Online Course management system

❖ Introduction:

Welcome to our Educational Management System, a comprehensive platform designed to streamline and enhance the learning experience for both students and instructors. This system encompasses a robust database structure that efficiently manages users, courses, modules, lessons, and related content. By facilitating seamless enrollment processes and grade management, our system ensures a smooth and interactive educational journey. Our goal is to provide an intuitive and user-friendly environment that promotes effective learning and teaching practices. Join us as we explore the capabilities and benefits of this innovative solution.

❖ Entities of the Online course Management Systems

- **User:**

The User entity represents individuals who interact with the CMS. This includes instructors and students. Each user has attributes such as a unique ID, username, password, role (for example instructor, student) ,created_date and email address.

- **Course:**

The Course entity represents the courses offered within the CMS. It includes attributes such as a unique course ID, title, description, hours,crs_amount, created_date ,and the instructor who teaches the course.

- **Module:**

A Module entity represents sections or units within a course. Modules help to organize course content into manageable format. Attributes may include a unique module ID, title, description and the course to which the module belongs (Course_id).

- **Lesson:**

The Lesson entity refers to individual learning units within a module. It contains attributes such as a unique lesson ID, title, content (for example lecture notes, videos) and the module to which the lesson belongs (Module_id) .

- **Content:**

Content represents the learning materials associated with lessons. This may include documents, videos, presentations, quizzes and assignments. Attributes could include a unique content ID, title, type (for example document, video), file path and the lesson to which the content belongs(Lesson_id).

- **Enrollment:**

The Enrollment entity records the enrollment of students in courses. It includes attributes such as a unique enrollment ID, student ID, course ID and enrollment date.

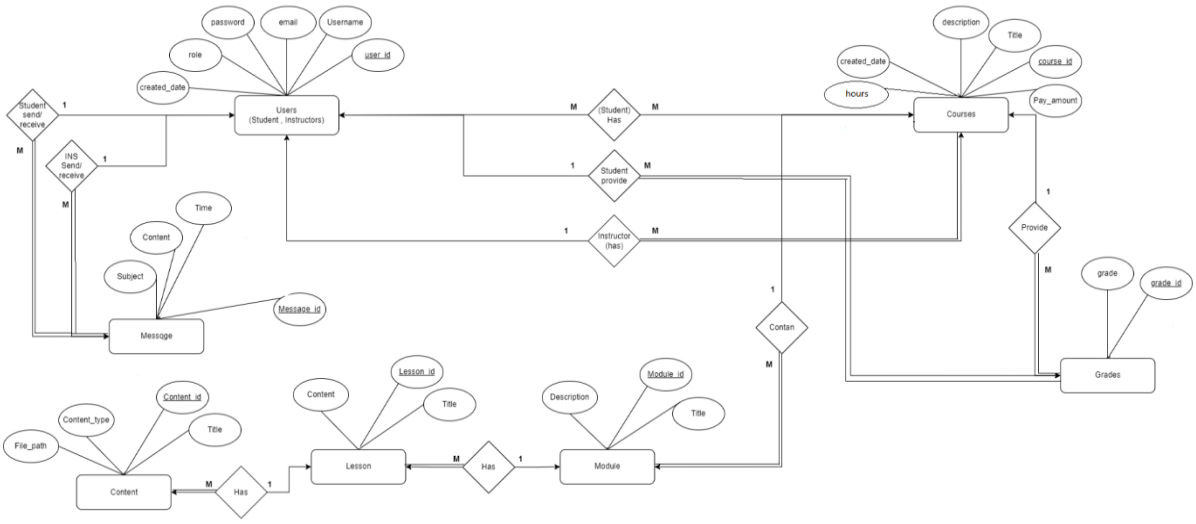
- **Grade:**

The Grade entity stores grades and assessment results for students. It includes attributes such as a unique grade ID, student ID, course ID and the actual grade or assessment result (grade).

