Complete all questions as scripts and turn in as one .sql file. Use the following format in your script

--========== Question 1
Your sql statement goes here for question 1
--========== Question 2
Your sql statement goes here for question 2

Spend some time reviewing the last page of this exam – it describes a system by which instructors are related to courses and courses have TA's. Instructors create exams and exams have questions. Write the following queries:

1. Return the first 10 Instructor and Course they teach and order the list by the Instructor name and then the Course. Return the following columns: Instructor.Name, Course.Name and name them: InstructorName, CourseName. Do not use OFFSET

2. Write the query that will return all the Instructors that are not assigned to a course. Do this without using a JOIN statement of any sort (meaning, do not use an implicit or explicit join) (hint: we did this in the beginning of the semester)

3. Write the **explicit** version to the above query

4. Write a UNION query that will return all the Instructor's name, ExamID, and the total number of points the exam is worth (this is calculated from the ExamQuestions) that was given between 1/1/2018 and 12/31/2018 and do not add up to 100 points. Include a field called "status" that will show "Under" if the points are less than 100 and "Over" if the points are more than 100. Also, disregard exams that don't have any questions.

5. Write the above query using a CASE statement

6. Return a list of TA's that are listed in more than 2 courses. Sort the list by the TA's name

7. Return a list of the number of exams by state and the average points for the exams. Order by state

8. Given the fact that I want to see 20 Instructors at a time write the code that will return the 3rd set of data (records 41-60). Return all columns and sort the list by state and then the Instructor's name.

9. The schema shows that an Instructor assigns a Rate of pay to their TA's. It also show what the Instructor will pay the TA during an exam (TARate in the Exams table). Given this information return a list of TA's and Exams that they were paid at a different rate then what the Instructor assigned them in the InstructorTAPayRate table. Return the ExamID, TA's name, their contract rate (Rate), and the rate they were given for the Exam (TARate). Sort the list by the TA's name

10. Return a list of Instructors and their TA's. If they don't have a TA then the TA's name should be "TA NEEDED". Note: an Instructor's TAs are defined by Instructors and TA's having the same course ID

11. Return a list of Instructor names and the TA's that proctored the Instructor's exam BUT are not their TA. In other words, show the list of TA's that proctored exams that they were not the TA for. Return the Instructor's name, the TA's name, and the ExamID. Sort the list by the TA's name

CourseDataDB Schema