Paper 200-25

Utilization of SAS® programs in the Business Environment

Sunil K. Gupta, Gupta Programming, Simi Valley, CA
Charles E. Shipp, Consultant, San Pedro, CA

ABSTRACT

Business can take advantage of all the features of the SAS System. By taking a task-oriented approach to address business issues, SAS programs can be written to support the functional requirements of business operations. The objective of many companies is to establish a data warehouse structure that incorporates total customer involvement with standards and documentation. The four main areas that will be reviewed include data access, data management, data analysis and data presentation.

Critical business decisions will benefit from these structured approaches.

DATA ACCESS

The first requirement for all businesses to manage information is to access the data. Typically in large companies, data files are stored and updated in a variety of formats. SAS procedures can access these files directly and also create a SAS view or a SAS data set based on these external files. Examples of these external files include Informix and Oracle. In addition, SAS programs can be written to read the contents of a Microsoft Excel file. For applications that collect data, a SAS Data Entry System can be created.

Question 1 - What is an effective procedure used to access external files for SAS data set creation?

Answer 1 - The Proc SQL procedure.

The SAS code below shows how a few statements of the Proc SQL can be used to create a SAS data set from an Informix database. While no subsetting is performed in this example, the user has great flexibility to subset and organize the data to fit the needs of the organization.

SAS Code

```
Proc SQL;
connect to informix as sasinf
(db = "//server_main/production");

create table sales as
select * from connect to sasinf
(select * from sales);

disconnect from sasinf
QUIT;
```

Question 2 - What procedure can be used to customize the input of an Excel file?

Answer 2 - The DATA STEP.

The SAS code below shows how the power of a data null step can be used to read data from an external file. A custom data null step can be written to locate and identify key text in the Excel file for conditional processing. The new SAS version 8 facilitates this process by use of the ODS feature. Note that SAS/Access to PC file format will be required to read from Excel files. Customers may maintain Excel files that can be processed and analyzed by SAS.

SAS Code

```
DATA EXCEL;
tab='09'x;
infile 'c:\customer1\product.xls' firstobs=2 dlm=tab dsd lrecl=6
missover;
input subject 1-2 sex $4.;
if sex = 'M';
RUN;
```

SAS Code Version 8

```
Proc IMPORT datafile = 'c:\customer1\product.xls'
 out=work.EXCEL
 DBMS = EXCEL97
 replace;
 getname = yes;
RUN;
```

Question 3 - What are some techniques available to check for data integrity?

Answer 3 - SAS/AF using SCL

Screen Control Language (SCL) allows you to add any logic and field validation for data entry. For example, if you have test dates that are only valid between a start date and an end date, your validation code could be as follows.

SAS Code

```
if (test_dt lt start_dt or test_dt ge end_dt) then do;
  error on test_dt;
  msg='Test date is invalid...please re-enter';
  end;
```

Give additional consideration to the new methods that use SAS internet tools. You can have field validation for your company intranets and internet web sites.

DATA MANAGEMENT

Once data is available as a SAS data set, business can establish a data warehouse structure to manage all the SAS data sets. While the SAS System is not a true relational database management system, business can construct rules for accessing files and for combining files. This structure not only makes programming more easy and straightforward, but it also makes the task of data set maintenance a minimum effort.

For collecting time sensitive data, a sophisticated mechanism can be developed to make data set name and variable names contain a time specific suffix. By the files having a unique identifier to relate all transactions, information can be combined for complete analysis. In addition, macros can be written to automate the listing of the most recent weeks to be analyzed.

Question 4 - What is an effective method to construct data sets for entity relationship?

Answer 4 - The use of primary key variables in a one-to-one and a one-to-many relationship data sets.

The list below shows a data set and sample variables of a distribution center keeping track of product sales. Having the data structure in this organization, there is no duplication of values or records. As more detailed information is required for any given product, time period or customer, more data sets can be extracted from that data.

