

2021-2022 database management term project



LINKEDIN-MOODLE DATABASE SYSTEM

05180000096 ÖZDAL CAN SAĞLAM

05180000064 BUSE ARGUŞ

05200000002 MERT AKÇAY

05190000074 PELİN DOYUK

SCOPE OF APPLICATIONS

*1)MOODLE*

Moodle is an education management system used by universities. Since Moodle aims to create a virtual university environment, its scope is broad.

Teachers and students can create profiles on Moodle and log into their school's system. Teachers can create and manage the pages of the courses they give, share lecture notes, give quizzes and homeworks, and upload various resources and files related to the course.

Students enrolled in the course can access these course pages, files and documents with their code/password provided by their teachers. They can also upload their assignments/quizes to the fields opened by their teachers on the same course page.

Teachers can also assign various projects related to the courses; and the management, follow-up and delivery of these projects can be done on the course page.

*2)LINKEDIN*

Linkedin is one of the most important platforms that ensures that job seekers do not miss this digital development in an increasingly digital world.

Many different people and organizations such as job seekers, employers, human resources, companies, universities use Linkedin regardless of their job type. It is one of the best ways to find a job, to see the needs of the sector in which the job is sought, and to improve oneself in that field.

Linkedin is basically a social networking platform. You can use Linkedin to individually create and share resumes, search for jobs, establish business contacts, and share posts. As an organization, you can create posts about jobs, and gather followers. Linkedin also offers features such as following, messaging, sharing, liking, and commenting like other social media platforms.

Every user has a profile on Linkedin and this profile is your CV. You can write about yourself, add your education life and your position, share your projects and skills. You can be a reference for each other with other users.

## ANALYSIS

# 1) MOODLE

The aim of Moodle is to provide an easily understandable and useful platform for students and teachers to share, follow and manage courses.

## *1.1) ENTITIES*

**INSTRUCTOR**

* Instructors are one of the main users.
* Each instructor has a unique instructor number, name, surname, e\_mail, phone\_number, major, rank.
* Each instructor has specializes on a certain field which is their major.
* Each instructor has a rank in their university.
* Instructors have departments they work for.

**STUDENT**

* Students are one of the main users.
* Each student has a unique student number, name, surname, e\_mail, phone\_number, GPA.
* Each student has a GPA indicating their current point in their department.
* Students have departments they study in.

**UNIVERSITY**

* Each university has a unique id, name, website and address.
* Each university has a particular faculty member who is the rector of the university.

**DEPARTMENT**

* The university is organized into academic departments.
* Each department has a unique id, name, office and phone.
* Each department has a particular faculty member who is the dean of the department.

**COURSE**

* Each department offers courses.
* Each course has a unique id, name and credit.

**SECTION**

* Courses are offered as sections.
* Each section is related to a single course and a single instructor.
* Each section has unique id, year, classroom, semester and days/times.

**PROJECT**

* Some courses may have require projects.
* Each project is related to a single section.
* Each project has a unique id, name, description, start date and end date.

**QUIZ**

* Each quiz is related to a single section.
* Each quiz has a unique id and a questions path that indicates where the question file is kept.

## *1.2) RELATIONSHIPS*

**BELONGS TO between UNIVERSITY and DEPARTMENT:**

* Departments belong to universities.
* A university can have many departments but each department belongs to a single university.

**EMPLOYS between INSTRUCTOR and DEPARTMENT:**

* Departments employ instructors.
* A department can employ many instructors but each instructor belongs to a single department.

**DEAN between INSTRUCTOR and DEPARTMENT:**

* Instructors could be deans of departments.
* A department must have one dean and an instructor could be the dean of only one department.

**RECTOR between INSTRUCTOR and UNIVERSITY:**

* Instructors could be rectors of universities.
* A university must have one rector and an instructor could be the rector of only one department.

**SUPERVISES between INSTRUCTOR and PROJECT:**

* Instructors supervise projects.
* An instructor can supervise many projects but each project is supervised by a single instructor.

**TEACHES between INSTRUCTOR and SECTION:**

* Instructors teach sections.
* An instructor teach many sections but each section is related to a single instructor.

**TAKES between STUDENT and SECTION:**

* Students take sections.
* A student can take many sections and a section could be taken by many students.

**TAKES between STUDENT and QUIZ:**

* Students take quizzes.
* A student can take many quizzes and a quiz could be taken by many students.

**WORKS ON between STUDENT and PROJECT:**

* Students work on projects.
* A student can work on many projects and a project could be done by many students.

**SECS between COURSE and SECTION:**

* Courses secs into sections.
* A course can divide into many sections but each section is related to a single course.

**HAS between SECTION and QUIZ:**

* Sections have quizzes.
* A section can have many quizzes but a quiz belongs to a single section.

**HAS between DEPARTMENT and STUDENT:**

* Departments have students.
* A department can have many students but each student belongs to a single department.

**OFFERS between DEPARTMENT and COURSE:**

* Departments offer courses.
* A department can offer many courses but each course belongs to a single department.

## *1.3) CONSTRAINTS*

* Students and instructors must have an account to use the system. In order to create an account, they must add their e-mail or phone number information to the system. A **STUDENT** or **INSTRUCTOR** cannot exist without an **e-mail** or a **phone number**.
* All sections are related to a single course and a single instructor. A **SECTION** cannot exist without a course and instructor.
* Courses belong to particular departments and are offered by only those departments. A **COURSE** cannot exist without a department.
* All projects belong to a single section. A **PROJECT** cannot exist without a section.
* All quizzes belong to a single section. A **QUIZ** cannot exist without a section.

# 2) LINKEDIN

The aim of Linkedin is to bring employers and job seekers together. Users apply to jobs given by companies, employers revies the CV’s on their profile pages created by job seekers, and the process proceeds. Linkedin offers all the facilities to do these operations.

## *2.1) ENTITIES*

**USER**

* This entity represents the individual users of the application.
* Each Linkedin user has a unique user id, name, surname, e-mail, phone number. Users can use their e-mail and/or phone number for authentication and authorization, and to comminucate with each other.
* Some users have companies they work for.

**CV (Curriculum Vitae)**

* Each user has a personal profile page and this is their CV.
* Each CV has a unique CV id.
* For each user; name, surname, current position, education, about, projects and skills are displayed on CV’s.
* Education keeps the last graduated school type.
* Users can write about themselves in about field.
* Users can know any number of skills, and made any number of projects.

**POST**

* Users can create post to show on their pages and interact with others’.
* Each post has unique post id, post title and post content.

**ORGANIZATION**

* Various organizations can use Linkedin.
* Each organization has a unique id, name, website and address.

**COMPANY**

* Companies use Linkedin to look for employees.
* Each company has a unique id, name, website and address.
* Each company has a manager.

**JOB**

* Companies hire people and give jobs.
* Each job as a unique id, name, description, seniority level, industry, skills.
* Each job requires some skills.

**SKILL**

* People has skills and use them to get jobs.
* Each skill has a unique id, name and category.

## *2.2) RELATIONSHIPS*

**CREATES between USER and POST:**

* Users create posts.
* A user can create many posts and a post must be created by users.

**COMMENTS ON between USER and POST:**

* Users comment on posts.
* A user can comment on many posts and a post can be commented on by many users.

**SHARES between USER and POST:**

* Users share posts.
* A user can share many posts and a post can be shared by many users.

**LIKES between USER and POST:**

* Users like posts.
* A user can like many posts and a post can be liked by many users.

**CREATES between ORGANIZATION and POST:**

* Organizations create posts.
* A user can create many posts and a post must be created by users.

