Web Technologies

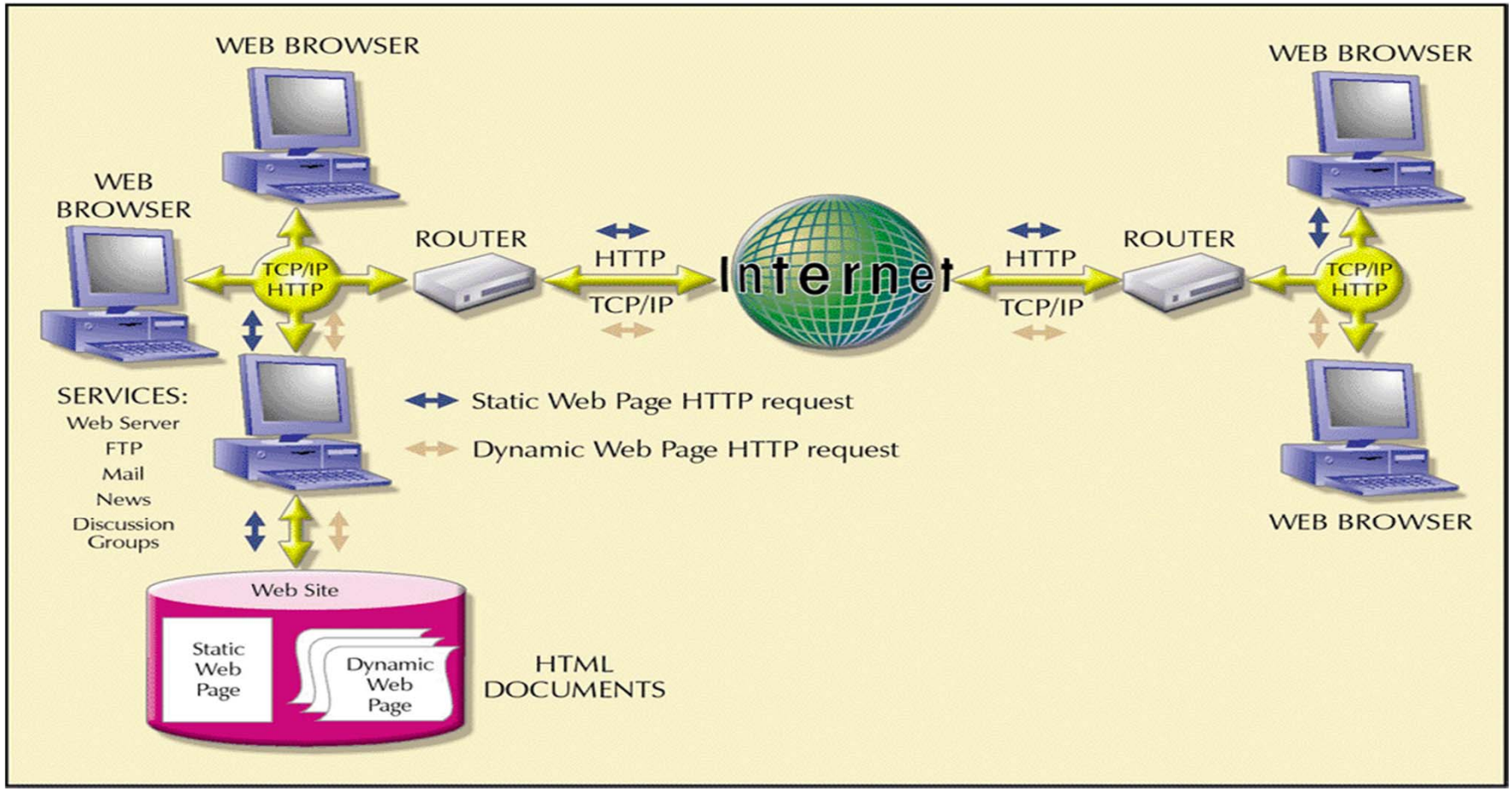
Internet Building Blocks

BUILDING BLOCK/SERVICE	DESCRIPTION
Internet	A worldwide network of networks. The Internet acts as a "super network" that connects thousands of smaller networks across the world. You can think of the Internet as the "highway" on which data travel, as in the phrase "the information superhighway." To connect thousands of heterogeneous networks, the Internet uses a standard network protocol known as *TCP/IP* and devices known as *routers*.
TCP/IP	Transmission Control Protocol/Internet Protocol. The basic network protocol that determines the rules used to create and route "packets" of data between computers in the same or different networks. Each computer connected to the Internet has a unique TCP/IP address. The TCP/IP address is divided into two parts used to identify the network and the computer (or host).
Router	Special hardware/software equipment that connects multiple and diverse networks. The router is in charge of delivering packets of data from a local network to a remote network. Routers are the traffic cops of the Internet, monitoring all traffic and moving data from one network to another.
World Wide Web (WWW or the Web)	Worldwide network collection of specially formatted and interconnected documents known as *Web pages*. The Web is just *one* of *many* services provided by the Internet.
Web page	A document containing text and special commands (or tags) written in *Hypertext Markup Language (HTML)*. A Web page can contain text, graphics, video, audio, and other elements.
Hypertext Markup Language (HTML)	Standard document-formatting language for Web pages. HTML allows documents to be presented in a Web browser in a standard manner.
Hyperlink	Web pages are linked to each other—that is, each Web page calls other Web pages—creating the effect of a "web." Because a link can connect to different types of documents such as text, graphics, animated graphics, video, and audio, it is known as a "hyperlink." A hyperlink is generally expressed as a *URL* in an HTML-formatted Web page.
Uniform Resource Locator (URL) or Web address	A URL identifies the address of a resource on the Internet. The URL is an abbreviation (ideally easily remembered) that uniquely identifies an Internet resource. Examples of URLs include *www.dell.com*, *www.ford.com*, *www.amazon.com*, *www.faa.gov*, and *www.mtsu.edu*.
Hypertext Transfer Protocol (HTTP)	The standard protocol used by the *Web browser* and *Web server* to communicate—to send requests and replies between servers and browsers. HTTP uses TCP/IP to transmit the data between computers on the Internet.
Domain Name System (Service)	DNS translates the "English-like" domain names (such as whitehouse.org or ebay.com) to the appropriate TCP/IP addresses. The DNS service lies at the heart of the Internet because most hyperlinks use URLs to refer to other Web pages.
Web browser	The end-user application used to *browse* or *navigate* (move from page to page) through the Internet. The browser is a graphical application that runs on the client computer, and its main function is to display Web pages. A client uses the Web browser (for example, Netscape Navigator, Microsoft Internet Explorer, and Opera) to request Web pages from a Web *server*.
Web server	A specialized application whose only function is to "listen" for client requests, process them, and send the requested Web page back to the client browser. The Web server and the Web client communicate using a special protocol known as *Hypertext Transfer Protocol* or *HTTP*.

Internet Building Blocks (2)

BUILDING BLOCK/SERVICE	DESCRIPTION
Web site	Term used to refer to the Web server and the collection of Web pages stored on the local hard disk of the server computer or an accessible shared directory.
Static Web page	A Web page whose contents remain the same (when viewed in a browser) unless the page is manually edited. An example of a static Web page is a standard pricelist posted by a manufacturer for inspection by the manufacturer's customers.
Dynamic Web page	A Web page whose contents are automatically created and tailored to an end user's needs each time the end user requests the page. For example, an end user can access a Web page that displays the latest stock prices for the companies selected by the end user.
File Transfer Protocol (FTP)	Protocol used to provide file transfer capabilities among computers in the Internet. An FTP client requests a file to an FTP server. The FTP server listens for clients' requests, processes them, and sends the requested files back to the client.
Electronic mail (e-mail)	Messages transmitted electronically among computers on the Internet. A mail server stores e-mail messages in end-user mailboxes. Mail clients retrieve e-mail from the mail server. When a client sends an e-mail, it is temporarily stored on the mail server, which in turn delivers the e-mail to the correct destination.
News and discussion group services	Specialized services that allow the creation of "virtual communities" in which users exchange messages regarding specific topics, for example, aviation, sports, and computers. This service allows end users to post information on shared bulletin boards for public access.

Basic Internet Services



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
  - Some general types:
    - Plug-ins: external application automatically invoked by the browse when needed, allow the Web server to properly handle data not originally supported
    - Java(Sun Microsystems) and JavaScript(Netscape): run on top of the Web browser software, call Java routines embedded inside the HTML page, JavaScript is simpler than Java
    - ActiveX(Microsoft) and VBScript: IE/Windows client browser specific, connect Windows "controls" to the web, interoperated by Microsoft's .NET framework

  Without such client validation → the entire data set must be validated by the client server (CPU waste)

Server-side Extensions: Web-to-Database Middleware:

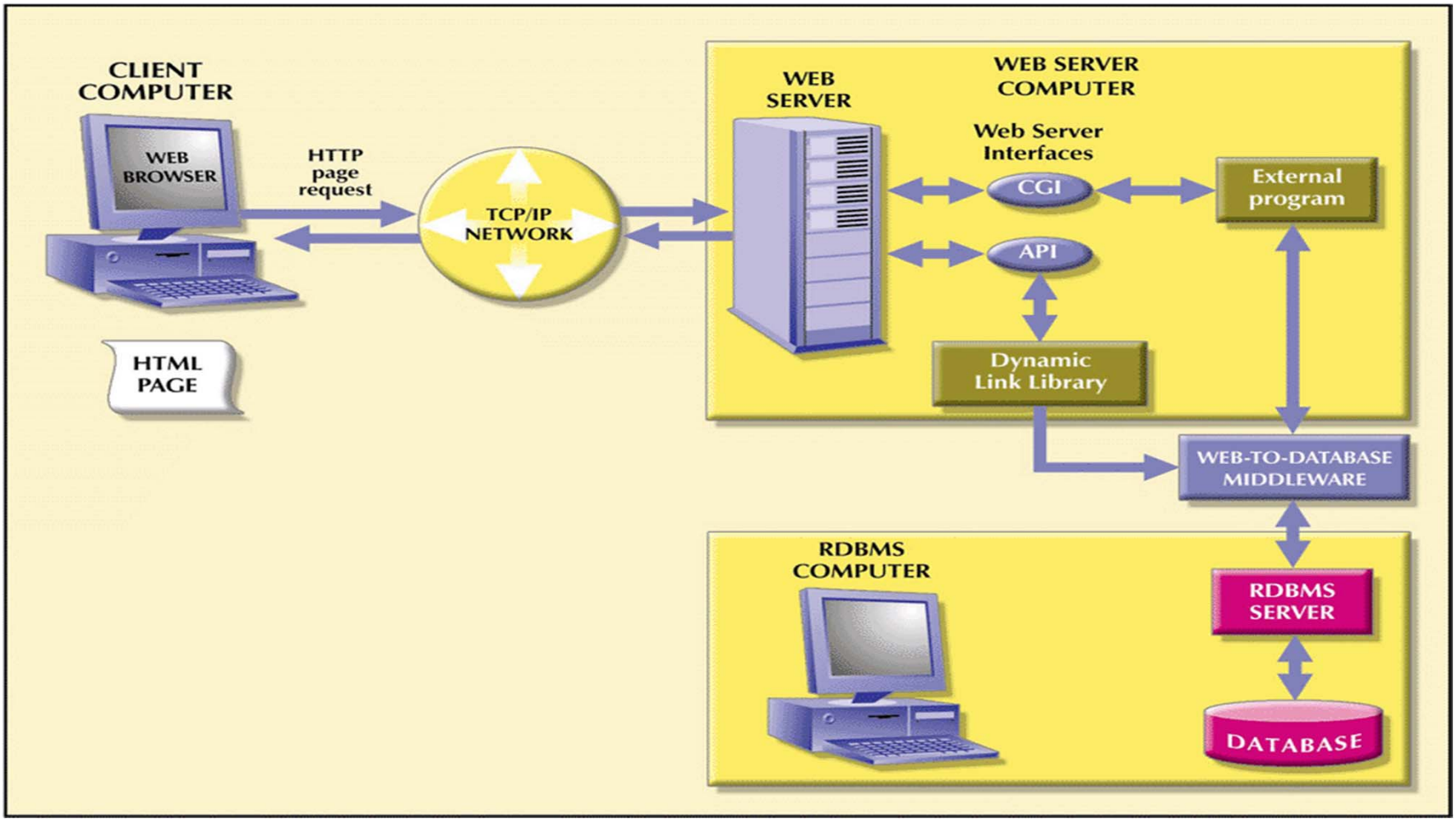
- Also known as Web-to-database middleware
- Program that interacts directly with Web server to handle specific types of requests
- Provides its services to the Web server in a way that is totally transparent to the client browser

Web Server Interfaces

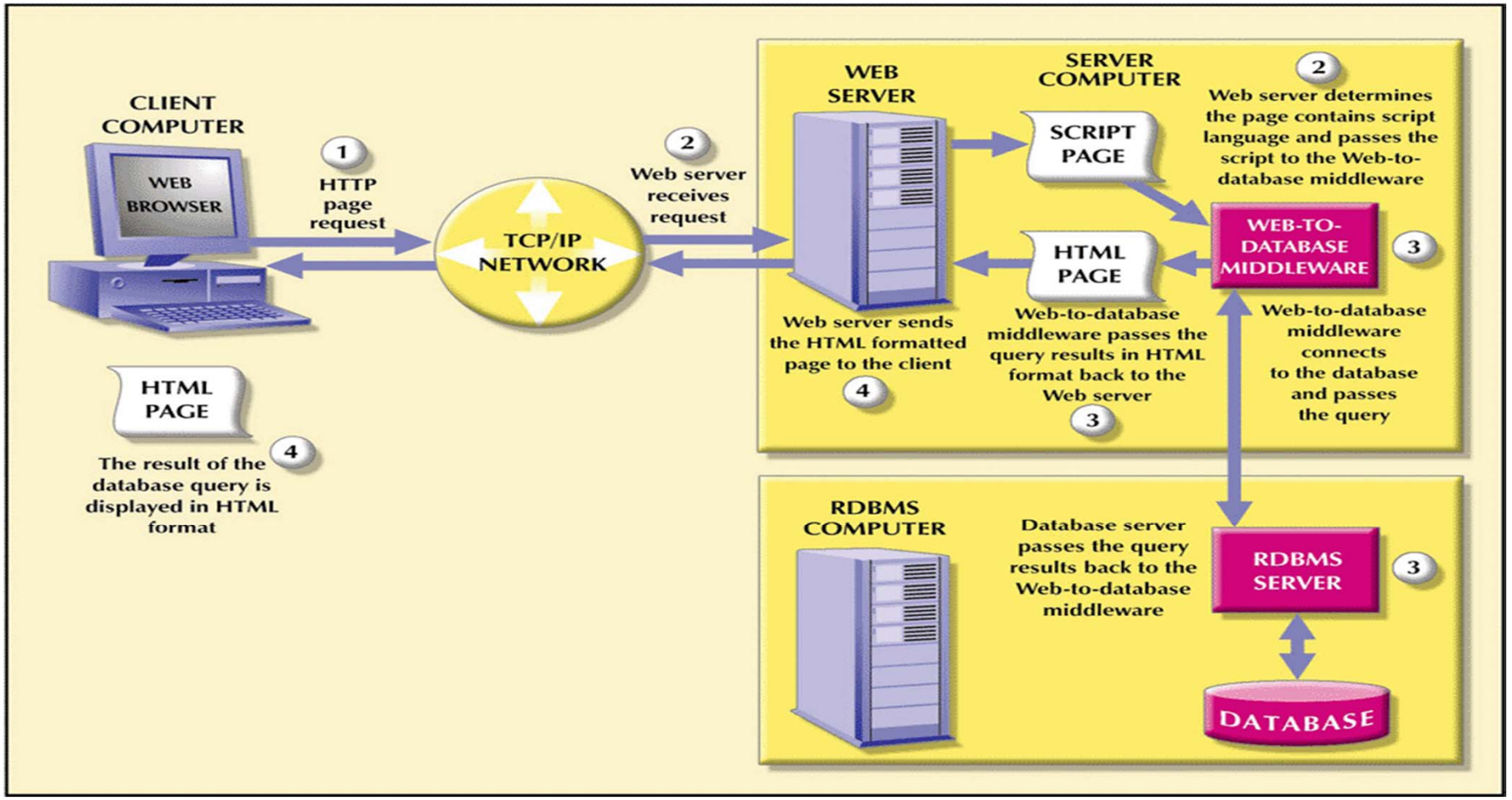
웹 서버: *이용자의 요구에 응답해 미리 준비된 정보를 전달*

CGI(*Common Gateway Interface*): *웹 서버 상에서 사용자 프로그램을 동작시키기 위한 조합 (예. 블로그, 게시판)*

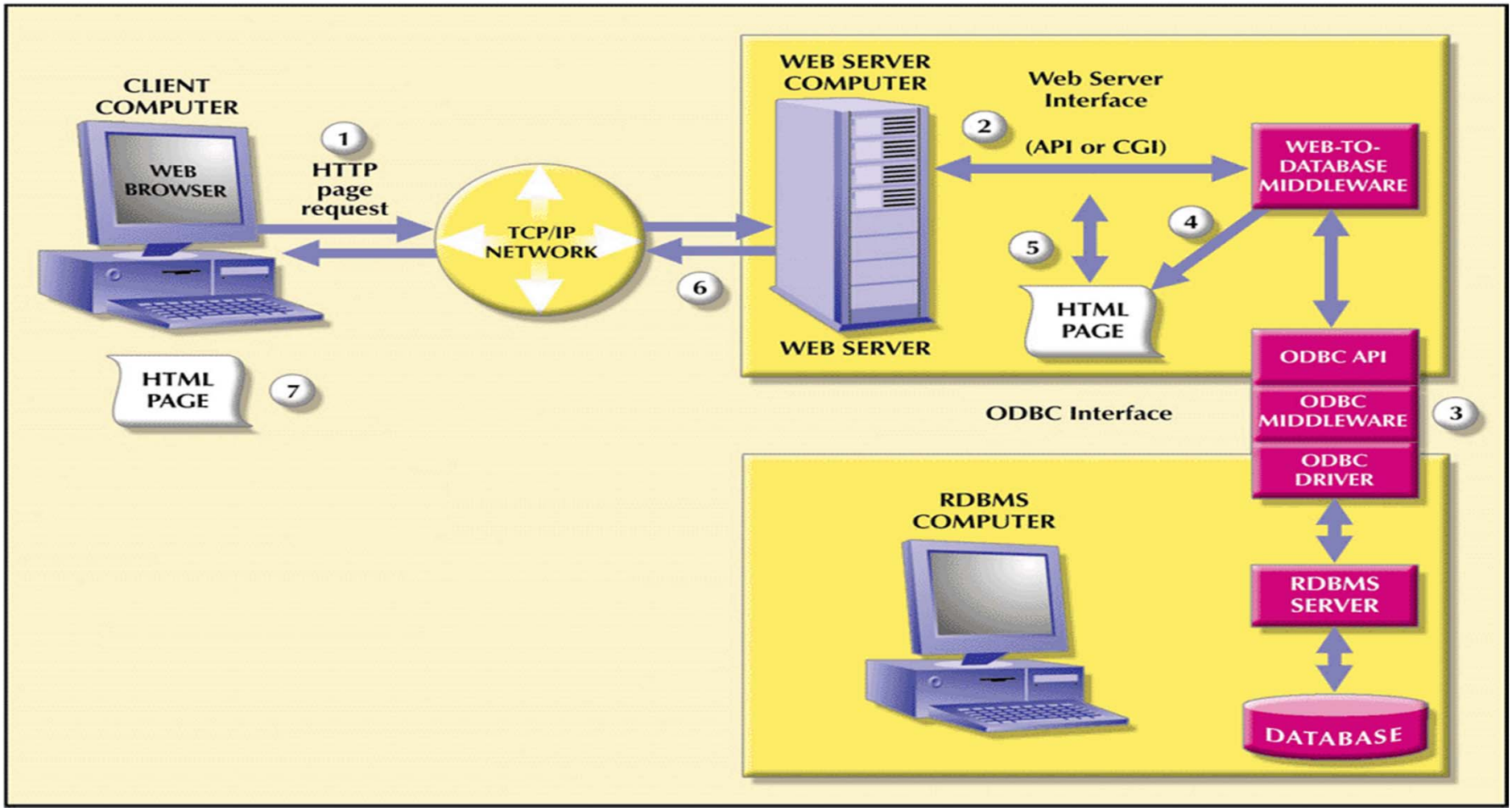
API(*Application Programming Interface*): *응용프로그램에서 사용할 수 있도록 운영체제나 프로그래밍 언어가 제공하는 기능을 제어 (파일제어, 창 제어, 문자 제어 등을 위한 인터페이스를 제공)*



