SAS® GLOBAL FORUM 2020
MARCH 29 - APRIL 1
WASHINGTON, DC
There is wildcarding allowed within where clause like expressions by using special characters. But if you want to search for those specific characters, you have to override them. If you forget to escape the character, you’ll get unexpected matches.

Topics include:

- proc sql; select x from y where x like %a/_b escape /;
- _ matches 1 character
- % matches any characters
- / escapes the next character (says next is actual character, not special meaning)
- set sorted_ddl (obs=max where=(name like '%/_ORD%' escape '/'));

Conclusion

There is wildcarding allowed within where clause like expressions by using special characters. But if you want to search for those specific characters, you have to override them. If you forget to escape the character, you’ll get unexpected matches.

Topics include:

- proc sql; select x from y where x like %a/_b escape /;
- _ matches 1 character
- % matches any characters
- / escapes the next character (says next is actual character, not special meaning)
- set sorted_ddl (obs=max where=(name like '%/_ORD%' escape '/'));
Basics of LIKE

- The general format is:

  Where variable like "%val%" escape "/";

  - % matches zero to many characters
  - _ matches exactly one character
  - The escape clause is optional
    - Use it if you want to match to a % or _
    - There is no default escape character

  Where variable like "%string/_with/_underscore" escape "/";

- If you make a mistake like:

  Where product_group like "%/%" escape "/";

  Or

  Where product_group like "%Kid/s%" escape "/";

  You’ll see

  ERROR: Like pattern is not a valid pattern due to the invalid use of the escape pattern.
Usability

- Like can be used in PROC SQL
  - Is part of ANSI SQL
  - That means it will pass through to your database engine
  - May make use of indexes within the database
    - The index will be scanned based on the string up first %
    - Then filter the result based on the rest of the string
    - This is usually better than a full table scan with filter

- Like can be used in DATA step
  - As a stand-alone Where statement
  - As part of the set statement
  - Not as part of an IF statement

- SAS® Indexes will be used where possible – DATA and PROC SQL!

Some Examples

- **Where name like like_string – or**
  - Where name like like_string escape ‘/’

<table>
<thead>
<tr>
<th>Initial List</th>
<th>Like String</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diana</td>
<td>Dav%</td>
<td>David</td>
</tr>
<tr>
<td>Diane</td>
<td>Dav%</td>
<td>Diane</td>
</tr>
<tr>
<td>Dianna</td>
<td>Dav%</td>
<td>Dianna</td>
</tr>
<tr>
<td>Dianthus</td>
<td>Dav%</td>
<td>Dianthus</td>
</tr>
<tr>
<td>Dyan</td>
<td>Dav%</td>
<td>Dyan</td>
</tr>
<tr>
<td>David</td>
<td>Dav%id</td>
<td>Dav(id)</td>
</tr>
<tr>
<td>Dav%id</td>
<td>Dav%id</td>
<td>Dav(id)</td>
</tr>
<tr>
<td>Dav/Id</td>
<td>Dav/%id</td>
<td>Dav(id)</td>
</tr>
<tr>
<td>Dav/%id</td>
<td>Dav/%id</td>
<td>Dav(id)</td>
</tr>
</tbody>
</table>

Some More Examples

- Like can be used in many contexts

```sql
proc print data=list (where=(name like 'David')); run;
```

```sql
proc print data=list (where=(name like 'Dav//id' escape '/'));
run;
```

```sql
data list2;
    set list (where=(name like 'Dav/%%' escape '/'));
run;
```

```sql
data list2;
    set list;
    where name like 'Dav/%%' escape '/';
run;
```
Alternatives

• Contains
  • Equivalent to like "%string%"
  • Unlikely to pass through to underlying database
  • Will use SAS Indexes

• Prxmatch
  • Allows extremely complex search strings via Regular Expressions
  • Definitely will not pass through
  • Will not use SAS indexes

• There is limited functionality with EQ:
  • if name eq: 'Dav';
  • Equivalent to name like "Dav/%" escape "/"

Cautions

In any commands in this presentation, the single and double quotation marks should be simple, not the “smart quotes” forced by Microsoft. The same applies to dashes or minus signs – they should not be “em dashes” (- versus –)

You can safely ignore this warning:

```sas
where product_group like "%Kid's%";
WARNING: Apparent invocation of macro KID not resolved.
```
4121 - Wildcarding in WHERE Clauses

David B. Horvath, CCP, MS

The Author can be contacted at:

David B. Horvath, CCP
504 Longbotham Drive, Aston PA 19014-2502, USA
Phone: 1-610-859-8826
Email: dhorvath@cobs.com
Web: http://www.cobs.com/
LI: http://www.linkedin.com/in/dbhorvath

All trademarks and service marks are the property of their respective owners.

References

https://documentation.sas.com/?docsetId=acreldb&docsetTarget=n05b4mygsvt845n1vnr6r5kchbjf.htm&docsetVersion=9.4&locale=en

The Basics of Using SAS® Indexes