**FOLLOWS between USER and ORGANIZATION:**

* Users follows organizations.
* A user can follow many organizations and an organization can be followed by many users.

**WORKS IN between USER and COMPANY:**

* Users work in companies.
* A company can employ many users but a user must work in a single company.

**KNOWS between USER and SKILL:**

* Users know skills.
* A user can know many skills and a skill can be known by many users.

**APPLIES TO between USER and JOB:**

* Users apply to jobs.
* A user can apply to many jobs and a job can be applied to by many users.

**REQUIES between JOB and SKILL:**

* Jobs require skills.
* A job can require many skills and a skill can be a requirement to many jobs.

**OFFERS between COMPANY and JOB:**

* Companies offer jobs.
* A company can offer many jobs but a job must be offered by a single company.

**CONNECTS between USERS:**

* Users connect with users.
* A user can make connections with many other users.

**REFERS TO between USERS:**

* Users refer to users.
* A user can be a reference to many other users.

**MESSAGES between USERS:**

* Users message to users.
* A user can message to many other users.

**HAS between USER and CV:**

* Users has CVs.
* A user must have one CV and a CV must belong to only one user.

**ADMINISTERS between USER and COMPANY:**

* Users administer companies.
* A user could administer one company and a company must belong to only one user.

## *2.3) CONSTRAINTS*

* Users must have an account to use the system. In order to create an account, user must add his/her e-mail or phone number information to the system. A **USER** cannot exist without an **e-mail** or a **phone number**.
* After the account is created, the user must create his/her personal page. This personal page is user’s CV. A **CV** has to belong to a user.
* Individuals and organizations can create and share posts. A **POST** cannot exist without a user or an organization that created it.
* Jobs are given by companies. A **JOB** cannot exist without a company.
* Users manage companies. A **COMPANY** cannot exist without a manager.

## DESIGN – CONCEPTUAL DESIGN

\*This diagram is available in the attached "LinkedinMoodle\_EER.drawio" and "LinkedinMoodle\_EER.png" files.

## DESIGN – LOGICAL MODEL

# Converting EER Diagram into Relational Model:

## *STEP 1: Mapping Regular Entities*

* USER (USER\_ID, NAME, SURNAME, E\_MAIL, PHONE\_NUMBER, USER\_TYPE)
* CV (CV\_ID, POSITION, ABOUT, PROJECTS, EDUCATION)
* POST (POST\_ID, POST\_TITLE, POST\_CONTENT)
* SKILL (SKILL\_ID, SKILL\_NAME, CATEGORY)
* JOB (JOB\_ID, JOB\_NAME, SENIORITY\_LEVEL, INDUSTRY, JOB\_DESCRIPTION)
* DEPARTMENT (DEPT\_ID, DEPT\_NAME, DEPT\_OFFICE, DEPT\_PHONE)
* ORGANIZATION (ORGANIZATION\_ID, WEBSITE, ADDRESS, ORGANIZATION\_TYPE)
* COURSE (COURSE\_CODE, COURSE\_NAME, CREDIT)

## *STEP 2: Mapping Weak Entities*

## *STEP 2.1:*

* SECTION (INSTRUCTOR\_NUMBER, COURSE\_CODE, SEC\_ID, YEAR, SEMESTER, CLASSROM, SEC\_DAYS)

## *STEP 2.2:*

* QUIZ (INSTRUCTOR\_NUMBER, COURSE\_CODE, SEC\_ID, QUIZ\_ID, QUESTION\_PATH)
* PROJECT (INSTRUCTOR\_NUMBER, COURSE\_CODE, SEC\_ID, PROJECT\_CODE, PROJECT\_DESCRIPTION, PROJECT\_NAME, START\_DATE, ESTIMATED\_END\_DATE)

## *STEP 3: Mapping 1:1 Relations*

* CV (CV\_ID, POSITION, ABOUT, PROJECTS, EDUCATION, USER\_ID)
* COMPANY (COMPANY\_ID, COMPANY\_NAME, MANAGER\_ID)
* UNIVERSITY (UNIVERSITY\_ID, UNIVERSITY\_NAME, RECTOR\_ID)
* DEPARTMENT (DEPT\_ID, DEPT\_NAME, OFFICE, PHONE, DEAN\_ID)

## *STEP 4: Mapping 1:N Relations*

*STEP 4.1:*

* USER (USER\_ID, NAME, SURNAME, E\_MAIL, PHONE\_NUMBER, COMPANY\_ID)
* JOB (JOB\_ID, JOB\_NAME, SENIORITY\_LEVEL, INDUSTRY, JOB\_DESCRIPTION, COMPANY\_ID)
* INSTRUCTOR (INSTRUCTOR\_NUMBER, RANK, MAJOR, DEPT\_ID)
* DEPARTMENT (DEPT\_ID, DEPT\_NAME, DEPT\_OFFICE, DEPT\_PHONE, UNIVERSITY\_ID)
* COURSE (COURSE\_CODE, COURSE\_NAME, CREDIT, DEPT\_ID)
* STUDENT (STUDENT\_NUMBER, GPA, DEPT\_ID)

*STEP 4.2:*

* PROJECT (INSTRUCTOR\_NUMBER, COURSE\_CODE, SEC\_ID, PROJECT\_CODE, PROJECT\_DESCRIPTION, PROJECT\_NAME, START\_DATE, ESTIMATED\_END\_DATE, SUPERVISOR\_ID)

## *STEP 5: Mapping M:N Relations*

*STEP 5.1:*

* COMMENTS\_ON (USER\_ID, POST\_ID, COMMENT)
* SHARES\_POST (USER\_ID, POST\_ID)
* LIKES\_POST (USER\_ID, POST\_ID)
* USER\_CREATES\_POST (USER\_ID, POST\_ID)
* ORGANIZATION\_CREATES\_POST (ORGANIZATION\_ID, POST\_ID)
* FOLLOWS\_ORGANIZATION (USER\_ID, ORGANIZATION\_ID)
* KNOWS\_SKILLS (USER\_ID, SKILL\_ID)
* APPLIES\_JOB (USER\_ID, JOB\_ID)
* REQUIRES\_SKILLS (JOB\_ID, SKILL\_ID)
* TAKES\_SECTION (STUDENT\_ID, SEC\_ID, GRADE)
* MESSAGE (USER\_ID, REC\_USER\_ID, MESSAGE\_TEXT)
* REFERS\_TO (USER\_ID, REFERENCE\_ID)
* CONNECTS (USER\_ID, CONNECTED\_USER\_ID)

*STEP 5.2:*

* TAKES\_QUIZ (STUDENT\_NUMBER, INSTRUCTOR\_NUMBER, COURSE\_CODE, SEC\_ID, QUIZ\_ID, ANSWER\_PATH, QUIZ\_GRADE)
* WORKS\_ON\_PROJECT (STUDENT\_NUMBER, INSTRUCTOR\_NUMBER, COURSE\_CODE, SEC\_ID, PROJECT\_CODE)

## *STEP 6: Mapping Multivalued Attributes*

* PROJECTS\_CV (CV\_ID, PROJECT\_ID, NAME, DESCRIPTION)

## *STEP 8: Mapping Multiple Relations :*

## *Superclass – Subclass*

* USER (USER\_ID, NAME, SURNAME, E\_MAIL, PHONE\_NUMBER, USER\_TYPE)

USER\_TYPE: ‘Student’ OR ‘Instructor’ OR NULL

**USER\_TYPE == ‘Instructor’:**

INSTRUCTOR (INSTRUCTOR\_NUMBER, RANK, MAJOR)

