

# ASSIGNMENT 2 - 600.315/415 - Database Systems

**Due date:** Friday, October 18, 2013, 3:00 PM

## What To Do:

- Students in 600.415 should implement 12 of the queries 1-14 and 11 of the queries 16-29 in MySQL as well as any 10 of the queries from 31-44. In addition, *all students in 600.415 must answer all parts of question 30*. Also, they should write the QBE equivalent *all* of the queries 16,17,22,23,24,29,35,37,39 (on paper).
- Students in 600.315 should implement 8 of the queries 1-14 and 9 of the queries 16-29 in MySQL, also query 30, 32, 33 and also any 5 of the queries from 34-44. Also, they should write the QBE equivalent to any 6 of the queries 16,17,22,23,24,29,35,37,39 (on paper).

## 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 using the online submission program with the following URL: <http://www.ugrad.cs.jhu.edu/cgi-bin/cgiwrap/cs415/submit.pl>. Instructions are provided there.

In addition, you should submit a *spool* log of your sql code running on the actual database tables in <http://www.cs.jhu.edu/~yarowsky/jhu.sql> or /home/cs415/jhu.sql (on dbase.cs.jhu.edu). Options for doing this (such as cutting and pasting the sql code into the MySQL interpreter are covered in class). You should also submit this spool file via the interface above.

Finally, you are required to submit a *hard copy* of your SQL queries **and** your QBE queries (on paper!), but **not** your SQL spool file. Please submit them as separately stapled bundles (marked HW2a and HW2b) with your name on each one. They will be graded in parallel.

If you are unable to attend class the day the assignment is due, the date/time of email submission will determine on-time submission, not the date/time of the hard copy submission.

## Queries:

The relational table specifications for all queries used in this assignment are given in the file /home/cs415/jhu.sql on dbase.cs.jhu.edu. They are specified in SQL CREATE TABLE syntax.

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

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.

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.

1. List the name and age of the youngest student in every major. If there is a tie, list all students tying.
2. For each major, list the name of the major, the student with the lowest GPA in the major, and the name of that student.
3. List the Student Id, name, major, gender and grades of any student who likes everyone in Algorithms.
4. List the course name, Major, Grade and name of students who love each other in the same course.
5. List the most common last name of all female student and the number of times that name appears (if there is a tie, list all.)
6. List all the trios of students (give this as name1, name2, name3) who satisfy the following constraints: A loves B, B loves C, and C loves A.
7. List the name and age of the youngest student in every major. If there is a tie, list all students tying.
8. List the most common first name of all the female students in the database, and the number of times that name appears (if there is a tie, list all)
9. List the name major and advisor of students whose age is less than the mean age of the class.
10. List the most popular minor for students in Mathematical Sciences and Computer Science departments
11. List all pairs of people living together who are incompatible - i.e. one smokes and other doesn't find it acceptable or they have different study habits. Note - just give out pairs of people, with each qualifying pair listed exactly once; you do not have to go into further details.
12. Select students who could possibly TA a course for next fall ( The students should have atleast a B+ overall and an A in the course they should TA
13. list the courses - as a suggestions to every student who loves another - so he can spend more time with him/her.

14. For each person in the database, find the difference between the number of people that the person likes, and the number of people that like the person. Give their name, as well as the numbers liked by them, liking them and the difference.
15. List the names and room number of all students who live alone in a dorm room
16. List the name, age and major of all students who do not live in a dorm.
17. List the average age of the students who have at least one parent on the JHU faculty.
18. List the name, age, and sex for all students enrolled in 600.315 or 600.415, and also enrolled in any computer science course in the 100 or 200 level (e.g. 600.224).
19. List all the faculty with a primary appointment in the Computer Science department in decreasing order by the total enrollment in all their classes (including their name and total combined enrollment).
20. List the names and total enrolled credits for all students who are enrolled in more than the average total number of enrolled credits for all students in the database (give the student name, advisor name, and total credits enrolled).
21. List the names and ages of all students who have an age more than one standard deviation greater than the average age of students in the database.
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.
25. For each course with total enrollment greater than 0, list the total number of A/A-/A+'s awarded in the class, the total enrollment in the class, and the percentage of awarded grades in the class that are A- or above (also include the course name and instructors name).
26. List the name, course id, and instructor for the course having the highest percentage of its students' grades below B-. You should write your query so that in case of a tie for this distinction, all of the tied courses are listed.
27. What is the average age among students who participate in more than 2 activities?
28. List names of all people who participate in the "extreme canasta" activity, including both faculty and students.
29. List names of all pairs of roommates. You can assume that no rooms contain more than 2 students, but be sure to write your query so that each pair of roommates is listed exactly once.

30. 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.

- (a) 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) (HKG,BOS,9000)
- (b) 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)

- (c) 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.
- (d) 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.

31. List the name of the dormitory whose resident’s homes have the greatest average distance from Baltimore.

32. List the name, age and major for all students taking courses outside their major who have a higher GPA in their non-major courses than in their major courses.
33. For each department, list the mean number of students advised by faculty members with primary appointment in that department. For example, if a department had 5 faculty members advising 2, 10, 3, 1 and 4 students respectively, their mean number of advisees per faculty member for that department is 4.
34. How many students participate in the most popular student activity, and what is the name of this activity?
35. List the names of all activities with no student participants which do have faculty participants.
36. List all students enrolled in a course with someone who is roommates with someone whose hometown is in Pennsylvania (PA).
37. Find all pairs of students who are roommates and from different countries. List each pair only once. (List their names).
38. For each major, list the average number of activities that the students in the major participate in. Only do this for majors with more than 1 student.
39. List the name of all faculty members who advise a student who is enrolled in a course with a student who is a roommate of a student who majors or minors in the same department that the faculty member is a member of.
40. List the names and total enrolled credits for all students who are enrolled in more than the average total number of enrolled credits for all students in their major (give the student name, advisor name, and total credits enrolled).
41. List the classname, instructor name and total enrollment of all courses whose enrollment is greater than the average enrollment in the database.
42. List the names, rooms and building locations of the faculty members with secondary affiliation to the CS department only if they are not located in NEB.
43. Find the total number of CS majors who are smokers and who do not like anyone in the database.
44. Invent a complex, interesting question of your choice and write a SQL query to compute the answer. Grading of this question will be based as much on your creativity as the correctness of your solution.