Orange Coast College
Business Division
CS/CIS Department
Fall 2004
CIS 182
Introduction to Database Concepts

Instructor
Dr. Martha Malaty
Text & Original Presentations
Chapter 14

Web Database Development

In this chapter, you will learn:

• How Internet databases are typically used
• About the architecture of Web-to-database middleware
• How Web-to-database middleware (ColdFusion) is used to integrate databases with the Internet
• What special considerations govern Web database development
Internet Technologies and Databases

- Web database connectivity allows new innovative services that:
  - Allow rapid response to competitive pressures by bringing new services and products to market quickly
  - Increase customer satisfaction through creation of Web-based support services
  - Yield fast and effective information dissemination through universal access from across the street or across the globe
- Many IS departments face the need to create data access architecture, based on Internet standards
- Having a web-based database interface doesn’t negate the database design & implementation issues
# Characteristics and Benefits of Internet Technologies

## Table 14.1 Characteristics and Benefits of Internet Technologies

<table>
<thead>
<tr>
<th>Internet Characteristic</th>
<th>Benefit</th>
</tr>
</thead>
</table>
| Hardware and software independence | Savings in equipment/software acquisition  
|                                 | Runs on most existing equipment  
|                                 | Platform independence and portability  
|                                 | No need for multiple platform development                               |
| Common and simple user interface | Reduced training time and cost  
|                                 | Reduced end-user support cost  
|                                 | No need for multiple platform development                               |
| Location independence           | Global access through Internet infrastructure  
|                                 | Reduced requirements (and costs!) for dedicated connections              |
| Rapid development at manageable costs | Multiple development tools available  
|                                 | Plug and play development tools (open standards)  
|                                 | Development is more interactive  
|                                 | Reduced development times  
|                                 | Tools are relatively inexpensive  
|                                 | Free client access tools (Web browsers)  
|                                 | Low entry costs—free Web servers are often available  
|                                 | Reduced costs of maintaining private networks  
|                                 | Distributed processing and scalability using multiple servers            |
Typical Uses of Internet Databases

• Share information that businesses are willing to make public
  – Shared information is likely to be based on non-critical data
    • Example: EDGAR
      – Holds financial information about businesses
• Provide marketing and customer support
• Provide corporate information to users inside the company (intranets)
• Provide relevant information to users of another company with which the company does business (extranets)
• Develop applications
### A Sample of Applications That Use Web-Database Technology

<table>
<thead>
<tr>
<th>Type of Application</th>
<th>Internally Available</th>
<th>Externally Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Corporate Information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phone Directories</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Suggestion Boxes</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Manufacturing Information</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Design</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Inventory Costing</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Project Tracking</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td><strong>Sales Information</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Catalogs</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Order Entry</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Customer Support</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Web-to-Database Middleware: Server-side Extensions

• Program that interacts directly with Web server to handle specific types of requests
• Provides its services to the Web server in a way that is totally transparent to the client browser
• Example: ColdFusion
• Web-server:
  – The main hub through which all Internet services are accessed
Web-to-Database Middleware: Scenario of a database query

- End user uses a Web browser to dynamically query a database
- Client browser requests a Web page
- Server receives the request
- Server looks for the page on the hard disk
- Server generates the Web page contents, including the result of the database query
- Server sends the page to the client
- Problem:
  - Neither the Web browser nor the Server knows how to connect & read data from the database
  ⇒ Web server capability must be extended so it can understand & process database requests
  - Done through “Server-side extension”
Server-side Extensions

- A program that interacts with the Web server to handle specific types of requests
- Makes it possible to retrieve & present query results
  - Retrieve data from the database
  - Pass retrieved data to the Web server
Web-to-Database Middleware (ColdFusion)

**Figure 14.1 Web-to-Database Middleware (ColdFusion)**

1. **HTTP Page Request**: The client computer sends an HTTP request to the server computer.
2. **Web Server Receives Request**: The server computer receives the request.
3. **Web-to-Database Middleware**: The middleware receives the request and processes it.
4. **HTML Page**: The middleware generates an HTML page.
5. **Database**: The middleware retrieves data from the database.
6. **Web Server Sends the HTML Formatted Page to the Client**: The server sends the HTML page back to the client.
7. **Script Page**: The server determines if the page contains script language and passes the script to the Web-to-database middleware.
8. **Web-to-database middleware connects to the database and passes the query**: The middleware connects to the database and passes the query.
9. **Database Server Passes the Query Results Back to the Web-to-database Middleware**: The middleware receives the query results.
10. **HTML Page**: The middleware formats the query results in HTML format.
11. **Web-to-Database Middleware**: The middleware processes the query results.
12. **Web Server**: The server sends the formatted page to the client.
13. **Client Computer**: The client computer displays the HTML page.
Web Server Interfaces

