SAS®
GLOBAL FORUM
2020

MARCH 29 – APRIL 1
WASHINGTON, DC

USERS PROGRAM

SAS GLOBAL FORUM 2020

MARCH 29 - APRIL 1
WASHINGTON, DC

Abstract
Basics
Methods
Alternatives
Conclusion

4121 - Wildcarding in WHERE Clauses

David B. Horvath, CCP, MS

- 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 '/'));

Main Author

SAS GLOBAL FORUM 2020
MARCH 29 - APRIL 1
WASHINGTON, DC

Abstract
Basics
Methods
Alternatives
Conclusion

4121 - Wildcarding in WHERE Clauses

David B. Horvath, CCP, MS

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.
```

SAS® GLOBAL FORUM 2020

MARCH 29 - APRIL 1 WASHINGTON, DC

- Abstract
- Basics
- **Methods**
- Alternatives
- Conclusion

4121 - Wildcarding in WHERE Clauses

David B. Horvath, CCP, MS

-

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 '/'

Initial List
Diana
Diane
Dianna
Dianthus
Dyan
David
Dav%id
Dav_id
Dav/id
Dav/%id

like_string	Result
David	David
like_string	name
Dav%	David
Dav%	Dav%id
Dav%	Dav_id
Dav%	Dav/id
Dav%	Dav/%id
like_string	name
Dav_%	David
Dav_%	Dav%id
Dav_%	Dav_id
Dav_%	Dav/id
Dav_%	Dav/%id
like_string	name
Dav/%	Dav/id
Dav/%	Dav/%id
like_string	name
Dav/%id	Dav/id
Dav/%id	Dav/%id

like_string	name
Dav/%id escape	Dav%id
like_string	name
Dav/%% escape	Dav%id
like_string	name
Dav/id	Dav/id
like_string	name
Dav//id escape	Dav/id
like_string	name
Dia__	Diana
Dia__	Diane
like_string	name
Di% or Dy%	Diana
Di% or Dy%	Diane
Di% or Dy%	Dianna
Di% or Dy%	Dianthus
Di% or Dy%	Dyan

like_string	name
D_a%	Diana
D_a%	Diane
D_a%	Dianna
D_a%	Dianthus
D_a%	Dyan
like_string	name
%a	Diana
%a	Dianna
like_string	name
%nn%	Dianna
like_string	name
Dav/% escape	NONE
like_string	name
Dav_	NONE

Some More Examples

- Like can be used in many contexts

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

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

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

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

SAS® GLOBAL FORUM 2020

MARCH 29 - APRIL 1
WASHINGTON, DC

Abstract
Basics
Methods
Alternatives
Conclusion

4121 - Wildcarding in WHERE Clauses

David B. Horvath, CCP, MS

Altenatives

- 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 my Microsoft. The same applies to dashes or minus signs – they should not be “em dashes” (- versus –)

You can safely ignore this warning:

```
where product_group like "%Kid's%";
WARNING: Apparent invocation of macro KID not
resolved.
```

4121 - Wildcarding in WHERE Clauses

David B. Horvath, CCP, MS

Abstract
Basics
Methods
Alternatives
Conclusion

Contact

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 servicemarks are the property of their respective owners.

References

SAS® 9.4 SQL Procedure User’s Guide, Fourth Edition:
https://documentation.sas.com/?docsetId=acreldb&docsetTarget=n05b4mygsvt845n1vnr6r5kchbjf.htm&docsetVersion=9.4&locale=en

The Basics of Using SAS® Indexes
https://support.sas.com/resources/papers/proceedings/proceedings/sugi30/247-30.pdf

SAS® GLOBAL FORUM 2020

USERS PROGRAM

MARCH 29 – APRIL 1 | WASHINGTON, DC | #SASGF