PROC SQL

Do PROC SQL Joins and DATA Step Merges produce the same results?

Be careful when combining datasets horizontally when there are columns with the same names in the contribution tables, but not specified in the join condition as the output will differ depending on the syntax used.

Sample datasets can be produced by submitting the following code:

```
data left ;
  key = '01' ;
  col1 = 999 ;
  col2 = 'left  ' ;
  output ;
  key = '02' ;
  col1 = . ;
  col2 = '' ;
  output ;
run ;

data right ;
  key = '01' ;
  col1 = 111 ;
  col2 = 'right1' ;
  output ;
  key = '02' ;
  col1 = 222 ;
  col2 = 'right2' ;
  output ;
run ;
```

Each dataset has the same columns, and matching values of the KEY variable. The remaining columns are common to both, but have no matching values.

LEFT table:

key	col1	col2
01	999	left
02	.	

(c) 2025 Alan D. Rudland <aland@pinkchameleon.co.uk> | 19-10-2025 21:12

URL: http://pinkchameleon.co.uk/index.php?action=artikel&cat=3&id=9&artlang=en

RIGHT table

key	col1	col2
01	111	right1
02	222	right2

Joining the datasets with a DATA Step MERGE:

```
data combine ;
  merge left
        right
 ;
    by key ;
run ;
```

will generate a table where the last-named dataset determines the values of the commonly-named columns:

COMBINE table

key	col1	col2
01	111	right1
02	222	right2

Using PROC SQL to combine the data (LEFT | RIGHT | FULL | INNER JOINs will determine which observations are included according to normal rules). Default action for PROC SQL is to produce a report, where multiple columns of the same name can be included:

```
proc sql ;
  select *
  from left  as l
       left join
       right as r
       on l.key = r.key
```


(c) 2025 Alan D. Rudland <aland@pinkchameleon.co.uk> | 19-10-2025 21:12

URL: http://pinkchameleon.co.uk/index.php?action=artikel&cat=3&id=9&artlang=en

```
      ;
quit ;
```

Report generated:

key	col1	col2	key	col1	col2
01	999	left	01	111	right1
02	.		02	222	right2

If however a table is generated using PROC SQL:

```
proc sql ;
  create table combine_sql as
  select *
  from left  as l
       full join
       right as r
       on l.key = r.key
       ;
quit ;
```

then WARNINGs are generated in the LOG:

```
WARNING: Variable key already exists on file WORK.COMBINE_SQL.
WARNING: Variable col1 already exists on file WORK.COMBINE_SQL.
WARNING: Variable col2 already exists on file WORK.COMBINE_SQL.
```

and the table which is generated takes it values from the first-named table only.

COMBINE_SQL table

key	col1	col2

(c) 2025 Alan D. Rudland <aland@pinkchameleon.co.uk> | 19-10-2025 21:12

URL: http://pinkchameleon.co.uk/index.php?action=artikel&cat=3&id=9&artlang=en

01	999	left
02	.	

Unique solution ID: #1008
Author: Alan D Rudland
Last update: 2017-03-03 15:46

(c) 2025 Alan D. Rudland <aland@pinkchameleon.co.uk> | 19-10-2025 21:12
URL: http://pinkchameleon.co.uk/index.php?action=artikel&cat=3&id=9&artlang=en