COMP2222 Database Systems
Spring 2023, Project Part#2, same teams of Part#1
Due: May 17, 2023 23:50

IŞIK UNIVERSITY
COMPUTER SCIENCE AND ENGINEERING

Part II DB Implementation

Part I solution files have been attached to this document. Both files are prepared in yEd graph editor. Please DO NOT change attribute and/or relation names.

Given the solution of Part I, steps:

a. Implement the relational schema on MySQL.
b. Once the implementation is over, populate every table with a reasonable number of tuples (at least 5 tuples for tables representing entities and 10 tuples for tables representing relationships).
c. Write SQL queries retrieving the information specified below.

Deliverables: SQL statements. Specifically, submit a text file including:

- "CREATE TABLE" queries for part (a),
- "INSERT INTO" queries for part (b), and,
- "SELECT FROM" queries for part (c).

Queries (capital letters represent variables, replace variables with an appropriate value from the domain):

Be careful with section and course distinction!

1. List the students of a department D in form [studentId, student name, e-mail(s), grad or undergrad]
2. List the advisors of students of a department D in form [studentId, student name, advisor name]
3. List the instructors of a department D.
4. List the courses of an instructor I in year Y, semester S in form [course Code, coursename, ects]
5. List the instructors who are not offering any course in year Y, semester S.
6. List the students taking course C in a given year Y and semester S, such as students taking COMP2222 in Spring 2022.
7. List the students taking a particular section S. Note that, particular section means that all the compound key fields of section is fixed, course C, instructor I, year Y, semester SE, section id ID like (students taking COMP2222.1 of Emine Ekin in Spring 2022)
8. Given a student S, list all courses in his/her curriculum in form [course code, course name, ects]
9. Given a student S, semester SE, year Y, display timetable in the form [coursecode, section id, day, hour] like
   COMP2222 1 Th 1
10. Given a student S, display his/her grade report in form [CourseCode, year, semester, grade] including the courses s/he has no grades yet, like
    COMP1111 2020 Fall BB
    COMP1112 2021 Spring CB
    COMP2222 2022 Spring
    COMP3401
    …
11. Display all grades of a course C in year Y semester S.

"No one in the brief history of the computing has ever written a piece of perfect software. It is unlikely that you'll be the first." Andy Hunt

12. Display all scores of a student S of a course C in the form [examname, score] like

Midterm1	40
Midterm2	55
Project1	81

Note that, if the student has repeated the course, there will be more than one Midterm1, 2 etc in the list as we haven't entered year, semester info.

13. Display all points of a certain exam E course C offered in a particular year Y, and semester S in the form [sssn, qNo, pointesEarned].
For instance, MATH111 in Fall 2022, Midterm 1

Student1	Q1	5
Student1	Q2	10
Student1	Q3	10
Student1	Q4	20
Student1	Q5	20
Student2	Q1	15
Student2	Q2	7
Student2	Q3	13
Student2	Q4	20
Student2	Q5	0

14. Given a section S, create free hours report for students registered in section S (difficult!).
15. List the projects controlled by a department D.
16. List all people working in a project P.
17. Assume for each hour of working in a project, instructors will be paid 100$ extra. Display extra payments of instructors working in a project P in form [instructor ssn, instructor name, extra payment]
18. Display overall extra payment of an instructor I in a given semester S and year Y (project working hours*100 + (total teaching hours in S,Y -10)*50 + (supervising gradstudent)*25).
19. Calculate average base salary of instructors of each department.

SOLUTION OF PART I

Assumptions /Explanations:

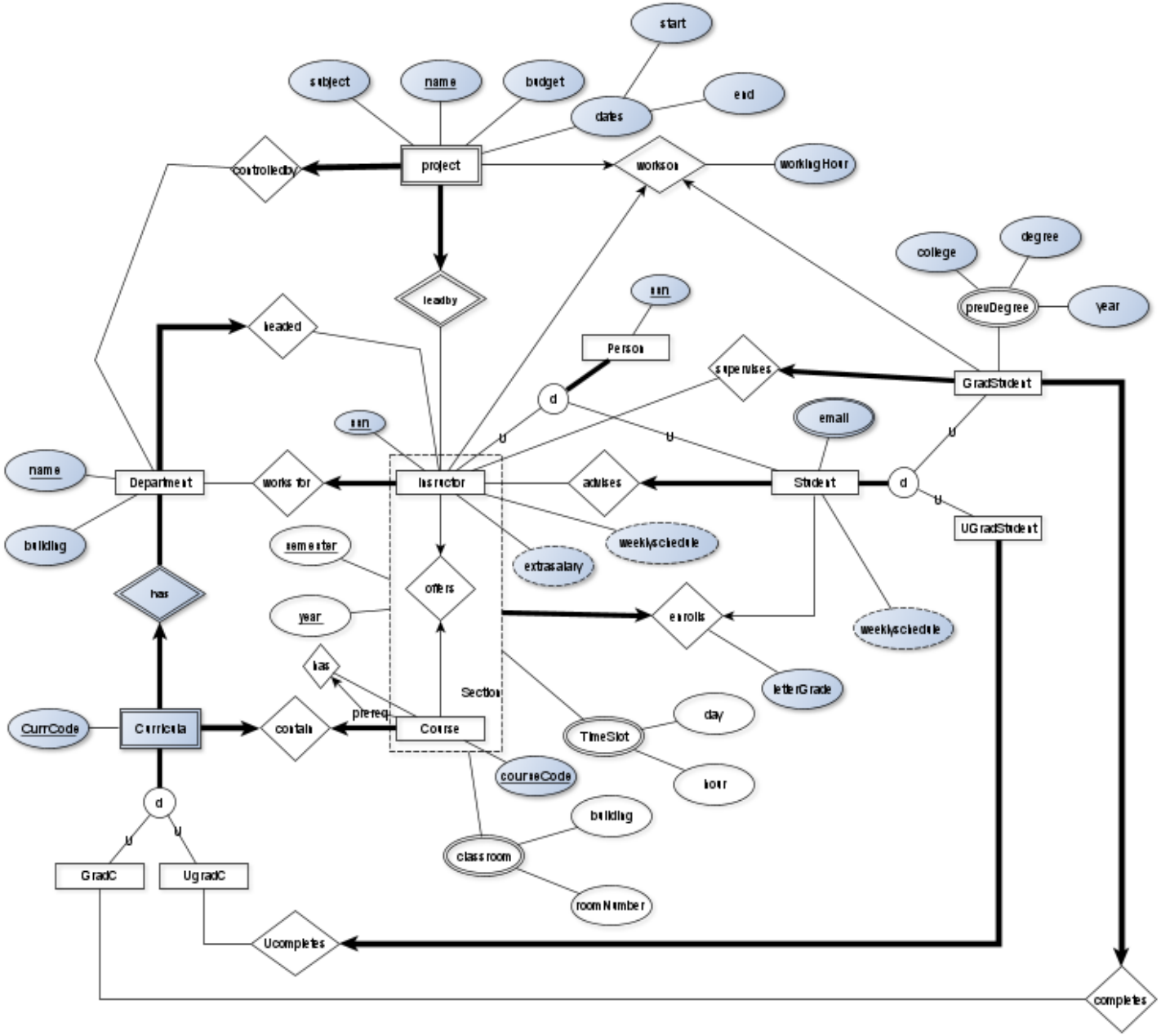
- Double major and minor options are ignored since not listed as requirement.
- Curricula is uniquely identified by its code, and department name.
- Direct Student-Department relationship is removed to prevent circular references. Dept of a student can be found by following curricula link, so as the students of a dept.
- Anyone can work on several projects.
- An instructor can lead to several projects.
- Students may have several e-mails, but emails are unique.

- Since every project has at least the lead instructor, "workson" is not total, i.e., there might be only one person working in the project who is the lead instructor.
- An instructor can be the head of several departments.
- Section Relation: In requirements "An instructor may offer several sections in a particular semester. " is given. It means that, an instructor can offer several sections of the same course in a particular semester/year. Therefore, (courseID, issn,semester,year) compound key cannot uniquely identify the tuples. To solve this issue sectionId is added, and included in compound key.
- "Curricula contain courses" relationship can be aggregate entity, say CurrCourse. Accordingly, instead of "Instructors offer Courses", one may prefer "Instructors offer CurrCourses". Although this approach is more correct, I haven't used it in my solution since Işık University is not following the mentioned approach, i.e., MATH1111 course for instance is not being offered separately for different departments. In other words, any section in our university can be taken by students of any department. However, it is still necessary to store curriculaCourses because it corresponds to your CCRs.
- There is no elective course, based on the requirements.
- Every student taking a particular section of a course will get a letter grade.
- Many obvious attributes have been omitted for the sake of clarity of the EER diagram where the relational model is complete.
- The weekly schedule which is a derived attribute of instructor and students in EER, can be found from enrollment relation in relational model.
- Exam is a weak entity, since you have many Midterm1, the name of the exam cannot be used as a unique identifier but only with the name of the course. Also, Question is a weak entity as every exam has question1, question2 etc.
- Following aggregations are required to obtain a sound model, many of them caused by weak relations[as there are many aggregations we'll try to be fair in grading].
    - Instructors offer courses →Section
    - Students enroll sections→Enrollment
    - Sections have Exams→ExamsOfSection
    - Exam has Questions→QuestionsofExam
    - Only the StudentsOfSection attends ExamsOfSection→StudentsTakingExam
    - Only the StudentsTakingExam gets points for QuestionsofExam→StudentGradesPerQuestion

  However, it was quite difficult to show the aggregations as they are supposed to be, I have repeated that part of diagram at a clear space.
- In relational schema, you will not see these aggregations, since the relationships (diamonds) are becoming tables as required.

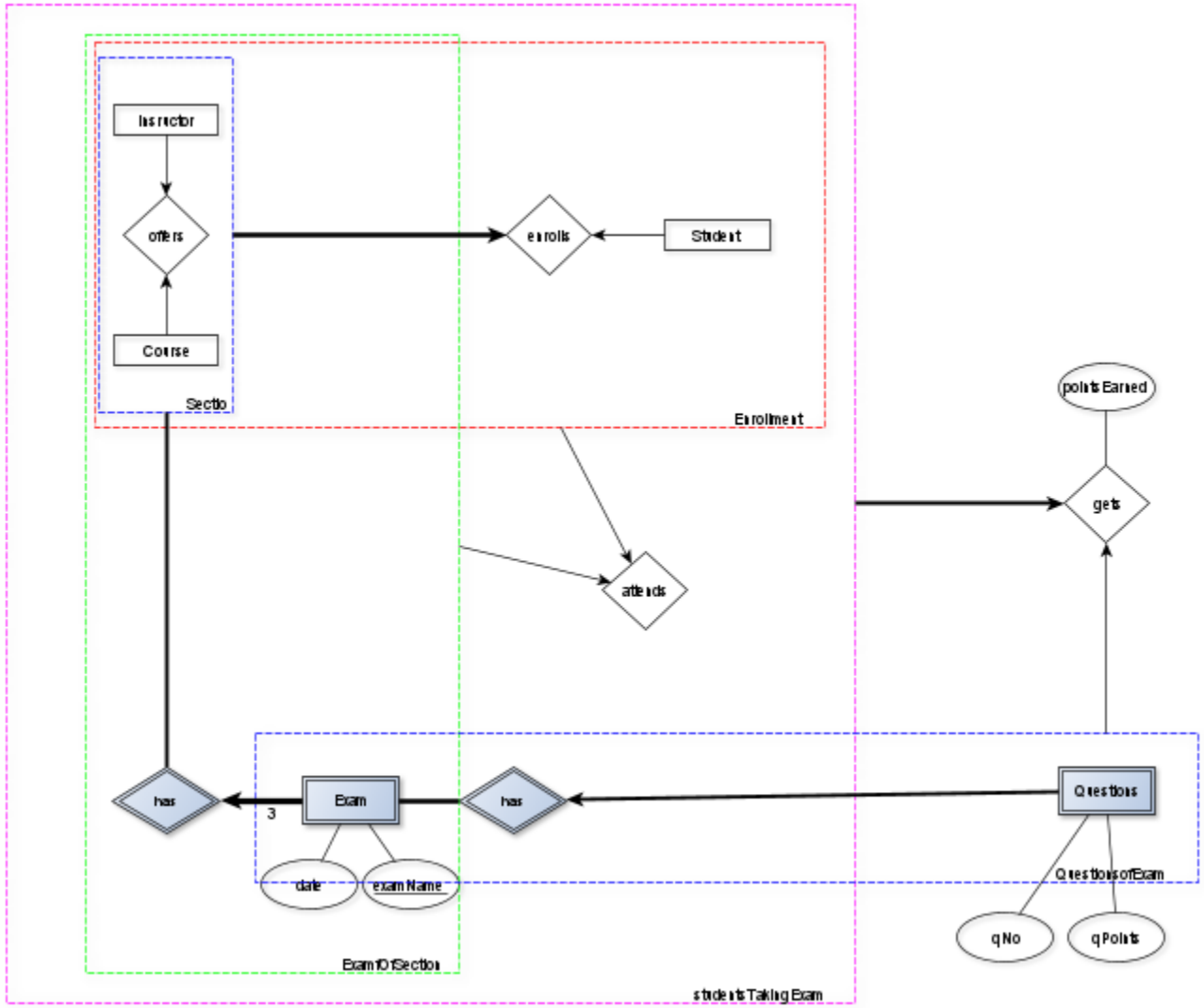
COMP2222 Database Systems
Spring 2023, Project Part#2
Due: May 17, 2023 23:50

