

ASSIGNMENT 2 - 601.315/415/615 - Databases

Due date: Friday, October 23, 2020, 1:00 PM (Baltimore time)

Date of Latest Acceptance if using an Extension: Tuesday, October 27, 2020, 1:00 PM (Baltimore time), so no more than 4 late days can be used on this assignment.

What To Do:

- Students in 601.415 and 601.615 should implement the following queries in **SQL**: 1, 2, 3, 4, 5, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49.

Also, Students in 601.415 and 601.615 should write the **QBE** equivalent of the queries: 1, 4, 5, 13, 22, 24, 26, 31, 34, 35, 42, 43, 49.

- Students in 601.315 should implement the following queries in **SQL**: 1, 2, 4, 6, 7, 8, 9, 10, 11, 12, 13, 15, 17, 18, 19, 22, 23, 24, 25, 26, 28, 31, 32, 34, 36, 38, 39, 40, 42, 44, 46, 47, 49.

Also, Students in 601.315 should write the **QBE** equivalent of the queries: 1, 4, 6, 13, 22, 24, 31, 34, 38, 42, 49.

All queries are based on the schema in `jhu.sql` on the class website.

Note that these queries are not necessarily listed in order of increasing difficulty. Also note that queries frequently build on other queries, and subsequent queries may require only a few changes from their predecessors. Question 17 is worth 4 times the points of the other questions.

Students in 601.415/615 have both more difficult queries and a greater number of queries than students in 601.315.

What To Hand In:

You should write all your SQL queries in a text editor, formatted clearly (preferably using `SELECT/FROM/WHERE/GROUP_BY` commands in all caps, with attribute names in all lower case and relation names with the first letter capitalized).

Submit this program as a plain text file named `yourusername.sql.txt` using gradescope as instructed.

You should also submit the output of your SQL code running on the actual database tuples in `jhu.sql` as instructed on piazza.

Finally, you are required to submit an electronic copy `yourusername.qbe.txt` of your QBE queries based on editing the plain-text schema we have provided for you in `jhu_schema.txt` on the class website, again using gradescope as instructed.

Queries:

The relational table specifications for all queries used in this assignment are given in the file `jhu.sql` on the class website.

You may *NOT* use views or create temporary tables to store intermediate results to simplify computation. Use nesting of expressions or derived relations in the FROM clause as appropriate.

You may not use the sql JOIN operator in the FROM clause for queries 1-32. It is permitted (but not required) for queries 33-49.

For the purposes of this exercise, the enrolled_in relation only contains one semester's worth of courses (student's classes for the most recent semester) and the students' grades for those courses. There are no semester/year attributes in the enrolled_in relation. Questions concerning class enrollment should use the entire relation.

When a question asks "List the name" of a person, give their first name followed by their last (family) name.

If two or more entities are tied for the answer condition (e.g. for maximum, minimum, etc.), then list all matches. You should never assume that there is only one minimum, maximum, etc.

1. List the names of all pairs of students who are from the same city and like each other but don't love each other.
2. List the name of the car-owning student in the database who owns the lowest MPG vehicle (of all the student-owned cars), along with the Manufacturer, Model, and MPG of that car.
3. List the name(s), age and major of student(s) who own every model of car manufactured by Nissan listed in the database. You cannot assume that Nissan only manufactures 3 models, as this can change.
4. List the names of students who live in a dorm and own more than one car.
5. List the names of students who live in a dorm and don't own a car but own a pet.
6. List the names of students who live in a dorm and own exactly two cars and own two or more pets.
7. What is the minimum, maximum and average MPG of all models of Porsche listed in the database?
8. What is the minimum, maximum and average age of students who live in the JHU dorms and do not own a car?
9. List the names and ages of all students who do not live in a dorm.
10. What is the average age of students who participate in more than 2 activities?
11. How many students participate in the most popular student activity, and what is the name of this activity?
12. List the names of all activities with no student participants which do have faculty participants.

