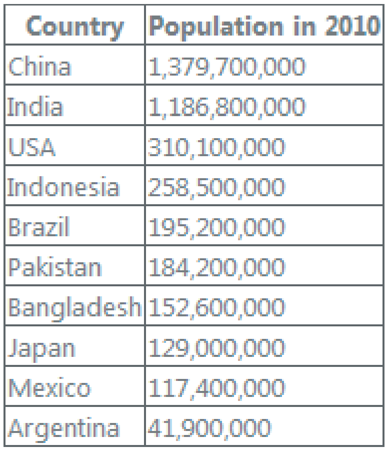
City	Population Increase
Dhaka	14.86%
Delhi	10.00%
Bombay	9.45%
Calcutta	8.97%
Shanghai	8.86%
Mexico City	3.59%
New York	3.09%
Sao Paulo	2.55%
Buenos Aires	2.29%
Tokyo	0.83%

12.



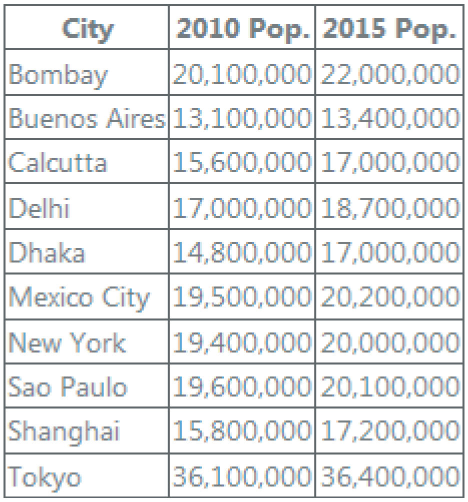
City	Population Increase
Dhaka	2,200,000
Bombay	1,900,000
Delhi	1,700,000
Calcutta	1,400,000
Shanghai	1,400,000
Mexico City	700,000
New York	600,000
Sao Paulo	500,000
Buenos Aires	300,000
Tokyo	300,000

13.



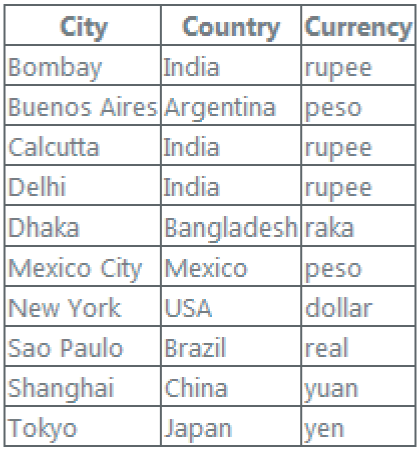
Country	Population in 2010
China	1,379,700,000
India	1,186,800,000
USA	310,100,000
Indonesia	258,500,000
Brazil	195,200,000
Pakistan	184,200,000
Bangladesh	152,600,000
Japan	129,000,000
Mexico	117,400,000
Argentina	41,900,000

14.



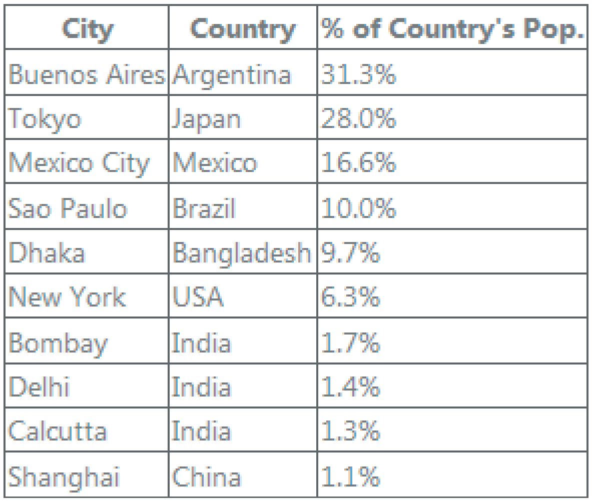
City	2010 Pop.	2015 Pop.
Bombay	20,100,000	22,000,000
Buenos Aires	13,100,000	13,400,000
Calcutta	15,600,000	17,000,000
Delhi	17,000,000	18,700,000
Dhaka	14,800,000	17,000,000
Mexico City	19,500,000	20,200,000
New York	19,400,000	20,000,000
Sao Paulo	19,600,000	20,100,000
Shanghai	15,800,000	17,200,000
Tokyo	36,100,000	36,400,000

15.