- Defines how a Web server communicates with external programs
- Two types
  - Common Gateway Interface (CGI)
  - Application Programming Interface (API)
Common Gateway Interface (CGI)

- Uses script files that perform specific functions based on the client’s parameters passed to the Web server
- Script file written usually in PERL, C++, or Visual Basic
- Script file’s contents are used to connect to the database, retrieve the data, convert it to HTML format, then pass it to the Web server
- Disadvantage:
  - The script file is an external program that is individually executed for each user request
  - Decreases system performance
  - Takes significant CPU & memory resources
  - Language used to create the script can affect system performance
**Application Programming Interface (API)**

- Much more efficient & faster than CGI
- Implemented as shared code or dynamic-link libraries (DDL)
- Treated as part of the Web server program that is dynamically invoked when needed
- Advantages (over CGI):
  - Code is resident in memory & there is no need to run an external program for each request
  - The same API serves all requests
  - Can use a shared connection to the database instead of creating a new one every time
- Disadvantages:
  - API error can bring down the server
Application Programming Interface (API)

• Three well-established APIs
  – Netscape API (NSAPI) for Netscape servers
  – Internet Server API (ISAPI) for Microsoft Windows Web servers
  – Web Site API (WSAPI) for O’Reilly Web Servers
• Regardless of the type of Web server interface used, the Web-to-database middleware program must be able to connect to the database
• Two ways to connect:
  – Open Database Connectivity (ODBC)
  – Object Linking and Embedding Database (OLEDB)
Web Server Interfaces
Open Database Connectivity (ODBC)

- Within the Windows environment, most productivity applications provide “hooks” that are used to access databases (Access, DB2, Oracle, SQL Server...)
- Connectivity is achieved through ODBC middleware
- ODBC allows any Windows application to access data sources using standard SQL
- Databases usually offer “native” database middleware to established the required database connections
- Example: Oracle SQL*Net
Windows Applications Use ODBC to Access Databases

**FIGURE 14.3** Windows Applications Use ODBC to Access Databases

Windows applications have access to multiple services through ODBC. Database vendors provide ODBC database drivers so Windows applications can access their respective databases.
Object Linking and Embedding Database (OLEDB)

- Shared Windows services are implemented as shared code that is dynamically linked into the Windows operating environment
- DDLs are stored as files with “.dll” extensions
- DDL speeds up code load & run times
Web-to-Database Middleware Uses ODBC to Access Databases

**FIGURE 14.4** Web-to-Database Middleware Uses ODBC to Access Databases

1. HTTP page request
2. Web Server Interface (API or CGI)
3. ODBC API
4. ODBC Middleware
5. ODBC Driver
6. HTML Page
7. HTML Page
The Web Browser

- Software that lets users navigate (browse) the Web
- Located in client computer
- End-user interface to the World Wide Web
- Interprets HTML code received from Web server
- Presents different page components in a standard way
Client-Side Extensions

• Client-side extensions
  – Add functionality to Web browser
  – Three general types:
    • Plug-ins
    • Java and JavaScript
    • ActiveX and VBScript
Using a Web-to-Database Production Tool: ColdFusion

- Web application server:
  - Middleware application that expands Web server functionality by linking it to a wide range of services
  - Provides consistent run-time environment for Web applications

- Website for ColdFusion Tutorials:
  - http://www.macromedia.com/support/coldfusion/tutorial_index.html
  - http://webmonkey.wired.com/webmonkey/programming/coldfusion/tutorials/tutorial2.html
  - http://www.sims.berkeley.edu/courses/is213/s99/Readings/cold-fusion-tutorial/tutorial/
Using a Web-to-Database Production Tool: ColdFusion (continued)

• ColdFusion application middleware can be used to:
  – Connect to and query a database from a Web page
  – Present database data in a Web page, using various formats
  – Create dynamic Web search pages
  – Create Web pages to insert, update, and delete database data
  – Define required and optional relationships
  – Define required and optional form fields
  – Enforce referential integrity in form fields
  – Use simple and nested queries and form select fields to represent business rules
How ColdFusion Works

