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

CS561 – Programming Assignment 1

Due Date: 4/1/2019 (Mon.)

Objectives

- In this assignment, you will *express* "complex" OLAP 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 (such as writing Java programs to produce the same results). Your mission (in addition to writing the SQL queries) is to consider the reasons for the gap (between the *expression* and *evaluation* of such queries) and how to narrow it.

Description

- Generate 4 separate reports based on the following queries (one report for query #1, one for query #2, one for query #3 and another for query #4):

1. 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 *states* in which the sale transactions took place. If there are >1 occurrences of the min or max, display all.

   For the same *customer*, compute the *average* sales quantity.
2. For *each of the 12 months* (regardless of the year), find the *most "popular"* and *least "popular" products* (those products with most and least total sales quantities) and the corresponding *total sales quantities (i.e., SUMs)*.
3. For each *product*, find the "*most favorable*" *month* (when most amount of the product was sold) and the "*least favorable*" *month* (when the least amount of the product was sold).
4. Show for each *customer* and *product* combination, the *average sales quantities for 4 quarters*, Q1, Q2, Q3 and Q4 (in four separate columns) – Q1 being the first 3 months of the year (Jan, Feb & Mar), Q2 the next 3 months (Apr, May & Jun), and so on – ignore the YEAR component of the dates (i.e., 3/11/2001 is considered the same date as 3/11/2002, etc.). Also compute the *average for the "whole" year* (again ignoring the YEAR component, meaning simply compute AVG) along with the *total quantities* (SUM) and the *counts* (COUNT).

The following is a sample output – quantities displayed are for illustration only (not the actual values). **For dates (e.g., MAX_DATE, MIN_DATE), you can display 'month', 'day' and 'year' as 3 separate columns – i.e., you don't need to concatenate them into MM/DD/YYYY format.**

Report #1:

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

. . . .

Report #2:

MONTH	MOST_POPULAR_PROD	MOST_POP_TOTAL_Q	LEAST_POPULAR_PROD	LEAST_POP_TOTAL_Q
1	Eggs	497214	Pepsi	55526
2	Milk	1874794	Banana	23126
3	Pepsi	974531	Milk	19958

. . . .

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

Report #3:

PRODUCT	MOST_FAV_MO	LEAST_FAV_MO
Egg	4	12
Apple	1	11
Banana	3	2
. . . .		

Report #4:

CUSTOMER	PRODUCT	Q1_AVG	Q2_AVG	Q3_AVG	Q4_AVG	AVERAGE	TOTAL	COUNT
Sam	Pepsi	1923	4241	2383	1325	2988	38848	13
Emily	Milk	239	9872	142	2435	2663	21307	8
Helen	Bread	2534	981	4239	1987	2781	25032	9
. . . .								

Grading

NOTE: A query with syntax errors will lose 50% of the points for the query.

Submission

Submit a file containing all of the 4 queries or 4 separate files with each query in a separate file with your name and CWID on it on Canvas. If you create 4 separate files, please place them in a ZIP file and submit the ZIP file.

Please include a "README" file if any special instructions are required.

I encourage you to discuss the "ideas" with your TAs (rather than your classmates, esp, if you have any specific questions), but the final queries must be your own work. If I determine that your queries are copies of someone else's, both you and that someone else will be disciplined (you will receive 0 for the entire assignment) and possibly receive additional penalties for the course.