

# How can the SAS SELECT-WHEN statement be used in a SAS program?

Authored by  
**stats writer**

June 29, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can the SAS SELECT-WHEN statement be used in a SAS program?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=160230>

The SAS SELECT-WHEN statement is a conditional statement that can be used in a SAS program to execute specific actions based on a series of conditions. It works by evaluating a series of logical expressions and executing the corresponding action when a condition is met. This statement is often used in conjunction with the SAS DO statement to create more complex logic in a SAS program.

For example, suppose we have a dataset containing information about employees, including their salary and job title. We want to create a new variable called "Salary Bracket" that categorizes each employee's salary into three groups: "Low", "Medium", and "High". We can use the SAS SELECT-WHEN statement to achieve this.

Example SAS code:

```
data employees;
set employees;
select;
when (salary = 50000 and salary = 100000) bracket = 'High';
end;
run;
```

In this example, the SELECT-WHEN statement evaluates the value of the "salary" variable for each employee and assigns the appropriate value to the "bracket" variable based on the conditions specified. This allows for efficient and concise code, making it easier to manage and maintain the program.

## Use SELECT-WHEN in SAS (With Example)

**You can use a SELECT-WHEN statement in SAS to assign values to a new variable based on the values of an existing categorical variable in a dataset.**

**This statement uses the following basic syntax:**

```
data new_data;
set my_data;
```

```
select (Existing_Column);  
when ('value1') New_Column=1;  
when ('value2') New_Column=2;  
when ('value3') New_Column=3;  
otherwise New_Column=4;  
end;  
run;
```

This syntax produces a new column called `New_Column` whose values are dependent on the values in `Existing_Column`.

The following example shows how to use a **SELECT-WHEN** statement in practice.

Example: **SELECT-WHEN** in SAS

Suppose we have the following dataset in SAS that contains information about various basketball players:

```
/*create dataset*/  
data my_data;  
input team $ rating $ points;  
datalines;  
Mavs Great 22
```

**Mavs Good 29**

**Mavs OK 15**

**Mavs Bad 8**

**Spurs Good 30**

**Spurs OK 15**

**Spurs OK 20**

**Spurs Bad 7**

;

**run; /\*view dataset\*/**

**proc printdata=my\_data;**

Obs	team	rating	points
1	Mavs	Great	22
2	Mavs	Good	29
3	Mavs	OK	15
4	Mavs	Bad	8
5	Spurs	Good	30
6	Spurs	OK	15
7	Spurs	OK	20
8	Spurs	Bad	7

**We can use the following SELECT-WHEN statement to create a new variable called Player\_Status whose values depend on the value in the rating column:**

**/\*create new dataset with Player\_Status column\*/**

```
data new_data;
set my_data;
select (rating);
when ('Great') Player_Status=1;
when ('Good') Player_Status=2;
when ('OK') Player_Status=3;
otherwise Player_Status=4;
end;
run;

/*view new dataset*/
proc printdata=new_data;
```

Obs	team	rating	points	Player_Status
1	Mavs	Great	22	1
2	Mavs	Good	29	2
3	Mavs	OK	15	3
4	Mavs	Bad	8	4
5	Spurs	Good	30	2
6	Spurs	OK	15	3
7	Spurs	OK	20	3
8	Spurs	Bad	7	4

**Here is how the values were generated in the new Player\_Status column:**

If rating was equal to "Great" then Player\_Status was assigned 1.

If rating was equal to "Good" then Player\_Status was assigned 2.

If rating was equal to "OK" then Player\_Status was assigned 3.

If rating was not equal to any of the previously specified values then Player\_Status was assigned 4.

**Note:** You can find the complete documentation for the **SELECT** statement in **SAS** .

#### Additional Resources

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