



4

Working with Databases

IN THIS CHAPTER, YOU LEARN THE BASICS of databases, including how they work, how ColdFusion interfaces with a database, and how to create a ColdFusion datasource. If you've been hesitant to try application servers because you're unfamiliar with databases, you should read this chapter.

Understanding Databases

In one form or another, you use databases every day. If you visit any popular web site, you're using a database through a web browser. If you use an ATM, you're using a database. If you use an automated call system to check the arrival time of a flight, you're using a database.

In short, databases are integral to all of our lives because of the enormous amount of information we access on a daily basis. Whatever form they take, databases store related information in an organized manner. This book uses a database installed with ColdFusion. To find the database, browse to the db folder of the CFusionMX directory, such as C:\CFusionMX\db, and find the cfexamples.mdb file.

If you have Microsoft Access installed, you can view the contents of the database file. As you will see as you explore the cfexamples.mdb file, databases

store information in tables, which contain rows and columns. As shown in Figure 4.1, the `tblEmployees` table contains a row for each employee. Columns separate the rows by information category, such as `FirstName`, `LastName`, and `Title`. In the rows, individual pieces of data are stored in fields.

DeptIDFK	EmployeeID	FirstName	LastName	Title	IsTemp	Email	Phone
B1AE73F-7D7	74DD66E-B0C	Raymond	Camden	Engineer	No	demodata@mai	555-1001
B1AE73F-7D7	74E0C347-B0D	Jacob	Camden	CIO	No	demodata@mai	555-1000
B1AE741-7D7	74E25C48-B0D	Alice	Bluetooth	Marketing Dron	No	demodata@mai	555-1010
B1AE742-7D7	74E2EF23-B0D	James	Nond	Money Person	Yes	demodata@mai	555-1011
B1AE73F-7D7	74E3E426-B0D	Luke	Skywriter	Engineer	No	demodata@mai	555-1012
B1AE73F-7D7	74E4AF1B-B0D	Rob	Brooks-Bilson	Engineer	No	demodata@mai	555-0001
B1AE73F-7D7	74E51567-B0D	Alexson	Boudreaux	Engineer	No	demodata@mai	555-0002
B1AE73F-7D7	74E591E3-B0D	Mel	Landry	Engineer	No	demodata@mai	555-003
B1AE742-7D7	74E738E2-B0D	Mike	Andler	Expenses	No	demodata@mai	555-0010
B1AE742-7D7	74E7B740-B0D	James	Moon	Income Director	No	demodata@mai	555-2000
B1AE743-7D7	74EC23DD-B0C	Sally	Oakenfold	Benefits Manag	No	demodata@mai	555-2001
B1AE743-7D7	74EE9276-B0D	Eve	Ralph	Pizza Provider	No	demodata@mai	555-2002
B1AE743-7D7	74EF72F9-B0D	Siouxie	Smith	Vice President	No	demodata@mai	555-2003
B1AE741-7D7	75039620-B0D	Lee	Alister	Marketing Dron	No	demodata@mai	555-2004
B1AE741-7D7	750421DE-B0D	Ford	Perfect	Marketing Dron	No	demodata@mai	555-2005
B1AE740-7D7	750771AF-B0D	Nathan	Turtledove	Web Gnu	No	demodata@mai	555-2006
B1AE740-7D7	750906A9-B0D	Joel	Canary	Web Gnu	No	demodata@mai	555-2007
B1AE740-7D7	750A6158-B0D	Alicia	Parrot	Web Gnu	No	demodata@mai	555-2008
B1AE740-7D7	750B0E0C-B0C	Ben	Jacobsen	Web Gnu	No	demodata@mai	555-2009
B1AE741-7D7	750B0C6F1-B0D	Claudia	Smith	Sales Drone	No	demodata@mai	555-2012
B1AE73F-7D7	B1AE744-7D7	Stephen	Cheng	Vice President	No	demodata@mai	555-1002
B1AE740-7D7	B1AE745-7D7	Joe	Benny	Intern	No	demodata@mai	555-1005
B1AE741-7D7	B1AE746-7D7	Adam	Lipinsky	Director	No	demodata@mai	555-1236
B1AE742-7D7	B1AE747-7D7	Lynne	Teague	Penny Counter	Yes	demodata@mai	555-1003
B1AE743-7D7	B1AE748-7D7	Victoria	Gilson	Paycheck Hand	No	demodata@mai	555-1238
B1AE740-7D7	B1AE749-7D7	Charles	Reiff	Manager	No	demodata@mai	555-1239

Figure 4.1 The `tblEmployee` table of the `cfexamples` database contains information related to employees. Notice that rows and columns organize the table.