INSTRUCTOR\_NUMBER == USER\_ID

**USER\_TYPE == ‘Student’:**

STUDENT (STUDENT\_NUMBER, GPA) STUDENT\_NUMBER == USER\_ID

* ORGANIZATION (ORGANIZATION\_ID, WEBSITE, ADDRESS, ORGANIZATION\_TYPE)

ORGANIZATION\_TYPE: ‘Company’ OR ‘Student’ OR NULL

**ORGANIZATION\_TYPE == ‘Company’:**

COMPANY (COMPANY\_ID, COMPANY\_NAME)

COMPANY\_ID == ORGANIZATION\_ID

**ORGANIZATION\_TYPE == ‘University’:**

UNIVERSITY (UNIVERSITY\_ID, UNIVERSITY\_NAME)

UNIVERSITY\_ID == ORGANIZATION\_ID

## IMPLEMENTATION – PHYSICAL MODEL

## *1) CREATING THE DATABASE AND ITS RELATIONAL MODEL USING SQL SERVER/T-SQL*

CREATE TABLE ORGANIZATION

(

organization\_id INT,

organization\_type VARCHAR(50),

website VARCHAR(50),

[address] VARCHAR(50) NOT NULL,

PRIMARY KEY(organization\_id)

);

CREATE TABLE [USER] (

user\_id INT,

name VARCHAR(50) NOT NULL,

surname VARCHAR(50) NOT NULL,

e\_mail VARCHAR(50) NOT NULL UNIQUE,

phone\_number VARCHAR(50),

user\_type VARCHAR(50),

company\_id INT,

PRIMARY KEY(user\_id)

);

CREATE TABLE UNIVERSITY

(

university\_id INT,

university\_name VARCHAR(50) NOT NULL,

rector\_id INT,

PRIMARY KEY(university\_id),

FOREIGN KEY(university\_id) REFERENCES ORGANIZATION(organization\_id),

FOREIGN KEY(rector\_id) REFERENCES [USER](user\_id)

);

CREATE TABLE DEPARTMENT

(

dept\_id INT,

dean\_id INT,

university\_id INT,

dept\_name VARCHAR(50) NOT NULL,

dept\_office VARCHAR(50) NOT NULL,

dept\_phone VARCHAR(50) NOT NULL,

PRIMARY KEY(dept\_id),

FOREIGN KEY(dean\_id) REFERENCES [USER](user\_id),

FOREIGN KEY(university\_id) REFERENCES UNIVERSITY(university\_id)

);

CREATE TABLE INSTRUCTOR

(

instructor\_number INT,

dept\_id INT,

rank VARCHAR(50) NOT NULL,

major VARCHAR(50) NOT NULL,

PRIMARY KEY(instructor\_number),

FOREIGN KEY (instructor\_number) REFERENCES [USER](user\_id),

FOREIGN KEY(dept\_id) REFERENCES DEPARTMENT(dept\_id)

);

CREATE TABLE STUDENT

(

student\_number INT,

dept\_id INT,

gpa FLOAT NOT NULL,

PRIMARY KEY(student\_number),

FOREIGN KEY (student\_number) REFERENCES [USER](user\_id),

FOREIGN KEY(dept\_id) REFERENCES DEPARTMENT(dept\_id)

);

CREATE TABLE CV

(

cv\_id INT,

user\_id INT,

position VARCHAR(50),

about VARCHAR(200),

education VARCHAR(200),

PRIMARY KEY(cv\_id),

FOREIGN KEY(user\_id) REFERENCES [USER](user\_id)

);

CREATE TABLE POST

(

post\_id INT,

post\_title VARCHAR(50) NOT NULL,

post\_content VARCHAR(200) NOT NULL,

PRIMARY KEY(post\_id)

);

CREATE TABLE SKILL

(

skill\_id INT,

skill\_name VARCHAR(50) NOT NULL,

category VARCHAR(50),

PRIMARY KEY(skill\_id)

);

CREATE TABLE COMPANY

(

company\_id INT,

company\_name VARCHAR(50) NOT NULL,

manager\_id INT,

PRIMARY KEY(company\_id),

FOREIGN KEY(company\_id) REFERENCES ORGANIZATION(organization\_id),

FOREIGN KEY(manager\_id) REFERENCES [USER](user\_id)

);

ALTER TABLE [USER]

ADD FOREIGN KEY(company\_id)

REFERENCES COMPANY(company\_id)

CREATE TABLE JOB

(

job\_id INT,

company\_id INT,

job\_name VARCHAR(50) NOT NULL,

seniority\_level VARCHAR(50),

industry VARCHAR(50),

job\_description VARCHAR(200) NOT NULL

PRIMARY KEY(job\_id)

FOREIGN KEY(company\_id) REFERENCES COMPANY(company\_id)

);

CREATE TABLE COURSE

(

course\_code INT,

course\_name VARCHAR(50) NOT NULL,

dept\_id INT,

credit INT NOT NULL,

PRIMARY KEY(course\_code),

FOREIGN KEY(dept\_id) REFERENCES DEPARTMENT(dept\_id)

);

CREATE TABLE SECTION

(

course\_code INT,

instructor\_number INT,

sec\_id INT,

year INT,

semester VARCHAR(50) NOT NULL,

sec\_days VARCHAR(50) NOT NULL,

classroom VARCHAR(50),

PRIMARY KEY(course\_code, instructor\_number, sec\_id),

FOREIGN KEY(instructor\_number) REFERENCES INSTRUCTOR(instructor\_number),

FOREIGN KEY(course\_code) REFERENCES COURSE(course\_code),

);

CREATE TABLE PROJECT

(

instructor\_number INT,

course\_code INT,

sec\_id INT,

project\_code INT,

project\_name VARCHAR(50),

start\_date DATE,

estimated\_end\_date DATE,

project\_description VARCHAR(200),

PRIMARY KEY(project\_code, course\_code, instructor\_number, sec\_id),

FOREIGN KEY(course\_code, instructor\_number, sec\_id) REFERENCES SECTION(course\_code, instructor\_number, sec\_id)

);

CREATE TABLE QUIZ

(

instructor\_number INT,

course\_code INT,

sec\_id INT,

quiz\_id INT,

question\_path VARCHAR(50),

PRIMARY KEY(quiz\_id, course\_code, instructor\_number, sec\_id),

FOREIGN KEY(course\_code, instructor\_number, sec\_id) REFERENCES SECTION(course\_code, instructor\_number, sec\_id)

);

CREATE TABLE COMMENTS\_ON\_POST

(

user\_id INT,

post\_id INT,

comment VARCHAR(200),

PRIMARY KEY(user\_id, post\_id),

FOREIGN KEY(user\_id) REFERENCES [USER](user\_id),

FOREIGN KEY(post\_id) REFERENCES POST(post\_id)

);

CREATE TABLE SHARES\_POST

(

user\_id INT,

post\_id INT,

PRIMARY KEY(user\_id, post\_id),

FOREIGN KEY(user\_id) REFERENCES [USER](user\_id),

FOREIGN KEY(post\_id) REFERENCES POST(post\_id)

);

CREATE TABLE LIKES\_POST

(

user\_id INT,

post\_id INT,

PRIMARY KEY(user\_id, post\_id),

FOREIGN KEY(user\_id) REFERENCES [USER](user\_id),

FOREIGN KEY(post\_id) REFERENCES POST(post\_id)

);

CREATE TABLE USER\_CREATES\_POST

(

user\_id INT,

post\_id INT,

PRIMARY KEY(user\_id, post\_id),

FOREIGN KEY(user\_id) REFERENCES [USER](user\_id),

FOREIGN KEY(post\_id) REFERENCES POST(post\_id)

