

ASSIGNMENT 2 - 600.315/415 - Database Systems

Due date: Tuesday, October 20, 2009, 3:00 PM

What To Do:

- Students in 600.415 should implement 7 of the queries 1-14 in SQL as well as all of the queries from 16-30, including queries 6 and 10. Also, they should write the QBE equivalent *all* of the queries 2,5,6,7,12,14,20,26,30 (on paper).
- Students in 600.315 should implement queries 6, 10, 10, 17, 19, 27, 28, 29, 30 in SQL, any four additional queries in 1-14 in SQL, and also any 5 of the queries from 20-26. Also, they should write the QBE equivalent to any 6 of the queries 2,5,6,7,12,14,20,26 (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](http://home.cs.jhu.edu/~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](http://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, major and gender of any student who likes everyone in the database.
2. List the names and gender of of all people in the database who either teach or are enrolled in a computer science course.
3. List the name, age and gender of all students who do not have a roommate.
4. List the name, age and major of all students along with the total number of credits they are enrolled in.
5. List the name and major of all students who are enrolled in more than 20 credits.
6. 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.
7. List the names of all students who like every other student who is in their major.
8. For each major, list the total number of credits taught by the major (the credits taught in a class is computed by the enrollment times the number of credits for the course).
9. List the name and age of the oldest student in every major. If there is a tie, list all students tying.
10. For each course with a total enrollment greater than 0, list the total number of grades lower than B in the class, the total enrollment in the class, and the percentage of grades in the class that are B or below (also include the course name and instructors name). To simplify this query, you can make use of the GRADECONVERSION relation and the fact that a B has a grade point of 3.0. Also, B- is lower than B.
11. List the names and student ID numbers of all pairs of student who love each other. (Do not list a pair more than once, regardless of the order of names).
12. List the names and genders of all pairs of student who love each other and are also roommates.
13. List the names of all students who love someone who does not love them back.
14. List the the names of all pairs of roommates where there exists a third party who loves them both.
15. List the names and genders of all students who like but do not love Linda Smith.

16. List the names of all students who love each other but have different SleepHabits or smoking preferences.
17. List the name and gender of the student who is liked by the most other students, including the total number of students who like him or her.
18. List the name, age, sex and major of all students along with the total number of students that they like, the total number of students who like them. Also add a final column showing their liked/disliked ratio (number of students who like them divided by the number of students who dislike them). You can assume that all students have at least one person who dislikes them.
19. List the name of the student liked by the greatest number of students. If a tie, list all tied winners.
20. For all students, list their name and the total number of students who either like or love them. Do not count the same individual twice. Sort by this total in descending order.
21. List the minimum, maximum and average number of students that the students in the database love.
22. List the name, major and age of all students whose age is greater than the mean age of students in the database.
23. 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).
24. List the classname, instructor name and total enrollment of all courses whose enrollment is greater than the average enrollment in the database.
25. List the names of all students who likes someone who likes someone who likes Linda Smith.
26. List the names of all students who either like or is enrolled in a class with someone who either likes or is enrolled in a class with Linda Smith.
27. List the total course enrollment for each division of the university.
28. List the most common first name in the database (including first names of both faculty and students), along with the number of times this first name appears in the database.
29. List the department with the smallest total student enrollment (summed over all of their courses), and also give this total enrollment and the department name.
30. 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.