



**EGE UNIVERSITY**  
**FACULTY OF ENGINEERING**  
**COMPUTER ENGINEERING DEPARTMENT**

**DATABASE MANAGEMENT**  
**2021–2022 FALL SEMESTER**

**PROJECT REPORT**  
**(LinkedInMoodle Database)**

**PREPARED BY**

Onur AYGÜN

Yağmur Durukan

Burak Taha Pekşen

Ozan Coşkun

## İçindekiler Tablosu

Analysis .....	3-6
EER .....	7
Design-Logical Model .....	8-10
Implementation-Physical Model .....	10-43

# ANALYSIS

## LINKEDIN

1. LINKEDIN is a social media application that people can create network by contacting other people or companies generally in professional purposes. By forming a profile with their interests, abilities, experiences, hobbies, they introduce themselves to the companies or people that are wanted to be connected. The application provides people to discover job opportunities, and easy apply method their profiles. Also, posts can be shared in any topic that users can react them, and users are able to send messages each other on LinkedIn. In addition, events, courses and groups for several purposes may be organised.
2.
  - a. The aim of LinkedIn is providing users a platform that they can stay connected with people and companies, be aware of trends and innovations, personal improvement. Also, the purpose might be finding a job for people, hiring for companies.
  - b. Main entities are member, user, organisation, university, company, organisation\_profile, profile, job\_advertisement, post , group.
  - c. Member → member\_id, login (member\_mail, password), member\_type  
Person → member\_id  
Organisation → member\_id  
Company → member\_id  
University → uni\_name  
Organisation\_profile → org\_id, org\_name, founding\_date, no\_of\_employee, {Location}, web\_site, sector, founder, org\_phone  
Profile → profile\_id, name (FName,LName), address, sex, birthday, phone  
Post → post\_id, member\_id  
Text → post\_id, text  
Event → post\_id, event\_type, event\_date, event\_name  
Photo → post\_id, tag  
Video → post\_id, thumbnail  
Group → group\_id, GName, GMember\_no  
Grade → student\_no, Student\_grade (grade\_point, letter\_grade)

- d. VIEW\_PROFILE (between MEMBER, MEMBER)
  - Each person can view all member's profile.
- VIEW (between MEMBER, JOB\_ADV)
  - Each member can view all of job advertisements.
- APPLY (between MEMBER, JOB\_ADV)
  - Each member can apply more than one job advertisements.
- LIKE (between MEMBER, POST)
  - Members may like posts.
- SHARE (between MEMBER, POST)
  - Members may share posts.
- COMMENT (between MEMBER, POST)
  - Members may comment posts.
- CONNECT (between MEMBER, MEMBER)
  - Each member can connect more than one member.
- CHAT (between MEMBER, MEMBER)
  - Each member can chat more than one member.
- PUBLISH (between ORGANISATION, JOB\_ADV)
  - An organisation can publish more than one job advertisements.
- ORG\_HAS\_PROFILE (between ORGANISATION, ORGANISATION\_PROFILE)
  - An organisation must have only one organisation profile.
- VIEW\_GROUP (between PERSON, GROUP)
  - Persons can view groups.
- JOIN (between PERSON, GROUP)
  - Persons can join more than one group.
- MANAGE\_GROUP (between PERSON, GROUP)
  - Persons can manage several groups.
- CREATE (between PERSON, GROUP)
  - A person can create more than one group.
- HAS\_PROFILE (between PERSON, PROFILE)
  - A person must have only one profile.

WORKS\_IN (between PERSON, JOB)

-Each person may work in a job.

BELONGS (between JOB, ORGANISATION)

-Jobs have to belong one organisation.

MANAGE (between PERSON, ORGANISATION)

-A person can manage more than one organisation.

e.

-Members must have unique and valid mail addresses.

-Member type can be only organisation or person.

-Each member must have a profile.

-University name must be unique.

-Person must join a group in order to manage a group.

-If a person working in a job, start date must be earlier than today's time.

-The number of members that apply the job advertisements must be greater than or equal to zero.

-The number of members in a group must be greater than zero.

-The person who wants to be a LinkedInMoodle member must born later than 2008-01-01.