Using Database Data Types

For storage efficiency, each column is assigned a data type, such as text, number, date/time, currency, and so on. Just like programming languages, databases use data types to maximize memory usage. Although you could store most information entered in a web browser as characters (text), using different data types increases database performance and promotes data validity.

The `tblEmployees` database table contains a variety of data types. For example, the `FirstName` and `LastName` columns are text data types. The `IsTemp` column is a Boolean (yes or no) data type, and the `StartDate` column is a date/time data type. Data types also restrict the information to the specified data type.

Understanding Relational Databases

You probably noticed that the `cfexamples` database contains multiple tables, including `tblDepartments`, `tblEmployees`, and `tblParks`. That's because the `cfexamples` database is considered a relational database, meaning that information in multiple tables is related, and the tables have relationships with each other.

Relationships represent links between tables. Usually, primary and foreign keys define relationships. Primary keys, such as the `EmployeeID` column in the `tblEmployees` table, identify the record, in this case an employee. The data stored in the primary key field of a record must be unique because other database tables will use this value to identify an employee. Don't worry, most database applications can automatically generate primary keys for you.

A foreign key lets you represent a primary key in another table. For example, the `DeptIDFK` column in the `tblEmployees` table contains foreign keys for the primary keys in the `DeptID` column of the `tblDepartments` table. Figure 4.2 shows the relationships among the tables of the `cfexamples` database.

Database Naming

A set of informal database-naming conventions have developed over time. For example, the "tbl" preface to a name indicates that the name refers to a database table. If a column contains primary keys, the column name is prefaced by "PK." If a column contains foreign keys, the column name is suffixed by "FK."



Figure 4.2 You can use Microsoft Access's Relationship feature to view a graphical representation of the relationships between tables. Notice that the links connect primary keys to foreign keys.

Selecting a Database Vendor

Usually, you don't get to choose the database vendor because ColdFusion is frequently used to put existing databases on the web. If you do get the chance to select a database vendor, you must carefully weigh the project requirements against the cost of purchase and maintenance.

If your project consists of creating a database for a company intranet, Microsoft Access is a popular choice, if for no other reason than most companies already own a license for Access through Microsoft Office. However, as a shared-file or desktop database, Access has some significant drawbacks, such as scalability and possible data corruption.

Client/server database programs, such as Microsoft SQL Server or Oracle 9i, provide robust, scalable enterprise database solutions. Client/server databases tightly control access to database information, offer excellent performance, and provide multiple safeguards against data corruption and failure. At the same time, client/server database systems are very expensive and are more difficult to set up and maintain. As an alternative, check out open-source databases, such as MySQL and PostgreSQL.

ColdFusion supports the majority of popular database programs used today, including Microsoft Access and SQL Server, Oracle, Informix, Sybase, IBM DB2, and MySQL. For a complete list of database programs that ColdFusion supports, check the Macromedia web site (www.macromedia.com).

Understanding ColdFusion Data Sources

For ColdFusion to interface with a database, you must create a ColdFusion data source. A data source is a connection to a database management system or database file. Just like a Windows data source, a ColdFusion data source represents a kind of ghost copy of the real database. Because ColdFusion runs as a Java application, it interfaces with a system database through Java Database Connectivity (JDBC), a Java technology for standardizing access to databases.

JDBC uses database drivers to communicate directly with the database. JDBC drivers are specifically designed for a particular database system, so, for example, there is an Oracle driver and a Sybase driver. As Figure 4.3 shows, JDBC provides a level of abstraction that lets you work with multiple databases.

