ADO Introduction

Objective: Learn how to use Active Server Page for dynamic content of the webpage.

What is ADO?

ADO is a Microsoft technology
ADO stands for ActiveX Data Objects
ADO is a Microsoft Active-X component
ADO is automatically installed with Microsoft IIS
ADO is a programming interface to access data in a database

Accessing a Database from an ASP Page

The common way to access a database from inside an ASP page is to:

1. Create an ADO connection to a database
2. Open the database connection
3. Create an ADO recordset
4. Open the recordset
5. Extract the data you need from the recordset
6. Close the recordset
7. Close the connection

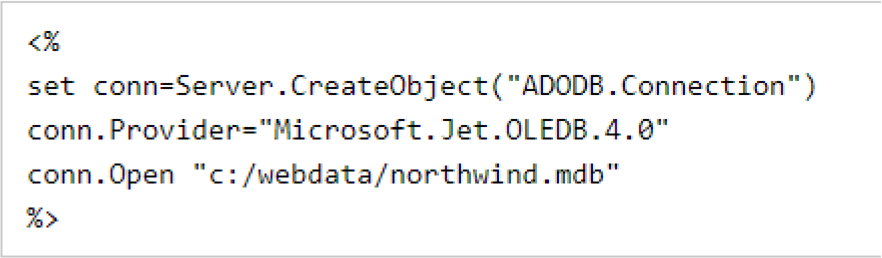
ADO Database Connection

Before a database can be accessed from a web page, a database connection has to be established.

Create a DSN-less Database Connection

The easiest way to connect to a database is to use a DSN-less connection. A DSN-less connection can be used against any Microsoft Access database on your web site.

If you have a database called "northwind.mdb" located in a web directory like "c:/webdata/", you can connect to the database with the following ASP code:



```
<%
set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open "c:/webdata/northwind.mdb"
%>
```

Note, from the example above, which you have to specify the Microsoft Access database driver (Provider) and the physical path to the database on your computer.

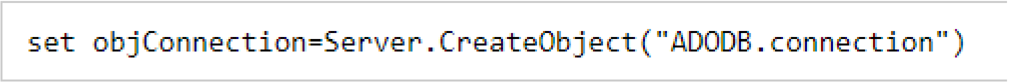
ADO Connection Object

Connection Object

The ADO Connection Object is used to create an open connection to a data source. Through this connection, you can access and manipulate a database.

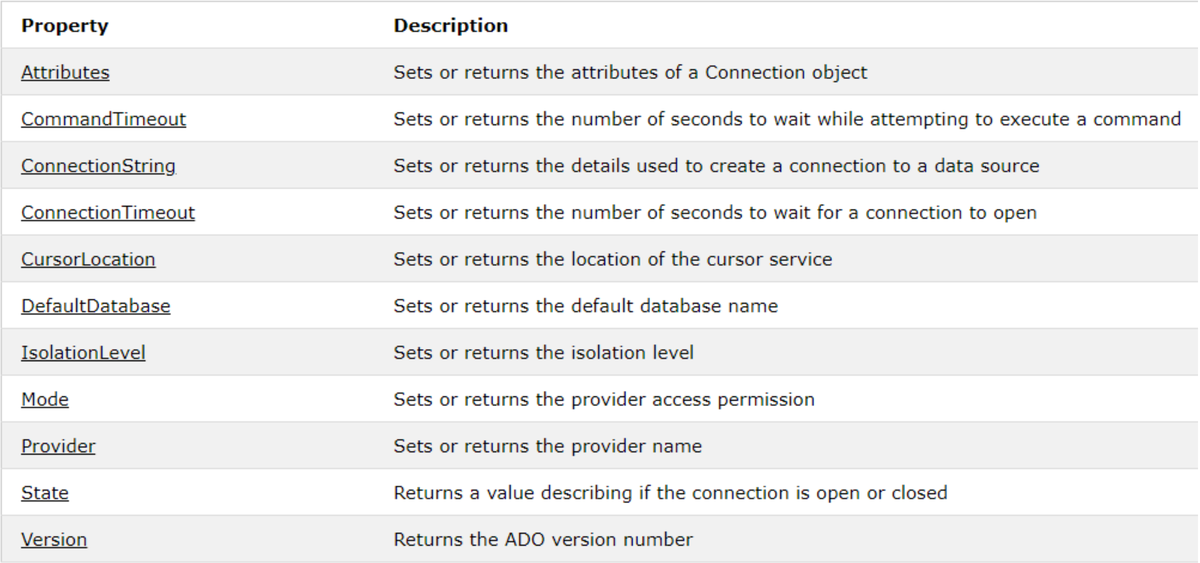
If you want to access a database multiple times, you should establish a connection using the Connection object. You can also make a connection to a database by passing a connection string via a Command or Recordset object. However, this type of connection is only good for one specific, single query.

ProgID



```
set objConnection=Server.CreateObject("ADODB.connection")
```

Properties



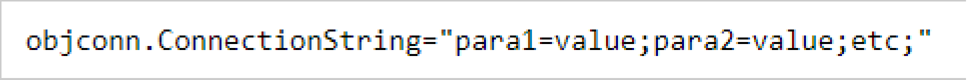
Property	Description
Attributes	Sets or returns the attributes of a Connection object
CommandTimeout	Sets or returns the number of seconds to wait while attempting to execute a command
ConnectionString	Sets or returns the details used to create a connection to a data source
ConnectionTimeout	Sets or returns the number of seconds to wait for a connection to open
CursorLocation	Sets or returns the location of the cursor service
DefaultDatabase	Sets or returns the default database name
IsolationLevel	Sets or returns the isolation level
Mode	Sets or returns the provider access permission
Provider	Sets or returns the provider name
State	Returns a value describing if the connection is open or closed
Version	Returns the ADO version number

ADO ConnectionString Property

The ConnectionString property sets or returns the details used to create a connection to a data source.

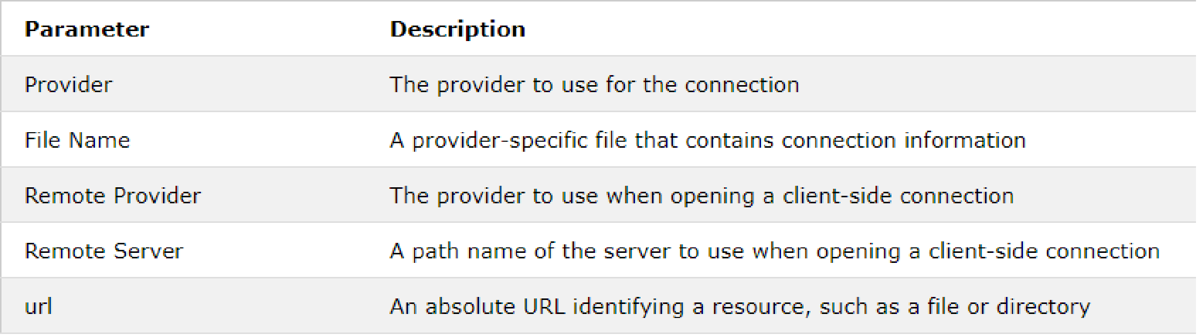
Note: You cannot use both the Provider and File Name parameters.