FIGURE 14.5 HOW COLDFUSION WORKS

1. HTTP page request
2. Server computer receives request
3. ColdFusion processes the script page and connects to the requested
4. ColdFusion sends HTML-formatted document to the Web server
5. Web server sends the HTML-formatted page to the client
6. The result of the database query is displayed in HTML format

Other services:
- Database server
- E-mail systems
- Directory systems
- File systems
- COM/DCOM
- Web servers
The Ch14_RobCor Database’s Relational Schema

**FIGURE 14.6** *The Ch14_RobCor Database’s Relational Schema*
A Simple Query Using CFQUERY and CFOUTPUT

**Script 14.1 (rc-1.cfm): A Simple Query Using CFQUERY and CFOUTPUT**

```html
<html>
<head>
<title>Rob & Coronel - ColdFusion Examples</title>
</head>
<body bgcolor="LIGHTBLUE">

<h1><center><b>Simple Query using CFQUERY and CFOUTPUT</b></center></h1>

<p><center><b>(Vertical Output)</b></center></p>

<hr>

<cfoutput>
  Your query returned #venlist.RecordCount# records
</cfoutput>

<cfoutput query="venlist">
  <pre>
  VENDOR_CODE:   #VEN_CODE#
  VENDOR_NAME:   #VEN_NAME#
  CONTACT_PERSON: #VEN_CONTACT_NAME#
  ADDRESS:       #VEN_ADDRESS#
  CITY:          #VEN_CITY#
  STATE:         #VEN_STATE#
  ZIP:           #VEN_ZIP#
  PHONE:         #VEN_PH#
  FAX:           #VEN_FAX#
  E-MAIL:        #VEN_EMAIL#
  CUSTOMER_ID:   #VEN_CUS_ID#
  SUPPORT_ID:    #VEN_SUPPORT_ID#
  SUPPORT_PHONE: #VEN_SUPPORT_PH#
  VENDOR WEB PAGE: #VEN_WEB_PAGE#
  </pre>
</cfoutput>

</body>
</html>
```
The rc-1.cfm Script Output (Vertical Listing)

**Figure 14.7** The rc-1.cfm Script Output (Vertical Listing)

<table>
<thead>
<tr>
<th>VENDOR CODE</th>
<th>CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>VENDOR NAME</td>
<td>Cartridge Family</td>
</tr>
<tr>
<td>CONTACT PERSON</td>
<td>James Young</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>110 Highland Dr.</td>
</tr>
<tr>
<td>CITY</td>
<td>Atlanta</td>
</tr>
<tr>
<td>STATE</td>
<td>GA</td>
</tr>
<tr>
<td>ZIP</td>
<td>77814</td>
</tr>
<tr>
<td>PHONE</td>
<td>4045565420</td>
</tr>
<tr>
<td>FAX</td>
<td>4045565771</td>
</tr>
<tr>
<td>E-MAIL</td>
<td><a href="mailto:Young@catridge.com">Young@catridge.com</a></td>
</tr>
<tr>
<td>CUSTOMER ID</td>
<td>J-900</td>
</tr>
<tr>
<td>SUPPORT ID</td>
<td>100</td>
</tr>
<tr>
<td>SUPPORT PHONE</td>
<td>8006576014</td>
</tr>
<tr>
<td>VENDOR WEB PAGE</td>
<td>catridge.com</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>VENDOR CODE</th>
<th>COEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>VENDOR NAME</td>
<td>Corporate Express</td>
</tr>
<tr>
<td>CONTACT PERSON</td>
<td>Penny Henze</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>508 South Military Trail</td>
</tr>
<tr>
<td>CITY</td>
<td>Deerfield Beach</td>
</tr>
<tr>
<td>STATE</td>
<td>FL</td>
</tr>
<tr>
<td>ZIP</td>
<td>33442</td>
</tr>
<tr>
<td>PHONE</td>
<td>6157264626</td>
</tr>
<tr>
<td>FAX</td>
<td>6152554772</td>
</tr>
<tr>
<td>E-MAIL</td>
<td><a href="mailto:pennychenze@ceo.com">pennychenze@ceo.com</a></td>
</tr>
<tr>
<td>CUSTOMER ID</td>
<td>11M_677</td>
</tr>
<tr>
<td>SUPPORT ID</td>
<td>401</td>
</tr>
<tr>
<td>SUPPORT PHONE</td>
<td>8008328333</td>
</tr>
</tbody>
</table>