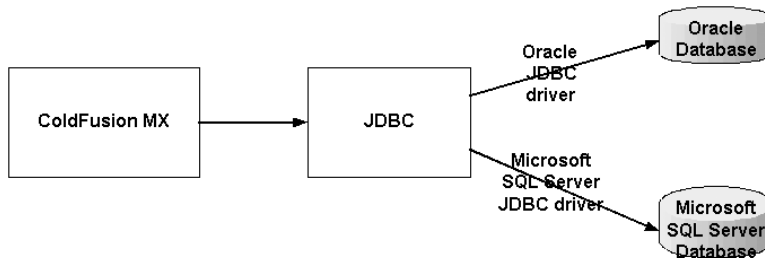


Figure 4.3 ColdFusion uses JDBC, a Java technology, to connect to different databases. JDBC drivers provide direct connections to a specific database type.

The Windows operating system uses Open Database Connectivity (ODBC) for database connections. Because Microsoft Access is a Windows-based application, it uses ODBC as well. If you are running ColdFusion on Windows, you must register the data sources in Windows first before you can use them in ColdFusion. For example, to open the Data Source Administrator in Windows 2000, select the Start menu, Settings, Control Panel, Administrative Tools, (ODBC) Data Sources.

Figure 4.4 shows the ODBC Data Source Administrator.

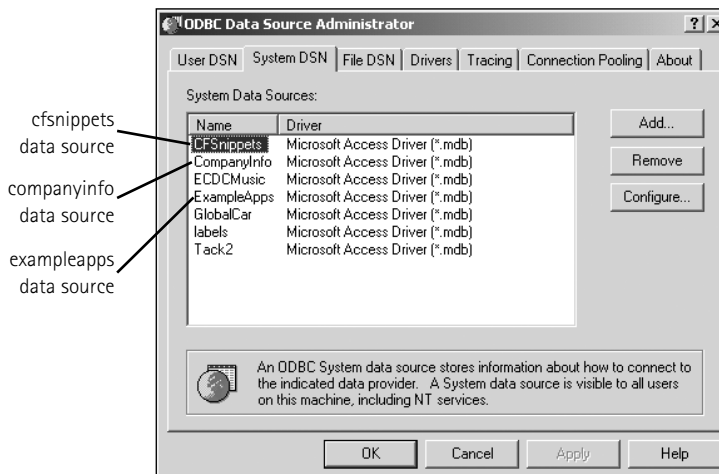


Figure 4.4 In Windows, the ODBC Data Source Administrator lets you review, modify, and create new ODBC data sources.

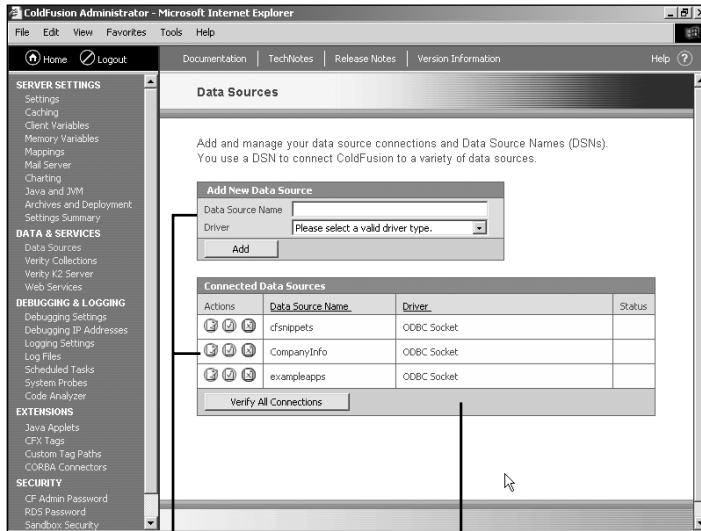
Notice that when you installed ColdFusion, three ODBC data sources were created: CFSnippets, ExampleApps (used in this book), and CompanyInfo.

You might be wondering how ColdFusion, which uses JDBC, works with an ODBC data source like Microsoft Access. ColdFusion contains a database driver named ODBC Socket. The ODBC Socket database driver lets JDBC interface with ODBC to simulate a JDBC connection.

Creating a ColdFusion Data Source

To work with ColdFusion data sources, you use the Data Sources page in the ColdFusion Administrator. You can open the ColdFusion Administrator in Windows by selecting Start, Programs, ColdFusion MX, Administrator. Once you log into the Administrator, click the Data Sources link in the left navigation bar.