Syntax



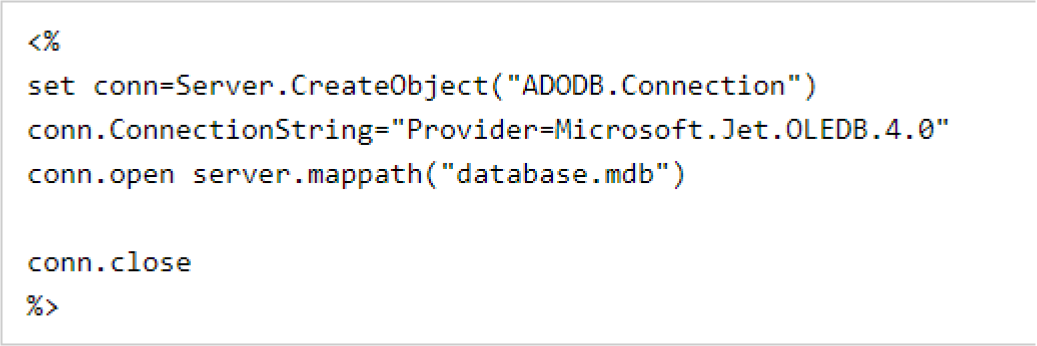
```
objconn.ConnectionString="para1=value;para2=value;etc;"
```

The ConnectionString property has these parameters:



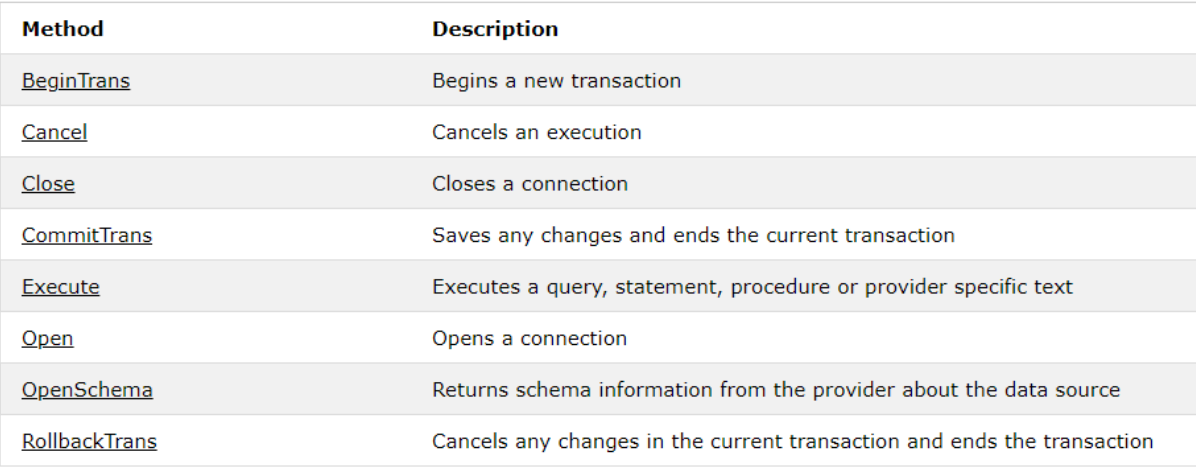
Parameter	Description
Provider	The provider to use for the connection
File Name	A provider-specific file that contains connection information
Remote Provider	The provider to use when opening a client-side connection
Remote Server	A path name of the server to use when opening a client-side connection
url	An absolute URL identifying a resource, such as a file or directory

Example Script



```
<%
set conn=Server.CreateObject("ADODB.Connection")
conn.ConnectionString="Provider=Microsoft.Jet.OLEDB.4.0"
conn.open server.mappath("database.mdb")

conn.close
%>
```

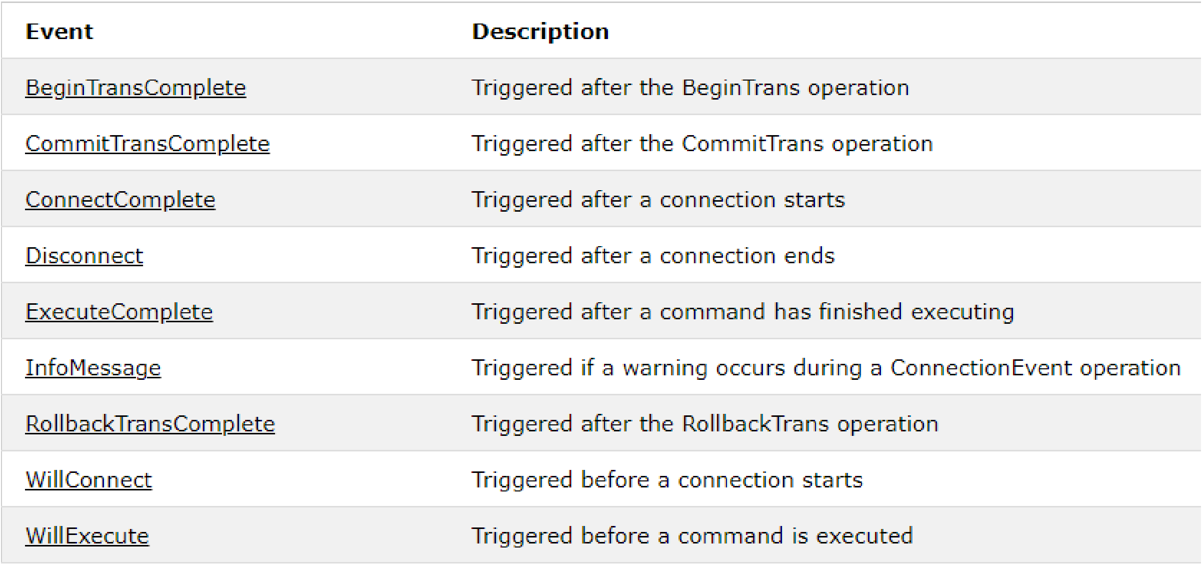
Methods



Method	Description
BeginTrans	Begins a new transaction
Cancel	Cancels an execution
Close	Closes a connection
CommitTrans	Saves any changes and ends the current transaction
Execute	Executes a query, statement, procedure or provider specific text
Open	Opens a connection
OpenSchema	Returns schema information from the provider about the data source
RollbackTrans	Cancels any changes in the current transaction and ends the transaction

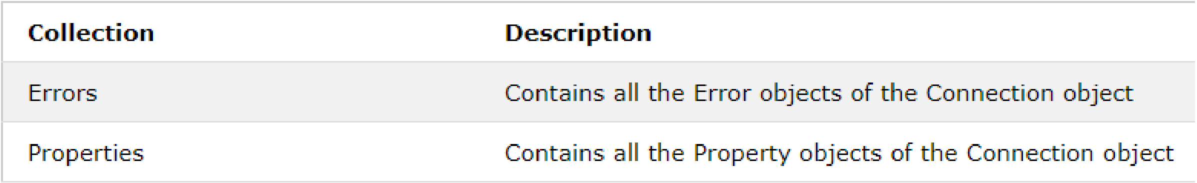
Events

Note: You cannot handle events using VBScript or JScript (only Visual Basic, Visual C++, and Visual J++ languages can handle events).



