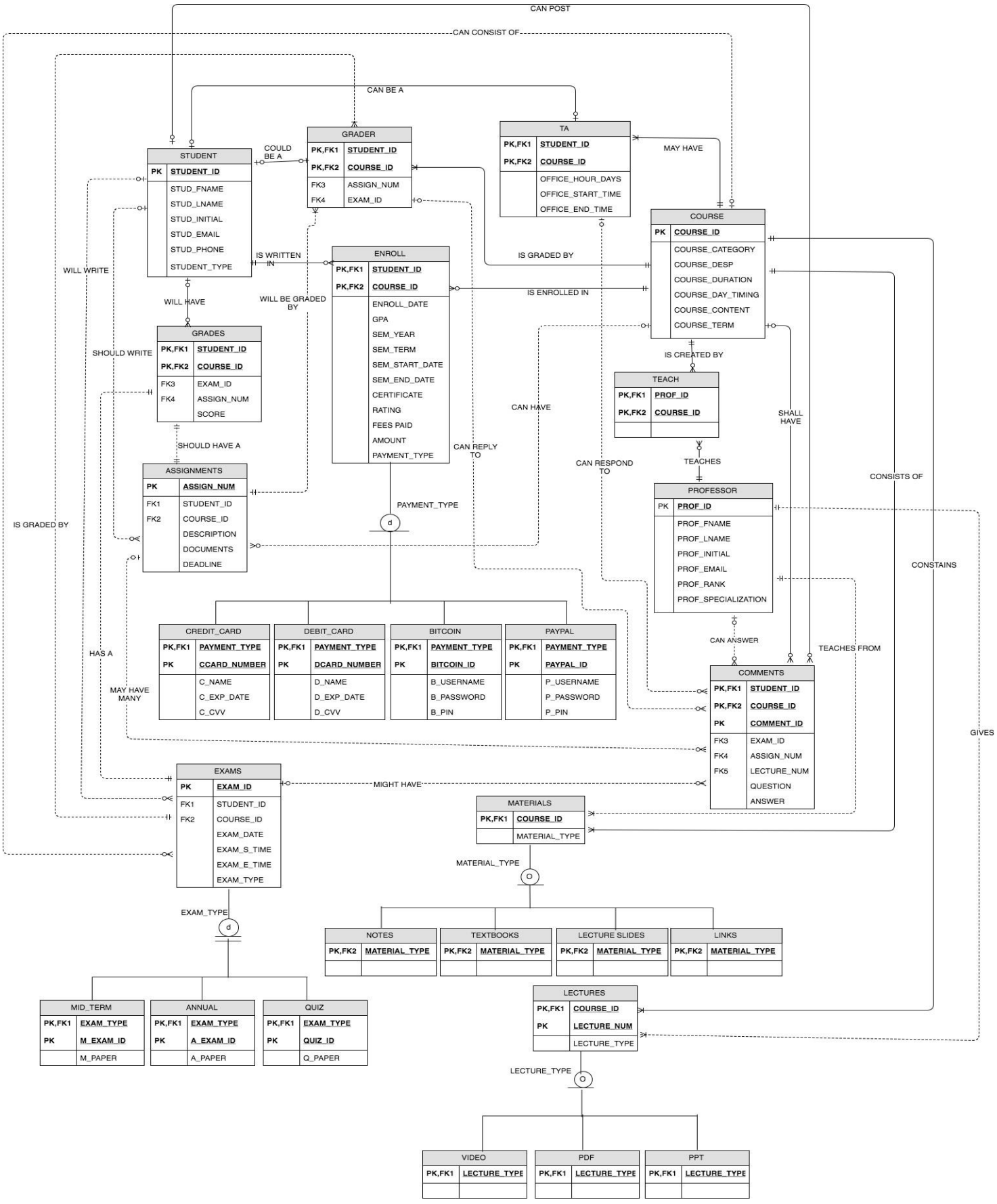
Database Systems: **EER Diagram for E-Learn**



Entities:

1. STUDENT: This table contains the details of a student/learner. **Attributes:** STUDENT_ID is the primary key. STUD_FNAME, STUD_LNAME, STUD_INITIAL, STUD_EMAIL, STUD_PHONE, STUD_TYPE: this can be grad or undergrad.

