**Homework, Proc SQL Additional Features**

**All tables are in the sql subdirectory.**

1. **Using PROC SQL Options and Displaying the Contents of a Dictionary Table**
	1. Write a query that retrieves **Memname** (table name) and **Memlabel** (description of the table) from **dictionary.Dictionaries**. Include only a single row per table name displayed.
	Title the report **Dictionary Tables**.
	2. Use the columns **memname**, **type**, and **length** from **dictionary.Columns** where **libname="ORION"** and the table contains a column (**Name**) named **Customer\_ID** (**upcase(Name)='CUSTOMER\_ID'**). Title the report **Tables containing Customer\_ID**.
2. **Using PROC SQL Options and Displaying Dictionary Table Information**
	1. Produce a report that includes **memname** (table name), **memlabel** (table description), and a count of the number of columns in each table, by querying **dictionary.Dictionaries**. Display only one row per table name. Title the report **Dictionary Tables**.
	2. List the table name (**memname**), number of rows (**nobs**), number of columns (**nvar**), file size (**filesize**), length of the widest column (**maxvar**), and length of the widest column label (**maxlabel**) by querying **dictionary.Tables**. Limit the list to tables in the **orion** library (**libname="ORION"**) and exclude views (**memtype ne 'VIEW'**) from your report. Order the report by table name. Give the columns appropriate labels. Name the report **Orion Library Tables**.
3. **Using PROC SQL Options, SAS System Options, and Dictionary Tables to Document Data Tables in the Orion Star Library**
	1. Use PROC SQL options, SAS system options, and **dictionary.Tables** to document the data tables in the **orion** library. In the report, exclude views and output only one row per table. For each numeric column, the query should place an asterisk beside each of the maximum values.

Hint: Consider using a CASE statement to generate the asterisks in the report.

Partial Output:



1. **Creating and Using Macro Variables**
	1. Write a query for the **Employee\_payroll** table that returns the highest value of **Salary** (**max(Salary)**). Title the report appropriately.
	2. Use %LET statements to create and assign values to two macro variables:
	**DataSet = Employee\_payroll**
	**VariableName = Salary**

Use a %PUT statement to write their values back to the SAS log in this context:

* 1. Modify the query you wrote in step **4.a**, and replace each hardcoded (typed) reference to **Employee\_payroll** (once in the title and once in the query) with a reference to the macro variable **DataSet** (**&DataSet**), and each hardcoded (typed) reference to **Salary** (once in the title and once in the query) with a reference to the macro variable **VariableName** (**&VariableName**). Resubmit the query, and verify that the results are identical to the results obtained in step **a.** above.
	2. Use the %LET statements to change the values of your macro variables:

**DataSet = Price\_List**
**VariableName = Unit\_Sales\_Price**

1. **Creating a Macro Variable from an SQL Query**
	1. Produce a report of **Country** and a new column named **Purchases** (**SUM (Total\_Retail\_Price)**). Group the report by **Country**. Include only orders placed in 2007. Order the report so that the highest values of **Purchases** sums are at the top. The data is in the following tables (columns of interest in parenthesis):
* **Order\_fact** (**Customer\_ID**, **Total\_Retail\_Price**)
* **Customer** (**Customer\_ID**, **Country**)

Name the report **2007 Purchases by Country**. Label the columns as indicated in the sample report:

* 1. Write a query similar to the first but modified to produce a report of **Purchases** by **Customer\_Name** for the year 2007. (Keep a copy of your first query for use in step **5.c**.) Subset the query so that only customers from the top-buying country (the one listed at the top of the last report; in this case, **US**) are included. Order the report so that customers with the highest purchases are at the top. The data is in the following tables:
* **Order\_fact** (**Customer\_ID**, **Total\_Retail\_Price**)
* **Customer** (**Customer\_ID**, **Customer\_Name**, **Country**)

Name the report **2007 US Customer Purchases** with a second title line of **Total US Purchases: $10,655.97** as indicated by the previous query. When you run the report, it should produce the following results:

* 1. Automate the report that you wrote in step **5.b**.
		1. Modify the query from step **5.a.** so that instead of producing output, it merely writes the values for **Country** and **Purchases** for the first returned row into macro variables called **Country** and **Country\_Purchases**, respectively.
		2. Modify the query that you wrote in step **5.b.** by substituting the macro variable reference **&Country** for each instance where you typed the value **US** in the query, and the macro reference **&Country\_Purchases** in the second title line in place of the **$10,655.97** value that you typed previously. When you are finished, run the modified queries. The results should be exactly as produced in step **5.b**.

Hint: Do not forget to use double quotation marks around macro variable references! Review Syntax 1 for inserting values into macro variables, if necessary.

* 1. Starting with the code from step **5.c.**, modify the first query so that the country with the lowest total purchases is read into the macro variable instead of the highest. Then rerun both queries. The queries should produce the following results without further modification:

