ASK THE EXPERT

How Do I Combine Data in SAS®?

Dominique Weatherspoon
Senior Associate Technical Training Consultant, SAS

Dominique Weatherspoon

Senior Associate Technical Training Consultant, SAS

Dominique teaches people from a variety of experience levels how to code in the Base SAS Programming Language. SAS Programming 1, SAS Programming 2, SAS SQL Essentials and SAS Visual Analytics are among some of the classes she teaches. When she’s not leading classes via Zoom, she contributes to the SAS Users YouTube page with upbeat tips and tricks for new and experienced SAS programmers.

Combining Data in SAS

I. Concatenation

II. Joins/Merges

III. Point-and-click Methods

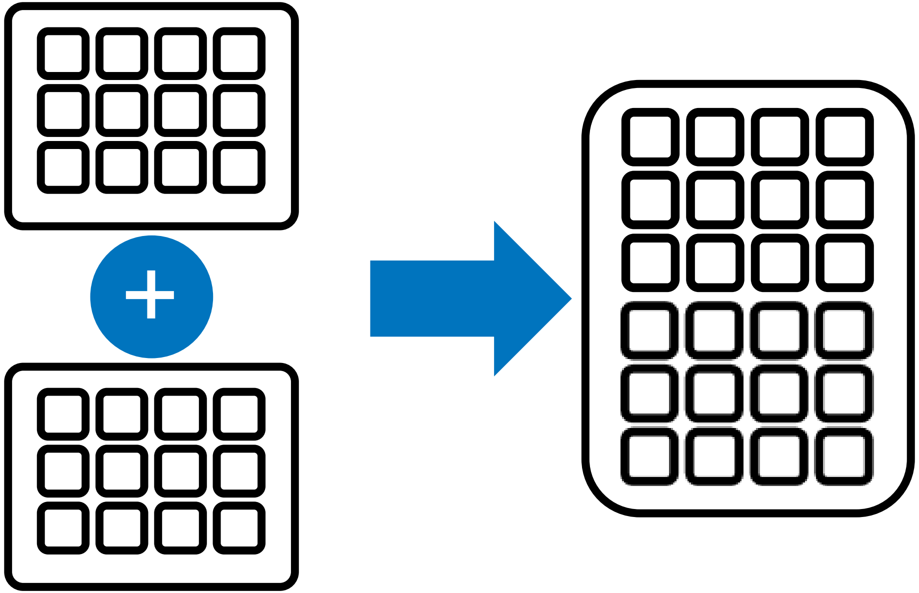
Combining Data in SAS

I. Concatenation

II. Joins/Merges

III. Point-and-click Methods

Concatenating Tables

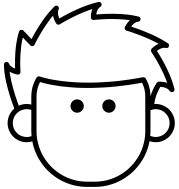


Concatenating Tables Using PROC SQL

```
SELECT query…
     OUTER UNION <CORR>
SELECT query…;
```

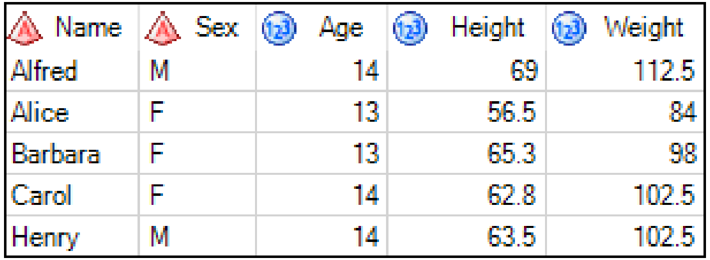
Use the Outer Union Set Operator

Use the SET statement when you want to combine tables with similar data in one output table.



```
proc sql;
create table class_current as
select * from sashelp.class
outer union corr
select * from mylib.class_new;
run
```