2. ENROLL: ENROLL is the composite entity that implements M:N relationship "STUDENT" is enrolled in "CLASS". This is a weak entity. **Attributes:** {STUDENT_ID, COURSE_ID} form the composite primary key. ENROLL DATE-student's enrollment date, GPA, SEM_YEAR- 1st,2nd or 3rd year, SEM_TERM- Fall, Spring or Summer, SEM_START_DATE- when semester begins, SEM_END_DATE-when semester ends(used since a course can be re-opened multiple times and each time is it re-opened Professors, TAs, Graders etc. can change.), CERTIFICATE- this is a Boolean variable which indicates whether the student has received a certificate or not, RATING- since students can rate a course, this variable holds the rating value. FEES_PAID- Since there is a constraint that only registered students can enroll, FEES PAID is a Boolean value which determines if a student has enrolled in a course or not. Only if the student has paid fees, he is enrolled in the course otherwise no. AMOUNT – this attribute holds the fees amount to be paid for each course. PAYMENT_TYPE- this is a supertype attribute which has subtypes CREDIT_CARD, DEBIT_CARD, PAYPAL, BITCOIN. The subtype tables contain required attributes for that particular subtype. This specifies the different payment options available for paying fees. The subtypes are disjoint and partial. It means that only one of the following methods can be used for payment each time and the more new methods of payment For example: paycheck etc. can be added.

3. COURSE: This table contains course details. **Attributes:** COURSE_ID is the primary key. COURSE_CATEGORY- this can be either undergrad level course, grad level course or optional course, required course etc. can be used to define pre-requisites. COURSE_DESP- this describes which major field the course belongs to and what other areas it comes under. COURSE_DURATION- how long is the course, how many months? COURSE_DAY_TIMINGS- what time does the course start/end and is scheduled on which days? COURSE_CONTENT- A short description about the course like topics covered in the course i.e. syllabus etc. COURSE_TERM- Fall, Spring or Summer.

4. PROFESSOR: This table holds Professor/Teacher details. **Attributes:** PROF_ID is the primary key. PROF_FNAME, PROF_LNAME, PROF_INITIAL, PROF_EMAIL, PROF_RANK- Assistant, Associate or Part-time lecturer. PROF_SPECIALIZATION- Professor's area of specialization.

5. TEACH: TEACH is the composite entity that implements M:N relationship "COURSE" is created by "PROFESSOR". {COURSE_ID, PROF_ID} form the composite primary key.

6. GRADER: Contains grader details. **Attributes:** {STUDENT_ID, COURSE_ID} form the composite primary key. STUDENT_ID, COURSE_ID, ASSIGN_NUM and EXAM_ID are the foreign keys.

7. TA: Contains TA details. **Attributes:** {STUDENT_ID, COURSE_ID} form the composite primary key. STUDENT_ID and COURSE_ID are the foreign keys. OFFICE_HOUR_DAYS, OFFICE_START_TIME, OFFICE_END_TIME- Since TAs can hold office hours, these attributes are used to determine office hour days and timings.

8. GRADES: This table is used to hold the scores for each individual assignment and exams. **Attributes:** {STUDENT_ID, COURSE_ID} form the composite primary key. STUDENT_ID, COURSE_ID, ASSIGN_NUM and EXAM_ID are the foreign keys. SCORE attribute is used to hold scores.

9. ASSIGNMENTS: This table is used to hold the assignment details for a particular course. **Attributes:** ASSIGN_NUM is the primary key. STUDENT_ID and COURSE_ID are the foreign keys. DESCRIPTION, DOCUMENTS and DEADLINE hold information about the assignment description, help related documents or questions for the assignment and the assignment's deadline.

10. EXAMS: This table is used to hold the exam details for a particular course. **Attributes:** EXAM_ID is the primary key. STUDENT_ID and COURSE_ID are the foreign keys. EXAM_DATE, EXAM_S_TIME(start time), EXAM_E_TIME(end time) and EXAM_TYPE hold exam related details. EXAM_TYPE is a supertype which has subtypes MID_TERM, ANNUAL and QUIZ which is used to determine what kind of an exam it is. The subtypes are disjoint and total. It means that each exam can only be one type and no new exam types can be added.

11. MATERIALS: this holds different material details for each course. **Attributes:** COURSE_ID is the primary key. MATERIAL_TYPE is a supertype attribute which has NOTES, TEXTBOOKS, LECTURE SLIDES, LINKS as the subtypes. The subtypes are overlapping and partial. Overlapping means MATERIAL_TYPE can be more than 1 subtype for a course.

