Table of Contents

Section I: Project Description

The idea for my project will be to focus on the management system for a school. This will handle the School itself, Students, Courses, Faculty, Enrollment, Grading, Library, Books, and Dropping Courses/Students. This will have to keep track of specific attributes for all of the entities, such as units of the course, the students in each course, which instructor is teaching which course, etc. It will also handle faculty relationships - such as Lecturers, Professors, Researchers, and Teaching Assistants and Research Assistants. The entities will also all have attributes related to them such as years teaching, salary, the courses they teach or research topics they study. Finally it will also handle enrollment for students and courses, as well as students dropping a course or instructors dropping a student from a course.

Section II: Use Cases

Use Case Title	Enroll in Class
Actors	Student
Description	This use case begins when a student enrolls in one or more classes. The student must be eligible for registration and also must meet the prerequisites for that class. If the student has these requirements met when registering, they will be enrolled in the class unless there is a waitlist or the class is full, otherwise they will be placed in the waitlist / notified that the class is full. If they don't meet the prerequisites or are not eligible for registration yet, they will also be notified of that and unable to register.

Use Case Title	Drop a Class
Actors	Student
Description	This use case begins when a student attempts to drop a class from their schedule. The deadline for dropping a class must not have passed, and as long as they attempt to drop the class before that date, it should remove that class from the students schedule and remove the student from that class roster.

Use Case Title	Pay Fee for Class
Actors	Student

Description	If a student is enrolled in one or more classes, they can attempt to pay the fee for their classes. If their form of payment is accepted, their fee will be paid.

Use Case Title	Drop Student from Class
Actors	Teacher, Student(?)
Description	If a students' attendance is missing the first day of class or the teacher deems it applicable for another reason, then the teacher can drop the student from the class.

Use Case Title	Publish Paper
Actors	Researcher
Description	This use case begins when a Researcher at the school wants to publish a paper.

Use Case Title	Publish Grade
Actors	Professor, Instructor
Description	This use case happens when an Instructor or Professor publishes a Grade for a Student.

I think the “Drop Student from Class” user case could have a lot of possible failures. I think the “or the teacher deems it applicable for another reason” is too vague and there should somehow be more concrete rules, but I also don’t know enough about the process through which teachers drop students from their class. This would be fixable by changing it to something like:
“If the student doesn’t meet the criteria set forth by the teacher the first few class sessions, the teacher will drop them from the class.”
This is also vague but I think some things like this are different from class to class, it's the ability to drop a student that stays the same.

Section III: Database Requirements (Business Rules)

1. School
    a. A School has many Faculty Members
    b. A School has many Students
    c. A school has a Library
2. Library
    a. A Library has one school
    b. A Library has many Books
3. Book
    a. A Book has one Library
    b. A Book can be Rented by a Student
4. Rent Book
    a. A Student can Rent a Book
    b. Rent Book has one Student
5. Student
    a. A Student can enroll in a class
    b. A Student can check graded class
    c. A Student can drop a class
6. Faculty Member
    a. A Faculty Member is either a Professor, Lecturer, or Researcher
    b. A Faculty Member has a department
7. Professor
    a. A Professor is a Faculty Member
    b. A Professor has a department
    c. A Professor has one or many courses
    d. A Professor can be a department chair
8. Lecturer
    a. A Lecturer is a Faculty Member
    b. A Lecturer has a department
    c. A Lecturer has one or many courses
9. Teacher Assistant
    a. A TA is a Faculty Member
    b. A TA has one or many Professors/Lecturers
    c. A TA has one or many Courses
10. Researcher
    a. A Researcher is a Faculty Member
    b. A Researcher has a department
    c. A Researcher has published papers
    d. A Researcher has zero, one, or many Research Assistants.
11. Research Assistant
    a. An Assistant is a Faculty Member
    b. An Assistant has a mentor (Researcher)

