

# ASSIGNMENT 2 - 600.315/415 - Database Systems

**Due date:** Thursday, October 16, 2008, 3:00 PM

If using the one free extension, assignments may be submitted no later than 5 days after the deadline above (3:00 PM on Tuesday, October 21, in class). Solutions will be distributed at that time.

## What To Do:

- Students in 600.415 should implement queries 1-18 in SQL as well as any 8 of the queries from 19-28. Also, they should write the QBE equivalent to 8 of the queries 1,2,5,10,15,16,17,18,19,20,28 (on paper).
- Students in 600.315 should implement 13 of the queries 1-18 in SQL, plus any 4 of the queries from 19-28. Also, they should write the QBE equivalent to 6 of the queries 1,2,5,10,15,16,17,18,19,28 (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 names of all students who are smokers or have environmental allergies (allergy type is "environmental").
2. List the names of all students with more than 1 food allergy (allergy type is "food").
3. List the names of students who suffer from every allergy type.
4. List the most common allergy name (and its allergy type) suffered by students older than 25.
5. Find all roommates who are not compatible on some dimension (at least one difference in smoking, sleep-time or music preferences). Note that for smoking the following pairs *are* compatible: Yes/no-accept and no/no-accept.
6. List the percentage of nonsmokers for all majors with at least 1 person in the database (include the number of smokers and the total in the major), and include the name and number of the majors.
7. List the name and number of the major with the greatest percentage of smokers (including only majors with 3 more more students). In the case of ties, you can list one or all winners.
8. List the name and type of the most common allergy of students from each state (ignore states with fewer than 2 people).
9. List the names of all students who suffer from environmental allergies and also are smokers.
10. List the name, age, and major of all students enrolled in a class taught by their advisor (also include the name of the advisor).
11. List the total course enrollment for each division of the university.
12. Produce a list of all distinct faculty first names and the number of times each of these names occurs (sorted in decreasing order of frequency and limited to first names occurring more than once).
13. 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.

14. 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.
15. List the student name, course name, instructor name, and letter grade for all classes enrolled in by students who are early risers and have no allergies and do not smoke.
16. List the name and age of both the oldest and the youngest student in the database (include in the same table).
17. For each department, list the total number of A's awarded in a course offered by the department (A/A-/A+), the total combined enrollment for all courses in the department, the percentage of awarded grades in the department that are A- or above, the department's name and department chair's name.
18. List all pairs of students enrolled in the same course and sharing the same first name (give FN1, LN1, FN2, LN2 where FN1=FN2). Make sure that students are not paired with themselves. Also, because of symmetry, each pair will appear twice in the result in reversed order (e.g. (John Smith, John Winters) and (John Winters, John Smith)). Eliminate this duplication (this can be done as a simple change when eliminating self pairings).
19. Find the total number of CS majors who are smokers and who do not like anyone in the database. (for the QBE question, find the *names*, not number.)
20. 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.
21. List all the students who have minored in Math, but also have an 'A+' or 'A' from a computer science course.
22. List all students who smoke and have working fireplaces in their dorm.
23. List all the courses Bruce Wilson is enrolled in, giving the course name, the number of credits offered by the class (e.g. 3), Bruce's letter grade in the class, and his numeric gradepoint for the class. For example:

CID	Course Name	Credits	LetGrade	GradePoint
340.108	Intermediate Basketweaving	3	A	4.0
220.209	Physics for Poets	3	B	3.0
600.117	Exploring the Internet	3	B+	3.3
340.500	Beginning Pet Grooming	2	A-	3.1

To help you with the letter-grade to gradepoint conversion, a relation called grade-conversion has been defined for you.

Lettergrade	GradePoint
A	4.0
A-	3.7
B+	3.3
...	...

24. Compute Bruce Wilson's grade point average (for all courses listed for him in the enrolled\_in relation), restricted to courses in his major. The GPA is defined as the sum of (grade\_point × course.credits) for all his major courses divided by the sum of his course.credits for all his major courses. For the example above, assuming his major is 340, his major GPA would be (12 + 6.2)/5. You need only to list his student ID number, total number of credits he has enrolled in and his major GPA.
25. List the first and last names of all students in the database and their GPA's (restricted to courses in the student's major and calculated as in the problem above). Format the GPA so that it shows only one value to the right of the decimal point.
26. For each department with greater than 3 student majors, list the most popular music preference of its students. If there is a tie, give any or all of the tying preferences.
27. Create a table (filled with appropriate values) that maps between a letter grade and the next lower letter grade. Assume that the grade lower than F is F. For example:

LetterGrade	NextLower
A+	A
A	A-
A-	B+
B+	B
B	B-

Also, write an SQL command that uses this table to lower the grades of all students enrolled in courses taught by their parents to the next lowest grade (e.g. B to B-). **Do NOT** actually run this last SQL command, just write it.

28. 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.