

How to Use LIKE Operator in SAS

Do you often need to search or subset your data using character or free-text variables based on keywords? Would you like to learn more or better understand how to conduct keyword searches and simple text mining exercises with SAS character variables? This article demonstrates how to use the LIKE operator with wildcards using both SAS Data Step and PROC SQL to enable you to conduct more in-depth keyword searches with your data.

In particular, this article will cover the following topics:

1. Using SAS data step

- Select values that start with a character string
 - Adjusting for Different Letter Cases
- Select values that end with a character string
- Select values the contain a character string
- Select values with multiple like conditions

2. Using PROC SQL

- Select values that start with a character string
- Select values that end with a character string
- Select values the contain a character string
- Select values with multiple like conditions

Software

Before we continue, make sure you have SAS Studio or SAS 9.4 installed. Don't have the software? Download [SAS Studio](#) now. It's free!

SAS Studio

Data Sets

The following datasets found in the SASHELP library are used in this article for various examples:

1. SHOES - Fictitious Shoe Company Data
2. CARS – Car Data
3. ORSALES – Orion Start Sports & Outdoor Sales Data

Using WHERE LIKE with SAS Data Step

With a SAS Data Step, the LIKE operator is used in conjunction with a WHERE statement while the WHERE statement is used subset an input dataset.

In this example, the WHERE statement is used to select the Products which will be kept in the output dataset, called MENS_PRODUCTS.

Let's start with a simple example where we will create a new dataset in the WORK library called MENS_PRODUCTS, which contains all the records from SASHELP.SHOES where the product name is "Men's Dress".

To do this, we simply need a WHERE statement after the SET statement. The WHERE statement includes the variable name (PRODUCT), the LIKE operator, and finally the character string to search for which is "Men's Dress" in this example.

Note that double quotes (") are used here since we need to include the apostrophe in "Men's Dress". Single quotes can be used when the text string does not contain an apostrophe or other single quotes. The syntax is as follows:

```
data mens_products;  
  set sashelp.shoes;  
  where product like "Men's Dress";  
run;
```

In this first example, the LIKE operator behaves exactly like an equals sign (=) since we have not yet introduced any wild cards. The resulting dataset contains all the records which have Men's Dress as products, as you can see in the partial output below:

Picture

Selecting Values that Start with a Character String

The LIKE operator starts to become useful when the concept of wildcards is introduced. In SAS, the percentage sign (%) is used as the wildcard to conduct keyword searches for character variables that start with, end with or even contain certain character strings.

Building upon the previous example, let's now select all the records for SHOES which have a product that starts with the word "Men's". To do this, we use the same syntax as previously, but now add the wildcard (%) sign after the word "Men's" inside the quotation marks:

```
data mens_products;  
set sashelp.shoes;  
  where product like "Men's%";  
run;
```

Now, in the the output dataset shown partially below, you can see all those records with products starting with the text string “Men’s”.

Picture

Adjusting for Different Letter Cases

When conducting text string searches with the WHERE statement and LIKE operator, it’s important to keep in mind that the search terms in quotation marks are case sensitive.

For example, by running the following code, you will not find any records in SASHELP.SHOES that contain products which start with “men’s” since all the values are written as “Men’s” and thus the code will not return any records in the output dataset:

```
data mens_products;  
set sashelp.shoes;  
  where product like "men's%";  
run;
```

When you know the values in your dataset contain a mixture of upper and lower case values or you are unsure of the cases used for the values in your dataset, you can use the UPPERCASE or LOWERCASE functions in conjunction with your WHERE statement and LIKE operator.

For example, by applying the UPPERCASE function to PRODUCT you can transform all the characters to uppercase temporarily on input so that you only need to search for values that start with “MEN” (in all upper case letters):

```
data mens_products;  
set sashelp.shoes;  
  where uppercase(product) like "MEN%";  
run;
```

Similarly, you can apply the LOWERCASE function to PRODUCT and search for values that start with “men” (in all lower case letters):

```
data mens_products;  
set sashelp.shoes;  
  where lowcase(product) like "men%";  
run;
```

With either scenario, you can achieve the desired result shown partially below, without having to worry about which case your values might be in:

Picture

Selecting Values that End with a Character String

Similar to selecting variables that start with a character string, the wildcard (%) can be moved to the front the character string in quotations to find records which contain variables ending with the specified character string.

In this example, we would like to keep all those records in a newly created dataset, CASUAL which have products ending with the word “Casual”. To achieve this, Casual is added in quotation marks with the wildcard (%) placed in front of the word “Casual”:

```
data casual;  
set sashelp.shoes;  
  where product like '%Casual';  
run;
```

As you can see in the WORK.CASUAL dataset shown partially below, we now have all the records which contain products ending in the word “Casual”:

Picture

Selecting Values that Contain a Character String

In addition to having the ability to select values that either start with or end with a character string, you can also retrieve records which contain a character string anywhere within the value.