);

CREATE TABLE ORGANIZATION\_CREATES\_POST

(

organization\_id INT,

post\_id INT,

PRIMARY KEY(organization\_id, post\_id),

FOREIGN KEY(organization\_id) REFERENCES ORGANIZATION(organization\_id),

FOREIGN KEY(post\_id) REFERENCES POST(post\_id)

);

CREATE TABLE FOLLOWS\_ORGANIZATION

(

user\_id INT,

organization\_id INT,

PRIMARY KEY(user\_id, organization\_id),

FOREIGN KEY(user\_id) REFERENCES [USER](user\_id),

FOREIGN KEY(organization\_id) REFERENCES ORGANIZATION(organization\_id)

);

CREATE TABLE KNOWS\_SKILLS

(

user\_id INT,

skill\_id INT,

PRIMARY KEY(user\_id, skill\_id),

FOREIGN KEY(user\_id) REFERENCES [USER](user\_id),

FOREIGN KEY(skill\_id) REFERENCES SKILL(skill\_id)

);

CREATE TABLE APPLIES\_JOB

(

user\_id INT,

job\_id INT,

PRIMARY KEY(user\_id, job\_id),

FOREIGN KEY(user\_id) REFERENCES [USER](user\_id),

FOREIGN KEY(job\_id) REFERENCES JOB(job\_id)

);

CREATE TABLE REQUIRES\_SKILLS

(

job\_id INT,

skill\_id INT,

PRIMARY KEY(job\_id, skill\_id),

FOREIGN KEY(job\_id) REFERENCES JOB(job\_id),

FOREIGN KEY(skill\_id) REFERENCES SKILL(skill\_id)

);

CREATE TABLE TAKES\_SECTION

(

student\_number INT,

course\_code INT,

instructor\_number INT,

sec\_id INT,

grade FLOAT,

PRIMARY KEY(student\_number, course\_code, instructor\_number, sec\_id),

FOREIGN KEY(student\_number) REFERENCES STUDENT(student\_number),

FOREIGN KEY(course\_code, instructor\_number, sec\_id) REFERENCES SECTION(course\_code, instructor\_number, sec\_id)

);

CREATE TABLE MESSAGE

(

user\_id INT,

receiver\_user\_id INT,

message\_text VARCHAR(200),

PRIMARY KEY(user\_id, receiver\_user\_id),

FOREIGN KEY(user\_id) REFERENCES [USER](user\_id),

FOREIGN KEY(receiver\_user\_id) REFERENCES [USER](user\_id)

);

CREATE TABLE REFERS\_TO

(

user\_id INT,

reference\_id INT,

PRIMARY KEY(user\_id, reference\_id),

FOREIGN KEY(user\_id) REFERENCES [USER](user\_id),

FOREIGN KEY(reference\_id) REFERENCES [USER](user\_id)

);

CREATE TABLE CONNECTS

(

user\_id INT,

connected\_user\_id INT,

PRIMARY KEY(user\_id, connected\_user\_id),

FOREIGN KEY(user\_id) REFERENCES [USER](user\_id),

FOREIGN KEY(connected\_user\_id) REFERENCES [USER](user\_id)

);

CREATE TABLE TAKES\_QUIZ

(

student\_number INT,

instructor\_number INT,

course\_code INT,

sec\_id INT,

quiz\_id INT,

quiz\_grade FLOAT,

answer\_path VARCHAR(50),

PRIMARY KEY(student\_number, instructor\_number, course\_code, sec\_id, quiz\_id),

FOREIGN KEY(student\_number) REFERENCES STUDENT(student\_number),

FOREIGN KEY(quiz\_id, course\_code, instructor\_number, sec\_id) REFERENCES

QUIZ(quiz\_id, course\_code, instructor\_number, sec\_id)

);

CREATE TABLE WORKS\_ON\_PROJECT

(

student\_number INT,

instructor\_number INT,

course\_code INT,

sec\_id INT,

project\_code INT,

PRIMARY KEY(student\_number, project\_code, instructor\_number, course\_code, sec\_id),

FOREIGN KEY(student\_number) REFERENCES STUDENT(student\_number),

FOREIGN KEY(project\_code, course\_code, instructor\_number, sec\_id) REFERENCES

PROJECT(project\_code, course\_code, instructor\_number, sec\_id)

);

CREATE TABLE PROJECTS\_CV

(

cv\_id INT,

project\_id INT,

project\_name VARCHAR(100),

project\_desc VARCHAR(200),

PRIMARY KEY(cv\_id, project\_id),

FOREIGN KEY(cv\_id) REFERENCES CV(cv\_id)

);

## *2) SAMPLE TUPLES*

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(2,'MERT','AKCAY','mert.akcay@example.com','525325262','Instructor',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(4,'BUSE','ARGUS','buse.argus@example.com','534523523','Instructor',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(6,'PELIN','DOYUK','pelin.doyuk@example.com','234243244','Instructor',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(1,'OZDAL','SAGLAM','ozdal.saglam@example.com','546524234','Instructor',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(3,'CEM','ULUS','cem.ulus@example.com','546546343','Instructor',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(5,'BEGUM','SAHIN','begum.sahin@example.com','546546456','Instructor',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(7,'BURAK','AKCAY','burak.akcay@example.com','546546456','Student',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(8,'MEHMET','YILMAZ','mehmet.yilmaz@example.com','546549456','Student',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(9,'MUSTAFA','KAYA','mustafa.kaya@example.com','546541556','Student',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(10,'HATİCE','DEMİR','hatice.demir@example.com','546541556','Student',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(11,'FATMA','YILMAZ','fatma.yilmaz@example.com','546541557',null,null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(12,'ALİ','DEMİR','ali.demir@example.com','546541553',null,null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(13,'CEM','AKCAY','cem.akcay@example.com','546541559',null,null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(14,'SİNEM','SAHİN','sinem.sahin@example.com','546541969',null,null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(15,'YUSUF','YILMAZ','yusuf.yilmaz@example.com','546541557',null,null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(16,'MEHMET','DEMİR','mehmet.demir@example.com','546541553',null,null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(17,'CEM','YILDIZ','cem.yildiz@example.com','546541559',null,null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(18,'SİNEM','ÇELİK','sinem.celik@example.com','546541969',null,null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(19,'SENA','SAGLAM','sena.saglam@example.com','546524234','Instructor',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(20,'DENİZ','ULUS','deniz.ulus@example.com','546546343','Instructor',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(21,'MERT','SAHIN','mert.sahin@example.com','546546456','Instructor',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(22,'YAGMUR','KAYA','yagmur.kaya@example.com','546541556','Student',null);

INSERT INTO [USER](user\_id,name,surname,e\_mail,phone\_number,user\_type,company\_id) VALUES(23,'ÖMER','DEMİR','omer.demir@example.com','546541556','Student',null);

INSERT INTO ORGANIZATION VALUES(1, 'Company', 'nokia.com', 'Espoo/Finland');

INSERT INTO ORGANIZATION VALUES(2, 'Company', 'samsung.com', 'Suwon/Korea');

INSERT INTO ORGANIZATION VALUES(3, 'Company', 'apple.com', 'USA');

INSERT INTO ORGANIZATION VALUES(4, 'University', 'egeuniversity.com', 'Bornova/İzmir');

INSERT INTO ORGANIZATION VALUES(5, 'University', 'ankarauniversity.com', 'Ankara');

INSERT INTO ORGANIZATION VALUES(6, 'University', 'gaziuniversity.com', 'Yenimahalle/Ankara');

