

## Using LIKE with special Characters (like the percent sign)

Posted At : December 27, 2005 6:29 PM | Posted By : Mark Kruger

Related Categories: MS SQL Server

### CF Muse Reader Asks:

How do you select a record containing a "%" using the LIKE keyword in sql

Ah... I love the easy ones. If you have a character column that contains the phrase "50% over gross" and "5% over gross" how could you construct a "LIKE" clause that would capture JUST the "5%" over gross if all you had to go on was "5%"? You see? If you did the following...

```
SELECT      *
FROM        grossup
WHERE       totalDesc LIKE '5%'
```

... it would naturally trap *both* the 50 and the 5 - because they both "begin" with 5. How do I get the database server to treat the percent sign as a character instead of a wildcard? Using the handy dandy bracket identifiers of course. This code:

```
SELECT      *
FROM        grossup
WHERE       totalDesc LIKE '5[%]'
```

... would do the trick and return just the "5%" description.

### Another Related tip - the underscore

Here's another tip. You can use the underscore like a wildcard character. If you had 500, 501 and 5000 as characters in a column and you wanted to get all the columns that were in the hundreds and began with 50 you could not do "LIKE '50%' - because it would pick up the 5000 as well. But you *could* use the underscore as in this example:

```
SELECT      *
FROM        Categories
WHERE       cat_num LIKE '50_'
```

...this would give you 500 and 501 - because they are both 3 characters beginning with 50. 5000 would be excluded.

### Regular expression pattern syntax

Finally, for those of you who are regex junkies out there you can use pattern matching type syntax as follows...

```
SELECT      *
FROM        Policies
WHERE       description LIKE 'Policy [A-C]%'
```

... which would give you "Policy A" and "Policy C" but not "Policy H". It's a neat trick when you need it. Personally I end up tearing out my hair trying to write regexes. I usually end up writing the **CF-Talk** list where Michael Dinowitz is the master of the regular expression and seems to pull what I need out of thin air in most cases (ha).