**HW 14 Macro SQL Interface**

1. **Creating Macro Variables Using SQL**
   1. Copy the program shown below into the Editor window. Submit the program and review the results.

**%let start=01Jan2007;**

**%let stop=31Dec2007;**

**proc means data=orion.order\_fact noprint;**

**var Total\_Retail\_Price;**

**output out=stats n=count mean=avg;**

**run;**

**data \_null\_;**

**set stats;**

**call symputx('orders',count);**

**call symputx('average',avg);**

**run;**

**proc gchart data=orion.order\_fact;**

**vbar3d Order\_Type**

**/ patternid=midpoint cframe=w shape=c discrete**

**sumvar=Total\_Retail\_Price type=mean ref=&average;**

**format Total\_Retail\_Price dollar4.;**

**label Total\_Retail\_Price='Average Order';**

**title1 h=1 "Report from &start to &stop";**

**title2 h=1 f=swiss "Orders this period: " c=b "&orders";**

**footnote1 h=1 f=swiss "Overall Average: " c=b**

**"%sysfunc(putn(&average,dollar4.))";**

**run;**

**quit;**

* 1. Replace the PROC MEANS step and the DATA step with a PROC SQL step. Create the macro variables ORDERS and AVERAGE based on the variable **Total\_Retail\_Price** in the **orion.order\_fact** data set.
  2. Remove any extraneous blanks from the values. Use the %PUT statement to display the values.
  3. Submit the PROC GHART step and verify that the output is the same.
  4. Modify the PROC SQL step to create a third macro variable named FMTAVG that contains the formatted version of AVERAGE. Remove any extraneous blanks from the value.
  5. In the FOOTNOTE statement, replace **%sysfunc(putn(&average, dollar4.))** with   
     the macro variable FMTAVG.

1. **Creating a List of Values in a Macro Variable Using SQL**
   1. Copy the program below into the Editor window and modify the SQL procedure to create a macro variable named TOP3 that contains the customer ID numbers of the top three customers by **Total\_Retail\_Price** in the **orion.order\_fact** data set. Separate each of the values with a comma and a blank. Use the OUTOBS= option to limit the number of output rows.

**proc sql;**

**select customer\_id**

**from orion.order\_fact**

**group by Customer\_ID**

**order by total descending;**

**quit;**

**Remember:** A GROUP BY clause is used to summarize the data by customer ID number, and an ORDER BY clause is used to sort the data.

* 1. Submit the program and review the results.

1. **Creating Multiple Macro Variables Using SQL**
   1. The **orion.customer\_type** data set contains the variable **Customer\_Type\_ID**, which holds the unique customer type codes. Use the SQL procedure to create a series of macro variables named CTYPE1 through CTYPE*xx*, where *xx* resolves to the number of rows that the query will return.

You will need two queries, one to return the number of rows that the query will return and the other to create CTYPE1 through CTYPE*xx.* Store the number of rows in a macrovariable called nrows. When resolving this macrovariable in the second query, you will need to use the %trim function to remove extraneous blanks.

* 1. Open the program shown below to display only the macro variables that begin with CTYPE.

**proc sql;**

**select name, value**

**from dictionary.macros**

**where name like "CTYPE%";**

**quit;**