SAS® GLOBAL FORUM 2016

IMAGINE. CREATE. INNOVATE.

Virtual Accessing of a SAS® Data Set Using OPEN, FETCH, and CLOSE Functions with %SYSFUNC and %DO Loops

Amarnath Vijayarangan
Emmes Services Pvt Ltd, India

#SASGF

Virtual Accessing of a SAS® Data Set Using OPEN, FETCH, and CLOSE Functions with %SYSFUNC and %DO Loops

Amarnath Vijayarangan

Emmes Services Pvt Ltd, India

Abstract

- In SAS®, there are, at a minimum three approaches to achieve any intended task and each approaches has its own pros and cons.
- Identifying and using of efficient SAS programming techniques are recommended and mandatory for the larger data sets.
- This paper describes the efficiency of virtual access and various situations to virtual access of the data sets using OPEN, FETCH and CLOSE functions with %SYSFUNC and %DO LOOPS.

SASHELP.SNACKS

	QtySold	Price	Advertised	Holiday	Date	Product
1	0	1.99	0	0	01JAN2002	Baked potato chips
2	0	1.99	0	0	02JAN2002	Baked potato chips
3	0	1.99	0	0	03JAN2002	Baked potato chips
4	0	1.99	0	0	04JAN2002	Baked potato chips
5	0	1.99	0	0	05JAN2002	Baked potato chips
6	0	1.99	0	0	06JAN2002	Baked potato chips
7	0	1.99	0	0	07JAN2002	Baked potato chips

Situation 1. Data Set Partition

```
%macro datasplit;
data
  %let dsid=%sysfunc(open(product));
  %do %while(not %sysfunc(fetch(&dsid)));
    snacks.pid%sysfunc(getvarn(&dsid, %sysfunc(varnum(&dsid, id))))
  %end;
  %let dsid=%sysfunc(close(&dsid));;
  set sashelp.snacks;
  %let dsid=%sysfunc(open(product));
  %let i=0;
  %do %while(not %sysfunc(fetch(&dsid)));
   %let i = %eval(&i + 1);
   %if &i=1 %then %let elsest=;
   %else %let elsest=else;
  &elsest if product = "%sysfunc(getvarc(&dsid, %sysfunc(varnum(&dsid, product))))"
  then output snacks.pid%sysfunc(getvarn(&dsid, %sysfunc(varnum(&dsid, id))));
  %end;
  %let dsid=%sysfunc(close(&dsid));
run;
%mend;
```

	Product	id
	Baked potato chips	1
	Barbeque pork rinds	2
	Barbeque potato chips	3

```
%datasplit
```

```
MPRINT(DATASPLIT):   data snacks.pid1 snacks.pid2 snacks.pid3 snacks.pid4 snacks.pid5 snacks.pid6
snacks.pid7 snacks.pid8 snacks.pid9 snacks.pid10 snacks.pid11 snacks.pid12 snacks.pid13
snacks.pid14 snacks.pid15 snacks.pid16 snacks.pid17 snacks.pid18 snacks.pid19 snacks.pid20
snacks.pid21 snacks.pid22 snacks.pid23 snacks.pid24 snacks.pid25 snacks.pid26 snacks.pid27
snacks.pid28 snacks.pid29 snacks.pid30 snacks.pid31 snacks.pid32 snacks.pid33 snacks.pid34
snacks.pid35 ;
MPRINT(DATASPLIT):   set sashelp.snacks;
MPRINT(DATASPLIT):   if product = "Baked potato chips                    " then output
snacks.pid1;
MPRINT(DATASPLIT):   else if product = "Barbeque pork rinds              " then output
snacks.pid2;
MPRINT(DATASPLIT):   else if product = "Barbeque potato chips            " then output
```

Situation 2. Variable Ordering

```
%macro fetchobs1(dsn=);
  %let dsid=%sysfunc(open(&dsn));
  %do %while(not %sysfunc(fetch(&dsid)));
    price%sysfunc(getvarn(&dsid, %sysfunc(varnum(&dsid, pid))))
    adv%sysfunc(getvarn(&dsid, %sysfunc(varnum(&dsid, pid))))
    qty%sysfunc(getvarn(&dsid, %sysfunc(varnum(&dsid, pid))))
  %end;
  %let dsid=%sysfunc(close(&dsid));
%mend fetchobs1;
data mdsn;
    length date %fetchobs1(dsn=ProductList) 8;
    set mdsn;
run;
```

Date of sale	price1	adv1	qty1	price2	adv2	qty2	price3
01JAN2002	1.99	0	0	1.49	0	3	1.49
02JAN2002	1.99	0	0	1.49	0	11	1.49
03JAN2002	1.99	0	0	1.49	0	1	1.49
04JAN2002	1.99	0	0	1.49	0	1	1.49
05JAN2002	1.99	0	0	1.49	0	13	1.49
06JAN2002	1.99	0	0	1.49	0	1	1.49