Data set Monthly Sales
w_28JAN

Variables
cust, product, type, w07jan00, w14jan00, w21jan00, w28jan00

SAS Code to identify the most recent Friday and past 3 weeks date.

```
Data _null_;
 length charv scharv $7;

 * go back past 7 days to find Friday date;
do I = 1 to 7;

* decrease today's date by 1 week and upto 7 days;
 fridt = ("&sysdate"d - I -7);

* process if new date is Friday;
 if weekday(fridt) = 6 then do;

 * save date;
 wk_dsd=fridt;

 * save past 3 weeks date;
 swk_dsd = fridt - 2*(7);

 end;
end;

* save dates as char values;
charv = put(wk_dsd, date7.);
scharv=put(swk_dsd, date7.);

* define data set name;
dddw_ds = 'W_' || substr(charv, 1, 5);

* define variables;
s_wk = 'w' || scharv;
e_wk = 'w' || charv;

* save as macro variables;
call symput('dddw_ds', dddw_ds);
call symput('s_wk', s_wk);
call symput('e_wk', e_wk);
run;

Proc PRINT data=&dddw_ds (obs=10) label;
 var &e_wk &s_wk;
run;
```

The SAS printa macro below from page 266 of the Guide to SAS® Macro Processing Book is very helpful to print all SAS data sets in a SAS Data Library.

SAS Code

```
Proc CONTENTS DATA=COMMON._ALL_;
Proc PRINT DATA=COMMON.DEMO (obs=20) label;
RUN;

%macro printa(libname, worklib=work);
%local num I;
Proc DATASETS library=&libname memtype=data;
contents out=&worklib..temp1(keep=memname) data=_all_
noprint;
run;
Data _null_;
set &worklib..temp1 end=final;
by memname notsorted;
if last.memname;
n+1;
if final then
call symput('num', put(n, 8.));
call symput('v'||left(put(n, 8.)), trim(memname));
run;
%do I=1 to &num;
Proc PRINT data=&libname..&&v&i;
title "Data Set &libname..&&v&i";
run;
%end;
%mend printa;
```

Once a data warehouse structure is established, keeping current and reliable information becomes very significant maintenance factor. Typically, business will need to assure duplicate records do not exist in the system, identify and correct missing values, and automate the archive of outdated data.

Question 5 - How do you check for duplicate records?

Answer 5 - Use Proc SQL.

The SAS code below shows how to check for duplicate records. Once identified, they can be deleted. Ideally, there should be checks for duplicate records throughout the data entry process.

SAS Code

```
* Identify duplicate records;
Proc SQL;
create table duptmp as
select orig, std, count(*) as freq
from int.adr
group by orig, std
having freq ge 2;
QUIT;

* Delete duplicate records;
Proc SQL;
create table unique as
select distinct * from int.adr;
QUIT;
```

Question 6 - How can you locate missing values?

Answer 6 -

The SAS code below shows an effective method to locate any missing values. This is specially helpful when business merges many files together into a single file. As a result of the data set merge, missing values could result due to missing values or variables in one of the data sets. This provides a quick check for missing variables.

SAS Code

```
Data All;
 merge w_31DEC w_28JAN w_25FEB;
 by cust product type;
run;

Proc TABULATE data=all format=6.2;
 class cust product type;
 var w31dec99 w28jan00 w25feb00;
 table cust*product*type, w31dec99 w28jan00 w25feb00 /
missprint;
RUN;
```

Question 7 - How can you archive data sets of the same name?

Answer 7 - Use Proc DATASETS.

The SAS code below shows an automated mechanism to archive outdated data. By using the age statement in the Proc datasets procedure, SAS automatically renames the data set to the pervious version data set and retires the oldest data set. SAS version 8 now has a generation feature to manage multiple versions of a data set.

SAS Code

```
Proc DATASETS LIBRARY=IN98;
 age today today1 - today3/memtype = data;
RUN;
```

SAS Code Version 8

This program creates upto 3 generations of the IN98.today data set. With each new data or procedure step, the IN98.today is renamed to the next sequence number. Thus the proc sort will rename the original IN98.today to IN98.today#001. In the next data step, the IN98.today#001 gets renamed to IN98.today#002.

```
DATA IN98.today (genmax=3);
 set IN98.past;
run;

Proc SORT data=IN98.today;
 by date;
run;

DATA IN98.today;
 set IN98.today IN98.past;
run;
```

DATA ANALYSIS

For data analysis, business often needs to identify and monitor the performance of its best products and customers. Reports can be generated to list the top products and top customers. Example 8 shows a method to identify companies' top 10 customers during a specific time period. Example 9 shows how to transfer descriptive statistics to a SAS data set for summary reports or analysis.

Question 8 - What procedure ranks records based on a numeric variable?

Answer 8 - The Proc RANK procedure.

