



HACETTEPE
University

Computer Science Engineering Department

Group Name&Surname: Mehmet Taha USTA & Burcu ÖZTAŞ

Identity Numbers: 21527472 & 21483435

Course: BBM-473 Database Management

Subject: Online Education System

Due Date: 05/04/2020 23:59

Main Idea

In this project, "Online Education System", Which is popular in recent days, has been designed. For the database project, The properties of multiple interrelated entities have been designed as tabular and relational. The contents of our project have been drawn as E / R diagram.

All the tables and their qualities on the diagram are verbally explained below.

TABLES

1. Account : Account records are stored.

- **AccountID:** Registered users id, Primary Key, Unique
- **Password:** Account Passwords
- **UserName:** Account Name, Unique
- **E-mail:** Account Email, Unique
- **CreatedDate:** The date the Account was created
- **Name:** Account Name
- **Surname:** Account Surname
- **Birthday:** Account Birthday
- **Phone:** Account Phone
- **Gender:** Account Gender

2. Support: Support records are stored.

- **SupportID:** Support ID, Primary Key, Unique
- **AccountID:** Account ID, Foreign Key
- **AdminID:** Administrator ID, Foreign Key
- **SenderName:** Account Name
- **Type:** Account Type, categorizing
- **Content:** Content, message
- **E-Mail:** Account E-Mail for communication

3. Learner: Learner records are stored.

- **LearnerID:** Account ID, Primary Key, Unique
- **Department:** Account Department
- **CourseCode:** Represents courses taken by the Account.

Take: A relationship between entity sets Learner and Exam. Learner takes Exam.

Give: A relationship between entity sets Learner and Feedback. Learner gives feedback.

Enroll: A relationship between entity sets Learner and Course. Learner enroll Course

Browse: A relationship between entity sets Learner and Content. Learner Browse Content

Contact: A relationship between entity sets Learner and Support. Learner Contact Support

Has: A relationship between entity sets Learner and Account. Learner has a account

4. **Exam:** Exam result records are stored.

- **ExamID:** Exam ID, Primary Key, Unique
- **Midterm1:** Learner Midterm1
- **Midterm2:** Learner Midterm2
- **Midterm3:** Learner Midterm3
- **Final:** Learner Final

Give: A relationship between entity sets Exam and Feedback. Exam gives a Feedback.

Provide/Grade: A relationship between entity sets Exam and The Exam Subsystem.

5. **Feedback:** Feedback records are stored.

- **ExamID:** ExamID, Foreign Key
- **LearnerID:** LearnerID, Foreign Key
- **Point:** Learner Point
- **Comment:** Learner Comment

Send: A relationship between entity sets Feedback and Instructor. Feedback send information to instructor.

6. **Course:** Course records are stored.

- **CourseID:** Course ID, Primary Key, Unique
- **CourseCode:** Course Code, Unique
- **Status:** Course Status. Active or not
- **Name:** Course Name
- **Description:** Description for course
- **CreatedDate:** The date the Course was created
- **CreatedBy:** Course creator

Has: A relationship between entity sets Course and Content.

Has: A relationship between entity sets Course and Statistic.

7. **Content:** Content records are stored.

- **ID:** Content ID, Primary Key, Unique

- **Type:** Content Type

8. **Administrator:** Administrator records are stored.

- **AdminID:** Admin ID, Primary Key, Unique

Provide: A relationship between entity sets Administrator and Support. Administrator deals with records on Support.

Has: A relationship between entity sets Administrator and Account. Administrator has a account.

Accept/Reject: A relationship between entity sets Administrator and Content. Admin accept or reject content.

9. **Instructor:** Instructor records are stored.

- **InstructorID:** Instructor ID, Primary Key, Unique
- **Department:** Instructor Department
- **Activation:** Instructor Activation. Account can be Instructor if activated.

Use: A relationship between entity sets Instructor and the Exam subsystem. Instructor use the Exam subsystem.

Create: A relationship between entity sets Instructor and Course. Instructor can create course.

Contact: A relationship between entity sets Instructor and Support. Instructor can contact support.

Has: A relationship between entity sets Instructor and Account. Instructor has a Account.

Uploading Material: A relationship between entity sets Instructor and Uploading Material. Instructor can upload material.

10. **The Exam Subsystem:** Exam records are stored.

- **ExamID:** ExamID, Primary Key, Unique
- **ExamName:** ExamName

Grant: A relationship between entity sets The Exam Subsystem and Certificate. The Exam Subsystem Grant Certificate.

11. **Certificate:** Certificate records are stored.

- **CertificateID:** CertificateID, Primary Key, Unique
- **LearnerID:** LearnerID, Foreign Key

- **InstructorID:** InstructorID, Foreign Key
- **CourseID:** CourseID, Foreign Key
- **CertificateDate:** The date the Certificate was created

Send: A relationship between entity sets Certificate and Learner. Certificate table sends certificate to Learner.

12. **Upload:** Uploaded material records are stored.

- **UploadID:** UploadID, Primary Key, Unique
- **CourseCode:** CourseCode, Foreign Key
- **UploadDate:** The date the material was uploaded
- **UploadBy:** Which instructor uploaded the material
- **UploadFile:** UploadFile

13. **Statistic:** Statistic records are stored.

- **CourseCode:** Shows which course your statistics belong to
- **DisplayCount:** Shows count number
- **AverageOfPoint:** Shows average of point
- **Date:** The date the statistic was created