INSERT INTO COMPANY(company\_id,company\_name,manager\_id) VALUES(1,'NOKIA',11);

UPDATE [USER] SET company\_id = 1 where user\_id = 11

UPDATE [USER] SET company\_id = 1 where user\_id = 16

UPDATE [USER] SET company\_id = 1 where user\_id = 7

INSERT INTO COMPANY(company\_id,company\_name,manager\_id) VALUES(2,'SAMSUNG',12);

UPDATE [USER] SET company\_id = 2 where user\_id = 12

UPDATE [USER] SET company\_id = 2 where user\_id = 15

UPDATE [USER] SET company\_id = 2 where user\_id = 9

INSERT INTO COMPANY(company\_id,company\_name,manager\_id) VALUES(3,'APPLE',13);

UPDATE [USER] SET company\_id = 3 where user\_id = 13

UPDATE [USER] SET company\_id = 3 where user\_id = 14

UPDATE [USER] SET company\_id = 3 where user\_id = 19

INSERT INTO UNIVERSITY(university\_id,university\_name,rector\_id) VALUES(4,'Ege Üniversitesi', 2);

INSERT INTO UNIVERSITY(university\_id,university\_name,rector\_id) VALUES(5,'Ankara Üniversitesi',4);

INSERT INTO UNIVERSITY(university\_id,university\_name,rector\_id) VALUES(6,'Gazi Üniversitesi',6);

INSERT INTO DEPARTMENT(dept\_id,dept\_name,dean\_id,dept\_office,dept\_phone,university\_id) VALUES(1,'Computer Engineering',1,'CS','546456645',4);

INSERT INTO DEPARTMENT(dept\_id,dept\_name,dean\_id,dept\_office,dept\_phone,university\_id) VALUES(2,'Physics',3,'PHYS','546645645',5);

INSERT INTO DEPARTMENT(dept\_id,dept\_name,dean\_id,dept\_office,dept\_phone,university\_id) VALUES(3,'Math',5,'MATH','546123645',6);

INSERT INTO INSTRUCTOR(instructor\_number,dept\_id,major,rank) VALUES(1,1,'AI','Asst. Prof');

INSERT INTO INSTRUCTOR(instructor\_number,dept\_id,major,rank) VALUES(3,2,'Astrophysics','Assc. Prof');

INSERT INTO INSTRUCTOR(instructor\_number,dept\_id,major,rank) VALUES(5,3,'Numerical analysis','Prof');

INSERT INTO INSTRUCTOR(instructor\_number,dept\_id,major,rank) VALUES(2,null,'Economics','Prof');

INSERT INTO INSTRUCTOR(instructor\_number,dept\_id,major,rank) VALUES(4,null,'Software Engineering','Prof');

INSERT INTO INSTRUCTOR(instructor\_number,dept\_id,major,rank) VALUES(6,null,'Industry Engineering','Prof');

INSERT INTO INSTRUCTOR(instructor\_number,dept\_id,major,rank) VALUES(19,1,'Data Structures','Asst. Prof');

INSERT INTO INSTRUCTOR(instructor\_number,dept\_id,major,rank) VALUES(20,2,'Astronomy','Assc. Prof');

INSERT INTO INSTRUCTOR(instructor\_number,dept\_id,major,rank) VALUES(21,3,'Linear Algebra','Prof');

INSERT INTO STUDENT(student\_number,dept\_id,gpa) VALUES(7,1,3.4);

INSERT INTO STUDENT(student\_number,dept\_id,gpa) VALUES(8,2,2.4);

INSERT INTO STUDENT(student\_number,dept\_id,gpa) VALUES(9,3,2.9);

INSERT INTO STUDENT(student\_number,dept\_id,gpa) VALUES(10,3,2.6);

INSERT INTO STUDENT(student\_number,dept\_id,gpa) VALUES(22,2,3.1);

INSERT INTO STUDENT(student\_number,dept\_id,gpa) VALUES(23,1,3.3);

INSERT INTO COURSE(course\_code,course\_name,credit,dept\_id) VALUES(1,'Data Structures',3,1);

INSERT INTO COURSE(course\_code,course\_name,credit,dept\_id) VALUES(3,'Artificial Intelligence',5,1);

INSERT INTO COURSE(course\_code,course\_name,credit,dept\_id) VALUES(5,'Molecular physics',4,2);

INSERT INTO COURSE(course\_code,course\_name,credit,dept\_id) VALUES(6,'Nuclear physics',5,2);

INSERT INTO COURSE(course\_code,course\_name,credit,dept\_id) VALUES(7,'Probability and Statistics',3,3);

INSERT INTO COURSE(course\_code,course\_name,credit,dept\_id) VALUES(9,'Differential equations',5,3);

INSERT INTO SECTION(sec\_id,course\_code,instructor\_number,classroom,sec\_days,semester,year) VALUES (1,1,19,'A101','Monday','Spring',2022);

INSERT INTO SECTION(sec\_id,course\_code,instructor\_number,classroom,sec\_days,semester,year) VALUES (2,1,19,'A101','Monday','Spring',2021);

INSERT INTO SECTION(sec\_id,course\_code,instructor\_number,classroom,sec\_days,semester,year) VALUES (3,3,1,'A105','Wednesday','Fall',2021);

INSERT INTO SECTION(sec\_id,course\_code,instructor\_number,classroom,sec\_days,semester,year) VALUES (4,5,3,'A109','Tuesday','Fall',2021);

INSERT INTO SECTION(sec\_id,course\_code,instructor\_number,classroom,sec\_days,semester,year) VALUES (5,6,20,'A111','Friday','Spring',2022);

INSERT INTO SECTION(sec\_id,course\_code,instructor\_number,classroom,sec\_days,semester,year) VALUES (6,6,20,'A111','Friday','Spring',2021);

INSERT INTO SECTION(sec\_id,course\_code,instructor\_number,classroom,sec\_days,semester,year) VALUES (7,7,21,'A107','Thursday','Spring',2022);

INSERT INTO SECTION(sec\_id,course\_code,instructor\_number,classroom,sec\_days,semester,year) VALUES (8,9,5,'A119','Tuesday','Fall',2021);

INSERT INTO SECTION(sec\_id,course\_code,instructor\_number,classroom,sec\_days,semester,year) VALUES (9,9,5,'A119','Tuesday','Fall',2022);

INSERT INTO PROJECT(course\_code,project\_code,sec\_id,project\_name,project\_description,instructor\_number,estimated\_end\_date,start\_date) VALUES(1,1,1,'Data Structures Final Project','Project for data structures course.',19,'01-03-2022','10-04-2022');

INSERT INTO PROJECT(course\_code,project\_code,sec\_id,project\_name,project\_description,instructor\_number,estimated\_end\_date,start\_date) VALUES(3,2,3,'Chess AI','AI project.',1,'01-01-2022','01-29-2022');

INSERT INTO PROJECT(course\_code,project\_code,sec\_id,project\_name,project\_description,instructor\_number,estimated\_end\_date,start\_date) VALUES(5,3,4,'Molecular Physics','Molecular physics research project',3,'02-02-2022','02-17-2022');

INSERT INTO QUIZ(instructor\_number,course\_code,sec\_id,quiz\_id,question\_path) VALUES (19,1,1,1,'questions/example');

INSERT INTO JOB(job\_id,company\_id,job\_name,seniority\_level,job\_description,industry) VALUES (1,1,'SOFTWARE DEVELOPER','JR','Software development job for jr engineers.','Telecommunications');

INSERT INTO JOB(job\_id,company\_id,job\_name,seniority\_level,job\_description,industry) VALUES (2,2,'MACHINE LEARNING ENGINEER','MID','Engineering job for our ML systems.','Consumer Electronics');