Event	Description
BeginTransComplete	Triggered after the BeginTrans operation
CommitTransComplete	Triggered after the CommitTrans operation
ConnectComplete	Triggered after a connection starts
Disconnect	Triggered after a connection ends
ExecuteComplete	Triggered after a command has finished executing
InfoMessage	Triggered if a warning occurs during a ConnectionEvent operation
RollbackTransComplete	Triggered after the RollbackTrans operation
WillConnect	Triggered before a connection starts
WillExecute	Triggered before a command is executed

Collections



Collection	Description
Errors	Contains all the Error objects of the Connection object
Properties	Contains all the Property objects of the Connection object

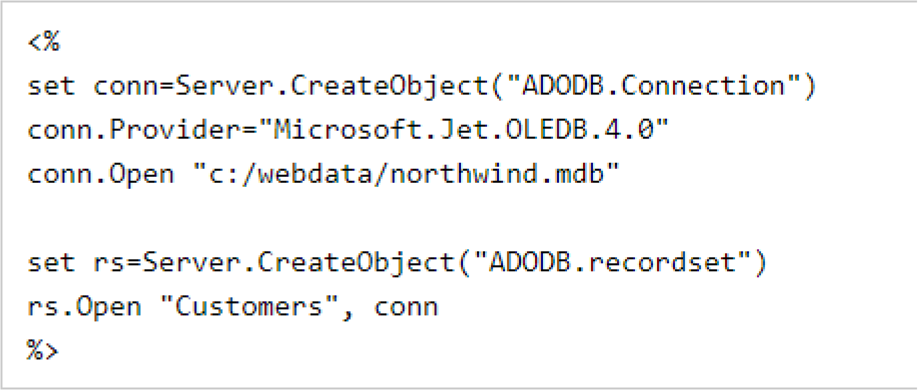
ADO Recordset

To be able to read database data, the data must first be loaded into a recordset.

Create an ADO Table Recordset

After an ADO Database Connection has been created, as demonstrated in the previous chapter, it is possible to create an ADO Recordset.

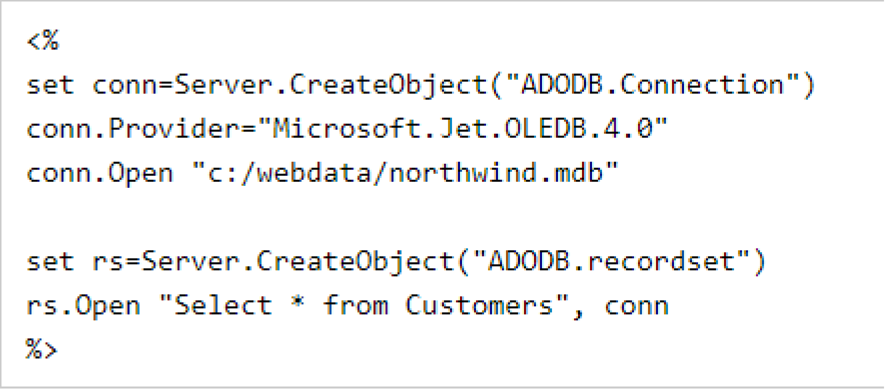
Suppose we have a database named "Northwind", we can get access to the "Customers" table inside the database with the following lines:



```
<%
set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open "c:/webdata/northwind.mdb"

set rs=Server.CreateObject("ADODB.recordset")
rs.Open "Customers", conn
%>
```

Create an ADO SQL Recordset

We can also get access to the data in the "Customers" table using SQL:

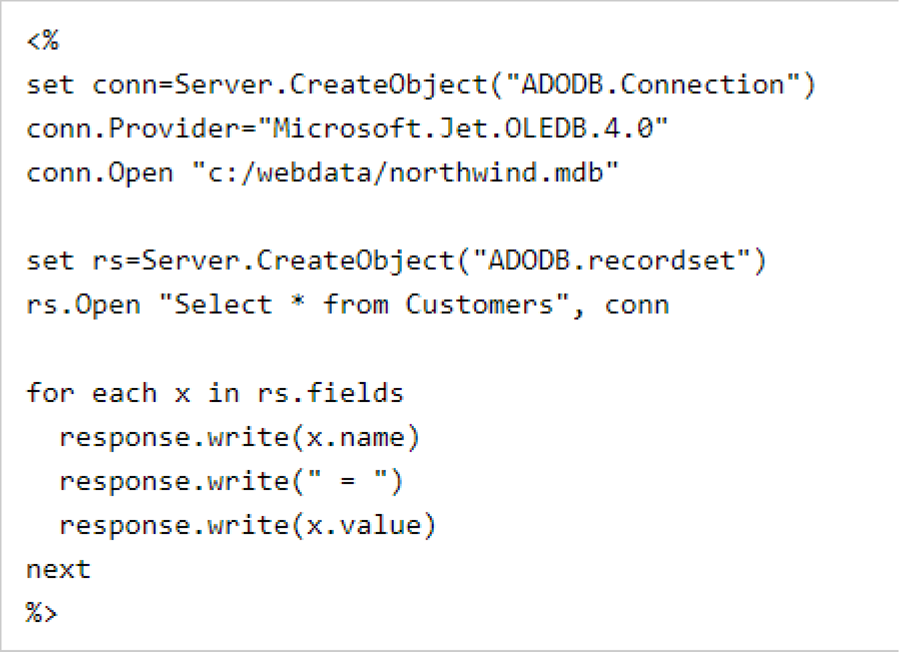


```
<%
set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open "c:/webdata/northwind.mdb"

set rs=Server.CreateObject("ADODB.recordset")
rs.Open "Select * from Customers", conn
%>
```

Extract Data from the Recordset

After a recordset is opened, we can extract data from recordset.

Suppose we have a database named "Northwind", we can get access to the "Customers" table inside the database with the following lines:



```
<%
set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open "c:/webdata/northwind.mdb"

set rs=Server.CreateObject("ADODB.recordset")
rs.Open "Select * from Customers", conn

for each x in rs.fields
  response.write(x.name)
  response.write(" = ")
  response.write(x.value)
next
%>
```

