Chapter 1: Building Master and Detail Pages

In This Chapter

- **Developing master and detail pages at the same time**
- **Building your master and detail pages separately**
- **Putting together master and detail pages for ASP.NET**
- **Making sure your master and detail pages work**

You can use Dreamweaver to build master and detail Web pages, which are a popular way to display information on your Web site. A *master page* displays a list of records and corresponding links for each record. A user can click a link to see further information about a record on its associated *detail page.*

Depending on your programming language, you can either build the set of master and detail pages all in one operation, or you can separately build the master pages and then the detail pages. You can find out more about both methods in this chapter.

The examples in this chapter use a simple Employee table, which you can see in Table 1-1. The first value, the `employee_id`, is a special field called a *key.* It always has a unique value. Therefore, if you query for a record in a table by using the key, you always get only the row that you're looking for because no two rows have the same key.

Table 1-1 The Employee Table

Column Name	*Type of Data*
`employee_id`	Integer
`first_name`	50 Character String
`last_name`	50 Character String
`aAddress1`	75 Character String
`aAddress2`	75 Character String
`cCity`	50 Character String
`sState`	50 Character String

(continued)

Table 1-1 *(continued)*	
zZip	5 Character String
pPhone	14 Character String
department_id	Integer

Building Master and Detail Pages in One Operation (PHP, ASP, JSP, ColdFusion)

For PHP, ASP, JSP, and ColdFusion, you can build sets of master and detail pages all in one operation. You can use the same method for all these languages. Creating both the master and detail pages at the same time is generally easier, but Dreamweaver gives you the flexibility of creating them separately, too.

For ASP.NET, you need to build the master and detail pages separately (as described in the section "Developing Master and Detail Pages for ASP.NET," later in this chapter). You can't build these pages at the same time in one operation for ASP.NET.

To create a master and detail page set for PHP, ASP, JSP, or ColdFusion, follow these steps:

1. **Create a new or open an existing PHP, ASP, JSP, or ColdFusion page.**

   A blank page opens in Dreamweaver; this page becomes your master page in the language that you selected.

2. **Define a recordset for the page.**

   For the lowdown on creating a recordset, check out Book VIII, Chapter 1.

   The recordset provides the data that's displayed on both the master and detail pages. Make sure that you include all the table columns that you need to create your master page, including the unique key (the record ID column) for each record and all the table columns that you need to create your detail page. Typically, you show more columns on the detail page than on the master page.

   For the example in this chapter, we created an Employee recordset and included all the columns in the Employee table (see Table 1-1).

3. **Save your changes to the master page.**

   The new recordset appears in the Bindings panel, and a small yellow code indicator is inserted in the Document window, as shown in Figure 1-1. This page allows a listing of employees to display on the master page, which provides links to detail pages with more data than the master page.

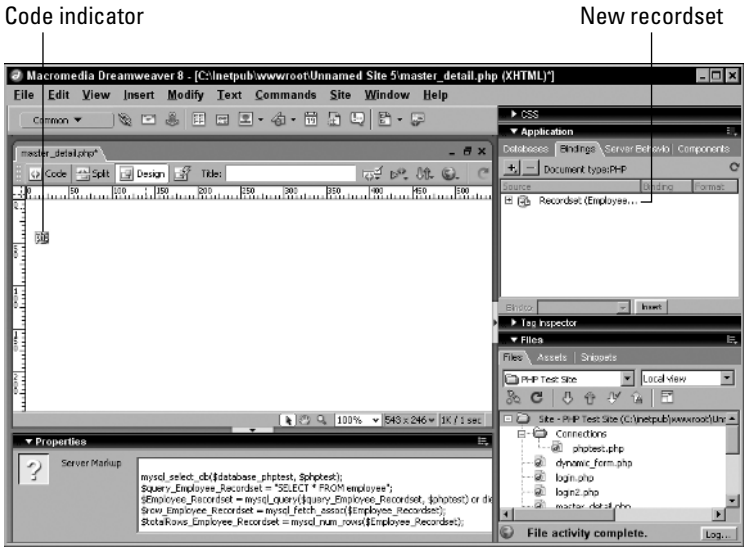

Figure 1-1: The Document window with a recordset defined.