13. List all students enrolled in a course with someone who is who is enrolled in a course with someone who roommates with someone whose hometown is in Pennsylvania (PA) and voted for Donald Trump in 2020.
14. List the names of students who share at least one activity with their advisor; exclude advisors who do not teach any classes. List the name of the advisor in addition to that of the student.
15. Find all pairs of students who are roommates and from different countries. List each pair only once. (List their names).
16. List the dormitory whose residents have the highest grade point average.
17. You have been given two tables relating to location of and distance between cities: `DIRECT_DISTANCE` and `CITY`. `CITY` contains the latitude and longitude of each city. `DIRECT_DISTANCE` contains the actual distance (great-circle distance, or in other words, shortest distance along the surface of the earth) between two cities. However, `DIRECT_DISTANCE` as it is given is an incomplete relation: not all city pairs are represented there. Your job is to augment `DIRECT_DISTANCE` with approximated distances where needed. First note that, conveniently, the direct distance between Baltimore and every other city is listed in the `DIRECT_DISTANCE` table. This will form the basis of the first distance approximation.
 1. First, create a table called `BALTIMORE_DISTANCE`. This table represents the “distance via Baltimore” between any two cities. For example, the Baltimore distance between Washington, DC and Philadelphia is: `direct distance from Washington to Baltimore + direct distance from Baltimore to Philadelphia`. In this case the direct distance and the Baltimore distance are comparable, at least. Now consider the Baltimore distance between San Diego and Los Angeles. In this case Baltimore distance is not a very good approximation of real distance. In any case, you are to create the table `BALTIMORE_DISTANCE` having the same attributes as `DIRECT_DISTANCE`, and write a query that populates it with the direct distance between every two cities. Be careful to make your query insert two tuples for every city pair: e.g., for Boston and Hong Kong, the tuples might look like `(BOS,HKG,9000)` or `(HKG,BOS,9000)`.
 2. Secondly, create a table called `RECTANGULAR_DISTANCE`. Write a query to populate it with the approximate distance between each city pair using the formula:

$$\sqrt{(d * lat(city1) - d * lat(city2))^2 + (d * long(city1) - d * long(city2))^2}$$
 where $d = 70$. In other words, this is the euclidian distance between the two points IF longitude and latitude are considered to be a rectangular coordinate system. As above, populate your table with two tuples for each city pair. (Note: 70 is the approximate number of miles per degree latitude)
 3. Thirdly, create a table called `ALL_DISTANCES`. This table should have tuples of the following type: `(city1,city2,direct_distance,baltimore_distance,rectangular_distance)`. Write a query to populate this table.

4. Finally, create a table called BEST_DISTANCE with tuples of this type: (city1,city2,distance).
Write a query that for each city pair takes the minimum of direct_distance, baltimore_distance and rectangular_distance and inserts it into BEST_DISTANCE. Clearly, direct_distance will always be selected if it is present. It is guaranteed to be less than or equal to the approximations we are using.
For the remaining queries that require a distance relation between two cities, use the BEST_DISTANCE relation you have just created.
18. List each city that is home to at least 2 students, and how many students live there.
19. Find all students living in dorms with fewer than 300 residents, whose homes are within 100 miles of someone else in the dorm they live in. List the name and city, state and country of origin for each of these students.
20. For every country, list the student(s) whose homes are in the city furthest from Baltimore.
21. List the name of the activity whose participants' homes have the greatest average distance from Baltimore.
22. List the names and ages of all female students minoring in a department in the engineering school and taught by a female professor whose primary appointment is in the engineering school.
23. List the names and student ID numbers of all students who are enrolled in every course taught by Paul Smolensky.
24. List the names and ID numbers of all students who are enrolled in a class with a student who is enrolled in a class that Linda Smith is enrolled in, and also lives in the same state that Linda Smith lives in and also voted for the same president that Linda Smith voted for in both 2016 and 2020.
25. List the names of courses that are enrolled with students who are not a member of any club, who have no allergies, and who likes people who is both in a club and has an allergy.
26. List the first and last name and dorm name of any student with at least one conduct violation, and the total number of his/her violations.
27. List the first and last name and dorm name the student with the most total conduct violations, along with that total. If there are multiple tied students tied, list all.
28. List the name, capacity and amenities of the dorm with the most alcohol conduct violations.
29. List the name and department of classes where the enrolled students have the most conduct violations (of any type).
30. List the names of student organizations whose members collectively have over 3 conduct violations.

