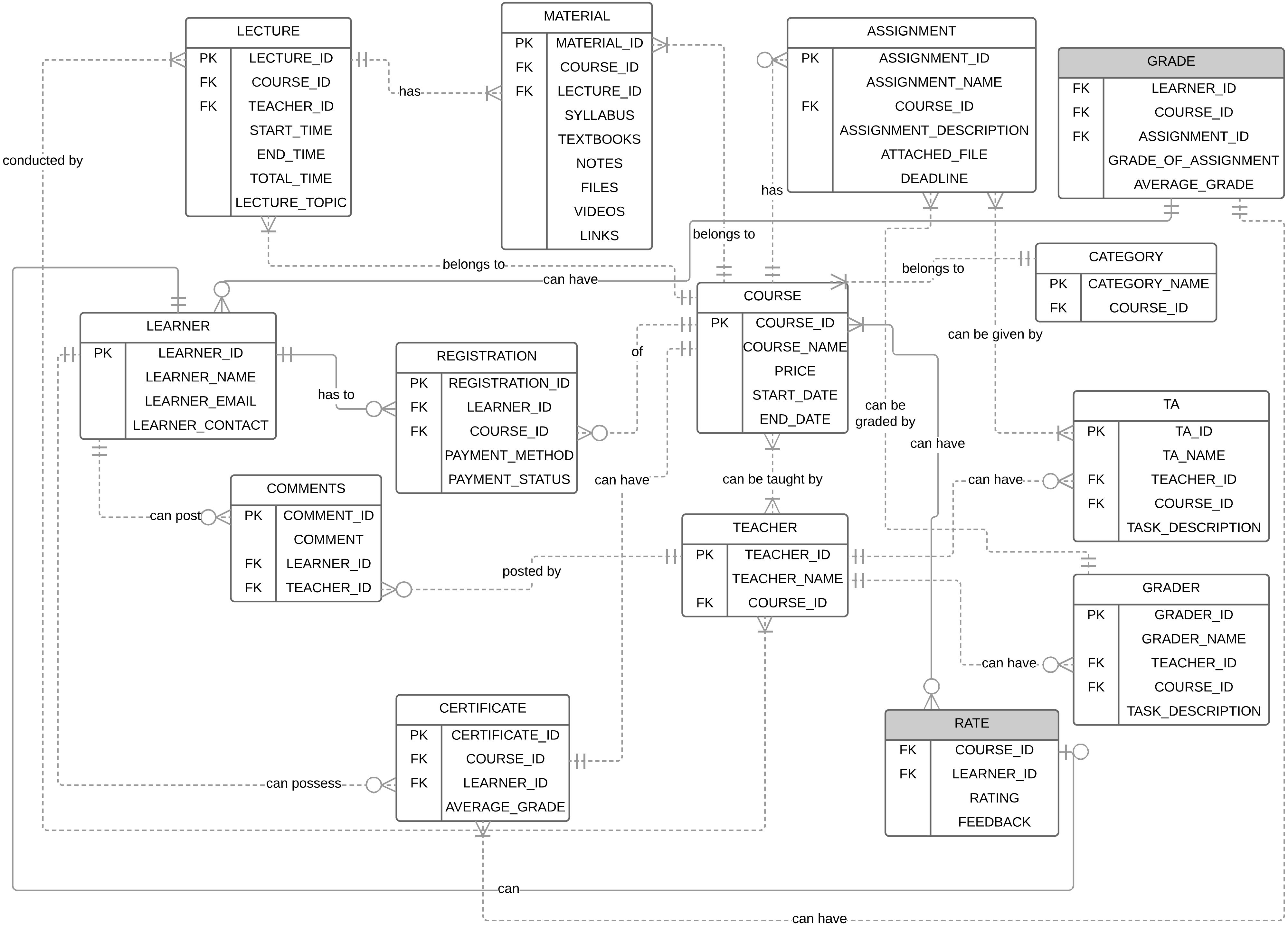

	LECTURE
PK	LECTURE_ID
FK	COURSE_ID
FK	TEACHER_ID
	START_TIME
	END_TIME
	TOTAL_TIME
	LECTURE_TOPIC

	MATERIAL
PK	MATERIAL_ID
FK	COURSE_ID
FK	LECTURE_ID
	SYLLABUS
	TEXTBOOKS
	NOTES
	FILES
	VIDEOS
	LINKS

	ASSIGNMENT
PK	ASSIGNMENT_ID
	ASSIGNMENT_NAME
FK	COURSE_ID
	ASSIGNMENT_DESCRIPTION
	ATTACHED_FILE
	DEADLINE

	GRADE
FK	LEARNER_ID
FK	COURSE_ID
FK	ASSIGNMENT_ID
	GRADE_OF_ASSIGNMENT
	AVERAGE_GRADE

	CATEGORY
PK	CATEGORY_NAME
FK	COURSE_ID

	COURSE
PK	COURSE_ID
	COURSE_NAME
	PRICE
	START_DATE
	END_DATE

	LEARNER
PK	LEARNER_ID
	LEARNER_NAME
	LEARNER_EMAIL
	LEARNER_CONTACT

	REGISTRATION
PK	REGISTRATION_ID
FK	LEARNER_ID
FK	COURSE_ID
	PAYMENT_METHOD
	PAYMENT_STATUS

	TA
PK	TA_ID
	TA_NAME
FK	TEACHER_ID
FK	COURSE_ID
	TASK_DESCRIPTION

	COMMENTS
PK	COMMENT_ID
	COMMENT
FK	LEARNER_ID
FK	TEACHER_ID

	TEACHER
PK	TEACHER_ID
	TEACHER_NAME
FK	COURSE_ID

	GRADER
PK	GRADER_ID
	GRADER_NAME
FK	TEACHER_ID
FK	COURSE_ID
	TASK_DESCRIPTION

	CERTIFICATE
PK	CERTIFICATE_ID
FK	COURSE_ID
FK	LEARNER_ID
	AVERAGE_GRADE

	RATE
FK	COURSE_ID
FK	LEARNER_ID
	RATING
	FEEDBACK

REPORT

ENTITY	PRIMARY KEY	FOREIGN KEY	MULTIVALUED ATTRIBUTE	DERIVED ATTRIBUTE	OTHER POSSIBLE PRIMARY KEY
LEARNER	learner_id	-	learner_contact	-	learner_email
REGISTRATION	registration_id	learner_id, course_id	course_id	payment_status	learner_id + course_id
COURSE	course_id (compound key)	-	start_date, end_date	-	course_name
CATEGORY	category_name	course_id	course_id	-	-
TEACHER	teacher_id	course_id	course_id	-	-
LECTURE	lecture_id	course_id, teacher_id	lecture_topic	total_time	-
MATERIAL	material_id	course_id, lecture_id	textbooks, notes, files, videos, links	-	course_id + lecture_id OR only lecture_id
ASSIGNMENT	assignment_id	course_id	attached_file	-	assignment_name
GRADE	learner_id + course_id + assignment_id	learner_id, course_id, assignment_id	assignment_id	-	-
COMMENTS	comment_id	learner_id, teacher_id	-	-	-
TA	ta_id	teacher_id, course_id	-	-	-
GRADER	grader_id	teacher_id, course_id	-	-	-
CERTIFICATE	certificate_id	course_id, learner_id	-	-	course_id + learner_id
RATE	course_id + learner_id	course_id, learner_id	rating	-	-

Figure – A brief outline about the ER Diagram

ENTITY

In the ER Diagram, the entities are: Learner, Registration, Course, Category, Lecture, Material, Assignment, Grade, Comments, Teacher, TA, Grader, Certificate, Rate.

LEARNER

