HW2: SQL

Total points: 6, plus 1 add'l point (related to Q5), but with a CAP of 6 points [if you score 7, your score would 'only' be a 6].

In this assignment, you will code solutions to the **five** SQL problems described below. Guess what - the questions relate back to HW1 - here, you're asked to create tables and data related to the COVID-19 tracing application you considered in HW1, and perform queries on the data.

ALL the SQL knowledge/commands you need to answer the questions have been covered in class! You **do NOT** need to learn more commands or techniques (eg. use of 'triggers') etc. on your own in order to do this HW set.

To run SQL code, you can use one of the three ways mentioned in the lecture notes [a locally installed DB, or a remote server-based DB via an online shell running in a browser page, or a cloud DB] to do the problems.

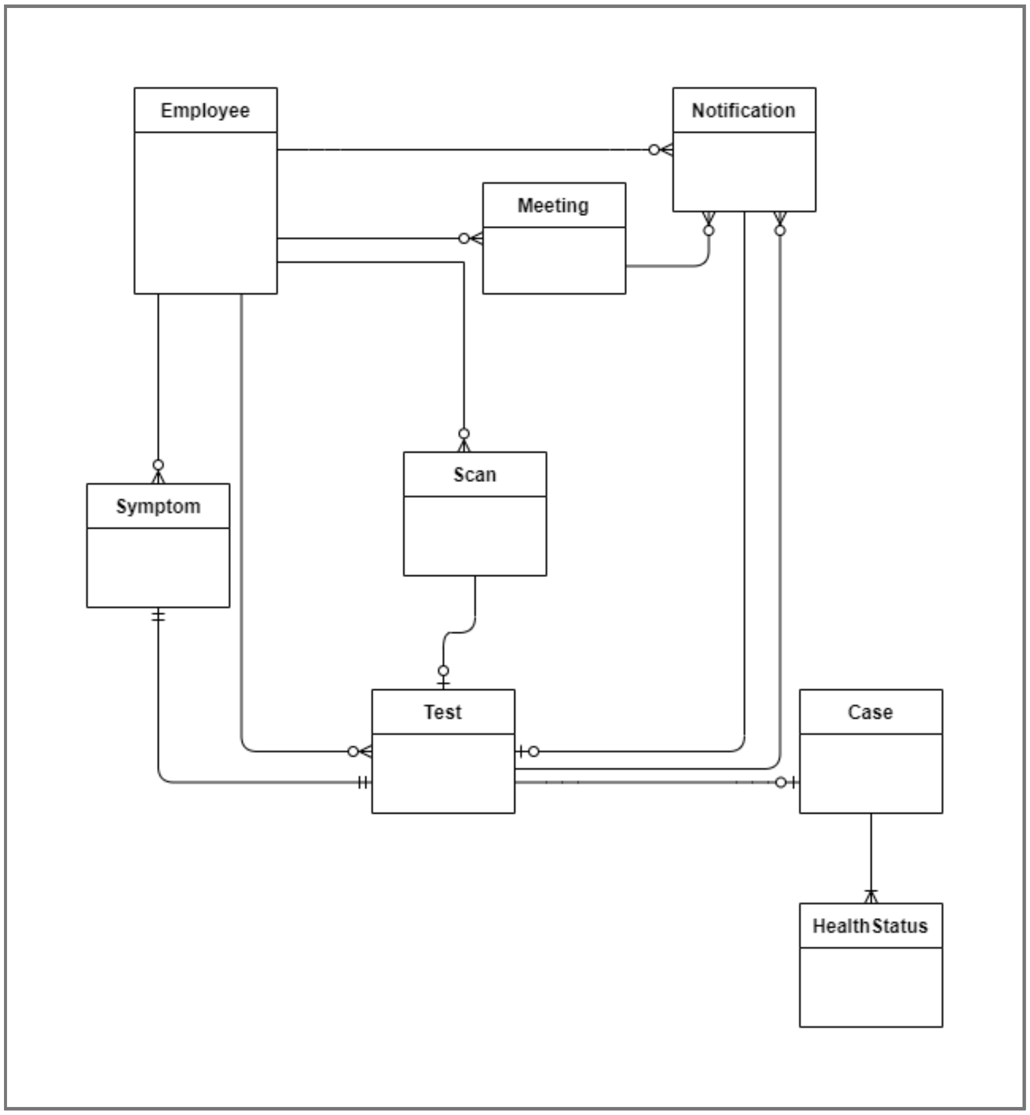
One more cloud-y way, not mentioned in class, to do your homework (and continue learning and practicing SQL) is to use https://livesql.oracle.com/ (https://livesql.oracle.com/); after you sign up for a free account, you can create tables, insert rows and do queries - you can save all your work in separate sessions, reload [any/all of] them after logging in again in order to recreate your tables+data and rerun queries, and also add more tables/data/code to the mix - it's very convenient and cool, try it!

