Database hw 1

Sunhao Liu 001050520

Uml link: https://www.lucidchart.com/invitations/accept/d49c3dab-69ca-4baf-9c55-a011dc2b6b88

1. Nouns:

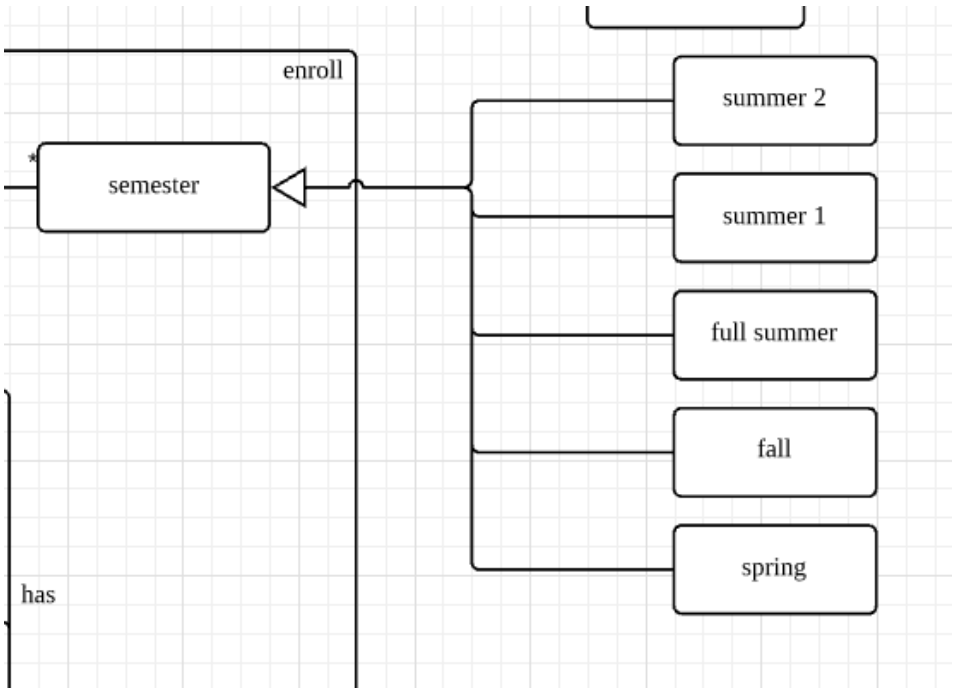
   Faculty, courses, modules lessons, order, calendar schedule, widget, topics, exams, progress, questions, types, office, sections, semester, students, seat capacity, faculty assigned, personal information, scholarships, GPA, grade, assignments, points, answers, level.

2. Verbs:

   Author, contain, rearrange, build, evaluate, answer, has, enroll, register, see, track, verify, update, break, review.

3. Generation means one “is a” relationship between two classes. In this statement, the relationship between different types of semester and the class semester is generation.

   Here is part of my class diagram.



4. composition:

the relationship between lesson and module

the relationship between module and course

the relationship between course and section

the relationship between section_grade and progress

the relationship between question and point

Lesson is generated from module and module is generated from course, they cannot exist without their parents, section is divided from course, so section cannot be created without course. Progress(grade) should be related to section, without section, the progress will not be meaning. Point is involved in question and cannot exist without question. So all the relationship mentioned above is composition.

Aggregation:

the relationship between exam and question

the relationship between question and student_question

the relationship between GPA and exam

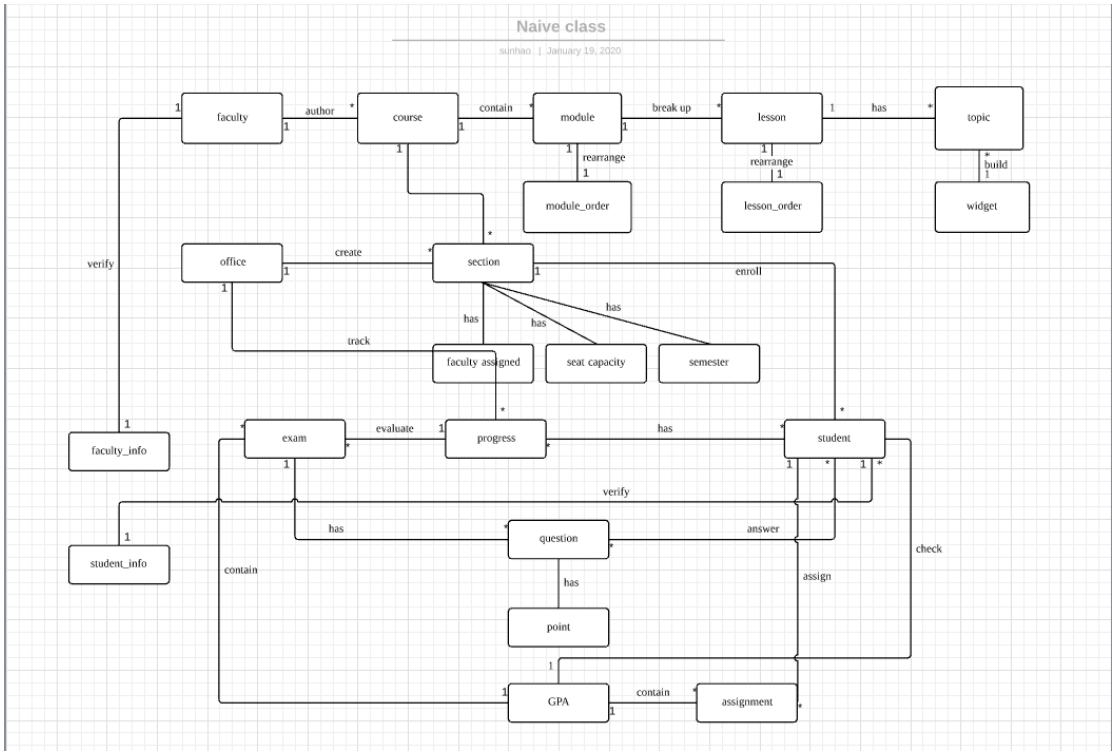
the relationship between GPA and assignment

the relationship between section and semester

the relationship between section and student

Exam consists of many questions, but question can be alive without exam, such as exist in textbook, so is the relationship between question and student_question. Exam and assignment is the component of GPA, but they can be alive without GPA. Semester and student are component of section, but they can both be alive without section. So all the relationship mentioned above is aggregation.

5.Naïve class diagram_version00:



After consideration, widget is better to become the attribute of the topic. Faculty assigned and seat capacity are better to exist as the attribute of the section since they do not have their own attributes and they are not very important component of the whole diagram.

The relationship between student_info and student is 1 to 1, so we need to merge information into student. So is the relationship between faculty_info and faculty, we need to do the same operation. (reify)

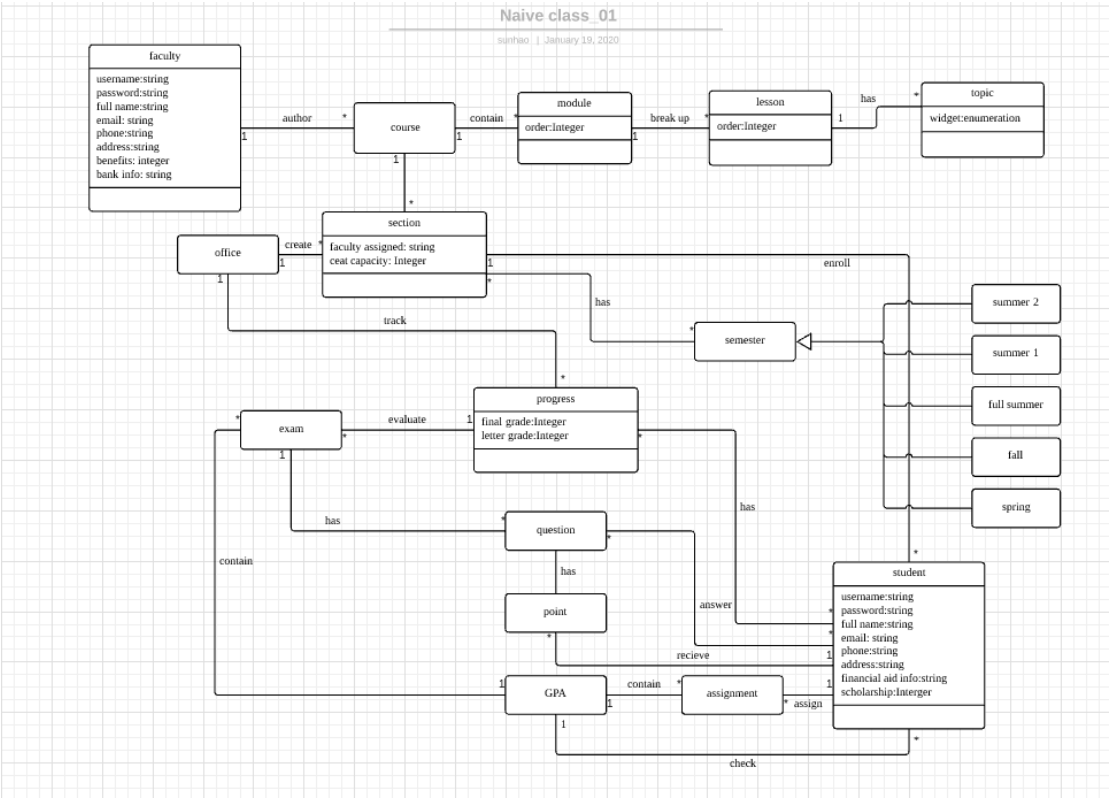
The relationship between module order and module is 1 to 1, so I merge module order into module. So is the relationship between lesson order and lesson and I made the same operation. (reify)