The ADO Recordset Object

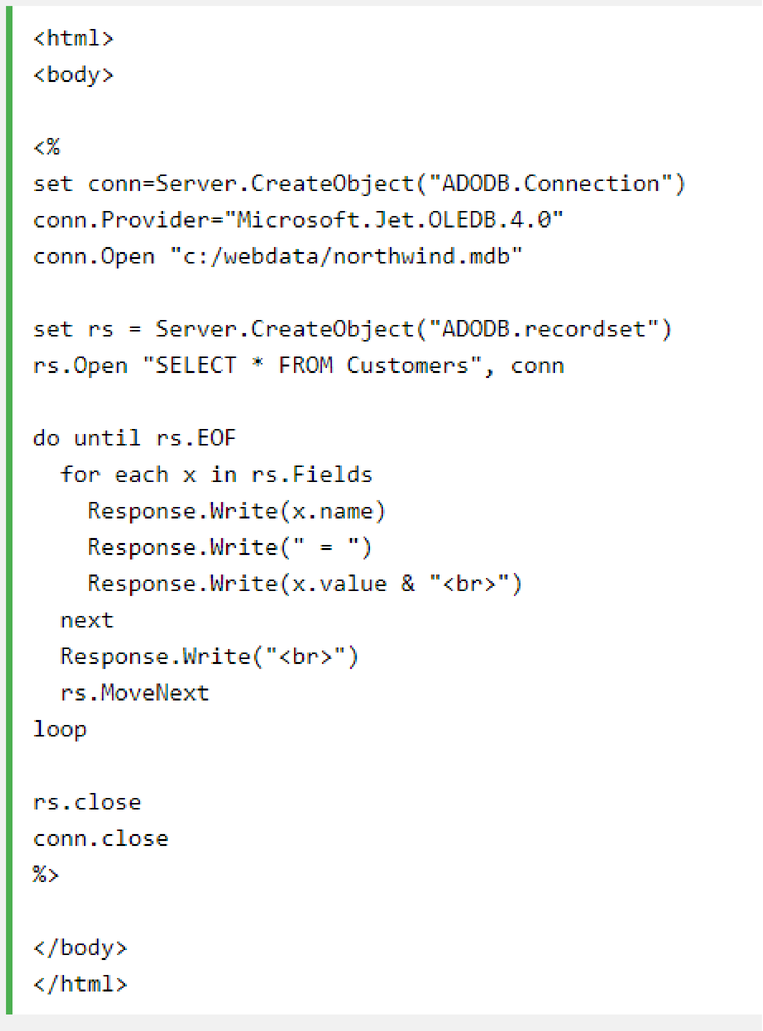
The ADO Recordset object is used to hold a set of records from a database table.

ADO Display

The most common way to display data from a recordset, is to display the data in an HTML table.

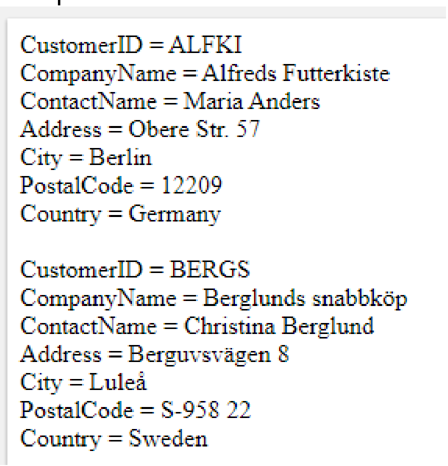
Display the Field Names and Field Values

We have a database named "Northwind" and we want to display the data from the "Customers" table (remember to save the file with an .asp extension):



```
<html>
<body>

<%
set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open "c:/webdata/northwind.mdb"

set rs = Server.CreateObject("ADODB.recordset")
rs.Open "SELECT * FROM Customers", conn

do until rs.EOF
  for each x in rs.Fields
    Response.Write(x.name)
    Response.Write(" = ")
    Response.Write(x.value & "<br>")
  next
  Response.Write("<br>")
  rs.MoveNext
loop

rs.close
conn.close
%>

</body>
</html>
```

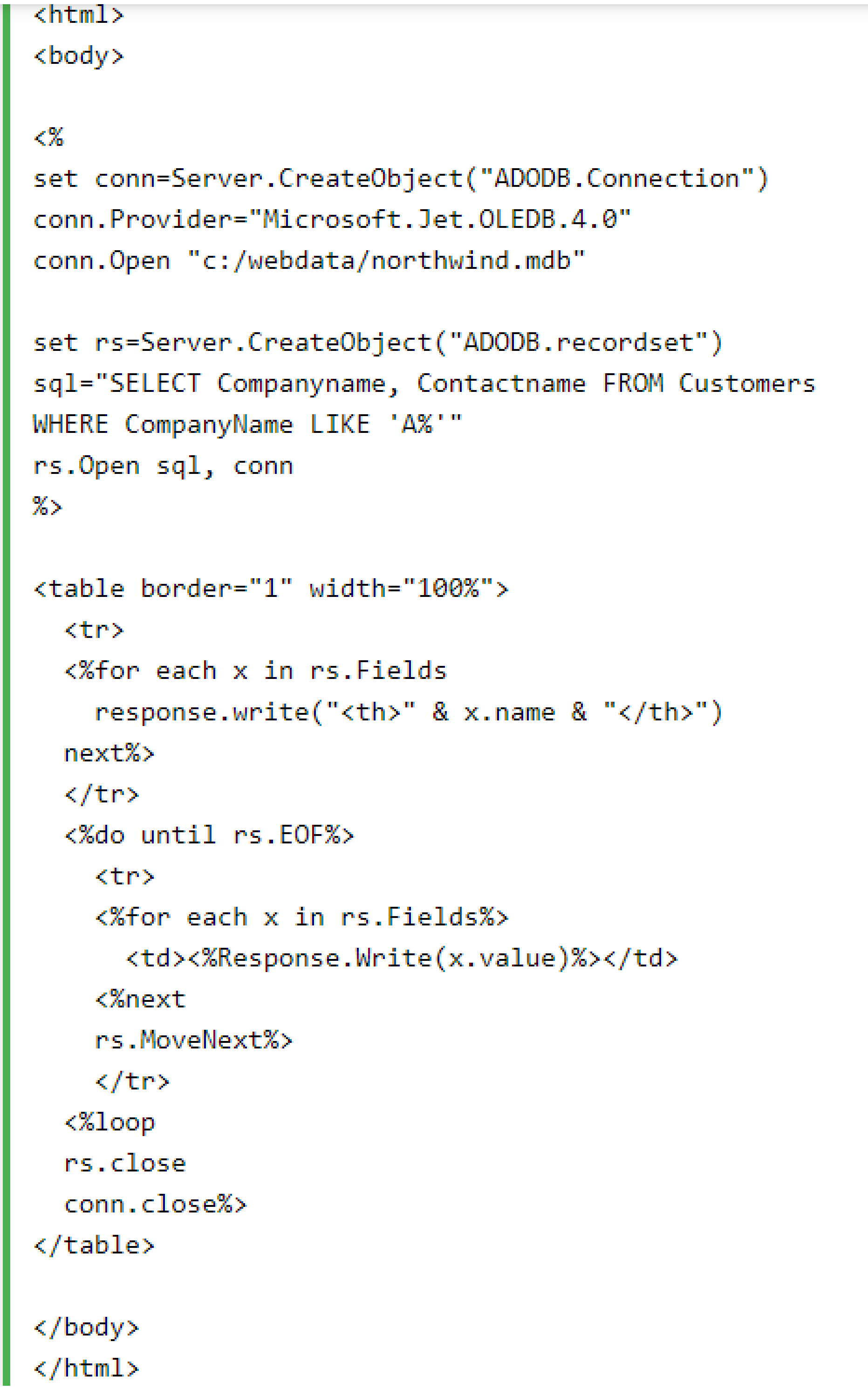
Sample Result:



```
CustomerID = ALFKI
CompanyName = Alfreds Futterkiste
ContactName = Maria Anders
Address = Obere Str. 57
City = Berlin
PostalCode = 12209
Country = Germany

CustomerID = BERGS
CompanyName = Berglunds snabbköp
ContactName = Christina Berglund
Address = Berguvsvägen 8
City = Luleå
PostalCode = S-958 22
Country = Sweden
```

ADO Queries
We may use SQL to create queries to specify only a selected set of records and fields to view.

