

Querying an Access database in ASP

by Manohar Kamath

Active Server Pages allow us an easy means of querying and updating a database from a webpage. This article describes the basics of using this technique in a nutshell. The examples given here utilize a Microsoft Access database, but the principles elucidated apply equally well to ASP integration with other databaes. (Access has been selected because it is widely familiar, easy to use, and adequate for many small to medium-sized projects.)

We will be using the *Connection* object throughout the examples since it is very easy to handle. In the following discussion we will see the two most common ways of using command object with Access database.

The Database



tbl_member
member-id
lname
fname
address
city
state
zip
login
passwd

The database is a simple one-table one, with a member information table. The table is a simulated library database, with the information about its members.

The member-id is the primary key and it contains the member information like last name, first name etc. as shown in the diagram on the left. Our goal is to create queries based on this table.

SQL Queries

These are the most simple to create. These are based on SQL statements in the Command object. All you need for this exercise is a fair understanding of SQL commands - SELECT, DELETE, and UPDATE.

E.g. Generate a detailed report of all the members living within the ZIP area 60194. Note that the zip field is a TEXT field of length 5

```
' First let us create Connection and Recordset objects
Set Conn = Server.CreateObject("ADODB.Connection")
Set Rs = Server.CreateObject("ADODB.RecordSet")
```

```
' Open the connection to the ODBC source, in this case
' the Access database
Conn.Open "your_ODBC_source"
```

```

' Now, create the SQL statement
sSQL = "SELECT * FROM tbl_member WHERE zip='60194'"

' Execute the SQL statement, and set the recordset object
' to the result of this execution. We obtain the resulting
' records in Rs object
Set Rs = Conn.Execute(sSQL)

' Use this RecordSet object to populate your HTML output stream
' In this example, we will just write out the last name field
Do While NOT Rs.EOF
    Response.Write(Rs.Fields("lName").value)

    ' Move to the next record in the resultset
    Rs.MoveNext
Loop

' Close the Recordset object and destroy it
Rs.Close
Set Rs = Nothing

' You might want to release the resources for connection object,
' unless you want to use the same connection again in the later code
Conn.Close
Set Conn = Nothing

```

With the above things (as in documentation) in mind, you could create any SQL query in ASP. However, there are a few things to note while working with ASP and Access.

- **Using LIKE and * operators in SQL statements**
Although * is a wild-card character for SQL with LIKE clauses, you need to replace this with % character. E.g. the following query is in standard SQL

```
SELECT * FROM tbl_member WHERE lName LIKE 'smith*'
```

When using this in ASP (or ADO in general), use the following syntax

```
SELECT * FROM tbl_member WHERE lName LIKE 'smith%'
```

- **Using 's (apostrophes) in text fields**
This is probably the most common problem discussed in the ASP newsgroups. One ' in a text field within the SQL query screws up the whole SQL statement to produce some esoteric ODBC errors. E.g. Say you have an SQL statement which runs like this

```
"INSERT INTO tbl_member VALUES ( " & iMemID & ", '" & sLName & "' .....
```

where iMemID, sLname.. are dynamically produced values (say, from a form).
When everything is all right, the final SQL produced may look like

```
INSERT INTO tbl_member VALUES (10, 'smith', .....
```

Now, a last name like D'Abo could result in a statement like

```
INSERT INTO tbl_member VALUES (10, 'D'Abo', ...)
```

Note the value 'D'Abo', which leaves one ' dangling. This is what causes the ODBC error since the SQL syntax is not right.

This problem could easily be corrected. Just replace one ' with two 's. Use the Replace function in VBScript.

```
sLname = Replace(sLname, "'", "'")
```

easily fixes this problem. Use Replace for each field, that makes up the SQL statement, which may contain apostrophes.