

Chapter 7: Combining SAS Data Sets

<SAS Certification Prep Guide - Base Programming for SAS 9>

Objectives : learn to

- perform one-to-one merging of data sets
- concatenate data sets
- interleave data sets
- match-merge data sets

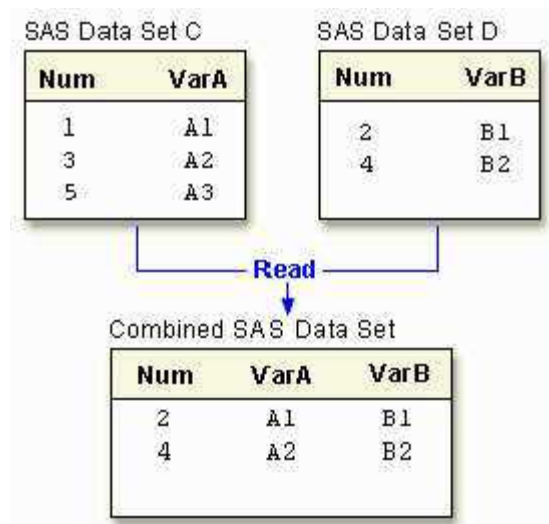
One-to-One merging

In one-to-one merging, you can read different data sets, or you can read the same data set more than once, as if you were reading from separate data sets.

■ How One-to-One Merging Selects Data

- the new data set contains all the variables from all the input data sets. If the data sets contain variables that have the same names, the values that are read in from the last data set overwrite the values that were read in from earlier data sets.
- the number of observations in the new data set is the number of observations in the smallest original data set. Observations are combined based on their relative position in each data set; that is, the first observation in one data set is joined with the first observation in the other, and so on. The DATA step stops after it has read the last observation from the smallest data set.

```
data one2one;  
    set c;  
    set d;  
run;
```



■ Example

```
data clinic.one2one;  
    set clinic.patients;  
    if age<60;  
    set clinic.measure;  
run;
```

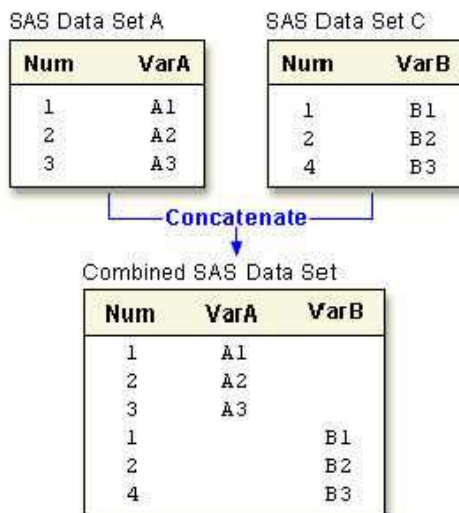
Concatenating

Another way to combine SAS data sets with the SET statement is concatenating, which appends the observations from one data set to another data set. To concatenate SAS data sets, you specify a list of data set names in the SET statement.

■ How Concatenating Selects Data

When a program concatenates data sets, all of the observations are read from the first data set that is listed in the SET statement. Then all of the observations are read from the second data set that is listed, and so on, until all of the listed data sets have been read. The new data set contains all of the variables and observations from all of the input data sets.

```
data concat;  
    set a b;  
run;
```



Notice that A and C contain a common variable named Num:

- Both instances of Num (or any common variable) must have the same type attribute, or SAS stops processing the DATA step and issues an error message stating that the variables are incompatible.
- However, if the length attribute is different, SAS takes the length from the first

data set that contains the variable. In this case, the length of Num in A determines the length of Num in Concat.

- The same is true for the label, format, and informat attributes: If any of these attributes are different, SAS takes the attribute from the first data set that contains the variable with that attribute.

■ Example

```
data clinic.concat;  
    set clinic.therapy1999 clinic.therapy2000;  
run;
```

Interleaving

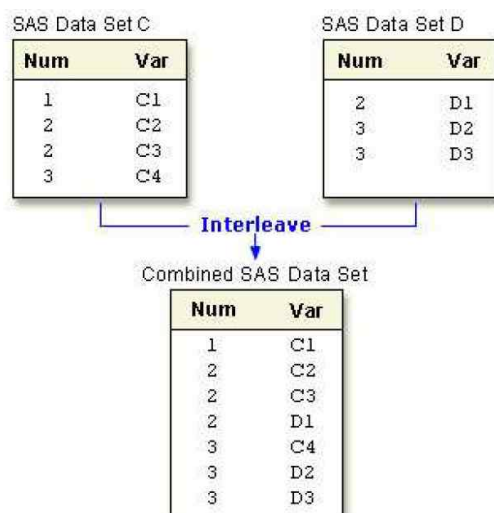
If you use a BY statement when you concatenate data sets, the result is interleaving. Interleaving intersperses observations from two or more data sets, based on one or more common variables.