Your query returned 2 records.
CFQUERY with Tabular CFOUTPUT

**Script 14.2 (rc-2.cfm): CFQUERY with Tabular CFOUTPUT**

```html
<html>
  <cfquery name="VENDATA" datasource="RobCorol">
    SELECT * FROM VENDOR ORDER BY VEN_CODE
  </cfquery>
  <head>
    <title>Rob & Coronel - ColdFusion Examples</title>
  </head>
  <body bgcolor="LIGHTBLUE">
    <h1>
      <b>&lt;center&gt;Simple Query using CFQUERY and CFOUTPUT&lt;/center&gt;&lt;/b&gt;
      &lt;center&gt;(Horizontal Output)&lt;/center&gt;&lt;/b&gt;
    </h1>
    <p>
      VENDOR CODE  VENDOR NAME  CONTACT PERSON  ADDRESS  CITY  STATE  ZIP  PHONE
    </p>
    <cfoutput query="VENDATA">
      #VEN_CODE#  #VEN_NAME#  #VEN_CONTACT_NAME#  #VEN_ADDRESS#  #VEN_CITY#  #VEN_STATE#  #VEN_ZIP#  #VEN_PH#<br>
    </cfoutput>
  </body>
</html>
```
The rc-2.cfm Script Output (Horizontal Listing)

**FIGURE 14.8** The rc-2.cfm Script Output (Horizontal Listing)

---

### Simple Query using CFQUERY and CFOUTPUT (Horizontal Output)

<table>
<thead>
<tr>
<th>VENDOR CODE</th>
<th>VENDOR NAME</th>
<th>CONTACT PERSON</th>
<th>ADDRESS</th>
<th>CITY</th>
<th>STATE</th>
<th>ZIP</th>
<th>PHONE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CART Cartridge Family</td>
<td>James Young</td>
<td>110 Highland Dr. Atlanta GA 77844 4045507458</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>COEX Corporate Express</td>
<td>Penny Henze</td>
<td>508 South Military Trail Deerfield Beach FL 33442 6157284626</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DELL Dell Computer</td>
<td>Kim Berringer</td>
<td>250 Main St. Round Rock TX 78682 8007815222</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GLOBAL Global Corporation</td>
<td>Michael Fox</td>
<td>4744 Rock Rd. Chicago IL 45787 8004571555</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LSREX Laser Express Company</td>
<td>Nicole Whether</td>
<td>5540 Pine St. Albany NY 41550 5164547110</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---
<HTML>
<HEAD>
<TITLE>Rob & Coronel - ColdFusion Examples</TITLE>
</HEAD>
<BODY BGCOLOR="LIGHTBLUE">
<H1><B>Simple Query Using CFQUERY and CFTABLE</B></H1>
<br/>
<TABLE BGCOLOR="Silver" BORDERCOLOR="Fuchsia" FRAME="BORDER">
<TR><HR/></TR>
<TR>
<CFTABLE QUERY="" VENDATA STARTROW="1" COLSPACING="1" COLHEADERS>
<CFCOL HEADER="" CODE"" WIDTH="8" TEXT="#VEN_CODE#">
<CFCOL HEADER="" VENDOR_NAME"" WIDTH="25" TEXT="#VEN_NAME#">
<CFCOL HEADER="" CONTACT_PERSON"" WIDTH="14" TEXT="#VEN_CONTACT_NAME#">
<CFCOL HEADER="" ADDRESS"" WIDTH="20" TEXT="#VEN_ADDRESS#">
<CFCOL HEADER="" CITY"" WIDTH="10" TEXT="#VEN_CITY#">
<CFCOL HEADER="" STATE"" WIDTH="" TEXT="#VEN_STATE#">
<CFCOL HEADER="" ZIP"" WIDTH="5" TEXT="#VEN_ZIP#">
<CFCOL HEADER="" PHONE"" WIDTH="10" TEXT="#VEN_PH#">
<CFCOL HEADER="" FAX"" WIDTH="10" TEXT="#VEN_FAX#">
<CFCOL HEADER="" E-MAIL"" WIDTH="10" TEXT="#VEN_EMAIL#">
<CFCOL HEADER="" CUSTOMER_ID"" WIDTH="8" TEXT="#VEN_CUS_ID#">
<CFCOL HEADER="" SUPPORT_PHONE"" WIDTH="6" TEXT="#VEN_SUPPORT_ID#">
<CFCOL HEADER="" WEB_PAGE"" WIDTH="14" TEXT="#VEN_WEB_PAGE#">
</CFTABLE>
</TR>
</TABLE>
</BODY>
</HTML>
### The rc-3.cfm Script Output (Formatted Horizontal Listing)