INSERT INTO JOB(job\_id,company\_id,job\_name,seniority\_level,job\_description,industry) VALUES (3,3,'BUSINESS ANALYST','SENIOR','Senior analyst.','Consumer Electronics');

INSERT INTO SKILL(skill\_id,category,skill\_name) VALUES(1,'Industry Knowledge','Deep Learning');

INSERT INTO SKILL(skill\_id,category,skill\_name) VALUES(2,'Industry Knowledge','AI');

INSERT INTO SKILL(skill\_id,category,skill\_name) VALUES(3,'Tools','.NET');

INSERT INTO SKILL(skill\_id,category,skill\_name) VALUES(4,'Tools','Java');

INSERT INTO SKILL(skill\_id,category,skill\_name) VALUES(5,'Interpersonal Skills','Teamwork');

INSERT INTO SKILL(skill\_id,category,skill\_name) VALUES(6,'Language','English');

INSERT INTO SKILL(skill\_id,category,skill\_name) VALUES(7,'Tools','SQL');

INSERT INTO SKILL(skill\_id,category,skill\_name) VALUES(8,'Industry Knowledge','Agile Methodologies');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(1,1,'Dean','Dean at EGE Uni. Computer Engineering Dept', 'Doctorate');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(2,2,'Rector','Rector at EGE Uni.', 'Doctorate');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(3,3,'Dean','Dean at ANKARA Uni. Physics Dept', 'Doctorate');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(4,4,'Rector','Rector at ANKARA Uni', 'Doctorate');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(5,5,'Dean','Dean at GAZİ Uni. Math Dept.', 'Doctorate');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(6,6,'Rector','Rector at GAZİ Uni', 'Doctorate');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(7,7,'Developer','Studying CS, developer at NOKIA','Bachelors degree');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(8,8,'Student','Studying Physics','Bachelors degree');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(9,9,'Student','Studying Math','Bachelors degree');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(10,10,'Student','Studying Maths','Masters degree');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(11,11,'Manager','Manager at NOKIA','Masters degree');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(12,12,'Manager','Manager at SAMSUNG','Masters degree');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(13,13,'Manager','Manager at APPLE','Masters degree');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(14,14,'Analyst','Analyst at APPLE','Masters degree');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(15,15,'Engineer','Engineer at SAMSUNG','Bachelors degree');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(16,16,'Developer','Developer at NOKIA','Bachelors degree');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(17,17,'Unemployed','Looking for a job...','Bachelors degree');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(18,18,'Unemployed','Looking for a job','Bachelors degree');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(19,19,'Asst. Prof','Asst. Prof at Ege Uni.','Doctorate');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(20,20,'Assc. Prof','Assc. Prof at Ankara Uni.','Doctorate');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(21,21,'Prof','Prof at Gazi Uni.','Doctorate');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(22,22,'Student','Studying Physics','Bachelors Degree');

INSERT INTO CV(cv\_id,user\_id,position,about,education) VALUES(23,23,'Student','Studing CENG','Bachelors Degree');

INSERT INTO POST(post\_id,post\_title,post\_content) VALUES(1,'POST1','POST1');

INSERT INTO POST(post\_id,post\_title,post\_content) VALUES(2,'POST2','POST2');

INSERT INTO POST(post\_id,post\_title,post\_content) VALUES(3,'POST3','POST3');

INSERT INTO POST(post\_id,post\_title,post\_content) VALUES(4,'POST4','POST4');

INSERT INTO POST(post\_id,post\_title,post\_content) VALUES(5,'POST5','POST5');

INSERT INTO USER\_CREATES\_POST(user\_id, post\_id) VALUES(10,1);

INSERT INTO USER\_CREATES\_POST(user\_id, post\_id) VALUES(17,2);

INSERT INTO USER\_CREATES\_POST(user\_id, post\_id) VALUES(22,3);

INSERT INTO ORGANIZATION\_CREATES\_POST(organization\_id, post\_id) VALUES(1,4);

INSERT INTO ORGANIZATION\_CREATES\_POST(organization\_id, post\_id) VALUES(6,5);

INSERT INTO COMMENTS\_ON\_POST(user\_id, post\_id) VALUES(5,1);

INSERT INTO SHARES\_POST(user\_id, post\_id) VALUES(3,2);

INSERT INTO LIKES\_POST(user\_id, post\_id) VALUES(11,5);

INSERT INTO FOLLOWS\_ORGANIZATION(user\_id, organization\_id) VALUES (18,3)

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(17, 7);

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(17, 8);

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(18, 1);

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(18, 5);

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(8, 5);

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(8, 4);

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(8, 1);

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(19, 8);

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(23, 8);

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(23, 7);

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(19, 1);

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(19, 7);

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(22, 2);

INSERT INTO KNOWS\_SKILLS(user\_id, skill\_id) VALUES(22, 1);

INSERT INTO APPLIES\_JOB(user\_id, job\_id) VALUES(17, 3);

INSERT INTO APPLIES\_JOB(user\_id, job\_id) VALUES(18, 2);

INSERT INTO APPLIES\_JOB(user\_id, job\_id) VALUES(8, 1);

INSERT INTO APPLIES\_JOB(user\_id, job\_id) VALUES(22, 2);

INSERT INTO APPLIES\_JOB(user\_id, job\_id) VALUES(23, 3);

INSERT INTO APPLIES\_JOB(user\_id, job\_id) VALUES(19, 1);

INSERT INTO REQUIRES\_SKILLS(job\_id, skill\_id) VALUES(3, 7);

INSERT INTO REQUIRES\_SKILLS(job\_id, skill\_id) VALUES(3, 8);

INSERT INTO REQUIRES\_SKILLS(job\_id, skill\_id) VALUES(2, 1);

INSERT INTO REQUIRES\_SKILLS(job\_id, skill\_id) VALUES(2, 2);

INSERT INTO REQUIRES\_SKILLS(job\_id, skill\_id) VALUES(1, 3);

INSERT INTO REQUIRES\_SKILLS(job\_id, skill\_id) VALUES(1, 4);

INSERT INTO REQUIRES\_SKILLS(job\_id, skill\_id) VALUES(1, 5);

INSERT INTO TAKES\_SECTION(student\_number, course\_code, instructor\_number, sec\_id, grade) VALUES(7, 1, 19, 1, 100);

INSERT INTO TAKES\_SECTION(student\_number, course\_code, instructor\_number, sec\_id, grade) VALUES(8, 5, 3, 4, 77);

INSERT INTO TAKES\_SECTION(student\_number, course\_code, instructor\_number, sec\_id, grade) VALUES(9, 7, 21, 7, 66);

INSERT INTO TAKES\_SECTION(student\_number, course\_code, instructor\_number, sec\_id, grade) VALUES(10, 9, 5, 9, 46);

INSERT INTO TAKES\_SECTION(student\_number, course\_code, instructor\_number, sec\_id, grade) VALUES(22, 5, 3, 4, 59);

INSERT INTO TAKES\_SECTION(student\_number, course\_code, instructor\_number, sec\_id, grade) VALUES(23, 1, 19, 1, 85);

INSERT INTO TAKES\_SECTION(student\_number, course\_code, instructor\_number, sec\_id, grade) VALUES(7, 3, 1, 3, 93);

INSERT INTO TAKES\_SECTION(student\_number, course\_code, instructor\_number, sec\_id, grade) VALUES(8, 6 ,20 ,5, 56);

INSERT INTO TAKES\_SECTION(student\_number, course\_code, instructor\_number, sec\_id, grade) VALUES(9, 9, 5, 8, 83);