31. List the names of all students who live in the same dorm room but voted for *different* candidates for US president in 2020, and also list the names of the presidential candidates that each voted for. Only include roommates who both voted for president in 2020 and only list each pair once.
32. List the name of the dorm with the largest total number of students who have voted for Donald Trump for US president in 2020 (along with that total number of students).
33. List the name of the dorm with the highest percentage of students who have voted for Donald Trump for US president in 2020 (as a percentage of the total number of students living in the dorm (NOT capacity), regardless of how many students in that dorm voted for president in 2020). For that dorm, include the number of votes for Donald Trump, the total number of students living in the dorm, and the percentage of votes for Donald Trump.
34. List the names and ages of students who have voted for different candidates for US president in 2016 and 2020, as well as the names of and political parties of the candidates they voted for in each election. The student should must have voted in both elections.
35. List the names and home state of students who have voted for different political parties for US president in different years (e.g. Republican in 2016 and Democrat in 2020). Ignore years when the student didn't vote.
36. List the names of all people who worked as an Intern (Intern anywhere in his/her job name) at the same time that they have lived abroad (i.e. where the dates overlap by at least one day). For example, the two ranges 2020-06-17 through 2020-08-23 and 2020-08-21 through 2020-09-12 overlap by 3 days (8/21, 8/22 and 8/23)).
37. List the names of all people who worked as an Intern (Intern anywhere in his/her job name) in at least two positions at the same time (i.e. where the dates overlap by at least one day).
38. List the names of all people who have studied abroad in the same country where they live now, along with the name of that country.
39. For each company, list the total number of JHU Computer Science majors who have worked for that company.
40. For every student internship (with Intern anywhere in the job title), list the name of the student, the company where they worked, the start and end dates of the internship and the total number of days the student worked there (including the start and end date).
41. List the name of the student with the longest total internship, as well as the name of the company they were an intern for and the total number of days the student worked there. As before, an internship is a job with the word Intern anywhere in the job title.

42. List the student name and dorm name of all the students who live in a dorm that has a pet living there of the same type that the student is allergic to. Assume that if a student both has a pet and lives in a dorm, that pet lives with the student in the dorm. The student and pet don't have to live in the same room.
43. List the names of every pair of students love each other and live with someone that has a pet with the same name as the first name of the person they love. Include the pet's name and list the person who matches the pet's name 1st.
44. List the name and age of the student who owns the oldest dog in the database. Also include the name of the pet and the age of the pet next to the student's name.
45. Assuming that dogs cannot live with cats, cats cannot live with parrots but parrots can live with dogs, which pair of students who own pets *cannot* potentially be roommates. List the student names, dorm name and dorm room number of both students, with each pair listed only once.
46. For each dorm, list the name of the dorm and the total number of occupied rooms in the dorm (rooms with at least one listed occupant), and the total capacity of each dorm.
47. For each dorm, list the name of the dorm and the total number of rooms in the dorm with at least one pet living in that room.
48. For each dorm, list the name of the dorm, the total number of pets living in the dorm, and the percentage of occupied rooms in the dorm which have at least one pet living in that room. Do not use the dorm capacity, but the total number of rooms with a listed occupant.
49. Invent a complex, interesting question of your choice using the data in the `jhu.sql` schema and write both a SQL and QBE query to compute the answer to your question. Grading of this question will be based as much on your creativity as the correctness of your solution.