Instructions

Before starting the test
If you have not done so already, run script LargeRelationsInsertFileUpdated.sql located on Canvas in Files in CreateDB directory.

For all problems
1) Use University Schema located on Canvas
2) Please read each problem CAREFULLY (recommended: more that once!)
3) Write ONE SQL statement for each problem below. Points will be deducted if more than one SQL query, even if results are correct.

Grading
Each problem (or sub-problem, if exists, e.g., 5A, 5B, etc.) is 20 points
You will get 100% for solving problems #1-7.
If you solve the bonus problem (#8), you will get an additional 5 bonus points ☺

To submit your answers:

Step 1 Create a "final_<your_firstname_lastname>.sql" file for your answers. Submit the file to canvas. Include the following:

a. Problem/sub-problem number
b. Problem formulation, commented out
c. Your answer (=SQL statement)
d. The number of rows effected.

Step 2. Create final_<your_firstname_lastname_problem_number>. csv files corresponding to each problem

- Do not include rows containing NULLs ONLY.
- Do NOT repeat the problem formulation in cvs files.
- Submit the files to Canvas.
- DO NOT ZIP the files.

Step 3. After you submitted the files to Canvas, you must let your breakout room TA know, and return to the Main room

Additional grading rubric details:
1. Missing CSV files: ***-5 points*** **for each CSV file missing**.
2. Missing SQL file: you will receive ***0 points for the test***
3. Partial SQL file: ***we will only grade questions with SQL queries***
3. Missing number-of-row-return comments: ***-2 points*** **for each** ***comment*** **missing**.
4. The result in the CSV file does not correspond to the number-of-row-return comment: ***-16 points for each.***
5. A Zip file is uploaded vs individual csv files: ***-1 point***
6. Using multiple queries for 1 question: ***-8 points each***

Problem 1.

Find all students from Math department who got at least one A+ in 2019. Output student ID and student name.

Problem 2.
Find all students from Math department who got more than one A+ in 2019. Output student ID, student name, and the number of A+ courses.

Problem 3.
Find all instructors (names and IDs) who taught at least one course in the 'Taylor' building in the Fall of 2019.

Problem 4.
Find student(s) with the maximum total credit. Output student name, ID, and department; order by department name.

Problem 5.
5A. Create the table **student2** with the schema:
student2((ID varchar(5), name varchar(20), dept_name varchar(20)).
Make ID a primary key. Enforce that name cannot be NULL.

5B.
Insert student records from the table **student** into the table **student2.** Include only records for students with the total credit > 100.

5C.
'Math', 'Statistics' and 'Comp. Sci.' departments merged into a ‘Data Science', department. Update the table **student2** to reflect that.

Problem 6.
For each course, calculate the number of times the students took that course. Output course id and the number of times; output 0, if nobody took that course. If a student took some course more than once, include all those times. Order the results from the largest number of times to the smallest.

Problem 7.
Calculate the number of courses each instructor taught per each year, each semester. Also include in the (same) output the instructor ID and the following count summaries:

- by instructor ID by semester
- by instructor ID by year
- by instructor ID by totals
- the total number of courses taught by all instructors in all semesters of all years

Bonus Problem (#8)
For each department, find the student(s) with the maximum total credit of all students from that department. Output student name, ID, department, and total credit. Order by department name.
Note: make sure that if more than one student in a department has a maximum total credit, all of them are included in the result set.