Lab 6: Database-Driven Visualization

Based on the lecture this week, you will be building a database-driven visualization for this week's lab. "Database-driven" visualization means that: (a) you will not read a data file, but will instead connect to a live database; (b) every time a user performs an action, your system will send out a database (SQL) query; and (c) given the data elements returned from your database, you will render your visualization accordingly.

Basic Requirements:

a) First, make sure that you can get the support code running. This means that you will need two libraries. One is the ControlP5 library for the user interface elements. The other is the MySQL Database Driver. We've put both of them in right folder(.../code) in the support code.
b) You will connect to a database that we have set up. The database connection code is provided for you, but you will need to be on the EECS network (Wi-Fi or wire if you use a desktop) in order to connect. Also note that the database setup to be "read-only", so you can't modify the content of it.
c) You should see the example in the support code to know how to execute a SQL. We've set every other thing for you. You don't worry about how to set up a connection etc.
d) Writing queries: there are two "types" of queries that you'll need to write
    1) During the initialization stage, you will need to contact the database to find the "min" and "max" values of some of the data dimensions.
    2) During runtime, your code will need to generate a SQL query based on the state of the user interface. Note that the buttons and sliders in the user interface all function as "filters", meaning that when nothing is selected, your system is effectively performing a "**SELECT** * **FROM** forestfire" (more on the database schema later). As the user clicks on more interface elements, your query will have various conditions in "WHERE" clause.
e) Building a visualization: When you run your support code, you will see a blank canvas. Your team will need to draw a scatterplot visualization on that canvas. There are two parts to this:
    1) The database will respond to your query with a ResultSet instance. You will need to parse through this ResultSet and find get the data you want out of it.
    2) With the data, you will need to populate your scatterplot visualization.

Important Notes:

a) Like most commercial databases, MySQL is case sensitive for the variable names, but not the commands. For example, "**select** * **from** MyTable" is the same as "**SELECT** * **FROM** MyTable", but not "**select** * **from** MYTABLE".

b) You need to "clean up" after yourself. You should close the ResultSet object after you are done with it, and you should close the database connection upon exit. We've provided you the structure to manipulate a ResultSet and closed the ResultSet for you. At the same time, we've done the code to close the connection when you clicked the "CLOSE" button on the interface or the close button at the top of the windows. But for those who use Processing IDE, clicking the stop button on IDE menu would not call the method to close the connection.

c) Place you can find the tutorials for writing SQL or using ResultSet:
1) Tuesday's lecture
2) http://dev.mysql.com/doc/refman/5.7/en/sql-syntax.html
3) http://www.w3schools.com/sql/
4) http://docs.oracle.com/javase/tutorial/jdbc/basics/retrieving.html

d) It is relevant to note that while we are writing this program in Processing, this program will NOT run in a browser due to security reasons. Direct connection from the client to a database is strictly forbidden because the connection requires keeping username and password to the database in plain text. Usually, the database connection code is done only on server side. In our use case, we are assuming that this visualization will be used as a desktop application.

e) As we understand that you only have an hour to do this lab, don't worry about how scalable your visualization is. Your primary concern should be about the connection between the database and the visualization.

Database Schema:

a) The database that you are connecting to only has one table named "forestfire". The number of data entries in the database is measured by hundred. Below you can see a screenshot of the structure of the dataset. Notice the id column is an int.

id	X	Y	month	day	temp	humidity	wind
1	6.309813732	4.512422446	mar	fri	8.2	51.0	6.7
2	6.047158094	4.386385792	oct	tue	18.0	33.0	0.9
3	7.035954043	4.459339531	oct	sat	14.6	33.0	1.3
4	7.737249752	5.386428048	mar	fri	8.3	97.0	4.0
5	7.046100744	6.737435854	mar	sun	11.4	99.0	1.8
6	8.919142921	6.793629357	aug	sun	22.2	29.0	5.4
7	7.053397737	5.256608123	aug	mon	24.1	27.0	3.1
8	7.844135233	6.069006865	aug	mon	8.0	86.0	2.2
9	7.356892432	5.00421248	sep	tue	13.1	63.0	5.4
10	7.175825767	4.844435807	sep	sat	22.8	40.0	4.0
11	6.446995638	5.012376277	sep	sat	17.8	51.0	7.2
12	7.424553874	4.708070852	sep	sat	19.3	38.0	4.0
13	5.935986111	5.264111839	aug	fri	17.0	72.0	6.7
14	5.701273314	5.719931311	sep	mon	21.3	42.0	2.2
15	6.861479813	4.962728113	sep	wed	26.4	21.0	4.5
16	5.238456825	5.786193985	sep	fri	22.9	44.0	5.4
17	5.172823052	5.342252744	mar	sat	15.1	27.0	5.4
18	8.384357261	4.962766554	oct	mon	16.7	47.0	4.9
19	6.81116425	3.14939022	mar	wed	15.9	35.0	4.0
20	5.60744989	3.193951325	apr	sat	9.3	44.0	4.5