What you need to submit are text files with the SQL commands that you come up with, one file for each question (Q1.sql, Q2.sql.. Q5.sql). PLEASE MENTION AT THE TOP OF EACH FILE, which database (eg. Oracle, SQLite..) you used for that question! Your grader(s) will execute the SQL commands from the text files you submit, using the same software you used, to see if they produce the expected results. You can also submit a README text file containing anything else you want to communicate to your grader, regarding your queries etc.

You can talk to your friends/classmates to informally discuss approaches to the problems, but DO NOT {'share'/look up/ask others for} actual code - that would be plagiarism, and will get you an 'F' in the course.

Please see your TAs/graders if you need one-on-one help (or see me). You can also post (and reply!) in the Piazza 'hw2' forum. Don't wait, start early. Good luck, have fun!

Consider the following 'barebones' design (showing just entity names and connections), for the contact tracing application:



Here are notes on the entities:

- Employee [contains info about employees]: ID, name, office number, floor number, phone number, email address etc.
- Meeting [contains meeting info, on every meeting between employees]: meeting ID, employee ID, room number, floor number, meeting start time (just an int between 8 and 18, standing for 8AM..6PM)
- Notification [based on contact tracing, to alert employees who might have been exposed]: notification ID, employee ID, notification date, notification type (mandatory, optional)
- Symptom [self-reported by employees, any of 5 symptoms]: row ID (1,2...), employee ID, date reported, symptom ID (1 through 5)
- Scan [random scans of employees' body temperatures]: scan ID, scan date, scan time, employee ID, temperature
- Test [to record test details]: test ID, location (company or hospital or clinic etc), test date, test time, employee ID, test result (positive or negative)
- Case [to record employees who test positive]: case ID, employee ID, date, resolution (back to work, left the company, or deceased)
- HealthStatus [self-reporting by employees]: row ID, employee ID, date, status (sick, hospitalized, well)
- time can be ints between 0 and 23, standing for 12AM to 11PM 'on the hour'
- the company has 10 floors, numbered 1 through 10

• do feel free to assume anything else you need, to answer the questions below (please list these in a README)

Q1 (2 points). Write SQL commands to create the above tables (create your own column names), and populate them with data (as many/few rows as you see fit, eg. 25 employees, 15 meetings... you can start small, and add more rows later, to help write queries for Q2..Q5).

Q2 (1 point). Write a query to output the most-self-reported symptom.

Q3 (1 point). Write a query to output the 'sickest' floor.

Q4 (1 point). The management would like stats, for a given period (between start, end dates), on the following: number of scans, number of tests, number of employees who self-reported symptoms, number of positive cases. Write queries to output these.

Q5 (1 point). Create your own query! What else would you like to learn, from the data? Describe/list the question, and come up with the query to answer it. You'll get 1 extra point if your query involves table division [be sure to indicate this in your README].

To reiterate, everything you need is in the slides (the material we went through in class) - you do not need to read ahead or look up more commands online! Look at each relational operator, each command, each function, each keyword that we covered, and ask yourself how it could be of use in constructing your query.

You'll **lose points** if you:

• don't name your files properly

• submit a .zip (you need to submit **individual** files instead)

• neglect to mention what DB software you used

• submit/resubmit after the deadline (be sure to do it ahead of time, and make sure you did)

Good luck, **have fun** working on the problems!!