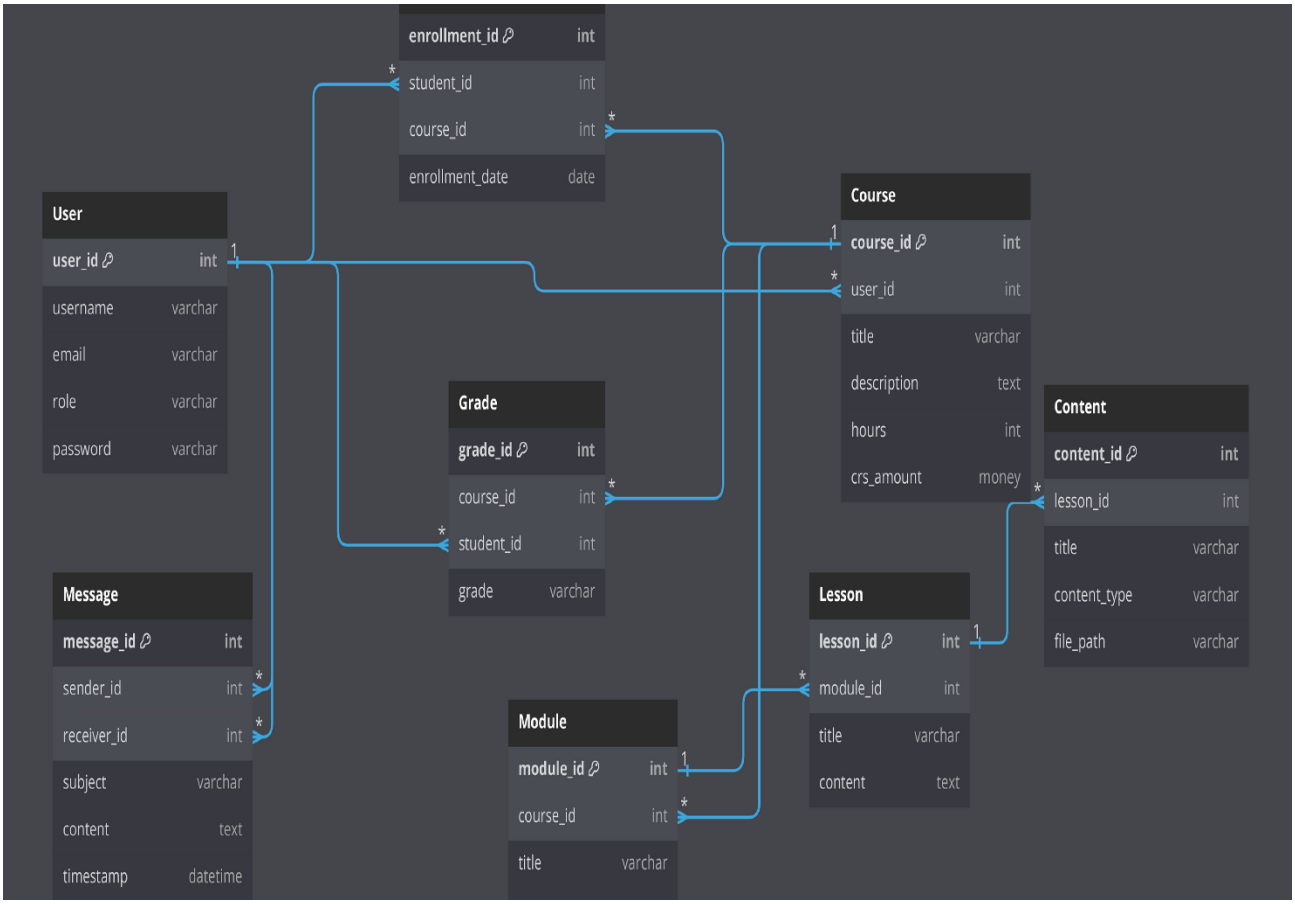
- **Message:**

The Message entity represents communication between users within the system. Attributes may include a unique message ID, sender ID, receiver ID, subject, content and timestamp.

❖ ERD


password
email
Username
role
user_id
created_date
Users
(Student , Instructors)
Student send/ receive
INS Send/ receive
Time
Content
Subject
Message_id
Message
(Student) Has
Student provide
Instructor (has)
description
Title
course_id
created_date
hours
Pay_amount
Courses
Provide
grade
grade_id
Grades
Contan
Lesson_id
Content
Title
Lesson
Has
Module_id
Description
Title
Module
Content_id
Content_type
Title
File_path
Content
Has

❖ Mapping:


enrollment_id int
student_id int
course_id int
enrollment_date date
User
user_id int
username varchar
email varchar
role varchar
password varchar
Course
course_id int
user_id int
title varchar
description text
hours int
crs_amount money
Grade
grade_id int
course_id int
student_id int
grade varchar
Content
content_id int
lesson_id int
title varchar
content_type varchar
file_path varchar
Message
message_id int
sender_id int
receiver_id int
subject varchar
content text
timestamp datetime
Lesson
lesson_id int
module_id int
title varchar
content text
Module
module_id int
course_id int
title varchar

❖ Topics

✓ **Views**

- View StudentsOnly() (to show All students only)
- View InstructorsOnly() (to show All Instructors only)
- View EnrolledCoursesOnly () (to show Enrolled Courses only)
- View NonEnrolledCoursesOnly() (to show Non- Enrolled Courses only)
- View InstructorsWithoutCourses() (to show Instructors Without Courses)
- View StudentsNotEnrolledInCourses() (to show Students Not Enrolled In any Courses)

✓ **stored procedure (Insert process)**

- Insert User (insert data to User table)
- InsertCourse (insert data to Course table)
- InsertModule (insert data to Module table)
- InsertLesson (insert data to Lesson table)
- InsertContent (insert data to Content table)
- InsertEnrollment (insert data to Enrollment table)
- InsertGrade (insert data to Grade table)
- InsertMessage (insert data to Message table)

✓ **stored procedure (Update process)**

- UpdateUser (update data in User table)
- UpdateCourse (update data in Course table)
- UpdateModule (update data in Module table)
- UpdateLesson (update data in Lesson table)
- UpdateContent (update data in Content table)
- UpdateEnrollment (update data in Enrollment table)
- UpdateGrade (update data in Grade table)