City	Country	Currency
Bombay	India	rupee
Buenos Aires	Argentina	peso
Calcutta	India	rupee
Delhi	India	rupee
Dhaka	Bangladesh	raka
Mexico City	Mexico	peso
New York	USA	dollar
Sao Paulo	Brazil	real
Shanghai	China	yuan
Tokyo	Japan	yen

16.



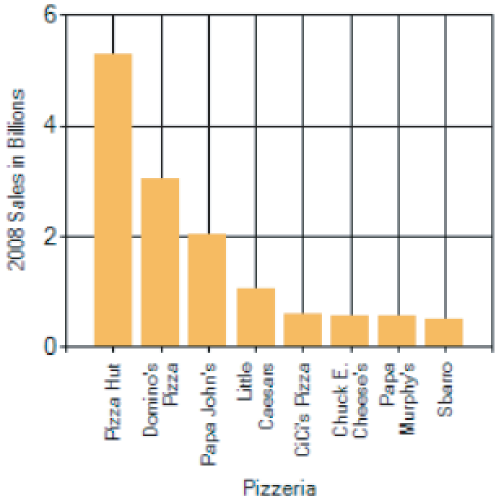
City	Country	% of Country's Pop.
Buenos Aires	Argentina	31.3%
Tokyo	Japan	28.0%
Mexico City	Mexico	16.6%
Sao Paulo	Brazil	10.0%
Dhaka	Bangladesh	9.7%
New York	USA	6.3%
Bombay	India	1.7%
Delhi	India	1.4%
Calcutta	India	1.3%
Shanghai	China	1.1%

The database Pizza.mdf has a single table *Pizzerias* that contains data for the eight leading pizza chains. The fields for the table are *name*, *sales2007*, *sales2008*, *numStores2007*, and *numStores2008*. (Sales are given in thousands of dollars.) Two records in the table are

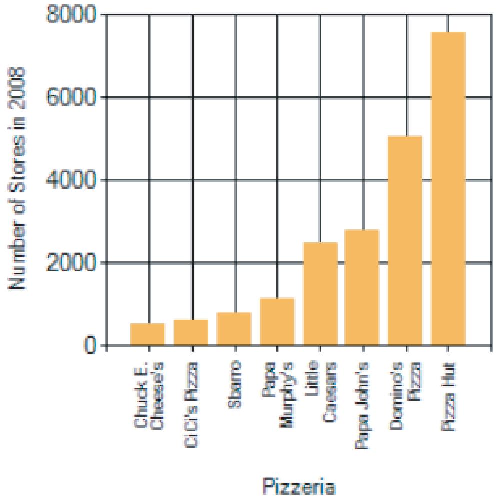
Pizza Hut	5100000	5300000	7466	7564
Domino’s Pizza	3194485	3037703	5136	5047

In Exercises 17 through 22, write a Web program that uses the database Pizza.mdf to generate the bar chart.

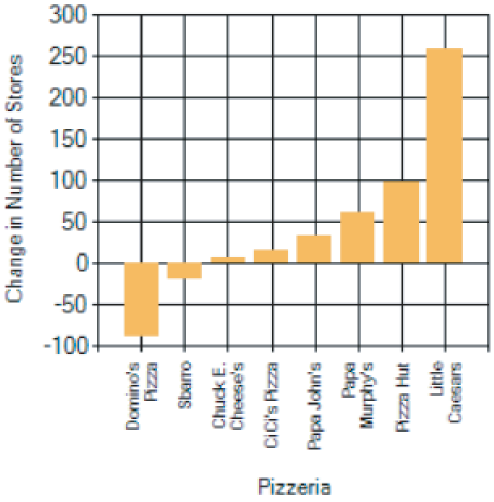
17.