In the Data Sources page, shown in Figure 4.5, you see the registered data sources and the driver used for each. To see the details of a data source, click on the Edit icon. This book uses the exampleapps data source, so click on its Edit icon. In the ODBC Socket page that appears, as shown in Figure 4.6, you see the details of the data source, such as the data source name, the database driver, and a description.



Create new data source

Registered data sources

Figure 4.5 The ColdFusion Administrator’s Data Sources page contains the available data sources and the capability to create new data sources.

The screenshot shows the 'ODBC Socket Data Source: exampleapps' configuration window. It is divided into two sections. The top section contains:

- CF Data Source Name: exampleapps
- ODBC DSN: ExampleApps
- Trusted Connection:
- Description: [Empty text box]

 Below this section are buttons for 'Hide Advanced Settings', 'Submit', and 'Cancel'. The bottom section, which is expanded, contains:

- Username: system
- Password: [Empty text box]
- Connection String: [Empty text box]
- Limit Connections: Restrict connections to: [Empty text box]
- Maintain Connections: -- Maintain connections across client requests.
- Timeout (min): 20 Interval (min): 7
- Disable Connections: -- Suspend all client connections.
- Login Timeout (sec): 30
- CLOB: -- Enable long text retrieval (CLOB).

Figure 4.6 The ODBC Socket page lets you see the details of an ODBC Socket data source, such as the ExampleApps data source. Click the Show Advanced Settings button to reveal additional settings.

Although this book uses the ExampleApps data source, the following procedure creates a new data source for the cfexamples.mdb file:

1. In the Data Sources page of the ColdFusion Administrator, enter a name for the data source, such as myDataSource, in the Data Source Name text box. The name that you enter here will be the name used in Dreamweaver and your CFML code.

In the Driver menu, select a database driver. Because you are using the cfexamples.mdb file (Microsoft Access), select the Microsoft Access driver. Notice that the ODBC socket driver was not used. The Microsoft Access driver in the Data Sources page is a JDBC driver, which provides better performance and stability than the ODBC socket driver.

Click the Add button.

2. In the Microsoft Access page that appears, as shown in Figure 4.7, click the Browse Server button to the right of the Database File text box. In the Browse Server page, shown in Figure 4.8, browse to the cfexamples.mdb file in the CFusionMX\db directory. Click the Apply button.
3. Make sure the Use Default Username check box is selected, and enter a short description in the Description text box.
4. Click the Submit button.

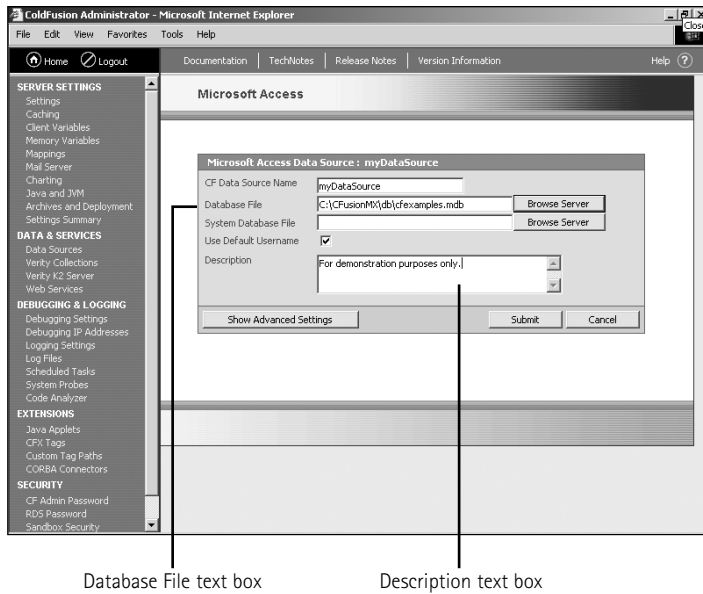


Figure 4.7 The Microsoft Access page lets you select the database file as well as enter a description for the data source. If you need additional options, such as security login information, click the Advanced Settings button.