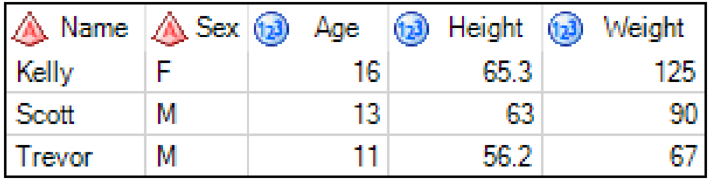
sashelp.class



Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5

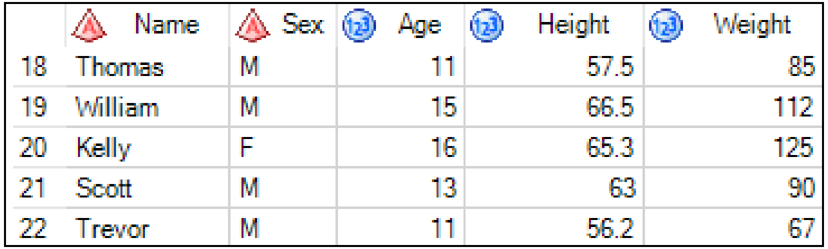
+

mylib.class_new



Name	Sex	Age	Height	Weight
Kelly	F	16	65.3	125
Scott	M	13	63	90
Trevor	M	11	56.2	67

class_current



	Name	Sex	Age	Height	Weight
18	Thomas	M	11	57.5	85
19	William	M	15	66.5	112
20	Kelly	F	16	65.3	125
21	Scott	M	13	63	90
22	Trevor	M	11	56.2	67

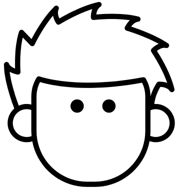
rows from second table added after rows from the first table

Concatenating Tables Using The DATA Step

```
DATA output-table;
      SET input-table1 input-table2 ...;
RUN;
```

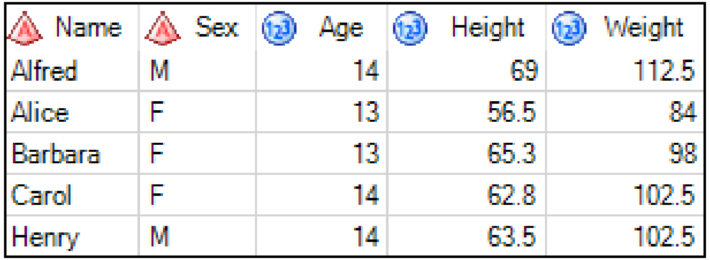
any number of input tables

Use the SET statement when you want to combine tables with similar data in one output table.



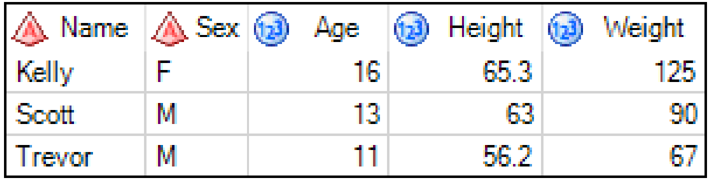
```
data class_current;
    set sashelp.class mylib.class_new;
run;
```

sashelp.class



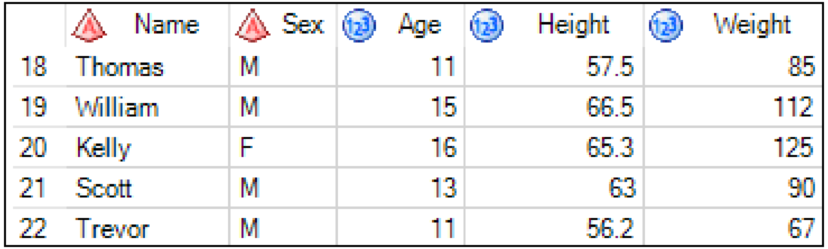
Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5

mylib.class_new



Name	Sex	Age	Height	Weight
Kelly	F	16	65.3	125
Scott	M	13	63	90
Trevor	M	11	56.2	67

class_current



	Name	Sex	Age	Height	Weight
18	Thomas	M	11	57.5	85
19	William	M	15	66.5	112
20	Kelly	F	16	65.3	125
21	Scott	M	13	63	90
22	Trevor	M	11	56.2	67

