

# PROC SQL

## How can I compare two datasets with similarly-named variables by joining them horizontally?

Rather than using PROC COMPARE to identify differences, which can sometimes generate 'too much' information, it is preferable to join datasets horizontally, then perform a manual comparison. The problem is that a MERGE will overlay same-named columns, therefore to perform the comparison variables must first be renamed.

This utility macro will identify all of the variables in a dataset make a copy of the dataset in a designated library and add a prefix to each variable.

```
%macro copy_and_preface( inlib  =
                        ,outlib =
                        ,dsn     =
                        ,pfx     =
                        ) ;

*** Cleanse parameters *** ;
%let inlib = %upcase(&inlib) ;
%let dsn   = %upcase(&dsn)   ;
%let pfx   = %lowercase(&pfx) ;

*** Build rename list *** ;
data _null_ ;
  length ren $ 2000 ;
  retain ren ;
  set sashelp.vcolumn end = eof ;
  where libname = "&inlib" and memname = "&dsn" ;
  ren = catt(ren, ' ' !! name, " = &pfx._", compress(name,'_')) ;
  if eof then call symputx('renop',ren) ;
run ;

*** Copy dataset *** ;
%if %sysfunc(exist(&outlib..&pfx._&dsn)) %then
%do ;
  %if %sysfunc(exist(&outlib..&pfx._&dsn._OLD)) %then
  %do ;
    %put WARNING: Previous archive dataset &outlib..&pfx._&dsn._OLD al
ready exists. Deleting... ;
    proc datasets lib = &outlib nolist ;
      delete &pfx._&dsn._OLD ;
    quit ;
  %end ;
  %put WARNING: Dataset &outlib..&pfx._&dsn already exists. Renaming a
s &outlib..&pfx._&dsn._OLD. ;
  proc datasets lib = &outlib nolist ;
    change &pfx._&dsn = &pfx._&dsn._OLD ;
  quit ;
%end ;
```

# PROC SQL

```
proc datasets lib = &inlib nolist ;
  copy outlib = &outlib ;
  select &dsn ;
quit ;

*** Reaname dataset & preface variables *** ;
proc datasets lib = &outlib nolist ;
  change &dsn = &pfx._&dsn ;
  modify &pfx._&dsn ;
  rename &renop ;
quit ;

%mend copy_and_preface ;

%copy_and_preface ( inlib  = sashelp
                   ,outlib = work
                   ,dsn    = class
                   ,pfx    = one
                   )
```

Call the macro on each of the required datasets, then join them horizontally, e.g.:

```
proc sql noprint ;
  create table compare as
  select  l.*
         ,r.*
  from one_class as l
       full outer join
       two_class as r
  on one_keyvar = two_keyvar
  ;
quit ;
```

Unique solution ID: #1029

Author: Alan D Rudland

Last update: 2017-06-07 10:30