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
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.

Situation 1. Data Set Partition

```sas
%macro datasplit;
  data;
  set sashelp.snacks;
  if product = "Baked potato chips" then output snacks.pid= fundkeyfunc( getvname(idd), product )
     snacks.pid= fundkeyfunc( getvname(idd), fundkeyfunc( getvname(idd), idd ) )
  ;
  run;
%mend;
```

Situation 2. Variable Ordering

```sas
%macro fetchchips(den, prefix=);%let datasplit=datasplit(open(idd));
%do while( not keyfunc( fetch(idd) ) ) ;
%let prices=fundkeyfunc( getvname(idd), keyfuncfunc( getvname(idd), idd ) )
%end;
%let datasplit=datasplit(close(idd));
%mend fetchchips;
```

Situation 3. Dynamic Variable List

```sas
proc reg data=mdsn plots=noplet;
  model qty1= fetchchips(den=product, prefix=price);
  %let fetchchips=datalines product=price
  %end;
```

Situation 4. Array Process

```sas
%macro fetchchips(den, prefix=);
%let datasplit=datasplit(open(idd));
%do while( not keyfunc( fetch(idd) ) ) ;
%let prices=fundkeyfunc( getvname(idd), keyfuncfunc( getvname(idd), idd ) )
%end;
%let datasplit=datasplit(close(idd));
%mend fetchchips;
```

**SASHELP:SNACKS**

<table>
<thead>
<tr>
<th>QtySold</th>
<th>Price</th>
<th>Advertised</th>
<th>Holiday</th>
<th>Date</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.99</td>
<td>0</td>
<td>01JAN2002</td>
<td>0</td>
<td>Baked potato chips</td>
</tr>
<tr>
<td>2</td>
<td>0.99</td>
<td>0</td>
<td>03JAN2002</td>
<td>0</td>
<td>Baked potato chips</td>
</tr>
<tr>
<td>3</td>
<td>0.99</td>
<td>0</td>
<td>03JAN2002</td>
<td>0</td>
<td>Baked potato chips</td>
</tr>
<tr>
<td>4</td>
<td>0.99</td>
<td>0</td>
<td>04JAN2002</td>
<td>0</td>
<td>Baked potato chips</td>
</tr>
<tr>
<td>5</td>
<td>0.99</td>
<td>0</td>
<td>05JAN2002</td>
<td>0</td>
<td>Baked potato chips</td>
</tr>
<tr>
<td>6</td>
<td>0.99</td>
<td>0</td>
<td>06JAN2002</td>
<td>0</td>
<td>Baked potato chips</td>
</tr>
<tr>
<td>7</td>
<td>0.99</td>
<td>0</td>
<td>07JAN2002</td>
<td>0</td>
<td>Baked potato chips</td>
</tr>
</tbody>
</table>
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

### Process Flow

**Situation 5. Correlation Matrix Report**

**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.

### 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

### 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

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

**Situation 5. Correlation Matrix Report**

**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.

### 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