## MOODLE

1. MOODLE is a platform that is used for providing online courses and learning management system. There are several courses at university departments, and students can enrol the courses that are given for their departments. Instructors upload the materials related to the course into the system, and students access them. These materials might be projects, quizzes, exams, videos or lesson notes. Students can follow the exam results and the announcements that are shared by instructor about the course in the system.

2.

a. The aim of Moodle is providing a platform that students can take courses and improve themselves, instructors give courses, manage the lessons and follow the activities of students via the application.

b. Main entities are person (student, instructor), profile, course, project, department, university.

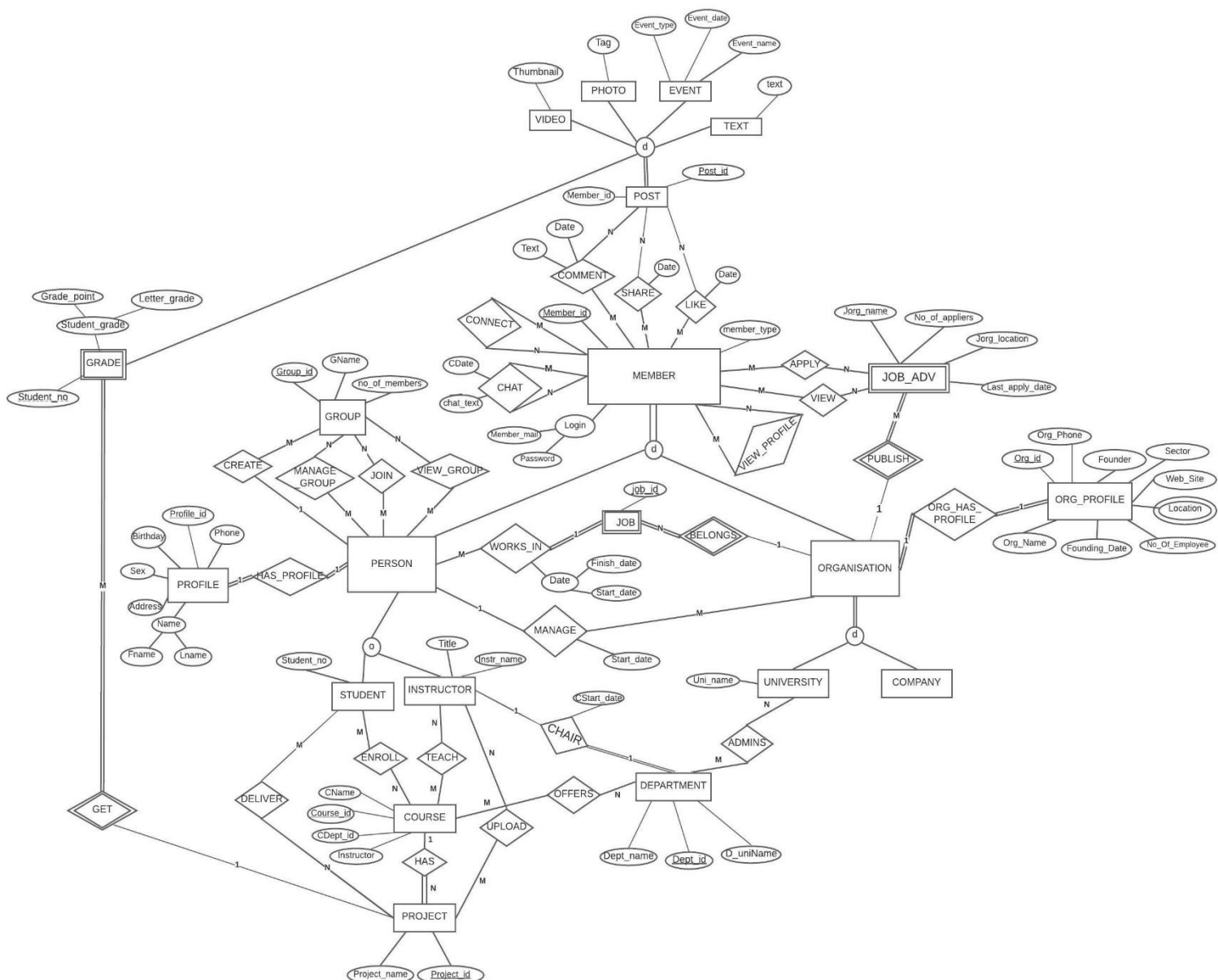
- c. Member → member\_id, login (member\_mail, password), member\_type  
Person → member\_id  
Profile → profile\_id, phone, birthday, sex, address, name (FName, LName)  
Student → student\_no  
Instructor → title, instr\_name  
Course → course\_id, CDept\_id, CName, Instructor  
Project → project\_id, project\_name  
Department → dept\_id, dept\_name, D\_uniName  
University → uni\_name  
Grade → student\_no, student\_grade (grade\_point, letter\_grade)

