

Dreamweaver CS3 Project

Introduction:

Students will be given **dream2.zip** file.

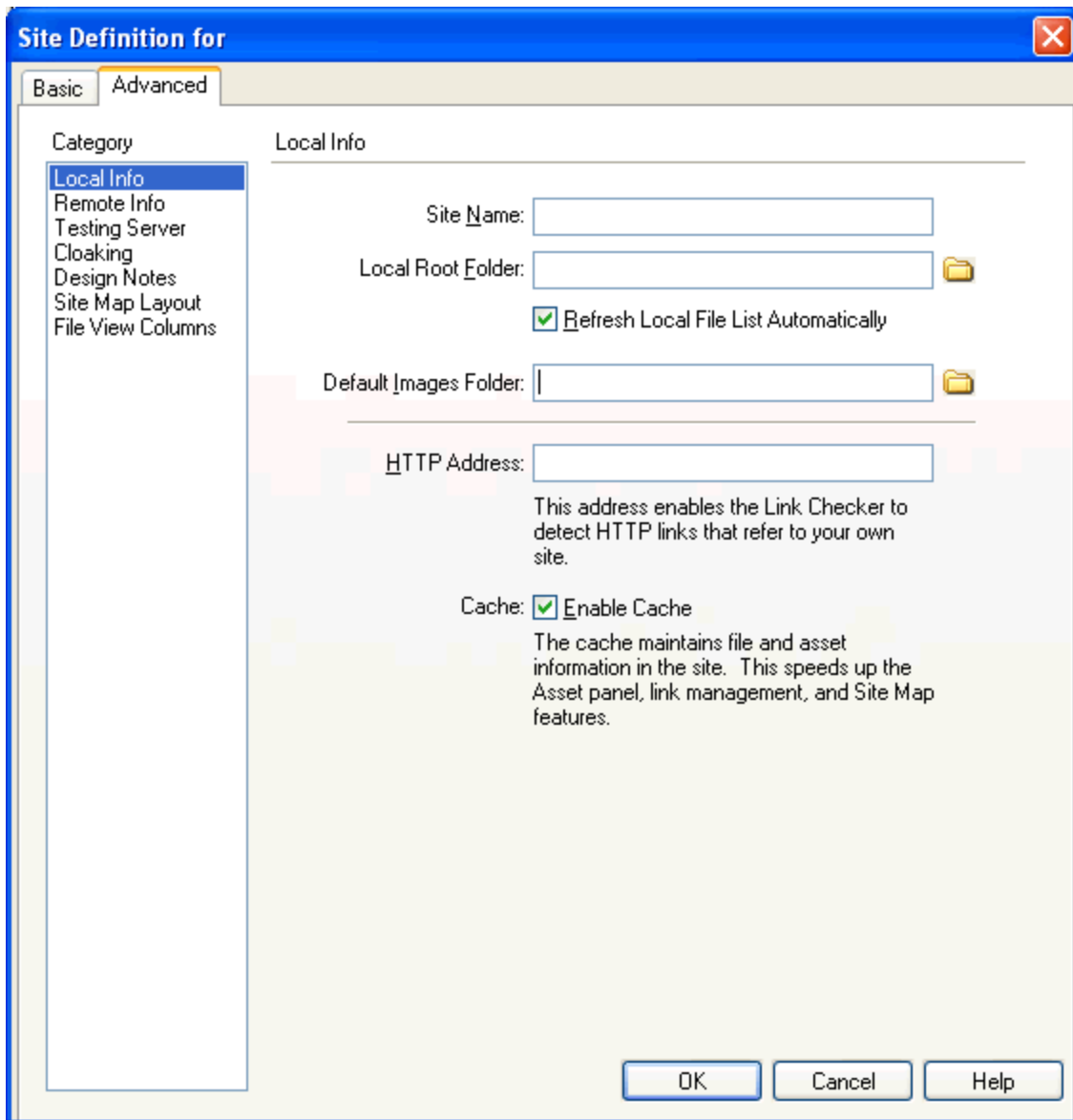
This file has 3 folders as: **Assests, images, and Template.**