Web-to-Database Middleware



Web-to-Database Middleware Uses ODBC to Access Databases



***ODBC**(Open Database Connectivity): DB에 접근하기 위한 소프트웨어의 표준규격*

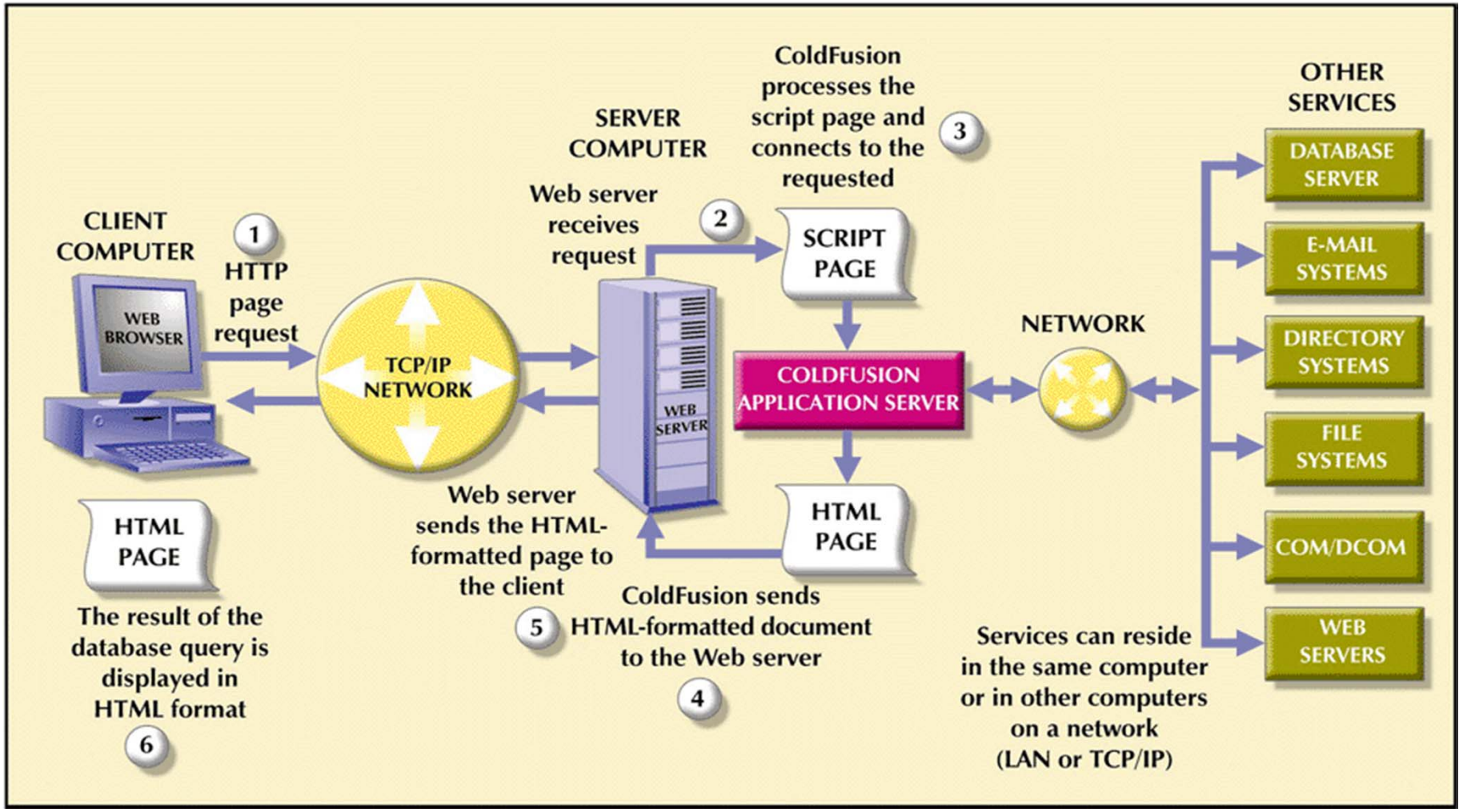
Using a Web-to-Database Production Tool: ColdFusion

- 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

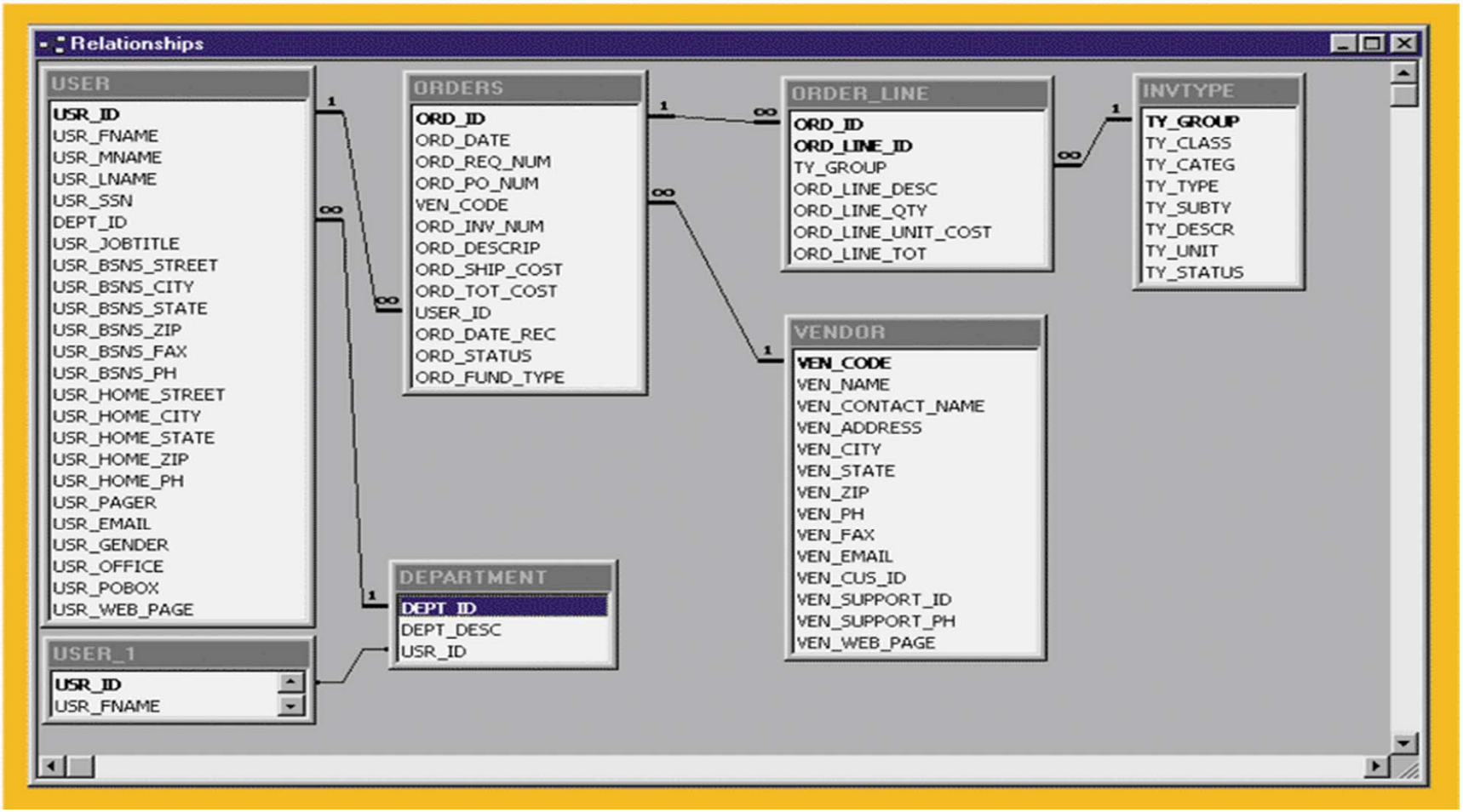
Using a Web-to-Database Production Tool: ColdFusion (2)

- 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
  - Other functions

How ColdFusion Works



The Ch14_RobCor Database's Relational Schema



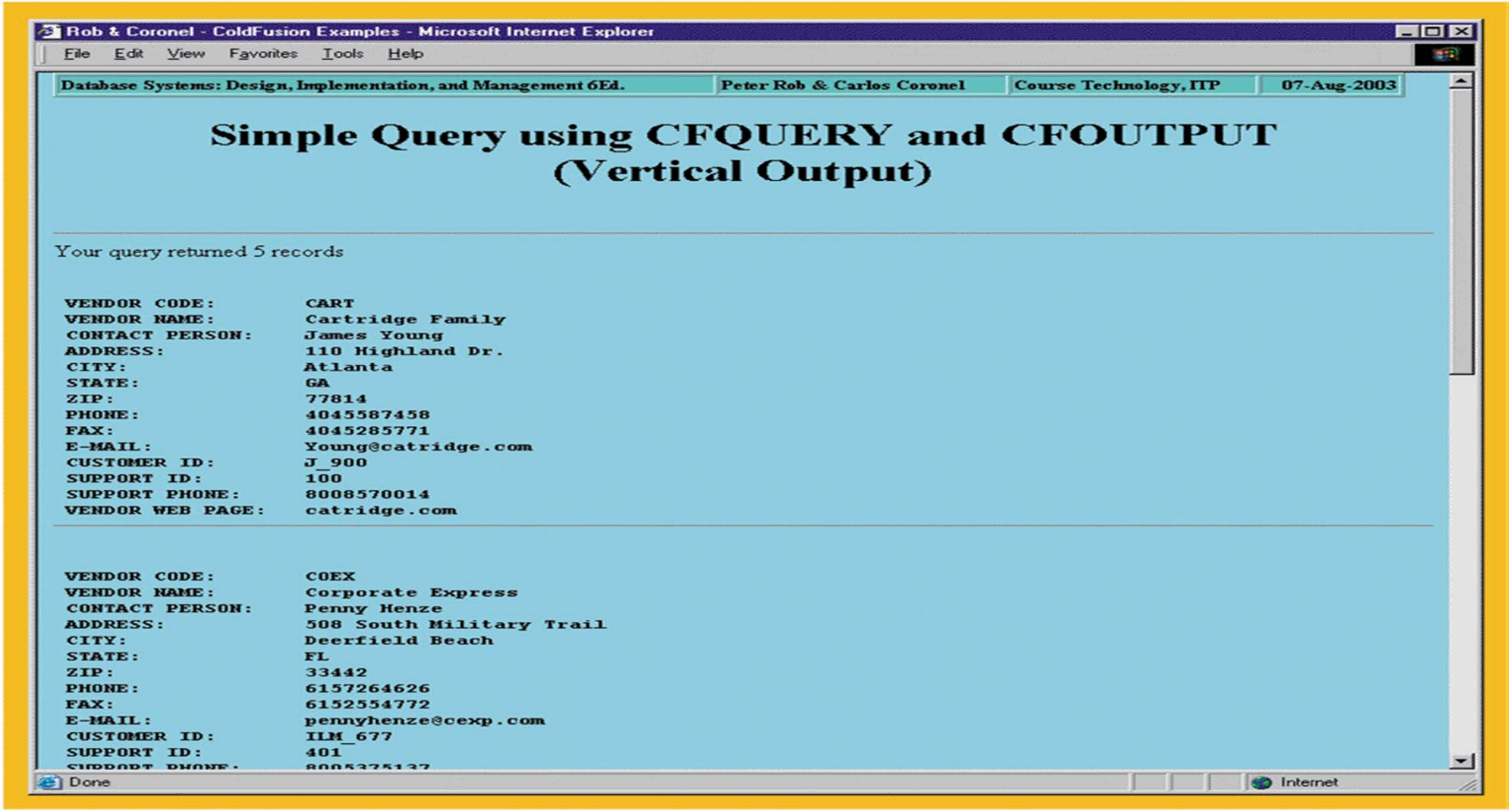
A Simple Query Using CFQUERY and CFOUTPUT

SCRIPT 14.1 (RC-1.CFM): A SIMPLE QUERY USING CFQUERY AND CFOUTPUT

```
<HTML>
<HEAD>
<TITLE>Rob & Coronel -  ColdFusion Examples</TITLE>
<CFQUERY NAME="venlist" DATASOURCE="RobCor">
    SELECT * FROM VENDOR ORDER BY VEN_CODE
</CFQUERY>
</HEAD>
<BODY BGCOLOR="LIGHTBLUE">
<H1>
<CENTER><B>Simple Query using CFQUERY and CFOUTPUT</B></CENTER>
<CENTER><B>(Vertical Output)</B></CENTER>
</H1>
<BR>
<HR>
<CFOUTPUT>
    Your query returned #venlist.RecordCount# records
</CFOUTPUT>
<CFOUTPUT QUERY="venlist">
<PRE><B>
 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#
<HR></B></PRE>
</CFOUTPUT>
<FORM ACTION="rc-0.cfm" METHOD="post">
    <INPUT TYPE="submit" VALUE="Main Menu ">
</FORM>
</BODY>
</HTML>
```

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

FIGURE 14.7 THE RC-1.CFM SCRIPT OUTPUT (VERTICAL LISTING)



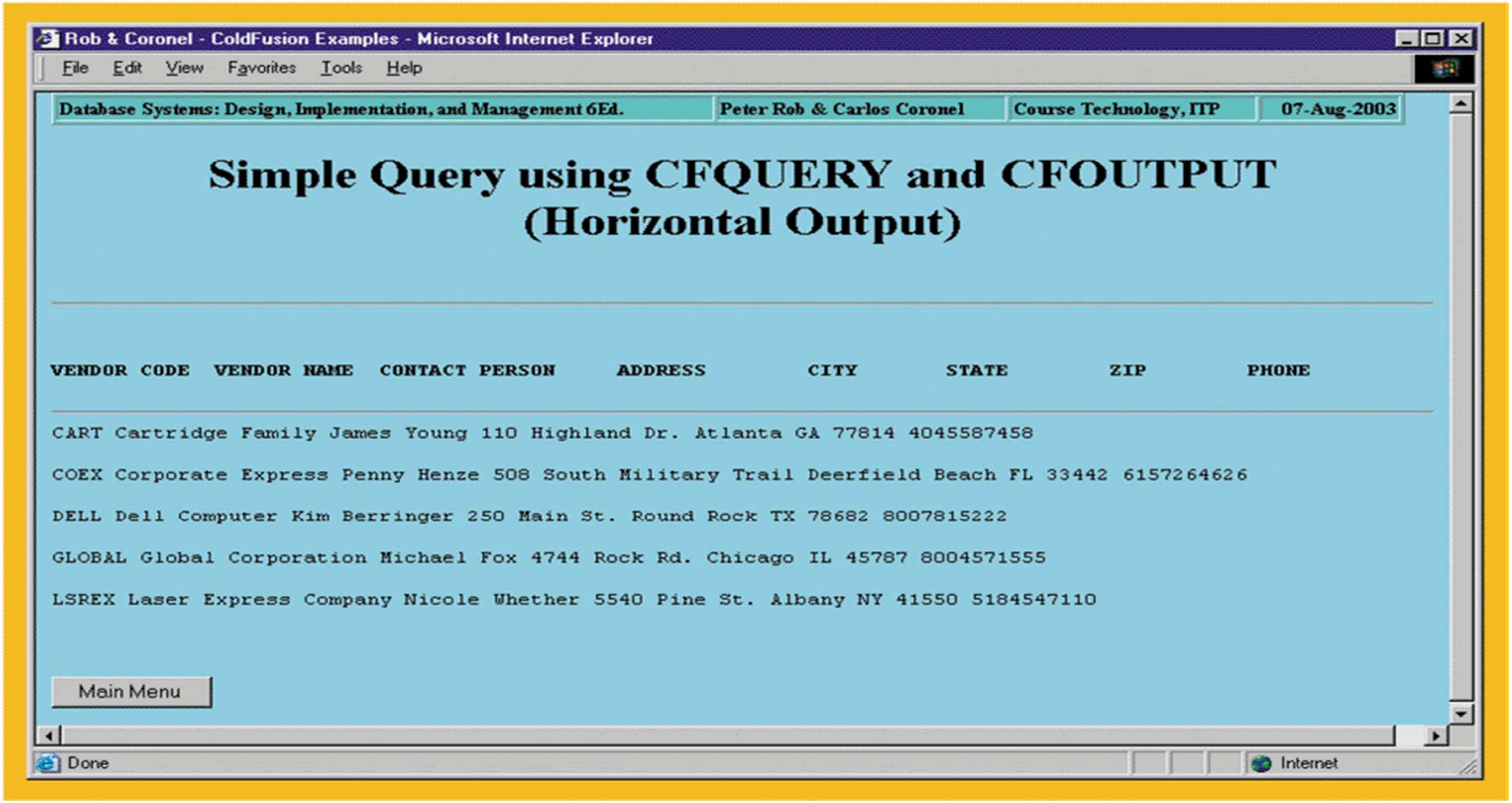
CFQUERY with Tabular CFOUTPUT

SCRIPT 14.2 (RC-2.CFM): CFQUERY WITH TABULAR CFOUTPUT

```
<HTML>
<CFQUERY NAME="VENDATA" DATASOURCE="RobCor">
        SELECT * FROM VENDOR ORDER BY VEN_CODE
</CFQUERY>
<HEAD>
<TITLE>Rob & Coronel -  ColdFusion Examples</TITLE>
</HEAD>
<BODY BGCOLOR="LIGHTBLUE">
<H1>
<B><CENTER>Simple Query using CFQUERY and CFOUTPUT</CENTER></B>
<B><CENTER>(Horizontal Output)</CENTER></B>
</H1>
<BR>
<HR>
<PRE>
<B>
VENDOR CODE  VENDOR NAME  CONTACT PERSON     ADDRESS        CITY       STATE        ZIP        PHONE
</B>
<HR>
<CFOUTPUT QUERY="VENDATA">
#VEN_CODE#   #VEN_NAME#   #VEN_CONTACT_NAME# #VEN_ADDRESS#  #VEN_CITY# #VEN_STATE#  #VEN_ZIP#  #VEN_PH#<BR>
</CFOUTPUT>
</PRE>
<FORM ACTION="rc-0.cfm" METHOD="post">
    <INPUT TYPE="submit" VALUE="Main Menu ">
</FORM>
</BODY>
</HTML>
```