rows from second table added after rows from the first table

Demonstration

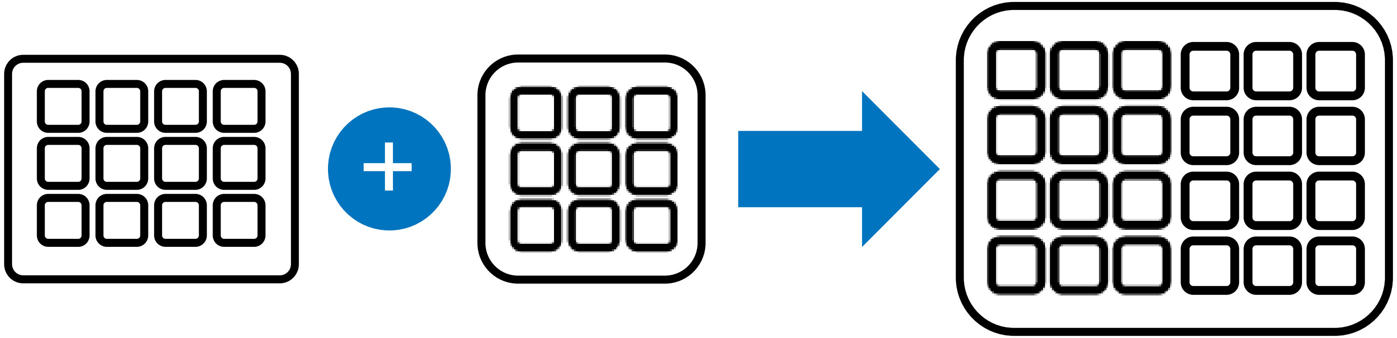
Combining Data in SAS

I. Concatenation

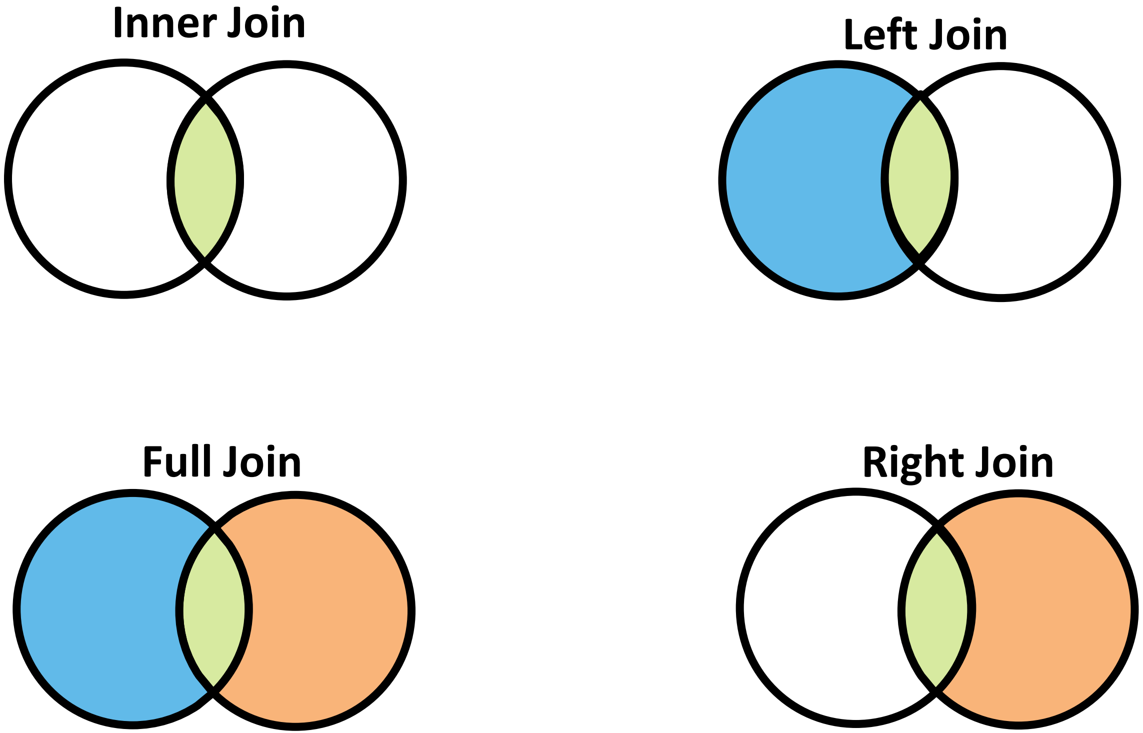
II. Joins/Merges

III. Point-and-click Methods

Merging Tables



Types of Joins



Inner Join

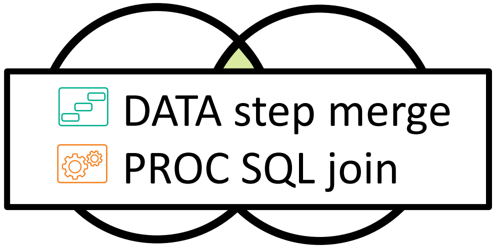
Left Join

Full Join

Right Join

Types of Joins

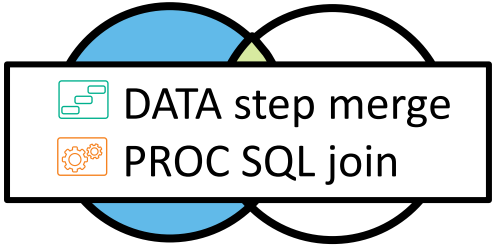
Inner Join



DATA step merge

PROC SQL join

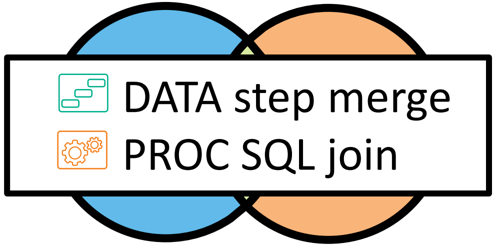
Left Join



DATA step merge

PROC SQL join

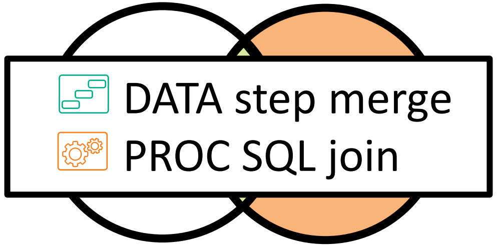
Full Join



DATA step merge

PROC SQL join