4. **Choose Insert➪Application Objects➪Master Detail Page Set to insert the master and detail pages all in one operation.**

   The Insert Master-Detail Page Set dialog box appears. You specify the properties for the master page in the top half of this dialog box and the properties for the detail page in the lower half.

5. **From the Recordset drop-down list, select the recordset that you want to use for the master page.**

   For our example, we selected Employee_Recordset.

   After you select a recordset, Dreamweaver fills in the rest of the fields with the columns from the recordset.

6. **In the Master Page Fields area, select which records you want to appear on the master page.**

   Click the plus (+) button to add a field, and click the minus (–) button to remove a field. In Figure 1-2, we selected the `first_name` and `last_name` fields. These fields will appear on the master page in a table format.

   Typically, fewer fields appear on the master page than the detail page.

7. **From the Link to Detail From drop-down list, select the field in the recordset that you want to serve as the link to the detail page.**

   For example, we selected the `last_name` field to serve as the link to the detail page for each record.

8. **From the Pass Unique Key drop-down list, select which field contains the values that you want to pass on to the detail page so it can identify the records.**

   Typically, you select the key field that ends in ID. This key field tells the detail page which record to display for the user. For our example, we selected the `employee_id` field.

9. **Specify the number of records that you want to show at one time on the master page.**

   In the example, we chose to show 10 records at a time.

10. **In the Detail Page Name text box, enter a name for the detail page or click the Browse button to select an existing file.**

    For the example, we entered `detail.php`. Dreamweaver will automatically use this name when it creates the detail page.

11. **In the Detail Page Fields area, select which records you want to appear on the master page.**

    Click the plus (+) button and minus (–) button to change the fields that appear on the detail page.

    Typically, more fields appear on the detail page than the master page. For the example, we selected all the fields listed in Table 1-1.

    Figure 1-2 shows the configuration of the Insert Master-Detail Page Set dialog box based on the example values.

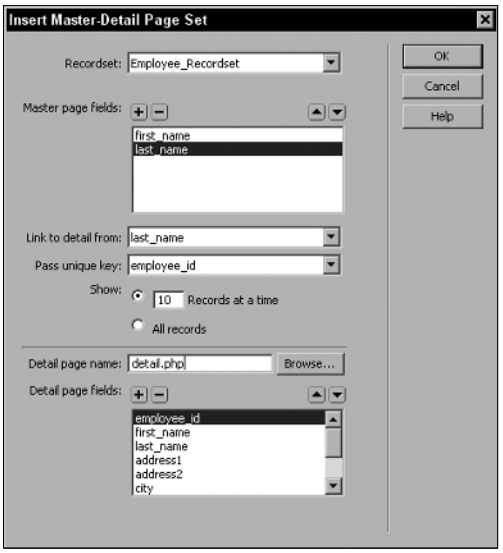

Figure 1-2: The Insert Master-Detail Page Set dialog box for the `Employee_Recordset`.

12. Click OK.

Dreamweaver creates the master and detail pages and includes dynamic content and server behaviors in both.

The Document window contains the automatically generated objects (a repeated region, navigation objects, record, counter, and link to the detail page), as shown in Figure 1-3.

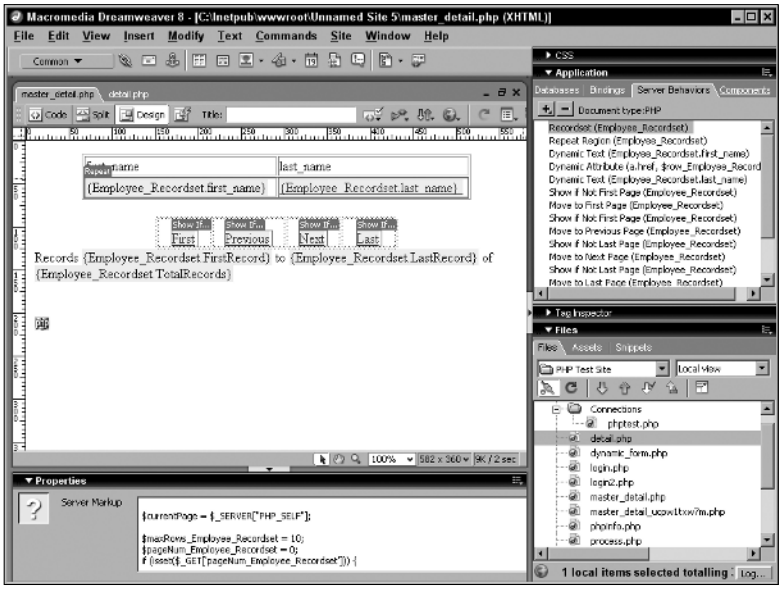

Figure 1-3: This Document window has a repeated region, navigation objects, record counter, and link to the detail page.

