How can I use the ORDER BY clause in PROC SQL to implement the SAS approach of sorting data in a specific order?

Authored by

stats writer

June 23, 2024

The ORDER BY clause is a powerful tool in PROC SQL that allows for the implementation of the SAS approach of sorting data in a specific order. By using the ORDER BY clause, users can specify the columns and their respective sorting order, ensuring that the data is organized in a desired manner. This feature is particularly useful when working with large datasets or when specific data arrangements are necessary for analysis or reporting purposes. By utilizing the ORDER BY clause, users can efficiently manipulate and present data in a structured and meaningful way, following the established SAS methodology.

SAS: Use ORDER BY in PROC SQL

You can use the ORDER BY statement in PROC SQL in SAS to order the results of a query by the values of one or more variables.

Here are three common ways to use the ORDER BY statement in practice:

Method 1: Order By One Variable Ascending

```
/*display results in ascending order by value in team column*/
proc sql;
select *
from my_data
order by team;
quit;
```

Method 2: Order By One Variable Descending

```
/*display results in descending order by value in team column*/
proc sql;
select *
from my_data
order by team desc;
quit;
```

Method 3: Order By Multiple Variables

```
/*display results in ascending order by team, then descending order by points*/
proc sql;
select *
from my_data
order by team, points desc;
quit;
```

The following examples show how to use each method in practice with the following dataset in SAS that contains information about various basketball players:

```
/*create dataset*/
data my_data;
```

```
input team $ position $ points assists;
datalines;
A Guard 14 4
B Guard 22 6
B Guard 24 9
A Forward 13 8
C Forward 13 9
A Guard 10 5
B Guard 24 4
C Guard 22 6
D Forward 34 2
D Forward 15 5
B Forward 23 5
B Guard 10 4
;
run;

/*view dataset*/
proc printdata=my_data;
```

Obs	team	position	points	assists
1	A	Guard	14	4
2	B	Guard	22	6
3	B	Guard	24	9
4	A	Forward	13	8
5	C	Forward	13	9
6	A	Guard	10	5
7	B	Guard	24	4
8	C	Guard	22	6
9	D	Forward	34	2
10	D	Forward	15	5
11	B	Forward	23	5
12	B	Guard	10	4

Example 1: Order By One Variable Ascending

The following code shows how to return every row in the dataset in ascending order by the values in the team column:

```
/*display results in ascending order by value in team column*/
proc sql;
select *
from my_data
order by team;
quit;
```

team	position	points	assists
A	Forward	13	8
A	Guard	14	4
A	Guard	10	5
B	Guard	24	4
B	Guard	24	9
B	Guard	10	4
B	Forward	23	5
B	Guard	22	6
C	Forward	13	9
C	Guard	22	6
D	Forward	15	5
D	Forward	34	2

Notice that the results are shown in ascending order by the values in the team column.

Example 2: Order By One Variable Descending

```
/*display results in descending order by value in team column*/
proc sql;
select *
from my_data
order by team desc;
quit;
```

team	position	points	assists
D	Forward	34	2
D	Forward	15	5
C	Forward	13	9
C	Guard	22	6
B	Guard	24	4
B	Guard	10	4
B	Forward	23	5
B	Guard	24	9
B	Guard	22	6
A	Guard	10	5
A	Guard	14	4
A	Forward	13	8

Notice that the results are shown in descending order by the values in the team column.

Example 3: Order By Multiple Variables

The following code shows how to return every row in the dataset first in ascending order by team, then in descending order by points:

```
/*display results in ascending order by team, then descending order by points*/
proc sql;
select *
from my_data
```

```
order by team, points desc;
quit;
```

team	position	points	assists
A	Guard	14	4
A	Forward	13	8
A	Guard	10	5
B	Guard	24	9
B	Guard	24	4
B	Forward	23	5
B	Guard	22	6
B	Guard	10	4
C	Guard	22	6
C	Forward	13	9
D	Forward	34	2
D	Forward	15	5

Notice that the results are shown first in ascending order by team, then in descending order by points.

The following tutorials explain how to perform other common tasks in SAS: