



## REPORT OF E-LEARN ERD

E-learn is an e-learning platform. It consists of many entities and each entity holds certain rules and attributes which are listed below:

1. **PERSON** entity shares common attributes of learner, teacher, ta and grader. All four have different set of permissions. Here, I'm assuming that all four are disjoint that is a learner is a learner it can't be teacher, ta or grader for that course. This is a major trade off in the design. **PERMISSION** entity is created which holds permission for all four. All four possess different permission like grader can only grades assignments and ta can only assist in teaching.
  - Here, P\_ID is a primary key while P\_ROLE in this table will help to define the role of Person. When P-ID is used as a foreign key in other tables this P\_ROLE will help to identify the relation.
  - **Weak relationship** will exist between the two entities because permission is strong entity as it does not contain primary key of person entity as its own primary key.
  - Cardinality of Person(ALL) is (1:1) and permission is (1:1)
  - PERSON entity contains P\_ROLE to store identity of person whether it's a TA, GRADER, Learner or Teacher. P\_LNAME and P\_FNAME contains first and last name of the person. P\_EMAIL and P\_PASS contains email address and password of the person which is used in the registration process. P\_ADDRESS is multivalued attribute as it may contain many addresses.
2. E-learn consists of learners, learners can explore, enroll or study. I'm assuming that learner is user and he will enroll to study and get certificates.
  - **PERSON (Learner type)** entity have (M: N) relationship with **COURSE** entity via **ENROLL** bridge entity. A learner can enroll in many courses, for instance as a student I can enroll in as many subjects I need in first semester. Similarly, a course may contain many students.
  - **Strong relationship** will exist between the three entities because enroll is weak entity as it contains primary key of both entities as its own primary key.
  - Cardinality of Person(learner) is (1:1), Enroll is (0:M) and Course is (1:1).
  - COURSE entity contains C\_DESC to store description of course. C\_CREDIT to store credit of course. C\_TITLE to store title of the course and C\_PRICE to tell price of that course.

- ENROLL will contain ENROL\_DATE to store date of enrolment, ENROL\_AMOUNT to tell the amount the learner has to pay in order to enroll in particular course and ENROL\_RECEIPT which will show the payment for course enrollment.

3. **PERSON (TEACHER Type)** have (M: N) relation with **COURSE**. For instance: A teacher can create many courses while one course can be created by many teachers. There are many teachers who work to create DB and many subjects like DB or Big Data Analytics can be created by one teacher. Bridge entity **CREATION** is used to break (M: N) relation.

- **Strong relationship** will exist between the three entities because creation is weak entity as it contains primary key of both entities as its own primary key.
- Cardinality of Person(learner) is (1:1), Creation is (1:M) because every is created by at least one teacher and Course is (1:1).
- CREATION contain CREATE\_DATE to store date of creation of course.

4. **CERTIFICATE** entity will contain the total grades of enrolled learners. There is a threshold to obtain a certificate. Learner can view their grades in detail. If the grades are below threshold then certificate link attribute will be null and if it is above the threshold then learner can obtain their certificate by going on the link provided. Certificate is linked to enroll entity. Here, I am assuming two things a course is consists of only assignments no final exam and course is not completed if a learner left any graded assignment. The relationship between **Certificate** and **Enroll** will be **(1:1)** because a certificate can be of one learner and one course while one learner in one course may or may not get the certificate. For instance: I'm enrolled in DB course and completed all assignment and my total grade is also above threshold so I will get one certificate for DB. Similarly, I can get certificates for different courses. On the other hand, DB certificate can belong only to me.

- Relationship between certificate and enroll will be **weak** because enroll is a strong entity.
- The cardinality at certificate entity will be (0,1) while at enroll will be (1,1).
- CERTIFICATE entity also contains the course feedback because a learner can only collect his certificate when course is completed.