Part 1: Setup the Site

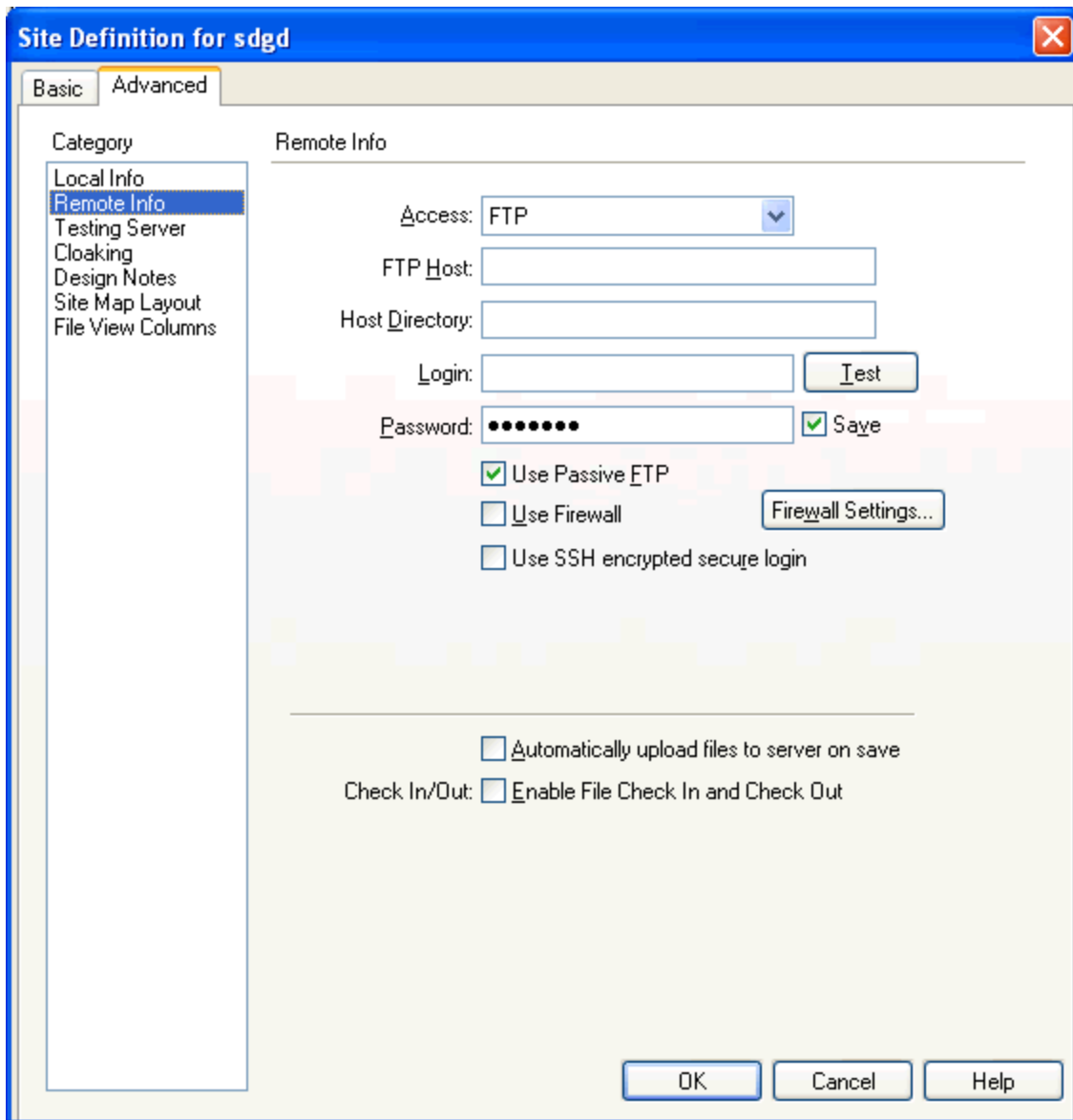
The very first thing you need to do when you are working on a new application is to create a site. Just like a static site, Dreamweaver needs to know where all your assets are, and where your server is as well as its URL.

You will create an application where your customer can view data from the database and can go to detail records for each piece of data. You can administer and change the data for each record in the database, and you can add more records to the database. You will learn how to use templates with dynamic data, bind tables to one another, how to use the Application Objects, and how to use one Recordset with another to select data to display in a form that updates another Recordset.

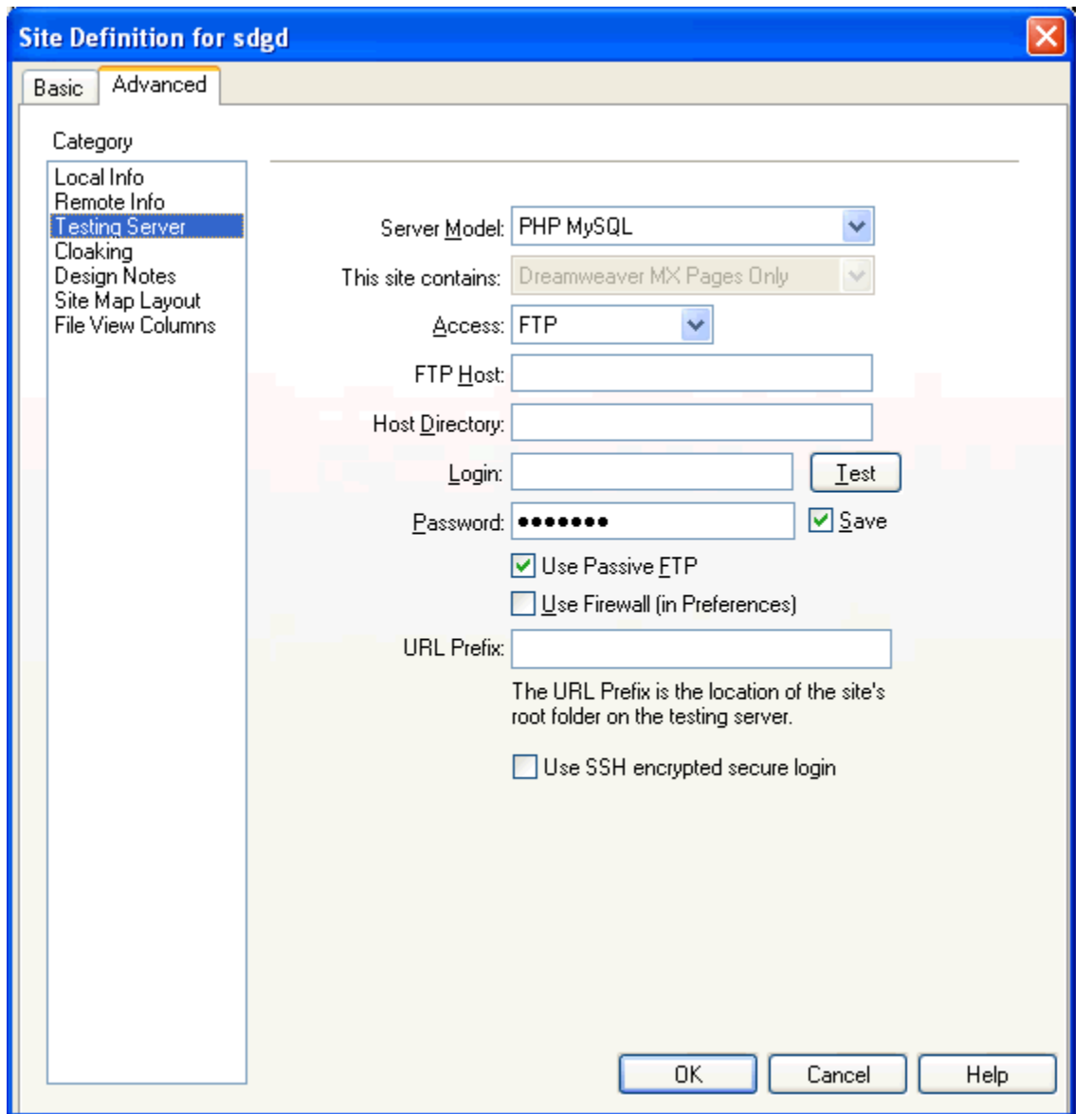
- 1. Create a folder on any drive such as C: or F: or H: drive and name it **dream2****
- 2. Open Dreamweaver and close the blank page.**
- 3. Click Site, New Site**
- 4. Click Advanced Tab**
- 5. Fill out the following categories based on your information sheet:**
 - **Local Files**
 - **Remote Info**
 - **Testing Server**