I have added two attributes to the class progress, since I think student feedback is ambiguous to progress, I choose to delete it.

Besides, I have added five types of semester and regard them as class since I hope to show the generalization relationship.

I have added some data types in classes.

Naive class diagram_version01:



6.

Inadequate relationship:

(1) the class diagram did not show the relationship between faculty and office, but actually office can decide to assign faculty so add decide_system to assign faculty.

(2) the class diagram did not show the relationship between section and progress, the accurate relationship should be" students receives progresses for each section" , so add section_grade class.

(3)the relationship between student and question is * to *, since one question can be answered by many students and one student can answer many questions. Add

one new class student_question to solve this problem.

Redundant relationship:

(1) student can answer questions and each question has its point, so the relation receive between student and point is redundant.

(2) Student can see section_grade evaluated by exam, which could decide GPA, to check their GPA, and student can also see assignment, which could also decide GPA, so the check relationship between student and GPA is redundant.

(3) Topic is irrelevant to the whole class diagram, so delete it .

Reify:

(1) add class: decide_system

(2) add class: section_grade

(3) add class: student_question

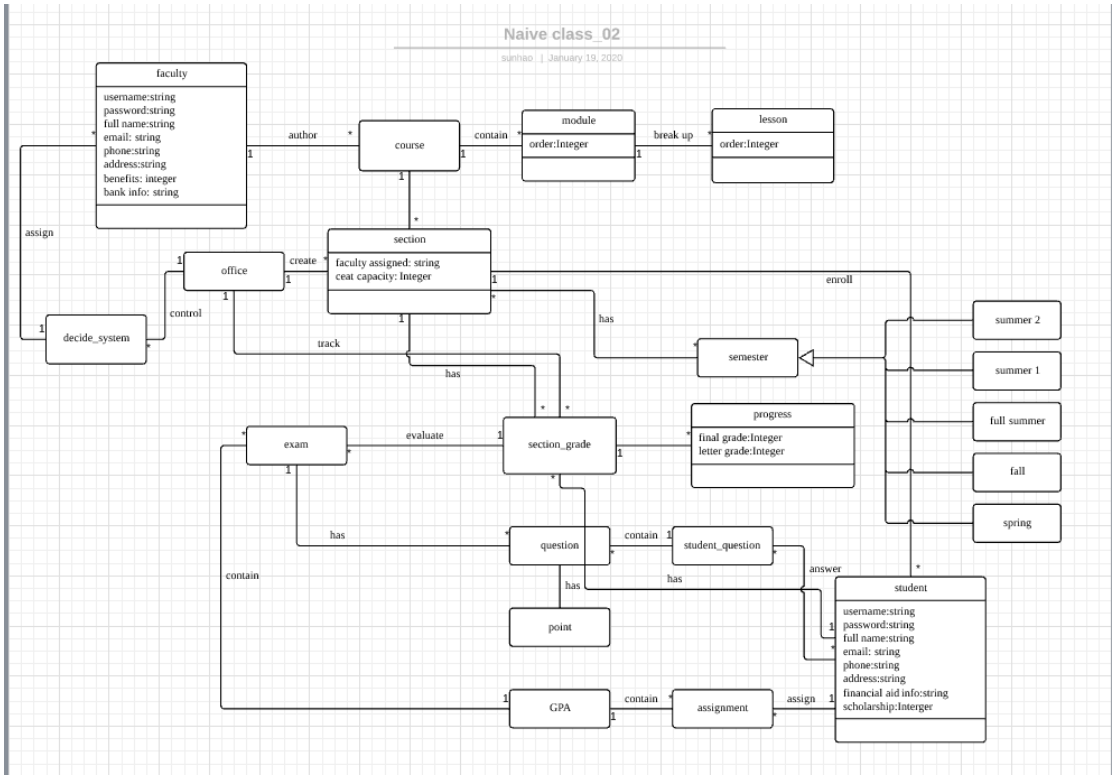
(4) merge module_order into module

(5) merge lesson_order into lesson

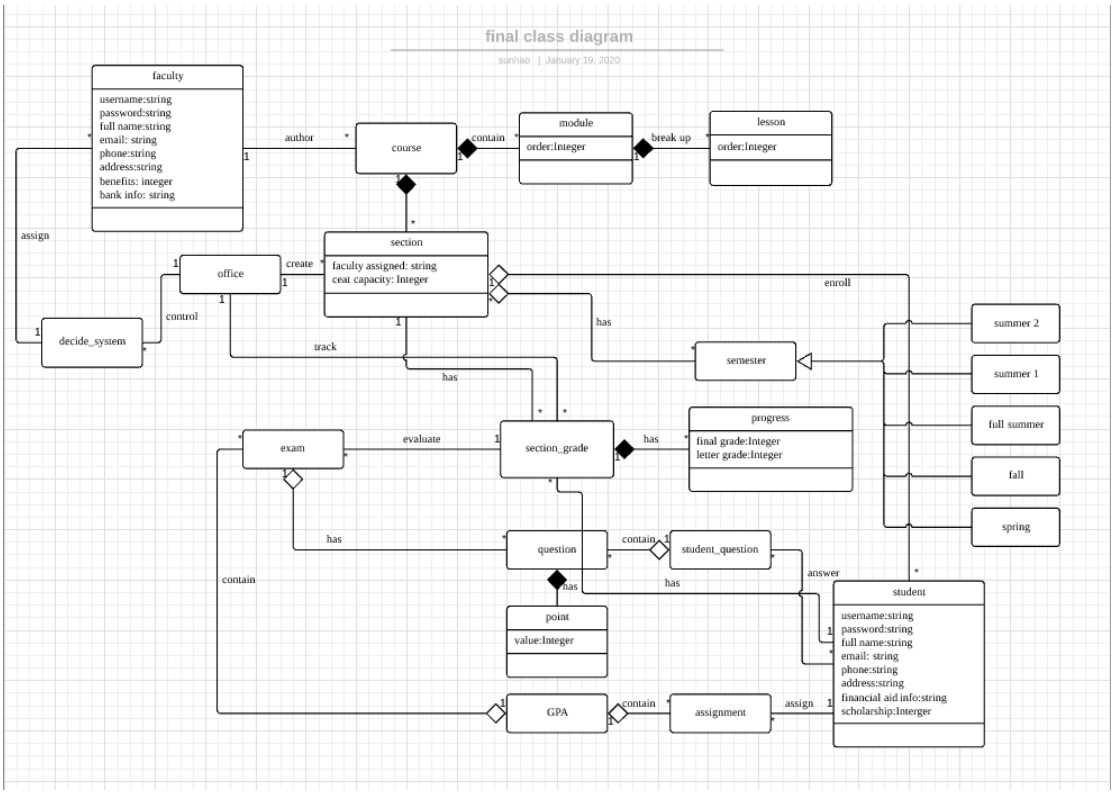
(6) merge faculty_info into faculty

(7) merge student_info into student

Here is the class diagram after the operation mentioned above,



7.adding the composition and aggregation relationship, here is the final class diagram:


final class diagram
sunhao | January 19, 2020
faculty
username:string
password:string
full name:string
email: string
phone:string
address:string
benefits: integer
bank info: string
author
course
contain
module
order:Integer
break up
lesson
order:Integer
assign
section
faculty assigned: string
ceat capacity: Integer
office
create
control
decide_system
track
has
enroll
semester
summer 2
summer 1
full summer
fall
spring
progress
final grade:Integer
letter grade:Integer
exam
evaluate
section_grade
has
question
contain
student_question
answer
contain
has
has
point
value:Integer
student
username:string
password:string
full name:string
email: string
phone:string
address:string
financial aid info:string
scholarship:Interger
GPA
contain
assignment
assign