CS-UY 3083, Introduction to Database Systems, Fall 2025 Prof Dey

Homework #4

Instructions: You may work alone or with a group of up to 3 people. Hand in your solutions via Gradescope as a single pdf file. **Follow the Gradescope instructions to mark your solution to each problem or subproblem as indicated in the outline; otherwise the graders will have trouble finding them and will apply a small penalty to your score.** See student workflow section in GradeScope help to learn how to do this. If you're working with a group, use GradeScope's group submission feature to indicate all members of your group.
Note: Execute each of the queries you write. Your answers should include the SQL query and the result of executing the query on the given data. (You can screenshot your GUI displays or print it to a pdf file.) Your queries should work on any data that satisfies the given constraints, not only on the sample data provided.

1. For this problem, Consider the **takes** and **student** tables from the university database that you imported in HW 0. **Execute each of the queries you write.** Your answers should include the SQL query and the result of executing the query on the given data. (You can screenshot your GUI displays or print it to a pdf file.)

   INCLUDE THE NAMES OF EVERYONE IN YOUR GROUP AS A COMMENT AT THE START OF THE QUERY.

   a. Find the ID, course_id of students who got 'A' grades in any course(s). (That is, find the ID of each student who got any A's and the course_ids of courses in which they got 'A's; the result should be a table with attributes 'id' and 'course_id').
   b. Find the names and IDs of students who got a 'B' or better in CS-101.
   c. Find the ID, course_id of each student who has NULL grade in any course
   d. Find the name of each student who took a (section of a) course given in Watson building

2. Find the ID, name, and total number of credits each student has passed (grades other than F or NULL), *based on the information in the takes table (not on the student.tot_cred attribute)*. The result should be a relation with attributes ID, name, and creditsPassed and should only include students who have passed at least 10 credits.

3.
   a. Create a table gradepoint(grade,point) to associate letter grades with points and fill it with the appropriate values ('A', 4.0), ('A-' ,3.7), etc.
   b. Use it to create a view StudentGPA(id, dept, gpa) showing each student's ID, department name, and grade point average.

Hint/Note: The following query would work **if** all courses had the same number of credits and **if** we didn't care about which department each student is in:

```
SELECT id, avg( points )AS gpa
FROM takes NATURAL JOIN gradepoint
GROUP BY id
```

You need to modify this so it's weighting the average by the number of credits per course (Sum (credits* points)/sum(credits)) and so that you can also include each student's department.

4. Find the ID and name of the Comp Sci student with the highest GPA (among Comp Sci students) (This is referring to the view you just created in problem 3)

5. Find the course id of each course that has been offered twice in one academic year: Fall of year y and Spring of year y+1

6. This problem will lead you toward writing a query to Find IDs and names of instructors who have taught every Bio course (i.e. courses with title LIKE (Bio%)). Do this with the following steps and show your each step in the work you hand in:

   a. Write a predicate that returns true if and only if instructor 12345 taught all Bio courses.

   b. Use a modified version of (B) with a correlated attribute instead of 12345 as part of a query to find the IDs and names of instructors who have taught every Bio course. (Hint: you might need to use a nested subquery)

7. Find the number of students that have taken each class but only include classes that were taken by at least 2 people. In your result, include the course_id, sec_id, semester and year (Hint: Group By and Having keywords might be helpful)