

# Can the SAS SAS: Use EXCEPT in PROC SQL be used as an alternative method for combining datasets in PROC SQL?

Authored by  
**stats writer**

June 25, 2024

## RECOMMENDED CITATION

stats writer (2024). *Can the SAS SAS: Use EXCEPT in PROC SQL be used as an alternative method for combining datasets in PROC SQL?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153079>

The "Can the SAS SAS: Use EXCEPT in PROC SQL be used as an alternative method for combining datasets in PROC SQL?" is a question that aims to explore the possibility of using the EXCEPT function in PROC SQL as an alternative way of merging datasets. This function allows for the comparison of two datasets and the exclusion of any matching rows from the resulting dataset. This potentially offers a different approach to combining datasets compared to traditional methods such as the JOIN function. By utilizing this feature, PROC SQL users can potentially achieve more efficient and precise results in their data merging processes. This inquiry seeks to determine the viability and benefits of using the EXCEPT function in PROC SQL for dataset combination purposes.

## **SAS: Use EXCEPT in PROC SQL**

**You can use the EXCEPT operator in the PROC SQL statement in SAS to only return rows from one dataset that are not in another dataset.**

**The following example shows how to use the EXCEPT operator in practice.**

**Example: Using EXCEPT in PROC SQL in SAS**

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

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

```
data data1;
```

```
input team $ points;
```

```
datalines;
```

```
A 12
```

**A 14**

**A 15**

**A 18**

**A 20**

**A 22**

**;**

**run;**

**/\*view first dataset\*/**

**proc printdata=data1;**

Obs	team	points
1	A	12
2	A	14
3	A	15
4	A	18
5	A	20
6	A	22

**And suppose we have another dataset in SAS that also contains information about various basketball players:**

**/\*create second dataset\*/**

**data data2;**

**input team \$ points;**

**datalines;**

**A 12**

**A 14**

**B 23**

**B 25**

**B 29**

**B 30**

;

**run;**

**/\*view second dataset\*/**

**proc printdata=data2;**

Obs	team	points
1	A	12
2	A	14
3	B	23
4	B	25
5	B	29
6	B	30

**We can use the EXCEPT operator in the PROC SQL statement to only return the rows from the first dataset that are not in the second dataset**

**/\*only return rows from first dataset that are not in second dataset\*/**

```
proc sql;  
title 'data1 EXCEPT data2';  
select * from data1  
exceptselect * from data2;  
quit;
```

**data1 EXCEPT data2**

team	points
A	15
A	18
A	20
A	22

Notice that only the rows in the first dataset that do not belong to the second dataset are returned.

We can also use the EXCEPT operator to only return the rows from the second dataset that are not in the first dataset:

```
/*only return rows from second dataset that are not in  
first dataset*/
```

```
proc sql;  
title 'data2 EXCEPT data1';  
select * from data2
```

```
exceptselect * from data1;  
quit;
```

**data2 EXCEPT data1**

team	points
B	23
B	25
B	29
B	30

**Notice that only the rows in the second dataset that do not belong to the first dataset are returned.**

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