Display Selected Data
We want to display only the records from the "Customers" table that have a "Companyname" that starts with an A (remember to save the file with an .asp extension):



```
<html>
<body>

<%
set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open "c:/webdata/northwind.mdb"

set rs=Server.CreateObject("ADODB.recordset")
sql="SELECT Companyname, Contactname FROM Customers
WHERE CompanyName LIKE 'A%'"
rs.Open sql, conn
%>

<table border="1" width="100%">
  <tr>
  <%for each x in rs.Fields
    response.write("<th>" & x.name & "</th>")
  next%>
  </tr>
  <%do until rs.EOF%>
    <tr>
    <%for each x in rs.Fields%>
      <td><%Response.Write(x.value)%></td>
    <%next
    rs.MoveNext%>
    </tr>
  <%loop
  rs.close
  conn.close%>
</table>

</body>
</html>
```

ADO Add Records

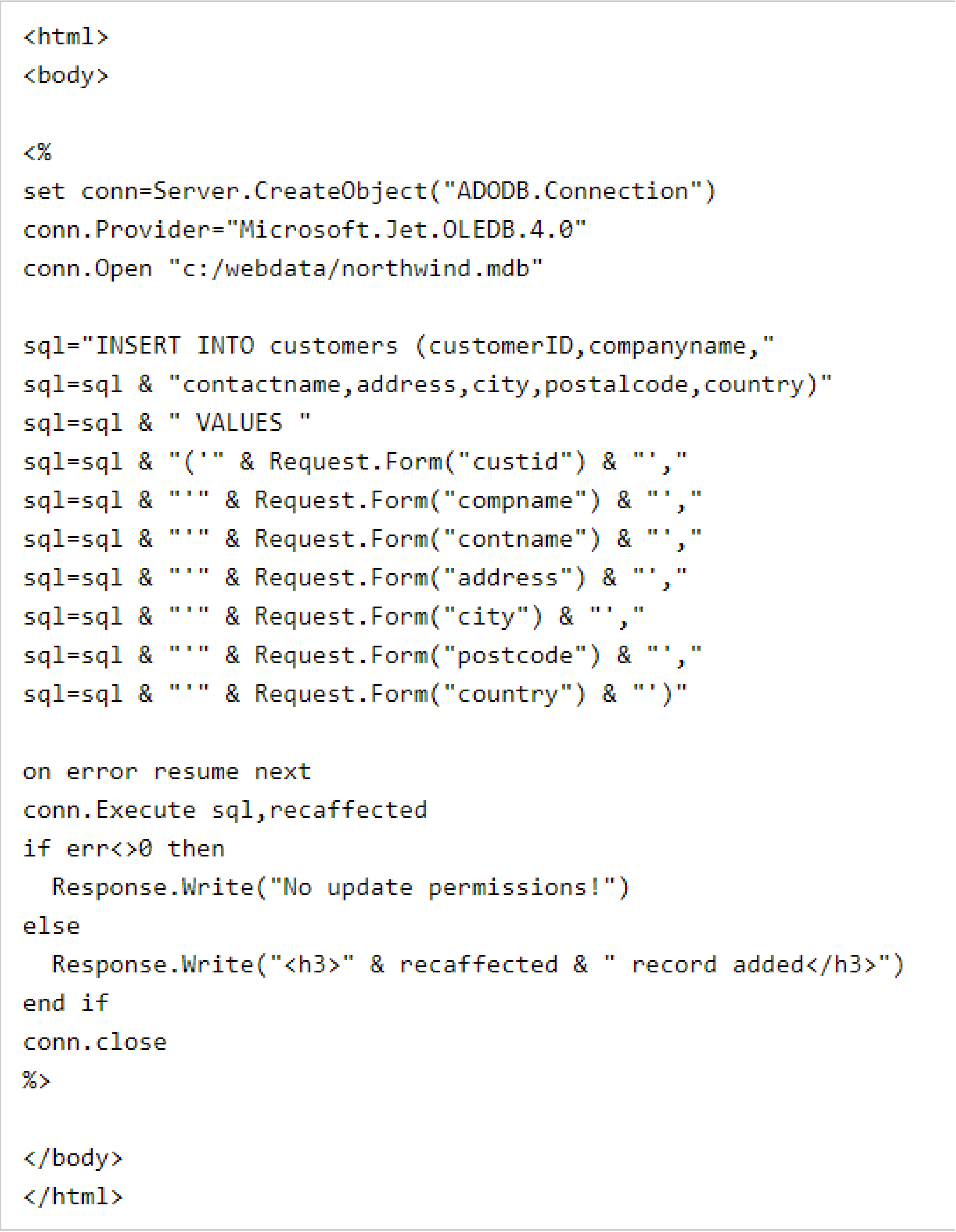
We may use the SQL INSERT INTO command to add a record to a table in a database.

Add a Record to a Table in a Database

We want to add a new record to the Customers table in the Northwind database. We first create a form that contains the fields we want to collect data from:

```
<html>
<body>

<form method="post" action="demo_add.asp">
<table>
<tr>
<td>CustomerID:</td>
<td><input name="custid"></td>
</tr><tr>
<td>Company Name:</td>
<td><input name="compname"></td>
</tr><tr>
<td>Contact Name:</td>
<td><input name="contname"></td>
</tr><tr>
<td>Address:</td>
<td><input name="address"></td>
</tr><tr>
<td>City:</td>
<td><input name="city"></td>
</tr><tr>
<td>Postal Code:</td>
<td><input name="postcode"></td>
</tr><tr>
<td>Country:</td>
<td><input name="country"></td>
</tr>
</table>
<br><br>
<input type="submit" value="Add New">
<input type="reset" value="Cancel">
</form>

</body>
```

When the user presses the submit button the form is sent to a file called "demo_add.asp". The "demo_add.asp" file contains the code that will add a new record to the Customers table:



```
<html>
<body>

<%
set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open "c:/webdata/northwind.mdb"

sql="INSERT INTO customers (customerID,companyname,"
sql=sql & "contactname,address,city,postalcode,country)"
sql=sql & " VALUES "
sql=sql & "('" & Request.Form("custid") & "',"
sql=sql & "'" & Request.Form("compname") & "',"
sql=sql & "'" & Request.Form("contname") & "',"
sql=sql & "'" & Request.Form("address") & "',"
sql=sql & "'" & Request.Form("city") & "',"
sql=sql & "'" & Request.Form("postcode") & "',"
sql=sql & "'" & Request.Form("country") & "')"

on error resume next
conn.Execute sql,recaffected
if err<>0 then
  Response.Write("No update permissions!")
else
  Response.Write("<h3>" & recaffected & " record added</h3>")
end if
conn.close
%>

</body>
</html>
```

Important
If you use the SQL INSERT command be aware of the following:

If the table contains a primary key, make sure to append a unique, non-Null value to the primary key field (if not, the provider may not append the record, or an error occurs)
If the table contains an AutoNumber field, do not include this field in the SQL INSERT command (the value of this field will be taken care of automatically by the provider)

What about Fields With no Data?