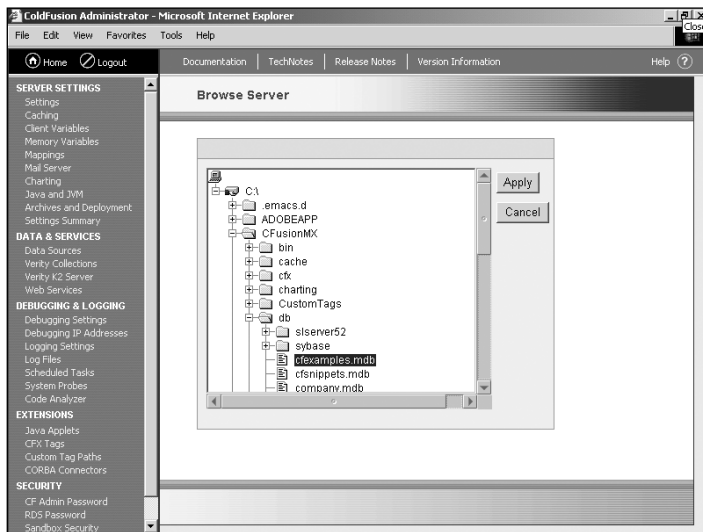


Figure 4.8 The Browse Server page lets you navigate your file system to find the database file. After you click the Apply button, the file path appears in the Database File text box.

Duplicate Data Sources

Avoid creating duplicate data sources for the same database. Duplicate data sources will likely cause confusion or mistakes later.

In the Connected Data Sources table on the Data Sources page, shown in Figure 4.9, you can do more than just see the available data sources. Using the Edit, Verify, and Delete icons in the Actions columns, you can manage your data sources easily. When you verify a data source to make sure it is functioning properly, OK appears in the Status column.

You can also see ColdFusion data sources in the Databases panel of Dreamweaver MX. As shown in Figure 4.9, the Databases panel displays the same list of databases as the Data Sources page in the ColdFusion Administrator. You can also expand and collapse the databases to reveal tables and columns.

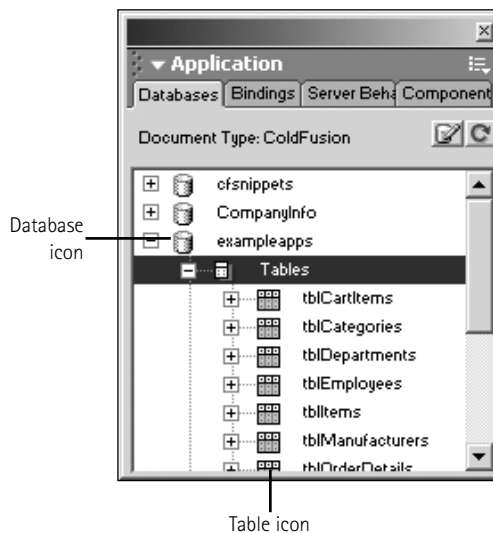


Figure 4.9 The Databases panel of the Application panel group lets you see the registered ColdFusion data sources, their tables, and their table columns.

Querying Databases with SQL

Although storing information is very useful, the true power of databases comes from their capabilities to accept queries and return specific records in a certain order. Queries consist of a communication sent from a program to a database

that specifies how a database should search its records. The *de facto* query language supported by almost every database vendor is Structured Query Language (SQL). The database takes the results of the query and creates a recordset, which contains all the records that match the criteria of the SQL query, and passes the recordset back to the program that requested the query.

Understanding SQL

SQL is a query language that uses basic statement syntax. In general, SQL statements consist of the following types:

- **SELECT**—List of one or more table column names. Must be used with a **FROM** statement.
- **FROM**—List of one or more table names. Must be used with a **SELECT** statement.
- **WHERE**—A comparison statement to filter the query created in the **SELECT** and **FROM** statements.
- **ORDER BY**—List of one or more column names to sort by. You can also use **ASC** (ascending) or **DESC** (descending) to specify which way to sort the records.

You use **SELECT** and **FROM** statements to specify one or more table columns from one or more database tables. The following example selects the `FirstName` column from the `tblEmployees` table:

```
<cfquery name="Recordset1" datasource="exampleapps">
    SELECT FirstName
    FROM tblEmployees
</cfquery>
```

As you can see, SQL can be very simple. To select more than one table column, simply add a comma and the column name, as the following example shows:

```
<cfquery name="Recordset2" datasource="exampleapps">
    SELECT FirstName, LastName, Email
    FROM tblEmployees
</cfquery>
```

To select all columns in the `tblEmployees` table, use the asterisk (*), as the following example shows:

```
<cfquery name="Recordset3" datasource="exampleapps">
    SELECT *
    FROM tblEmployees
</cfquery>
```

To filter a query, you use the **WHERE** statement to filter the records using a comparison statement, such as comparing a record to a value or another record.

