PROC SQL

Is there a facility in PROC SQL to repeat a value down a BY-GROUP in a column, similar to the RETAIN Statement in DATA Step?

Sometimes when data is combined from different sources it is possible to end up with undesirable missing values. Run the following code against the attached CSV file to generate a dataset demonstrating this issue:

```
data miss_val ;
  infile 'miss_val.csv' dsd missover ;
  input name  :$30.
        email :$30.
        first :$15.
 last  :$15.
 age
 ;
run ;
```

The dataset looks like:

name	email	first	last	age
John Doe	john_doe@acme.com	John		.
John Doe	john_doe@acme.com		Doe	.
John Doe	john_doe@acme.com	John		18
John Doe	john_doe@acme.com	John	Doe	.
Joe Smith	joe_smith@acme.com	Joe	Smith	.
Joe Smith	joe_smith@acme.com	Joe	Smith	.
Becky Smith	becky_smith@acme.com	Becky	Smith	.
Becky Smith	becky_smith@acme.com		Smith	34
Charlie Thomas	charlie_thomas@acme.com	Charlie	Thomas	9
Charlie Thomas	charlie_thomas@acme.com	Charlie	Thomas	.
Charlie Thomas	charlie_thomas@acme.com		Thomas	9

As can be seen there are missing values of the first, last and age variables in each name BY-GROUP. The objective is to populate the missing fields, based on the non-missing fields for the same BY-GROUP. The RETAIN statement could be used in conjunction with FIRST. and LAST. variables for each of the columns in turn, re-initializing the value only as the BY-GROUP changes.

PROC SQL re-merging

Alternatively the re-merge facility within PROC SQL can produce the same result in a more succint form. The re-merge facility causes a summary statistic (calculated down a column) to be re-merged back on to the detail records, with a message to that effect written to the LOG. e.g. consider the dataset:

cvar	nvar
A	100
B	500
C	300

and the code:

```
proc sql noprint ;
  create table two as
  select  cvar
         ,nvar
         ,mean(nvar) as avg
  from one
  ;
quit
```

will generate the output:

cvar	nvar	avg
A	100	300
B	500	300
C	300	300

It might be assumed that the 'summary statistic' column can only be a numeric variable, however certain of the Statistical Functions can be used with character variables, and it is this feature which allows us to generate our desired outcome.

PROC SQL

Use the code:

```
proc sql noprint ;
  create table no_miss as
  select  name                /* GROUPing variable */
         ,email               /* Un-Summarized data to ensure data lin
es are not collapsed to a single entry */
         ,max(first) as first /* Summarized variable - re-
merged back against detail rows */
  ,max(last)  as last  /* Summarized variable - re-
merged back against detail rows */
  ,max(age)   as age   /* Summarized variable - re-
merged back against detail rows */
  from miss_val
  group by name
  ;
quit ;
```

In this example, the name column is the BY-GROUP variable, the email variable is neither in the BY-GROUP, nor is it a summarized column, and therefore causes the re-merge feature to operate. The MAX function returns the largest non-missing value, and because it lists only a single variable, the search is performed DOWN the column (ANSI Standard default action) according to the GROUP BY clause. The returned value is then re-merged back against the detail records for the corresponding BY-GROUP in a new computed column which shares a name with the original incomplete column.

Unique solution ID: #1037
Author: Alan D Rudland
Last update: 2017-09-27 17:15