In a MS Access database, you can enter zero-length strings ("") in Text, Hyperlink, and Memo fields IF you set the AllowZeroLength property to Yes.

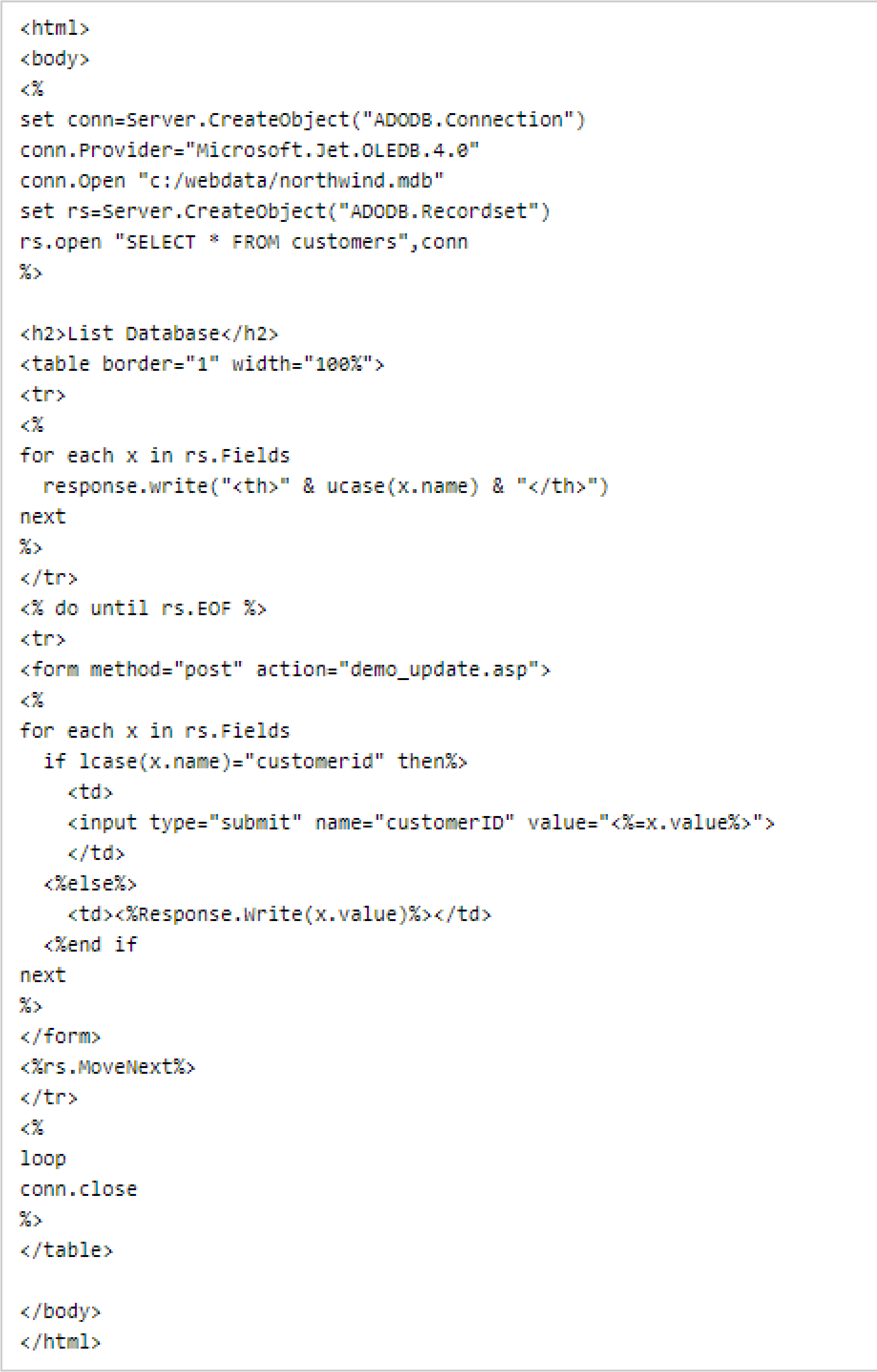
Note: Not all databases support zero-length strings and may cause an error when a record with blank fields is added. It is important to check what data types your database supports.

ADO Update Records

We may use the SQL UPDATE command to update a record in a table in a database.