- d. HAS\_PROFILE (between PERSON, PROFILE)  
-A person must have only one profile.  
DELIVER (between STUDENT, PROJECT)  
-Each student can deliver more than one projects.  
UPLOAD (between INSTRUCTOR, PROJECT)  
-Each instructor can upload more than one projects.  
TEACH (between INSTRUCTOR, COURSE)  
-Instructors can teach several courses.  
ENROLL (between STUDENT, COURSE)  
-Students can enroll many courses.  
HAS (between COURSE, PROJECT)  
-There must be a project belonging to the courses.  
OFFERS (between DEPARTMENT, COURSE)  
-Departments can offer courses.  
CHAIR (between INSTRUCTOR, DEPARTMENT)  
-A department must have a chair.  
ADMINS (between DEPARTMENT, UNIVERSITY)  
-Universities can admin more than one department.

e.

- Instructors must upload a project so students can submit a project.
- Department's chair must have 'Prof' title.
- University name must be unique.
- Grades of projects must be between 0 and 100. (0 and 100 included)
- Letter grades can only be "AA", "BA", "BB", "CB", "CC", "DC", "DD", "FF".
- The person who wants to be a LinkedInMoodle member must born later than 2008-01-01.

3.



#### 4.

- We designed a system that allows students to share their grades on LinkedInMoodle.
- Students and teachers registered on Moodle, are directly connected to LinkedIn as a person.
- In addition, universities registered on Moodle, are also integrated into the organization section of LinkedIn. They can even post a job advertisement.
- Profiles of students, teachers, and universities connected to Moodle, are integrated with their LinkedIn accounts.

### 5. DESIGN-LOGICAL MODEL

#### Iteration 1:

1- PROJECT(Project\_id, Project\_name)

COURSE(Course\_id, CName, Instructor, CDept\_id)

DEPARTMENT(Dept\_id, Dept\_name, D\_uniName)

PROFILE(Profile\_id, FName, LName, Phone, Birthday, Sex, Address)

GROUP(Group\_id, GName, No\_Of\_Members)

ORG\_PROFILE(Org\_id, Org\_Name, Org\_Phone, Founder, Sector, Web\_Site,  
No\_Of\_Emp, Founding\_Date)

2,3- No weak entity and 1:1 relationship

4- PROJECT(Project\_id, Project\_name, *Course\_id*) // HAS

5- OFFERED\_COURSE(Course\_id, Dept\_id) // OFFER

6- ORG\_LOCATIONS(Org\_id, OLocation)

7- No N-ary relationship

8-

8C- MEMBER(Member\_id, Member\_mail, Member\_Password, Member\_Type)

8A- POST(Post\_id)

TEXT(Post\_id, Text)

EVENT(Post\_id, Event\_Type, Event\_Date, Event\_Name)

PHOTO(Post\_id, Tag)



VIDEO(Post\_id, Thumbnail)

GRADE(Post\_id, Letter\_Grade, Grade\_Point)

9- No union type

### Iteration 2:

1-No strong entity

2- JOB\_ADV(Member\_id, Jorg\_Name, No\_Of\_Appliers, Jorg\_Location, Last\_Apply\_Date)

JOB(Member\_id, Job\_id)

GRADE(Post\_id, Project\_id, Student\_no, Letter\_grade, Grade\_Point)

3- PROFILE(Profile\_id, FName, LName, Phone, Birthday, Sex, Address,

UMember\_id) // HAS\_PROFILE

ORG\_PROFILE(Profile\_id, Org\_Name, Org\_Phone, Founder, Sector, Web\_Site,  
No\_Of\_Emp, Founding\_Date, OMember\_id) // ORG\_HAS\_PROFILE