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

FIGURE 14.8 THE RC-2.CFM SCRIPT OUTPUT (HORIZONTAL LISTING)



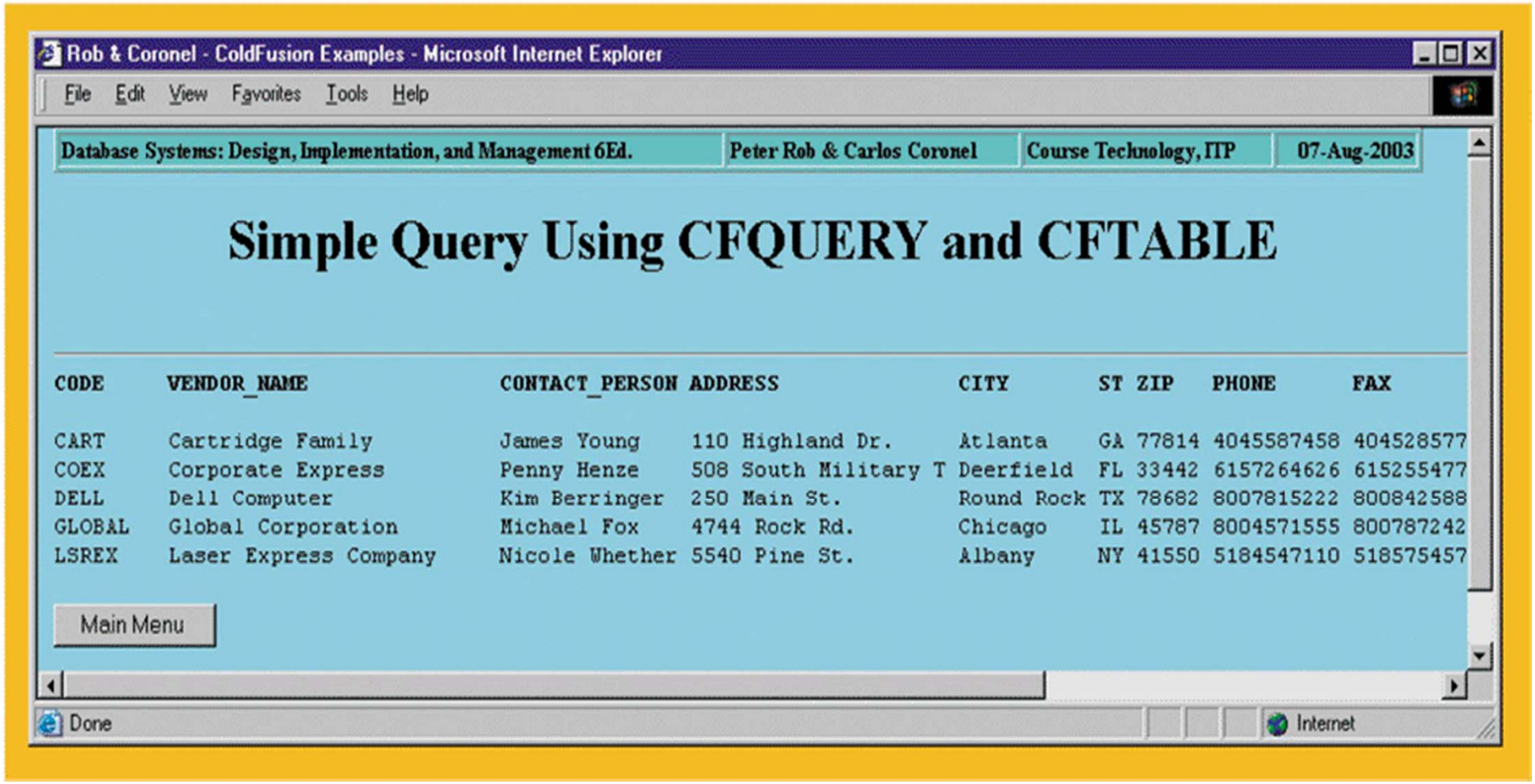
CFQUERY with CFTABLE

SCRIPT 14.3 (RC-3.CFM): CFQUERY WITH CFTABLE

```
<HTML>
<HEAD>
<TITLE>Rob & Coronel -  ColdFusion Examples</TITLE>
<CFQUERY NAME="VENDATA" DATASOURCE="RobCor">
        SELECT * FROM VENDOR ORDER BY VEN_CODE
</CFQUERY>
</HEAD>
<BODY BGCOLOR="LIGHTBLUE">
<H1>
<B><CENTER>Simple Query Using CFQUERY and CFTABLE</CENTER></B>
</H1>
<BR>
<TABLE BGCOLOR="Silver" BORDERCOLOR="Fuchsia" FRAME="BORDER">
<TR><HR></TR>
<TR>
<CFTABLE QUERY="VENDATA" STARTROW="1" COLSPACING="1" COLHEADERS>
<CFCOL HEADER="<B>CODE</B>" WIDTH="8" TEXT="#VEN_CODE#">
<CFCOL HEADER="<B>VENDOR_NAME</B>" WIDTH="25" TEXT="#VEN_NAME#">
<CFCOL HEADER="<B>CONTACT_PERSON</B>" WIDTH="14" TEXT="#VEN_CONTACT_NAME#">
<CFCOL HEADER="<B>ADDRESS</B>" WIDTH="20" TEXT="#VEN_ADDRESS#">
<CFCOL HEADER="<B>CITY</B>" WIDTH="10" TEXT="#VEN_CITY#">
<CFCOL HEADER="<B>STATE</B>" WIDTH="2" TEXT="#VEN_STATE#">
<CFCOL HEADER="<B>ZIP</B>" WIDTH="5" TEXT="#VEN_ZIP#">
<CFCOL HEADER="<B>PHONE</B>" WIDTH="10" TEXT="#VEN_PH#">
<CFCOL HEADER="<B>FAX</B>" WIDTH="10" TEXT="#VEN_FAX#">
<CFCOL HEADER="<B>E-MAIL</B>" WIDTH="10" TEXT="#VEN_EMAIL#">
<CFCOL HEADER="<B>CUSTOMER_ID</B>" WIDTH="8" TEXT="#VEN_CUS_ID#">
<CFCOL HEADER="<B>SUPPORT_PHONE</B>" WIDTH="6" TEXT="#VEN_SUPPORT_ID#">
<CFCOL HEADER="<B>WEB PAGE</B>" WIDTH="14" TEXT="#VEN_WEB_PAGE#">
</CFTABLE>
</TR>
<TR>
<FORM ACTION="rc-0.cfm" METHOD="post">
    <INPUT TYPE="submit" VALUE="Main Menu ">
</FORM>
</TR>
</TABLE>
</BODY>
</HTML>
```

The rc-3.cfm Script Output (Formatted Horizontal Listing)

FIGURE 14.9 THE RC-3.CFM SCRIPT OUTPUT (FORMATTED HORIZONTAL LISTING)



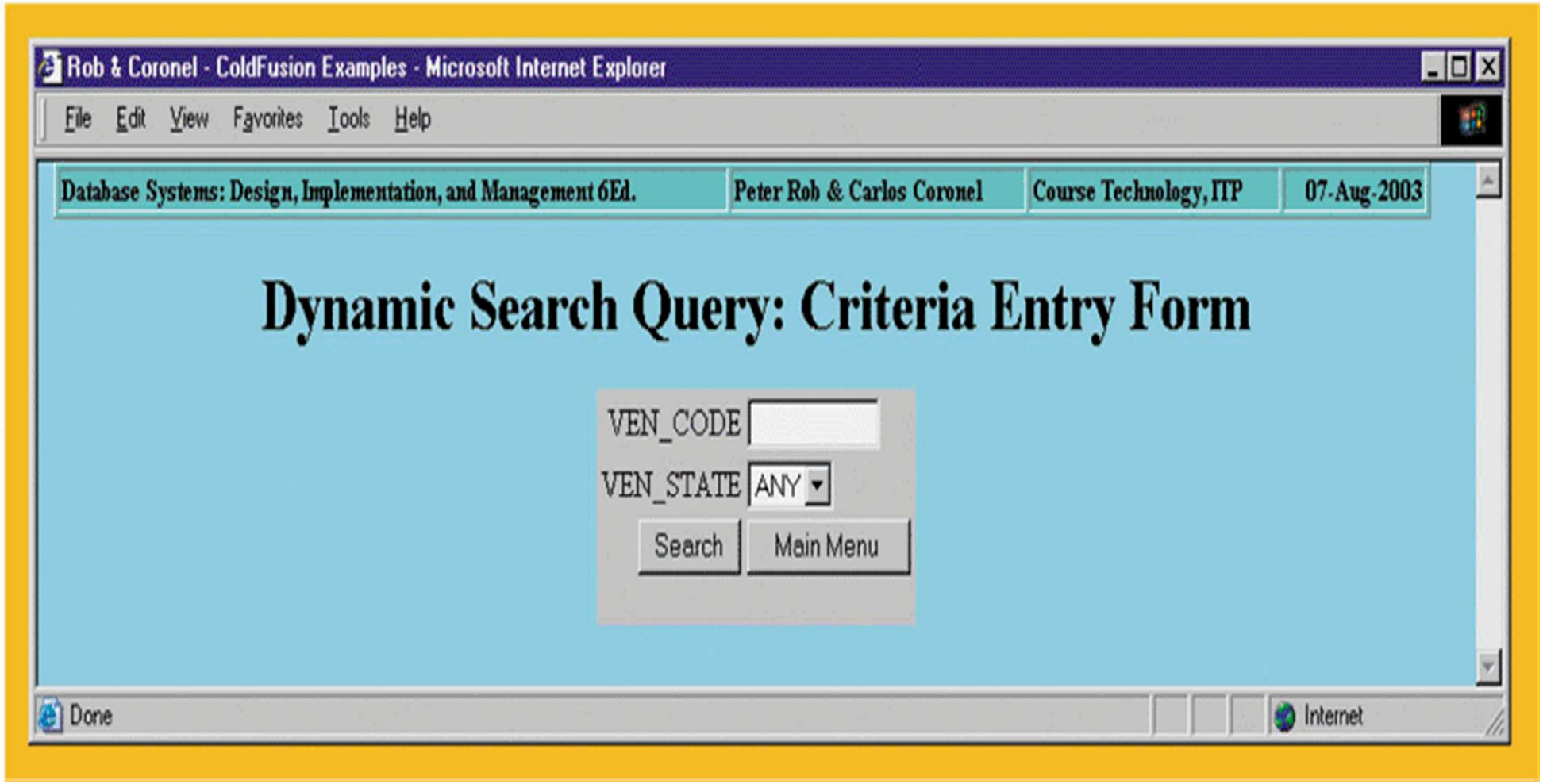
Dynamic Search Query: Criteria Entry Form

SCRIPT 14.4A (RC-4A.CFM): DYNAMIC SEARCH QUERY: CRITERIA ENTRY FORM

```
<HTML>
<CFQUERY NAME="STATELIST" DATASOURCE="RobCor">
     Select VEN_STATE from VENDOR Order by VEN_STATE
</CFQUERY>
<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
           <CFOUTPUT QUERY="STATELIST">
           <OPTION VALUE="#STATELIST.VEN_STATE#">#VEN_STATE#
           </CFOUTPUT>
           </SELECT>
     </TD>
     </TR>
     <TR>
     <TD ALIGN="right" VALIGN="middle">
          <INPUT TYPE="submit" VALUE="Search">
          </FORM>
     </TD>
     <TD ALIGN="right" VALIGN="middle">
          <FORM ACTION="rc-0.cfm" METHOD="post">
               <INPUT TYPE="submit" VALUE="Main Menu ">
          </FORM>
     </TD>
     </TR>
</TABLE>
</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)



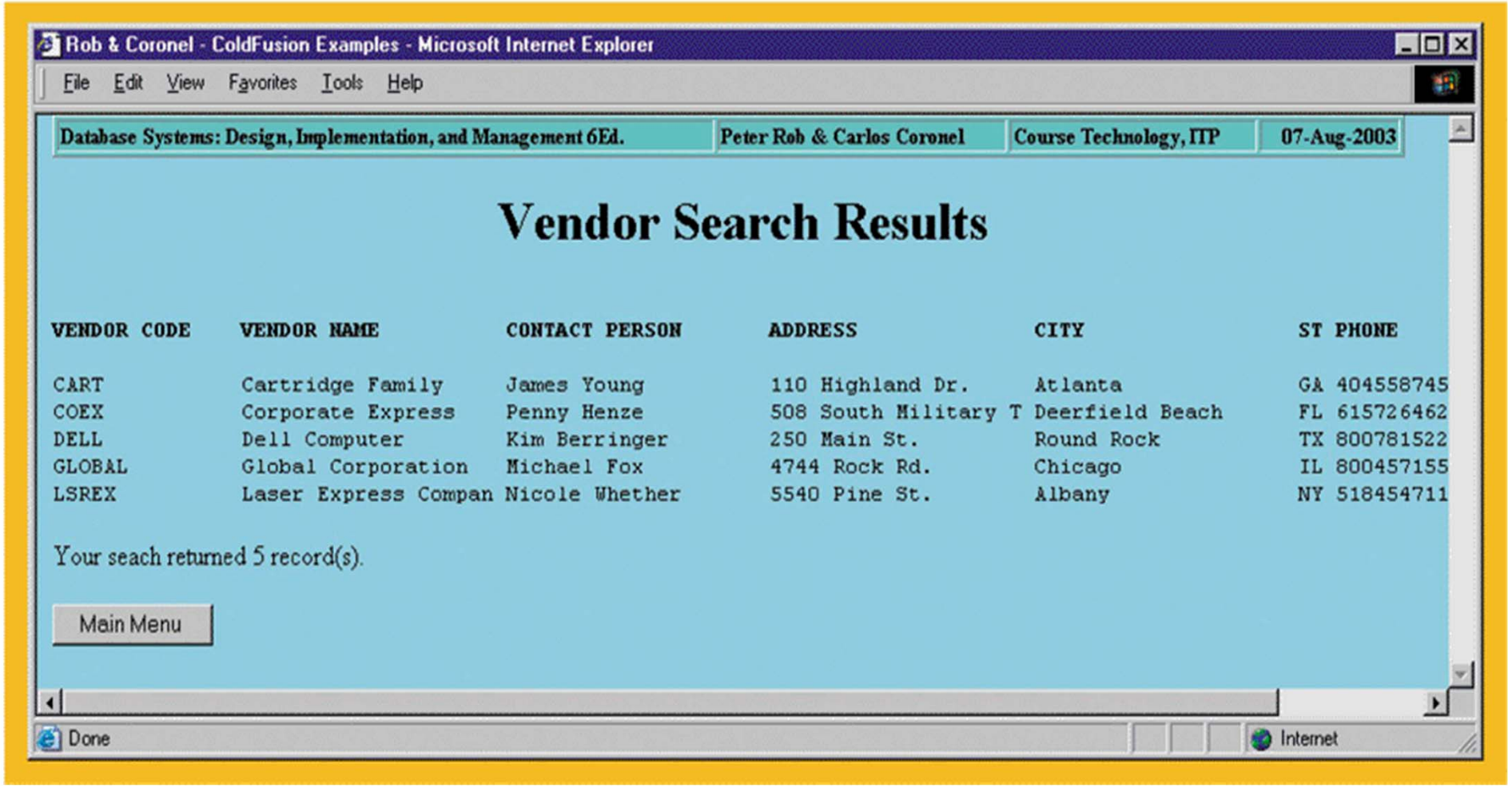
The VENDOR Search Results

SCRIPT 14.4B (RC-4B.CFM): THE VENDOR SEARCH RESULTS