13. Modify the design of the master and detail pages.

You can modify your dynamic fields just like you'd edit any other object.

When you finish designing the pages, you're ready to view them in a browser. See the section "Testing Your Master and Detail Pages," later in this chapter, for details.

Developing Master and Detail Pages Block by Block

You can develop a master page block by block for PHP, ASP, JSP, and ColdFusion. While you usually create the master and detail pages at the same time, you can create them block by block to have complete control over the placement of the blocks.

Creating the master page

To create a dynamic master page, follow these steps:

1. **Create a page and define a recordset.**

   Turn to Book VIII, Chapter 1 to find out how to define a recordset.

   The Recordset dialog box varies slightly between dynamic page types. Specifically, ColdFusion calls the database connection a data source and includes optional Username and Password fields for the database. The dialog box also has a button to access ColdFusion components. The rest of the dialog box works exactly the same as the dialog box for other development code types.

2. **In the Document window, place the insertion point where you want the records to appear on the page.**

3. **Choose Insert➪Application Objects➪Dynamic Data➪Dynamic Table.**

   The Dynamic Table dialog box opens, as shown in Figure 1-4.

Figure 1-4: The Dynamic Table dialog box for the `Employee Recordset`.

4. **From the Recordset drop-down list, select the name of the recordset that you want to appear on the master page.**

   For the example, we selected the EmployeeRecordset.

5. **Specify the number of records that you want to show at one time on the master page.**

   In the example, we wanted to show 10 records at a time.

6. **(Optional) Specify border, cell padding, and cell spacing.**

7. **Click OK to close the dialog box.**

   The master page is created.

If you don't want users to see some of the columns on the master page (such as the Record ID column, which describes the record to display but is not useful for the end user), delete the column from the table by following these steps:

1. **In Design view, click anywhere on the master page.**
2. **Put the pointer near the top of the Record ID column so that the column's entries are outlined in red. Then click to select the column.**
3. **Click the Delete button to delete the column from the table.**

   Generally, fewer records appear on the master page than the detail page.

Setting up links to open a detail page

After you create the master page (as the preceding section describes), you need to create links that open the detail page and communicate which record the user selected so that only the detail for that record displays.

To set up links to open a detail page, follow these steps:

1. **Open the master page in the Document window.**
2. **In the table, select the placeholder for the dynamic content on which you want to create a link.**
3. **In the Properties inspector, click the folder button next to the Link field.**

   The Select File dialog box appears (see Figure 1-5).

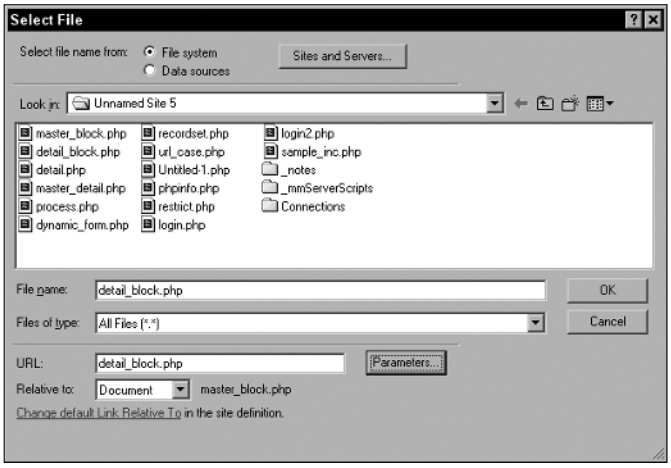

Figure 1-5: Use the Select File dialog box to configure which page to link to and which parameters to send.

4. **Browse to and select the detail page.**

5. **Click the Parameters button to the right of the URL field.**

   The Parameters dialog box opens.

