HW1: ERD

STUDENT
- PK USER_ID
- USER_FNAME
- USER_LNAME

ENROLLMENT
- PK FK USER_ID
- PK SURROGATE_KEY
- PAYMENT_METHOD
- GRADE
- FEEDBACK

COURSE
- PK SURROGATE_KEY
- COURSE_ID
- USER_ID
- COURSE_STIME
- COURSE_ETIME
- COURSE_PRICE

EE
- PK COURSE_ID
- ...

CS
- PK COURSE_ID
- ...

FIN
- PK COURSE_ID
- ...

FACULTY
- PK FK SURROGATE_KEY
- PK USER_ID
- FACU_TYPE

CONTENT
- PK FK SURROGATE_KEY
- PK USER_ID
- PK CONTENT_ID
- PK MODIFY_DATE
- OPER_DETAIL

MATERIAL
- PK CONTENT_ID
- MAT_DETAIL

LECTURE
- PK CONTENT_ID
- LEC_DETAIL
- LEC_ID

ASSIGNMENT
- PK CONTENT_ID
- ASS_DESCRIPTION
- ASS_ID
- DEADLINE_DATE

ASS_GARDE
- PK FK ASS_ID
- PK USER_ID
- AGRADE

LEC_COMMENT
- PK LCOMMENT_ID
- LEC_ID
- USER_ID
- LCOMMENT_DETAIL

ASSIGN_COMMENT
- PK ACOMMENT_ID
- ASS_ID
- USER_ID
- ACOMMENT_DETAIL

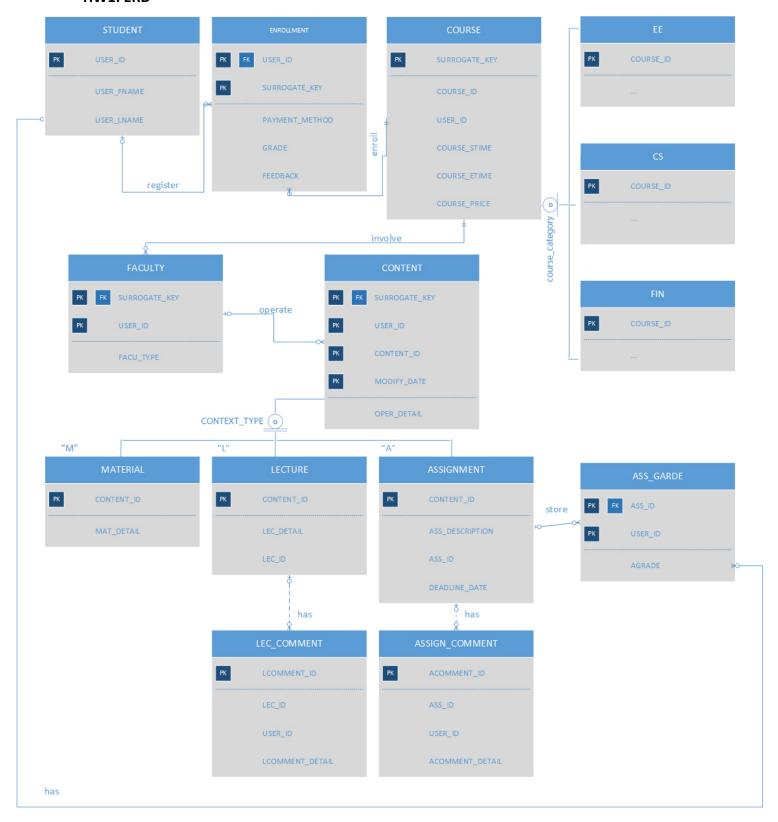Relationships: register, enroll, involve, operate, course_category, CONTEXT_TYPE ("M", "L", "A"), store, has, has, has

In the ERD above, I define that a user who owns a unique user id can be one or more roles as below:

Student, Faculty (Teacher, TA, Grader…)

For instance, a teacher can also be a student.

In my ERD, there are 14 tables in total:

STUDENT (PK: USER_ID)

ENROLLMENT (PK: (USER_ID, SURROGATE_KEY))

COURSE
(PK: SURROGATE_KEY,
   SECONDERY_KEY:
      (COURSE_ID,
       USER_ID,
       COURSE_STIME,
       COURSE_ETIME)
)

Generate a surrogate key to distinguish different courses or courses with same id but in different time periods. Teacher (USER_ID) creates this course.

FACULTY
(PK: (SURROGATE_KEY,USER_ID))

Faculty type (FACU_TYPE) includes Teacher, TA, Grader, etc.

CONTENT
(PK: (SURROGATE_KEY,USER_ID,CONTENT_ID,MODIFY_DATE))

Operation detail includes create, delete, modify, etc.

MATERIAL
(PK: CONTENT_ID)

Material detail can be a textbook, syllabus, etc.

LECTURE
(PK: CONTENT_ID)

Lecture detail can be videos, notes, and materials, etc.

ASSIGNMENT
(PK: CONTENT_ID)

Each assignment has a description (ASS_DESCRIPTION), attached files and a deadline (DEADLINE_DATE).

LEC_COMMENT
(PK: LCOMMENT_ID)

ASSIGN_COMMENT
(PK: ACOMMENT_ID)

ASS_GARDE
(PK:(ASS_ID,USER_ID))

EE
(PK:COURSE_ID)

CS
(PK:COURSE_ID)

FIN
(PK:COURSE_ID)

Relationships:

1) **STUDENT-ENROLLMENT-COURSE**
   In my design, table ENROLLMENT is the bridge table connects table STUDENT and table COURSE.
   For the relationship between student and enrollment is 1:N, which means a student can register multiple courses.
   For the relationship between course and enrollment is also 1:N, which means a course can be enrolled multiple times.

2) **COURSE-EE/CS/FIN**
   According to homework description, courses are grouped into multiple categories and a course may be presented in different categories. I use overlap and partial to describe the relationships between table COURSE with table CS, table EE and table FIN.

   Therefore, a course can exist in table COURSE, but not shown in CS, EE or FIN. In addition, a course can belong to CS, EE AND FIN at the same time, or two tables of them or one table of them.

3) **COURSE-FACULTY**
   The relationship between course and faculty is one to many. In that case, multiple teachers can get involved in a course. There are also other roles (TA, graders) who have different permissions on different course operations.

4) **FACULTY-CONTENT**
   Table Content is used to keep track of each content related to corresponding course, which can be added, deleted or modified.
   The relationship between faculty and content is one to many. Therefore, a faculty can operate with the same content multiple times.

5) **CONTENT-MATERIAL/LECTURE/ASSIGNMENT**
   According to homework description, some content may belong to both material and lecture. Therefore, I use overlap and total to describe the relationships between table CONTENT with table MATERIAL, table LECUTRE and table ASSIGNMENT.

6) **LECTURE-LEC_COMMENT**
   1:N weak relationship. One lecture can have zero or more comments.

7) **ASSIGNMENT-ASSIGN_COMMENT**
   1:N weak relationship. One assignment can have zero or more comments.

8) **ASSIGNMENT-ASS_GRADE-STUDENT**
   In my design, table ASS_GRADE is the bridge table connects table ASSIGNMENT and table STUDENT.
   For the relationship between assignment and assignment grade is 1:N, which means one assignment can have zero or more grade records.
   For the relationship between student and assignment grade is also 1:N, which means a student can have zero or more assignment grades.

For the categories (CS/EE/FIN), the attributes are not specified. They could include USER_ID to keep track each user's category preferences.