Local Info	
Site Name	Record Store
Local Root Folder	H:\kim
Default Images Folder	H:\kim\images
HTTP address:	http://maxtips.netfirms.com/



Remote Info	
Access	FTP
FTP Host	
Host Directory	
Login	
Password	
Use Passive FTP	√



Testing Server	
Server Model	
Access	FTP
FTP Host	
Host Directory	
Login	
Password	
Use Passive FTP	√
URL Prefix	

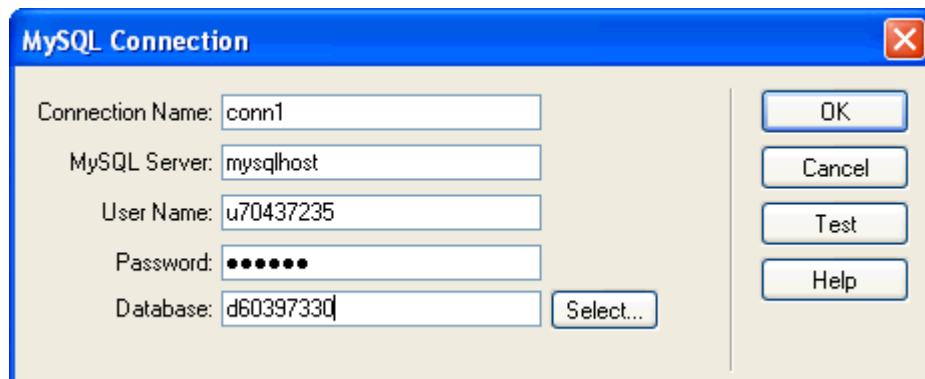
Part 2: Creating the catalog page

Now that you have a site to work in, you can start creating your application.

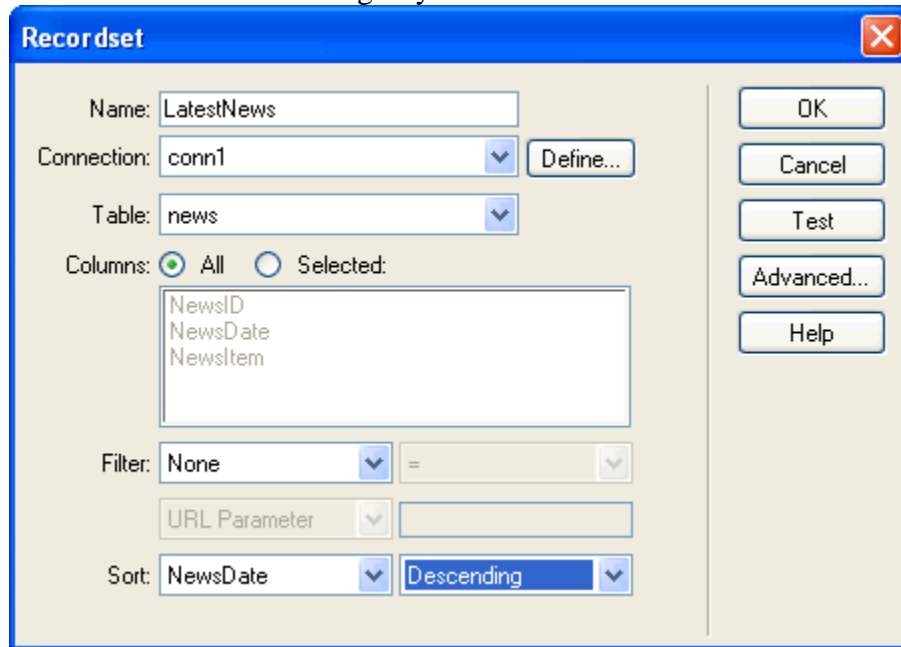
1. From the File menu, select New or press Ctrl-N to get the New Document dialog box.
 2. In the New Document Dialog box, select the Template tab. Dreamweaver lists all the templates for the sites that you have set up. Select the RecordStorePHPStart site in the Templates For column, and you will see one template for recordstore called recordstore_template. Select it, and click the Create button.
 3. When you create a page from a template, Dreamweaver makes a page with all the features of the template ready for you to type into. First, you should save the page and give it a name.
 4. Select File Save. Save the file to your **dream2** folder and name it
 5. **catalog.php.**
-