6. **Click the plus (+) button to add a parameter.**

   This parameter tells the detail page which row to display. Select the key value column as this parameter.

7. **In the Name column, enter the column name.**

   For the Employee table (Table 1-1) example, the key field is `employee_id`, as shown in Figure 1-6.

Figure 1-6: Configuring the URL parameter and its recordset.

8. **Click in the Value column and then click the Dynamic (lightning bolt) button.**

   The Dynamic Data dialog box displays. This is where you select the column from the recordset.

9. **Expand the recordset, click the key field, and then click OK.**

   In this case, the key field is `employee_id`.

   After you click OK, the Parameters dialog box displays the new parameter and the code that places it into the page dynamically, as shown in Figure 1-6.

10. **Click OK to close the Parameters dialog box.**

    The URL field in the Select File dialog box is pre-populated with the new parameter.

Each dynamic page type has different code that appears because each programming language uses a slightly different syntax to display a URL variable dynamically. Fortunately, because Dreamweaver is generating the code for you, you don't need to know the syntax differences.

11. **Click OK to close the Select File dialog box.**

    You return to the Document window. The name of the detail page appears in the Link field in the Properties inspector. The placeholder for the dynamic content is now a link.

12. **Save your changes to the master page.**

    You now have a complete master page.

Read on to find out how make the detail page.

Building detail pages

To create a detail page for PHP, ASP, JSP, and ColdFusion page types, follow these steps:

1. **Create a new or open an existing PHP, ASP, JSP, or ColdFusion page.**
2. **In the Bindings panel, click the plus (+) button and select Recordset (Query) from the menu that appears.**

   The simple Recordset dialog box appears.

   If you want to write your own SQL statements, click the Advanced button to display the advanced Recordset dialog box.

3. **In the Name text box, enter a name for your recordset.**

   You can use only letters, numbers, and underscores in the name.
4. **Select a database connection for obtaining the data that you want to display.**
5. **Select a table name for obtaining the data that you want to display.**

   After you select a table name, the database columns appear in the Columns list. For example, Figure 1-7 shows the columns for the Employee table (see Table 1-1).

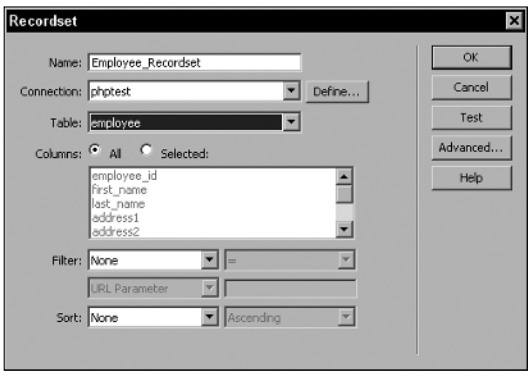

Figure 1-7: The Recordset dialog box after selecting a database table.

Depending on your dynamic page type, the Recordset dialog box may appear slightly different from the one in Figure 1-7. ColdFusion, in particular, calls the Connection field a data source and includes the Username and Password fields for the database. However, these differences don't change the following steps.

6. **Select which columns will provide the record data to display.**

   To use all columns, select All; otherwise, choose Selected and Ctrl+click (Windows) or ⌘+click (Mac) in the list to indicate which columns you want to use.

   Typically, your detail page uses more columns than your master page. You want the recordset for your detail page to contain at least one column (generally the record ID column) that matches the column that you use for the master page.

7. **Complete the Filter sections as follows:**

   - **The first Filter field:** Select the database column name that contains values to match against the URL parameter. You use the filter to find and display the record specified by the URL parameter passed from the master page.
   - **The second Filter field:** Select the equals (=) symbol, if it's not already selected. This requires the fields to be equal, which they must be to display only the record that is detailed.
   - **The third Filter field:** Select the URL parameter.
   - **The fourth Filter field:** Enter the name of the URL parameter that you want the master page to pass to the detail page.

   The recordset now returns only the data for the employee who's been selected on the master page.

8. **Click the Test button.**

   The Test Value dialog box appears.

9. **Enter a value in the Test Value field and click OK.**

   This value represents which detail record displays. This test helps you check that the detail page brings back the data you expect.