#### Figure 14.9

**Simple Query Using CFQUERY and CFTABLE**

<table>
<thead>
<tr>
<th>CODE</th>
<th>VENDOR_NAME</th>
<th>CONTACT_PERSON</th>
<th>ADDRESS</th>
<th>CITY</th>
<th>ST</th>
<th>ZIP</th>
<th>PHONE</th>
<th>FAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>CART</td>
<td>Cartridge Family</td>
<td>James Young</td>
<td>110 Highland Dr.</td>
<td>Atlanta</td>
<td>GA</td>
<td>73814</td>
<td>4045557458</td>
<td>404528577</td>
</tr>
<tr>
<td>COEX</td>
<td>Corporate Express</td>
<td>Penny Henze</td>
<td>508 South Military T Deerfield</td>
<td>FL</td>
<td>33442</td>
<td>6157264626</td>
<td>615255477</td>
<td></td>
</tr>
<tr>
<td>DELL</td>
<td>Dell Computer</td>
<td>Kim Berringer</td>
<td>250 Main St.</td>
<td>Round Rock</td>
<td>TX</td>
<td>78662</td>
<td>8007815222</td>
<td>800842598</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>Global Corporation</td>
<td>Michael Fox</td>
<td>4744 Rock Rd.</td>
<td>Chicago</td>
<td>IL</td>
<td>45787</td>
<td>8004571555</td>
<td>800787242</td>
</tr>
<tr>
<td>LSREX</td>
<td>Laser Express Company</td>
<td>Nicole Whethe</td>
<td>5540 Pine St.</td>
<td>Albany</td>
<td>NY</td>
<td>41550</td>
<td>5184547110</td>
<td>518575437</td>
</tr>
</tbody>
</table>

---

Dynamic Search Query: Criteria Entry Form

Script 14.4a (rc-4a.cfm): Dynamic Search Query: Criteria Entry Form

```html
<html>
<head>
<title>Rob & Coronel - ColdFusion Examples</title>
</head>
<body bgcolor="lightblue">
<h1>
<center><b>Dynamic Search Query: Criteria Entry Form</b></center>
</h1>
<form action="rc-4b.cfm" method="post">
<table align="center" bgcolor="silver">
<tr>
<td align="right">VEN_CODE</td>
<td>
<input type="text" name="VEN_CODE" size="10" maxlength="10"></td>
</tr>
<tr>
<td align="right">VEN_STATE</td>
<td>
<select name="VEN_STATE" size=1>
<option selected value="ANY">ANY</option>
<option value="STATELIST">STATELIST</option>
</select></td>
</tr>
</table>
</form>
</body>
</html>
```
The rc-4a.cfm Script Output (State Search Criteria Entry Form)

**Figure 14.10** The rc-4a.cfm Script Output (State Search Criteria Entry Form)
The VENDOR Search Results

Script 14.4b (rc-4b.cfm): The VENDOR Search Results

<cfquery name="SearchVendor" data source="RobCor">
   SELECT VEN_CODE, VEN_NAME, VEN_CONTACT_NAME, VEN_ADDRESS, VEN_CITY, VEN_STATE, VEN_PH
   FROM VENDOR
   WHERE 0=0
   <cfif #form.VEN_CODE# is not "">
      AND VENDOR.VEN_CODE LIKE "#form.VEN_CODE#"
   </cfif>
   <cfif #form.VEN_STATE# is not "ANY">
      AND VENDOR.VEN_STATE LIKE "#form.VEN_STATE#"
   </cfif>
   ORDER BY VEN_CODE
</cfquery>