Let us start from the Learner entity. A learner will have his/her id, name, email, contact. learner_id is the attribute that uniquely identifies a learner and hence it is the primary key. Here, we can also make learner_email as primary key for the Learner Entity, as email id of every learner will be unique. learner_contact has to be the multivalued attribute as there can be multiple contact numbers of a particular learner.

REGISTRATION

A learner has to make registration of course. For registering, a learner will require his/her unique id (foreign key), the course_id of the course he/she wants to register (foreign key), the payment_method. If the payment is done then the payment_status should be true and the course is registered. So we can derive the value of payment_status.

Here, we can either make the combination of two foreign keys (i.e. learner_id and course_id) as our primary key for registration table, which would make strong entity relationship between learner and registration. Instead we have made a registration_id attribute as the primary key, forming a weak entity relationship. Still it is dependent on existence of learner and course table (because if there is no learner or course, then who will register or what will the person register for?).

COURSE

Registration of a course is done. Course entity will have its own unique course_id which is actually a compound key. This key will have its first two characters derived from the Subject name and then a 3 digit course number. So it can be split into two parts. The course_name can also be made a primary key because there won’t be any two courses with the same name. The other attribute in the entity is the price of the course. Also, start_date and end_date can have multiple values as the course can be reopened again. Registration works as a bridge table between Learner and Course tables.

CATEGORY

There can be no two categories with same name, hence we can make category_name as the primary key as well. Under a particular category, there can be multiple courses, hence the course_id foreign key attribute works as a multivalued attribute here.

TEACHER

A teacher is uniquely identified by teacher_id. A teacher can teach multiple courses, hence course_id will be a multivalued attribute. Also course_id works as a foreign key which matches the teacher with the courses he/she teaches.

LECTURE

Each lecture is uniquely identified with its lecture_id. It will have information like course_id, teacher_id, start_time of the lecture, end_time of the lecture, and from the start and end time we can derive total_time of the lecture. Other attribute it has is lecture_topic which is a multivalued attribute as there can be multiple topics covered in a lecture.