   A table displaying data from the recordset appears.

10. **Click OK.**

    The Test SQL Statement window closes.

11. **In the Recordset dialog box, click OK.**

    Figure 1-8 shows the recordset in the Bindings panel of the detail page.

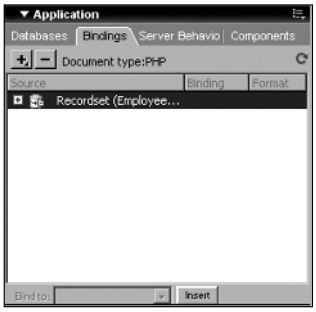

Figure 1-8: The Employee Recordset appears in the Bindings panel.

12. **To bind the columns in the recordset to the detail page, select the columns in the Bindings panel and drag those columns onto the detail page.**

    Your detail page can now process requests from the master page. Figure 1-9 shows the Document window after adding fields from the recordset.

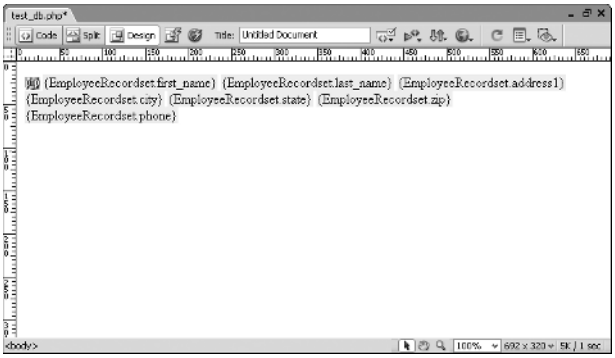

Figure 1-9: The Document window after dragging several fields from the recordset onto it.

See the section "Testing Your Master and Detail Pages," later in this chapter, to find out how to preview your master and detail pages in a browser.

Developing Master and Detail Pages for ASP.NET

For ASP.NET, you can use the DataSet Web control to specify table columns and the DataGrid Web control to list the database records to display on the master page. The Web controls provide an easy way to display database data with controls for paging between multiple pages of records.

You need to define a database connection for the site before you create the master page. See Book VII for details.

Creating a master page

To create an ASP.NET master page, follow these steps:

1. **Create a new or open an existing ASP.NET page in Dreamweaver.**
2. **In the Bindings panel, click the plus (+) button and select DataSet (Query) from the menu that appears.**

   The DataSet dialog box appears.
3. **Complete the DataSet dialog box and then click OK.**

   Make sure to include all table columns that you need to create your master page, including the unique key (Record ID column) for each record.

   A dataset is essentially the same thing as a recordset; see Book VIII, Chapter 1 for more information on recordsets.

   The new dataset appears in the Binding panel.
4. **In the Server Behaviors panel, click the plus (+) button and select DataGrid from the menu that appears.**

   The DataGrid dialog box opens.
5. **Select the dataset source from the DataSet drop-down list and click OK.**

   You can leave the default column type as Simple Data Field. The DataGrid dialog box for our example looks like Figure 1-10.

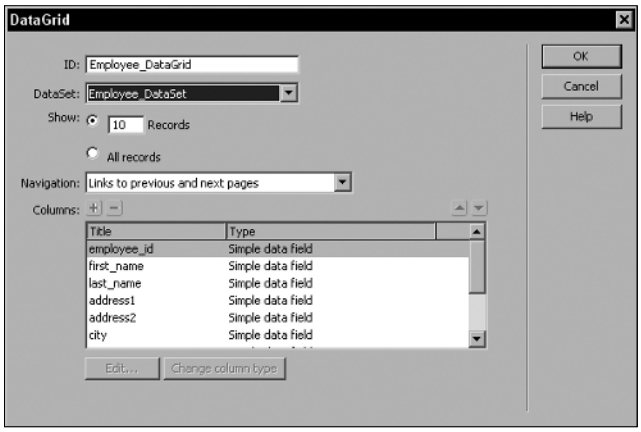

Figure 1-10: The DataGrid dialog box has the `Employee_Dataset` selected.

Figure 1-11 shows the DataGrid object created for the example.