<title>Rob & Coronel - ColdFusion Examples</title>

<body bgcolor="lightblue">
<h1>Vendor Search Results</h1>
<h2>No records were found matching your criteria</h2>
<form action="rc-0.cfm" method="post">
   <input type="submit" value="Main Menu"/>
</form>
</body>
</html>
Figure 14.11 The rc-4b.cfm Script Output (Vendor Search Results: All States)
The Vendor List for the Condition
VEN_STATE = “GA”
Script 14.5a (rc-5a.cfm): Insert Query: Data Entry Script

```
<HTML>
<HEAD>
<TITLE>Rob & Coronel - ColdFusion Examples</TITLE>
<CFQUERY NAME="USRLIST" DATASOURCE="RobCor"> 
SELECT USR_ID, USR_LNAME, USR_FNAME, USR_MNAME 
    FROM USER 
    WHERE USR_ID NOT IN (SELECT USR_ID FROM DEPARTMENT WHERE USR_ID > 0) 
    ORDER BY USR_LNAME, USR_FNAME, USR_MNAME 
</CFQUERY>
</HEAD>
<BODY bgcolor="LIGHTBLUE">

<H1>Insert Query: Data Entry Screen</H1>
<form action="rc-5b.cfm" method="post"> 
<h3>Department Data</h3>  
---- the following code defines required fields ---->
<form action="rc-5b.cfm" method="post"> 
<h3>Department Data</h3>  
---- the following code defines required fields ---->
<textarea name="DEPT_DESC" rows="10" cols="50">Manager : <br>
select user from list --
</textarea>

<INPUT type="hidden" name="DEPT_ID_required" value="You must enter a DEPT_ID">
<INPUT type="hidden" name="DEPT_DESC_required" value="You must enter a description">
<table align="CENTER" bgcolor="Silver">
<tr>
<td></td>
<td>
Department ID: <br>
<INPUT type="text" name="DEPT_ID" size="10" maxLength="10"><br>Description:
<INPUT type="text" name="DEPT_DESC" size="35" maxLength="35"><br>Manager: <br>
<select name="USR_ID">  
<option value=""></option>
<option value="#usrID#">[#usrID#] #usrLNAME#, #usrFNAME#, #usrMNAME#
</select>
</td>
</tr>
</table>

</form>
<form action="rc-0.cfm" method="post"> 
</form>
</body>
</html>
```
The rc-5a.cfm Script Output (Department Data Entry Screen)

FIGURE 14.13 THE RC-5A.CFM SCRIPT OUTPUT (DEPARTMENT DATA ENTRY SCREEN)
The Insert Query Form:
Server-Side Validation Error Message

**FIGURE 14.14** The Insert Query Form: Server-Side Validation Error Message

Form Entries Incomplete or Invalid

One or more problems exist with the data you have entered.

- You must enter a description

Use the Back button on your web browser to return to the previous page and correct the listed problems.
Insert Query: Confirmation Script

**SCRIPT 14.5B (rc-5B.cfm): INSERT QUERY: CONFIRMATION SCRIPT**

```html
<HTML>
<HEAD>
<TITLE>Rob & Coronel - ColdFusion Examples</TITLE>
<!---- inserting record in table ---->
<CFINSERT DATASOURCE="RobCor" TABLENAME="DEPARTMENT">
</HEAD>
<BODY bgcolor="LIGHTBLUE">
<H1>
<CENTER><B>Insert Query: Result Confirmation</B></CENTER></H1>
You have added the following record:
</CFOUTPUT>
<pre>
DEPT_ID : #DEPT_ID#
DEPT_DESC: #DEPT_DESC#
USR_ID   : #USR_ID#
</pre>
</CFOUTPUT>
<form action="rc-0.cfm" method="post">
  <input type="submit" value="Main Menu ">
</form>
</BODY>
</HTML>
```
The rc-5b.cfm Script Output (Department Data Entry Confirmation)

**Figure 14.15** The rc-5b.cfm Script Output (Department Data Entry Confirmation)

User enters data on the department data entry form.
Manager is selected from a drop-down select list.
ColdFusion validates data and inserts the new record in the database.
The Insert Query: ODBC Integrity Violation Error

FIGURE 14.16  THE INSERT QUERY: ODBC INTEGRITY VIOLATION ERROR

Insert Query: Data Entry Screen
Department Data

Trying to add an existing department causes an ODBC database constraint violation error message to be displayed.

Error Occurred While Processing Request

Error Diagnostic Information
ODBC Error Code = 23000 (Integrity constraint violation)

The changes you requested to the table were not successful because they would create duplicate values in the index, primary key, or relationship. Change the data in the field or fields that contain duplicate data, remove the index, or redefine the index to permit duplicate entries and try again.

The error occurred while processing an element with a general identifier of (CFINSERT), occupying document position (5.1) to (5.53).

