Java App Notes

Tasks:

JAVA

**~~Step 1: Get the correct DPI scaling of the application~~**

* DPI scaling changed using Resource Hacker run on admin: http://ignitiontips.blogspot.com/2016/07/java-and-high-dpi-displays.html

SIMPLE DB

**~~Step 0: App connect to MSSQL DB~~**

**~~Step 1: App displays simple table database~~**

* When making buttons that can only be selected one, make sure to set them as a group
* Configuring a table in swing: click on table in design > table contents > columns then configure rows (decrement count if you want to hide them)
* Make sure you change the variable name of the table in Display so that you can connect to it with DefaultTableModel

**~~Step 2: App updates table upon insertion of new data into db~~**

* Can’t update without restarting: to change: use defaulttablemodel to get the model, empty the table and then reshow the table using the show\_user() (this is O(n^2) time)
* Shouldn’t need to delete (or really insert) from table since that will be handled with SQL queries

**~~Step 3: App displays certain rows from the simple database (same database as before but with the same column headings – USE SQL Query to select)~~**

* ArrayList<User>: class User defines the parameters of the chart connects to the database and gets the values
* Show\_user(): takes the values acquired from userList and actually displays it in a table
* Make sure SQL query is correct

**~~Step 4: App displays certain rows to another table~~**

* ^This is included in the step above

GRAPH DB  
 **~~Step 1: App displays single table from graph db~~**

* Display query from node and edge table

**~~Step 2: App queries and displays simple extended graph table from graph db~~**

* Done -> make sure to change the class that displays the array depending on what is in the columns (use previous examples to guide you)

**Step 3: App displays information from clicked row into another table**

* Make a new table with all the users
* When a user is clicked, another table should appear (or be populated), showing restaurants they like, rating, and where they live??

**Step 4: App displays food liking scenario**

* See above

select string\_agg(p.name, ','), r.name

from Person p, likes l , restaurant r

where match(p-(l)->r)

group by r.name

having count(r.name) > 1

// needed to add grouping!!

select string\_agg(p.name, ','), r.name, r.city

from Person p, likes l , restaurant r

where match(p-(l)->r)

group by r.name, r.city

having count(r.name) > 1

--select string\_agg(p.name, ','), r.name, r.city

select count(p.name), r.name, r.city

from Person p, likes l , restaurant r

where match(p-(l)->r)

group by r.name, r.city

having count(r.name) > 1

--select string\_agg(p.name, ','), r.name, r.city

select count(p.name) as 'num ppl', string\_agg(l.rating, ',') as 'avg rtg', r.name, r.city

from Person p, likes l , restaurant r

where match(p-(l)->r)

group by r.name, r.city

having count(r.name) > 1

Table still missing location and state from ‘locatedIn table’

VISUALIZATION

**Step 1: When row is clicked, add a node of all the people (see tutorial found)**

Visualization: currently uses code found online

* Graph generation uses: start -> boots background
* Add if condition to “addGraphComponents()” in main to display the correct information
* Store information from SQL query and add it to the nodes
* Does it generate randomly?

**Step 2: Come up with better design than that of Neo4j**

Outline:

When row in table is clicked, a node is generated

Hovering over the node

After making a JFrame form, make sure to change the variable names

* This allows you to call them in your SQL set code as they appear in generated code, otherwise for categories (ex. Buttons = male, female, category = gender) you need to declare them in the public class

In database, make sure to set primary key and data type values

Tasks:

D3.JSON

R -> iGraph library