Situation 3. Dynamic Variable List

```
%macro fetchobs2(dsn=,prefix=);
  %let dsid=%sysfunc(open(&dsn));
  %do %while(not %sysfunc(fetch(&dsid)));
    &prefix.%sysfunc(getvarn(&dsid, %sysfunc(varnum(&dsid, pid))))
  %end;
  %let dsid=%sysfunc(close(&dsid));
%mend fetchobs2;
proc reg data=mdsn plots=none;
  model qty1= %fetchobs2(dsn=Product,prefix=price)
              %fetchobs2(dsn=Product,prefix=adv);
quit;
```

```
85187
85188
85189   proc reg data=mdsn plots=none;
85190     model qty1= %fetchobs2(dsn=Product,prefix=price)
MPRINT(FETCHOBS2):   price1 price2 price3 price4 price5 price6 price7 price8 price9 price10
price11 price12 price13 price14 price15 price16 price17 price18 price19 price20 price21 price22
price23 price24 price25 price26 price27 price28 price29 price30 price31 price32 price33 price34
price35
85191                %fetchobs2(dsn=Product,prefix=adv);
MPRINT(FETCHOBS2):   adv1 adv2 adv3 adv4 adv5 adv6 adv7 adv8 adv9 adv10 adv11 adv12 adv13 adv14
adv15 adv16 adv17 adv18 adv19 adv20 adv21 adv22 adv23 adv24 adv25 adv26 adv27 adv28 adv29 adv30
adv31 adv32 adv33 adv34 adv35
85192   quit;
```

Situation 4. Array Process

```
data mdsn;
    set mdsn;
    array price(*) %fetchobs2(dsn=Product,prefix=price);
    array qty(*) %fetchobs2(dsn=Product,prefix=qty);
    array dsale(*) %fetchobs2(dsn=Product,prefix=dsale);
    do i=1 to dim(qty);
       if price(i) and qty(i) then dsale(i)=price(i)*qty(i);
    end;
    drop i;
run;
```

```
85193
85194   data mdsn;
85195     set mdsn;
85196     array price(*) %fetchobs2(dsn=Product,prefix=price);
MPRINT(FETCHOBS2):   price1 price2 price3 price4 price5 price6 price7 price8 price9 price10
price11 price12 price13 price14 price15 price16 price17 price18 price19 price20 price21 price22
price23 price24 price25 price26 price27 price28 price29 price30 price31 price32 price33 price34
price35
85197     array qty(*) %fetchobs2(dsn=Product,prefix=qty);
MPRINT(FETCHOBS2):   qty1 qty2 qty3 qty4 qty5 qty6 qty7 qty8 qty9 qty10 qty11 qty12 qty13 qty14
qty15 qty16 qty17 qty18 qty19 qty20 qty21 qty22 qty23 qty24 qty25 qty26 qty27 qty28 qty29 qty30
qty31 qty32 qty33 qty34 qty35
85198     array dsale(*) %fetchobs2(dsn=Product,prefix=dsale);
MPRINT(FETCHOBS2):   dsale1 dsale2 dsale3 dsale4 dsale5 dsale6 dsale7 dsale8 dsale9 dsale10
dsale11 dsale12 dsale13 dsale14 dsale15 dsale16 dsale17 dsale18 dsale19 dsale20 dsale21 dsale22
```

Virtual Accessing of a SAS® Data Set Using OPEN, FETCH, and CLOSE Functions with %SYSFUNC and %DO Loops

Amarnath Vijayarangan

Emmes Services Pvt Ltd, India

Situation 5. Correlation Matrix Report

```
*** Perform correlation analysis;
proc corr data=mdsn out=Corr(where=(_TYPE_='CORR'));
    var price:;
run;

*** Report;
%macro corr_rpt;
ods tagsets.excelxp  file="CorrelationMatrix.xls"  style=statistical;
proc print data = corr2  label noobs ;
    label %let dsid=%sysfunc(open(product));
        %do %while(not %sysfunc(fetch(&dsid)));
        price%sysfunc(getvarn(&dsid, %sysfunc(varnum(&dsid, id))))=
        %nrbquote(%sysfunc(getvarc(&dsid, %sysfunc(varnum(&dsid, product)))))
        %end;;
        %let dsid=%sysfunc(close(&dsid));;
    var product price:;
run ;
ods tagsets.excelxp close;
%mend corr_rpt;
%corr_rpt;
```

Product name	Baked potato chips	Barbeque pork rinds	Barbeque potato chips	Bread sticks	Buttery popcorn
Baked potato chips	1	-0.14938	-0.11403	0.16121	-0.22581
Barbeque pork rinds	-0.14938	1	0.29435	-0.01687	0.63369
Barbeque potato chips	-0.11403	0.29435	1	-0.18389	0.79684
Bread sticks	0.16121	-0.01687	-0.18389	1	-0.36409
Buttery popcorn	-0.22581	0.63369	0.79684	-0.36409	1

Usual Approach

- Count the number of products
- Create separate macro variable for each product to pass the product id and product name
  - Too many macro variables are created
- Create a macro variable to store product list and then scan it for the product name
  - May run into a macro variable length issue on a larger data set.
  - May be the product list can split into several macro variables, need more variables to be created

Conclusion

- While processing data sets especially larger data sets, speed and time are of essence.
- Parallel processing, dynamic variable creation, variable list creation and ordering the variables using a reference key data set helps
  - To reduce the execution time
  - Amount of coding and
  - Also avoids possible errors that can creep in while handling the larger data sets with numerous variables.
- Various situations were described in which the combination of OPEN, FETCH, and CLOSE Functions with %SYSFUNC and %DO Loops becomes handy and more efficient.

SAS® GLOBAL FORUM 2016

IMAGINE. CREATE. INNOVATE.

LAS VEGAS | APRIL 18-21

#SASGF