STSCI 5060 HW4

Due on 12/7/2018

This homework is composed of classroom/lab practices. When coding, you should mark different components appropriately and leave at least one blank line between different code sections. Your code should start with the following comments:

```
/* Fall 2018 STSCI 5060 HW4           */
/* Student Name: First_Name Last_Name */
/* NetID:                             */
/* Section:                           */
```

For each question, you should start something like below to mark the beginning of a question and every output. If the comments and/or titles are missing, up to 10 points will be taken.

```
title "Question 1";
title2 "Output 1";
```

Important:

- Only use the data files that I posted on the course website for this homework. Do not use the built-in SAS data sets since these may have been modified already.
- Create a libref called **orasas** to store all your files.

You should wait after all your code is working and run it to produce an HTML output at once for your submission. You are required to submit the following to the course website, which should be compressed into one single file, e.g., HW4_LastName_FirstName.7z.

- your SAS code (HW4_LastName_FirstName.sas),
- your HTML output (HW4_HTML_LastName_FirstName.mht),
- your SAS log file (HW4_SAS_log_LastName_FirstName.txt) and,
- the screenshot from Question 5 (in PDF or MS Word format).

1. Use the LIBNAME statement to create a libref called **myora** to associate with the Oracle database you created before. In order for you to have the same tables as others for this homework, you are required to run the **CREATEPVFC.SQL** script at the beginning; refer to Lab 2 if needed. From the Customer_t table find all available information of the customers who live in New York and Pennsylvania; sort your result first by state and then by name. Then you update the Customer_t table by adding a new customer with the following data: customerid=100, customername= Cool Furture, customeraddress=123 Lake Side Drive, customercity=Ithaca, customerstate=NY, and customerpostalcode=14850. Display the whole table to confirm the update, and sort your result by customerid in descending order. Disassociate the libref you created.
2. Use the SQL Pass-through facility to connect to the Oracle database you created earlier by assigning an alias called "**myconn**." Use the EXECUTE…BY statement to create a table called **PTF_t** in the orasas library, which is the inner join of the **customer_t** and **order_t**, containing the following columns: customerid, customername, customerstate and customerpostalcode. Disconnect from the Oracle database. Display the new PTF_t table in the orasas library (Hint: consider creating PTF_t in Oracle first and then copying it over to SAS).

3. You will be using the Oracle command line interface, Oracle SQLDEVELOPER, and SAS 9.4 to do the following practice (Questions 3-6). For how to save your files in the Oracle command line interface, refer to early lab session instructions. In your own Oracle account, create and populate a large DBMS table called **Rawmaterials_t** by running the script "**Mariterials.sql**" available on the course website (you are required to add a SAS comment at the beginning of this SAS code showing the SQLPlus commend for running this script). Check the number of rows inserted into the Rawmaterials_t table by running query containing the COUNT() function. Do this within SAS through the SAS LIBNAME statement method. You just report the number not the table.
4. In SAS 9.4, create a SAS data set, **Rawmaterial**, in the orasas library containing the same contents as the Rawmaterials_t table in your Oracle database using the LIBNAME statement and a PROC SQL procedure. Then use a PROC PRINT procedure to display your result. You should only report the first 10 rows of the dataset.
5. From within SAS create an Oracle table called "**cheapoak_t**" which is saved in the same Oracle account by using a PROC SQL procedure through querying the Rawmaterial dataset in the orasas library. The cheapoak_t table contains all the Oak materials whose standard prices are less than $500. Display cheapoak_t in SQLDEVELOPER and submit **a SQLDEVELOPER screenshot**. You are required to create a new connection named **myora** in the Connections panel and just show the first 10 rows.
6. Use in-database processing and PROC SORT to sort Rawmaterials_t by materialstandardprice then by materialid and output your result to a SAS dataset in the orasas library. Was the in-database processing actually used by SAS when PROC SORT was executing? Use PROC PRINT to display the table contents. You should only report the first 10 rows of the dataset.