eLearn Database Design

INTRODUCTION:

eLearn is an e-learning platform. Learners can explore, enroll, study and get certificates from a variety of courses. Each course has a unique course id and consists of materials, lectures and assignments. Learner is a student. Other users are TA, Graders, and Teachers. Student can be faculty also. All Users have different permission levels for each course. Student has to pay to enroll in a course. He will get grade, certificate after completion of the course. Student can provide feedback or give rating to a course. There is discussion forum where students and teachers can comment regarding assignments and lectures.

IDENTIFYING ENTITIES:

1. **USERS** – Contains all the user IDs and User names – Students and staff.
2. **TYPE_OF_USER** - list of User types.
3. **STUDENT** – list of Students.
4. **TA** – list of TAs.
5. **GRADER** – list of Graders.
6. **TEACHER** – list of Teachers.
7. **COURSE** – Contains all the course ID's and their names.
8. **COURSE_DETAILS** – Gives information about, to which all category, a particular course belongs to.
9. **COURSE_CATEGORY** – list of course categories.
10. **MATERIALS** – Contains details of all the materials related to a course.
11. **LECTURE** – contains lecture details of a course.
12. **LECTURE_MATERIAL** – gives information about lecture materials.
13. **LM_TYPE** – list of lecture material type.
14. **ASSIGNMENT** – contains assignment related information under each course.
15. **COURSE_OFFERING** – Gives information on which courses are offered in a year at different season (Spring, Summer, Fall ...)
16. **COURSE_TERM** – Holds all different seasons in a year (spring, summer, and fall).
17. **COURSE_CREATOR** – Contains information on which course is created by whom.
18. **COURSE_INVOLVEMENT** – Which are teachers are involved in a particular course.
19. **ENROLL** – holds course registration information.
20. **PAYMENT_MODE** – list of different types of payments types available.
21. **STUDENT_COURSE_DETAILS** – gives information about courses to which students registered and their grades. Also students can provide rating to a course and provide feedback.
22. **STUDENT_GRADING** – contains grading scale (ex A, B, C, D, F)
23. **RATING** – contains rating scale (ex 0, 1, 2, 3, 4, 5)
24. **ASSIGNMENT_SUBMISSION** – holds assignment submission details.
25. **TA_RELATED_TO_COURSES** – Gives list of TAs who are related to each course offered in each term of the year and their permission types.

26. **GRADER_RELATED_TO_COURSES** - Gives list of GRADERs who are related to each course offered in each term of the year and their permission types.
27. **TEACHER_RELATED_TO_COURSES** - Gives list of TEACHERs who are related to each course offered in each term of the year and their permission types.
28. **PERMISSION** – Contains list of permission Types given to TAs, Graders, and Teachers.
29. **INVOLVE** – is a bridge table between TEACHER entity and COURSE_INVOLVEMENT entity.
30. **DISCUSSION_BOARD_FOR_LECTURES** – It is a discussion forum for all the lectures where students and teachers can comment.
31. **DISCUSSION_BOARD_FOR_ASSIGNMENT** - It is a discussion forum for all the Assignments where students and teachers can comment.

IDENTIFYING RELATIONSHIPS:

Above ERD has 1:M, 1:1, M:N Relationships between the entities.

M:N relationship is decomposed into two 1:M relationships.

For example there is a M:N relationship between STUDENT and COURSE_OFFERING entities.

ENROLL is the bridge table used to decompose it.

Specialization Hierarchy

We have hierarchy in our design. USERS is **Supertype** (Parent Entity) and Student, TA, Grader, and Teacher are **Subtypes** (Child Entities).

STUDENT -> USERS: Student is a User (is-a Relationship)

TA -> USERS: TA is a User (is-a Relationship)

GRADER -> USERS: Grader is a User (is-a Relationship)

TEACHER -> USERS: Teacher is a User (is-a Relationship)

OPTIONAL ATTRIBUTES:

1. STUDENT_COURSE_DETAILS entity has Feedback as Optional Attribute. It can be NULL.
2. ASSIGNMENT entity has Attached Files as Optional Attribute.

Assumption: Few assignments may contain only description without any attachments.

TRADE-OFFs IN THE ABOVE DESIGN:

Made **PAYMENT_MODE** as different entity. Because it gives more flexibility to update different Payment Types without affecting the other entities.

Similarly PERMISSION, COURSE_TERM, RATING, STUDENT_GRADING are the new entities.

We may have more entities in the design phase but it is helpful post implementation.

Here disadvantage is, it creates more tables. I have made this trade-off to have **data modularity**.

MULTI-VALUED ATTRIBUTES:

1. Lecture Material type is a multi-valued attribute. A material may have video, PDfs, Links etc
2. COURSE_OFFERING entity has TA, GRADER, TEACHER for a Particular Course, but we have only USER_ID attribute to refer. I have created 3 new attributes 1. USER_ID_TEACHER 2. USER_ID_TA 3. USER_ID_GRADER.

ATTRIBUTES INCLUDED IN THE ERD:

Time Stamp is included in Discussion Entity which helps in sorting the comments.

Status Attribute - to check before giving certificate to student.

ATTRIBUTES MISSING IN THE ERD:

Certificate attribute is not present in the above ERD. Based on the status attribute in STUDENT_COURSE_DETAILS entity, we can get the certificate.

ASSUMPTIONS:

1. Student is a Learner.
2. Each User must belong to one of User Type mentioned. (STUDENT, TA, GRADER, TEACHER) – **Total Overlap**.
3. Every Course must have Materials, Lectures, and Assignments.
4. If the Course is offered in a particular term, there will a Course Creator for that.
5. At least one professor will be involved in a Course other than course creator.
6. A course should belong to at least one Category.
7. Course may present in multiple Categories.
8. Only Enrolled students can comment.
9. Student get certificate based on his Course Status. Whether he completed the course or not.
10. Every TA, GRADER, TEACHER should have permission type.
11. Student will get grade irrespective of his completion status.