Part 3: Linking to the database

1. Place your cursor in the editable area of the Title Bar Sidebar, delete the sample text, and type "**Latest News**".
2. Place your cursor in the editable area for the Title Bar Main Area, delete the sample text, and type "**Catalog**".
3. Open the Bindings panel. If it is not open, select Bindings from the Window menu.
4. In the **Bindings** panel, you see some instructions on how to add data to the page. If step 2 is not checked, click the link for "document type", and select PHP from the Choose Document Type dialog box.
5. Click the "+" button to add a Recordset.
The Recordset is the code that requests data from the database, so that it can be displayed on the page.
6. Before you can create a Recordset, you need to create a connection. To do this, click the Define... Button.
7. Click on the **New** button and fill out the dialog as follows.
8. In the MySQL Connection dialog, name the Connection **conn1**.
Fill out the boxes based the following info:



9. Fill out the Recordset dialog as you see below.



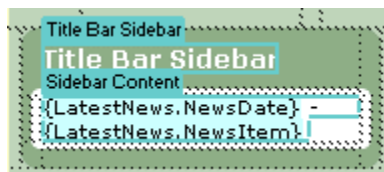
Click **OK** to create the Recordset and insert it in your page.

10. Save your file.


Part 4: Applying dynamic data to the page

Now you have a Recordset on the page, and you can now bind the columns of that Recordset to various places on your page. In this example, you want to bind the data to the Sidebar Content so that the sidebar always shows the latest piece of news.

1. Place your cursor in the Side Bar Content editable area and delete the placeholder content.
2. Make sure that you can see the Bindings tab of the Application panel. If you can't see it, open the Bindings panel from the Window menu.
3. Expand the LatestNews Recordset on the Bindings panel. Each entry represents a column in the database. Each column is a piece of data that you can bind to things on the page.
4. Select the NewsDate item, and drag it to the empty editable area of the Side Bar Content. You should now see a placeholder on your page that indicates that the spot is going to be replaced by the date from the database.
5. Select the NewsItem item, and drag it next to the NewsDate placeholder on your page. Add a dash between the two placeholders to make it easier to see the end of the date and the beginning of the news.



To see your data live, you can preview your page in the browser. However, it is easier to just turn on the Dreamweaver Live Data feature.

To do that, click the Live Data Mode icon in the toolbar. 



6. **Save your page.**
7. **To preview your page in the browser**, you need to upload Connections folder as well as catalog.php
8. Start you Internet Explorer, the type the proper url to see your page:
<http://www.icdf.ca/dream2/catalog.php>

Information:

For the Catalog, you want to display a complete set of information. You need to display the following items: a thumbnail of the cover of the album, the name of the album, the name of the band, the type of music and the price.

There are a couple ways to think of this data. You can think of it as based on the name of the album, or you can think of it as based on the name of the group. If you think of the data as records based on the name of the album, then there are as many records in the database table as there are albums. If you think of the data as based on the names of the group, there would be as many entries in that table as there are groups, and some of those would have more than one album they have released.

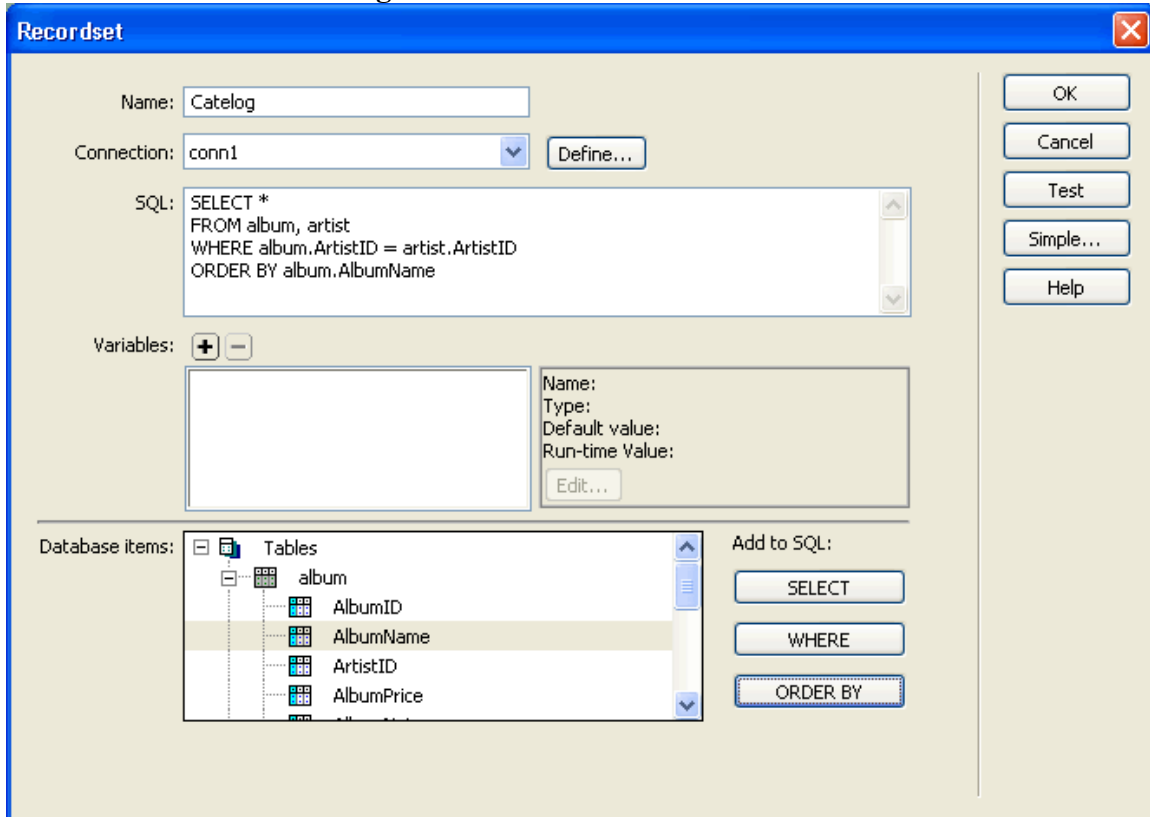
One way of handling that is to join two tables. That way, you can keep track of both the albums and the groups separately and relate the two sets of data. You want to join tables if you wanted to keep parts of your data accessible for change. If, for instance, my favorite band, the Grape People, added a new band member or changed their name, I would want to change that information in one place, not for each of the many albums they have produced.