- CERTIFICATE entity contains CERT\_TOTGRADE to store total grade of assignments, CERT\_DATE to tell when certificate is created and CERT\_LINK that store link of certificate.
5. **PAYMENT** entity will contain primary key of enroll entity as its foreign key. There will **(1:M) one to many** relationships between payment and enroll. Payment entity only consists of payment methods through which learner can pay its amount. A learner can pay through many methods but one method payment is done by only one learner.
- I have created a separate entity for payment because it is clearly mentioned in the statement that the payment entity is flexible to add and delete. It is also handled from an outside unit. Outside unit will take primary key of this entity to make changes in the entity easily. Another trade off in the design.
  - Relationship between enroll entity and payment entity will be **weak** because payment is a **strong** entity as it does not contain enroll primary as its own primary key.
  - The cardinality at enroll entity will be (1,1) while at payment will be (1, M).
  - PAYMENT entity contains PAY\_NAME which store type of payment used and PAY\_WEBADD stores link on which payment is done.
6. A course can be presented in different categories and many courses can be grouped in one category. There exists **(M: N) relation** between **COURSE** and **CATEGORY**. A bridge entity called **PRESENT** is used to implement (M: N) relation. For instance, Data Analytics is offered in Computer Science, Business Analytics, Management and many more so it will be (1:M) relation between course and categories. A category (CS, BA) can have many different courses again (1:M) which will create (M: N) relationship. I have to use category entity because if I place category as an attribute of course entity then it creates redundancy.
- Between course and present **strong** relationship will exist as present entity contains primary key of course and category as its primary key. Similarly, **strong** relationship exists between present and category entity.
  - Cardinality of Course is (1:1), Present is (0:M) and Category is (1:1).

- CATEGORY contains CAT\_NAME, CAT\_TYPE which stores the name of category and type of category. PRESENT entity contains only PRE\_CBRANCH that stores the branch in which course is taught.

7. **ASSIGNMENT** entity is linked with **PERSON**(Learner) **through (M:M) relationship**. One learner can submit many assignment and one assignment can come from many learners. For instance: DB assignment will have submitted by 490 students and 490 students can submit DB as well as algorithm assignment. There are many students who do not submit algorithm assignment but they are enroll A bridge will be created to remove M:M relationship. A bridge of name called **SUBMISSION** is created.

- **Submission entity** contain SUB\_MAIL that learner will get after submitting. SUB\_GRADE that will store the grade of the assignment. SUB\_DATE will store date of submission and SUB\_FBACK will store learner's feedback regarding assignment.
- **Assignment entity** have ASS\_DESC that will contain description of the assignment. ASS\_FILES will have the assignment, ASS\_DEAD contains deadline of assignment, ASS\_NUMB counts number of assignment of a particular course.
- Cardinality of Assignment (1:1), submission will be (0:M) and Learner will be (1:1).
- **Strong relationship** will exist between the three entities because submission is weak entity as it contains primary key of both entities as its own primary key.

8. **MATERIAL** is linked with **LECTURE** entity. A **(1:M)** relationship is present between them. One material can contain many lectures but one lecture can contain only one material. For instance: CS585 course material have many lectures but one lecture of CS585 consists of only one type of material.

- **MATERIAL entity** contain MAT\_GENERAL it contains syllabus and textbook. It is a multivalued attribute and MATERIAL\_LINKS contains links only.
- **LECTURE** entity contain LECTURE\_VIDEOS, LECTURE\_NOTES and LECTURE\_FILES. They store videos, notes and files of lecture.
- **Weak relationship** will exist between the two entities because Material is strong entity as it does not contain primary key of lecture entity as its own primary key.

