

# Can the SAS (Statistical Analysis System) use the “in=a” syntax in a Merge statement?

Authored by  
**stats writer**

June 23, 2024

## RECOMMENDED CITATION

stats writer (2024). *Can the SAS (Statistical Analysis System) use the “in=a” syntax in a Merge statement?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=149385>

The SAS (Statistical Analysis System) is a widely used software for data analysis and management. It offers various tools and functions to manipulate, merge, and analyze data. One of the commonly used functions is the "Merge" statement, which combines data sets based on matching values in specified variables. In this statement, the "in=a" syntax allows for the creation of an indicator variable that indicates whether a particular observation was present in the merged data set or not. This feature is useful for tracking and identifying the source of observations in the merged data set. Therefore, the SAS is capable of using the "in=a" syntax in a Merge statement.

## **SAS: Use (in=a) in Merge Statement**

**When merging two datasets in SAS, you can use the IN statement to only return rows where a value exists in a particular dataset.**

**Here are a few common ways to use the IN statement in practice:**

**Method 1: Return Rows where Value Exists in First Dataset (in = a)**

```
data final_data;  
merge data1 (in=a) data2;  
by ID;  
if a;  
run;
```

**This particular example merges the datasets called data1 and data2 and only returns the rows where a**

**value exists in data1.**

## **Method 2: Return Rows where Value Exists in Second Dataset (in = b)**

```
data final_data;  
merge data1 data2 (in=b);  
by ID;  
if b;  
run;
```

**This particular example merges the datasets called data1 and data2 and only returns the rows where a value exists in data2.**

## **Method 3: Return Rows where Value Exists in Both Datasets (in = a) and (in = b)**

```
data final_data;  
merge data1 (in = a) data2 (in=b);  
by ID;  
if a and b;  
run;
```

**This particular example merges the datasets called**

**data1 and data2 and only returns the rows where a value exists in *both* data1 and data2.**

**The following examples show how to use each method in practice with the following two datasets:**

```
/*create first dataset*/
```

```
data data1;
```

```
input ID Gender $;
```

```
datalines;
```

```
1 Male
```

```
2 Male
```

```
3 Female
```

```
4 Male
```

```
5 Female
```

```
;
```

```
run;
```

```
title "data1";
```

```
proc printdata = data1;
```

```
/*create second dataset*/
```

```
data data2;
```

```
input ID Sales;
```

```
datalines;
```

**1 22**

**2 15**

**4 29**

**6 31**

**7 20**

**8 13**

**;**

**run;**

**title "data2";**

**proc printdata = data2;**

ARABPSYCHOLOGY.COM

**data1**

Obs	ID	Gender
1	1	Male
2	2	Male
3	3	Female
4	4	Male
5	5	Female

**data2**

Obs	ID	Sales
1	1	22
2	2	15
3	4	29
4	6	31
5	7	20
6	8	13

### Example 1: Return All Rows

We can use the following merge statement without any IN statement to merge the two datasets based on the value in the ID column and return all rows from both datasets:

```
/*perform merge*/  
data final_data;  
merge data1 data2;  
by ID;
```

```
run;
```

```
/*view results*/
```

```
title "final_data";
```

```
proc printdata=final_data;
```

Obs	ID	Gender	Sales
1	1	Male	22
2	2	Male	15
3	3	Female	.
4	4	Male	29
5	5	Female	.
6	6		31
7	7		20
8	8		13

Notice that all rows from both datasets are returned, regardless if there are missing values due to an ID value not existing in both datasets.

Example 2: Return Rows where Value Exists in First Dataset (in = a)

We can use the following merge statement with (in = a) to merge the two datasets based on the value in the ID column and return only the rows where a value exists in

**the first dataset:**

```
/*perform merge*/  
data final_data;  
merge data1 (in = a) data2;  
by ID;  
if a;  
run;
```

```
/*view results*/  
title "final_data";  
proc printdata=final_data;
```

**final\_data**

Obs	ID	Gender	Sales
1	1	Male	22
2	2	Male	15
3	3	Female	.
4	4	Male	29
5	5	Female	.

**Notice that only the rows where a value exists in the first dataset are returned.**

**Example 3: Return Rows where Value Exists in Second Dataset (in**

= b)

We can use the following merge statement with (in = b) to merge the two datasets based on the value in the ID column and return only the rows where a value exists in the second dataset:

```
/*perform merge*/  
data final_data;  
merge data1 data2 (in = b);  
by ID;  
if b;  
run;  
  
/*view results*/  
title "final_data";  
proc printdata=final_data;
```

**final\_data**

Obs	ID	Gender	Sales
1	1	Male	22
2	2	Male	15
3	4	Male	29
4	6		31
5	7		20
6	8		13

**Notice that only the rows where a value exists in the second dataset are returned.**

**Example 4: Return Rows where Value Exists in Both Datasets (in = a) and (in = b)**

**We can use the following merge statement with (in = a) and (in = b) to merge the two datasets based on the value in the ID column and return only the rows where a value exists in both datasets:**

```
/*perform merge*/  
data final_data;  
merge data1 (in = a) data2 (in = b);  
by ID;  
if a and b;  
run;
```

```
/*view results*/  
title "final_data";  
proc printdata=final_data;
```

**final\_data**

Obs	ID	Gender	Sales
1	1	Male	22
2	2	Male	15
3	4	Male	29

**Notice that only the rows where a value exists in both datasets are returned.**

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

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