Date/Time: 08/07/03 15:30:30
Browser: Mozilla/4.0 (compatible; MSIE 6.0; Windows 98; T312461)
Remote Address: 161.45.248.24
HTTP Referrer: http://labnt1.bizlab.mtsu.edu/robcor/rc-5a.cfm
Update Query: Record Selection Script

**SCRIPT 14.6A (rc-6a.cfm):** UPDATE QUERY: RECORD SELECTION SCRIPT

```html
<HTML>
<HEAD>
<TITLE>Rob & Coronel – ColdFusion Examples</TITLE>
</HEAD>
<BODY BGCOLOR="LIGHTBLUE">
<H1>
<CENTER><B>Update Query: Record Selection Screen</B></CENTER>
</H1>
<TABLE ALIGN="CENTER" BGCOLOR="Silver">
<tr VALIGN="TOP">
<td>
<form ACTION="rc-6b.cfm" METHOD="post">
<select NAME="DEPT_ID" SIZE=1>
<option VALUE="#DEPT_ID#">[#DEPT_ID#] - #DEPT_DESC#</option>
</select>
</form>
<input TYPE="hidden" NAME="DEPT_ID_required" VALUE="DEPT_ID is required">
</td>
<td>
<input TYPE="submit" VALUE="Edit ">
</form>
</td>
</tr>
</TABLE>
</BODY>
</HTML>
```
The rc-6a.cfm Script Output (Record Selection Screen)
Update Query: Edit Record Script

```
<html>
<head>
<title>Rob & Coronel - ColdFusion Examples</title>
</head>
<body bgcolor="lightblue">
<h1>Update Query: Edit Record Screen</h1>
<form action="rc-6b.cfm" method="post">
<table align="center" bgcolor="silver" bordercolor="blue">
<tr>
<td><input type="hidden" name="dept_id" value="#deptdata.dept_id#"></td>
</tr>
<tr>
<td>Department ID: <b>&lt;input type="text" name="dept_id" size=35 max_length=35&gt;</b><br>
Description: <b>&lt;input type="text" name="dept_desc" value="#deptdata.dept_desc#" size=35 max_length=35&gt;</b><br>
Manager: <b>&lt;select name="usr_id" size=10&gt; ---- select user from list ------ &lt;/select&gt;</b><br>
</td>
</tr>
<tr><td><input type="submit" value=""/></td></tr>
</table>
</form>
</body>
</html>
```
The rc-6b.cfm Script Output
(Edit Record Screen)

**FIGURE 14.18 The rc-6b.cfm Script Output (Edit Record Screen)**

This form enables the end user to assign a new manager to the transportation department. Note that the existing manager appears as the default selection.
Update Query: Result Confirmation Script

**SCRIPT 14.6C (rc-6c.cfm): UPDATE QUERY: RESULT CONFIRMATION SCRIPT**

```html
<html>
<head>
<title>Rob & Coronel - ColdFusion Examples</title>
<cfupdate datasource="RobCor"tablename="Department"/>
</head>
<body bgcolor="LIGHTBLUE">
<h1><center><b>Update Query: Result Confirmation</b></center></h1>
<cfooutput>
You have successfully updated the following data:
<pre>
DEPARTMENT ID: <b>#DEPT_ID#</b>
DESCRIPTION : <b>#DEPT_DESC#</b>
MANAGER     : <b>#USR_ID#</b>
</pre>
</cfooutput>
<form action="rc-0.cfm" method="post">
  <input type="submit" value="Main Menu ">
</form>
</body>
</html>
```
The rc-6c.cfm Script Output (Update Query Result Confirmation)

