Using LIKE Wildcards in Your Queries

Posted At : October 27, 2005 10:32 AM | Posted By : Mark Kruger
Related Categories: MS SQL Server, Coldfusion & Databases

I often comment that learning the ins and outs of your database platform can do just as much for your productivity as learning advanced Coldfusion coding techniques. Here's another example. Suppose you have a "LIKE" expression in your query that checks against a stored character value. You use the expression to allow a user to do a search against a character column in the database. If the column in question contains an underscore you might end up scratching your head at the results.

For example, let's say you are doing a search for user ID's and you need to pull out the user ID "joe_bubbaloui", but you can't remember Joe's last name. So you try this:

```
<Cfquery name="getUsers" datasource="#dsn#">
      SELECT   *
      FROM   users
      WHERE   username LIKE 'joe_%'
   </CFQUERY>
```

That *should* return everything beginning with "joe_" right? Well, not exactly. You see the underscore is a special character in a LIKE expression. It's a placeholder for "any single character". So the query above is *actually* asking for "any username of at least 3 characters beginning with joe". Joe_bubbaloui, Joesmith, joey and joenofark would all be returned. This can be quite useful. For example, let's say you know of a username ending in smith, but you can't remember the first name. If you happened to remember that it was a short first name - say 3 letters long (bob, rob, sue etc.) then you could search for LIKE '___smith' and it would return suesmith, robsmith or bobsmith. It would *not* return janesmith or josmith. You get the idea.

Now back to our original problem. How do we search *for the underscore* without having the server interpret it as a pattern? Simple, use the square brackets. The square brackets have a sort of special function (in MS SQL) as the "quoted identifier". They tell the server "treat this as a literal". Modify your query as follows:

```
<Cfquery name="getUsers" datasource="#dsn#">
      SELECT   *
      FROM   users
      WHERE   username LIKE 'joe[_]%'
   </CFQUERY>
```

Now you will get *only* usernames beginning with "joe_".

Some Other Uses

The square brackets in a LIKE query can be used in other ways too. You can match a pattern or series of characters. For example, let's say you have a date value that is stored as character data in a string that looks like 20050311 (this happens often with mainframe data) and you wanted to find all the records for 2002 through 2005, but you wanted to exclude 2003 (for some reason). You might try casting the data as date types:

```
SELECT CAST(
         CAST(left(dt,4) AS char(4)) + '-'
         + CAST(substring(dt,5,2) AS char(2))
         + '-' + CAST(right(dt,2) AS char(2))
         AS SmalldateTime) AS newDT
```

... and then select the range from your new data type. You might have other reasons for doing this - but it's admittedly kludgy and awkward. Using LIKE with a pattern match you could just as easily do:

```
SELECT *
   FROM   myTable
   WHERE   dt LIKE '200[235]%'
```

The result would be all the strings beginning with 2002, 2003 and 2005 - 2004 would be excluded.

You can also use the carrot symbol (^) to do a "NOT" inside of the pattern. For example:

```
SELECT *
   FROM   myTable
   WHERE   dt LIKE '200[^4]%'
```

... would give you everything that did NOT begin with 2004.

SQL Wildcards is just one more example of how utilizing the power of your database platforms can save you headaches and workarounds in your Coldfusion code. The examples above come from MS SQL server. If you have some examples from other platforms feel free to post them and add to this explanation.