```
<HTML>
<CFQUERY NAME="SearchVendor" DATASOURCE="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>
<HEAD>
<TITLE>Rob & Coronel -  ColdFusion Examples</TITLE>
</HEAD>
<BODY BGCOLOR="LIGHTBLUE">
<H1>
<CENTER><B>Vendor Search Results</B></CENTER>
</H1>
<BR>
<CFTABLE QUERY="SearchVendor" STARTROW="1" COLSPACING="1" COLHEADERS>
<CFCOL HEADER="<B>VENDOR CODE</B>" WIDTH="14" TEXT="#VEN_CODE#">
<CFCOL HEADER="<B>VENDOR NAME</B>" WIDTH="20" TEXT="#VEN_NAME#">
<CFCOL HEADER="<B>CONTACT PERSON</B>" WIDTH="20" TEXT="#VEN_CONTACT_NAME#">
<CFCOL HEADER="<B>ADDRESS</B>" WIDTH="20" TEXT="#VEN_ADDRESS#">
<CFCOL HEADER="<B>CITY</B>" WIDTH="20" TEXT="#VEN_CITY#">
<CFCOL HEADER="<B>STATE</B>" WIDTH="2" TEXT="#VEN_STATE#">
<CFCOL HEADER="<B>PHONE</B>" WIDTH="20" TEXT="#VEN_PH#">
</CFTABLE>
<CFIF #SEARCHVENDOR.RECORDCOUNT# IS 0>
    <H2>No records were found matching your criteria </H2>
<CFELSE>
    <CFOUTPUT>Your seach returned #SearchVendor.RecordCount# record(s).</CFOUTPUT>
</CFIF>
<FORM ACTION="rc-0.cfm" METHOD="post">
    <INPUT TYPE="submit" VALUE="Main Menu ">
</FORM>
</BODY>
</HTML>
```

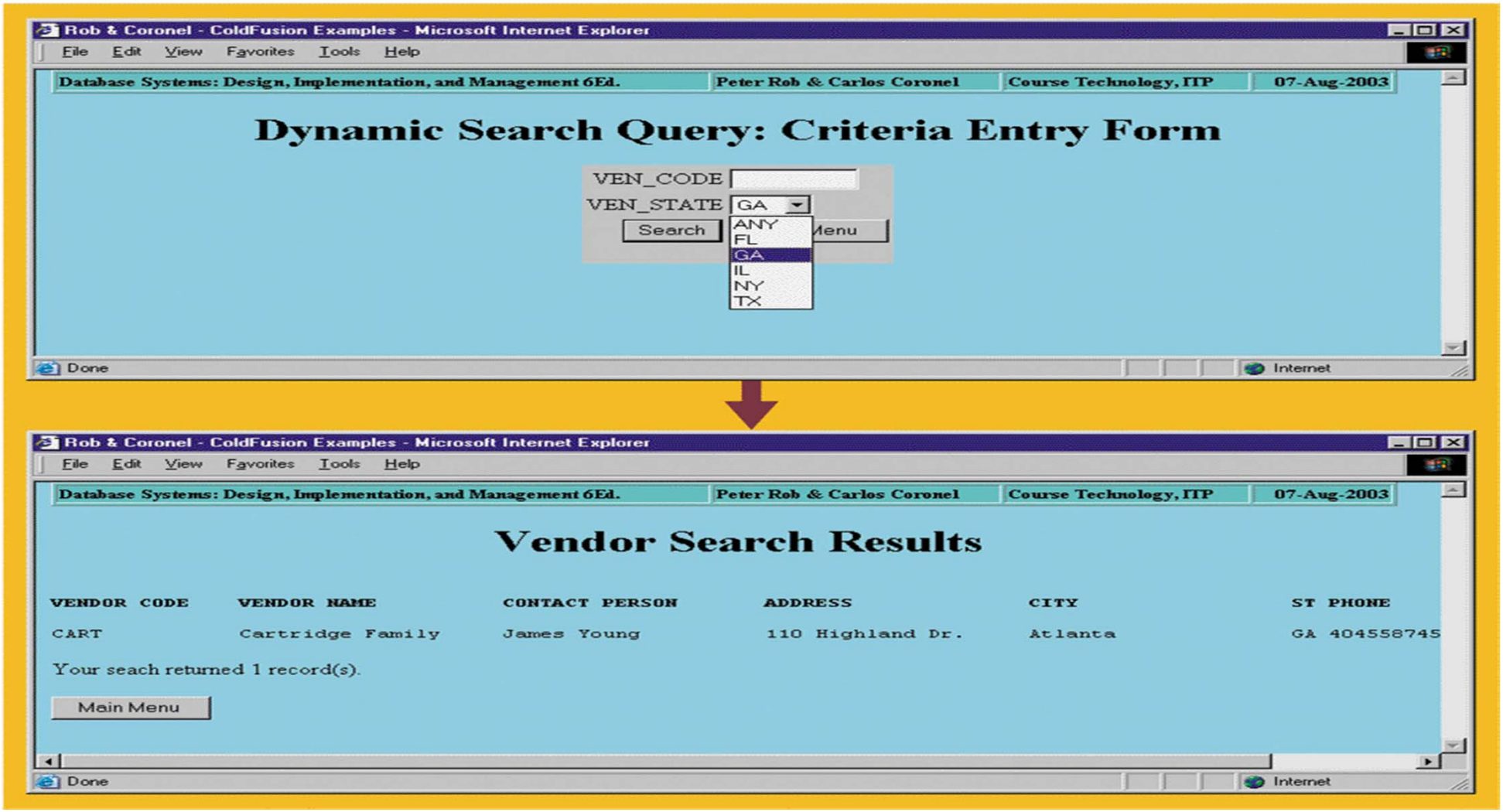
The rc-4b.cfm Script Output (Vendor Search Results: All States)

FIGURE 14.11 THE RC-4B.CFM SCRIPT OUTPUT (VENDOR SEARCH RESULTS: ALL STATES)



The Vendor List for the Condition VEN_STATE = "GA"

FIGURE 14.12 THE VENDOR LIST FOR THE CONDITION VEN_STATE = "GA"



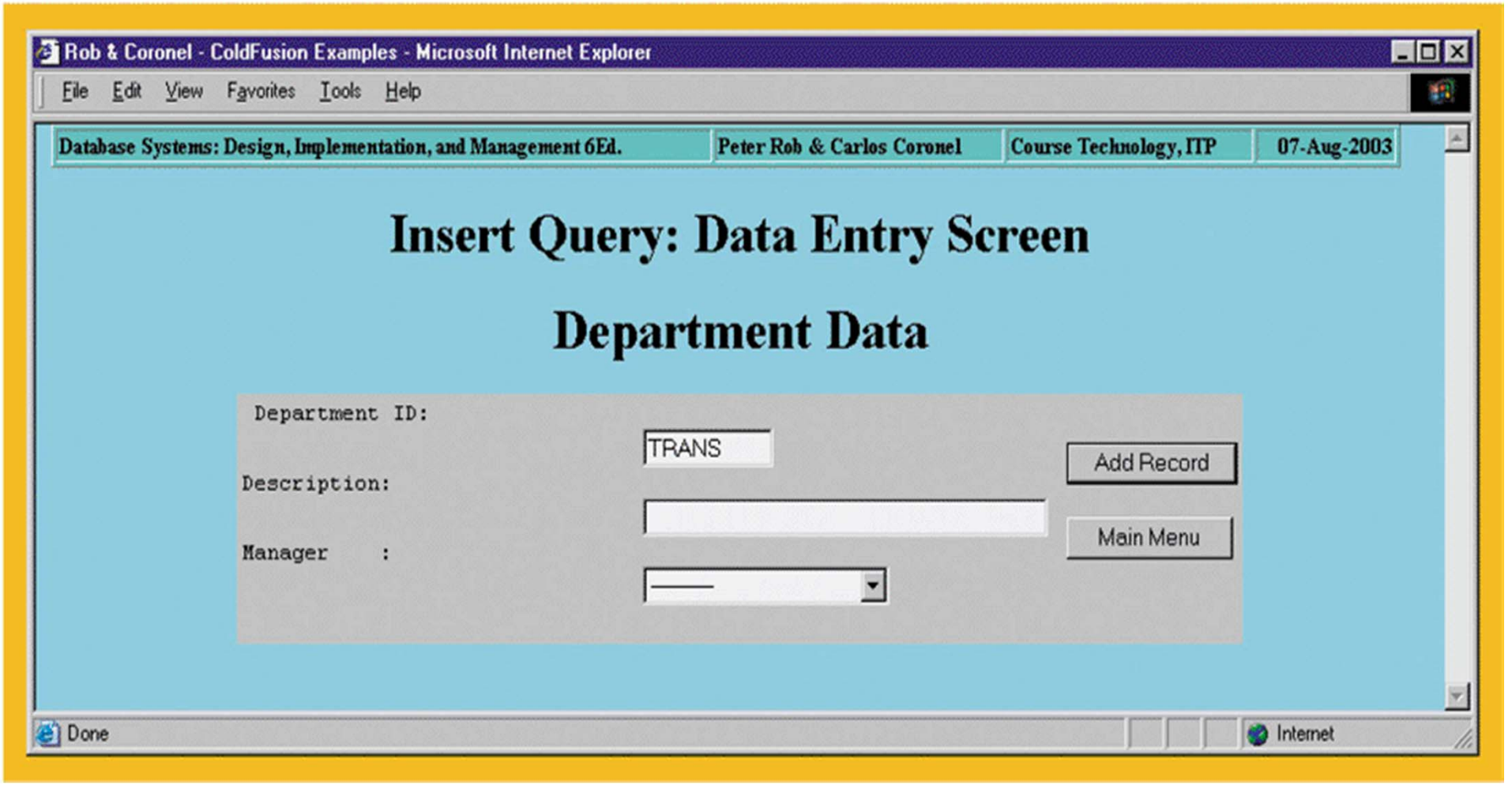
Insert Query: Data Entry Script

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>
<CENTER><B>Insert Query: Data Entry Screen</B></CENTER></H1>
<FORM ACTION="rc-5b.cfm" METHOD="post">
<CENTER><H1>Department Data</H1></CENTER><!--- the following code defines required fields --->
    <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>
            <PRE>
 Department ID:
                <INPUT TYPE="text" NAME="DEPT_ID" SIZE="10" MAXLENGTH="10"><BR>Description:
                <INPUT TYPE="text" NAME="DEPT_DESC" SIZE="35" MAXLENGTH="35"><BR>Manager    :
                <SELECT NAME="USR_ID" SIZE="1"><!--- select user from list --->
                    <OPTION VALUE="">----------
                    <CFOUTPUT QUERY="USRLIST">
                        <OPTION VALUE="#UsrList.USR_ID#">[#USR_ID#] #USR_LNAME#, #USR_FNAME#, #USR_MNAME#
                    </CFOUTPUT>
                </SELECT></PRE>
            </TD>
            <TD>
                <INPUT TYPE="submit" VALUE="Add Record">
</FORM>
<FORM ACTION="rc-0.cfm" METHOD="post">
    <INPUT TYPE="submit" VALUE=" Main Menu ">
</FORM>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

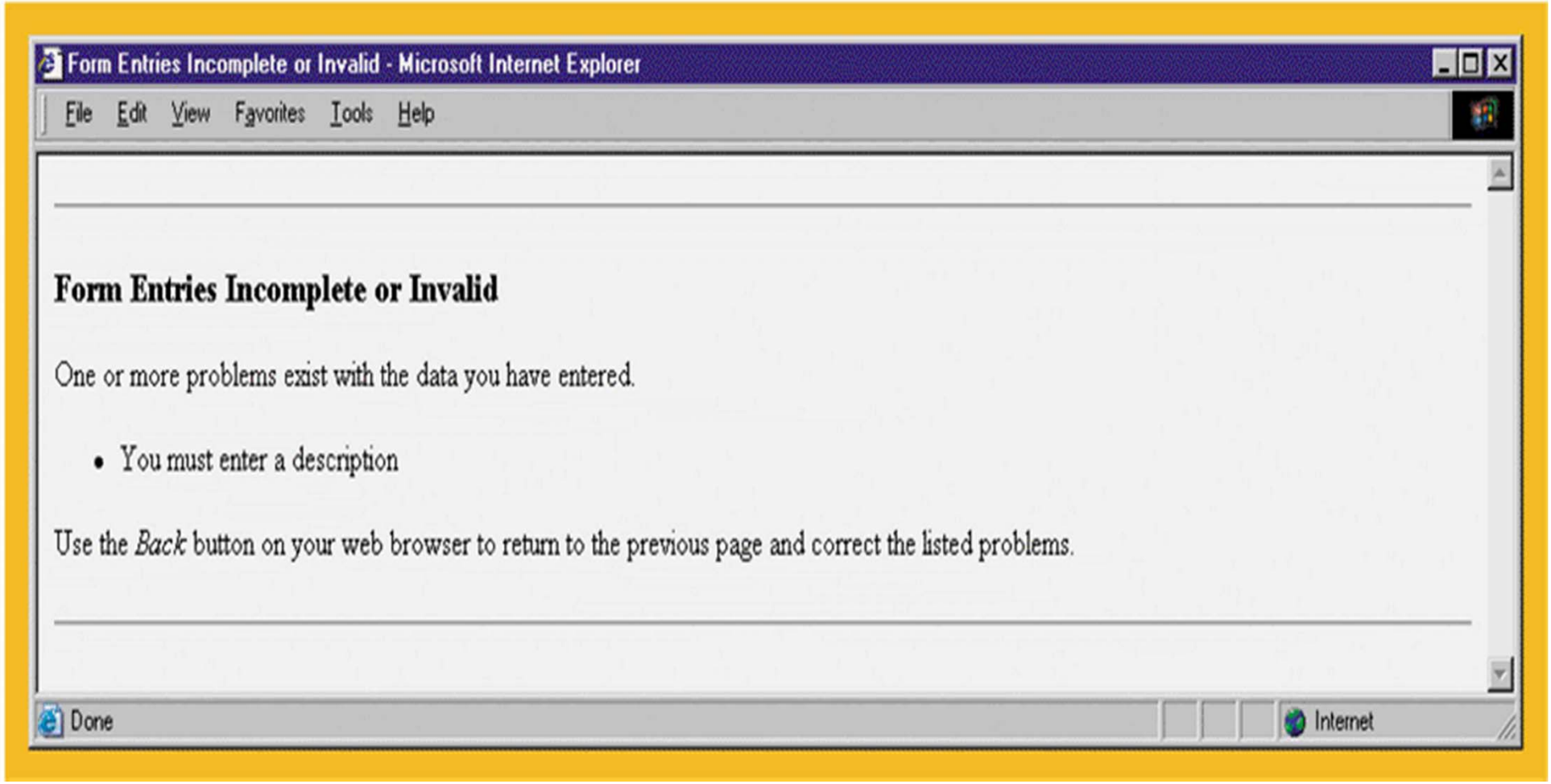
The rc-5a.cfm Script Output (Department Data Entry Screen)

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

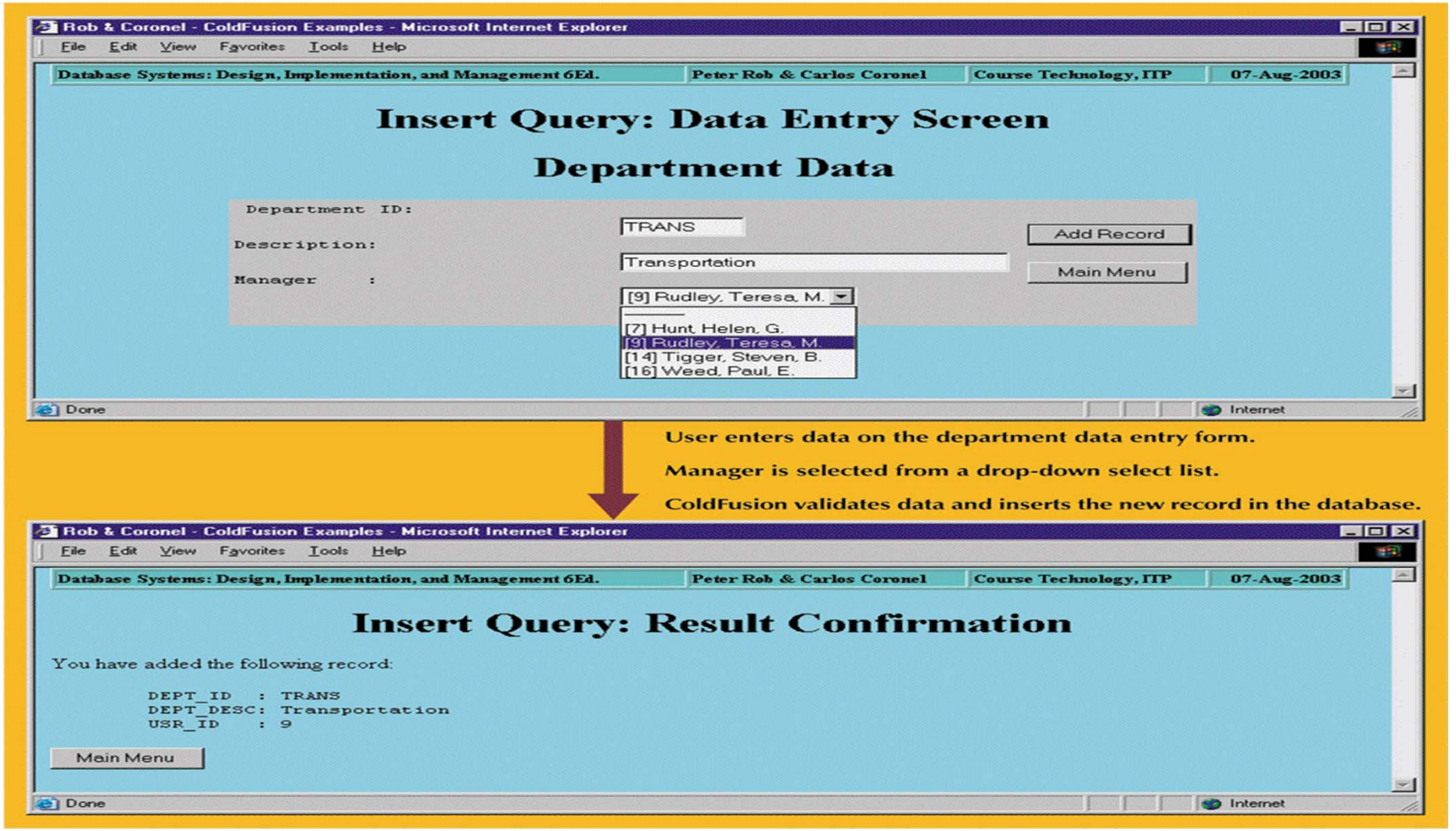


Insert Query: Confirmation Script

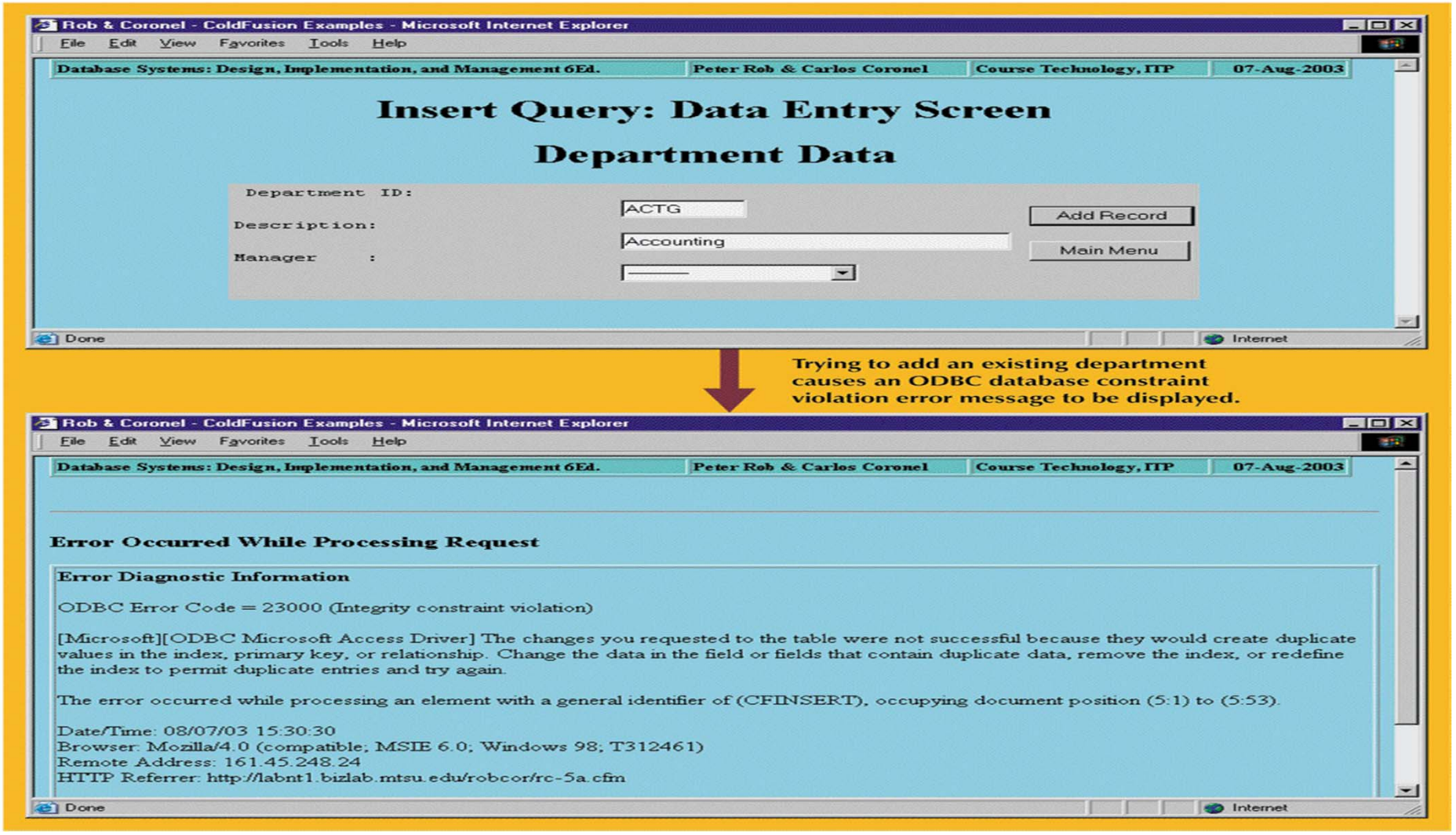
SCRIPT 14.5B (RC-5B.CFM): INSERT QUERY: CONFIRMATION SCRIPT

