Stevens Institute of Technology
Castle Point on Hudson
Hoboken, NJ 07030

CS561 – Programming Assignment 1

Due Dates: Sec. A-10/19/2016 (Wed.) & Sec. B-10/20/2016 (Thu.)

Objectives

- In this assignment, you will write programs to *evaluate* relatively simple report queries and produce the output, and also *express* the queries in SQL. The key point of the exercise is to observe a large gap between the complexity of expressing the type of such queries and that of evaluating them. Your mission (in addition to writing the programs and SQL queries) is to consider the reasons for the gap (between the *expression* and *evaluation* of such queries) and how to narrow it.

Description

"Simple Database Application Program #1" (`sdap1.pgc`)

- Generate 2 separate reports based on the following queries (one report for query #1 and another for query #2):
  1. For each *product*, compute the *maximum* and *minimum* sales quantities along with the *corresponding customers, dates* (i.e., customers who made those maximum and minimum purchases, and the dates when those maximum and minimum sales quantities were made) and the *states* in which the sale transactions took place. If there are >1 occurrences of the max or min, choose one – do not display all.

     For each *product*, also compute the *average* sales quantity.
  2. For each combination of *customer* and *product*, output the *maximum sales quantities for CT* and *minimum sales quantities for NY and NJ in 3 separate columns*. Like the first report, display the *corresponding dates* (i.e., dates of those maximum and minimum sales quantities). Furthermore, for CT, include only the sales that occurred between 2000 and 2005; for NY and NJ, include all sales.

For this assignment, you can use a simple data structure (e.g., an array) to maintain the list of "information" being computed/captured (we will discuss the type of information you will need to compute/capture and maintain internally for the report over the next couple of lectures).

The following is a sample output – quantities displayed are for illustration only (not the actual values).

```
PRODUCT   MAX_Q  CUSTOMER  DATE        ST  MIN_Q  CUSTOMER  DATE        ST  AVG_Q
========  =====  ========  ==========  ==  =====  ========  ==========  ==  =====
Pepsi      2893  Bloom     01/01/2006  NJ     12  Sam       09/25/2001  NY   1435
Milk        159  Sam       02/15/2002  NJ      1  Emily     03/23/2004  CT     56
Bread      3087  Emily     07/01/2005  NY      2  Helen     02/02/2001  NJ   1512
```

```
CUSTOMER  PRODUCT  CT_MAX  DATE        NY_MIN  DATE        NJ_MIN  DATE
========  =======  ======  ==========  ======  ==========  ======  ==========
Sam       Egg        1908  01/11/2001     234  07/24/2005       2  11/03/2008
Helen     Cookies     392  03/31/2002    2342  09/14/2000      11  07/23/2002
Bloom     Butter     7045  09/22/2003     923  03/10/2004       8  09/11/2006
```

Make sure that:

1. "**`select * from sales`**" is the ONLY SQL statement allowed in your program.
2. Character string data (e.g., customer name and product name) are left justified.
3. Numeric data (e.g., Maximum/minimum Sales Quantities) are right justified.
4. The Date fields are in the format of `MM/DD/YYYY` (i.e., 01/02/2002 instead of 1/2/2002).
5. The reports must be generated with a single scan of the 'sales' table.

Grading

- (80 pts.) Logic/Correctness
- (10 pts.) Programming Style (e.g., comments, indentation, use of functions, etc.)
- (10 pts.) SQL statements to generate the same two reports

NOTE: A program with compilation errors will earn no more than 50 points.

Sample Command Line

```
$ sdap1 [sales], where 'sales' is an optional argument for the table name.
```

Submission

Submit your source code (file) (with your name and CWID on it) on Canvas.

Please include a "README" file with detailed instructions on how to compile and run the code, especially if you are using a language other than C, C++ or Java.

In addition to the source code, submit **SQL queries** to generate the same output (10% of the grade) – you should use the SQL queries to check for the correctness of your program output.

Please remember the following points when you're working on your programming assignments:

1. Your program must compile and execute based on the instructions provided in the README file (i.e., if your programs contain special functions for other compilers and does not compile based on README, you WILL lose 50% of the grade for the assignment).
2. Programming style is 10% of the grade. Please make sure to provide comments for the program, functions, etc. as well as in-line comments as needed. Also, make sure to use meaningful names for your classes, variables, methods/functions, etc. Use proper indentation.
3. In the header comments for your program (i.e., at the beginning of your program), please provide:
    a. General instructions on how to execute your program (e.g., command line for the program and whatever arguments it requires). This can be a simple copy & paste of the README file, or you can provide a simplified bullet listing of the steps for compiling and executing the code.
    b. Justification of your choice of data structures for your program – e.g., if you're using a linked list to maintain whatever information necessary for your program, justify why it's a data structure of your choice, as opposed to, say, arrays. If you're using other more sophisticated data structures, please provide a brief description of the data structures and again justify as to why you chose the data structures for your program.
    c. A detailed description of the algorithm of your program, e.g., how you're computing and maintaining the aggregates (e.g., min, max, avg) for your query output. You can do this with a detailed pseudo code.
4. Remember the only SQL statement allowed in your program is the simple select statement, "select * from sales". Points will be deducted if you use any other SQL statements in your programs.
5. You are NOT allowed to read in the entire table ('sales') and store them in memory before processing the rows. Instead, you need to read each row (one row at a time), process it and discard it. 50% of the overall grade (50 points out of 100) will be deducted, if any of the rows, other than the current row, are saved in memory (e.g., in a simple variable or an array).

Most importantly, **make sure it's your own work**! If we determine that your program is a copy of someone else's, both you and that someone else will receive 0 for the assignment and possibly additional penalties for the course.

Stevens Institute of Technology
Castle Point on Hudson
Hoboken, NJ 07030

Student's Name:_____

Major Area	Item	Max	Deduct	Score	%	Total
Logic	Correct min( ) for NY	5				
	Correct min( ) for NJ	5				
	Correct max( ) for CT	5				
	Correct max( )	5				
	Correct min( )	5				
	Correct max & min DATES	10				
	Correct PRODS and STATES	10				
	Correct avg( )	10				
	Output Format	20				
	Single Scan (YES/NO)	25				
	Total	**100**			**80%**	
Style	Header Comment	30				
	Function Comment	10				
	Line Comment	10				
	Indentation	10				
	Line/Block Spacing	10				
	Meaningful Identifier Names	20				
	Other	10				
	Total	**100**			**10%**	
SQL	**Total**	**100**			**10%**	
Sub-Total		**100**				
Penalties	If compilation fails or 'sales' table is cached into memory (subtract 50); For using anything more than 'select * from sales' for programming (vs. for your SQL queries), 25 points will be deducted.					**- 50**
Total						