# Assignment 2

## Submission

***Your submission will consist of ONE sql file as a complete creation script containing all requirements. All files must contain your names, student id’s in an appropriate header.***

Only one submission per group please.

## Group Work

This assignment is to be completed in groups of 3 or 4. Please only one submission per group.

It is suggested that you **ALL do it individually** and then meet to compare answers. Those not doing the work may be barred from your group resulting in a zero and incomplete on the assignment.

**VERY IMPORTANT:**

Being part of a group is the same as being a part of a team for these assignments. When you submitted your work as part of a group you are saying that:

* you understood what was submitted and that you fully participated with ALL the group members.
* It does not mean letting others do your work for you.
* It does not mean watching the others do the work.
* For your full participation, you get a mark equal to all the others in the group.
* If on the test, which is very much like the assignment, you cannot answer it strongly indicates that you didn’t participate and understand the assignment but depended on others for the mark you received. That is very much like submitting their work and claiming it is your work.

## Business Use Case Scenario

* ***Seneca College*** contains many departments.
* Each department has many programs but every program belongs to only one department.
* Every department has many professors.
* A professor can work for only one department.
* A program has many courses.
* A course can be a requirement of many programs and the term in which students typically take it (i.e. term 4 is second term in year 2).
* A professor can teach many courses. A course can also be taught with many professors.
* A program has many students studying in that program.
* A student can study multiple programs. However, a student has to register in least in one program.
* A student may take many courses.
* A student, however, can be off from school and do not take any courses.
* A course can be taken by many students.
* A new course may not be available yet so the course may not be taken by any student.
* A student may have a program advisor. Having an advisor is optional for students so some students may not have any advisor.
* An advisor can have one or more students assigned to them.
* Courses are offered in no, one or several section(s) each term
* Course sections are only offered once, each term the section will receive a NEW code.
* Each section is taught by a single professors (required)

## Tasks

Each group will produce a single SQL file that:

* creates all tables, fields, constraints and relationships in the database
* contains insertion scripts for inserting their current program of study (i.e. all the courses you have taken since starting Seneca until you graduate (you should have 4 student records when you are done).  
  Do some research and find all the professors teaching the courses you are taking this term and add this information as well (hint: in the fall DBS301 is taught by Clint MacDonald, Ron Tarr, and Robert Stewart)   
  For courses you have not yet taken, you may assign an unknown professor (i.e. add a professor named “unknown”). It is okay to provide fake data for personal information for students and professors.
* Create individual views that will produce:
  1. A student transcript that contains all the courses taken and the marks obtained for each student
  2. A Class list for students enrolled in a particular course section
  3. The average mark for each section and the professor whom instructed it
  4. An output of all the courses required for each program sorted by program name and the term required.
  5. A term specific list of which professors are teaching which courses and how many students are enrolled in each course section. Remember that professors can teach more than one section of the same course each term.

Note:

* All tables and views should use the prefix “a2”. Example: “a2Properties”
* Use the documented style guide for all naming, SQL scripts and ERD formatting.
* Use advanced joins where required, simple joins (ANSI-89) are not acceptable.
* Create meaningful names for all objects, including constraints (see style guide).

## Marking Schema

Marks will be given in a top down format. i.e. you start with a perfect mark and marks are removed for incorrect, incomplete and insensible work. Bonus marks will be given for work that goes beyond the scope of the requirements, but only for work that is exceptional, creative and beneficial to the client.