- Cardinality of Material is (1:M) and of Lecture is (1:1).
9. **COMMENT** entity will contain Person, course, lecture entity primary key as foreign key. Before describing relationship, I'm assuming one thing that same comments will not be posted. Similar as written in this assignment that we can post our queries but we should not post similar queries.
- **COMMENT** relationship with **PERSON** will be (1:M). As one learner either learner or teacher can either post answers or questions many or zero but one comment will be of only one learner and teacher. Relationship will be **weak** as comment entity does not contain primary key of learner as its own primary key. Cardinality of comment will be (0:M) and of person will be (1:1).
  - **COMMENT** relationship with **COURSE** will be (1:M). As one course can have many comments one comment will be of only one course. Relationship will be **weak** as comment entity does not contain primary key of course as its own primary key. Cardinality of comment will be (0:M) and of course will be (1:1).
  - **COMMENT** relationship with **LECTURE** will be (1:M). As one Lecture can have many comments one comment will be of only one lecture. Relationship will be **weak** as comment entity does not contain primary key of lecture as its own primary key. Cardinality of comment will be (0:M) and of lecture will be (1:1).
  - **COMMENT** entity contains COM\_QUES to store the question, COM\_ANS to store answer of the question and COM\_DATE to store date on which comment is posted.
10. **SCHEDULE** entity is used to depict the schedule of Person (teachers, ta, graders, learner), course content, assignments. Schedule entity will link with all the given entity.
- **SCHEDULE** relationship with **PERSON** teachers, ta, graders, students will be (M: N). A learner, TA, Grader, Teacher can appear in many schedule and one schedule may contain many of them. **TIMELINE** bridge entity is used to break them into (1:M) relationship. The relationship between them is strong as TIMELINE is a weak entity. Cardinality of Person is (1:1), Timeline is (0:M) and Schedule is (1:1)
  - **SCHEDULE** relationship with **COURSE** is (1:M). As one course may be present in many schedules but one schedule contains one particular type of course. Like DB is offered in Fall and Spring but in Fall one DB is offered. **Weak** relationship will exist between them as

Schedule does not contain primary key of course as its own primary key. Cardinality of Course will be (1:1) and of schedule will be (0:M).

- **SCHEDULE** relationship with **ASSIGNMENT** is **(1:M)**. As many assignments may present in one schedule. Assuming here that assignment changes in each schedule. But each assignment belongs to one schedule only. **Weak** relationship will exist between them as Schedule does not contain primary key of assignment as its own primary key. Cardinality of ASSIGNMENT will be (0:M) and of schedule will be (1:1).
- **SCHEDULE** relationship with **MATERIAL** is **(1:M)**. As many materials may present in one schedule. Assuming here that material changes in each schedule. But one material belongs to one schedule only. **Weak** relationship will exist between them as Schedule does not contain primary key of material as its own primary key. Cardinality of MATERIAL will be (0:M) and of schedule will be (1:1).
- **SCHEDULE** entity consists of start time of the schedule SCHE\_STARTTIME and end time of the schedule SCHE\_ENDTIME. **TIMELINE** bridge entity contains TIME\_SEMESTER that will store semester in which subject is offered.

## COMPONENTS OF E-LEARN ERD

ENTITY	RELATIONSHIP	CONNECTIVITY	ENTITY
PERSON	enrolls in	M: N	COURSE
PERSON	has	1:M	PERMISSION
PERSON	creates	M: N	COURSE
PERSON	submits	1:M	SUBMISSION
PERSON	schedules in	M: N	SCHEDULE
PERSON	can ask	1:M	COMMENT
ENROLL	receives	1:1	CERTIFICATE
ENROLL	pays via	1:1	PAYMENT
COURSE	presents	M: N	CATEGORY
COURSE	generates	1:M	ASSIGNMMENT
COURSE	schedules	1:M	SCHEDULE
COURSE	have	1:M	COMMENT
COURSE	consists of	1:M	MATERIAL
ASSIGNMENT	submits	1:M	SUBMISSION
MATERIAL	stores	1:M	LECTURE
LECTURE	gives rise to	1:M	COMMENT
ASSIGNMENT	appears in	1:M	SCHEDULE
LECTURE	presents in	1:M	SCHEDULE