Homework 2 (SQL 1&2)

PLEASE READ CAREFULLY BEFORE STARTING YOUR WORK

1. *Before starting the HW, if you have not done so after the updates/deletes, run the scripts DDL & LargeRelationsInsertFileUpdated.sql located on Canvas in Files.*
2. *For all problems, use the University Schema located on Canvas in the same directory.*
3. *Start with problems 1-5.*
   After you are done with those, rerun scripts DDL & LargeRelationsInsertFileUpdated.sql to get your database in the original state.
4. *HW is 90 points plus there is a bonus problem for additional points, if desired, for additional points* ☺

Submission rules:

- Make sure you use the setup we use in Class, MySQL version we specified in Lecture 1 connected to Amazon; points will be deducted if any other setup is used.
- Please upload your answers to Canvas. Only typed answers will be accepted for this HW.
- Your answers to each problem need to be in the submitted in TWO files, details are below. Points will be deducted if the format differs or we get different number of files.
- SQL File: a file with your SQL queries. We will run this file for grading
  a. .sql format
  b. Name of the file: hw2_your_lastname_firstname_problem#.sql. It includes:
     i. SQL problem and its full formulation (formulation commented out)
     ii. Your query
     iii. the number of rows returned (or 0), commented out
- CSV file: A file with your queries' results (a separate folder for each set or a separate file for each problem, marked by problem #)
  a. Name of the file hw2_result_sets_your_lastname_firstname_problem#.csv
  b. Include Result sets for a problem. If no result set, please specify a problem # and 0 results

Assignment: Write and run a SQL query for each problem.

1. Create a new course "CS-001", titled "Weekly Seminar", with 3 credits. **(4 points)**
2. Create a section of the course from Problem 1 in Fall 2017, with sec_id of 1, and with the location of this section not yet specified. **(4 points)**

3. Enroll every student in the "Comp. Sci." department in the section from Problem 2. **(4 points)**

4. Delete enrollment in the section from Problem 3 where the student's ID is 1402. **(4 points)**

5. a) Delete the course "CS-001". (**2 points**)
b) Note that deleting from the table in the problem 5a caused automatic cascading deletes in some other tables in order to preserve referential integrity (foreign key constraints).
What tables are affected and how exactly? (**6 points)**

Rerun script LargeRelationsInsertFileUpdated.sql to get your database in the original state.

6. Find the name and ID of those students in the "Accounting" department who are advised by an instructor in the "Physics" department. **(7 points)**

7. Find the names of those departments whose budget is higher than that of "Geology" department. List them in alphabetic order. **(7 points)**

8. For each student who has retaken a course at least twice (i.e., the student has taken the course at least three times), find the course ID and the student's ID. Display your results in order of course ID, and do not display duplicate rows. **(7 points)**

9. Find the ID and title of each course in "Psychology" department that has had at least one section with afternoon hours (i.e., ends at or after 12:00pm). Eliminate duplicates if any.
(7 points)

10. Find the number of students in each section. The result columns should appear in the order "course_id, sec_id, year, semester, num". You do not need to output sections with no students.
(7 points)

11. Rewrite the query

```
select *
from section natural join classroom
```

without using a *natural join* but instead using an *inner join* with a *using* condition. **(7 points)**

12. Find out if there are students who have never taken a course at the university. Do this using SQL query with no sub-queries and no set operations (Hint: use an outer join). **(7 points)**

13. Create a view *tot_credits* (*year, num_credits*) that shows the total number of credits taken by all students in each year. Use that view to output total credits by year. **(7 points)**

14. a) Find the ID and name of each instructor who has never given an A grade in any course she or he has taught. (Instructors who have never taught a course trivially satisfy this condition.)
(10 points)
b) BONUS points: Rewrite the query from 14.a) to only include instructors who have given at least one (other than A) non-null grade in some course. **(2 points)**