ASSIGNMENT 2 - 601.315/415/615 - Databases

Due date: Monday, October 27, 2025, 11:59 PM (Baltimore time)

What To Do:

- Students in 600.415/615 should implement in SQL all queries 4-28 plus **all** of the queries 51, 53, 54, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 68, 69, 71, 73, 75, 78, 80, 81, 82, 83, 84, 86, and 87.

  Also, they should write the QBE equivalent *all* of the queries 2,5,6,7,12,14, and also 56,57,61,67,68,84,88.

- Students in 600.315 should implement queries 1-5, 6, 10, 11, 17, 19, 20-25, 50, 52, 54, 55, 59, 60, 61, 62, 63, 64, 65, 66, 67, 70, 72, 74, 76, 77, 79, 82, 83, 84, 85, and 87 in SQL. Also, they should write the QBE equivalent to the queries 2,5,6,7,12,14,88 and also any 3 of the queries 56,61,67,68,84.

- 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`.

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.

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 roomate 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 gradepoint 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 total course enrollment for each division of the university.

20. 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.

21. 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.

22. List the names of students who have taken different class taught by Yair Amir (i.e. any two of them have not taken the same course taught by Yair Amir), have played video games, and have not played the same video games.

23. Increase the GPA of students who represent JHU in sports by .1 and print out all students and their new GPAs.

24. List the name of students who have food allergies as well as are taking more than 3 subjects in a semester

25. What is the average GPA for students playing each sport?

26. Which sports involve the minimum and maximum hours played each week by students at JHU?

27. What is the difference in the average number of games played by scholarship and non-scholarship students?

28. For each department, list the number and average GPA of students who play a sport.

29. What percentage of early risers play sports?

30. How many amenities does the dorm which houses the most sports-playing students have, and is it a male or female dorm?

31. How many women play basketball or tennis, and also study music?

32. How many students in each department have an allergy and also smoke, but do not play any sports.

33. Assuming a course workload is 10 hours a week, which student(s) have the highest workload across both sports and academics, and how many hours a week do they spend on both these activities combined?

34. List the name of the student liked by the greatest number of students. If a tie, list all tied winners.

35. List the names of courses with video game players who have played at least 50 hours in total. Then sort descending by the number of players.

36. List the dorm names of those that have at least 1 video game player and none of whom have played the game "Europe is the Universe."

37. Print the total number of hours of video games played by students who do not smoke and like at least one person in the database.

38. List the names of students who do not play video games and have received an 'A' or better in a class offered by the Computer Science Department.

39. Print the average number of games played by students who do not participate in any activity and do not live in the same dorm.

40. List the names and department names of all students who are a CA for a course not offered by their department.

41. List the names of all students who have office hours on Monday or Friday after 3 PM.

42. List the average GPA and average enrollment of all CAs in the database.

43. List the students who have written papers with faculty whose course they are taking

44. Name all the competitions that have a deadline before 12/12/2019.

45. Name all the students who are participating in the MedHacks with a team of 2 or 3 members.

46. List the faculty teaching students enrolled in Database Systems and is applying to any competition with a team more than 1.

47. List the student names, course names, and restaurants of all students who visited a restaurant instead of attending a class (visited a restaurant at the same time that they had a class).

48. List the names and home countries of all international students majoring in Computer Science.

49. List the names of faculty who have produced over 5 papers in a year.

50. For each student, list the name and birthyear of the musical artist that they listen to who has the lowest average monthly listeners.

51. For each student, list the name and birthyear of the musical artist that they listen to who has the lowest average monthly listeners, and in the same result table list the political party that the student voted for in the 2016 election.

52. Find the names of all students who listen to any of the same musical artists that the student who has the highest number of hours played in sports each week also listens to.

53. Find the names of all students who likes to any of the same musical artists that the student who has the highest number of hours played in sports each week also likes, where liking an artist is defined as having a rating greater than 5 for that artist.