IŞIK UNIVERSITY
COMPUTER SCIENCE AND ENGINEERING



COMP2222 Database Systems
Spring 2023, Project Part#2
Due: May 17, 2023 23:50

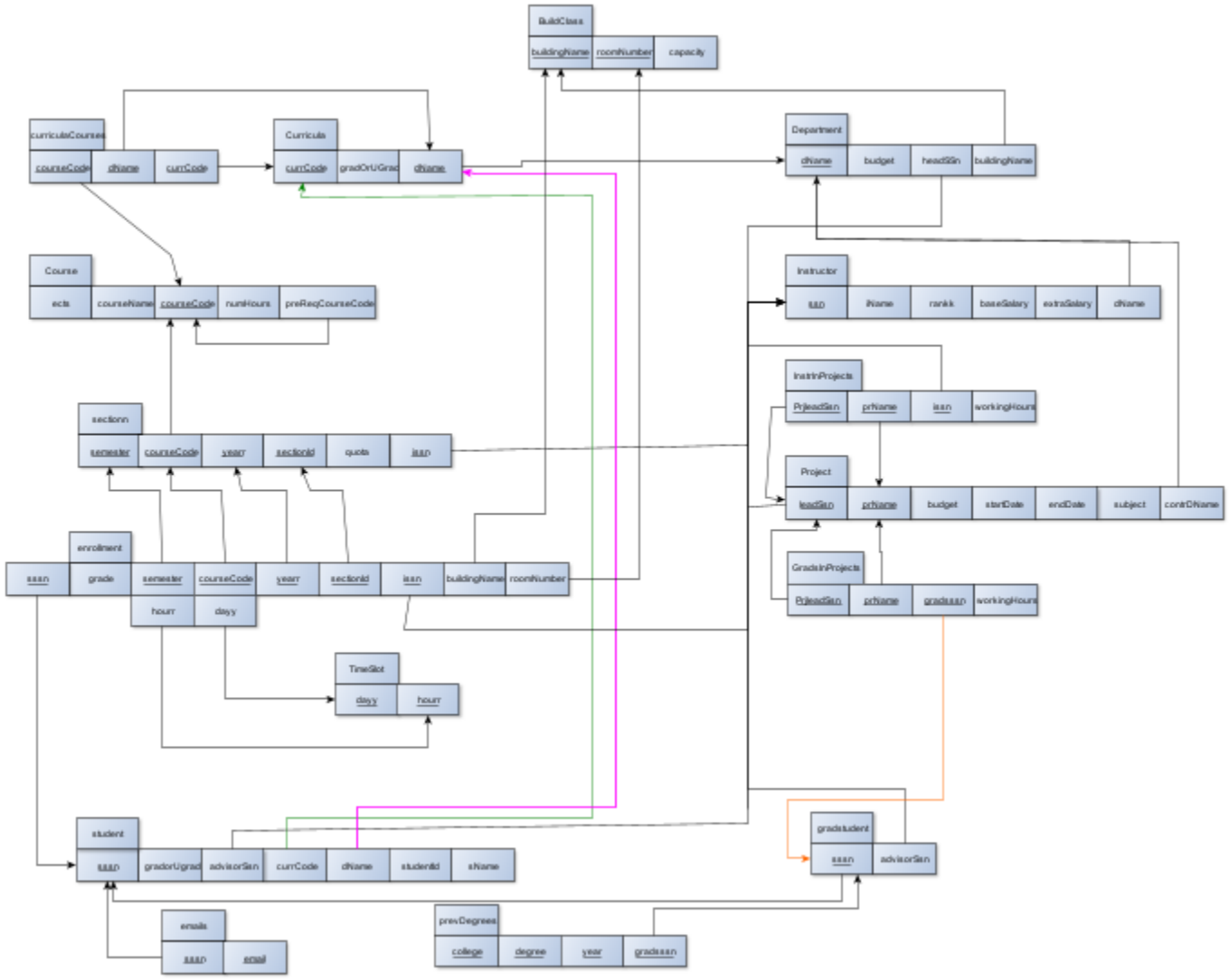
IŞIK UNIVERSITY
COMPUTER SCIENCE AND ENGINEERING



"First, solve the problem. Then write the code." John Johnson

COMP2222 Database Systems