INSERT INTO TAKES\_SECTION(student\_number, course\_code, instructor\_number, sec\_id, grade) VALUES(22, 6, 20, 6, 67);

INSERT INTO MESSAGE(user\_id, receiver\_user\_id, message\_text) VALUES(5, 1, 'Hello!');

INSERT INTO MESSAGE(user\_id, receiver\_user\_id, message\_text) VALUES(4, 9, 'HI');

INSERT INTO MESSAGE(user\_id, receiver\_user\_id, message\_text) VALUES(13, 21, 'Thanks!');

INSERT INTO REFERS\_TO(user\_id, reference\_id) VALUES(22, 5);

INSERT INTO REFERS\_TO(user\_id, reference\_id) VALUES(12, 13);

INSERT INTO REFERS\_TO(user\_id, reference\_id) VALUES(6, 8);

INSERT INTO REFERS\_TO(user\_id, reference\_id) VALUES(5, 22);

INSERT INTO REFERS\_TO(user\_id, reference\_id) VALUES(13, 12);

INSERT INTO REFERS\_TO(user\_id, reference\_id) VALUES(9, 20);

INSERT INTO REFERS\_TO(user\_id, reference\_id) VALUES(1, 4);

INSERT INTO REFERS\_TO(user\_id, reference\_id) VALUES(2, 18);

INSERT INTO CONNECTS(user\_id, connected\_user\_id) VALUES(1, 5);

INSERT INTO CONNECTS(user\_id, connected\_user\_id) VALUES(4, 9);

INSERT INTO CONNECTS(user\_id, connected\_user\_id) VALUES(21, 6);

INSERT INTO CONNECTS(user\_id, connected\_user\_id) VALUES(20, 6);

INSERT INTO CONNECTS(user\_id, connected\_user\_id) VALUES(9, 8);

INSERT INTO CONNECTS(user\_id, connected\_user\_id) VALUES(1, 2);

INSERT INTO CONNECTS(user\_id, connected\_user\_id) VALUES(3, 4);

INSERT INTO CONNECTS(user\_id, connected\_user\_id) VALUES(5, 6);

INSERT INTO CONNECTS(user\_id, connected\_user\_id) VALUES(19, 16);

INSERT INTO CONNECTS(user\_id, connected\_user\_id) VALUES(17, 20);

INSERT INTO CONNECTS(user\_id, connected\_user\_id) VALUES(22, 23);

INSERT INTO TAKES\_QUIZ(student\_number, instructor\_number, course\_code, sec\_id, quiz\_id, answer\_path, quiz\_grade)

VALUES(23, 19, 1, 1, 1, 'answers23/example', 80);

INSERT INTO TAKES\_QUIZ(student\_number, instructor\_number, course\_code, sec\_id, quiz\_id, answer\_path, quiz\_grade)

VALUES(7, 19, 1, 1, 1, 'answers7/example', 85);

INSERT INTO WORKS\_ON\_PROJECT(student\_number, project\_code, instructor\_number, course\_code, sec\_id) VALUES(23,1,19,1,1);

INSERT INTO WORKS\_ON\_PROJECT(student\_number, project\_code, instructor\_number, course\_code, sec\_id) VALUES(23,2,1,3,3);

INSERT INTO WORKS\_ON\_PROJECT(student\_number, project\_code, instructor\_number, course\_code, sec\_id) VALUES(7,2,1,3,3);

INSERT INTO WORKS\_ON\_PROJECT(student\_number, project\_code, instructor\_number, course\_code, sec\_id) VALUES(22,3,3,5,4);

INSERT INTO PROJECTS\_CV(cv\_id, project\_id, project\_name, project\_desc) VALUES(7, 1, 'Web Development', 'Web project with .NET');

INSERT INTO PROJECTS\_CV(cv\_id, project\_id, project\_name, project\_desc) VALUES(15, 2, 'Mobile App.', 'Mobile application development');

INSERT INTO PROJECTS\_CV(cv\_id, project\_id, project\_name, project\_desc) VALUES(12, 3, 'Research Project', 'Research project for thesis');

INSERT INTO PROJECTS\_CV(cv\_id, project\_id, project\_name, project\_desc) VALUES(23, 4, 'Library Applicaton', 'Library app. for school homework');

INSERT INTO PROJECTS\_CV(cv\_id, project\_id, project\_name, project\_desc) VALUES(17, 5, 'DL Classification', 'Classification project using deep learning.');

## *3) TRIGGERS*

## *3.1) Trigger that prevents the user from taking sections from departments other than her/his own*

CREATE TRIGGER SECTION\_CHECK ON TAKES\_SECTION

FOR INSERT

AS

IF(EXISTS((SELECT dept\_id FROM COURSE c , inserted i WHERE i.course\_code = c.course\_code) EXCEPT

(SELECT dept\_id FROM STUDENT s , inserted i WHERE i.student\_number = s.student\_number)))

BEGIN

RAISERROR ('The student does not study in the department where this section is given.', 16, 1);

ROLLBACK TRANSACTION;

END

*3.2) Trigger that prevents connection with previously connected person*

CREATE TRIGGER CONNECTS\_CHECK ON CONNECTS

FOR INSERT

AS

IF(EXISTS(SELECT \* FROM CONNECTS c, inserted i WHERE i.user\_id = c.user\_id AND i.connected\_user\_id = c.connected\_user\_id) OR

EXISTS(SELECT \* FROM CONNECTS c, inserted i WHERE i.user\_id = c.connected\_user\_id AND i.connected\_user\_id = c.user\_id ))

BEGIN

RAISERROR ('These users have been connected before.', 16, 1);

ROLLBACK TRANSACTION;

END

## *3.3) Trigger that prevents the project from being taken after the deadline*

CREATE TRIGGER PROJECTDATE\_CHECK ON WORKS\_ON\_PROJECT

FOR INSERT

AS

IF(EXISTS(SELECT \* FROM PROJECT p, inserted i WHERE i.project\_code = p.project\_code

AND p.estimated\_end\_date < cast(GETDATE() as date)))

BEGIN

RAISERROR ('The project cannot be taken after the estimated end date passed.', 16, 1);

ROLLBACK TRANSACTION;

END

## *4) CHECK CONSTRAINTS & ASSERTIONS*

ALTER TABLE POST

ADD CONSTRAINT POST\_TITLE\_LENGHT CHECK (LENGHT(POST\_CONTENT) < 240)

ALTER TABLE MESSAGE

ADD CONSTRAINT MESSAGE\_LENGHT CHECK (LENGHT( MESSAGE\_TEXT ) < 240)

ALTER TABLE COMMENTS\_ON\_POST

ADD CONSTRAINT COMMENT\_LENG CHECK (LEN( MESSAGE\_TEXT ) < 240)

ALTER TABLE PROJECT

ADD CONSTRAINT DATE\_CHECK CHECK (START\_DATE < ESTIMATED\_END\_DATE)

ALTER TABLE SECTION

ADD CONSTRAINT SEMESTER\_CHECK (SEMESTER = 'FALL' | SEMESTER = 'SPRING')

ALTER TABLE JOB

ADD CONSTRAINT SENIOR\_CHECK (SENIORITY\_LEVEL = 'JR' | SENIORITY\_LEVEL = 'MID' | SENIORITY\_LEVEL = 'SENIOR')

## *5) INSERT/UPDATE/DELETE STATEMENTS*

**INSERT STATEMENTS**

INSERT INTO REFERS\_TO(user\_id, reference\_id) VALUES(2, 5);