12. Department
    a. A department will have at least one Professor or Lecturer
    b. A department will have at least one Course
13. Course
    a. A Course will have many Students
    b. A Course will have one Professor or Lecturer
    c. A Course will have a cost
    d. A Course has one or many Graders
14. Enrollment
    a. Enrollment will have at least one student
    b. Enrollment will have one course
    c. Enrollment will also have a section
    d. Enrollment will have a semester
15. Drop Course
    a. A Student can drop a Course
    b. A Professor can drop a student from a Course
    c. A Lecturer can drop a student from a Course
16. Drop Student
    a. A Professor can drop a Student from a Course
    b. A Lecturer can drop a Student from a Course
17. Grader
    a. A Grader will have one or many Courses
    b. A Grader will have zero or one faculty id
    c. A Grader will have zero or one student id (so the Grader is either a student (TA) or a Professor/Lecturer
18. Grade
    a. A Grade will have one Student
    b. A Grade will have one Course
19. Publish Grade
    a. Publish Grade will have one or many graders (Professor or Lecturer)
    b. It will have one Student
    c. It will have one Course
    d. It will have one date and time
20. Fee Due
    a. Fee will have a status (paid/unpaid)
    b. Fee will have a total net amount due
    c. Fee will have one or many courses
    d. Fee will have a student
21. Pay Fee
    a. Pay Fee will have an amount (possibly less than Fee Due's total amount due)
    b. Pay Fee will have a student
22. Papers
    a. A Paper will have one title
    b. A Paper will have one or many Researchers

    c. A Paper will have one date
23. Publish Paper
    a. Publish will have one or many Papers
    b. Publish Paper will have one or many Researchers

Section IV: Detailed List of Main Entities, Attributes and Keys

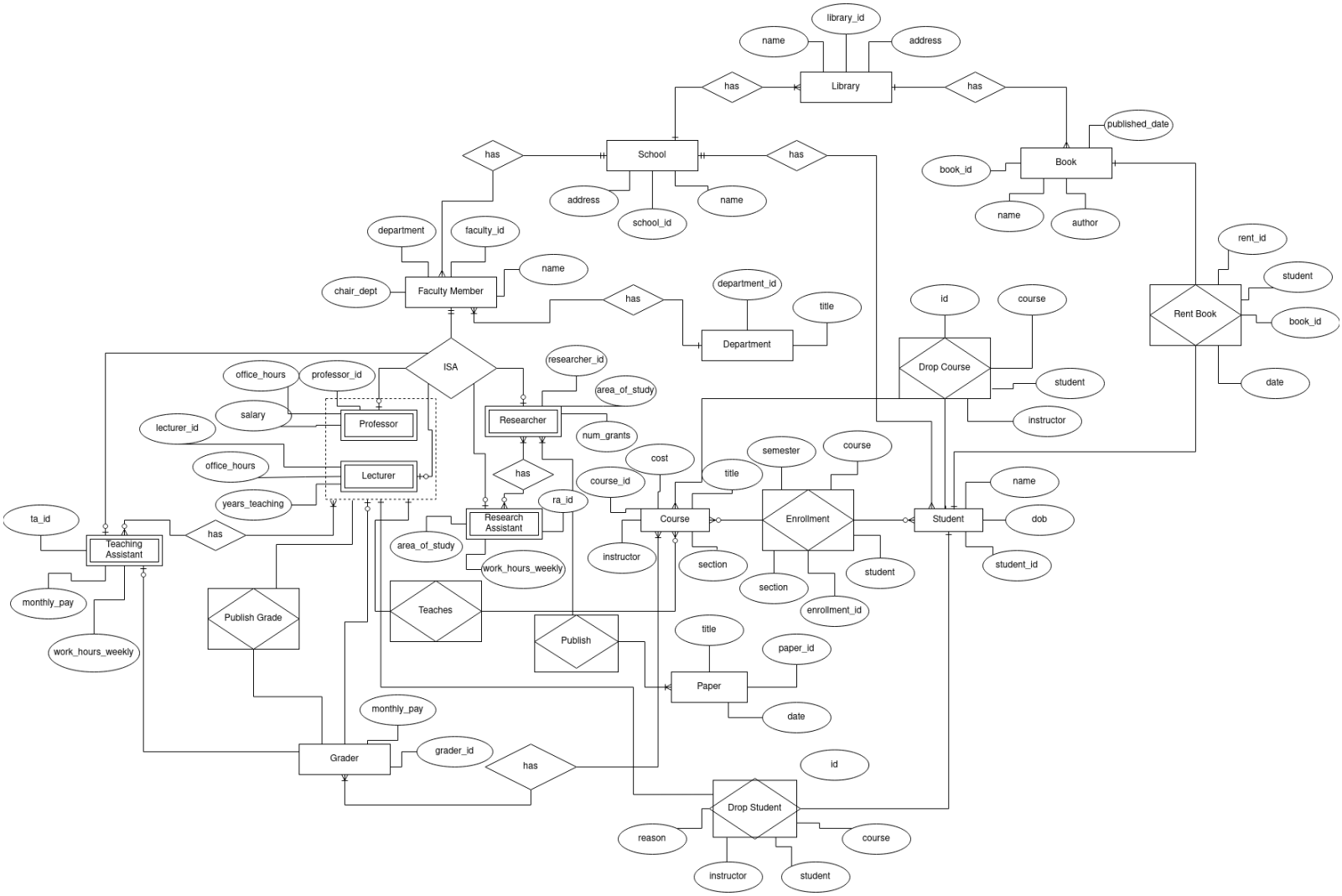
1. School (Strong)
    - school_id: key, numeric
    - name: composite, alphanumeric
    - address: composite, alphanumeric
2. Library (Strong)
    - library_id: key, numeric
    - name: composite, alphanumeric
    - address: composite, alphanumeric
3. Book (Strong)
    - book_id: key, numeric
    - name: composite, alphanumeric
    - author: composite, alphanumeric
    - published_date: multivalue, timestamp
4. Rent Book (Strong)
    - rent_id: key, numeric
    - student: composite, alphanumeric
    - book_id: numeric
    - date: multivalue, timestamp
5. Student (Strong)
    - student_id: key, numeric
    - name: composite, alphanumeric
    - dob: multivalue, timestamp
6. Faculty Member (Strong)
    - faculty_id: key, numeric
    - name: composite, alphanumeric
    - department: composite, alphanumeric
    - dob: multivalue, timestamp
7. Professor (Weak)
    - professor_id: key, numeric
    - office_hours: numeric
    - salary: numeric
8. Lecturer (Weak)
    - lecturer_id: key, numeric
    - office_hours: composite, alphanumeric
    - years_teaching: numeric
9. Teacher Assistant (Weak)
    - ta_id: key, numeric
    - monthly_pay: numeric
    - work_hours_weekly: numeric
10. Researcher (Weak)
    - researcher_id: key, numeric