12. LECTURES: this holds lecture details for each course. **Attributes:** {COURSE_ID,LECTURE_NUM} form the composite primary key. Since we want each lecture to be identified separately in each course, therefore we have made LECTURE_NUM as also the primary key. LECTURE_TYPE is a supertype attribute which has VIDEO,PDF,PPT as the subtypes. The subtypes are overlapping and partial. Overlapping means LECTURE_TYPE can be more than 1 subtype for a course.

13. COMMENTS: This holds details for the comments portal. **Attributes:** {STUDENT_ID,COURSE_ID} form the composite primary key. Same student can post multiple comments for a particular lecture, exam or assignment and so can the Professors, graders and TA's respond multiple times to same question. Therefore, STUDENT_ID, COURSE_ID, ASSIGN_NUM, LECTURE_NUM and EXAM_ID are the foreign keys. QUESTION attribute holds different questions or exam related queries asked by a learner for each course and ANSWER attribute holds responses by Professor, TAs and Graders.

COMPONENTS OF ENTITY RELATIONSHIP MODEL FOR E-LEARN			
ENTITY	RELATIONSHIP	CONNECTIVITY	ENTITY
STUDENT	is enrolled in	M:N	COURSE
STUDENT	will have	1:M	GRADES
STUDENT	can be a	1:1	TA
STUDENT	could be a	1:1	GRADER
STUDENT	should write	1:M	ASSIGNMENTS
STUDENT	will write	1:M	EXAMS
STUDENT	can post	1:M	COMMENTS
COURSE	may have	1:M	TA
COURSE	is graded by	1:M	GRADER
COURSE	can consist of	1:M	EXAMS
COURSE	can have	1:M	ASSIGNMENTS
COURSE	Is created by	M:N	PROFESSOR
COURSE	consists of	1:M	MATERIALS
COURSE	contains	1:M	LECTURES
COURSE	shall have	1:M	COMMENTS

PROFESSOR	teaches from	1:M	MATERIALS
PROFESSOR	gives	1:M	LECTURES
PROFESSOR	can answer	1:M	COMMENTS
EXAM	is graded by	1:M	GRADER
ASSIGNMENTS	will be graded by	1:M	GRADER
GRADER	can reply to	1:M	COMMENTS
TA	can respond to	1:M	COMMENTS
ASSIGNMENTS	should have a	1:1	GRADES
ASSIGNMENTS	may have many	1:M	COMMENTS
EXAMS	has a	1:1	GRADES
EXAMS	might have	1:M	COMMENTS

NOTE:

1. **STUDENT : COURSE** – **ENROLL** is the composite entity that implements M:N relationship "STUDENT" is enrolled in "CLASS". Since many students can be enrolled in a single course and each student can enroll in multiple courses.
2. **COURSE : PROFESSOR** – **TEACH** is the composite entity that implements M:N relationship "COURSE" is created by "PROFESSOR". Since there can be many professors involved in creating and teaching a course and also a single professor can teach multiple courses based on category(Grad, undergrad etc.).

Relationships:

1. STUDENT : COURSE – this is M:N relationship since I have assumed as multiple students can take multiple courses. ENROLL is the composite entity which implements this relationship.
2. STUDENT : GRADES – this is a 1:M relationship since each student can have multiple grades. EXAM_ID and ASSIGN_NUM are foreign keys here because each student can have only one score for each exam and each assignment.
3. STUDENT : TA – this is 1:1 relationship since each student can either be a TA or not be a TA because only enrolled students can be TAs and therefore Cardinality on both sides is [0,1], meaning non-enrolled students cannot be TAs. Which suggests it is optional.
4. STUDENT : GRADER – this is 1:1 relationship since each student can either be a grader or not be a grader because only enrolled students can be graders and therefore Cardinality on both sides is [0,1], meaning non-enrolled students cannot be graders. Foreign keys here are STUDENT_ID, COURSE_ID, EXAM_ID and ASSIGN_NUM. This is to keep track of which grader grades which assignment or exam and in which course. Not all graders can grade all the exams and assignments. Each grader will be given a particular assignment and exam to grade.
5. STUDENT : ASSIGNMENTS – this is 1:M relationship because each enrolled student has to write either assignments but non-enrolled students need not write any assignment. Therefore the cardinality is [0,1]:[0,M] on both sides. Which suggests it is optional.
6. STUDENT : EXAMS - this is 1:M relationship because each enrolled student has to write exams but non-enrolled students need not write any exam. Also some courses may not have any exams. Therefore the cardinality is [0,1]:[0,M]. Which suggests it is optional.
7. STUDENT : COMMENTS – this is 1:M relationship because each student can post multiple comments regarding an exam or assignment.
8. COURSE : TA – this is 1:M relationship since each student can be a TA for only one course, But a course can have multiple TAs. Each course should have atleast one TA. Therefore cardinality is [1,1]:[1,M].

