

# CS561 – Programming Assignment 1

**Due Date: 3/20/2017 (Mon.)**

## 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):
  - For each *customer* compute the minimum and maximum sales quantities along with the corresponding products (purchased), dates (i.e., dates of those minimum and maximum sales quantities) and the state in which the sale transaction took place. If there are >1 occurrences of the min or max, choose one – do not display all.  
For the same *customer*, compute the average sales quantity.
  - For each combination of *customer* and *product*, output the maximum sales quantities for NJ and minimum sales quantities for NY and CT in 3 separate columns. Like the first report, display the corresponding dates (i.e., dates of those maximum and minimum sales quantities). Furthermore, for NY and NJ, include only the sales that occurred earlier than 2009; for CT, include all sales.

For this assignment, you can use a simple data structure (e.g., an array) to maintain the list of "information" being captured (we will discuss the type of information you will need to 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). (NOTE – the following output must be generated with a single scan of the 'sales' table).

CUSTOMER	MIN_Q	MIN_PROD	MIN_DATE	ST	MAX_Q	MAX_PROD	MAX_DATE	ST	AVG_Q
=====	=====	=====	=====	==	=====	=====	=====	==	=====
Bloom	12	Pepsi	01/01/2006	NJ	2893	Apple	09/25/2001	NY	1435
Sam	1	Milk	02/15/2002	NJ	259	Banana	03/23/2004	CT	56
Emily	2	Bread	07/01/2005	NY	3087	Milk	02/02/2001	NJ	1512

CUSTOMER	PRODUCT	NJ_MAX	DATE	NY_MIN	DATE	CT_MIN	DATE
=====	=====	=====	=====	=====	=====	=====	=====
Sam	Egg	1908	01/11/2001	2	07/24/2005	2	11/03/2008
Helen	Cookies	392	03/31/2002	42	09/14/2000	11	07/23/2002
Bloom	Butter	7045	09/22/2003	23	03/10/2004	8	09/11/2006

Make sure that:

- "select \* from sales" is the ONLY SQL statement allowed in your program.
- Character string data (e.g., customer name and product name) are left justified.
- Numeric data (e.g., Maximum/minimum Sales Quantities) are right justified.
- The Date fields are in the format of MM/DD/YYYY (i.e., 01/02/2002 instead of 1/2/2002).

### **Grading**

- (50 pts.) Logic/Correctness
- (10 pts.) Programming Style (e.g., comments, indentation, use of functions, etc.)
- (40 pts.) SQL queries

**NOTE: A program with compilation errors will lose 30 points (out of 60).**

### **Sample Command Line**

`$ sdapl [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 – 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.

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.

Student's Name: \_\_\_\_\_

Major Area	Item	Max	Deduct	Score	%	Total
<i>Logic</i>	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				
	<b>Total</b>	<b>100</b>			<b>50%</b>	
<i>Style</i>	Header Comment	30				
	Function Comment	10				
	Line Comment	10				
	Indentation	10				
	Line/Block Spacing	10				
	Meaningful Identifier Names	20				
	Other	10				
	<b>Total</b>	<b>100</b>			<b>10%</b>	
<i>SQL</i>	<b>Total</b>	<b>100</b>			<b>40%</b>	
<i>Sub-Total</i>		<b>100</b>				
<i>Penalties</i>	If compilation fails or 'sales' table is cached into memory (subtract 30); For using anything more than 'select * from sales' for programming (vs. for your SQL queries), 15 points will be deducted.					<b>- 30</b>
<i>Total</i>						