```
<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)



The Insert Query: ODBC Integrity Violation Error



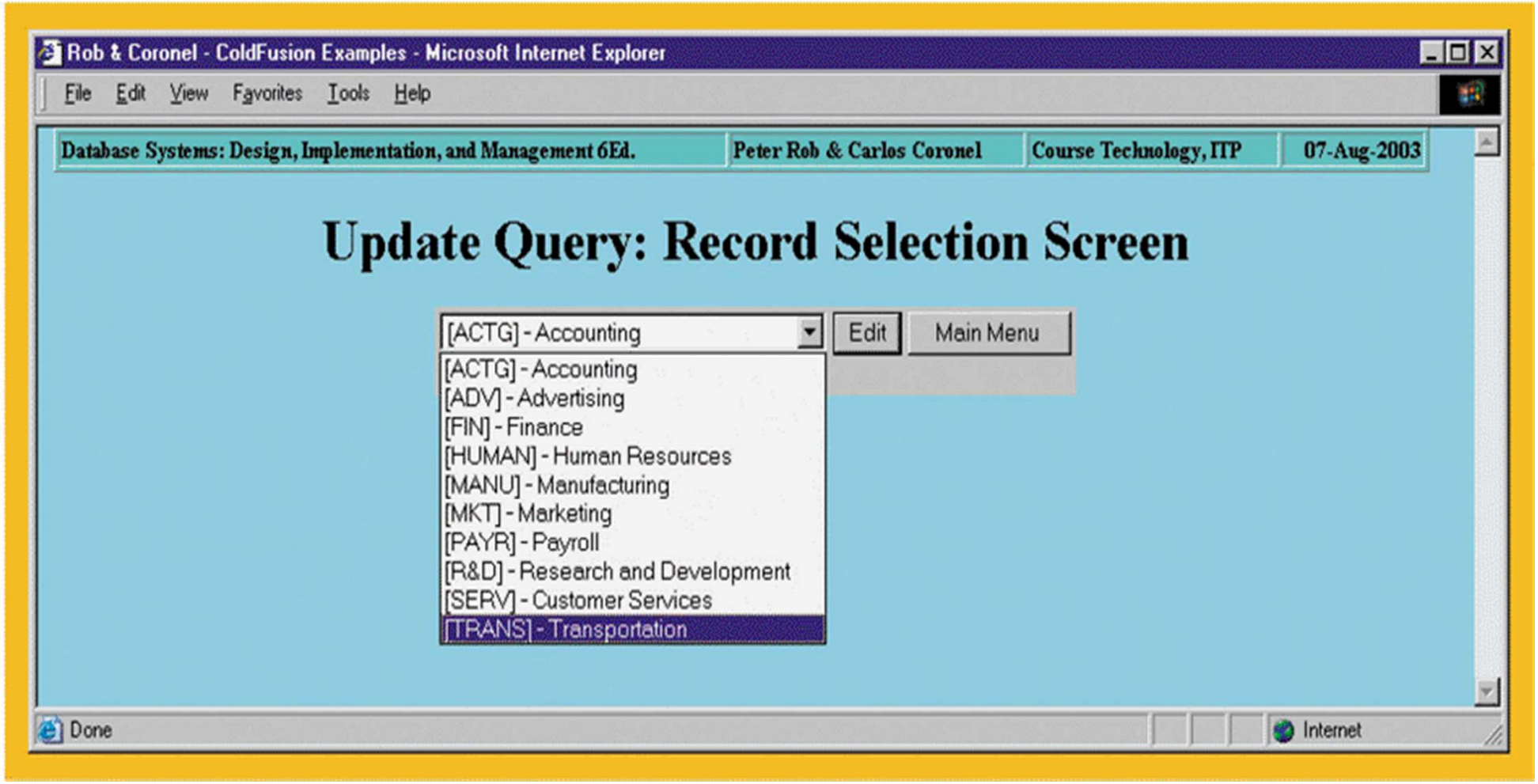
Update Query: Record Selection Script

SCRIPT 14.6A (RC-6A.CFM): UPDATE QUERY: RECORD SELECTION SCRIPT

```
<HTML>
<HEAD>
<TITLE>Rob & Coronel -  ColdFusion Examples</TITLE>
<CFQUERY NAME="Deptlist" DATASOURCE="RobCor">
    SELECT * FROM DEPARTMENT ORDER BY DEPT_ID
</CFQUERY>
</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>
    <CFOUTPUT QUERY="Deptlist">
        <OPTION VALUE="#DEPT_ID#">[#DEPT_ID#] - #DEPT_DESC#
    </CFOUTPUT>
    </SELECT>
    <INPUT TYPE=HIDDEN NAME="DEPT_ID_required" VALUE="DEPT_ID is required">
</TD>
<TD>
<INPUT TYPE="submit" VALUE=" Edit ">
</FORM>
</TD>
<TD>
<FORM ACTION="rc-0.cfm" METHOD="post">
    <INPUT TYPE="submit" VALUE="Main Menu ">
</FORM>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

The rc-6a.cfm Script Output (Record Selection Screen)

FIGURE 14.17 THE RC-6A.CFM SCRIPT OUTPUT (RECORD SELECTION SCREEN)



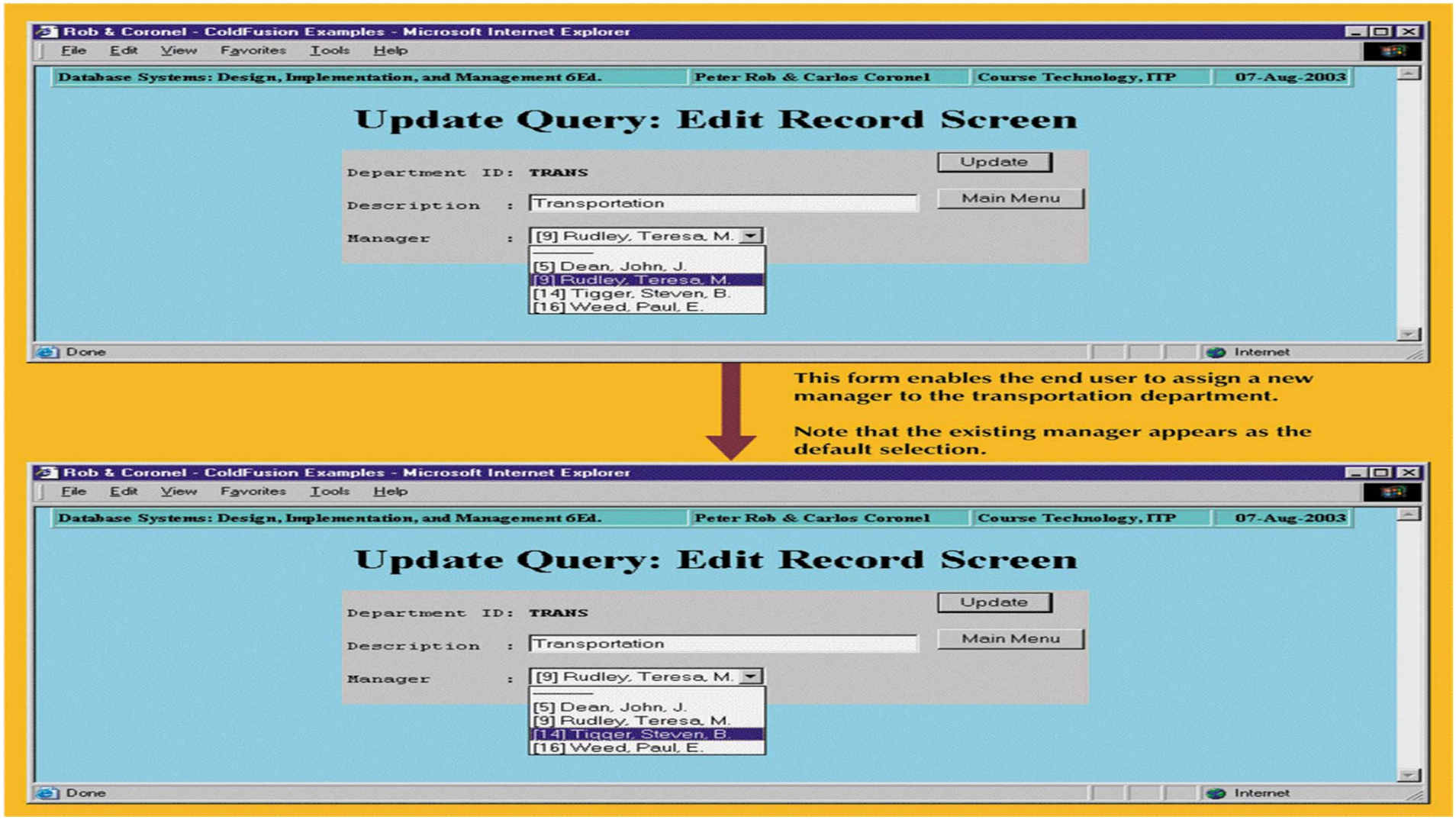
Update Query: Edit Record Script

SCRIPT 14.6B (RC-6B.CFM): UPDATE QUERY: EDIT RECORD SCRIPT

```
<HTML>
<HEAD>
<TITLE>Rob & Coronel -  ColdFusion Examples</TITLE>
<CFQUERY NAME="DeptData" DATASOURCE="RobCor">
    SELECT * FROM DEPARTMENT WHERE (DEPARTMENT.DEPT_ID='#form.DEPT_ID#')
</CFQUERY>
<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 AND DEPT_ID <> '#form.DEPT_ID#')
      ORDER BY USR_LNAME, USR_FNAME, USR_MNAME
</CFQUERY>
</HEAD>
<BODY BGCOLOR="LIGHTBLUE">
<H1>
<CENTER><B>Update Query: Edit Record Screen</B></CENTER>
</H1>
<FORM ACTION="rc-6c.cfm" METHOD="post">
<TABLE ALIGN="CENTER" BGCOLOR="Silver" BORDERCOLOR="Blue">
<TR>
<TD>
<CFOUTPUT QUERY="DeptData">
<PRE>
<INPUT TYPE="hidden" NAME="DEPT_ID" VALUE="#DEPTDATA.DEPT_ID#">
Department ID: <B>#DEPT_ID#</B><BR>
Description  : <INPUT TYPE="text" NAME="DEPT_DESC" VALUE="#DEPT_DESC#"SIZE=35 MAXLENGTH=35><BR>
Manager      : <SELECT NAME="USR_ID" SIZE=1><!--- select user from list --->
               <OPTION <CFIF #DEPTDATA.USR_ID# EQ "">SELECTED</CFIF> VALUE="">----------
</CFOUTPUT>
<CFOUTPUT QUERY="USRLIST">
               <OPTION <CFIF #DEPTDATA.USR_ID# EQ #USRLIST.USR_ID#>SELECTED</CFIF> VALUE="#UsrList.USR_ID#">
               [#USR_ID#] #USR_LNAME#, #USR_FNAME#, #USR_MNAME#
</CFOUTPUT>
                </SELECT>
</PRE>
</TD>
<TD VALIGN="TOP">
<INPUT TYPE="submit" VALUE=" Update ">
</FORM>
<FORM ACTION="rc-0.cfm" METHOD="post">
    <INPUT TYPE="submit" VALUE="Main Menu">
</FORM>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

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

FIGURE 14.18 THE RC-6B.CFM SCRIPT OUTPUT (EDIT RECORD SCREEN)



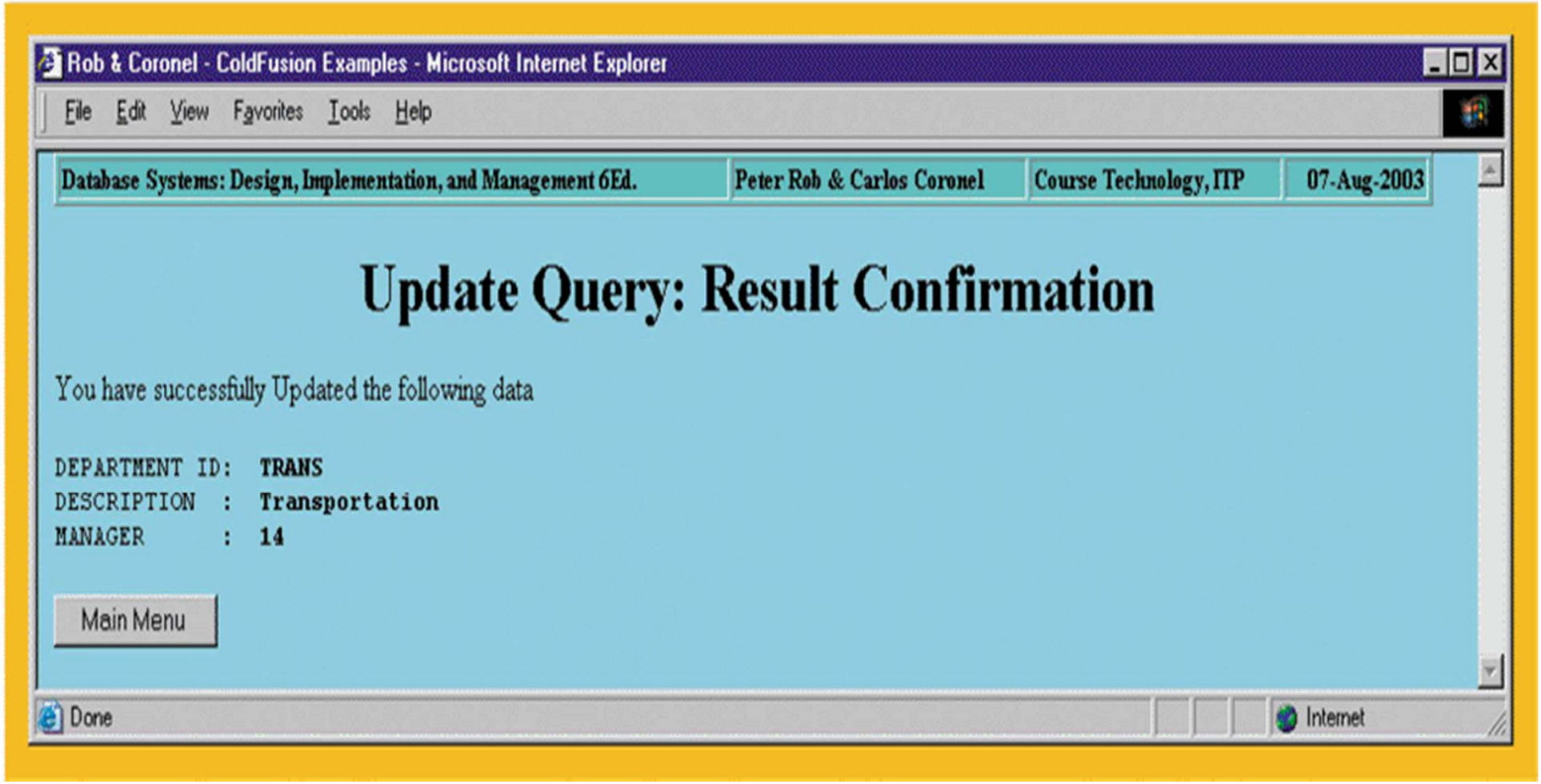
Update Query: Result Confirmation Script

SCRIPT 14.6C (RC-6C.CFM): UPDATE QUERY: RESULT CONFIRMATION SCRIPT

```
<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>
<CFOUTPUT>
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>
</CFOUTPUT>
<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

SCRIPT 14.7A (RC-7A.CFM): DELETE QUERY: RECORD SELECTION SCRIPT

```
<HTML>
<HEAD>
<TITLE>Rob & Coronel -  ColdFusion Examples</TITLE>
<CFQUERY NAME="Deptlist" DATASOURCE="RobCor">
    SELECT * FROM DEPARTMENT ORDER BY DEPT_ID
</CFQUERY>
</HEAD>
<BODY BGCOLOR="LIGHTBLUE">
<H1>
<CENTER><B>Delete Query: Record Selection Screen</B></CENTER>
</H1>
<TABLE ALIGN="CENTER" BGCOLOR="Silver">
<TR VALIGN="TOP">
<TD>
<FORM ACTION="rc-7b.cfm" METHOD="post">
    <SELECT NAME="DEPT_ID" SIZE=1>
    <CFOUTPUT QUERY="Deptlist">
        <OPTION VALUE="#DEPT_ID#">[#DEPT_ID#] - #DEPT_DESC#
    </CFOUTPUT>
    </SELECT>
    <INPUT TYPE=HIDDEN NAME="DEPT_ID_required" VALUE="DEPT_ID is required">
</TD>
<TD>
    <INPUT TYPE="submit" VALUE="Delete">
</FORM>
</TD>
<TD>
<FORM ACTION="rc-0.cfm" METHOD="post">
    <INPUT TYPE="submit" VALUE="Main Menu">
</FORM>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

The rc-7a.cfm Script Output (Record Selection Screen)

FIGURE 14.20 THE RC-7A.CFM SCRIPT OUTPUT (RECORD SELECTION SCREEN)

Delete Query: Show Record Script

SCRIPT 14.7B (RC-7B.CFM): DELETE QUERY: SHOW RECORD SCRIPT

```
<HTML>
<HEAD>
<TITLE>Rob & Coronel -  ColdFusion Examples</TITLE>
</HEAD>
<CFQUERY NAME="DeptData" DATASOURCE="RobCor">
    SELECT * FROM DEPARTMENT WHERE (DEPARTMENT.DEPT_ID='#form.DEPT_ID#')
</CFQUERY>
<CFIF #DEPTDATA.USR_ID# IS NOT "">
    <CFQUERY NAME="Usrdata" DATASOURCE="RobCor">
        SELECT USR_ID, USR_LNAME, USR_FNAME, USR_MNAME FROM USER
         WHERE (USER.USR_ID = #deptdata.usr_id#)
    </CFQUERY>
</CFIF>
<CFQUERY NAME="UsrTot" DATASOURCE="RobCor">
    SELECT COUNT(*) AS T1 FROM USER
      WHERE (USER.DEPT_ID = '#form.DEPT_ID#')
</CFQUERY>
</HEAD>
<BODY BGCOLOR="LIGHTBLUE">
<H1><CENTER><B>Delete Query: Show Record Screen</B></CENTER></H1>
<FORM ACTION="rc-7c.cfm" METHOD="post">
<CFOUTPUT QUERY="DeptData">
<INPUT TYPE="hidden" NAME="DEPT_ID" VALUE="#deptdata.DEPT_ID#">
<INPUT TYPE="hidden" NAME="DEPT_DESC" VALUE="#deptdata.DEPT_DESC#">
<INPUT TYPE="hidden" NAME="USR_ID" VALUE=#DEPTDATA.USR_ID#>
<TABLE ALIGN="CENTER" BGCOLOR="Silver" BORDERCOLOR="Blue">
<TR>
<TD>
<PRE>
Department ID: #DEPT_ID#<BR>
Description  : #DEPT_DESC#<BR>
Manager      : <CFIF #DEPTDATA.USR_ID# IS NOT "">#Usrdata.USR_LNAME# #Usrdata.USR_FNAME# #Usrdata.USR_MNAME#</CFIF>
</CFOUTPUT>
</PRE>
</TD>
<TD VALIGN="TOP">
<CFIF #USRTOT.T1# EQ 0>
    <INPUT TYPE="submit" VALUE=" Delete ">
<CFELSE>
    <SMALL><B>We cannot delete this record <BR>because there are dependent <BR>users assigned to this department</B></SMALL>
</CFIF>
</FORM>
<FORM ACTION="rc-0.cfm" METHOD="post"><INPUT TYPE="submit" VALUE="Main Menu"></FORM>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

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

FIGURE 14.21 THE RC-7B.CFM SCRIPT OUTPUT (SHOW RECORD SCREEN)

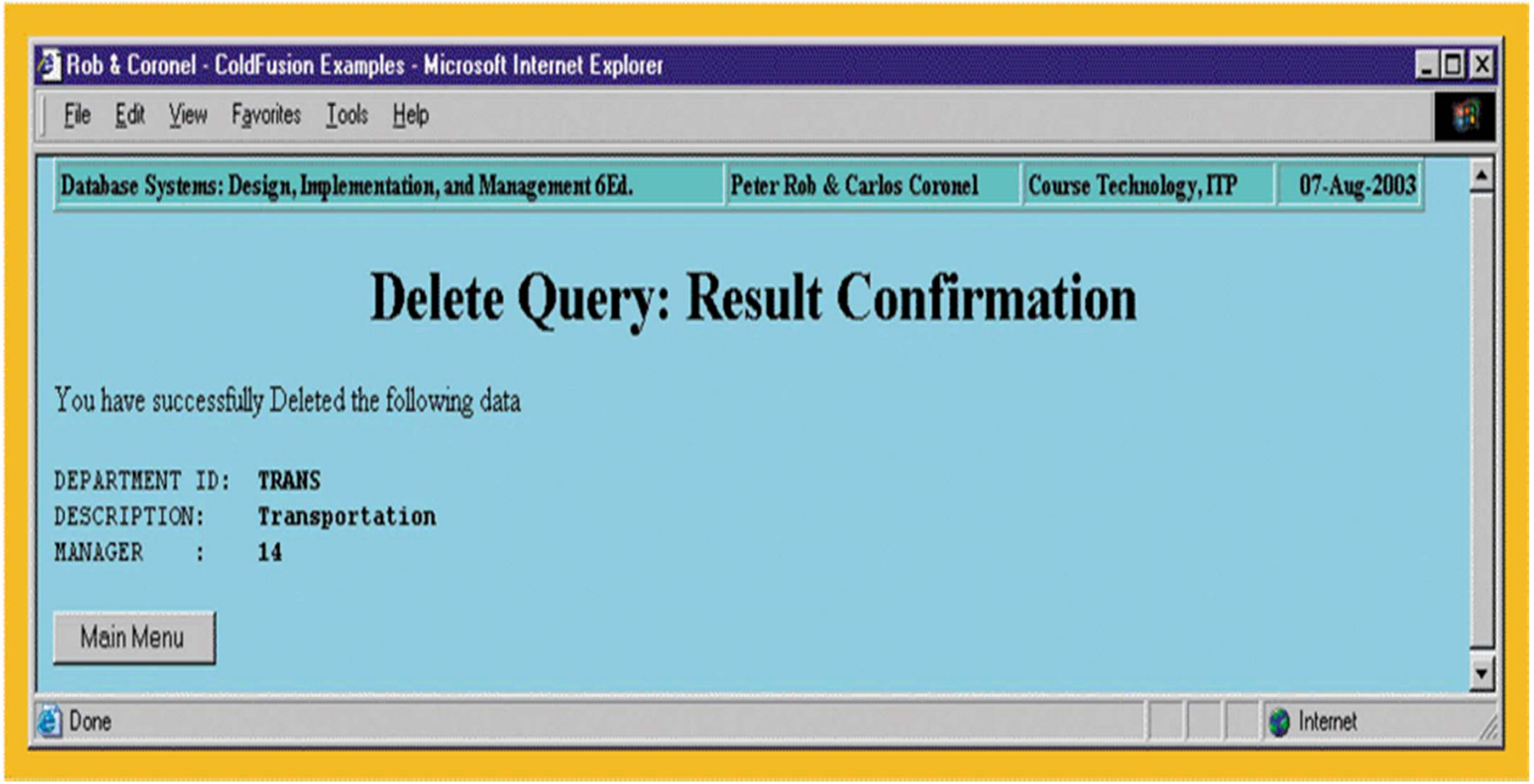
Delete Query: Result Confirmation Script

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

```
<HTML>
<HEAD>
<TITLE>Rob & Coronel -  ColdFusion Examples</TITLE>
<CFQUERY NAME="DeleteDept" DATASOURCE="RobCor">
    DELETE FROM DEPARTMENT WHERE (DEPT_ID = '#FORM.DEPT_ID#')
</CFQUERY>
</HEAD>
<BODY BGCOLOR="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>
```

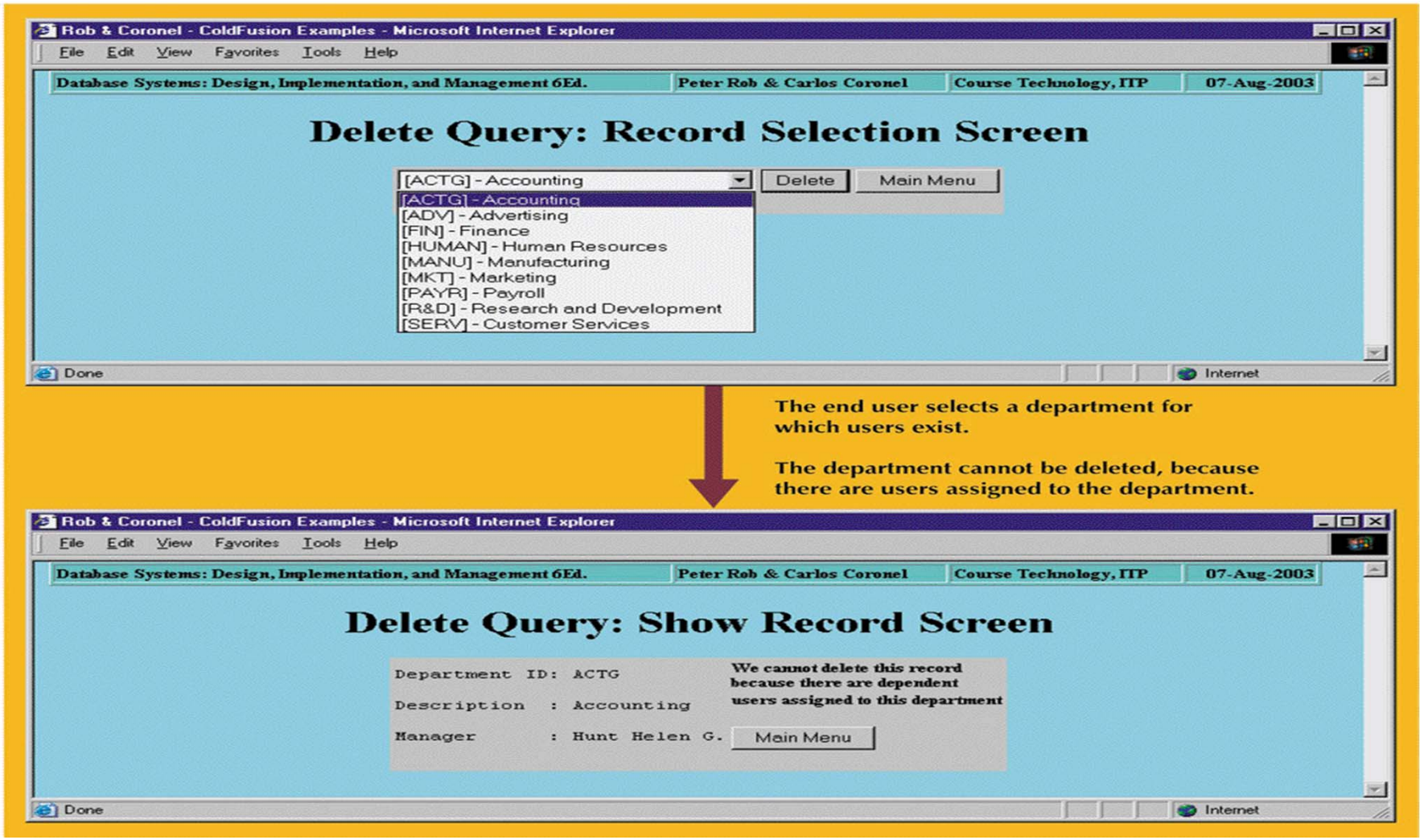
The rc-7c.cfm Script Output (Delete Query Confirmation Screen)

FIGURE 14.22 THE RC-7C.CFM SCRIPT OUTPUT (DELETE QUERY CONFIRMATION SCREEN)



The Delete Record Validation

FIGURE 14.23 THE DELETE RECORD VALIDATION



E-Business / E-Commerce

What is Electronic Commerce?

- Use of electronic networked computer-based technology to:
  - Bring new products, services, or ideas to market
  - Support and enhance business operations (including sales of products/services over the Web)
- Most e-commerce transactions take place among businesses
- Now recognized as a prime revenue source

The Road to Electronic Commerce

- Key to e-commerce is using computer networks, especially the Internet, to automate and streamline business transactions
- 1960s: banks created private telephone network to do electronic funds transfers
- 1970s: banks created services to provide after hours services to their customers
- Late 1970s and early 1980s: Electronic Data Interchange (EDI) emerged
  - Communications protocol that enabled companies to exchange business documents over private phone networks

The Road to Electronic Commerce (2)

- Early 1980s and through the 1990s: personal computer facilitated rapid expansion of the Internet and ultimately provided the spark that led to the explosive use of the World Wide Web
- Late 1990s and early 2000s: networking technologies blossomed and expanded the reach, speed, and in some cases, security of Internet-based communications and transactions

E-Commerce Benefits

- Easy comparison shopping
- Reduced costs and increased competition
- Convenience
- 24 × 7 × 365 operation
- Global access
- Lower entry barriers
- Increased market (customer) knowledge

E-Commerce Disadvantages

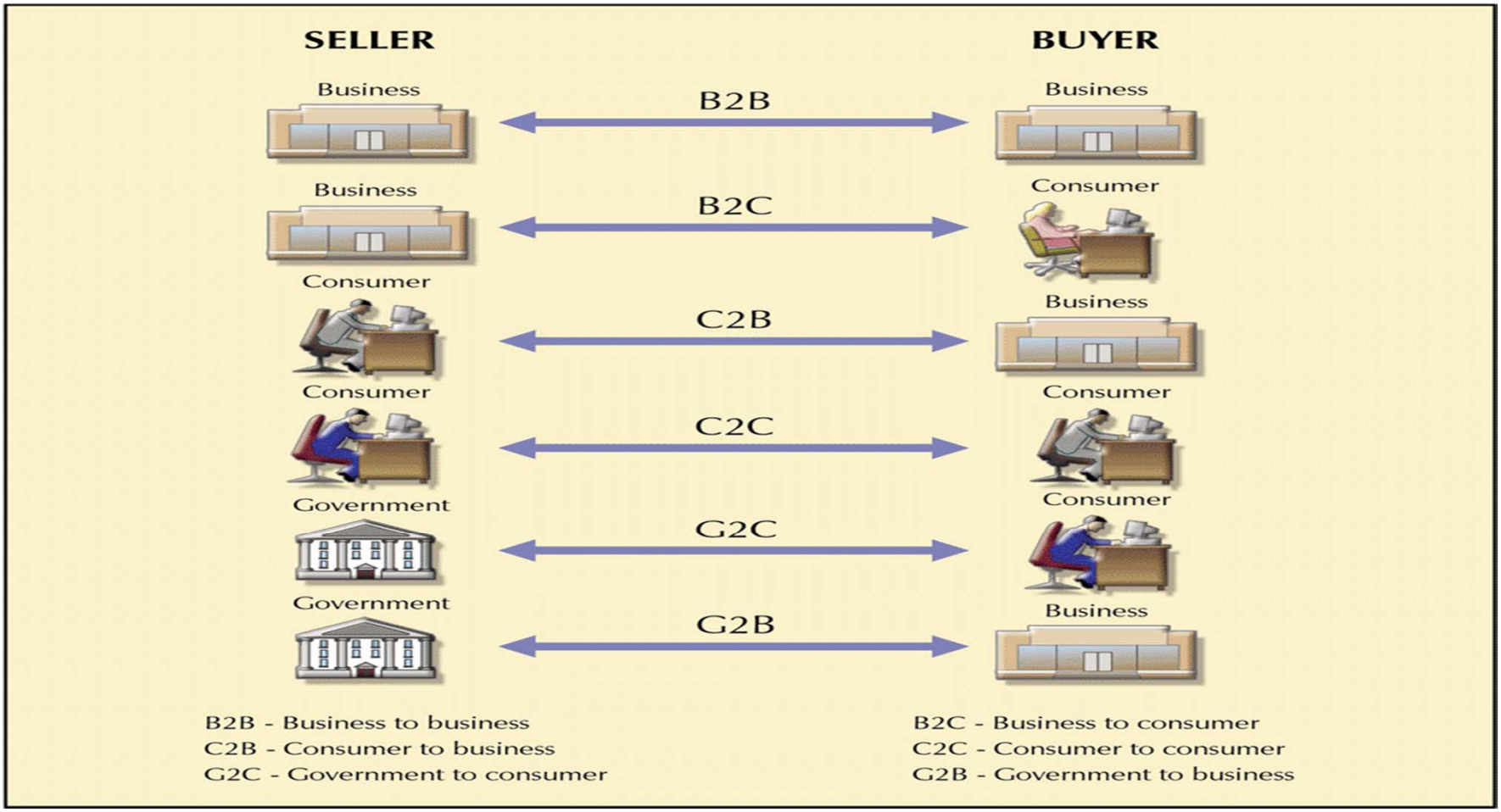
Technical Disadvantages

- Lack of system security, reliability or standards owing to poor implementation of e-Commerce
- Software development industry is still evolving and keeps changing rapidly.
- In many countries, network bandwidth might cause an issue as there is insufficient telecommunication bandwidth available.
- Special types of web server or other software might be required by the vendor setting the e-commerce environment apart from network servers.
- Sometimes, it becomes difficult to integrate E-Commerce software or website with the existing application or databases.
- There could be software/hardware compatibility issue as some E-Commerce software may be incompatible with some operating system or any other component.

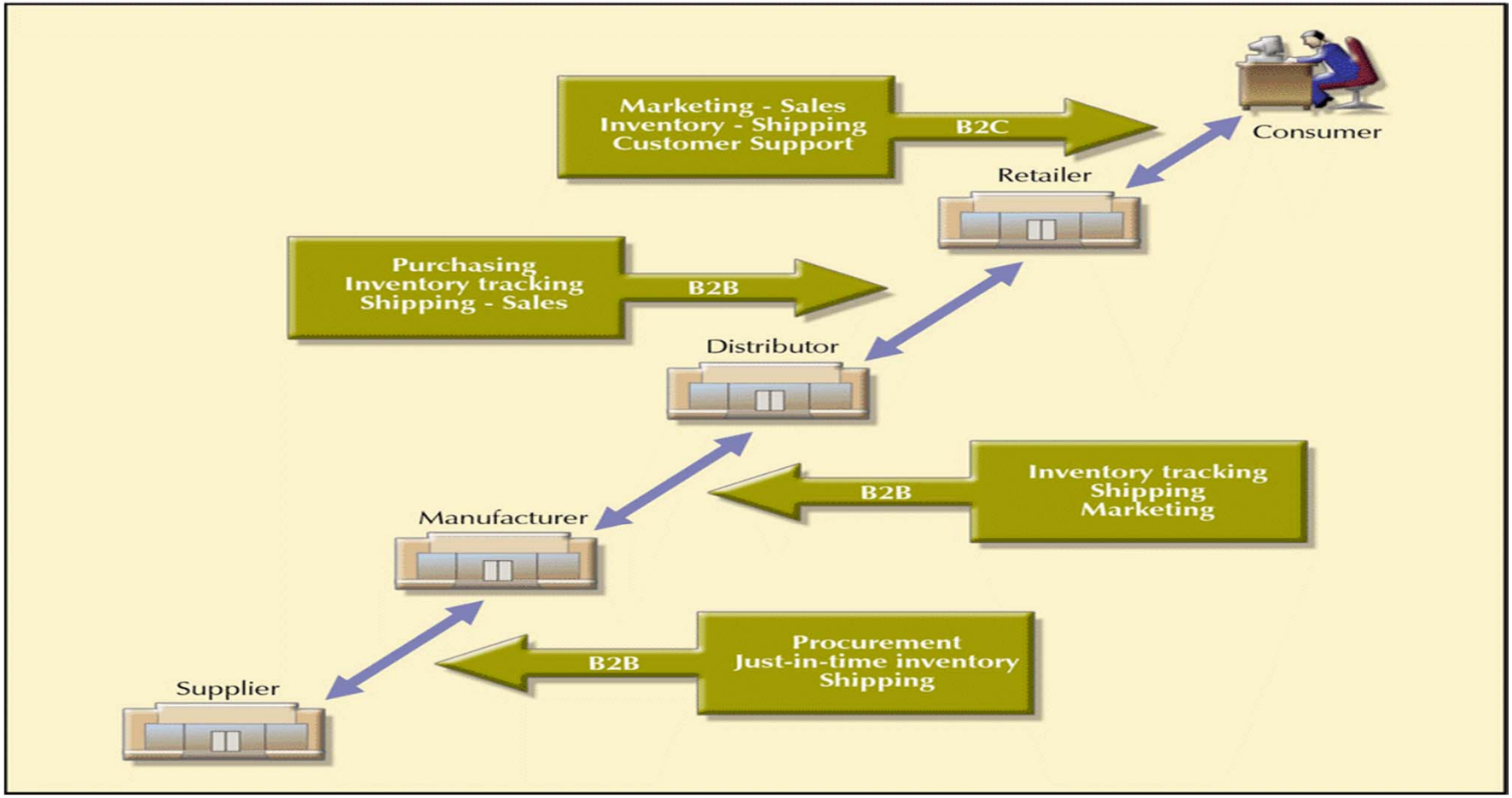
Non-Technical Disadvantages

- Initial cost: The cost of creating/building E-Commerce application in-house may be very high + Maintenance cost + Cost of staying in business + Hidden cost
- There could be delay in launching the E-Commerce application due to mistakes, lack of experience.
- User resistance: User may not trust the site being unknown faceless seller. Such mistrust makes it difficult to make user switch from physical stores to online/virtual stores.
- Security/Privacy: Difficult to ensure security or privacy on online transactions.
- Lack of touch or feel of products during online shopping.
- Internet access is still not cheaper and is inconvenient to use for many potential customers like one living in remote villages.
- Possible legal issues

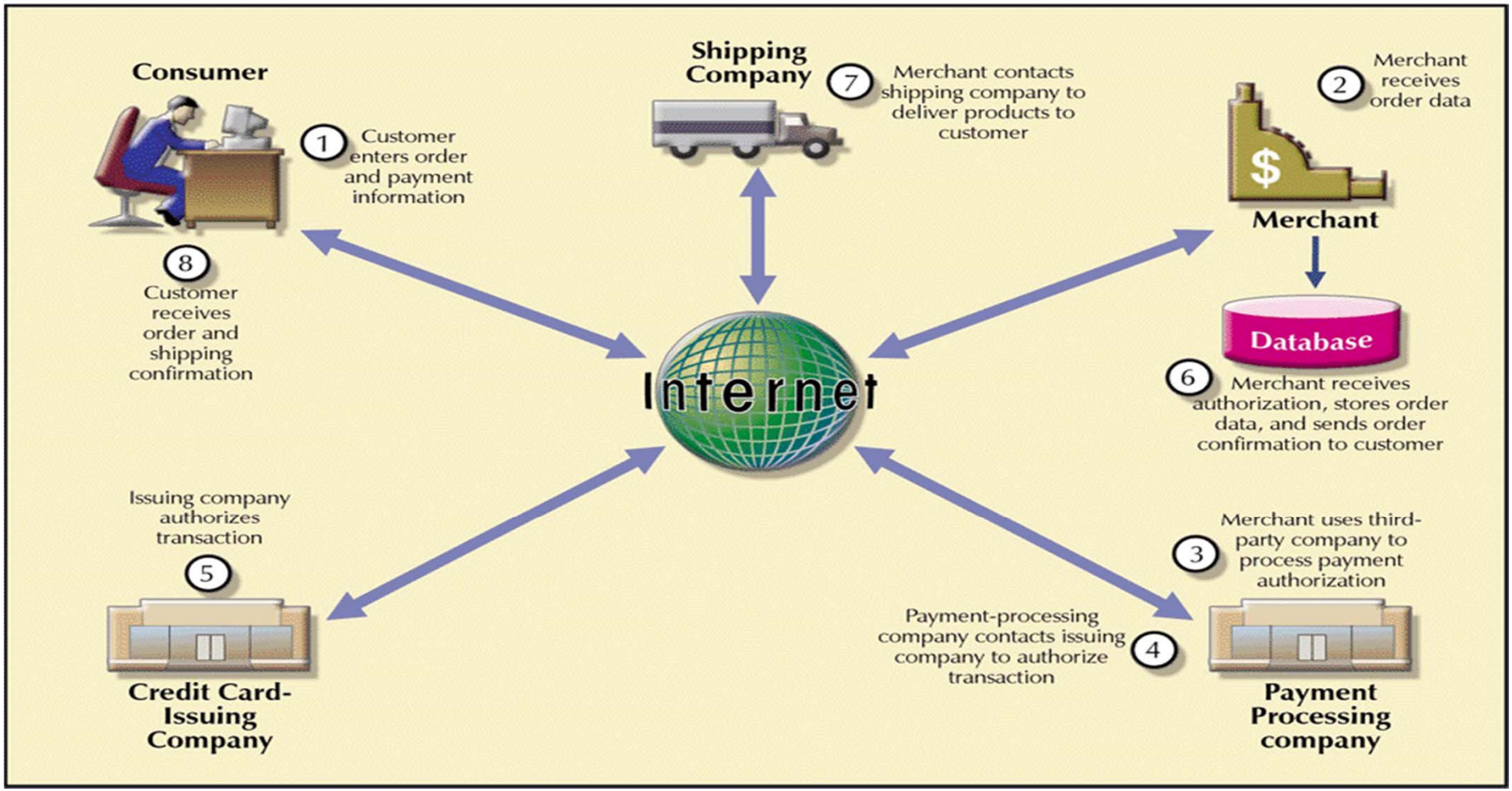
E-Commerce Styles

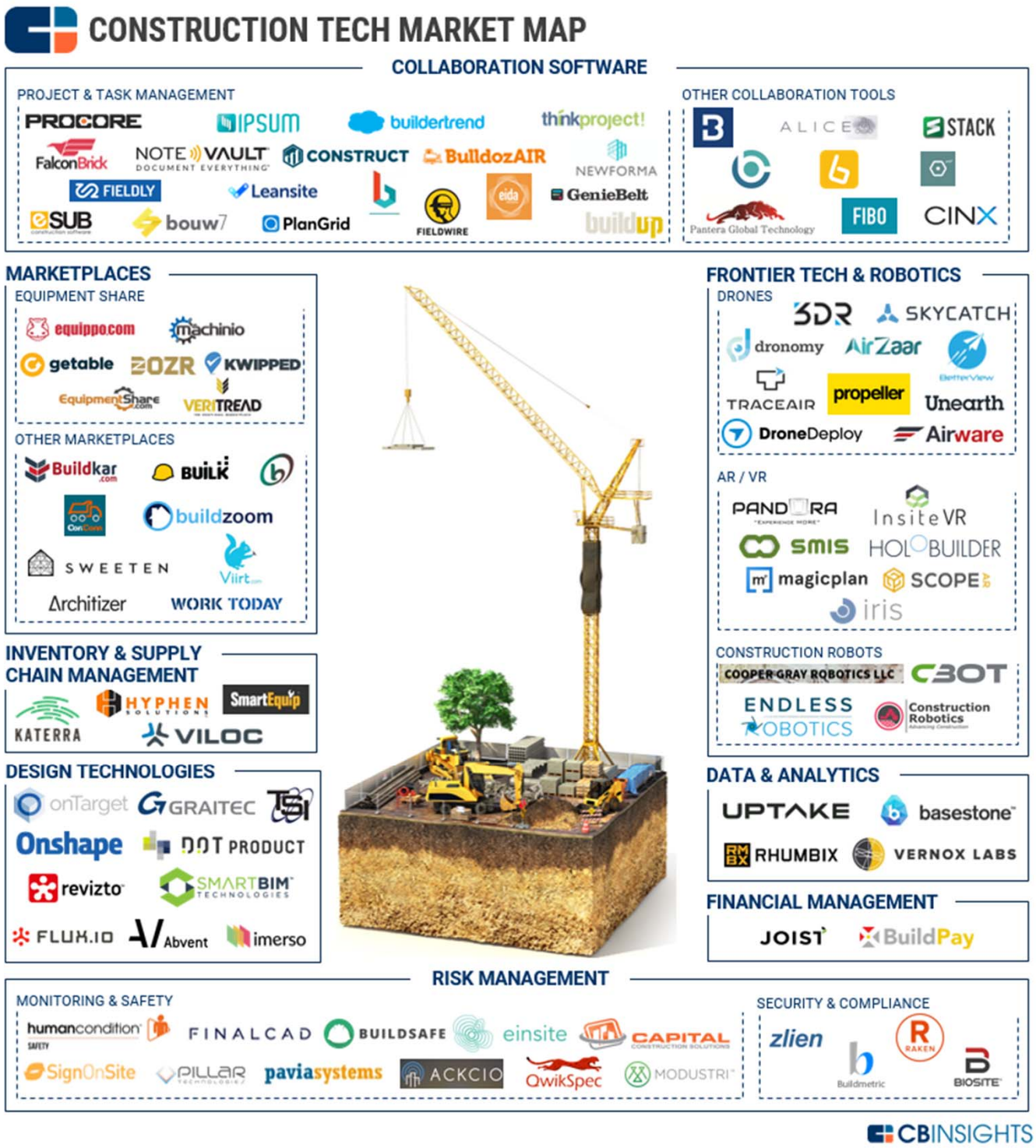


E-Commerce Automation of Supply Chain



A Sample E-Commerce Transaction





미국 CB insight의 100대 건설 스타트업

- **Collaboration Software**
  - Project & Task Management
  - Other Collaboration Tools
- **Market Places**
  - Equipment Share
  - Other Market Places
- **Inventory & Supply Chain Management**
- **Design Technologies**
- **Risk Management**
  - Monitoring & Safety
  - Security & Compliance
- **Financial Management**
- **Data & Analytics**
- **Frontier Tech & Robotics**
  - Drones
  - AR/VR
  - Construction Robots

Database Design for E-commerce Applications

- Define scope of database
- Establish some basic business rules and their effect(s) on the design
- Define tables required to support the e-commerce activities
- Identify basic attributes for each table

Extensible Markup Language (XML)

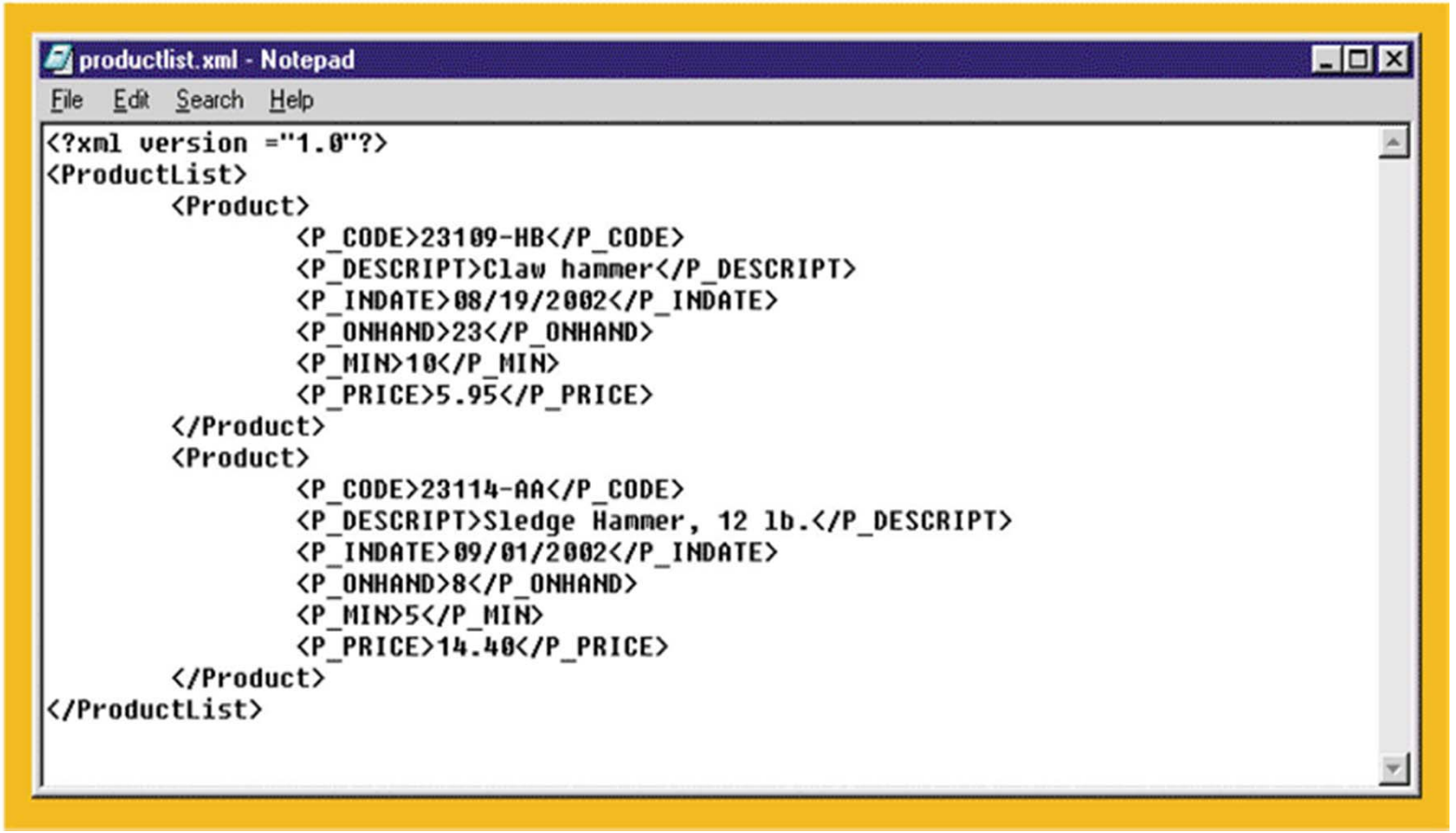
Markup: 웹 페이지를 만드는데 사용되는 프로그래밍 언어로 태그라는 요소를 사용하여 구성
HTML: Hypertext Markup Language
태그: 해당영역이 어떤 역할을 하는지 웹 브라우저의 렌더링 엔진에 전달(예. p, /p태크로 감싸진 문장: 문단으로 인식)

인터넷에 연결된 시스템끼리 데이터를 쉽게 주고 받을 수 있게 하여 HTML의 한계를 극복 (상호보완적 관계)
XML: 검색효율을 높이기 위한 데이터의 효율적 추출, 전자 상거래 데이터 전송, 웹 페이지와 DB간 데이터 교환 등

(HTML)데이터 표현에 목적 VS (XML)데이터 교환을 위한 구조정의에 목적
(HTML)정해진 테그를 가지고 표현 VS (XML)사용자가 태그를 정의해서 사용 (예. <ProductId>12-AA</ProductID>)
(HTML)인터넷 웹 환경에서 작동(웹 브라우저에서 실행) VS (XML)환경적 구애를 받지 않음(무선환경에서도 가능)
(HTML)데이터를 표현 VS (XML)데이터와 그 구조만을 가지고 있을 뿐 화면에 표현하지 않음

- Meta-language used to represent and manipulate data elements
- Designed to facilitate the exchange of structured documents such as orders or invoices over the Internet
- World Wide Web Consortium published the first XML 1.0 standard definition in 1998

ProductList.xml Document



The first line represents the XML document declaration and it is mandatory

Every XML document has a root element. The second line declares the ProductList root element

The root element contains child elements or sub-elements. Line 3 declares Product as a child element of ProductList

Each element can contain sub-elements. Each Product element is composed of P_CODE, etc.

The XML document reflects a hierarchical tree structure where elements are related in a parent-child relationship: each parent element can have many children elements.

ProductList.dtd Document

productlist.dtd - Notepad

File Edit Search Help

