

# Database Management Systems

## Data Modeling Assignment

### Introduction

Consider the problem statement below. Create a UML class diagram using any UML tool. If your tool supports online sharing, submit the link to the shared diagram. You can also submit the diagram as an image or PDF file. Create a class diagram following the techniques described in class. Provide a naive class diagram and a final class diagram. For each of the requirements below, provide a paragraph or two describing your thoughts

(10pts) List nouns that are candidate classes or attributes

(10pts) List verbs as candidate relations between classes

(10pts) Generalization/specialization (inheritance, if applicable, explain) - show parts of your diagram that specifically illustrates the use of inheritance

(10pts) Associations, aggregation and/or composition, e.g., empty or filled in diamonds (1 to \* or 1 to 1..\*, if applicable, explain) - capture any lifecycle dependencies between classes using aggregation or composition.

(20pts) Classes vs. attributes analysis - if you are not familiar with a particular domain, e.g., a particular industry such as education, or nuclear energy, or telecommunications, or astrophysics, you might be unfamiliar with the vocabulary and so any noun might be a potential class. You might create a 'naive' class diagram that makes no distinction between classes and attributes. As you research the domain and learn more about it you realize the relationships between nouns where some might just be describing other nouns or some are compositions of other nouns. Create a naive class diagram and diagram your process of identifying certain classes as actually being attributes of other classes. Or vice versa, where some attributes might actually be better modeled as classes. Document how the class diagram evolved from a naive first approximation to the final result. Explain your decisions and support them with relevant portions of the class diagram. Show your final class diagram as one single diagram.

(5pts) Correct data types, e.g., Date, String, Integer, List, Array, Enumeration, etc.

(10pts) Cardinality - for every single association, show the number of instances participating in a relation

(10pts) Remove any inadequate or redundant relationships, entities or attributes (if applicable, explain) - if you identify redundant associations, entities, or attributes, explain how/why you removed it. For instance, the problem statement might have irrelevant information that you might need to ignore. Also, the text might describe contradictory or ambiguous descriptions. Finally the statement might use different terms to refer to the same thing. Make sure you make a compelling argument for your decisions to ignore a particular noun or verb as irrelevant or redundant or an overloaded term.

(10pts) Reify (if applicable, explain) - if you have association classes or other UML artifacts that don't readily map to relational schemas, explain how you transformed it to a concrete relational schema representation

(5pts) Prose - in proper english, for each of the above, provide a couple of paragraphs that support your decisions and train of thought.

# Problem Statement

Consider an application such as Blackboard, an online learning management system (LMS) that provides tools for faculty and students to create and share a learning experience. Faculty can author courses that contain learning modules broken up into lessons. Modules and lessons can be rearranged into a different order based on the calendar schedule. An LMS should provide a set of rich content widgets to build each of the various topics in a particular lesson. Widgets come in variety of types: youtube videos, slides, text documents, raw HTML, evaluations, and many more. Evaluation widgets can be a simple essay assignment, a submission assignment, or an exam. Exams are used to evaluate the student's progress as they answer the various types of questions in an exam, such as essay questions, multiple choice questions, fill in the blank questions, and many more types of questions. Based on the popularity of courses, the registrar's office creates several sections for a course for a given semester. There are 5 types of semesters: fall, spring, full summer, summer 1 and summer 2. Some of the less popular courses are only taught in particular semester in a given academic year. Students enroll in different sections for a course. When registering for a course, students can see a section's seat capacity, and the faculty assigned to teach the section for a particular course. Undergrad students tend to enroll for many more courses than graduate students. The registrar's office keeps track of student progress in the enrollment, such as the final grade, letter grade, and student feedback. From time to time, the University asks everyone to verify their personal profile information such as username, password, first name, last name, emails, phones, and addresses. Users can provide multiple emails, addresses and phones. Faculty additionally need to update their benefits, tenure status, parking, and bank account info. Students have to verify their financial aid info, work-study, and scholarship. Students with scholarships are always keeping an eye on their gpa that it does not drop below a certain threshold. Students can see their final grades for a particular course they were enrolled in. Grades are neatly broken by the various assessments such as assignments and exams. Even down to the points they lost on a particular question based on a rubric that keeps track on how much each question was worth on an exam or how much a particular part of an assignment was worth. Students often go to office hours to review an evaluation with their instructor or teaching assistant. They go question by question reviewing their answers and where they might have gone wrong.