Another great reason to separate the different sets of information into more than one database table is to keep people from entering information into the database with spelling errors or content errors.

Part 5: For the next Recordset, you are going to join two tables.

1. Open the **Bindings** panel. If it is not open, select Bindings from the Window menu.
2. Click the "+" button and add a Recordset. This time, you want to click the **Advanced** button to get to the advanced version of the Recordset dialog box.

Fill out the Recordset dialog box as follows:



3. Set the name of the Recordset to "**Catalog**". This is how you are going to refer to this Recordset when you want to access the information that this Recordset returns.
4. Set the connection to **conn1**
5. You could simply type in the SQL if you know SQL. Instead, you are going to **build** the Recordset in a couple steps.
6. You don't need any page parameters, so leave that area **blank**.
7. In the **Database** Items area, expand the **Tables** entry. You should see **three** tables there: **Artist, Album, and News**.
8. Select the **Album** table without expanding it, and press the **Select** button. The SQL Recordset now reads:

```
SELECT *  
FROM Album
```

9. Select the **Artist table** without expanding it and click the **Select** button. Now, the SQL Recordset reads:

```
SELECT *  
FROM Album, Artist
```

Now you have selected all the data from both tables, but there is no relationship between the tables, there is no way to use them together.

10. Expand the **Album table**, select the **ArtistID** entry, and press the **Where** button. Now the Recordset reads:

```
SELECT *  
FROM Album, Artist  
WHERE Album.ArtistID
```

11. Expand the **Artist table** and select the **ArtistID** entry and press the **Where** button. Now the SQL field of the Recordset reads:

```
SELECT *  
FROM Album, Artist  
WHERE Album.ArtistID AND Artist.ArtistID
```

**Within the SQL field of the Recordset Dialog,
change the AND to a = sign. (Equal sign)**

12. Now you need to set the sort order. Expand the **Album** table, select the **AlbumName** entry, and click the **Order By** button.

13. Click the **OK** button to insert the Recordset into the page.

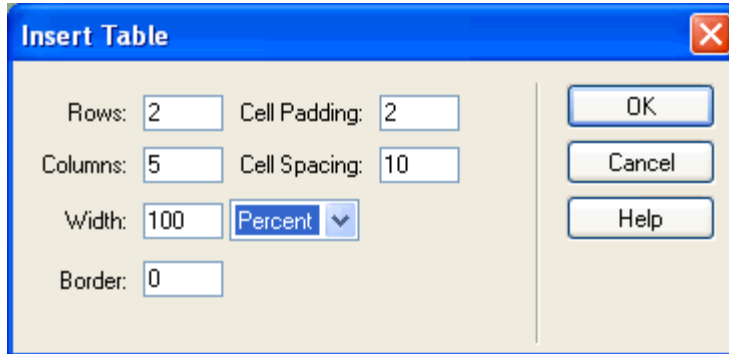
You've created a Recordset that includes all the information from the **Album** and **Artist** tables and joins them where the **ID** of the Artist is the same in both tables.

Part 6: For the next Recordset, you are going to join two tables.

With the first Recordset, we accessed the data directly, and put it in the page. This time, you are going to create an entire table of all the albums at one time, and you will create a detail page at the same time using an application object. This will allow you to click on

an entry and go to a detail page.

1. Click in the **Main Area Content** editable area and **delete** the content. Leave the cursor in the editable area.
2. Insert a table. Click the table icon. Fill out the boxes as below and click **Ok**.



3. In the table that you inserted, leave the first column empty. In the first row second column, type "**Album**". In the third column, type "**Artist**". In the fourth, type "**Style**", and in the last column of the first row, type "**Price**". These are going to be the **table heads** for your data.
4. Select all the cells by dragging across them, and set the type to **bold** by selecting Text > Style > Bold from the menu. Or, click the bold icon on the property inspector.

Now, you are going to bind the data to the cells of the table.

On the **Bindings** panel, expand the **Catalog** Recordset

Drag the **AlbumName** to the second column of the second row.

Drag the **ArtistGroupName** to the third column of the second row.

Drag the **ArtistStyle** to the fourth column of the second row.

Drag the **AlbumPrice** to the fifth column of the second row.