INSERT INTO MESSAGE(user\_id, receiver\_user\_id, message\_text) VALUES(2, 5, 'Hello!');

INSERT INTO SKILL(skill\_id,category,skill\_name) VALUES(3,'Tools','SQL Server’);

**UPDATE STATEMENTS**

UPDATE [MESSAGE]

SET message\_text = 'New Message For Update'

WHERE user\_id = 13 AND receiver\_user\_id = 21

UPDATE ORGANIZATION

SET address = 'USA/Seattle'

WHERE website LIKE 'apple%'

UPDATE SECTION

SET classroom = 'A111'

WHERE sec\_id = 2 AND instructor\_number = 19

**DELETE STATEMENTS**

DELETE FROM [CV] WHERE [CV].user\_id = 18;

DELETE FROM [SKILL] WHERE [SKILL].skill\_name = 'English'

DELETE FROM [MESSAGE] WHERE user\_id = 4 AND receiver\_user\_id = 9

## *6) SELECT STATEMENTS*

## *6.1) SELECT STATEMENTS WITH ONE TABLE*

**To list the names, surnames and phone numbers of instructors:**

SELECT name,surname,e\_mail,phone\_number

FROM [USER]

WHERE [USER].user\_type LIKE 'Instructor'

**To list courses with more than 3 credits:**

SELECT course\_name

FROM [COURSE]

WHERE [COURSE].credit > 3

**To list all skills in “Tools” category:**

SELECT skill\_name

FROM SKILL

WHERE category = 'Tools '

## *6.2) SELECT STATEMENTS WITH MINIMUM TWO TABLES*

**To list name, surname and phone number of the rector of Ege:**

SELECT UNIVERSITY\_NAME,NAME,SURNAME,PHONE\_NUMBER

FROM UNIVERSITY

JOIN [INSTRUCTOR] ON UNIVERSITY.RECTOR\_ID = INSTRUCTOR\_NUMBER

JOIN [USER] ON [USER].USER\_ID = INSTRUCTOR.INSTRUCTOR\_NUMBER

WHERE UNIVERSITY\_NAME LIKE 'EGE%'

**To list users’ and their reference’s name and surname:**

SELECT refere.name,refere.surname,[USER].name,[USER].surname

FROM [USER]

JOIN (SELECT name,surname,reference\_id

FROM [USER]

JOIN REFERS\_TO AS refer ON [USER].user\_id = refer.user\_id) AS refere

ON refere.reference\_id = [USER].user\_id

**To list connections’ senders and receivers’ name and surname**

SELECT conect.name AS IstekAtanIsım,conect.surname AS IstekAtanSoyısım,[USER].name AS IstekAlanIsım,[USER].surname AS IstekAlanSoyIsım

FROM [USER]

JOIN (SELECT name,surname,connected\_user\_id

FROM [USER]

JOIN CONNECTS AS C ON [USER].user\_id = C.user\_id) AS conect

ON conect.connected\_user\_id = [USER].user\_id

**To list post creators’ name, surname ,and title and content of the posts**

SELECT U.name,U.surname,post\_title,post\_content

FROM [USER] AS U

JOIN SHARES\_POST ON U.user\_id = SHARES\_POST.user\_id

JOIN POST ON POST.post\_id = SHARES\_POST.post\_id

## *6.3) SELECT STATEMENTS WITH MINIMUM THREE TABLES*

**To list users’ name, surname and companies’ names that they follow:**

SELECT name,surname,company\_name

FROM [USER] AS A

JOIN FOLLOWS\_ORGANIZATION ON A.user\_id = FOLLOWS\_ORGANIZATION.user\_id

JOIN ORGANIZATION ON ORGANIZATION.organization\_id = FOLLOWS\_ORGANIZATION.organization\_id

JOIN COMPANY ON COMPANY.company\_id = ORGANIZATION.organization\_id

**To list name, surname, gpa, id and grade of the users whose course code is 3, gpa is more than 3 and grade is more than 85**

SELECT NAME,SURNAME,GPA,[USER].user\_id,grade

FROM [STUDENT]

JOIN [USER] on [USER].user\_id = [STUDENT].student\_number

JOIN [TAKES\_SECTION] ON [STUDENT].student\_number = [TAKES\_SECTION].student\_number

JOIN [SECTION] ON [SECTION].course\_code = [TAKES\_SECTION].course\_code

WHERE gpa > 3 AND SECTION.course\_code = 3 AND grade > 85

**To list university names having the computer department**

select UNIVERSITY.university\_name

from UNIVERSITY

join [INSTRUCTOR] on UNIVERSITY.rector\_id = [INSTRUCTOR].instructor\_number

join [USER] on [USER].user\_id = [INSTRUCTOR].instructor\_number

join DEPARTMENT on UNIVERSITY.university\_id = DEPARTMENT.university\_id

WHERE dept\_name LIKE 'Computer %'

## *6.4) OTHER CRITICAL SELECT STATEMENTS*

**To list all message contents, names of senders and receivers:**

SELECT Sender\_Name AS SenderName,message\_text AS Message,name AS ReceiverName

FROM [USER]

JOIN (SELECT name AS Sender\_Name,message\_text ,receiver\_user\_id AS Receiver\_name

FROM [USER]

JOIN [dbo].[MESSAGE] AS sender ON [USER].user\_id = sender.user\_id) AS Sender

ON Sender.Receiver\_name = [USER].user\_id

**To list all students’ name with a gpa greater than 3 and not working in a company:**

SELECT DISTINCT(NAME)

FROM [STUDENT]

JOIN [USER] on [USER].user\_id = [STUDENT].student\_number

JOIN [TAKES\_SECTION] ON [STUDENT].student\_number = [TAKES\_SECTION].student\_number

JOIN [SECTION] ON [SECTION].course\_code = [TAKES\_SECTION].course\_code

WHERE company\_id IS NULL AND gpa > 3

**To list users who have the skills required for a job:**

SELECT DISTINCT([USER].NAME),[USER].SURNAME,[USER].USER\_ID

FROM [USER],COMPANY

JOIN JOB ON JOB.COMPANY\_ID = COMPANY.COMPANY\_ID

JOIN REQUIRES\_SKILLS ON REQUIRES\_SKILLS.JOB\_ID = JOB.JOB\_ID

JOIN SKILL ON SKILL.SKILL\_ID = REQUIRES\_SKILLS.SKILL\_ID

JOIN KNOWS\_SKILLS ON KNOWS\_SKILLS.SKILL\_ID = SKILL.SKILL\_ID

WHERE SKILL.SKILL\_NAME IN (

SELECT SKILL.SKILL\_NAME

FROM [USER]

JOIN KNOWS\_SKILLS ON [USER].USER\_ID = KNOWS\_SKILLS.USER\_ID

JOIN SKILL ON KNOWS\_SKILLS.SKILL\_ID = SKILL.SKILL\_ID)

**To list university adress for cargo:**

SELECT university\_name,address

FROM ORGANIZATION

JOIN UNIVERSITY ON university\_id = ORGANIZATION.organization\_id

WHERE [ORGANIZATION].organization\_type LIKE 'University'

**To list student information who takes more than grade 80 on quizzes:**

SELECT name,surname,[USER].user\_id, QUIZ.quiz\_id, quiz\_grade

FROM STUDENT

JOIN [USER] ON [USER].user\_id = STUDENT.student\_number

JOIN TAKES\_QUIZ ON STUDENT.student\_number = TAKES\_QUIZ.student\_number

JOIN QUIZ ON QUIZ.quiz\_id = TAKES\_QUIZ.quiz\_id

WHERE quiz\_grade > 80