HW 5 – CS 340
Submit the written HW 5 assignment and exported SQL code to
Canvas

1. Give the SQL code to create the table

   CollegeStats (cName: VARCHAR(20), appCount: int, minGPA: decimal(3,2), maxGPA: decimal(3,2).

This table will contain for each college in the College table the number of applications, minimum GPA of all students who applied and the maximum GPA of all students that applied to that college.

2. Give the SQL code to insert the cNames of the colleges from the College table into CollegeStat along with the appCount, minGPA and maxGPA.

3. Write a stored procedure called updateCollegeStats(cName) that takes as a parameter a college name and then updates the count of total applicants, minimum GPA and the maximum GPA of all students that applied to the college cName. Give the code and resulting table.

4. Write a stored procedure called updateCollegeStatsAll to update the entire the entire CollegeStats table with the count of total applicants, minimum GPA and the maximum GPA for all of the colleges in the College table. Give the code and resulting table.

5. Write a trigger called updateApplyStats for the Apply table. This trigger will update CollegeStats when a new record is inserted into the Apply table.

   a) Begin by incrementing the appCount in CollegeStats that corresponds to the college (cName) in the new Apply record. Test by inserting the record (543, OSU, CS, Y) into Apply. Give the resulting tables for Apply and CollegeStats and code for the trigger.

   b) After part a) is working correctly, add code to the trigger to update the minGPA and maxGPA in CollegeStats if the GPA of the new Apply record changes either of these values. Test by inserting the record (567, U of O, CS, N) into Apply. Give the resulting tables for Apply and CollegeStats and code for the trigger.

6. Write a stored function called studentRank that classifies students according to their GPA. The input to the function is the GPA: real and the output is a VARCHAR(10) with the following values:
   GOOD if GPA > 3.5, AVERAGE if 3.5 >= GPA >= 3.0, POOR if GPA < 3.0.
Give the code for the function and the results of the following query .

```
SELECT sName, GPA, studentRank(GPA)
FROM Student
```