**FIGURE 14.19** THE rc-6c.cfm SCRIPT OUTPUT (UPDATE QUERY RESULT CONFIRMATION)
Delete Query: Record Selection Script

```html
<SCRIPT 14.7A (rc-7a.cfm): DELETE QUERY: RECORD SELECTION SCRIPT

1. <HTML>
2.   <HEAD>
3.     <TITLE>Rob & Coronel - ColdFusion Examples</TITLE>
4.     <CFQUERY NAME="Deptlist"  DATASOURCE="RobCor">
5.       SELECT * FROM DEPARTMENT ORDER BY DEPT_ID
6.     </CFQUERY>
7.   </HEAD>
8. <BODY BGCOLOR="LIGHTBLUE">
9.   <H1>
10.  <CENTER><B>Delete Query: Record Selection Screen</B></CENTER></H1>
11. </BODY>
12. <TABLE ALIGN="CENTER" BGCOLOR="Silver">
13.  <TR VALIGN="TOP">
14.  <TD>
15.   <FORM ACTION="rc-7b.cfm" METHOD="post">
16.       <SELECT NAME="DEPT_ID" SIZE=1>
17.         <CFOUTPUT QUERY="Deptlist">
18.           <OPTION VALUE="#DEPT_ID#">[#DEPT_ID#] - #DEPT_DESC#</OPTION>
19.         </CFOUTPUT>
20.       </SELECT>
21.       <INPUT TYPE=HIDDEN NAME="DEPT_ID_required" VALUE="DEPT_ID is required">
22.   </TD>
23.   <TD>
24.     <INPUT TYPE="submit" VALUE="Delete"/>
25.   </TD>
26. </FORM>
27. </TD>
28.   </TR>
29. <FORM ACTION="rc-0.cfm" METHOD="post">
30.     <INPUT TYPE="submit" VALUE="Main Menu">
31. </FORM>
32. </TD>
33. </TR>
34. </TABLE>
35. </BODY>
36. </HTML>
```
The rc-7a.cfm Script Output
(Record Selection Screen)
Delete Query: Show Record Script

```html
<html>
<head>
<title>Rob & Coronel - ColdFusion Examples</title>
</head>
<body bgcolor="LIGHTBLUE">
<form action="rc-7c.cfm" method="post">
  <cfoutput query="DeptData">
  <p>Department ID: #DEPT_ID##BR>
  <p>Description : #DEPT_DESC##BR>
  <p>Manager: #CFIF #DEPTDATA.USR_ID# IS NOT ""##USRdata.USR_LNAME# #USRdata.USR_FNAME##USRdata.USR_MNAME##CFI菲##BR>
  </cfoutput>
</form>
</body>
</html>
```

The rc-7b.cfm Script Output (Show Record Screen)

FIGURE 14.21 THE rc-7b.cfm SCRIPT OUTPUT (SHOW RECORD SCREEN)
Delete Query: Result Confirmation Script

**SCRIPT 14.7C (RC-7C.CFM): DELETE QUERY: RESULT CONFIRMATION SCRIPT**

```html
<HTML>
<HEAD>
<TITLE>Rob & Coronel - ColdFusion Examples</TITLE>
<CFQUERY NAME="DeleteDept" DATASOURCE="RobCor">
    DELETE FROM DEPARTMENT WHERE (DEPT_ID = '#FORM.DEPT_ID#')
</CFQUERY>
</HEAD>
<BODYbgcolor="LIGHTBLUE">
<H1>
< CENTER><B>Delete Query: Result Confirmation</B><CENTER>
</H1>
<CFOUTPUT>
You have successfully deleted the following data
<pre>
DEPARTMENT ID: <B>#DEPT_ID#</B>
DESCRIPTION: <B>#DEPT_DESC#</B>
MANAGER : <B>#USR_ID#</B>
</pre>
</CFOUTPUT>
</FORM ACTION="rc-0.cfm" METHOD="post">
    <INPUT TYPE="submit" VALUE="Main Menu" />
</FORM>
</BODY>
</HTML>
```
The rc-7c.cfm Script Output
(Delete Query Confirmation Screen)
The Delete Record Validation

FIGURE 14.23  THE DELETE RECORD VALIDATION

Delete Query: Record Selection Screen

The end user selects a department for which users exist.

The department cannot be deleted, because there are users assigned to the department.

Delete Query: Show Record Screen

We cannot delete this record because there are dependent users assigned to this department.
Internet Database Systems: Special Considerations

- Internet database systems involve more than just the development of database-enabled Web applications.
- Database systems development requires sound database design and implementation.
- Concurrent database access by multiple heterogeneous clients affects how transactions are defined and managed.
- If database systems are to be developed and managed intelligently, today’s database administrator must understand the Internet-based business environment to successfully cope with the issues that drive the development, use, and management of Web-to-database interfaces.
Summary

• Internet is a worldwide network of computers running the TCP/IP suite of network protocols
• Web server provides standard interfaces such as the Common Gateway Interface (CGI) and the application programming interface (API)
• Database access through the Web is achieved through database-to-middleware software
• ColdFusion is a Web application server that provides database access (among other services) to the Web
• Developing applications for the Internet/intranets/extranets means overcoming limitations of the Web interface