

Due Date: July 18th, 2017

The Database system you created in part-1 must satisfy the following Requirements:

You are to create a Database Management System which must have the following **tables** and **fields**:

Student:

Id
First Name
M.I.
Last name
ID "W#"
DOB (MM/DD/YY)
Major
Degree Program
Graduate/Undergraduate

Course:

Id
Abbreviation
Name
Number
Description
Credit Hours
Semester_Id

Semester:

Id
Abbreviation
Name

Courses Taken:

Id
Student_Id
Course_Id
Grade("A", "B", "C", "D", "F")

Requirements for Project Part 2 "Student's Transcript":

- Your DBMS must be able to run the following SQL-ish queries (in addition to the queries from Part 1):
 - <result> = SELECT <table1-name> JOIN <table2-name> ON <column1-name> = <column2-name>
(Store the resulting set, <result>, to use for additional queries)
 - SELECT COUNT FROM <table-name> WHERE <column-name> = <value>

- List all the courses a student has taken.
- Compute the current course hours of the last semester for a student.
- Compute what student has an "A" in CMPS 339 for SUMMER '17.
- (Bonus) Compute the cumulative GPA using queries alone. You may execute queries on stored result sets, which must be done like any of the other queries you can execute. This is not an easy bonus; you are expected to put in the work.

Important Notes:

- You will give a demo of the database and the queries.
 - A team member demo should suffice for the whole group.
 - I am available on campus on **Tuesdays** so the team member demoing the project should contact the Professor to assign a time slot for the project demonstration.
 - Students must inform the Professor immediately who the team members are per group.
 - Be able to explain both the concept and code for your approach to this program.
- Part 2 builds on the application that you have been building all semester. If you are still working on the requirements from Part 1, complete these before beginning Part 2.
- Do not procrastinate and remember that **Google** is your friend. This is a group project; you are expected to work as a team.

Important Suggestions:

- Design your new system changes as fully as you can before implementing changes.
- If you are using source control, each team member should create a feature branch off the master branch, and merge the changes in after the feature is complete. This will keep your master branch as functional as possible.
- If you opted to use Entity Framework, setting up the database should now be the easiest part of the assignment. Concentrate on building a sturdy query parser to translate user input into Entity Framework commands.