**----------------MOODLE ANALİZ----------------**

1-Write a brief explanation using your own words (in English) about these applications in terms  
of their scope::

Moodle (stands for, *modular object-oriented dynamic learning environment)* is a free open-source education management system widely used in distance education and online teaching systems. It was first developed by Martin Dougiamas, to help educators create and manage online courses, 19 years ago. Moodle has course specific customizable management features. Educators can modify their websites with variety of plugins and customization settings -such as themes- available. There are several roles of individuals can be in a Moodle course where each role provides access to a set of capabilities. We will be explaining them on later parts. Basically, system works between teachers-course assistants-students. On their site, teachers can post, make announcements, see students enrolled in the course, assign assignments, and rate them. Course assistant is a special role that can only be assigned to an already-enrolled user from higher hierarchy. Course assistants can access the assignments uploaded by students and can grade them, but they don’t have access to the gradebook and they can’t share posts like teachers. Students can participate in course activities and can download course material that are shared by teachers. Activities like assignments are submitted by one student only. This is the main logic behind the Moodle.

* **Site Administrator**
  + **This type of user can do anything.**
* **Manager**
  + **Is similar to site administrator but with a few tweaks**
* **Course Creator**
  + **This user creates courses. By default a course creator is the Teacher.**
* **Teacher**
  + **Can manage and edit the contents of courses**
* **Non-Editing Teacher**
  + **This type of user can grade courses but can’t do much else. Sort of like course assistans.**
* **Student**
  + **Students can access and participate in courses and performsa assigned tasks.**
* **Guest**
  + **Guests can view courses but they can’t be a part of them**
* **Authenticated User**
  + **This is a meta role. All logged in users have this role.**

2. Write an analysis report for each web application:  
a. What is the aim of each application?  
b. What are the main entities of them?  
c. What are the characteristics of each entity?  
d. What relationships exists among the entities?  
e. What are the constraints related to entities, their characteristics, and the relationships among them?

2/A:

Moodle is an online education platform that is designed for educators and learners, where you can build customizable learning environments, assign tasks, share data, participate in events and courses. It’s designed for both teaching and learning. This means that Moodle delivers a set of learner-centric tools and collaborative learning environments that empower both teaching and learning. It’s also very easy to use with simple interface and drag-drop features so any age of user with any experience can easily use this application.

2/B:

In actual Moodle database there are way more entities. But in this project, we will be focusing on these key entities:

**PERSON**

* + A person is the main user of this system.
* **ACCOUNT**
  + Every person has some kind of account.
* **COURSE**
  + This is where classes happen.
* **CONTENT**
  + Any informational material that is needed for participation or understanding classes.
* **QUIZ**
  + A test of knowledge. Series of questions that is asked to students.
* **ASSIGNMENT**
  + Long term homework given to students.
* **QUESTIONS**
  + Contents of a quiz or an assignment.
* **CERTIFICATE**
  + A certificate of achievement that you get for successfully completing all the tasks.
* **MESSAGAES**
  + Communication box between two people. (a person can send and receive messages- unary relationship)
* **FEEDBACK**
  + Students can send feedback to teachers about things that they don’t like.

2/C:

**PERSON:**

A person is the main user of this system. This person can be a teacher, student, assistant or administrator. Apart from the differences, each person has their username (first name, middle name initials, last name), date of birth, address, phone number, e-mail and a unique user id stored in this database.

* **Teacher:**

If this person is a teacher, their expertness, department ,salary and diploma information (diploma given date, college obtained) is stored.

* **Student:**

If this person is a student, their gpa, enrolled courses and their department is stored.

* **Assistant:**

If this person is an assistant, their thesis and diploma information is stored.

* **Administrator:**

If this person is an administrator, there is no extra information is stored.

Map

Description automatically generated

**ACCOUNT:**

Every user listed above must have some kind of an account. And every account has their person’s role and a unique account id stored in them.

Diagram

Description automatically generated

**COURSE:**

Every course has their credit hours, level of education, description and unique course code and course name kept in this database.

Diagram

Description automatically generated

**CONTENT:**

Materials that are used in the courses are called, course contents. It can be of different file types. A content has a file type, availability date range (upload date and self-destruct date), and a unique content no.

Diagram

Description automatically generated

**EVALUATION:**

Evaluation is done by QUIZ and ASSIGNMENTS. Every Quiz is graded, has a number of questions and has it’s unique quiz no. Every Assignment is also graded and has a number of questions. They also have a user’s number of attempts, due date and a unique assignment id. And Evaluation has a feedback process as well.

Diagram

Description automatically generated

**QUESTIONS:**

Every Evaluation process is made out of questions. Every question entity has a question type.

Diagram

Description automatically generated

**CERTIFICATE:**

A certificate is given after a student achieves necessary amount achievements for a course. A certificate has given date, course name and a unique certificate id.

Diagram

Description automatically generated

**FORUM:**

Users share their ideas and ask and answer each other’s posts. Every user has to post with their user id and every post has it’s unique post id.