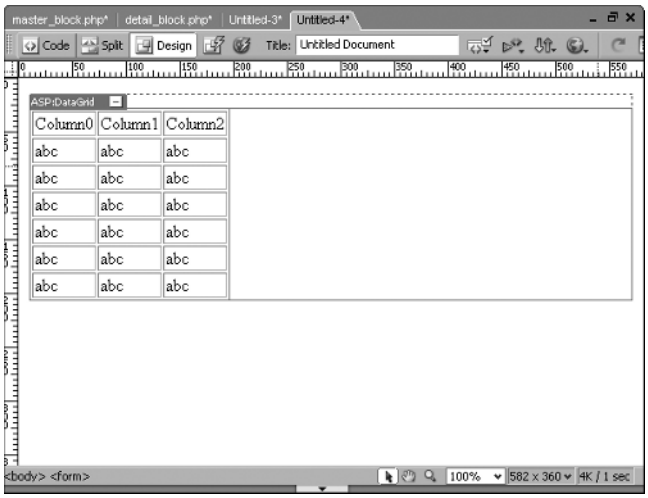

Figure 1-11: The Document window with the DataGrid object.

Creating links that open the detail page

After you create the ASP.NET master page (as described in the preceding section), you need to create links that open the detail page and communicate which record the user selected. To open an ASP.NET detail page and display the specified record, follow these steps:

1. **Open the master page in the Document window.**
2. **In the Server Behaviors panel, double-click DataGrid.**

   The DataGrid dialog box appears.
3. **In the Columns list, select the column that you want to serve as the link to the detail page.**

   You can use any field that identifies the record for expanding the detail of the record.
4. **Click the Change Column Type button.**
5. **Select Hyperlink from the pop-up menu.**

   The Hyperlink Column dialog box appears.
6. **In the Hyperlink Text section, specify the text that you want to display in the hyperlink column.**

Here are your choices:

- **Static Text:** Select this radio button if you want to use a generic description for this column. Enter text for the link, such as **Details**.
- **Data Field:** Select this radio button if you want to add text for a link based on a hyperlink column. Then from the drop-down list, select a data field in your dataset. In the example, we selected the `last_name` column, as shown in Figure 1-12.
- **Format String:** This field is automatically generated and shows the format of the URL text.

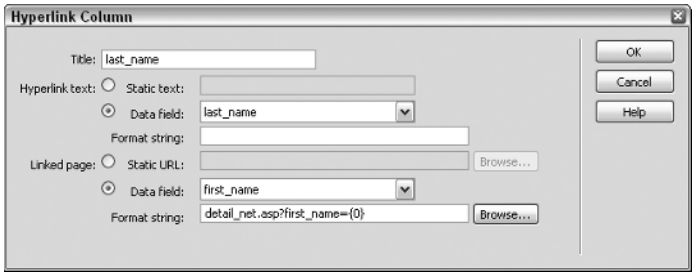

Figure 1-12: This Hyperlink Column dialog box uses the `first_name` as the parameter and `detail_net.asp` as the page.

7. **In the Linked Page section, specify the URL for the hyperlink column text.**

   Here are your choices:

   - **Static Text:** Select this radio button if you want to use a generic link for this column. Enter the URL for the link, such as **Details.aspx**.
   - **Data Field:** Select this radio button if you want to add a link for data displayed in the hyperlink column. Then from the drop-down list, select a data field in your dataset. In the example, we selected the `first_name` column, as shown in Figure 1-12.
   - **Format String:** This field is automatically generated and shows the format of the URL link.

   The URL opens the detail page and uniquely identifies the record to display on the detail page.

8. **In the Linked Page section, click the Browse button next to the Format String box.**

9. **Locate and select which detail page you want to display.**

   Note the following when selecting your link page:

   - When you select a detail page, Dreamweaver adds information to the URL that specifies a parameter to use to identify the record.
   - Dreamweaver automatically names this parameter based on your database field name, but you can change the name to something else if you want to.
   - In any case, be sure to note the name of this URL parameter because you need that name when you create the detail page (which you can read about in the following section, "Creating a detail page").
   - Dreamweaver uses a `{0}` placeholder to indicate where it places the value of the unique identifier when someone accesses the page.

10. **Click OK to close the Hyperlink Column dialog box.**

    Figure 1-13 shows the updated DataGrid dialog box for the example.