Spring 2023, Project Part#2
Due: May 17, 2023 23:50

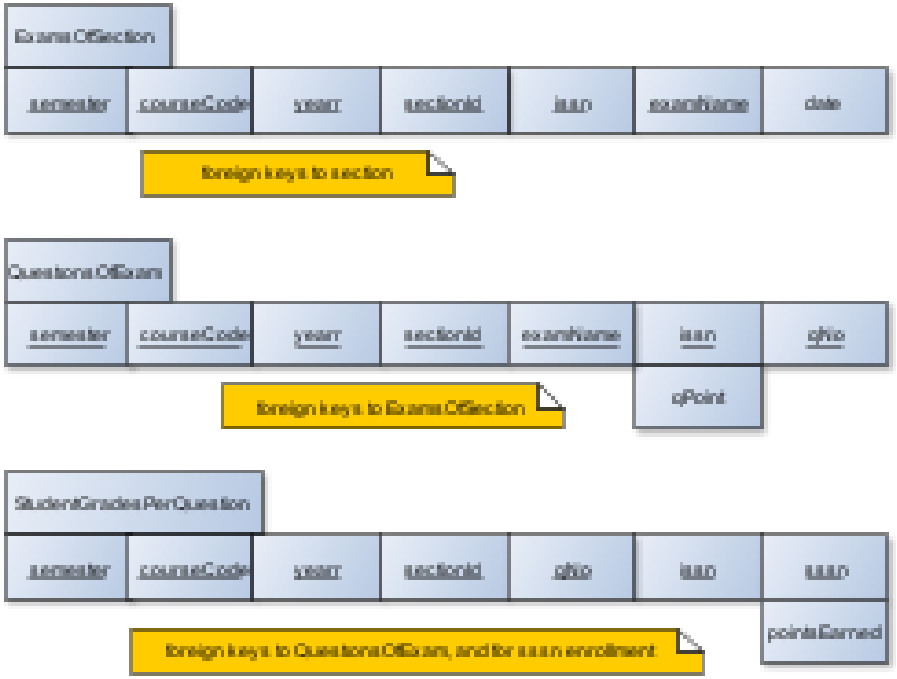
IŞIK UNIVERSITY
COMPUTER SCIENCE AND ENGINEERING



"First, solve the problem. Then write the code." John Johnson

COMP2222 Database Systems
Spring 2023, Project Part#2
Due: May 17, 2023 23:50

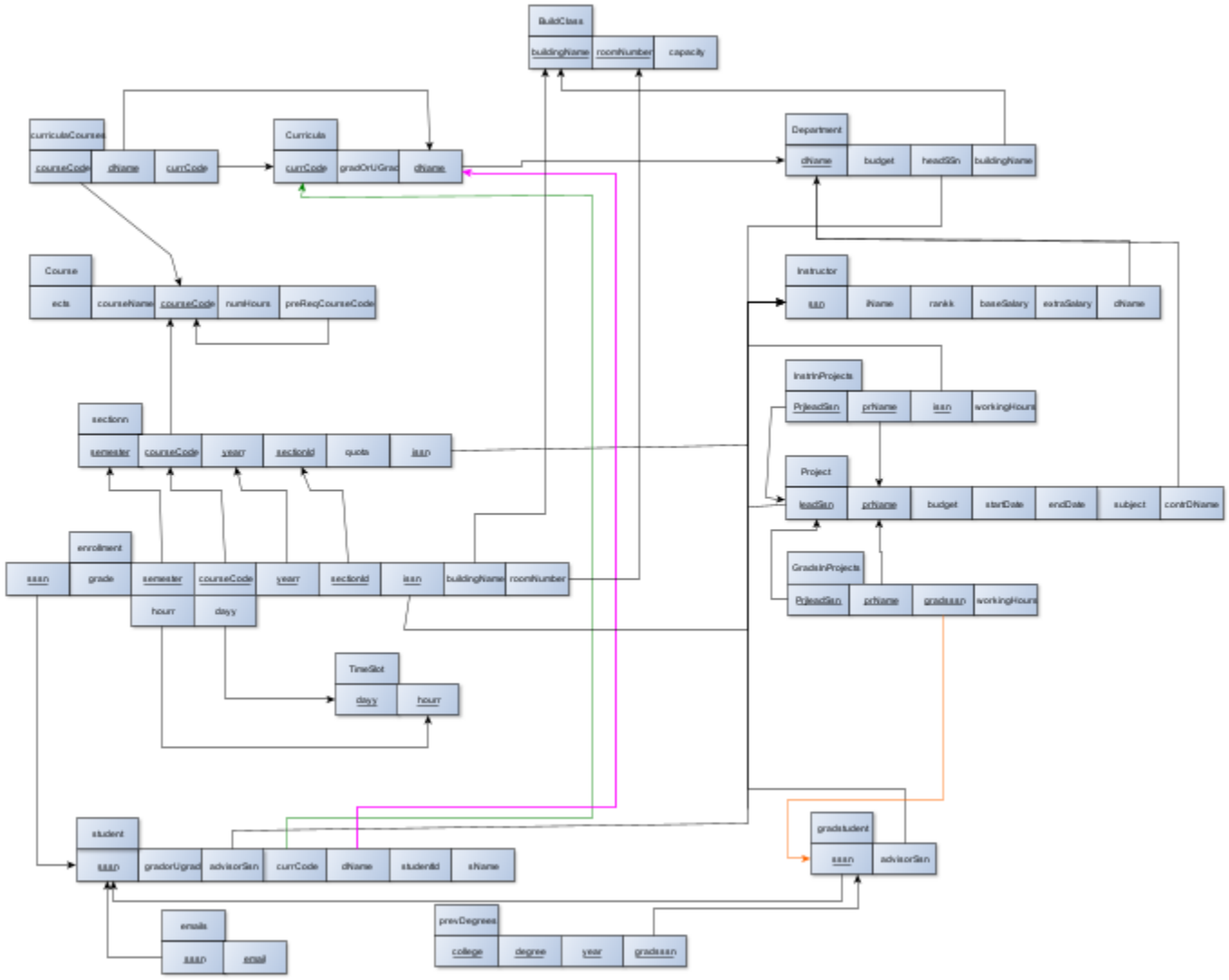
IŞIK UNIVERSITY
COMPUTER
SCIENCE AND
ENGINEERING



"First, solve the problem. Then write the code." John Johnson

COMP2222 Database Systems
Spring 2023, Project Part#2
Due: May 17, 2023 23:50

IŞIK UNIVERSITY
COMPUTER SCIENCE AND ENGINEERING



"First, solve the problem. Then write the code." John Johnson