4- GROUP(Group\_id, GName, No\_Of\_Members, Creator\_Member\_id) // CREATE

PERSON(Member\_id, Start\_Date, Finish\_Date, Job\_id) // WORKS\_IN

ORGANISATION(Member\_id, Start\_Date, UMember\_id) // MANAGE

5- COMMENTPOST(Member\_id, Post\_id, Text, Date) // COMMENT

SHAREPOST(Member\_id, Post\_id, Date) // SHARE

LIKEPOST(Member\_id, Post\_id, Date) // LIKE

CONNECTMEM(Member\_id, UMember\_id) // CONNECT

CHATMEM(Member\_id, UMember\_id, CDate, Chat\_Text) // CHAT

APPLYJOB(Member\_id, Jorg\_Name) // APPLY

VIEWJOB(Member\_id, Jorg\_Name) // VIEW

VIEWPROFILE(Member\_id, Other\_Member\_id) // VIEW\_PROFILE

GROUPMANAGER(Group\_id, Member\_id) // MANAGE\_GROUP

JOINGROUP(Group\_id, Member\_id) // JOIN

GROUPVIEW(Group\_id, Member\_id) // VIEW\_GROUP

6,7- No multivalued attribute and N-ary relation

8-

8B- STUDENT(Member\_id, Student\_no)

INSTRUCTOR(Member\_id, Instr\_Name, Title)

8B- UNIVERSITY(Member\_id, Uni\_Name)

COMPANY(Member\_id)

### Iteration 3:

1,2- No strong and weak entity

3- DEPARTMENT(Dept\_id, Dept\_Name, D\_uniName, UMember\_id, CStartDate) // CHAIR

4- No one to many relationship

5- DELIVERPROJECT(Project\_id, Member\_id) // DELIVER

UPLOADPROJECT(Project\_id, Member\_id) // UPLOAD

ENROLLCOURSE(Course\_id, Member\_id) // ENROLL

TEACHCOURSE(Course\_id, Member\_id) // TEACH

ADMINSDEPT(Dept\_id, Member\_id) // ADMINS

## IMPLEMENTATION-PHYSICAL MODEL

### 6. SQL Scripts

```
CREATE TABLE IF NOT EXISTS MEMBER (  
    member_id INT PRIMARY KEY NOT NULL,  
    mail VARCHAR(100) NOT NULL UNIQUE,  
    Mpassword VARCHAR(100) NOT NULL,  
    member_type VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS ORGANISATION (  
    member_id INT PRIMARY KEY NOT NULL,  
    start_date DATE NULL,  
    Umember_id INT NULL,  
    FOREIGN KEY (Umember_id)  
        REFERENCES MEMBER (member_id)  
    ON DELETE CASCADE ON UPDATE CASCADE
```