To interleave SAS data sets, specify a list of data set names in the SET statement, and specify one or more BY variables in the BY statement.

■ How Interleaving Selects Data

When SAS interleaves data sets, observations in each BY group in each data set in the SET statement are read sequentially, in the order in which the data sets and BY variables are listed, until all observations have been processed. The new data set includes all the variables from all the input data sets, and it contains the total number of observations from all input data sets.

```
data interlv;  
    set c d;  
    by num;  
run;
```



■ Example

```
data clinic.interlv;  
    set clinic.therapy1999 clinic.therapy2000;  
    by mont;  
run;
```

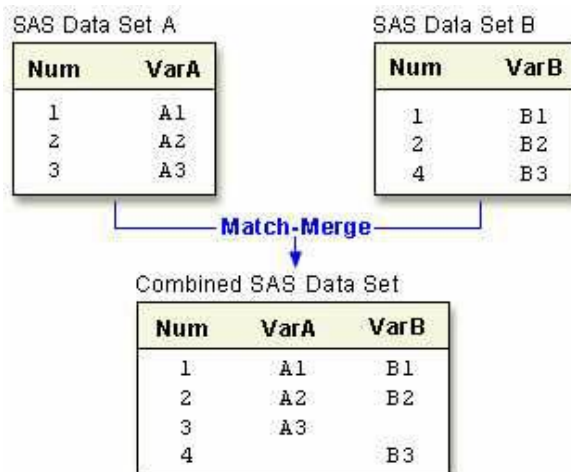
Simple Match-Merging

So far in this chapter, you've learned how to combine data sets based on the order of the observations in the input data sets. But sometimes you need to combine observations from two or more data sets into a single observation in a new data set according to the values of a common variable. This is called match-merging.

■ How Match-Merging Selects Data

Generally speaking, during match-merging SAS sequentially checks each observation of each data set to see whether the BY values match, then writes the combined observation to the new data set.

```
data merged;  
    merge a b;  
    by num;  
run;
```



observations in all input data sets. (You can add statements and options to select only observations that match for two or more specific input data sets.)

If an input data set doesn't have any observations for a particular value of the same-named variable, then the observation in the output data set contains missing values for the variables that are unique to that input data set.

■ Example: Merging Data Sets That Are Sorted in Ascending Order

```
proc sort data=clinic.demog;
    by id;
run;
proc print data=clinic.demog;
run;
proc sort data=clinic.visit;
    by id;
run;
proc print data=clinic.visit;
run;
data clinic.merged;
    merge clinic.demog clinic.visit;
    by id;
run;
proc print data=clinic.merged;
run;
```

Renaming Variables

Sometimes you might have same-named variables in more than one input data set. In this case, DATA step match-merging overwrites values of the like-named variable in the first data set in which it appears with values of the like-named variable in subsequent data sets.

For example, Clinic.Demog contains the variable Date (date of birth), and Clinic.Visit also contains Date (date of the clinic visit in 1998). The DATA step below overwrites the date of birth with the date of the clinic visit.

```
data clinic.merged;
    merge clinic.demog clinic.visit;
    by id;
run;
```

To prevent overwriting, you can rename variables by using the RENAME= data set option in the MERGE statement.

In the following example, the RENAME= option renames the variable Date in Clinic.Demog to BirthDate, and it renames the variable Date in Clinic.Visit to VisitDate.

```
data clinic.merged;
    merge clinic.demog(rename=(date=BirthDate))
          clinic.visit(rename=(date=VisitDate));
    by id;
run;
proc print data=clinic.merged;
run;
```

sas-code

```
data clinic.one2one;
    set clinic.patients;
    if age<60;
    set clinic.measure;
run;

data clinic.concat;
    set clinic.therapy1999 clinic.therapy2000;
run;

data clinic.interlv;
    set clinic.therapy1999 clinic.therapy2000;
    by month;
run;

proc sort data=clinic.demog;
    by id;
run;
proc print data=clinic.demog;
run;
proc sort data=clinic.visit;
    by id;
run;
proc print data=clinic.visit;
run;
data clinic.merged;
    merge clinic.demog clinic.visit;
    by id;
run;
proc print data=clinic.merged;
run;

data clinic.merged;
    merge clinic.demog(rename=(date=BirthDate))
          clinic.visit(rename=(date=VisitDate));
    by id;
run;
proc print data=clinic.merged;
run;
```