DBMS Lab 2019-20 Spring Semester
Lab Day 6 (February 11, 2020) – 50 Marks

[Penalty for plagiarism/copying: You will be awarded 0 for all the problems for the lab day you were involved in plagiarism/copying and an additional 5 marks will be deducted out of the total of 40 in Lab. All persons involved will be awarded the same penalty irrespective of who has copied from whom. Decision of the lab teachers is final in this respect.]

1. Consider that we have the following tables in a database (primary keys are underlined):

   Student (roll_no int not null, name varchar(30) not null, cgpa decimal(7,2) not null with default 0.00, credits_cleared int not null with default 0)

   Student_course (roll_no int not null, course_cd char(2) not null, grade_point int not null with default 0.00)

   Course (course_cd char(2) not null, course_name not null, credits int not null)

   Prerequisite (course_cd char(2) not null, prereq_course_cd char(2) not null)

a. Write a trigger on the Student_course table so that whenever a row is inserted or value of the column grade_point is updated, or a row is deleted, the value for credits_cleared of the corresponding student will be updated using the Course table. Note that credits cleared will be considered to be non-zero if the grade point obtained is greater than or equal to 5. **[10]**

b. Write a trigger on the Course table for update of the column credits so that whenever its value is changed, for all the students in the Student table, who have taken that course, the credits_cleared column will be appropriately updated. Note that credits cleared will be considered to be non-zero if the grade point obtained is greater than or equal to 5. **[10]**

2. Consider the same tables used in Problem 1.
   Write a database procedure/function that will take a roll_no as input and update the value of the cgpa column using the Course table and the Student_course table. The value of cgpa should also be returned as output. **[20]**

3. Consider the same tables used in Problem 1. Assume that for any course that does not have a pre-requisite, will not exist in the Prerequisite table

   ODD PC Nos.
   Write a recursive query which will take a course_cd (say AB) as a hardcoded input (i.e., somewhere in your SELECT, there will be a clause like “AND course_cd = AB” or “WHERE course_cd = AB”. This input is not in the sense of an input in a function/procedure) and return the total number of credits one can complete if AB is completed. If AB is not the prerequisite for any other course, it should return 0. Note, you need to make use of the Course table and the Prerequisite table only. **[10]**

   EVEN PC Nos.
   Write a recursive query which will take a course_cd (say AB) as a hardcoded input (i.e., somewhere in your SELECT, there will be a clause like “AND course_cd = AB” or “WHERE course_cd = AB”. This input is not in the sense of an input in a function/procedure) and return the total number of credits one needs to complete in order to complete AB. If AB does not have a prerequisite, it should return 0. Note, you need to make use of the Course table and the Prerequisite table only. **[10]**

Through Moodle, submit a file containing all your SQL and PL/SQL statements. (Name it as Lab6_<Roll_no>.sql).