For example, in the SASHELP.CARS dataset, cars with 2 doors contain the string “2dr” somewhere within the model name. Sometimes “2dr” is found at the end of the string, or sometimes it is in the middle, as you can see below:

Picture

To retrieve all the records from SASHELP.CARS which are 2 door models, we can simply add the wildcard (%) both before and after our search term “2dr” in the WHERE statement as shown below:

```
data two_door_cars;  
set sashelp.cars;
```

```
where model like '%2dr%';  
run;
```

As you can see in the Output Data shown partially below, we now have a new dataset in WORK, TWO_DOOR_CARS, which contains all those cars with “2dr” somewhere within the Model name:

Picture

Selecting Values with Multiple LIKE Conditions

The LIKE operator can also be used in conjunction with other operators, such as the AND or OR operators.

If for example you wanted to retrieve 2 door convertible cars from SASHELP.CARS, you can use multiple LIKE operators with an AND operator to apply both conditions when sub setting the data. To achieve this, simply add an AND after the first LIKE and then include a second LIKE which contains the second sub setting condition.

```
In the syntax below, the cars containing both “2dr” and “convertible” in the values for  
MODEL are being selected from the input dataset to create a new dataset,  
TWO_DR_CONVERTIBLES:  
data two_dr_convertibles;  
set sashelp.cars;  
  where model like '%2dr' and model like '%convertible%';  
run;
```

In the TWO_DR_CONVERTIBLES output data set shown partially below, you can see that SASHELP.CARS has now been subset to only include those records with both “2dr” and “convertible” somewhere in the value for MODEL:

Picture

Using WHERE LIKE with PROC SQL

The same principles for WHERE LIKE with SAS Data Step can also be applied to PROC SQL. Using the SASHELP.ORDSALES dataset which contains product sales information for a sports and outdoor store, let’s look at a few examples using the LIKE operator with PROC SQL.

Selecting Values that Start with a Character String

In this first example, let's look at how to select the QUARTER, PRODUCT_CATEGORY, PRODUCT_GROUP and PROFIT variables from SASHELP.ORSALES where the values of PRODUCT_GROUP start with "Golf".

The syntax using PROC SQL is essentially identical to that of the Data Step. The wildcard (%) is added after the word "Golf" in quotations with the only difference being that now the WHERE statement and LIKE operator are placed after a PROC SQL SELECT statement instead of a dataset SET statement:

```
proc sql;  
  select quarter, product_category, product_group, profit  
    from sashelp.orsales  
   where product_group like 'Golf%'  
  ;  
quit;
```

As you can see in the partial Results shown below, all those records with PRODUCT_GROUP values starting with the word "Golf" have been selected:

Picture

Selecting Values that End with a Character String

Of course, you can also use the LIKE operator with a PROC SQL SELECT statement to select those values that end with a character string.

To select all those records which have a PRODUCT_GROUP value ending in "Clothes", we can simply add the wildcard (%) in front of the word clothes in the PROC SQL call, as shown here:

```
proc sql;  
  select quarter, product_category, product_group, profit  
    from sashelp.orsales  
   where product_group like '%Clothes'  
  ;  
quit;
```

In the Results shown below, you can see that all the records with PRODUCT_GROUP ending in "Clothes" have been selected:

Picture

Selecting Values that Contain a Character String

As with the Data Step, you can select values which contain a character string anywhere within a value by adding the wildcard (%) both before and after the desired character string. For example, to select all those records which contain “Kid’s” somewhere within the PRODUCT_GROUP variable values, you would use the following syntax:

```
proc sql;  
  select quarter, product_category, product_group, profit  
  from sashelp.orsales  
  where product_group like "%Kid's%"  
  ;  
quit;
```

After running the code above, you will likely notice this WARNING in your SAS Log:

Picture

This is because the % sign is also used to denote macros within SAS, and so SAS is warning you that there is no macro called KID found in your SAS session. Despite the WARNING in this case, the desired results are still achieved, as you can see in the RESULTS shown partially below:

Picture

Selecting Values with Multiple LIKE conditions

In the previous example, you may have noticed when looking through the entire SASHELP.ORSALES dataset that Kid’s can be found written as “Kid’s” or “Kids”. To ensure that we select records under both scenarios, we can combine multiple LIKE operators using an OR statement, similar to how the AND operator was used in the data step example earlier in this article.

To select those records with PRODUCT_GROUP values which contain either “Kid’s” or “Kids”, simply add the OR operator as shown in the syntax below:

```
proc sql;  
  select quarter, product_category, product_group, profit  
  from sashelp.orsales  
  where product_group like "%Kid's%"  
  or product_group like "%Kids%"  
  ;  
quit;
```

In the Results shown partially below, you can see that we have now selected this records with PRODUCT_GROUP values containing either “Kid’s” or “Kids”:

Picture