STSCI 5060 HW3

Due at 11:59 PM, 11/28/2018

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

```
/* Fall 2018 STSCI 5060 HW3           */
/* 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 by other users already.
- Create a libref called HW4 to store all your files.

You should wait after all your code is working and run it to produce an HTML output at once. You are required to submit your SAS code (HW4_LastName_FirstName.sas) and your HTML output (HW4_HTML_LastName_FirstName.html) to the course website, which should be compressed into one single file, HW4_LastName_FirstName.7z.

1. Create a report that displays the employee identification number of current Level III and Level IV sales staff hired in 2004, who made at least one sale by the end of 2005. The **Order_fact** table contains information on all sales, and the **Sales** table contains information about current sales employees, including job titles and hire dates. (10 points)

2. Think about how you can use three columns to display the employee numbers, job codes, and salaries of all mechanics working for an airline respectively. The mechanic job has three levels and there is a separate table containing data for the mechanics at each level: ***Mechanicslevel1*, *Mechanicslevel2*, and *Mechanicslevel3*.** These tables all contain the same three columns. Write your PROC SQL code to realize it. (10 points)

3. Code the following business situation: You want to display vertically the following summarized data for members of a frequent-flyer program: total points earned, total points used, and total miles traveled. All three values can be calculated from columns in the table **Frequentflyers** by using summary functions. How about if you just want to display these results horizontally? (15 points)

4. Create a simple index on the Employee_ID column of the **Employee_addresses** table and a composite index on city, state, and country columns of the **Employee_addresses** table. Write a query to find out the employees (including all the columns) who are in Miami-Dade in

Florida by using "`City='Miami-Dade' and state='FL'`" in the WHERE clause; you should only output the **first 10 rows**. Find out if any index was used in your query; include this information as a footnote in the output (hint: use the MSGLEVEL=I system option to show if an index is used and you find out this index usage info from the SAS log). (20 points)

5. You have the following business scenario: Tom is a sales manager (Manager_ID=120102) who frequently needs access to personnel information for his direct reports, including name, job title, salary, and years of service. The data Tom needs can be obtained from these tables: Employee_Addresses, Employee_Payroll, and Employee_Organization. You are required to create a view, Tom_V, containing personnel information for Tom's direct reports to provide the information that Tom needs while avoiding inadvertent access to data for employees who do not report to him. In your view, use an alias "Name" and format=$25. for Employee_Name, and an alias "Title" and format=$15. for Job_Title; use an label "Annual Salary" and format=comma10.2 for Salary; define a column called "YOS" to hold the values of the years of service you calculate (hint: you use the today() function for those who are working; for those who were terminated earlier, use the column Employee_Term_Date), and label it as "Years of Service". Use a query to display the contents of the view you just created, and sort the values by Title and then by YOS, both in descending order. Now, use Tom_V to produce simple descriptive statistics: minimum, mean and maximum salaries of different job titles. You are required to use two methods to achieve the same result as shown on the lecture slide (Slide 11 of file L2-7-preclass-posting): the PROC MEANS procedure and the PROC SQL procedure. (hint: a PROC SQL view can be used as a SAS data set) (45 points)