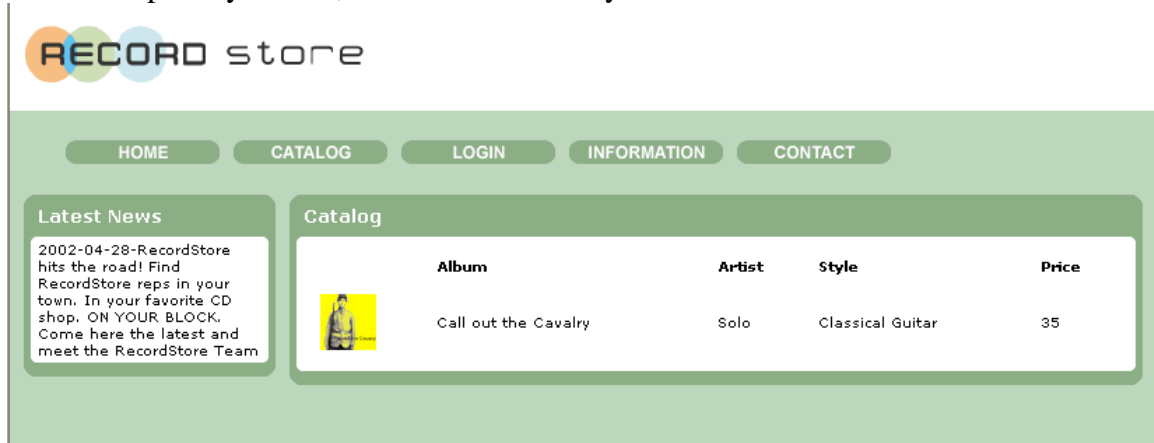
This gets all the placeholders for the data lined up under the headings in the table.

Now that the table is well formatted, you need to make a final change to the table.

You are going to display a thumbnail of the album in the first column, so the user can see the cover of the album they are interested in.

5. Place your cursor in the first cell of the second row of the table and insert an image.
6. At the top of the Insert Image dialog box, there is a radio button that allows you to insert an image from the file system or from the data sources you have set up for your site. Select the Data Sources radio button.
7. Expand the Catalog Recordset, and you will see the same list of fields that you would see in the Bindings panel.
8. Select **AlbumCoverThumbnailURL** from the list and click **OK**. You should see the thumbnails in the first columns of the table. If your live data is not turned on, click the live data button now to turn it on.

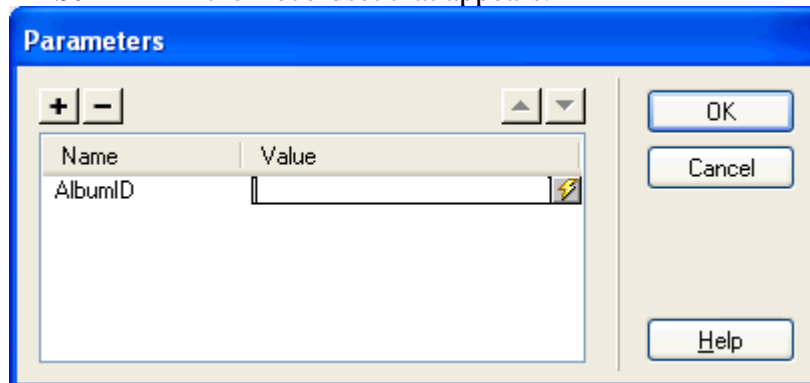
9. Now you want to get all the records showing. To do that, you need to select the row of the table that contains the data by selecting any item in the second row of the table and clicking on the <TR> in the tag selector at the bottom of the window.
10. From the Server Behaviors panel, click the "+" sign, and select the Repeat Region server behavior.
11. In the Repeat Region dialog box, set the Recordset to Catalog and select Show 10 Records at a Time radio button or All records. (Your choice)
12. Click OK.
13. Save and upload your file, then check it out in your browser:

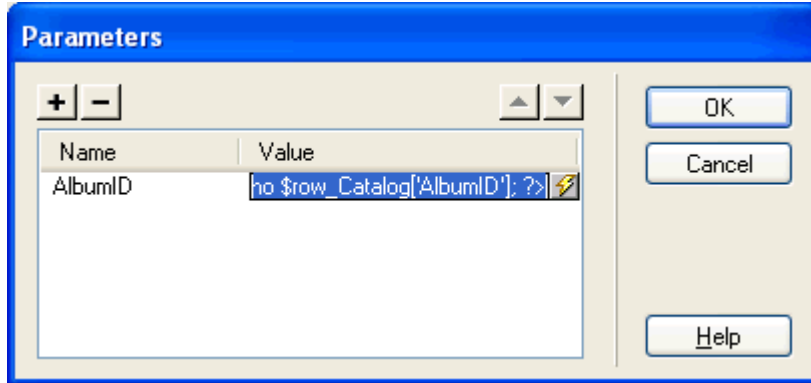


Part 7: Record Store Detail Page

To get to the detail page, you will need a dynamic link that is unique for each record whose detail page you are trying to view.

1. Select the placeholder for the Album Name that you just added in the table. In the property inspector, click the folder icon to the right of the Link drop down menu.
2. In the File name of the Link dialog box, type "**albumdetail.php**". You are going to create that file in the following steps, but it does not exist now.
3. Click the Parameters button, and type "**AlbumID**" in the name, and tab to the value field. Instead of typing in a value, click the lightning bolt icon and select "**AlbumID**" in the Recordset that appears.





What you are doing here is setting the link to pass the **AlbumID** URL parameter to the **albumdetail.php** page. This is the unique key for the table, and it will allow the Recordset on the **albumdetail.php** page to catch the **AlbumID** and show only that data that matches that record.