Right Join

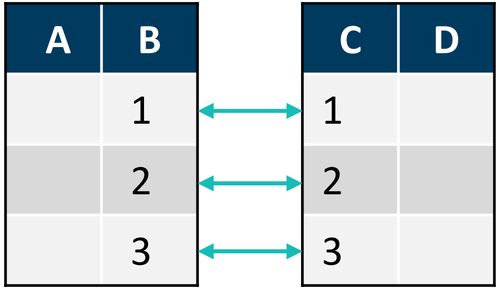


DATA step merge

PROC SQL join

Table Relationships

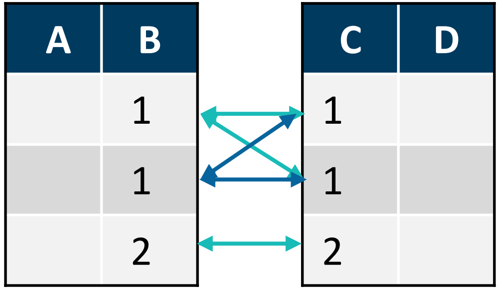
One-to-One



A	B
	1
	2
	3

C	D
1	
2	
3	

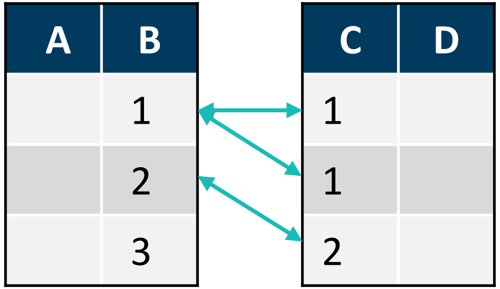
Many-to-Many



A	B
	1
	1
	2

C	D
1	
1	
2	

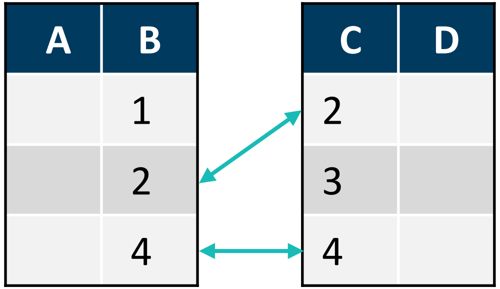
One-to-Many



A	B
	1
	2
	3

C	D
1	
1	
2	

Nonmatches

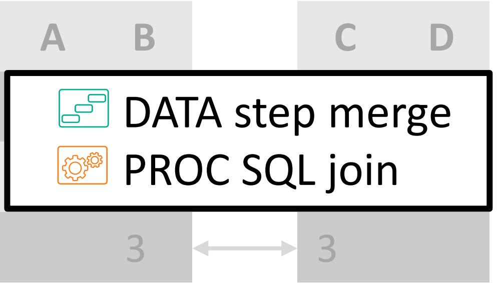


A	B
	1
	2
	4

C	D
2	
3	
4	

Table Relationships

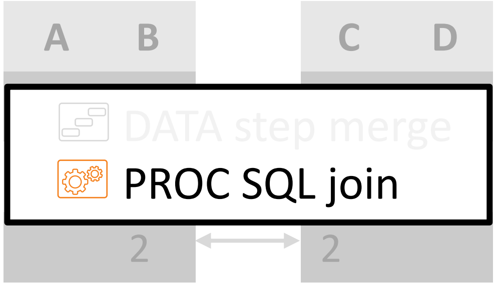
One-to-One



DATA step merge

PROC SQL join

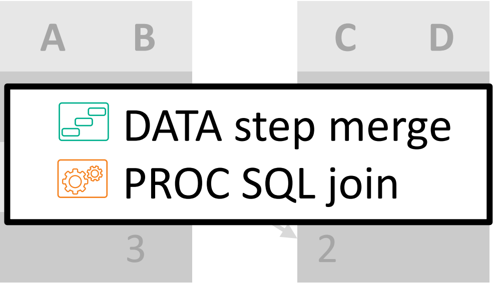
Many-to-Many



DATA step merge

PROC SQL join

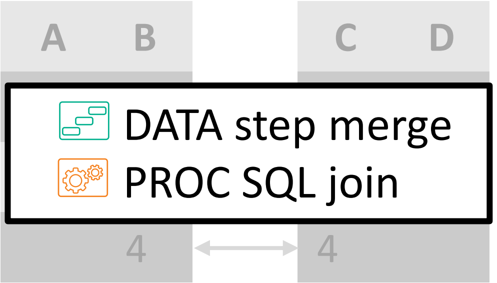
One-to-Many



DATA step merge

PROC SQL join

Nonmatches



DATA step merge

PROC SQL join

SQL Join Syntax

```
SELECT col-name, col-name
     FROM table1 INNER | LEFT| RIGHT | FULL JOIN> table2
     ON table1.column = table2.column;
```

Specify the join type in the FROM clause.

s103d01

SQL Join Syntax

SELECT *col-name, col-name*

FROM *table1* **INNER | LEFT| RIGHT | FULL JOIN>** *table2*

ON *table1.column* **=** *table2.column***;**

Specify the join criteria in the ON clause. You can use other comparison operators, such as the **greater than**, **less than**, or **special where** operators.