MATERIAL

Material can be uniquely identified by its material_id. It requires information like which course and lecture it belongs to. Material can have multiple textbooks, notes, files, videos, links, so these attributes are multivalued. Material also has the syllabus or overview for every particular lecture.

ASSIGNMENT

Assignment can be uniquely identified by its assignment_id. Assignment name can also be unique, hence another possible primary key for this particular entity is assignment_name. An assignment belongs to a course, hence we use course_id as foreign key. An assignment can have multiple files attached, hence attached_file is a multivalued attribute here. Other attributes include the description and deadline of the assignment.

GRADE

A grade can be uniquely identified by the combination of learner_id, course_id, and assignment_id i.e. a particular learner's particular course's particular assignment is graded. Grade_of_assignment attribute represents the grade for that particular assignment. assignment_id directly represents the total number of assignments so far. average_grade attribute is kept to keep the updated record of average grade with each new assignment.

COMMENT

A comment can be uniquely identified by its comment_id. It can be posted by a learner as well as a teacher, and the comment attribute is not made a multivalued attribute as for a particular comment_id, there will be a single comment. Every learner and teacher can post multiple comments, and each comment will have its own id.

TA

A TA has his/her own unique id, name, which faculty he/she works under, which particular course he/she assists for, and the description of his/her task. A TA can works under a single faculty, one TA does not belong to multiple faculties. A TA sets assignments for learners.

GRADER

A grader has his/her own unique id, name, under which faculty he/she is assigned to, which particular course he/she grades for, and the description of his/her task. A grader can works under a single faculty, one grader does not belong to multiple faculties. A grader grades assignments for learners.

CERTIFICATE

A certificate can be uniquely identified by its id. We can use the composite key (combination of learner_id and course_id) to uniquely identify a certificate that belongs to a particular learner for a particular course. It also has average_grade attribute from the grade entity.

RATE

Rate entity includes ratings of a particular course by a particular learner. A learner can also give feedback for the course he/she has taken. We can enhance this by adding total_ratings attribute and deriving average_ratings attribute. There can be an addition of rating with each learner in the same course, total_learners for particular course can then be divided by the total _ratings and we derive the average rating for that course.

RELATIONSHIP

1. A learner can make zero or many registrations for pursuing different courses. A single registration can be made by a single learner only.
2. A single registration can be made for a single course only, while a single course can have multiple registrations.
3. A course belongs to a single category, and a category can have multiple courses in it.
4. A course can be taught by multiple teachers and a teacher can teach multiple courses.
5. A lecture can be conducted by many teachers and a teacher can conduct many lectures.
6. A lecture belongs to one and only course and a course can have multiple lectures.
7. A lecture can have at least one or more material while a particular material belongs to a single lecture.
8. A course can have multiple material while a single material belongs to one and only one course.
9. A teacher can have zero or many TAs while a TA can be assigned to one and only one teacher.
10. A teacher can have zero or many graders while a grader can be assigned to one and only one single teacher.
11. A teacher may or may not post comments (and if posted then the teacher may post many comments) while a single comment belongs to a single teacher.
12. A learner can post zero or more comments, while a single comment belongs to one and only one learner.
13. A learner can possess zero or more certificates based on number of courses he/she pursues and eligibility. A single certificate can belong to one and only one learner.
14. A learner can give zero or one rating and feedback to a particular course. A single rating or feedback can be given by one and only one learner.
    If a learner does not exist then there can be no ratings, as the rating can be uniquely identified using the composite key (course_id + learner_id). Hence Learner and Rate forms a strong relationship, and Rate becomes the weak entity.
15. A learner can get one and only one grade for an assignment of a particular course. Similarly, a single grade can belong to zero or more learners.
    learner_id, course_id, assignment_id are the foreign keys for Grade entity and it does not have its own primary key to uniquely identify any of its rows, so the existence of each Grade depends on these three entities, which forms a strong relationship with learner, and Grade becomes the weak entity.
16. A particular assignment can belong to that particular course only. A course may not have assignments or may have any number of assignments.
17. A single course can have one and only one certificate, and a single certificate belongs to that particular course.
18. A course can have zero or more ratings. A rating or feedback can belong to one or more courses.
    If a course does not exist then there can be no ratings, as the rating can be uniquely identified using the composite key (course_id + learner_id). Hence Course and Rate forms a strong relationship, and Rate becomes the weak entity.

19. An assignment can be given by one or more TAs, and a TA can give one or more assignments.
20. An assignment can be graded by one and only one grader. (Here, we are not considering the possibility where each question of an assignment can be graded by a different grader. In that case, it will be a many to many relationship.) A grader can grade one or more assignments.
21. A certificate can have one and only one grade for the learner, while a grade can belong to multiple certificates.

In the ER Diagram, all weak entities are represented using shaded header here.