9. COURSE : GRADER – this is 1:M relationship since each student can be a grader for only one course, But a course can have multiple graders. Each course should have atleast one grader. Therefore cardinality is [1,1]:[1,M].
10. COURSE : EXAMS – this is 1:M relationship since each course can have multiple exams, but some courses may have only assignments and no exams. Also particular course may not be offered in a particular semester Therefore cardinality is [0,1]:[0,M].
11. COURSE : ASSIGNMENTS – this is 1:M relationship since each course can have multiple assignments and also a particular course may not be offered in a particular semester. Therefore cardinality is [0,1]:[0,M].
12. COURSE : PROFESSOR – this is a M:N relationship. TEACH is the composite entity that implements M:N relationship "COURSE" is created by "PROFESSOR".
13. COURSE : MATERIALS – this is 1:M relationship since each course can have multiple materials. Cardinality is [1,1]:[1,M].
14: COURSE : LECTURE – this is 1:M relationship since each course can have multiple lectures. Cardinality is [1,1]:[1,M].
15: COURSE : COMMENTS – this is a 1:M relationship since each course can have multiple comments about it. Cardinality is [0,1]:[0,M].
16. PROFESSOR : MATERIALS – this is a 1:M relationship since each Professor gives more than one materials. And each professor has to give atleast 1 material. Which makes it mandatory and hence the cardinality is [1,1]:[1,M].
17. PROFESSOR : LECTURES – this is a 1:M relationship since each Professor gives more than one lectures. And each professor has to give atleast 1 lecture. Which makes it mandatory and hence the cardinality is [1,1]:[1,M].
18. PROFESSOR : COMMENTS – this is a 1:M relationship since each professor can post multiple comments. The professor may or may not post comments as well. So this is optional and therefore cardinality is [0,1]:[0,M].
19. EXAMS : GRADER – this is a 1:M relationship since each exam can be graded by multiple graders. Each exam should have atleast 1 grader. Therefore cardinality is [1,1]:[1:M].
20. ASSIGNMENTS : GRADER - this is a 1:M relationship since each assignment can be graded by multiple graders. Each assignment should have atleast 1 grader. Therefore cardinality is [1,1]:[1:M].
21. GRADER : COMMENTS - this is a 1:M relationship since each grader can post multiple comments. The grader may or may not post comments as well. So this is optional and therefore cardinality is [0,1]:[0,M].
22. TA : COMMENTS - this is a 1:M relationship since each TA can post multiple comments. The TA may or may not post comments as well. So this is optional and therefore cardinality is [0,1]:[0,M].
23. ASSIGNMENTS : GRADES – this is 1:1 relationship since each assignment can have only one grade. Therefore cardinality is [1,1]:[1,1].
24. ASSIGNMENTS : COMMENTS – this is 1:M relationship since multiple comments can be there on each assignment. Comments may not be there sometimes. Therefore cardinality is [0,1]:[0,M].
25. EXAMS : GRADES – this is 1:1 relationship since each exam can have only one grade. Therefore cardinality is [1,1]:[1,1].
26. EXAMS : COMMENTS – this is 1:M relationship since multiple comments can be there on each exam. Comments may not be there sometimes. Therefore cardinality is [0,1]:[0,M].

Multi-valued Attributes:

There are two multi-values attributes in my EER-Diagram, they are: MATERIAL_TYPE and LECTURE_TYPE. Each course material can contain several overlapping types such as NOTES, TEXTBOOKS, LECTURE SLIDES, LINKS etc. per semester. Similarly each course lectures can contain many different types of materials such as

VIDEOs, PDFs and PPTs etc. for every chapter. Therefore, I have considered these two attributes as multi-valued attributes.

Trade-offs:

There are two trade-offs in this design. I have considered COURSE : GRADER as 1:M meaning each course can have multiple graders grade it, but a grader cannot grade multiple courses. That means a student can only be a grader for just 1 course and not multiple courses for each semester. Although same person can be a grader for another course for next semester. This could have been M:N relationship as well but it would require creating composite bridge entity which would be redundant and make the design more complicated. Hence for simplicity I have taken it as 1:M.

Similar to the above, I have considered COURSE: TA also as 1:M. This could have also been M:N but to avoid redundancy and for simplicity of design I have considered it as 1:M.