    - num_of_grants: numeric
    - area_of_study: composite, alphanumeric
11. Research Assistant (Weak)
    - ra_id: key, numeric
    - work_hours_weekly: numeric
    - area_of_study: composite, alphanumeric
    - monthly_pay: numeric
12. Department (Strong)
    - dept_id : key, numeric
    - name: composite, alphanumeric
    - department_chair: composite, alphanumeric
13. Course (Strong)
    - course_id: key, numeric
    - name: composite, alphanumeric
    - department: composite, alphanumeric
    - section: composite, alphanumeric
    - semester: composite, alphanumeric
    - units: numeric
    - cost: numeric
14. Enrollment (Strong)
    - enroll_id: key, numeric
    - student: composite, alphanumeric
    - course: composite, alphanumeric
    - section: composite, numeric
    - semester: composite, alphanumeric
15. Drop Course (Strong)
    - drop_course_id: key, numeric
    - student: composite, alphanumeric
    - course: composite, alphanumeric
    - instructor: composite, alphanumeric
16. Drop Student (Strong)
    - drop_student_id: key, numeric
    - student: composite, alphanumeric
    - course: composite, alphanumeric
    - instructor: composite, alphanumeric
    - reason: composite, alphanumeric
17. Grader (Strong)
    - grader_id: key, numeric
    - course: composite, alphanumeric
    - student_id: key, numeric
    - faculty_id: key, numeric
18. Grade (Strong)
    - grade_id: key, numeric
    - student: composite, alphanumeric

    - course: composite, alphanumeric
19. Publish Grade (Strong)
    - publish_grade_id: key, numeric
    - student: composite, alphanumeric
    - course: composite, alphanumeric
    - grader: composite, alphanumeric
    - datetime: multivalue, timestamp
20. Fee Due (Strong)
    - fee_due_id: key, numeric
    - status: boolean
    - amount_due: numeric
    - student: composite, alphanumeric
21. Pay Fee (Strong)
    - pay_fee_id: key, numeric
    - amount_paid: numeric
    - student: composite, alphanumeric
22. Papers (Strong)
    - papers_id: key, numeric
    - title: composite, alphanumeric
    - Researcher: composite, alphanumeric
23. Publish Paper (Strong)
    - publish_paper_id: key, numeric
    - paper: composite, alphanumeric
    - date: multivalue, timestamp

Section V: Entity Relationship Diagram (ERD)

I will also have a PDF and draw.io file of this uploaded in my GitHub repository with this document (so it is easier to read)



Section VI: Testing Table

Rule	Entity A	Relation	Entity B	Cardinality	Pass/Fail	Error Description
1	School	has	Library	1-to-1 or Many	Pass	None
2	School	has	Faculty Member	1 and only 1-to-Many	Fail(?)	Faculty Members could also possibly work at another school(?)
3	School	has	Student	1 and only 1-to-Many	Fail(?)	Similar to above, Students could have multiple schools they attend(?)
4	Library	has	Book	1-to-Many	Pass	None
5	Student	Rents	Book	1-to-1	Fail	Student doesn't have to rent book
6	Student	Drops	Course	1-to-Many	Fail	Student doesn't have to drop course
7	Student	Enrolls	Course	0 or Many-to-0 or Many	Pass	None
8	Faculty Member	has	Depart-ment	1 or Many-to-1	Fail	Faculty Member could work under multiple departments
9	Instructor	Publish	Grade	1-to-Many	Pass	
10	Researcher	has	Research Assistant	1 or Many-to-Zero or Many	Pass	None
11	Instructor	has	Teacher Assistant	1 or Many-to-Zero or Many	Pass	None
12	Grader	has	Course	1 or Many-to-1 or Many	Pass	None
13	Researcher	Publish	Paper	1 or Many-to-1 or Many	Fail	Researcher doesn't have to publish a paper (so zero or many?)
14	Instructor	Drops	Student	1-to-1	Fail	Instructor doesn't have to drop a student

Section VII: Database Model/EER

Table	FK	ON DELETE	ON UPDATE	Comments
Book	Library	SET NULL	CASCADE	When a book is deleted, the library will no longer have that book
Teacher Assistant	Instructor	SET NULL	CASCADE	When a TA is deleted, the instructor no longer has a TA
Researcher	Faculty Member	CASCADE	CASCADE	A Researcher is a Faculty Member, so both must be deleted if one is.
Research Assistant	Researcher	SET NULL	CASCADE	When a research assistant is deleted, the researcher no longer has an assistant
RentBook	Student	SET NULL	CASCADE	If a student is deleted, then there is no longer a student who needs to rent a book
RentBook	Book	SET NULL	CASCADE	If a book is deleted then there is no longer a book to rent out
PublishPaper	Researcher	CASCADE	CASCADE	The paper should still be able to be published if a researcher is deleted
PublishPaper	Papers	CASCADE	CASCADE	When a paper is deleted, there is no longer a paper to publish
PublishGrade	Student	CASCADE	CASCADE	If a student is deleted, there is no longer a grade to publish
PublishGrade	Course	CASCADE	CASCADE	If a course is deleted, there is no longer a grade to publish
PublishGrade	Grader	NO ACTION	CASCADE	If a grader is deleted, the student should still receive a grade
Professor	Faculty Member	CASCADE	CASCADE	A Professor is a Faculty Member, so if one is deleted then both need to be
Pay Fee	Student	CASCADE	CASCADE	When a student is deleted, they can no longer pay a fee
Library	School	CASCADE	CASCADE	If the school is deleted, so is the library

