How can I use the SAS UPDATE statement within PROC SQL for data manipulation?

Authored by

stats writer

June 25, 2024

The SAS UPDATE statement, used within PROC SQL, allows users to modify or update data within a SAS dataset. This statement is useful for manipulating data in a more efficient and streamlined manner, as it can perform multiple updates in a single step. It follows a specific syntax and requires the use of a WHERE clause to specify which data rows should be updated. With the SAS UPDATE statement, users can easily make changes to their data without having to create new datasets, saving time and improving data management processes.

SAS: Use UPDATE Within PROC SQL

You can use the UPDATE statement within PROC SQL in SAS to update the values in one or more columns of dataset.

Here are the most common ways to use the UPDATE statement in practice:

Method 1: Update Values in Column Based on One Condition

```
proc sql;
update my_data
set var1='new_value'
where var1='old_value';
quit;
```

Method 2: Update Values in Column Based on Multiple Conditions

```
proc sql;
update my_data
set var1 =
case when var1>25 then 100
when var1>20 then 50
else 0
end;
quit;
```

The following examples show how to use each method in practice with the following dataset in SAS:

```
/*create dataset*/
data my_data;
input team $ position $ points;
datalines;
A Guard 22
A Guard 20
A Guard 30
A Forward 14
A Forward 11
B Guard 12
B Guard 22
B Forward 30
```

```
B Forward 9
B Forward 12
B Forward 25
;
run;

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

Obs	team	position	points
1	A	Guard	22
2	A	Guard	20
3	A	Guard	30
4	A	Forward	14
5	A	Forward	11
6	B	Guard	12
7	B	Guard	22
8	B	Forward	30
9	B	Forward	9
10	B	Forward	12
11	B	Forward	25

Example 1: Update Values in Column Based on One Condition

We can use the following UPDATE statement within PROC SQL to update each of the values in the team column to be 'Atlanta' where the existing values are equal to 'A':

```
/*update values in team column where team is equal to 'A'*/
proc sql;
update my_data
set team='Atlanta'
where team='A';
quit;

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

Obs	team	position	points
1	Atlanta	Guard	22
2	Atlanta	Guard	20
3	Atlanta	Guard	30
4	Atlanta	Forward	14
5	Atlanta	Forward	11
6	B	Guard	12
7	B	Guard	22
8	B	Forward	30
9	B	Forward	9
10	B	Forward	12
11	B	Forward	25

Notice that each value in the team column that used to be equal to 'A' is now equal to 'Atlanta.'

Any values that were not equal to 'A' in the team column were simply left unchanged.

Example 2: Update Values in Column Based on Multiple Conditions

We can use the following UPDATE statement within PROC SQL to update each of the values in the points column based on several conditions:

```
/*update values in points column based on multiple conditions*/
proc sql;
update my_data
set points =
case when points>25 then 100
when points>20 then 50
else 0
end;
quit;

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

Obs	team	position	points
1	A	Guard	50
2	A	Guard	0
3	A	Guard	100
4	A	Forward	0
5	A	Forward	0
6	B	Guard	0
7	B	Guard	50
8	B	Forward	100
9	B	Forward	0
10	B	Forward	0
11	B	Forward	50

We used the UPDATE statement along with a CASE WHEN statement to update the values in the points column.

In particular:

If the existing value in the points column was greater than 25, we updated it to be 100.Else, if the existing value in the points column was greater than 20, we updated it to be 50.Else, we updated the value in the points column to be 0.

Note that we only used three conditions in the CASE WHEN statement but you can use as many conditions as you'd like.

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