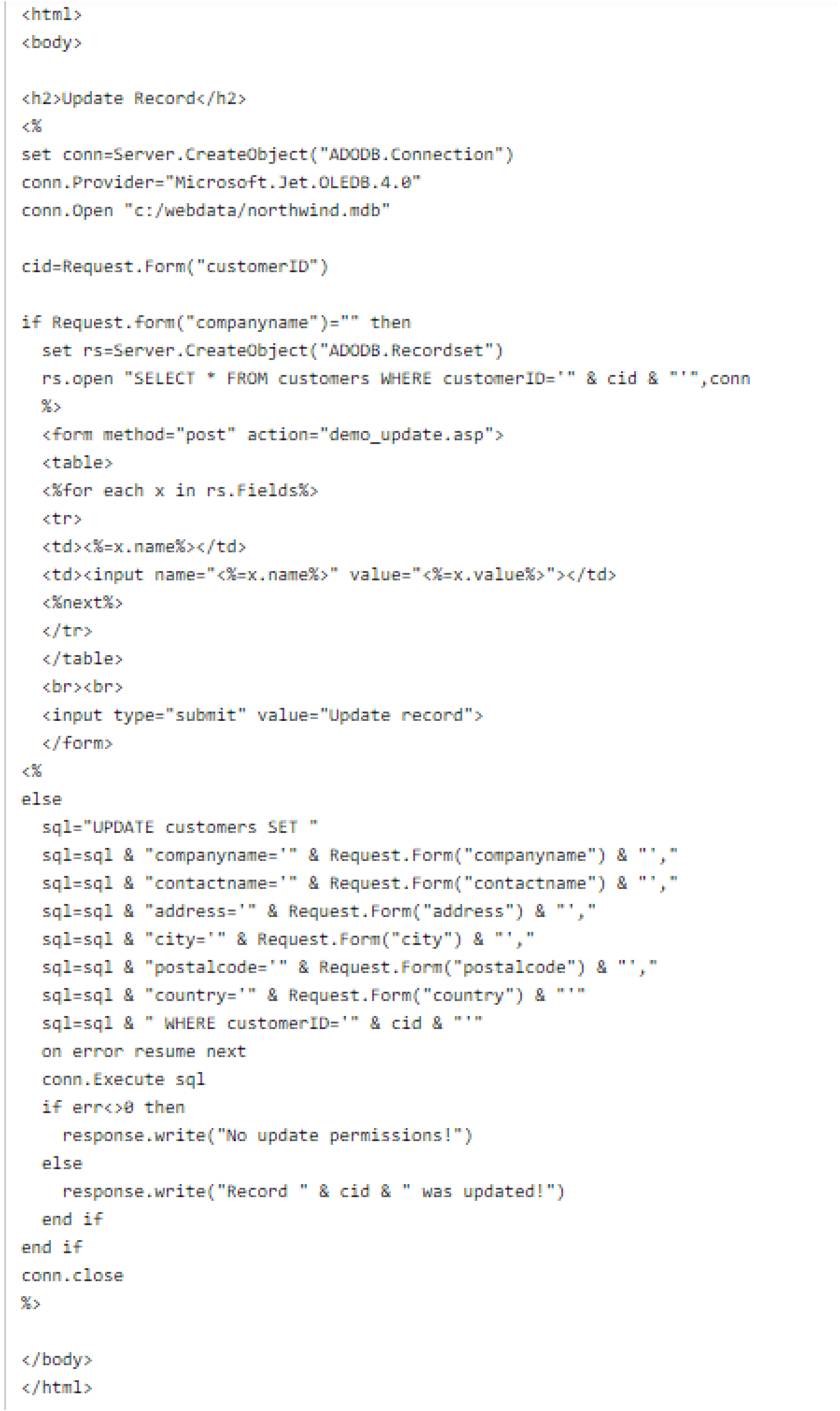
Update a Record in a Table

We want to update a record in the Customers table in the Northwind database. We first create a table that lists all records in the Customers table:



```
<html>
<body>
<%
set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open "c:/webdata/northwind.mdb"
set rs=Server.CreateObject("ADODB.Recordset")
rs.open "SELECT * FROM customers",conn
%>

<h2>List Database</h2>
<table border="1" width="100%">
<tr>
<%
for each x in rs.Fields
  response.write("<th>" & ucase(x.name) & "</th>")
next
%>
</tr>
<% do until rs.EOF %>
<tr>
<form method="post" action="demo_update.asp">
<%
for each x in rs.Fields
  if lcase(x.name)="customerid" then%>
    <td>
    <input type="submit" name="customerID" value="<%=x.value%>">
    </td>
  <%else%>
    <td><%Response.Write(x.value)%></td>
  <%end if
next
%>
</form>
<%rs.MoveNext%>
</tr>
<%
loop
conn.close
%>
</table>

</body>
</html>
```

If the user clicks on the button in the "customerID" column he or she will be taken to a new file called "demo_update.asp". The "demo_update.asp" file contains the source code on how to create input fields based on the fields from one record in the database table. It also contains a "Update record" button that will save your changes:



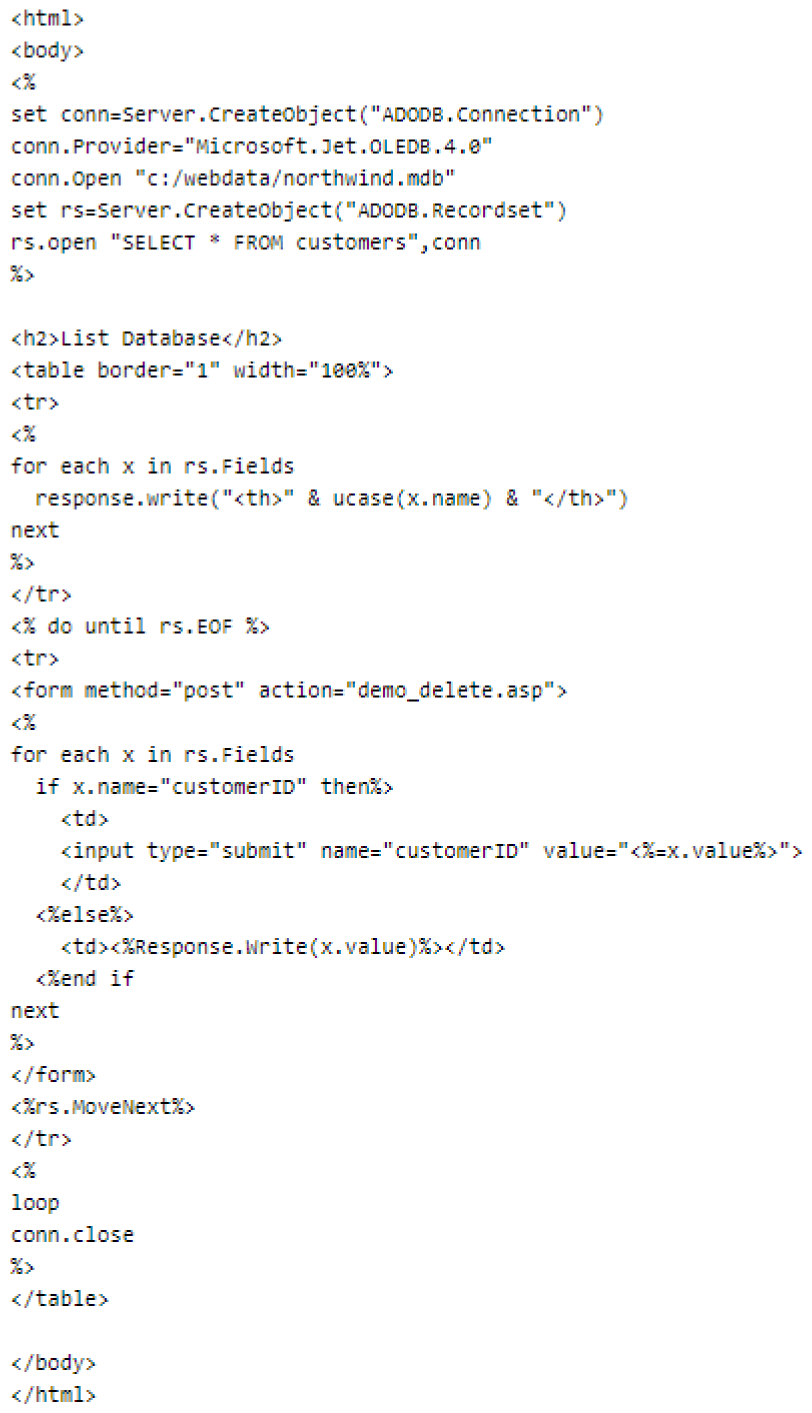
```
<html>
<body>

<h2>Update Record</h2>
<%
set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open "c:/webdata/northwind.mdb"

cid=Request.Form("customerID")

if Request.form("companyname")="" then
  set rs=Server.CreateObject("ADODB.Recordset")
  rs.open "SELECT * FROM customers WHERE customerID='" & cid & "'",conn
  %>
  <form method="post" action="demo_update.asp">
  <table>
  <%for each x in rs.Fields%>
  <tr>
  <td><%=x.name%></td>
  <td><input name="<%=x.name%>" value="<%=x.value%>"></td>
  <%next%>
  </tr>
  </table>
  <br><br>
  <input type="submit" value="Update record">
  </form>
<%
else
  sql="UPDATE customers SET "
  sql=sql & "companyname='" & Request.Form("companyname") & "',"
  sql=sql & "contactname='" & Request.Form("contactname") & "',"
  sql=sql & "address='" & Request.Form("address") & "',"
  sql=sql & "city='" & Request.Form("city") & "',"
  sql=sql & "postalcode='" & Request.Form("postalcode") & "',"
  sql=sql & "country='" & Request.Form("country") & "'"
  sql=sql & " WHERE customerID='" & cid & "'"
  on error resume next
  conn.Execute sql
  if err<>0 then
    response.write("No update permissions!")
  else
    response.write("Record " & cid & " was updated!")
  end if
end if
conn.close
%>

</body>
</html>
```

ADO Delete Records

We may use the SQL DELETE command to delete a record in a table in a database.

