# ASSIGNMENT 2 - 601.315/415/615 - Databases

**Due date:** Thursday, October 21, 2021, 3 PM (Baltimore time)

Students will be given 3 extra optional/free late dates to use on any homework assignment during the course if desired (for a total of 8). However, solutions to HW2 will be distributed at 3PM on Tuesday, October 26, 2021 in preparation for the exam, and no submissions are allowed after that date, so a maximum of 5 late days can be used for this assignment.

## What To Do:

- Students in 601.315 should implement the following queries in SQL: 1,2,3,4,7,9,10,11,13,14, 16,17,18,19,22,24,25,28,31,32,33,34,37,40,41,42,43,44,45,47,50,53,54,55,57,58,59,60,61,62, 64,66,67,68,70,71

- Students in 601.415/615 should implement the following queries in SQL: 1,2,3,5,6,8,9,12,13, 15,16,18,20,23,25,26,27,28,29,30,32,33,34,35,36,37,38,39,40,41,44,45,46,48,49,50,52,53,54, 55,56,49,61,63,65,66,67,69,70,71

- Also, All students in 601.315/415/615 should write the QBE equivalent of *all* of the queries 3,18,19,40,41,54,60,71 using the empty jhu.schema provided (this involves very few keystrokes, a strength of QBE).

- Please note that while the students in 601.315 have a relatively similar number of queries to students in 601.415/615, their queries are substantially simpler and shorter to write.

- Please also note that a target goal for this class, for both homeworks and the exams, is that students are able to write solutions to queries on their level in roughly 5 minutes each, on average. Based on this goal, this homework should take 4-6 hours to complete, and is good practice towards reaching that goal.

- 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.

## 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 using gradescope as instructed.

In addition, you should submit a spool log of the output 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 the undergraduate network). 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 gradescope.

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 `http://www.cs.jhu.edu/-yarowsky/jhu.schema`. 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 views or use 'CREATE TABLE' to store intermediate results to simplify computation. Instead, you may 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, at a minimum give their first name followed by their last (family) name.

1. List the student name, age, sex and full name of their home city for all students who have voted for Donald Trump in both 2016 and 2020.

2. List the student name and major name of all students who are enrolled in a course taught by Jason Eisner or are advised by Jason Eisner.

3. List the student name and major name of all students who are **NOT** enrolled in a course taught by Jason Eisner and **NOT** advised by Jason Eisner.

4. List the name, age and sex of all students with no roommates.

5. List the student names and major names of all students who are not enrolled in any courses from his/her major.

6. List the names of activities where at least 3 students participate. Include the activity name along with the number of participants. Output should be sorted first by the number of participants (descending), and then by the activity name (dictionary order).

7. List the student name and major name of the students who participate in at least two activities including Baseball.

8. List the student name and major name of all students who participate in at least 2 activities, and also have never received a grade below B- in any of the courses s/he is enrolled in.

9. Find the student(s) who is (are) loved by the highest number of people. Report the students' name, major name, and the total number of activities s/he participates in.

10. Find the name of all faculty members whose office building is different from his/her department's building. Report the faculty name, department name, faculty office building, and department building.

11. List the dormname and student capacity for female dorms where there are no amenities.

12. List the dormid and student capacity for the dorm with the most amenities. If there is a tie, list all tying dorms.

13. List average age of the students who do not participate in any activity.

14. List the student name, major name and advisor name of the student in the database who lives the farthest direct distance from Baltimore, MD. You should not include the city code for Baltimore in the query, just the name "Baltimore" and state "MD".

15. List the student name, major name and advisor name of the student enrolled in "Computer Vision" who lives the farthest direct distance from Baltimore, MD. You should not include the city code for Baltimore in the query, just the name "Baltimore" and state "MD".

16. List dorm id and dorm name whose resident home is farthest from Baltimore, MD. You should also include the name and sex of that student, and you should not include the city code BAL in your query.

17. List all students who smoke and have working fireplaces in their dorm.

18. Print the names of all students from New York, who live in Wolman, who major in Computer Science, who are allergic to peanut butter and who are majors in the Computer Science department.