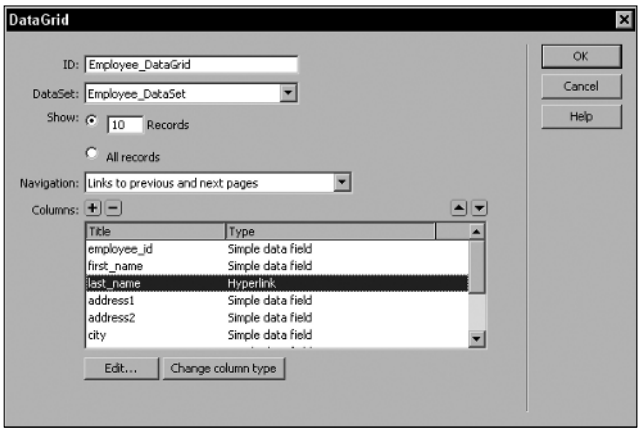

Figure 1-13: The DataGrid dialog box displays the `last_name` column as a Hyperlink type.

11. **Click OK.**

    The DataGrid dialog box closes. The DataGrid on your page is updated.

Creating a detail page

After you create a master page for ASP.NET (as described in the two previous sections), you need to create a detail page to display the record. To do

so, you need to define a dataset for the record and bind its columns to the detail page.

When creating the detail page, you need to know the database column name that you want to reference and the URL parameter that the master page uses to find and display that column's record on the detail page. If you don't remember the URL parameter name, open the master page, go to the Bindings panel, and look under the DataSet listing.

To create an ASP.NET detail page, create a new ASP.NET page in Dreamweaver and follow Steps 2 through 12 in the earlier section, "Building detail pages." Note that although the earlier steps show you how to fill out the Recordset dialog box, the steps are the same for the DataSet dialog box, which you fill out when creating an ASP.NET detail page.

Testing Your Master and Detail Pages

After you create a master and detail page set, you need to test those pages. Follow these steps to preview the pages in a browser:

1. **Open the master page.**
2. **Choose File➪Preview in Browser➪*Your browser type*.**
3. **When Dreamweaver asks you if it's okay to copy the file to the testing site, click OK.**

   Your browser launches with your master page (see Figure 1-14).

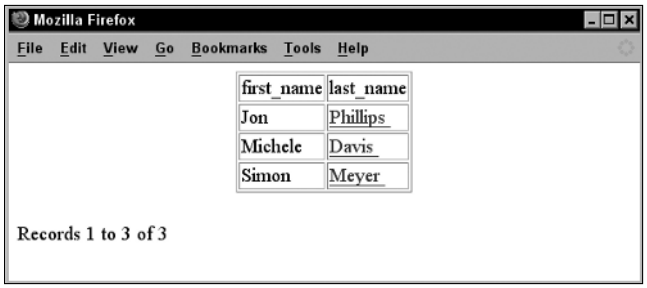

Figure 1-14: The browser displays the master page that uses the data from the database.

4. **Click a hyperlink to view the associated detail page for that record.**

   In the employee example, the linked field is the employee's last name. When you click a link, the browser page changes to expand the record and display the detail page, as shown in Figure 1-15.

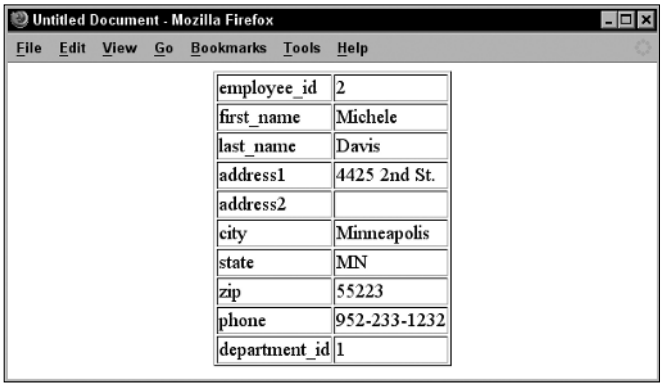

Figure 1-15: The browser displays the detail page that uses the testing employee data.

Be sure that both your master page and detail page transfer to the testing server. If not, when you click a link in the master page, you get a `Page not found` error message.