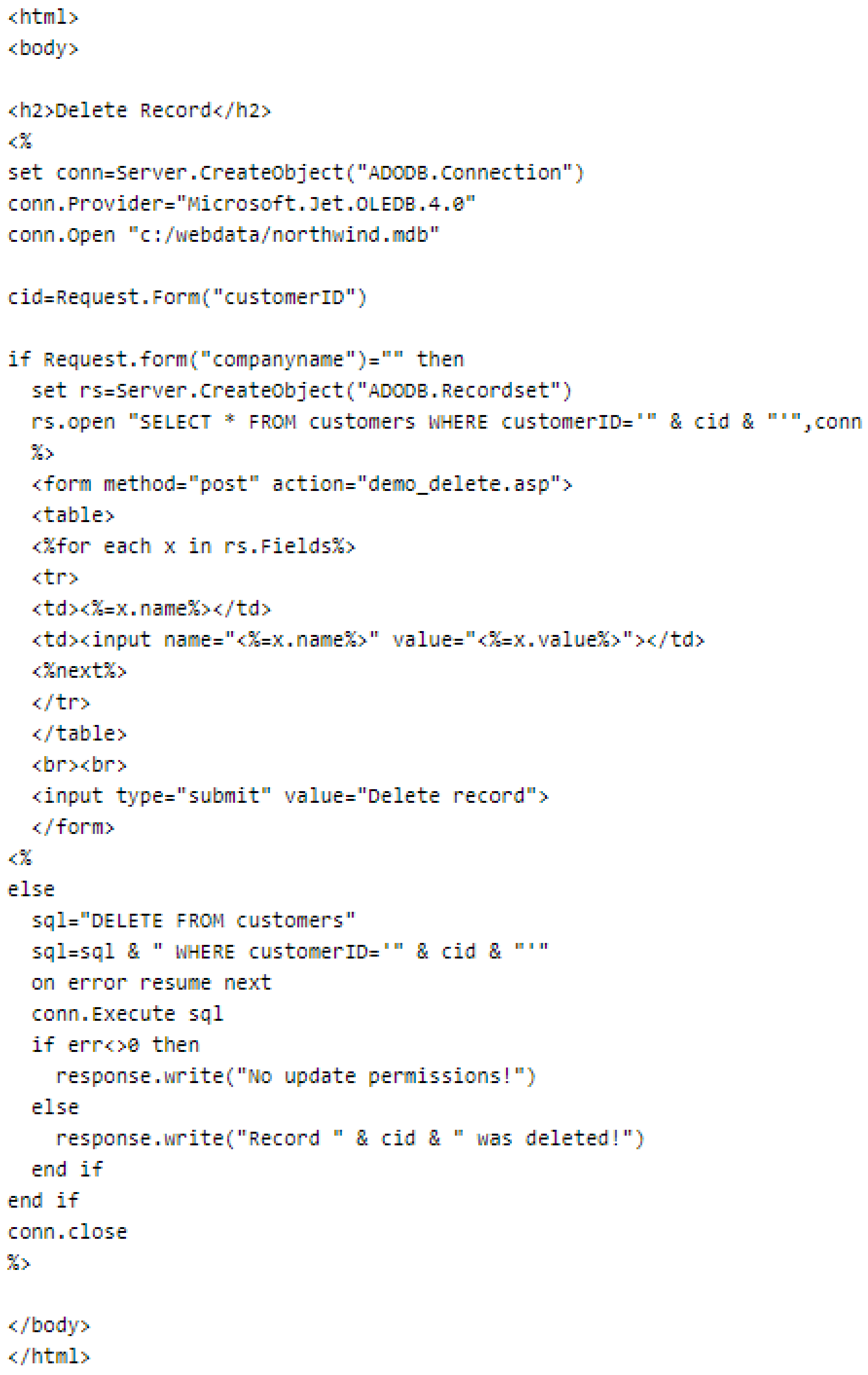
Delete a Record in a Table

We want to delete a record in the Customers table in the Northwind database. We first create a table that lists all records in the Customers table:



```
<html>
<body>
<%
set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open "c:/webdata/northwind.mdb"
set rs=Server.CreateObject("ADODB.Recordset")
rs.open "SELECT * FROM customers",conn
%>

<h2>List Database</h2>
<table border="1" width="100%">
<tr>
<%
for each x in rs.Fields
  response.write("<th>" & ucase(x.name) & "</th>")
next
%>
</tr>
<% do until rs.EOF %>
<tr>
<form method="post" action="demo_delete.asp">
<%
for each x in rs.Fields
  if x.name="customerID" then%>
    <td>
    <input type="submit" name="customerID" value="<%=x.value%>">
    </td>
  <%else%>
    <td><%Response.Write(x.value)%></td>
  <%end if
next
%>
</form>
<%rs.MoveNext%>
</tr>
<%
loop
conn.close
%>
</table>

</body>
</html>
```

If the user clicks on the button in the "customerID" column he or she will be taken to a new file called "demo_delete.asp". The "demo_delete.asp" file contains the source code on how to create input fields based on the fields from one record in the database table. It also contains a "Delete record" button that will delete the current record:



```
<html>
<body>

<h2>Delete Record</h2>
<%
set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open "c:/webdata/northwind.mdb"

cid=Request.Form("customerID")

if Request.form("companyname")="" then
  set rs=Server.CreateObject("ADODB.Recordset")
  rs.open "SELECT * FROM customers WHERE customerID='" & cid & "'",conn
  %>
  <form method="post" action="demo_delete.asp">
  <table>
  <%for each x in rs.Fields%>
  <tr>
  <td><%=x.name%></td>
  <td><input name="<%=x.name%>" value="<%=x.value%>"></td>
  <%next%>
  </tr>
  </table>
  <br><br>
  <input type="submit" value="Delete record">
  </form>
<%
else
  sql="DELETE FROM customers"
  sql=sql & " WHERE customerID='" & cid & "'"
  on error resume next
  conn.Execute sql
  if err<>0 then
    response.write("No update permissions!")
  else
    response.write("Record " & cid & " was deleted!")
  end if
end if
conn.close
%>

</body>
</html>
```

Reference: www.w3schools.com - https://www.w3schools.com/asp/ado_intro.asp