4. Click **OK** to set the link, Click **OK** again.

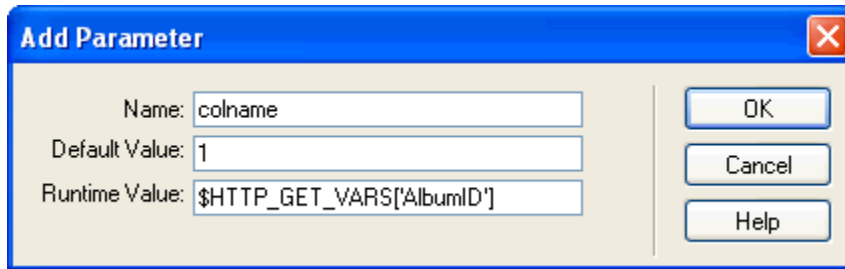
5. **Save your file.**

Part 8: Creating the album detail page

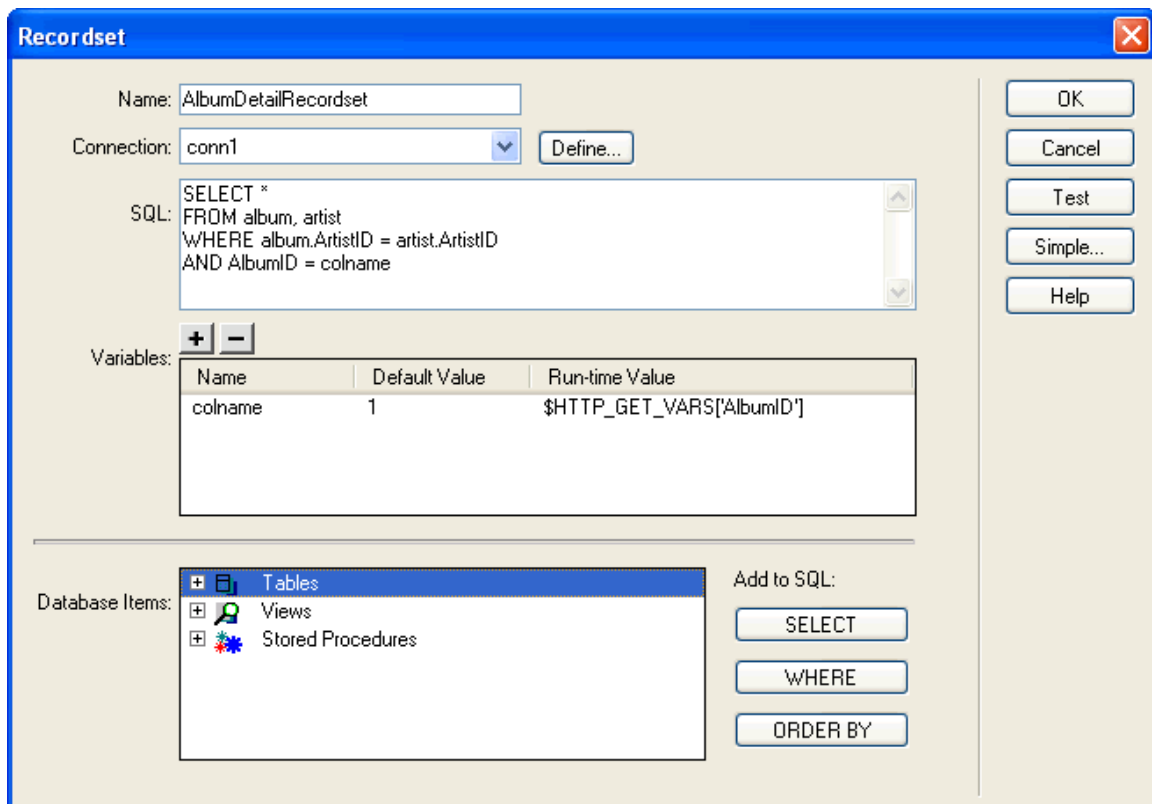
1. Create a new file by selecting File > New. In the New File dialog box, select the General tab. In the General column, select Dynamic Page. In the Dynamic Page column, select PHP and click Create to create a new page.
2. Save the file as albumdetail.php.
3. Insert a new Recordset This Recordset has to get all the records from the Album and the Artist tables, and it has to join them so that the right artist information is associated with the right album. It also has to filter through all those records and find the one record that matches the AlbumID. This record ID will be passed in the URL to the detail page from the catalog.php page when the user clicks on the Album title link.
4. Set the Name of the Recordset to "**AlbumDetailRecordset**".
5. Set the Connection to "**conn1**".
6. Type the following SQL statement in the SQL field:


```
SELECT *
FROM Album, Artist
WHERE Album.ArtistID = Artist.ArtistID
AND AlbumID = colname
```

- This selects all the records from the Artist and the Album tables joined on the ArtistID column. This also sets a requirement for the AlbumID to equal the the AlbumID being passed in from the catalog.php.
- Click on the "+" button in the **Variables** area. Fill out the fields and click OK.



You are going to set a single line of PHP that tells the query to get the value of the AlbumID that was passed to the page from the catalog.php page. Set that up by adding a new Parameter called **colname** with a default of **1** and a Run-time Value of **\$HTTP_GET_VARS['AlbumID']**

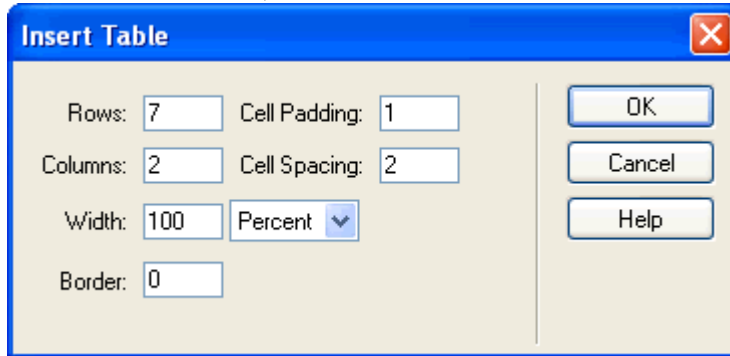


- Click **OK** to finish the Query and insert the Recordset

Now, you have the Recordset ready to catch the variable that the catalog.php is passing. The Recordset knows how to use that variable to get a single record from the database with data combined from two tables. The next step is to build a table to display that