A picture containing text, athletic game

Description automatically generated **MESSAGES:**

It’s a communication process between two people. A person can both send and receive messages. Every message has information about both sender and receiver and time sent.

Diagram

Description automatically generated

2/D

A person must have an account and an account must be bound to one person only. A person can message with another person as well. A person can share posts in forums and forums may have multiple people posting on them. If this person is a teacher, they may be giving more than one course, but a course must have only one teacher. A teacher and assistants can upload contents to courses, naturally they can upload more than one content per course. Students can use these contents via downloading them. Accounts earn certificates. A certificate may be possessed by more than one person and a person may achieve more than one certificate. Administrators provide support and they contact with other group of people.

Teachers will make evaluations for students. So, a course must have at least one of these evaluation tasks. And students are evaluated. Quizzes and assignments can be graded by both teacher and assistants. Both of these evaluation elements are made of multiple questions. These evaluation tasks are answered by students. A student may have more than one evaluation per course. Evaluations grant students, certificates. After the exams are done, students get to give feedback about them. A teacher will receive these feedbacks.

PERSON 1 HAS 1 ACCOUNT

PERSON M EARNS M CERTIFICATE

PERSON 1 MESSAGES 1 PERSON

PERSON M POSTS ON M FORUM

TEACHER 1 GIVES M COURSE

TEACHER 1 DOES M EVALUATIONS

TEACHER 1 UPLOADS M CONTENT

TEACHER 1 GRADES M EVALUATIONS

TEACHER 1 RECIEVES M FEEDBACK

ASSISTANT 1 GRADES M EVALUATIONS

ASSISTANT 1 UPLOADS M CONTENT

STUDENTS 1 TAKE M EVALUATIONS

STUDENTS M USE M CONTENT

STUDENTS M ENROLL TO M COURSE

STUDENTS 1 GIVE M FEEDBACK

ADMINISTRATORS M SUPPORT M PERSON ?????

EVALUATIONS M GRANTS 1 CERTIFICATE

EVALUATIONS M HAS M QUESTIONS

COURSE CONTAINS CONTENT

COURSE HAS EVALUATIONS

2/E

Constraints:

Users:

CONSTRAINT pk\_USERS primary key (UserID),

CONSTRAINT uq\_EMail UNIQUE (EMail)

PhoneNumber:

CONSTRAINT pk\_USER\_PHONE primary key (PhoneUsersID, PhoneNumber),

CONSTRAINT pf\_USER\_PHONE\_USERS foreign key (PhoneUsersID) references USERS(UserID)

Address:

CONSTRAINT pk\_USER\_ADDRESS primary key (AddressUsersID, Address),

CONSTRAINT pf\_USER\_ADDRESS\_USERS foreign key (AddressUsersID) references USERS(UserID)

Person:

CONSTRAINT pk\_PERSON primary key (PersonUserID),

CONSTRAINT fk\_PERSON\_USERS foreign key (PersonUserID) references USERS(UserID)

Messages:

CONSTRAINT pk\_MESSAGES primary key (SenderUserID, RecieverUserID),

CONSTRAINT fk\_MESSAGES\_PERSON1 foreign key (SenderUserID) references PERSON(PersonUserID),

CONSTRAINT fk\_MESSAGES\_PERSON2 foreign key (RecieverUserID) references PERSON(PersonUserID)

Administrator:

CONSTRAINT pk\_ADMINISTRATOR primary key (AdminID),

CONSTRAINT fk\_ADMINISTRATOR\_USERS foreign key (AdminID) references PERSON(PersonUserID)

Teaching Staff:

CONSTRAINT pk\_TEACHING\_STAFF primary key (TeachingUserID),

CONSTRAINT fk\_TEACHING\_STAFF\_USER foreign key (TeachingUserID) references PERSON(PersonUserID)

Student:

CONSTRAINT pk\_STUDENT primary key (StudentUserID),

CONSTRAINT fk\_STUDENT\_USERS foreign key (StudentUserID) references Person(PersonUserID)

Teacher:

CONSTRAINT pk\_TEACHER primary key (TeacherUserID),

CONSTRAINT fk\_TEACHER\_USER foreign key (TeacherUserID) references TEACHING\_STAFF(TeachingUserID)

Feedback:

CONSTRAINT pk\_FEEDBACK primary key (FeedbackID),

CONSTRAINT fk\_FEEDBACK\_STUDENT foreign key (SendingStudentUserID) references STUDENT(StudentUserID),

CONSTRAINT fk\_FEEDBACK\_TEACHER foreign key (RecievingTeacherUserID) references TEACHER(TeacherUserID)

Assistant:

CONSTRAINT pk\_ASSISTANT primary key (AssistantUserID),

CONSTRAINT fk\_ASSISTANT\_USERS foreign key (AssistantUserID) references TEACHING\_STAFF(TeachingUserID)

Course:

CONSTRAINT pk\_COURSE primary key (CourseCode),

CONSTRAINT fk\_COURSE foreign key (CourseTeacherUserID) references TEACHER(TeacherUserID)