19. List the names of all activities that at least one female likes and no female loves.

20. List the names and course numbers of all courses for which there are fewer than 3 students are enrolled.

21. List the names of all students having an "environmental" allergy and are loved by at-least one other student.

22. List the name and number of the course that has the most students enrolled in it living in "Bud Jones Hall".

23. Print the total number of A's or A+'s received by students in each class offered by the Computer Science Department or Electrical and Computer Engineering Department.

24. List the names of all students with more than 1 food allergy (allergy type is "food").

25. List the names of students who suffer from every allergy type.

26. List the most common allergy name (and its allergy type) suffered by students older than 25.

27. Find all roomates 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.

28. 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.

29. 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.

30. List the name and type of the most common allergy of students from each state (ignore states with fewer than 2 people).

31. List the names of all students who suffer from environmental allergies and also are smokers.

32. List the name, age, and major of all students enrolled in a class taught by their advisor (also include the name of the advisor).

33. List the total course enrollment for each division of the university.

34. 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).

35. 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.

36. 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.

37. 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.

38. List the name and age of both the oldest and the youngest student in the database (include in the same table).

39. 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.

40. 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).

41. 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.)

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. List all the students who have minored in Math, but also have an 'A+' or 'A' from a Computer Science course.

44. 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 |
| ... | ... |

45. 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 (gradepoint × 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.

46. 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.

47. How many students who are both smokers and late nighters have a GPA greater than 3.0?

48. List all dorms and the average GPA of their residents, sorted by GPA in descending order.

49. 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 tieing preferences.

50. 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.

51. List the name and dorm-gender of the dorm where the most number of students voted for Donald Trump in *either* 2016 or 2020.

52. List the name and dorm-gender of the dorm where the highest percentage of students voted for Donald Trump in *either* 2016 or 2020, ignoring dorms with no students.

53. List the name and total occupancy of the dorm with the largest number of students who live that dorm (based on Lives_in), along with that dorm's official capacity.

54. List the names, ages and home city name of all students who live in the dorm with the largest number of students (based on Lives_in, not capacity).

55. List the dorm names and official student capacity of dorms where the total number of students actually living in that dorm (based on Lives_in) exceeds the official student capacity of that dorm.

56. List the dorm names and official student capacity of dorms where the total number of students actually living in that dorm (based on Lives_in) exceeds the official student capacity of that dorm, along with that actual number of students, and the percentage above capacity (e.g. 55 students living in a 50 capacity dorm is 10% over capacity).

57. What is the most common research area among students that have interned as a software developer?

58. List the club memberships of the faculty member who mentors the most student researchers.

59. List the research topics and respective student majors of students who are doing research with a faculty mentor who is not in the department they are majoring or minoring in.

60. List the names and ages of all students who have not had any close contacts with people that they like.

61. List all students who have had a close contact of at least 30 minutes with students who have received a positive covid test in the last 14 days.

62. List all individuals who were in a class where someone slept.

63. List all pairs of students (student one, student two) where it is likely that student one gave student two covid (either they do the same activity, or student 1 has had a close contact with student 2 after receiving a positive test for covid less than 14 days ago).

64. List the most stimulating professor (professor that has taught the fewest classes where students have fallen asleep during class), excluding those professors who do not teach any classes.

65. List the most stimulating professor (professor that has the lowest percentage of students who have fallen asleep during class), excluding those professors who do not teach any classes.

66. List the names, age and major of students who have enrolled in no classes offered by the computer science department.

67. List the names, age and major of students who have enrolled in all classes offered by the computer science department.

68. List the names, age and major of students who received the highest grade in Distributed Systems, along with that grade.

69. For every class listed in the database, list the names, age and major of students who received the highest grade for every class listed in the database, along with that course name, number and the student's grade. Ignore courses with no enrollments, and if there is a tie, list all tying students on separate lines (i.e. one student/course pair per line).

70. List the names, age and major of students who have enrolled in 2 or more courses in computer science, and the total number of courses in computer science that they have enrolled in.

71. 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.