Lecturer	Faculty Member	CASCADE	CASCADE	A Lecturer is also a Faculty Member so both need to be deleted if one is
Instructor	Faculty Member	CASCADE	CASCADE	An Instructor is also a Faculty Member so both need to be deleted if one is
Instructor	Course	SET NULL	CASCADE	If an Instructor is deleted, the course no longer has an instructor
Grader	Course	NO ACTION	CASCADE	If a Course is deleted, then the Grader should not be deleted
Grader	Faculty Member	NO ACTION	CASCADE	If a Faculty Member is deleted, then the Grader should not be deleted (could be instructor or student)
Grader	Student	NO ACTION	CASCADE	If a Student is deleted, then the Grader should not be deleted
Grade	Student	CASCADE	CASCADE	If a Student is deleted, their grade should also be deleted
Grade	Course	NO ACTION	CASCADE	If a Course is deleted, a Student still needs the grade for that Course
Fee Due	Student	NO ACTION	CASCADE	If a Student is deleted, they may still have an amount due
Faculty Member	School	CASCADE	CASCADE	If the School is deleted, the faculty member must also be deleted
Faculty Member	Department	NO ACTION	CASCADE	If a Department is deleted, the faculty member doesn't need to be deleted
Enrollment	Student	CASCADE	CASCADE	If a Student is deleted, the enrollment for that student is also deleted
Enrollment	Course	CASCADE	CASCADE	If a Course is deleted, any enrollment for that course is also deleted
Drop Student	Student	CASCADE	CASCADE	If a Student is deleted, they can no longer be dropped
Drop Student	Instructor	NO ACTION	CASCADE	If an Instructor is deleted, set to null because a new instructor may take the course
Drop Student	Course	CASCADE	CASCADE	If the Course is deleted, a student can't be dropped from it
Drop Course	Student	CASCADE	CASCADE	If a Student is deleted, they can no longer drop a course

Drop Course	Instructor	NO ACTION	CASCADE	If an Instructor is deleted, a Student can still drop a course
Drop Course	Course	CASCADE	CASCADE	If a Course is deleted, a Student can no longer drop it

Section XI: Testing Table

To preface this part, I fixed many of the errors I encountered when testing the table without realizing I was supposed to keep them. There are still a few errors, but many I fixed already and it seemed pointless to un-fix them if I have to in M3.

Entity	SQLQuery	Pass/Fail	Error Description	Possible Solution
School	Delete	Pass		
School	Update	Pass		
Library	Delete	Pass		
Library	Update	Pass		
Book	Delete	Pass		
Book	Update	Pass		
Student	Delete	Pass		
Student	Update	Pass		
Department	Delete	Pass		
Department	Update	Pass		
FacultyMember	Delete	Fail	Cannot delete or update a parent row: foreign key fail	Remove foreign key from grader?
FacultyMember	Update	Fail	Cannot delete or update a parent row: foreign key fail	Remove foreign key from grader?
Course	Delete	Pass		
Course	Update	Pass		
Enrollment	Delete	Pass		
Enrollment	Update	Pass		
TeacherAssistant	Delete	Pass		
TeacherAssistant	Update	Pass		
ResearchAssistant	Delete	Pass		
ResearchAssistant	Update	Pass		

PublishPaper	Delete	Pass		
PublishPaper	Update	Pass		
Grader	Delete	Pass		
Grader	Update	Pass		
Grade	Delete	Pass		
Grade	Update	Pass		
PublishGrade	Delete	Pass		
PublishGrade	Update	Pass		
RentBook	Delete	Pass		
RentBook	Update	Pass		
FeeDue	Delete	Fail	Cannot add or update a child row: foreign key constraint failed	Remove foreign key Student from FeeDue?
FeeDue	Update	Fail	Cannot add or update a child row: foreign key constraint failed	Remove foreign key Student from FeeDue?
PayFee	Delete	Pass		
PayFee	Update	Pass		
DropCourse	Delete	Pass		
DropCourse	Update	Pass		

Section XII: Business Rules & Discord Commands

1) Show every student and their grade level who are taking the specified Course.
   ```
   getStudents Analysis of Algorithms
   getStudents Operating Systems
   ```

2) Show the grade average for the specified Course and specified Semester (“Spring”, “Summer”, etc.)
   ```
   gradeAverage Operating Systems -Spring
   gradeAverage Intro to Java -Summer
   ```

3) Find all books written by Kyle Gilbert stored only in the specified library.
   ```
   writtenBy Kyle Gilbert -JP
   ```

4) Show all faculty member names who are working in the specified department
   showStaff Computer Science
   showStaff Physics

5) Find every paper published by every researcher with more than the specified number of grants
   getPapers 0
   getPapers 3

6) Find all books currently checked out by Students from the J.P. Library
   -JP Book Checked Out

7) Show every paper that was published at the AI Conference
   -AI Conference Papers