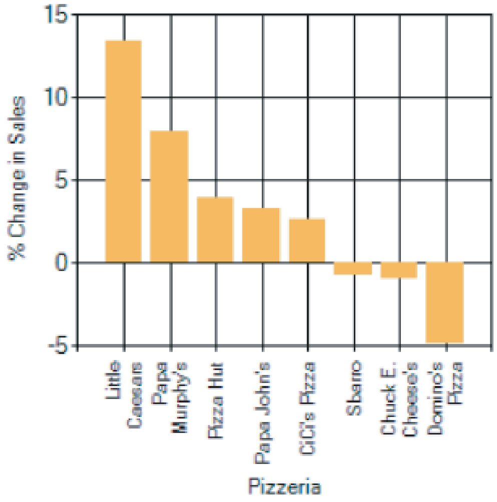
18.



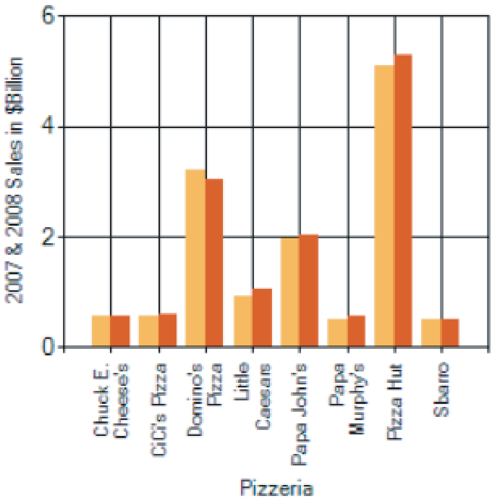
19.



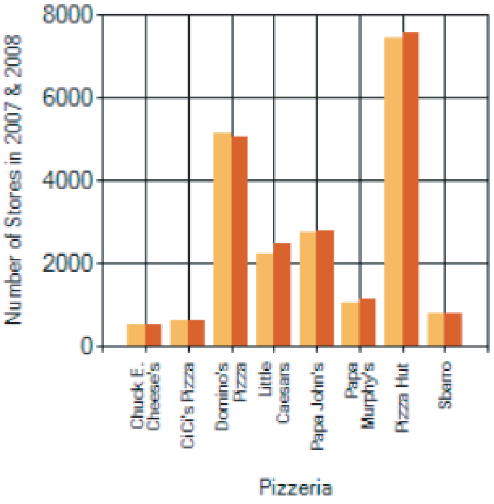
20.



21.

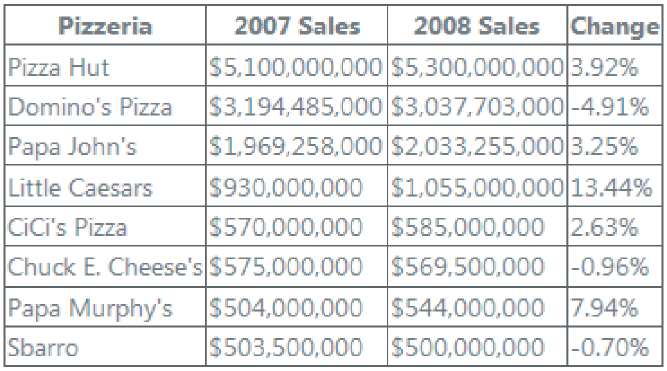


22.



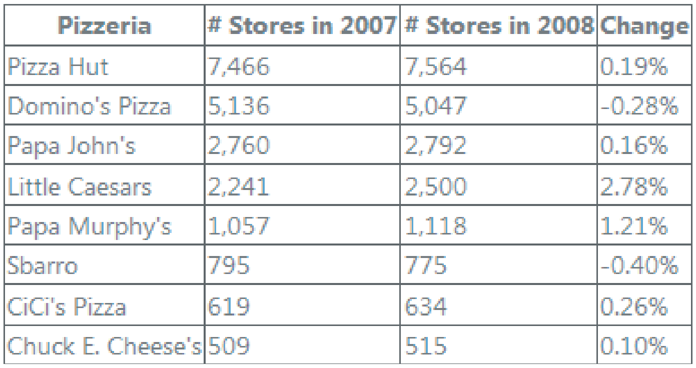
In Exercises 23 and 24, write a Web program that uses the database Pizza.mdf to generate the grid.

23.