The following example compares the `ItemCost` column to a numeric value:

```
<cfquery name="Recordset3" datasource="exampleapps">
    SELECT ItemCost, ItemName
    FROM tblItems
    WHERE ItemCost > 500
</cfquery>
```

If the value in an `ItemCost` column cell is greater than the value, the record is kept. If not, the record is removed. If you want to compare against text, you use single quotes (‘’) to surround the comparison value, such as the following example:

```
<cfquery name="Recordset5" datasource="exampleapps">
    SELECT LastName
    FROM tblEmployees
    WHERE LastName = 'Moon'
</cfquery>
```

If you want to compare column values in one table to column values in another, you simply add the table name to the `FROM` statement and preface the column names in the `SELECT` statement with their table name, as the following example shows:

```
<cfquery name="Recordset6" datasource="exampleapps">
    SELECT tblEmployees.FirstName, tblEmployees.LastName,
    tblEmployees.Email, tblEmployees.DeptIDFK, tblDepartments.DepartmentID,
    tblDepartments.DepartmentName
    FROM tblEmployees, tblDepartments
    WHERE tblEmployees.DeptIDFK = tblDepartments.DepartmentID
    ORDER BY tblEmployees.LastName ASC
</cfquery>
```

This SQL statement associates a department name with each employee. The `tblEmployees` table does not contain the department name, but does include a foreign key to the `tblDepartments` column. By prefixing the column names in the `SELECT` statement with the table name, you identify the column with a particular table, thereby avoiding errors.

The `WHERE` clause checks for equivalence between the `DeptIDFK` column in the `tblEmployees` table, which contains foreign keys, to the `DepartmentID` column in the `tblDepartments` table. Notice that the `LastName` column is sorted by the `ORDER BY` statement.

For more information on displaying or manipulating recordsets in ColdFusion pages, see Chapter 6, “Creating Pages with Dynamic Elements” and Chapter 8, “Displaying Records in a Dynamic Table.”

Note

All SQL contains `SELECT` and `FROM` statements, but you can also use other programming language constructs, such as the `AND`, `OR`, and `EXISTS` operators, or evaluation precedence.

Using Dreamweaver to Build Query Statements

In ColdFusion, you use SQL inside a `CFQUERY` statement. The `CFQUERY` tag supplies a name for the recordset created by the SQL statement as well as the ColdFusion data source name for the query. As the following example shows, the `CFQUERY` tag encapsulates your SQL:

```
<cfquery name="lastNameQuery" datasource="exampleapps">
    SELECT LastName
    FROM tblEmployees
</cfquery>
```

As you can see, the `CFQUERY` tag creates a name for the recordset created by the SQL, `lastNameQuery`, and specifies the database to use, `exampleapps`.

In Dreamweaver MX, you use the Recordset dialog box, accessible from the Bindings or Server Behaviors panel in the Application panel group, to create SQL queries. To open the Recordset dialog box, click the plus (+) button in the Bindings or Database panel. In the submenu that appears, select Recordset (Query).

In the Recordset dialog box, shown in Figure 4.10, you build the query using a set of menus and text boxes. In the Name text box, you specify the name for the recordset, which translates to the `name` attribute of the `CFQUERY` tag. In the Data Source menu, you select the ColdFusion data source to use, which translates to the `datasource` attribute of the `CFQUERY` tag. If required, the User Name and Password text boxes let you enter security credentials to the database.

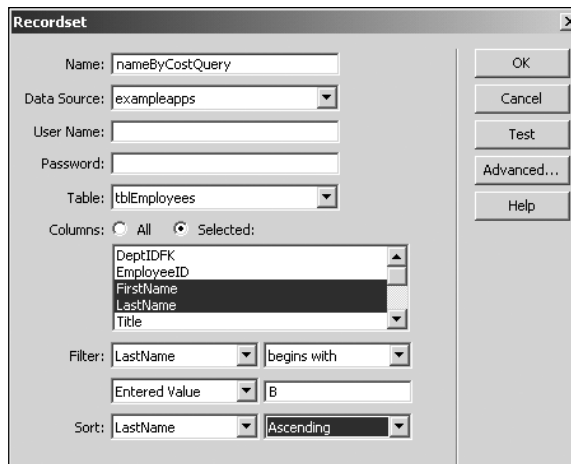


Figure 4.10 The Recordset dialog box lets you create SQL queries using menus and text boxes. You'll notice that what you select in one control dynamically changes the values in other controls.