```
<!ELEMENT ProductList (Product+)>
<!ELEMENT Product (P_CODE, P_DESCRIPT, P_INDATE?, P_ONHAND, P_MIN?,P_PRICE)>
<!ELEMENT P_CODE        (#PCDATA )>
<!ELEMENT P_DESCRIPT    (#PCDATA )>
<!ELEMENT P_INDATE      (#PCDATA )>
<!ELEMENT P_ONHAND      (#PCDATA )>
<!ELEMENT P_MIN         (#PCDATA )>
<!ELEMENT P_PRICE       (#PCDATA )>
```

dtd(Document Type Definition): composition of the database's logical model, must develop and share for data sharing

The first line declares the ProductList root element.

The ProductList root element has on child , the Product element

The plus symbol indicates that Product occurs one or more times within ProductList

An asterisk would mean that the child element occurs zero or more times

A question mark would mean that the child element is optional

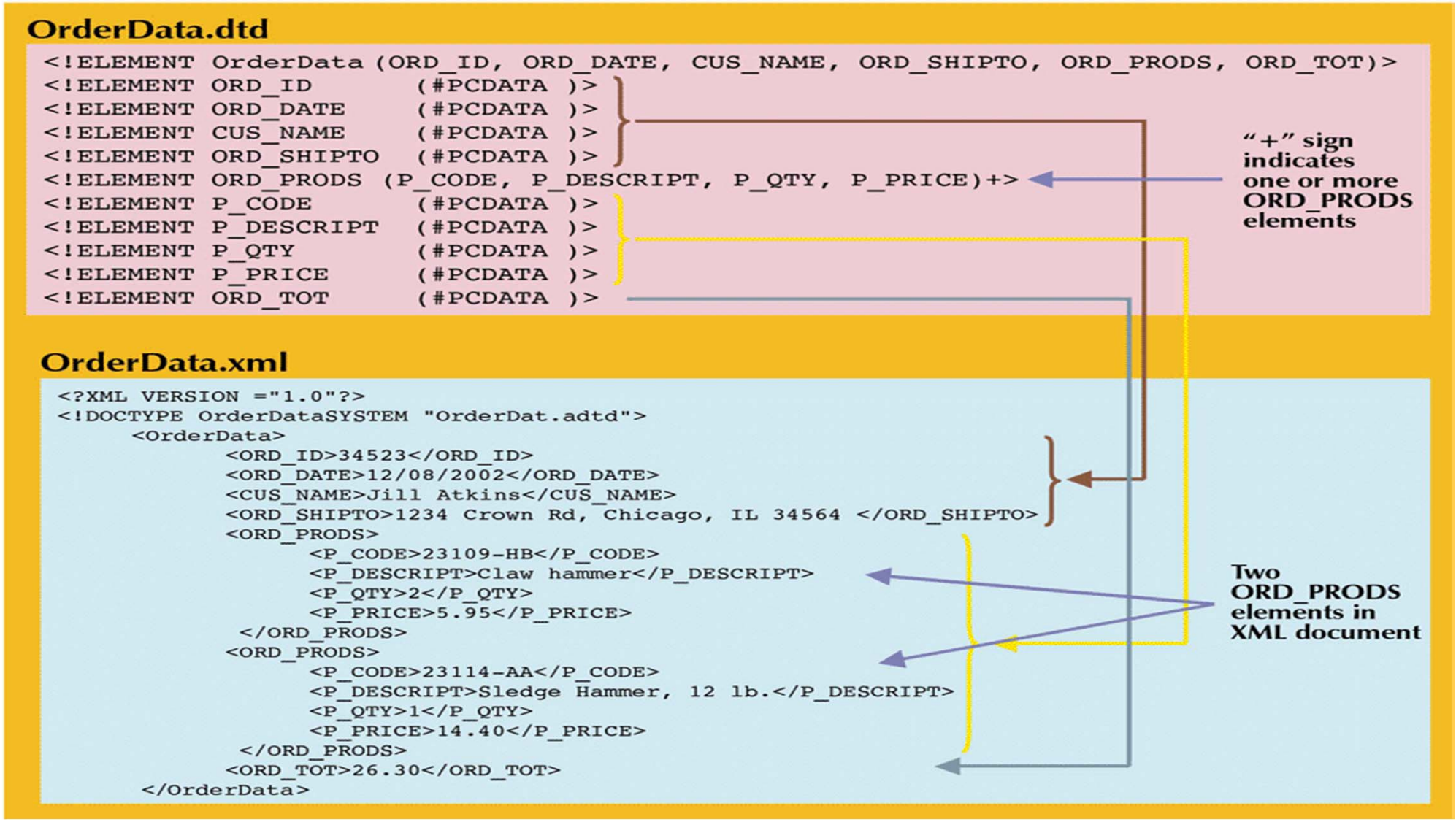
The second line describes the Product element

The third through eighth lines show that the Product element has six children sub-elements

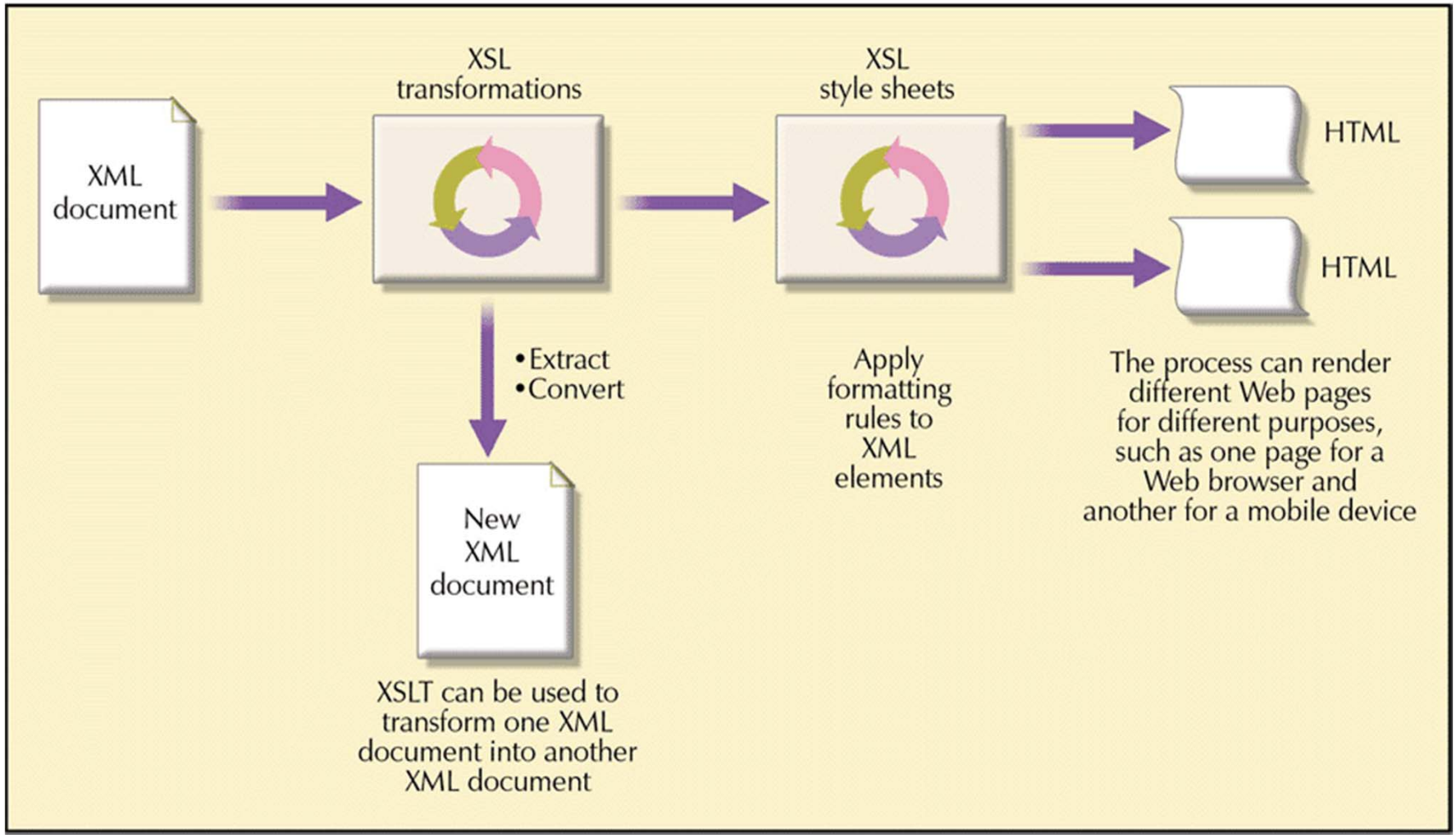
The #PCDATA keyword represents the actual text data

In the previous example: Add second line <!DOCTYPE ProductList SYSTEM "ProductList.dtd">

DTD and XML Documents for Order Data



Framework for XML Transformation



XSL(Extensible Style Language)*: specification used to define the rules by which XML data are formatted and displayed*

XSLT(Extensible Style Language Transformation)*: extract data from an XML document → convert it into a text file, an HTML Web page, or a Web page formatted for a mobile device. User sees an actual view(or HTML representation) of the actual XML data*

e.g. XSLT can be used to extract product codes and product prices to create a product catalog

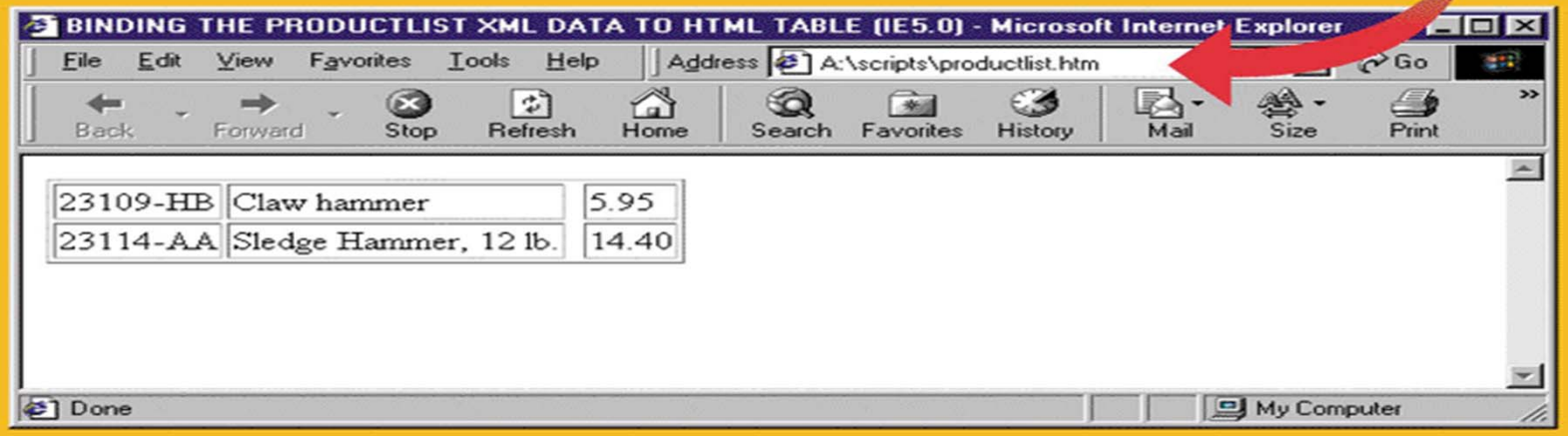
XSL Style Sheet*: define the presentation rules applied to XML elements like presentation templates*

XML Data Binding

productlist.htm - Notepad

File Edit Search Help

