CS 425 – Database Organization
Summer 2016
Homework Assignment 3

Programming Assignment –
The purpose of this assignment is to give you some hands-on experience writing code that accesses a relational database. Since you already have a lot of work to do with the project assignment, I've tried to structure this assignment such that it will help you also make progress on the project assignment.

As such, the goal of this assignment is to write a program that will…

Run all 22 TPC-H queries against a database 3 times each and gather the elapsed time of each execution of each query. When all the queries finish executing, the program should give you the average elapsed time for each query (the average of the 3 runs of that query), plus the total elapsed time given by summing all 22 averages together.

Your program should first check to see if the TPC-H tables exists. You can assume that if 1 of the tables exists, that they all exist. So, you can do something simple like SELECT COUNT(*) FROM REGION. If the query succeeds, they table must be there. If it fails, just assume that means the tables do not exist. In this case, your program should issue the necessary DDL calls to create them. This must at least include the necessary CREATE TABLE commands, but can include indexes or other objects if you'd like. Once they are created, your program should issue the necessary LOAD or COPY commands to load them. Once they are loaded, your program should issue the necessary ANALYZE commands to generate statistics on them. If the original SELECT COUNT(*) FROM REGION query succeeds but returns a count of zero, that indicates that your tables exist but are not loaded. In that case, you don't need to issue the DDL, but you do need to load them and generate statistics.

Your program should print some kind of status messages when it's doing this DDL / LOAD / statistics work.

Your program should provide a way for the user to specify the directory that contains the files to use to load the tables. Please use the standard file names that come out of dbgen. Your program should also provide a way for the user to specify all the relevant database connection information, this information could vary slightly depending on the database you choose but would at least include hostname (or ip), port number, userid, and password.

Once your program is sure that it has loaded tables with statistics, it can run the TPC-H queries in a loop that looks something like this:

```
for (i = 0; i < 22; i++)
{
	for (j = 0; j < 3; j++)
	{
		results[i][j] = runIt(queries[i]);
	}
}
```

This is going to run all the queries and save their elapsed times. After this completes, you can compute the averages and the total and display those.

Your runIt() method should do the following:

1) Get the current system time in milliseconds

2) Issue the query to the database
3) Print a header line to the screen showing column names (just separate the column names with tabs)
4) Print a line to the screen to separate the header line from the rows, i.e. "-------------------------"
5) For each row, print a line to the screen displaying the values of the columns in that row. You can just separate the value of each column with a tab character. Don't worry about trying to align things to look nice.
6) Get the current system time in milliseconds.
7) Subtract this end time from the start time, and that is the query elapsed time in milliseconds. Return this.

Why do we actually need to go through and print all the rows to the screen like this? The problem is that most databases attempt to stream results back to the client application. In other words, the executeQuery() command (in JDBC terms) returns before the database is actually done processing the entire query. It's not until your application tries to position on a certain row in the result set that the database actually has to have that row available and ready for the client application. Thus we need to iterate through the result set in its entirety and do something with the rows to get a valid value for actual query elapsed time.

Your output should look similar to the output you would get if you ran the query in psql or mysql, but don't worry about formatting it so that everything is nice and aligned.

What to turn in:

1) Source code
2) Instructions for compiling (if it's java, you can alternatively just send me the jar)
3) If it's Java – I need the JDBC driver jar file
4) If it's ODBC – I need instructions for getting the driver and installing it
5) Instructions for running your program, including instructions for how to set the various parameters that the user is allowed to specify (directory to load from, db connection parms)
6) The output of 1 run of your program

If you are newer to programming, I would strongly suggest that you use Java/JDBC as it is much easier than ODBC. I also have much more experience with it and can help you quicker.