s103d01

SQL Join Syntax

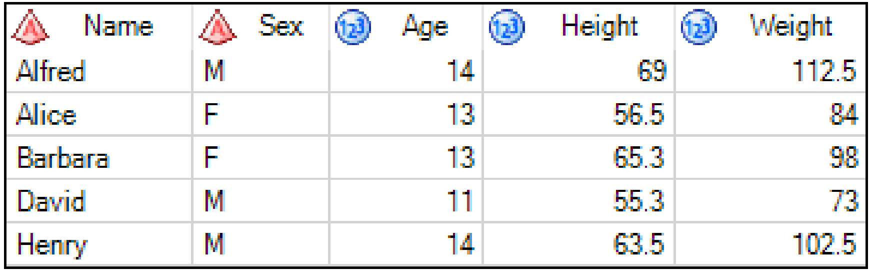
```
SELECT col-name, col-name
    FROM table1 INNER | LEFT| RIGHT | FULL JOIN> table2
    ON table1.column = table2.column;
```

Qualify the column names to specify the location of each column.

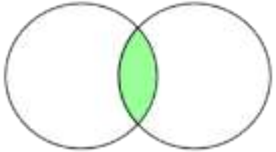
s103d01

```
proc sql;
create table class_combine as
select class_update.Name, Sex, Age, Height, Weight, Grade, Teacher
    from mylib.class_update inner join mylib.class_teachers
    on class_update.Name = class_teachers.Name;
quit;
```

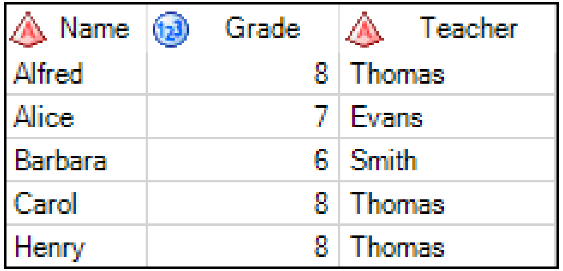
mylib.class_update



Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
David	M	11	55.3	73
Henry	M	14	63.5	102.5

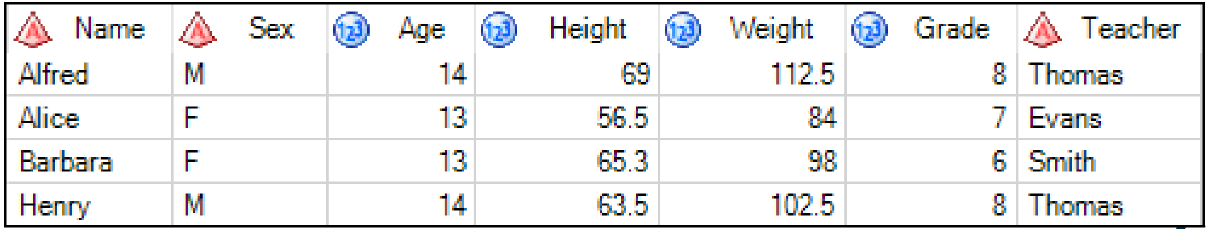


mylib.class_teachers



Name	Grade	Teacher
Alfred	8	Thomas
Alice	7	Evans
Barbara	6	Smith
Carol	8	Thomas
Henry	8	Thomas

class_combine



Name	Sex	Age	Height	Weight	Grade	Teacher
Alfred	M	14	69	112.5	8	Thomas
Alice	F	13	56.5	84	7	Evans
Barbara	F	13	65.3	98	6	Smith
Henry	M	14	63.5	102.5	8	Thomas

Only students in both input tables are included.

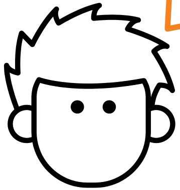
DATA Step Merge Syntax

list any number of input tables with one or more common columns

```
DATA output-table;
     MERGE input-table1 input-table2 ...;
     BY BY-column(s);
RUN;
```

list the common column or columns

The input tables must be sorted by the column(s) listed in the BY statement. This can be accomplished with a PROC SORT step.