);

```
CREATE TABLE IF NOT EXISTS JOB (  
    member_id INT NOT NULL,  
    job_id INT UNIQUE,  
    PRIMARY KEY (member_id , job_id),  
    FOREIGN KEY (member_id)  
        REFERENCES ORGANISATION (member_id)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS PERSON (  
    member_id INT PRIMARY KEY NOT NULL,  
    start_date DATE NULL,  
    finish_date DATE NULL,  
    job_id INT  
);
```

```
CREATE TABLE IF NOT EXISTS COMPANY (  
    member_id INT PRIMARY KEY,  
    FOREIGN KEY (member_id)  
        REFERENCES ORGANISATION (member_id)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS UNIVERSITY (  
    member_id INT PRIMARY KEY,  
    uni_name VARCHAR(45) UNIQUE,  
    FOREIGN KEY (member_id)
```

```
REFERENCES ORGANISATION (member_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS DEPARTMENT (
    dept_id INT PRIMARY KEY,
    dept_name VARCHAR(45),
    Cstart_date DATE NULL,
    Umember_id INT,
    FOREIGN KEY (Umember_id)
        REFERENCES PERSON (member_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS STUDENT (
    member_id INT PRIMARY KEY,
    student_no INT UNIQUE
);
```

```
CREATE TABLE IF NOT EXISTS INSTRUCTOR (
    member_id INT PRIMARY KEY,
    instr_name VARCHAR(45),
    title VARCHAR(45) NULL
);
```

```
CREATE TABLE IF NOT EXISTS ORG_PROFILE (
    org_id INT PRIMARY KEY NOT NULL,
    org_name VARCHAR(45),
    org_phone VARCHAR(45),
```

```

founder VARCHAR(45),
sector VARCHAR(45),
website VARCHAR(100),
no_of_employee INT,
found_date DATE,
Omember_id INT,
FOREIGN KEY (Omember_id)
    REFERENCES ORGANISATION (member_id)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS COURSE (
    course_id INT PRIMARY KEY,
    Cname VARCHAR(45),
    instr VARCHAR(45),
    Cdept_id INT,
    FOREIGN KEY (Cdept_id)
        REFERENCES DEPARTMENT (dept_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS PROJECT (
    project_id INT PRIMARY KEY,
    project_name VARCHAR(45),
    course_id INT,
    FOREIGN KEY (course_id)
        REFERENCES COURSE (course_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```
CREATE TABLE IF NOT EXISTS PROFILE (  
    profile_id INT PRIMARY KEY NOT NULL,  
    Fname VARCHAR(45),  
    Lname VARCHAR(45),  
    phone VARCHAR(45),  
    birthday DATE,  
    sex CHAR(10),  
    address VARCHAR(45),  
    Umember_id INT,  
    FOREIGN KEY (Umember_id)  
        REFERENCES PERSON (member_id)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS POST (  
    post_id INT PRIMARY KEY,  
    member_id INT,  
    FOREIGN KEY (member_id)  
        REFERENCES MEMBER (member_id)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS GRADE (  
    post_id INT,  
    project_id INT,  
    student_no VARCHAR(45),  
    letter_grade CHAR(10),  
    grade_point INT,
```

```

member_id INT,
PRIMARY KEY (post_id , project_id , student_no),
FOREIGN KEY (member_id)
    REFERENCES MEMBER (member_id)
    ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (post_id)
    REFERENCES POST (post_id)
    ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (project_id)
    REFERENCES PROJECT (project_id)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS VIDEO (
    post_id INT PRIMARY KEY,
    thumbnail VARCHAR(45),
    member_id INT,
    FOREIGN KEY (member_id)
        REFERENCES MEMBER (member_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (post_id)
        REFERENCES POST (post_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS PHOTO (
    post_id INT PRIMARY KEY,
    tag VARCHAR(45),
    member_id INT,

```

```
FOREIGN KEY (member_id)
REFERENCES MEMBER (member_id)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (post_id)
REFERENCES POST (post_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS EVENT (
    post_id INT PRIMARY KEY,
    event_type VARCHAR(45),
    event_date DATE,
    event_name VARCHAR(45),
    member_id INT,
    FOREIGN KEY (member_id)
        REFERENCES MEMBER (member_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (post_id)
        REFERENCES POST (post_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS TEXT (
    post_id INT PRIMARY KEY,
    text VARCHAR(45),
    member_id INT,
    FOREIGN KEY (member_id)
        REFERENCES MEMBER (member_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
```



```
FOREIGN KEY (post_id)
REFERENCES POST (post_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS DGROUP (
group_id INT PRIMARY KEY,
Gname VARCHAR(45),
no_of_members INT,
Umember_id INT,
FOREIGN KEY (Umember_id)
REFERENCES PERSON (member_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS ADMINSDPT (
dept_id INT,
member_id INT,
PRIMARY KEY (dept_id , member_id),
FOREIGN KEY (dept_id)
REFERENCES DEPARTMENT (dept_id)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (member_id)
REFERENCES UNIVERSITY (member_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS TEACHCOURSE (
course_id INT,
```

```

member_id INT,
PRIMARY KEY (course_id , member_id),
FOREIGN KEY (course_id)
    REFERENCES COURSE (course_id)
    ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (member_id)
    REFERENCES INSTRUCTOR (member_id)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS ENROLLCOURSE (
    course_id INT,
    member_id INT,
    PRIMARY KEY (course_id , member_id),
    FOREIGN KEY (course_id)
        REFERENCES COURSE (course_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (member_id)
        REFERENCES STUDENT (member_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS UPLOADPROJECT (
    project_id INT,
    member_id INT,
    PRIMARY KEY (project_id , member_id),
    FOREIGN KEY (project_id)
        REFERENCES PROJECT (project_id)
        ON DELETE CASCADE ON UPDATE CASCADE,

