PROC SQL

Can I use the FIRSTOBS= and OBS= options in PROC SQL?

Yes, but there are two PROC SQL options which you might also want to consider.

The FIRSTOBS= and OBS= options are normally used on a SET statement to specify the first observation to be processed and the last observation to be processed respectively - the total number of observations being (obs - firstobs) + 1. If the FIRSTOBS= option has not been specified, the default is 1.

To use these options in PROC SQL, add them to any data source listed on the FROM clause:

```
proc sql ;
  create table class_sql as
  select *
  from sashelp.class (firstobs = 3 obs = 8)
  ;
quit ;
```

which generates the following in the LOG:

```
NOTE: Table WORK.CLASS_SQL created, with 6 rows and 5 columns.
```

As an alternative to using these options, consider using the PROC SQL options

INOBS= to determine the number of rows to be processed.

OUTOBS= to determine the number of rows to be included in the report / output dataset.

For example, the code:

```
proc sql inobs = 8 ;
  create table class_sql_2 as
  select *
  from sashelp.class
  ;
quit ;
```

generates the following in the LOG:

```
WARNING: Only 8 records were read from SASHELP.CLASS due to INOBS= opt
ion.
NOTE: Table WORK.CLASS_SQL_2 created, with 8 rows and 5 columns.
```

It is important to understand the timing for the selection, e.g. the code:

```
proc sql outobs = 6 ;
  create table class_sql_3 as
  select *
  from sashelp.class
  where sex = 'F'
  ;
quit ;

proc sql inobs = 8 ;
  create table class_sql_4 as
  select *
  from sashelp.class
  where sex = 'M'
  ;
quit ;
```

generates the following in the LOG:

```
WARNING: Statement terminated early due to OUTOBS=6 option.
NOTE: Table WORK.CLASS_SQL_2 created, with 6 rows and 5 columns.

WARNING: Only 8 records were read from SASHELP.CLASS due to INOBS= opt
ion.
NOTE: Table WORK.CLASS_SQL_4 created, with 8 rows and 5 columns.
```

i.e. the selection of rows for the output dataset using either the INOBS= or OUTOBS= options is made ***after*** the WHERE clause has selected rows on an internal, virtual table.

Unique solution ID: #1025
Author: Alan D Rudland
Last update: 2017-05-09 14:39