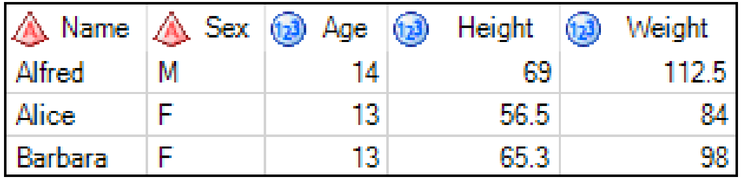


Merging Tables using the DATA Step

```
data class2;
    merge sashelp.class mylib.class_teachers;
    by Name;
run;
```

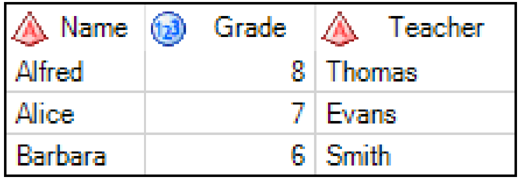
Columns are combined in the new table by matching values of **Name**.

sashelp.class



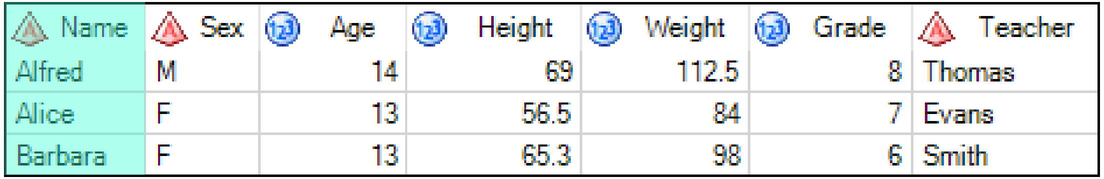
Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98

mylib.class_teachers



Name	Grade	Teacher
Alfred	8	Thomas
Alice	7	Evans
Barbara	6	Smith

class2

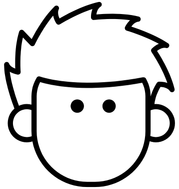


Name	Sex	Age	Height	Weight	Grade	Teacher
Alfred	M	14	69	112.5	8	Thomas
Alice	F	13	56.5	84	7	Evans
Barbara	F	13	65.3	98	6	Smith

Merging Tables with Nonmatching Rows

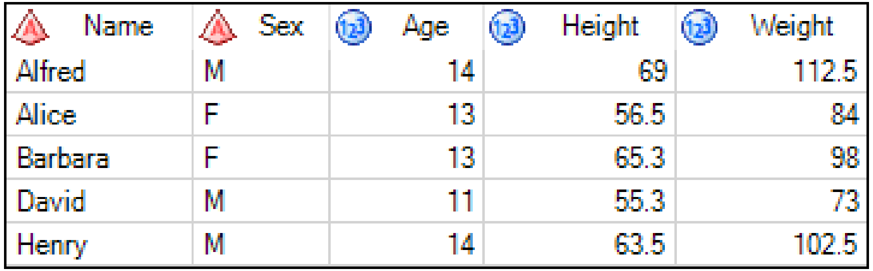
```
DATA output-table;
     MERGE input-table1(IN=variable)
           input-table2(IN=variable) ...;
     BY BY-column(s);
RUN;
```

The IN= data set option can be used to identify matching and nonmatching rows.

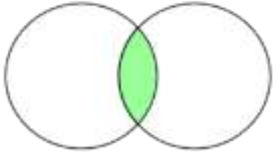


```
data class_combine;
    merge mylib.class_update(in=inUpdate)
          mylib.class_teachers(in=inTeachers);
    by Name;
   if inUpdate and inTeachers;
run;
```

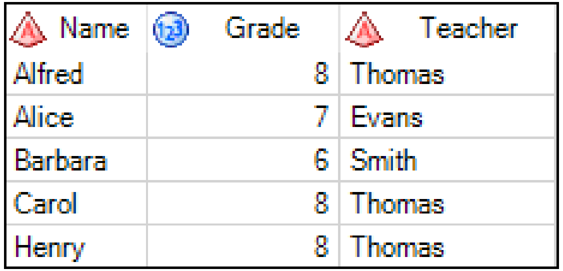
mylib.class_update



Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
David	M	11	55.3	73
Henry	M	14	63.5	102.5

