PharmaSUG2011 - Paper TU05

A Many to Many Merge, Without SQL?

David Franklin, TheProgrammersCabin.com, Litchfield, NH

ABSTRACT

As a SAS® Programmer, one of our common tasks is to merge data from two or more datasets. Most merges are 1-to-1 or 1-to-many, i.e. there is at least one dataset with a sequence of variables that create a unique record identifier. But what about the case where there is not a unique record in both cases, known as a many-to-many merge?

Typically the advice is to use PROC SQL, however this paper looks at a way (and the code is presented) that a SAS datastep can be used to achieve the same result, with the advantage that you as a programmer have more control over the way SAS processes it.

The technique introduced here will probably not change your reliance on PROC SQL to do a many-to-many merge, but it will add a valuable “utility” to your toolkit.

INTRODUCTION

Merging two datasets is a common task carried out by a SAS Programmer. The common format of the data when it is merged is either a 1-to-1 or 1-to-many. A many-to-many merge is usually done using a PROC SQL call. But what if we could merge the data using a datastep instead of an SQL? The aim of this paper is to show exactly this.

OUR DATA

Lets look at some data first -- this is in the form of SAS card statements:

```
*----------------------------------------------------*;
* Lets load some data;
*----------------------------------------------------*;
data ae;  ** Adverse Event Data;
   infile cards;
   input ptnum $ 1-3 @5 date date9. event $ 15-35;
   format date date9.;
cards;
001 16NOV2009 Nausea
002 16NOV2009 Heartburn
002 16NOV2009 Acid Indigestion
002 18NOV2009 Nausea
003 17NOV2009 Fever
003 18NOV2009 Fever
005 17NOV2009 Fever
;
run;

data cm;  ** Concomitant Medication Data;
   infile cards;
   input ptnum $ 1-3 @5 date date9. medication $ 15-35;
   format date date9.;
cards;
001 16NOV2009 Dopamine
002 16NOV2009 Antacid
002 16NOV2009 Sodium bicarbonate
002 18NOV2009 Dopamine
003 18NOV2009 Asprin
004 19NOV2009 Asprin
005 17NOV2009 Asprin
;
run;
```

The merge here will be done here by PTNUM (subject number) and DATE.

Some notes on the raw data to merge:

- Subjects 001 and 005 have a one-to-one match.

- Subject 002 has a many to many match on 16NOV2009, but a one to one match on 18NOV2009.
- Subject 003 has no match for 17NOV2009 but does have a match for 18NOV2009.
- Subject 004 has no match.

NOW LETS START MERGING

Lets jump ahead to what we are looking for when we merge the two datasets -- note that we are doing an inner join, i.e. where there is no match the observation is excluded:

```
Obs    ptnum         date    event               medication

 1      001     16NOV2009    Nausea              Dopamine
 2      002     16NOV2009    Heartburn           Antacid
 3      002     16NOV2009    Heartburn           Sodium bicarbonate
 4      002     16NOV2009    Acid Indigestion    Antacid
 5      002     16NOV2009    Acid Indigestion    Sodium bicarbonate
 6      002     18NOV2009    Nausea              Dopamine
 7      003     18NOV2009    Fever               Asprin
 8      005     17NOV2009    Fever               Asprin
```

When first looking at merging the two datasets, the first thought may be to use a simple MERGE statement as shown below:

```
** SAS Code;
*-----------------------------------------------------*;
* This will not work!;
*-----------------------------------------------------*;
data all0;
   merge ae cm;
   by ptnum date;
run;
title1 "Merge using the MERGE statement -- this fails";
proc print data=all0;
run;

** SAS Output;
Merge using the MERGE statement -- this fails

Obs    ptnum         date    event               medication

 1      001     16NOV2009    Nausea              Dopamine
 2      002     16NOV2009    Heartburn           Antacid
 3      002     16NOV2009    Acid Indigestion    Sodium bicarbonate
 4      002     18NOV2009    Nausea              Dopamine
 5      003     17NOV2009    Fever
 6      003     18NOV2009    Fever               Asprin
 7      004     19NOV2009                        Asprin
 8      005     17NOV2009    Fever               Asprin
```

As can be seen, this clearly does not work, so how can a many-to-many merge be done successfully?

