

Many-to-Many Matching Using the Inner Join

Tom Winn
State Auditor's Office,
State of Texas

Every SAS programmer knows how to combine observations from multiple data sets by matching the corresponding observations according to the values of a BY-variable which is common to each of the data sets, and which previously was used to sort each of the data sets. This is a match-merge situation, and the MERGE statement works every time . . . Right?

Well, no. For some reason, MERGE does not work the way you might think that it should for many-to-many merges involving large data files. In such cases, the customary MERGE step may fail to return all of the pertinent matches. The solution is to replace the MERGE-logic with an inner-join and a DATA step. The inner join is used to return all matching rows from the Cartesian product of two tables, and a DATA step is used to eliminate all of the matches which don't "belong" together, according to certain criteria.

Here is an example: A human resources management system (which happens to exist in a DB2 relational database environment) contains a table which includes leave-accounting transactions for employees, as well as a different table which includes historical data regarding employee job assignments, including the employee's assigned work-group. Each work-group is identified by a variable, which is called an ORGCODE. A programmer decides to use the job assignment history table to identify the beginning and ending dates for job assignments in each ORGCODE for each employee. Then, these data are combined with the leave-accounting data, by match-merging them according to the values of the employee-number, and by selecting only those corresponding observations for which the employee's leave start-date is contained in an interval of ORGCODE beginning- and ending-dates for the employee. Doesn't this seem like a reasonable approach?

Here is the SAS code which the programmer used:

```
*** COMBINE LEAVE ACCOUNTING AND EMPLOYEE ASSIGNMENT DATA ***;
DATA COMBO;
  MERGE LVTKN(IN=A) ASGNHIST;
  BY EMPLOYEE;
  IF A;
  *** IDENTIFY CORRECT ORGCODE ACCORDING TO LEAVE *****;
  *** STARTDT (MATCH RECORDS BY ENSURING STARTDT IS ***;
  *** IN-RANGE, BETWEEN BEGINNING- AND ENDING-DATES ***;
  *** FOR THE EMPLOYEE IN THE ORGCODE).*****;
  IF ORGBEG <= STARTDT <= ORGEND;
  DROP ORGBEG ORGEND;
```

Unfortunately, the preceding lines of SAS code failed to return all of the pertinent matches.

Here is the SAS code which achieved the desired result:

```
*** COMBINE LEAVE ACCOUNTING AND EMPLOYEE ASSIGNMENT DATA**;
*** NOTE -- THIS IS MANY-TO-MANY MATCHING, NOT ONE-TO- *****;
***          ONE MATCHING. (THE INNER JOIN RETURNS ALL *****;
***          MATCHING ROWS FROM THE CARTESIAN PRODUCT)*****;
PROC SQL;
  CREATE TABLE COMBO AS
  SELECT COALESCE(LVTKN.EMPLOYEE,ASGNHIST.EMPLOYEE)AS EMPLOYEE,
         RPTNAME, STARTDT, LVTYPE, LVACTYCD, LVTIME, ENDDT,
         ORGCODE, ORGBEG,  ORGEND
  FROM LVTKN, ASGNHIST
  WHERE LVTKN.EMPLOYEE=ASGNHIST.EMPLOYEE
  ORDER BY EMPLOYEE;

*** GET RID OF MATCHES WHICH DON'T BELONG TOGETHER *****;
DATA COMBO;
  SET COMBO;
  BY EMPLOYEE;
  *** IDENTIFY CORRECT ORGCODE ACCORDING TO LEAVE *****;
  *** STARTDT (MATCH RECORDS BY ENSURING STARTDT IS ***;
  *** IN-RANGE, BETWEEN BEGINNING- AND ENDING-DATES ***;
  *** FOR THE EMPLOYEE IN THE ORGCODE).*****;
  IF ORGBEG <= STARTDT <= ORGEND;
  DROP ORGBEG ORGEND;
```

The PROC SQL inner join returns all combinations of rows with matching key-values; that is, for each employee, all leave accounting transactions are combined with all of the ORGCODEs which the employee ever belonged to. The DATA step is used to eliminate those combinations for which the ORGCODE is incorrect. Although an employee may have worked in several ORGCODEs during his/her period of service with the employer, for each employee, there will be a beginning- and an ending-date associated with each job assignment. The DATA step selects only the ORGCODE for which the leave date is interior to the interval of dates during which the employee had job assignments in that particular ORGCODE.

The lesson to be learned from this example is: when you are combining data using many-to-many merges involving large data files, don't use the MERGE statement. Instead, utilize an inner join to return all matching rows from the Cartesian product of the tables to be combined, followed by a DATA step to eliminate any unwanted matches.

AUTHOR INFORMATION:

Thomas J. Winn Jr.
10906 Rustic Manor Lane
Austin, TX 78750

Telephone: (512) 250-8610
E-Mail: TWIN504@aol.com