For more information on displaying recordsets in ColdFusion pages, see Chapter 6.

In the Table menu, you select the table to query, which translates to the `FROM` SQL statement. In the Columns section, you can select some or all of the columns in a table, which translates to the `SELECT` SQL statement. The Filter menus and text boxes let you construct simple `WHERE` statements, and the Sort menu lets you specify a column to sort by.

To test the SQL before you create the recordset, click the Test button. In the Test SQL Statement dialog box, shown in Figure 4.11, Dreamweaver runs the query and displays the results. This feature is especially helpful when developing more complex SQL statements.

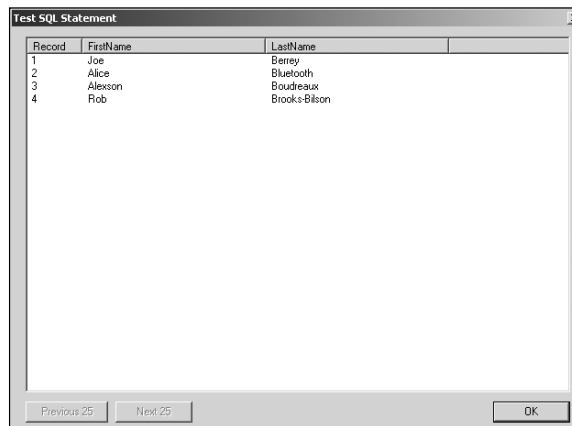


Figure 4.11 The Test SQL Statement dialog box displays the results of your SQL, letting you quickly edit and test SQL statements during development.

To create queries that use multiple tables, you must use the Advanced Recordset dialog box. To switch to the Advanced layout, click the Advanced button in the Recordset dialog box.

In the Advanced Recordset dialog box, shown in Figure 4.12, you can write your SQL by hand in the SQL text box, or you can use the Database Items section and the `SELECT`, `WHERE`, and `ORDER BY` buttons to build the SQL visually.

You can also specify page parameters in your SQL. Parameters are variables passed from external sources, such as other ColdFusion pages, Flash movies, application and session scope variables, and so on. For more information about passing parameters, see Chapter 6.

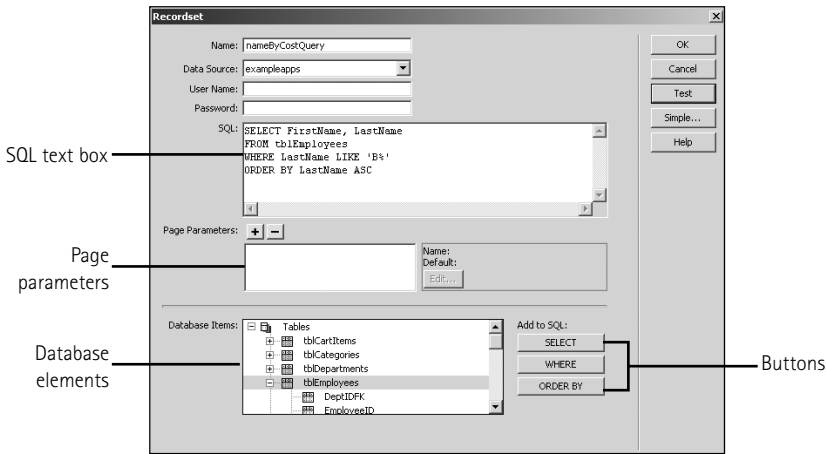


Figure 4.12 The Recordset dialog in Advanced mode provides controls for writing SQL by hand or building SQL visually using the Database Items tree control and the WHERE, SELECT, and ORDER BY buttons.

Summary

In this chapter, you learned the basics of databases, including terminology, SQL, and database structure. In the next chapter, you jump into ColdFusion development by building simple form and action pages.