Content:

CONSTRAINT pk\_CONTENT primary key (ContentNo),

CONSTRAINT fk\_CONTENT\_TEACHING\_STAFF foreign key (UploaderUserID) references TEACHING\_STAFF(TeachingUserID)

Uploads:

CONSTRAINT pk\_UPLOADS primary key (UppingTeachingStaffUserID),

CONSTRAINT fk\_UPLOADS\_TEACHING\_STAFF foreign key (UppedContentNo) references TEACHING\_STAFF(TeachingUserID)

Evaluation:

CONSTRAINT pk\_EVALUATION primary key (EvaluationID, TakingStudentUserID, MakerUserID, CourseID, GraderUserID),

CONSTRAINT fk\_EVALUATION\_TEACHING\_STAFF1 foreign key (MakerUserID) references TEACHING\_STAFF(TeachingUserID),

CONSTRAINT fk\_EVALUATION\_TEACHING\_STAFF2 foreign key (GraderUserID) references TEACHING\_STAFF(TeachingUserID),

CONSTRAINT fk\_EVALUATION\_COURSE foreign key (CourseID) references COURSE(CourseCode),

CONSTRAINT fk\_EVALUATION\_STUDENT foreign key (TakingStudentUserID) references STUDENT(StudentUserID)

Assignment:

CONSTRAINT pk\_ASSIGNMENT primary key (ATakingStudentUserID, AEvaluationID),

CONSTRAINT fk\_ASSIGNMENT\_STUDENT foreign key (ATakingStudentUserID) references STUDENT(StudentUserID),

CONSTRAINT fk\_ASSIGNMENT\_EVALUATION foreign key (AEvaluationID) references EVALUATION(EvaluationID)

Quiz:

CONSTRAINT pk\_QUIZ primary key (QTakingStudentUserID, QEvaluationID),

CONSTRAINT fk\_QUIZ\_STUDENT foreign key (QTakingStudentUserID) references STUDENT(StudentUserID),

CONSTRAINT fk\_QUIZ\_EVALUATION foreign key (QEvaluationID) references EVALUATION(EvaluationID)

Enrolls:

CONSTRAINT pk\_ENROLLS primary key (EnrolledCourseCode, EnrollingStudentUserID),

CONSTRAINT fk\_ENROLLS\_COURSE foreign key (EnrolledCourseCode) references COURSE(CourseCode),

CONSTRAINT fk\_ENROLLS\_STUDENT foreign key (EnrollingStudentUserID) references STUDENT(StudentUserID)

Take:

CONSTRAINT pk\_TAKE primary key (TStudentUserID, TEvaluationID, EMakerID, EGraderID, EvalTakenCourseID),

CONSTRAINT fk\_TAKE\_STUDENT foreign key (TStudentUserID) references STUDENT(StudentUserID),

CONSTRAINT fk\_TAKE\_EVALUATION foreign key (TEvaluationID) references EVALUATION(EvaluationID),

CONSTRAINT fk\_TAKE\_TEACHING\_STAFF1 foreign key (EMakerID) references TEACHING\_STAFF(TeachingUserID),

CONSTRAINT fk\_TAKE\_TEACHING\_STAFF2 foreign key (EGraderID) references TEACHING\_STAFF(TeachingUserID),

CONSTRAINT fk\_TAKE\_COURSE foreign key (EvalTakenCourseID) references COURSE(CourseCode)

Certificate:

CONSTRAINT pk\_CERTIFICATE primary key (CertificateID),

CONSTRAINT fk\_CERTIFICATE\_COURSE foreign key (CertificateCourseID) references COURSE(CourseCode),

CONSTRAINT fk\_CERTIFICATE\_STUDENT foreign key (CertificatedStudentUserID) references STUDENT(StudentUserID),

CONSTRAINT fk\_CERTIFICATE\_EVALUATION foreign key (CertificatingEvaluationID) references EVALUATION(EvaluationID),

CONSTRAINT fk\_CERTIFICATE\_TEACHING\_STAFF1 foreign key (EMakerID) references TEACHING\_STAFF(TeachingUserID),

CONSTRAINT fk\_CERTIFICATE\_TEACHING\_STAFF2 foreign key (EGraderID) references TEACHING\_STAFF(TeachingUserID)

Earns:

CONSTRAINT pk\_EARNS primary key (EarningPersonUserID, EarnedCertificateID),

CONSTRAINT fk\_EARNS\_PERSON foreign key (EarningPersonUserID) references PERSON(PersonUserID),

CONSTRAINT fk\_EARNS\_CERTIFICATE foreign key (EarnedCertificateID) references CERTIFICATE(CertificateID)

Questions:

CONSTRAINT pk\_QUESTIONS primary key (QuestionID, ExamEvaluationID),

CONSTRAINT fk\_QUESTIONS\_EVALUATION foreign key (ExamEvaluationID) references EVALUATION(EvaluationID)