54. For each student, list the student's name and age and the name and genre of the musical artist who they like the most (their rating for that artist is highest), also including their rating. If ties, list both on separate lines (duplicating the student's name and age).

55. List the name, age and major of students who like every pop musical artist in the database.

56. List the name, age and major of students who like every pop musical artist in the database, but does not like any rap artist with a rating of greater than 3.

57. List the name, age and major of all students who like at least one pop musical artist that no one else in the database likes.

58. List the name of all activities where number of faculty participating is greater than the number of students particpating, also including the total number of participating faculty and total number of participating students.

59. List the highest compensation that any student receives from an offer from any company.

60. List the name company that a student who has published at least one paper received an offer from.

61. List the name, sex and age of students who loves someone who loves another one that applied for a job at Facebook.

62. List the professor teaching the students who had the most work experience (in months) on average

63. List the name and department (primary appointment) of faculty members and the total work experience (across all companies) of students who they advise.

64. List the name of the Computer Science faculty member who has advised the student with the greatest total work experience (across all companies)

65. List the name of all Computer Science faculty members who advise students with some work experience, along with the combined total months of work experience of the students they advise.

66. List the name of all Computer Science faculty members who advise students with some work experience, along with the combined total months of work experience of the students they teach in some class

67. List the name of the Computer Science faculty member who has either advised or taught a student with some work experience in a company that focuses on software.

68. List the name of the Computer Science faculty member who has either advised or taught at least 3 distinct students with at least 6 months of work experience each in a company that focuses on finance.

69. List the name of the Computer Science faculty member who has advised the most total students with some work experience for a company that focuses on finance or accounting.

70. List the name, age and major of students who have at least 6 months work experience for a single company focused on the domain of finance in a job description containing the terms software, analyst, programmer, coder, or engineer in any part of the job description.

71. List the name, age and major of students who have at least 6 months total work experience across all companies focused on the domain of finance in a job description containing the terms software, analyst, programmer, coder, or engineer in any part of the job description.

72. List the name, age and advisor of the single student with the longest total work experience for any single company in the database focused in the domain of finance in a job description containing the terms software, analyst, programmer, coder, or engineer in any part of the job description.

73. List the name, age and advisor of the single student with the longest total work experience in the database across all companies in the database focused in the domain of finance in a job description containing the terms software, analyst, programmer, coder, or engineer in any part of the job description.

74. List the name, age and advisor of the single student with the longest total work experience in the database across all companies in the database focused in the domain of finance.

75. List the name, age and advisor of the single student with the longest total work experience in the database across all companies in the database focused in the domain of finance.

76. List the full name and age of the student who slept the greatest number of times across all courses taught by Russell Taylor. If there is a tie, report the full names of all such students.

77. List the name and student capacity of all dorms along with their total number of female student atheletes in that dorm (including if 0).

78. List the name and student capacity of all dorms along with their total number of female student atheletes in that dorm (including if 0) and the total number of computer science majors (including if 0).

79. List the name and student capacity of the dorm which has the greatest number of student athletes. If there is a tie, report data for all such dorms.

80. List the name and student capacity of the dorm which has the greatest number of female student athletes but no computer science majors. If there is a tie, report data for all such dorms.

81. List the name and student capacity of the dorm which has the highest total number of computer science majors or minors. If there is a tie, report data for all such dorms.

82. List the first name, last name, email and major of all the students who play at least 1 video game and also have more than one A+ across all the classes they are enrolled in.

83. List the name and location of the building which has rooms for the most number of faculty members.

84. List the full name, rank and sex of all faculty members who both teach at least one course in computer science and also coaches a sport.

85. List the full name, rank and sex of the faculty member who teaches the most courses in computer science

86. List the full name, rank and sex of the faculty member who advises the most female students in computer science.

87. 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.

88. Invent a complex, interesting question of your choice and write a QBE query to compute the answer. Grading of this question will be based as much on your creativity as the correctness of your solution. Your query may be the same as used in question 87, or different.