```
<HTML>
<HEAD>
<TITLE>BINDING THE PRODUCTLIST XML DATA TO HTML TABLE (IE5.0)</TITLE>
</HEAD>

<BODY>

<XML ID="PRODLIST" SRC="PRODUCTLIST.XML"></XML>

<TABLE BORDER="1" DATASRC="#PRODLIST">
        <TR>
        <TD><SPAN DATAFLD="P_CODE"></SPAN></TD>
        <TD><SPAN DATAFLD="P_DESCRIPT"></SPAN></TD>
        <TD><SPAN DATAFLD="P_QOH"></SPAN></TD>
        <TD><SPAN DATAFLD="P_PRICE"></SPAN></TD>
        </TR>

</TABLE>

</BODY>
</HTML>
```



Data binding of XML data to HTML documents: Bind an XML document to an HTML table!

<xml> tag to include the XML data in the HTML document to later bind it to the HTML table

ID: 이름정의, SRC: 연결할 외부 파일명

Mobile & Pervasive Computing

Mobile Computing

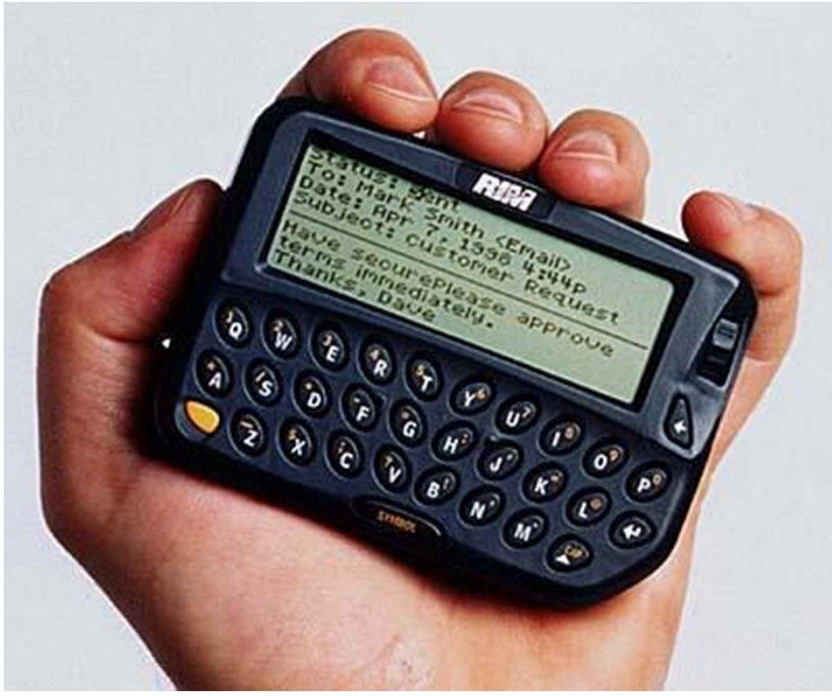
In the traditional computing environment it was necessary to come to the computer to do some work on it. All computers were connected to each other, to networks, servers, etc. via wires.

- The first phase was to make computers small enough so they can be easily carried - Mobile devices
- The second solution to the need for mobile computing was to replace wires with wireless communication media.
- The third phase was a combination of the first two, namely to use mobile devices in a wireless environment. Referred to as wireless mobile computing, this combination enables real-time connections between mobile devices and other computing environments.

Ubiquitous Computing – computing anytime anywhere

Mobile Computing Infrastructure – Hardware

- Cellular phones
- Personal digital assistants (PDAs)
- Interactive pagers
- Other



Mobile Computing Infrastructure – Hardware

M-commerce also requires the following hardware which is essential for wireless connectivity:

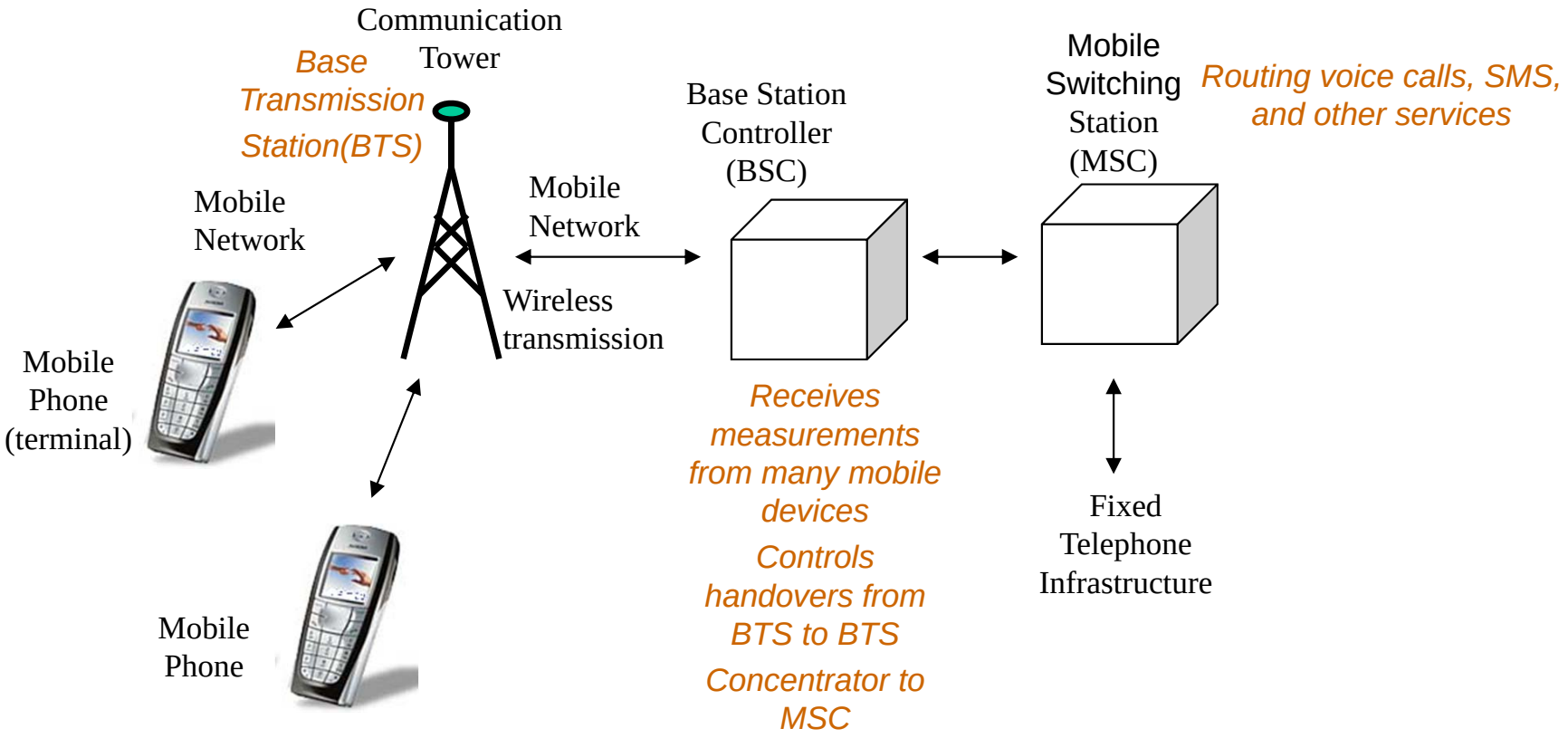
- A WAN modem
- A wireless LAN or MAN (metro-area network) adapter
- A Web server with wireless support
- A WAP gateway
- A communications server
- An application or database server
- An enterprise application server
- A GPS locator

Mobile Computing Infrastructure – Software

Software	Description
Microbrowser	A browser with limited bandwidth and memory requirements. Provides wireless access to the Internet
Operating system (OS) for mobile-client	An OS for mobile devices. Examples: Palm OS, Pocket PC, Win CE. Specialized OS's: Blackberry and Web browser.
Bluetooth	Chip technology for short-range communication among wireless devices. See *bluethooth.com.*
User interface	Application logic for handheld devices.
Application middleware	Provides connecting among applications, databases, and Web-based servers.
Wireless middleware	Links wireless networks to application servers.
Wireless Application Protocol (WAP)	A set of communication protocols that enables wireless devices to "talk" to a server on a mobile network, so users can access the Internet. Specially designed for small screen. (see *wapforum.org*).
Wireless Markup Language	An XML-based scripting language for creating content for wireless systems.
Voice XML	An extension of XML designed to accommodate voice.

Mobile Computing Infrastructure – WWAN's

At the core of most mobile computing applications are mobile networks. These are of two general types: the wide area and the local area. The wide area networks for mobile computing are known as **wireless wide area networks** (WWAN).



Mobile Computing Infrastructure – WLAN's

Wireless local area networks (**WLAN):**

- **Wireless access point** - a transmitter with an antenna, connected to a wired LAN that provides an Internet connection. (A wireless access point provides service to a number of users within a small geographical perimeter known as a "hot spot")
- Wireless network card incorporated with laptops, desktops, or PDAs will provide access
- WLAN's employ the Wi-Fi (wireless fidelity) standard developed by the IEEE
  - 802.11b Speeds up to 11Mbps
  - 802.11a and 802.11g Speeds up to 54 Mbps
  - Wireless Encryption Protocol (WEP) a built-in security system in Wi-Fi encrypts the communications between a client machine and a wireless access point.

Mobile Computing – Shopping

Shopping from wireless devices enables customers to perform quick searches, compare prices, use a shopping cart, order, and view the status of their order using their mobile wireless devices.

Some shopping applications include:

- Restaurant chains enabling consumers to place an order for pick up or delivery virtually any time, anywhere.
- eBay offers “anywhere wireless” services as does Amazon.com
- Purchasing movie tickets by wireless device

Mobile Computing – Advertising

Knowing the current location of mobile users (using GPS) and their preferences or surfing habits, marketers can send user-specific advertising messages to wireless devices.

This location-sensitive advertising, will informing a user about:

- sales at a specific shop or mall
- today's specials at a restaurant
- loyalty programs
- and much more

all when a potential buyer is within close proximity.

Mobile Computing – Mobile Portals

These are customer channels, optimized for mobility, that aggregates and provides content and services to mobile users.

The services provided by mobile portals include:

- News
- Sports
- E-mail
- Entertainment
- Travel information
- Restaurants
- Event information
- Leisure-related services (e.g., games, TV and movie listings)
- Community services
- Stock trading

Mobile Computing – Enterprise Applications

- **Support Of Mobile Workers:** are those working outside the corporate premises, such as service technicians, sales personnel, delivery workers, etc.
- **Wearable Devices:** Employees may be equipped with a special form of mobile wireless computing devices
  - Camera
  - Screen
  - Keyboard/Touch-panel display
  - Speech translator

Mobile Computing – Enterprise Applications

- **Job Dispatch**. To assign jobs to mobile employees, along with info about the task.
  - transportation (delivery of food, oil, newspapers, cargo, courier services)
  - Utilities measurement (gas, electricity, phone, water)
  - Field service (computer, office equipment, home repair)
  - Health care (visiting nurses, doctors, social services)
  - Security (patrols, alarm installation).

Mobile Computing – Location-based Commerce

Location-based commerce (L-commerce) refers to the localization of products and services. From a consumer's viewpoint, L-commerce offers safety. From a business supplier's point of view, L-commerce offers an opportunity to provide services that meet customers' needs.

ONCE A WIRELESS DEVICE IS DETECTED IN A LOCATION, ADVERTISEMENT IS DIRECTED TO THE DEVICE!

(e.g., Syrup, OK Cashback, Korail, ……)

The L-commerce services revolve around five key areas:

- ***Location***: determining the basic position of a person or a thing (e.g., car or boat).
- ***Navigation***: plotting a route from one location to another.
- ***Tracking***: monitoring the movement of a person or a thing (e.g., a package or vehicle).
- ***Mapping***: creating maps of specific geographical locations.
- ***Timing***: determining the precise time at a specific location. online language translation

Mobile Computing – Location-based Commerce

- There are many applications related to Location Based Commerce:
  - Location-based advertising.
    - The wireless device is detected, and similar to a pop-up ads on a PC, advertising is directed towards the PC.
    - A dynamic billboard ad will be personalized specifically for the occupant of an approaching car.
    - Ads on vehicles (taxicabs, trucks, buses) will change based on the vehicles location.
  - E-911 emergency cell phone calls
  - Telematics and telemetry applications: integration of computers and wireless communications in order to improve information flow (OnStar system by GM) https://www.onstar.com/us/en/home.html

Telematics: *무선통신과 GPS가 결합되어 자동차에서 위치정보, 안전운전, 오락 등 다양한 이동통신 서비스를 제공하는 것*
Telemetry: *automated communication process by which measurements are made and other data collected at remote points*

"No matter where you're headed – OnStar services keep you safe, connected and ready for the road ahead."
→ *Emergency, Security, Navigation, Connections, Vehicle Managers*

Mobile Computing – Pervasive Computing

Pervasive = Ubiquitous 어디에나 있는

A world in which virtually every object has processing power with wireless or wired connections to a global network. The user doesn't have to think about how to use the processing power in the object; rather, the processing power automatically helps the user perform a task (*Invisible Computing Everywhere*).

- **RFID (radio frequency identification)** tag attached to items for sale.
- **Active badges** worn as ID cards by employees.
- **Memory buttons** are nickel-sized devices that store information relating to whatever it is attached to.
- **Contextual computing,** refers to the process of understanding the user's interactions within a valid context, to better understand what the consumer needs, and what products or services they might possibly be interested in at this time. **Context awareness** refers to capturing a broad range of contextual attributes to better understand those needs.