Pizzeria	2007 Sales	2008 Sales	Change
Pizza Hut	$5,100,000,000	$5,300,000,000	3.92%
Domino's Pizza	$3,194,485,000	$3,037,703,000	-4.91%
Papa John's	$1,969,258,000	$2,033,255,000	3.25%
Little Caesars	$930,000,000	$1,055,000,000	13.44%
CiCi's Pizza	$570,000,000	$585,000,000	2.63%
Chuck E. Cheese's	$575,000,000	$569,500,000	-0.96%
Papa Murphy's	$504,000,000	$544,000,000	7.94%
Sbarro	$503,500,000	$500,000,000	-0.70%

24.



Pizzeria	# Stores in 2007	# Stores in 2008	Change
Pizza Hut	7,466	7,564	0.19%
Domino's Pizza	5,136	5,047	-0.28%
Papa John's	2,760	2,792	0.16%
Little Caesars	2,241	2,500	2.78%
Papa Murphy's	1,057	1,118	1.21%
Sbarro	795	775	-0.40%
CiCi's Pizza	619	634	0.26%
Chuck E. Cheese's	509	515	0.10%

In Exercises 25 through 28 write a program that uses the database Pizza.mdf and displays the stated information for the year 2008 in a read-only text box or in a list box.

25. The total number of stores for the top eight pizza chains.

26. The names of the top three pizza chains in sales.

27. The name of the pizza chain having the least number of stores of any of the top eight.

28. The name of the pizza chain having the greatest increase in sales of any of the top eight.

Solution to Practice Problems 12.3

1. There is no need to carry out the four stages from the walkthrough, since we have already created an object model and LinqDataSource control for this database. The following steps create the new program, and Fig. 12.30 shows the bar chart that is produced:

   (a) Start a new Web program with the name PracticeProb and then close the program.

   (b) Use the process for building on an existing Web program discussed at the end of Section 12.1 to make PracticeProb a clone of the program created in the walkthrough.

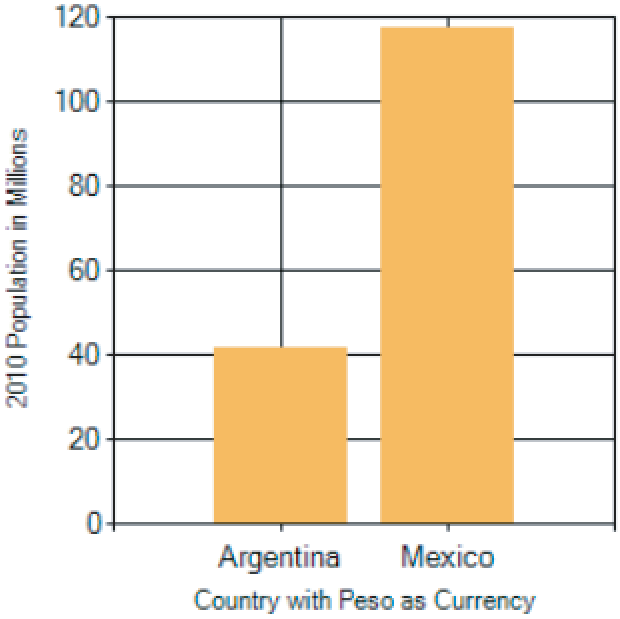


FIGURE 12.30 **Output of Practice Problem.**

   (c) Open PracticeProb in VWD and change the query to the following:

```
Dim query = From country In mcDC.Countries
            Where country.monetaryUnit = "peso"
            Select country.name, country.pop2010
```

   (d) Change the string in the next to last line of the code from “City” to “Country with Peso as Currency”.

CHAPTER 12 SUMMARY

1. *Visual Web Developer* (VWD) is used to create programs that reside on a server and run in the Web browser of a client computer.
2. There are many similarities between creating a Web program with VWD and creating a Visual Basic Windows program. However, there are some important differences. For one thing, text can be typed and formatted directly into Web pages. Also, not as many controls are available in VWD as in Visual Basic, and controls that are common to both have fewer properties and events in VWD than in Visual Basic.
3. The *table control* is helpful for aligning text and controls in VWD.
4. *Validation controls* allow the browser to check input before it is sent to the server.
5. The *hyperlink control* is used to request different pages from the server.