The SAS code below sorts the records based on numeric variable w28jan00.

SAS Code

```
Proc RANK DATA=w_28JAN OUT=RANKA TIES=LOW;
 by cust;
 var w28jan00;
 ranks rnk
RUN;

Data ranka;
 set ranka;
 if rnk <= 10;
run;
```

Question 9 - What procedure can be used to save the descriptive statistics to a data set?

Answer 9 - One procedure that generates descriptive statistics and saves the results to a data set is the MEANS procedure.

The SAS code below saves the count, mean and standard deviation to the S_28JAN data set.

SAS Code

```
Proc MEANS data=w_28jan n mean stdv;
 var w28jan00;
 output out=s_28JAN n=count mean=meanv stdv=stdv;
run;
```

DATA PRESENTATION

Once the quality of the data has been assured and the analysis is complete, business can then focus on the presentation of information to upper management. SAS provides many ways to create custom layout reports using the Data Null Step as shown in example 9. In addition, for generation of rich text format output to be read by Microsoft word, SAS provides the necessary codes required for RTF files. SAS Version 8 automates this facility with the ODS System. Based on what the customer requests, this could be an iterative process.

Question 10 - How can you create a customized report?

Answer 10 - Use Data Null step.

The SAS code below on page 154 in the SAS Applications Guide shows an example of how to format an output.

SAS Code

```
Data _NULL_;
 retain month;
 set payroll end=eof;
 by dept;
 if _n_=1 then
 month=scan(put(today(), worddate18.), 1);
 file print header=h linesleft=ll notitles;
 if first.dept then do;
  nettot=0;
  grosstot=0;
  if ll<14 then put _page_;
 end;
 put @7 dept 3. @19 name $15. @32 number 5. @42 sex $1.
@48 netpay 8.2 @63 grosspay 9.2;

 nettot+netpay;
 grosstot+grosspay;
 if last.dept then do;
  put @46 '________' @62 '_____________'
        / 'Department Total' @44 nettot dollar12.2
       @60 grosstot dollar12.2 //;
  grandnet + nettot;
  grandgro+grosstot;
 end;
 if eof then put 'Overall Total' @43 grandnet dollar13.2 @59
grandgro dollar13.2;
return;

h: put // @22 'ABC Manufacturing Compnay'
         / @23 'Payroll R eport for ' Month
        /// @19 'Employee' @31 'Employee' @50 'Net' @65
'Gross'
         / @3 'Department' @21 'Name' @32 'Number' @41
'Sex' @50 'Pay' @66 'Pay' //;
return;
RUN;
```

Question 11 - How can I use the Output Delivery System of SAS Version 8 for quality output?

Answer 11 - The Output Delivery System can be used for quality presentation graphics or for web site page tables.

Refer to The Little SAS Book or to Painless Windows for details to the following example:

SAS Code Version 8

```
ods listing close;
ods html file = 'path-to-html-destination';

Data ABC;
  .... defining SAS statements ... ;
RUN;

Proc PRINT;
run;

ods html close;
```

SUMMARY

By understanding the power of the SAS programming language and the latest technology available from SAS Institute, business can utilize SAS software to establish an effective information delivery system. Not only will the functional requirements of the business operation be met, but total customer involvement with standards and documentation can be incorp orated.

TRADEMARK INFORMATION

SAS® Guide to Macro Processing, version 6, second edition
SAS® Applications Guide, 1987 Edition.

ABOUT THE AUTHORS

Sunil K. Gupta, Gupta Programming
SAS Institute Quality Partner™
http://www.GuptaProgramming.com
213 Goldenwood Circle, Simi Valley, CA 93065
Phone: (805)-577-8877
E-mail: Sunil@GuptaProgramming.com

Sunil is a senior consultant at Gupta Programming. His consulting projects with pharmaceutical companies include the development of a Macro-Based Application for Report Generation. He has been using SAS® software for over 9 years and is a SAS Institute Quality Partner™.

Charlie Shipp, Charles E. Shipp, Consultant
1650 Barrywood, San Pedro, CA 90731-1254
Phone: (310) 833-9186
E-mail: charlieshipp@compuserve.com

Charlie Shipp has been a SAS programmer for more than 18 years. His areas of expertise include Base/SAS, SAS/GRAPH®, SAS/AF®, SAS/FSP®, SAS/STAT® and web design.