THE USUAL WAY, PROC SQL

The most common way that this match is done is with an SQL call, as the following code demonstrates:

```
** SAS Code;
*-----------------------------------------------------*;
* SQL, the most common way that it is seen done;
*-----------------------------------------------------*;
proc sql;
   create table all0 as
      select a.*, b.medication
      from ae a inner join cm b
      on a.ptnum=b.ptnum and
         a.date=b.date;
   quit;
run;
title1 "Merge using SQL -- most common way this is seen done";
proc print data=all0;
run;

** SAS Output;
Merge using SQL -- most common way this is seen done
```

```
Obs    ptnum         date    event               medication

 1      001     16NOV2009    Nausea              Dopamine
 2      002     16NOV2009    Heartburn           Antacid
 3      002     16NOV2009    Heartburn           Sodium bicarbonate
 4      002     16NOV2009    Acid Indigestion    Antacid
 5      002     16NOV2009    Acid Indigestion    Sodium bicarbonate
 6      002     18NOV2009    Nausea              Dopamine
 7      003     18NOV2009    Fever               Asprin
 8      005     17NOV2009    Fever               Asprin
```

This works, but it is not what we want -- we do not want to use SQL in the solution.

THE DATASTEP, GOING LOOPY!

The datastep with POINT option with the SET statement is not often seen but it gives the most control. Key to this is that the datastep that takes each observation in AE and then tries to match this with each observation in CM -- this is basically a loop within a loop!

```
** SAS Code;
data all1;
   set ae;
   drop _:;  ** Drop temporary variables;
   match=0;  ** Match flag;

   ** Our loop within a loop -- output if match;
   do i=1 to xnobs;
      ** Need to rename the "merging" variables within the CM
         dataset;
      set cm (rename=(ptnum=_ptnum date=_date)) nobs=xnobs point=i;
         ** Have to rename matching variables so that they do not overwrite
            the original values in AE;
      if ptnum=_ptnum and date=_date then do;
         match=1;  ** Yes, there is a match my the "merging" variables;
         output;
      end;
   end;

run;
title1 "Merge using using the POINT option in a SET statement";
proc print data=all1;
run;

** SAS Output;
Merge using using the POINT option in a SET statement

Obs    ptnum         date    event               match    medication

 1      001     16NOV2009    Nausea                1      Dopamine
 2      002     16NOV2009    Heartburn             1      Antacid
 3      002     16NOV2009    Heartburn             1      Sodium bicarbonate
 4      002     16NOV2009    Acid Indigestion      1      Antacid
 5      002     16NOV2009    Acid Indigestion      1      Sodium bicarbonate
 6      002     18NOV2009    Nausea                1      Dopamine
 7      003     18NOV2009    Fever                 1      Asprin
 8      005     17NOV2009    Fever                 1      Asprin
```

This is a lot of code. An important note here is that I have had to rename the matching variables in CM so that they do not overwrite the original values in AE (very important) but use the DROP statement to get rid of these when the dataset ALL1 is created. Note also that the variable MATCH is used so that it is easy to see where a match is made but is not necessary.

CONCLUSION

A many-to-many merge without using SQL was acomplished.

There are a couple of other ways to do this merge without SQL, notably a variation on the "Loading Unique Dataset into an Array" which was presented at PharmaSUG 2010 entitled “Countdown of the Top 10 Ways to Merge Data”. However, the technique shown above is a shorter method with similar control (however the use of the array will do a many-to-many merge with the equivalent SQL OUTER JOIN).

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

David Franklin
TheProgrammersCabin.com
16 Roberts Road
Litchfield, NH 03052
Cell: 603-275-6809
Email: dfranklin@theprogrammerscabin.com
Web: http://www.TheProgrammersCabin.com