✓ **stored procedure (Delete process)**

- DeleteUser (delete data from User table)
- DeleteCourse (delete data from Course table)
- DeleteModule (delete data from Module table)
- DeleteLesson (delete data from Lesson table)
- DeleteContent (delete data from Content table)
- DeleteEnrollment (delete data from Enrollment table)
- DeleteGrade (delete data from Grade table)

- DeleteMessage (delete data from Message table)

✓ **Function:**

- GetTopNCoursesWithHighestSold (to get Top N Courses With Highest Sold)
- FUNCTION show_course_with_grade (show a specific course with it's grade)
- FUNCTION show_student_courses_and_grades (show his all courses and his grades)

✓ **Trigger:**

- Stop_deleting (stop all the delete process from enrollment table)
- Log_data (Store all the log information for the insertion on enrollment table)
- FUNCTION show_instructor_courses_and_enrollment_count (show all his courses and the count of student enrolled in)
- FUNCTION show_instructor_courses_with_highest_enrollment (show all his courses with the highest enrolled)
- FUNCTION show_instructor_courses_with_no_enrollments (show all his courses with no enrollments)
- FUNCTION search_instructors_by_name (show all the instrucors name with he searches for)
- FUNCTION show_instructor_courses_and_student_count (show all the courses to a spesific instructor)
- FUNCTION search_courses_by_name (Show all courses name with he searches for)
- get_course_details (show The Course data , course's modules , course's lessons and course's contents to a specific course)

✓ **Index: (**Indexes are used to make the search process Faster):

- IDX_User_Username (index on username in the User table)
- IDX_Course_Title (index on Title in the Course table)

✓ **Cursor**

- **GetUserAndGradesInCourse** (return all the student names and grades that enrolled in this course)

✓ **Rule**

- **Create Rule Grade_Values AS @values IN('A' , 'B' , 'C' , 'D' , 'F')** (putting a Rule to the Grade to make sure that what we will insert will be a real grade).

- ✓ **Constraints**
- **Alter Table [User] ADD CONSTRAINT check_role CHECK (role IN ('student' , 'instructor')** (Putting a constraint to make sure that we will have just a student or instructor in the User table)

❖ Business questions and answers:

Q1) how can we display students who are not enrolled in any courses in the online course system?

A) we created a view called "StudentsNotEnrolledCourses" that displays username and email when rule is "Student for students who are not enrolled in any courses.

Q2) how can the system display instructors who do not have any courses assigned to them?

A) we created a view called "InstructorsWithoutCourses" that displays username and email when rule is " instructor" instructors without any assigned courses.

Q3) how can the system display all instructors within the online course system?

A) we created a view called "InstructorsOnly" that displays username and email when rule is " instructor" for all instructors registered in the online course system.

Q4) how can the system display all students enrolled in the online course system?

A) we created a view called"StudentsOnly" that displays username and email rule when rule is " Students" for all students enrolled in the online course system.

Q5) how can the system show courses with the highest n of courses ordered by the count of students enrolled in?

A) we created a function called " GetTopNCoursesWithHighestSold " that take @TopN as a parameter and return TOP (@TopN) course_id, title, COUNT (c ourse_id) To fetches the top N courses with the highest number of students enrolled.

Q6) how can the system display courses where no students are currently enrolled?

A) we created a view called " NonEnrolledCoursesOnly " that desplay course_id, title,and description for courses it fetches all courses where no students are currently registered.

Q7) how the student will show a specific course with it's grade?

A) We created function called ‘show_course_with_grade’ that will take the student_id nd course_id and will return to you a table contains the course data.

Q8) how the student will show his all courses and his grades?

A) we created a function called ‘show_student_courses_and_grades’ that will take a student_id and returns a table contain all the courses and grades data.

Q9) how the instructor will show all his courses and the count of student enrolled in ?

A) we created a function called 'show_instructor_courses_and_enrollment_count’ that will take an instructor_id and return the table contain the data.

Q10) how the instructor will show all his courses with the highest enrolled?

A) we created a function called ‘show_instructor_courses_with_highest_enrollment’ that will take an instructor_id and will return a table contain the data

Q11) how the instructor will show all his courses with no enrollments?

A) we created a function called ‘show_instructor_courses_with_no_enrollments’ that will take an instructor_id and return a table contail the data

Q12) how the search bar will show all the instrucors name with he searches for?

A) we created a frunction called ‘search_instructors_by_name’ that will take the string that user search with and give it to this fuction to return a table caontain with all the names like his sting to make his search easer.

Q13) how the instructor will show all the courses to a spesific instructor?

A) we created a function called ‘show_instructor_courses_and_student_count’ that will take an instructor_id and retuen a table contain the data.

Q14) how the system will show all courses name with he searches for?

A) we created a function called ‘search_courses_by_name’ that will take a search_term and this is a string which user searches with, and then give you a table contain all the similar courses name like the search string.

Q15) how the system will show The Course data , course's modules , course's lessons and course's contents to a specific course?

A) we created a Function called ‘get_course_details’ that will take the course_id and return a table contain all the courses , modules, lessons and contain data to this course.