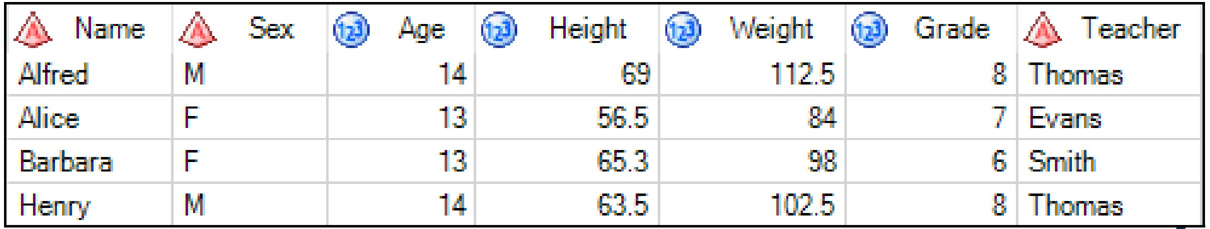


mylib.class_teachers



Name	Grade	Teacher
Alfred	8	Thomas
Alice	7	Evans
Barbara	6	Smith
Carol	8	Thomas
Henry	8	Thomas

class_combine



Name	Sex	Age	Height	Weight	Grade	Teacher
Alfred	M	14	69	112.5	8	Thomas
Alice	F	13	56.5	84	7	Evans
Barbara	F	13	65.3	98	6	Smith
Henry	M	14	63.5	102.5	8	Thomas

Only students in both input tables are included.

Demonstration

Considerations

DATA step merge

- requires sorted input data
- matching columns must have the same name
- manually set column attributes

PROC SQL join

- does not require sorted data
- matching columns do not need the same name
- use STIMER or FULLSTIMER to see timing
- SQL optimizer

Benefits

DATA step merge

- create multiple output tables in one step for matches and nonmatches
- control processing
- Data step debugger

PROC SQL join

- join multiple tables in one query
- create a Cartesian product for many-to-many joins
- Non-equijoins
- common DBMS syntax

Combining Data in SAS

I. Concatenation

II. Joins/Merges

III. Point-and-click Methods

Demonstration

SAS Programming 2: Data Manipulation Techniques

SAS® SQL 1: Essentials

Additional Resources

What if you want to ...

. . . use PROC SQL to join tables?

- Read the book *PROC SQL by Example*.

. . . compare the DATA step merge and the PROC SQL join?

- Read the blog post Life saver tip for comparing PROC SQL join with SAS data step merge.
- Read the paper MERGING vs. JOINING: Comparing the DATA Step with SQL.

. . . view examples of different methods for combining data?

- Read the book *Combining and Modifying SAS Data Sets: Examples*.
- Take the SAS Programming 3: Advanced Techniques and Efficiencies course.

Thank you for joining!

Q&A

Please submit your questions using the Q&A icon located in the menu at the bottom of your screen

Explore Helpful Resources

Ask the Expert
View other user webinars that provide insights into using SAS products to make your job easier.

FREE Training
Learn from home – free for 30 days. Get software labs to practice and online support if needed.

SAS Support Communities
Ask questions, get answers and share insights with SAS users.

SAS Analytics Explorers
An exclusive platform to collaborate, learn and share your expertise. Gain access to a diverse network to advance your career. Special rewards and recognition exclusively for SAS users.

SAS Users YouTube Channel
A plethora of videos on hundreds of topics, just for SAS users.

Newsletters
Get the latest SAS news plus tips, tricks and more.

Users Groups
Meet local SAS users, network and exchange ideas – virtually.

SAS Profile
If you haven't already done so, create your SAS Profile to access free training, SAS Support Communities, technical support, software downloads, newsletters and more.

Thank you

for joining us for this SAS webinar