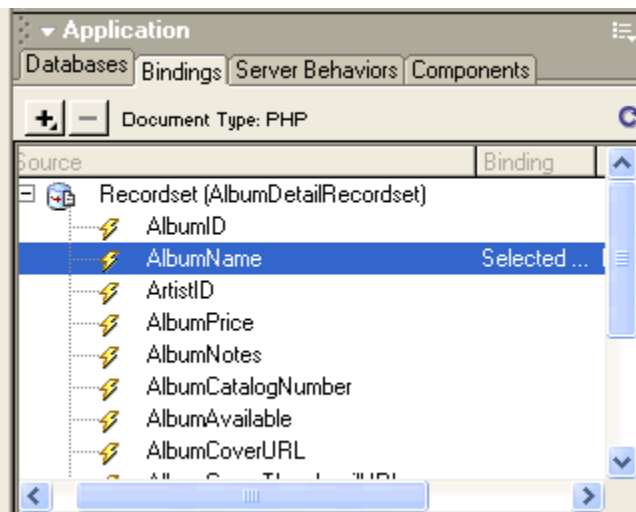
record.

- Click the **table** icon, to create a table with seven rows and two columns as below:



- Type the labels in the first column, and drag the binding data from the recordset from the Bindings panel to the rows in the second column as shown below. Notice that the data is all coming from the one **AlbumDetail Recordset**, however, the Recordset is pulling data from both the artist and the album tables.

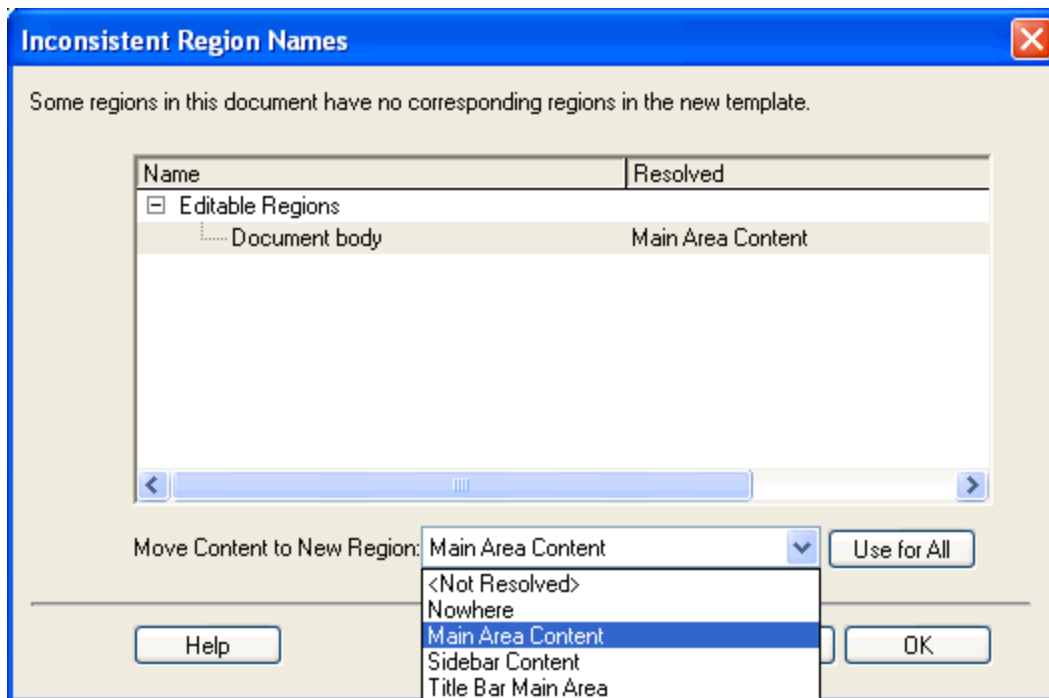
Album	{AlbumDetailRecordset.AlbumName}
Artist	{AlbumDetailRecordset.ArtistGroupName}
Style	{AlbumDetailRecordset.ArtistStyle}
Artist Names	{AlbumDetailRecordset.ArtistNames}
Notes	{AlbumDetailRecordset.AlbumNotes}
Price	{AlbumDetailRecordset.AlbumPrice}



- In the second column on the last row, insert an image, and set the data source equal to the ArtistPhoto in the AlbumDetail Recordset. This will show the picture of the band.

for the selected album.

13. Save your file.
14. Apply the template to this page to make it look like the rest of the site. Open the Assets panel. In case you don't see the Assets panel it is grouped with the Site panel by default. If you need to you can go to the Window menu and select Assets.
15. In the Assets panel, click the icon second from the bottom. If you leave your cursor over the image, you should see a tool tip identifying it as Templates. Last time you used a template, you did it from the New File dialog box. This time, you are going to apply an existing template to an existing page.
16. Highlight the icon for the **recordstore_template**, and click the Apply button at the bottom of the panel.
17. The Inconsistent Region Names dialog box appears. Here, you can select areas of your page to flow into a specific area of a template. This is useful if you are going to apply templates to documents that were built with other templates, and the names of the editable regions are not the same.



18. For the Document body of the **albumdetail.php** page, select the Move Content to New Region drop down, select Main Area Content, and click OK.
19. Save and upload this page. Go to your browser and check out your work.