

CS561 – Programming Assignment 2

Due Dates: 11/25/2015 (Wed.) for Sec. A & 11/26/2015 (Thu.) for Sec. B

Objectives:

- To become familiar with the concept of *database application programming* and *query processing for complex OLAP/BI queries*.

Description:

"Simple Database Application Program #2" (sdap2.cpp)

- Generate reports based on the following queries:
 - For each customer and product, compute (1) the customer's average sale of this product, (2) the average sale of the product for the other customers and (3) the customer's average sale of the other products.
 - For customer and product, show the average sales before and after each quarter (e.g., for Q2, show average sales of Q1 and Q3. For "before" Q1 and "after" Q4, display <NULL>). The "YEAR" attribute is not considered for this query – for example, both Q1 of 1997 and Q1 of 1998 are considered Q1 regardless of the year.
 - For customer and product, count for each quarter, how many sales of the previous and how many sales of the following quarter had quantities between that quarter's average sale and minimum sale. Again for this query, the "YEAR" attribute is not considered.

For this assignment, you can write either 3 separate programs, one for each of the 3 reports, or one program generating all of the 3 reports.

Again, the only SQL statement you're allowed to use for your program is:

```
select * from sales;
```

That is, no where clauses, **no aggregate functions** (e.g., avg, sum, count), etc.

And, you cannot store the 'sales' table in memory.

The following are sample report output (NOTE: the numbers shown below are not the actual aggregate values. You can write simple SQL queries to find the actual aggregate values).

Report #1:

CUSTOMER	PRODUCT	CUST_AVG	OTHER_CUST_AVG	OTHER_PROD_AVG
Helen	Bread	243	268	1493
Emily	Milk	1426	478	926
.

Report #2:

CUSTOMER	PRODUCT	QUARTER	BEFORE_AVG	AFTER_AVG
Bloom	Bread	Q1	<NULL>	2434
Sam	Milk	Q3	254	325
.

Report #3:

CUSTOMER	PRODUCT	QUARTER	BEFORE_TOT	AFTER_TOT
Emily	Bread	Q4	23	<NULL>
Bloom	Milk	Q2	45	35

Make sure that:

1. Character string data (e.g., customer name and product name) are left justified.
2. Numeric data (e.g., Maximum/minimum Sales Quantities) are right justified.
3. The Date fields are in the format of MM/DD/YYYY (i.e., 01/02/1992 instead of 1/1/1992).

Grading:

- (80 pts.) Logic/Correctness
- (20 pts.) Programming Style (e.g., comments, indentation, use of functions, etc.). You must include a program header, function header, etc. to clearly state what your program and functions are designed to do. Also for inline comments, please state clearly the purpose of those statements – for you as the programmer and to help others better understand your programming logic.

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

**Sample
Command
Line**

\$ sdap2 [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.

Major Area	Item	Max	Deduct	Score	%	Total
<i>Compilation</i>	If fails, subtract ...	50				
<i>Logic</i>	Query/Report #1	30				
	Query/Report #2	20				
	Query/Report #3	50				
	"Minimal Scan" implementation (YES/NO)					
	e.g., 1 scan for queries 1 & 2 and 2 scans for query 3					
	Total	100			80%	
<i>Style</i>	Header Comment	20				
	Function Comment	20				
	Line Comment	20				
	Indentation	10				
	Strings – Left Justified	15				
	Numbers – Right Justified	15				
	Total	100			20%	
<i>Total</i>		100			100%	