```

```
FOREIGN KEY (member_id)
REFERENCES INSTRUCTOR (member_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS DELIVERPROJECT (
project_id INT,
member_id INT,
PRIMARY KEY (project_id , member_id),
FOREIGN KEY (project_id)
REFERENCES PROJECT (project_id)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (member_id)
REFERENCES STUDENT (member_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS GROUPMANAGER (
group_id INT,
member_id INT,
PRIMARY KEY (group_id , member_id),
FOREIGN KEY (group_id)
REFERENCES DGROUP (group_id)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (member_id)
REFERENCES PERSON (member_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```

CREATE TABLE IF NOT EXISTS VIEWGROUP (
    group_id INT,
    member_id INT,
    PRIMARY KEY (group_id , member_id),
    FOREIGN KEY (group_id)
        REFERENCES DGROUP (group_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (member_id)
        REFERENCES PERSON (member_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS JOINGROUP (
    group_id INT,
    member_id INT,
    PRIMARY KEY (group_id , member_id),
    FOREIGN KEY (group_id)
        REFERENCES DGROUP (group_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (member_id)
        REFERENCES PERSON (member_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS VIEWPROFILE (
    member_id INT,
    Umember_id INT,
    PRIMARY KEY (member_id , Umember_id),
    FOREIGN KEY (Umember_id)

```

```

REFERENCES MEMBER (member_id)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (member_id)
REFERENCES MEMBER (member_id)
ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS JOB_ADV (
    member_id INT,
    Jorg_name VARCHAR(45),
    no_of_appliers INT,
    Jorg_location VARCHAR(45),
    last_apply_date DATE,
    PRIMARY KEY (member_id , Jorg_name),
    FOREIGN KEY (member_id)
        REFERENCES ORGANISATION (member_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS VIEWJOB (
    member_id INT,
    Jorg_name VARCHAR(45),
    PRIMARY KEY (member_id , Jorg_name),
    FOREIGN KEY (member_id)
        REFERENCES MEMBER (member_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS APPLYJOB (

```

```

member_id INT,
Jorg_name VARCHAR(45),
PRIMARY KEY (member_id , Jorg_name),
FOREIGN KEY (member_id)
    REFERENCES MEMBER (member_id)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS CHATMEM (
    member_id INT,
    CDate DATE,
    chat_text VARCHAR(45),
    UMember_id INT,
    PRIMARY KEY (member_id , UMember_id),
    FOREIGN KEY (member_id)
        REFERENCES MEMBER (member_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS CONNECTMEM (
    member_id INT,
    UMember_id INT,
    PRIMARY KEY (member_id , UMember_id),
    FOREIGN KEY (member_id)
        REFERENCES PERSON (member_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS LIKEPOST (

```

```

member_id INT,
post_id INT,
PRIMARY KEY (member_id , post_id),
Ldate DATE,
FOREIGN KEY (member_id)
    REFERENCES MEMBER (member_id)
    ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (post_id)
    REFERENCES POST (post_id)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS SHAREPOST (
    member_id INT,
    post_id INT,
    PRIMARY KEY (member_id , post_id),
    Sdate DATE,
    FOREIGN KEY (member_id)
        REFERENCES MEMBER (member_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (post_id)
        REFERENCES POST (post_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS COMMENTPOST (
    member_id INT,
    post_id INT,
    PRIMARY KEY (member_id , post_id),

```

```

Ltext VARCHAR(45),
Ldate DATE,
FOREIGN KEY (member_id)
    REFERENCES MEMBER (member_id)
    ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (post_id)
    REFERENCES POST (post_id)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS ORG_LOCATIONS (
    org_id INT,
    Olocation VARCHAR(45),
    PRIMARY KEY (org_id , Olocation),
    FOREIGN KEY (org_id)
        REFERENCES ORG_PROFILE (org_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS OFFEREDCOURSE (
    course_id INT,
    dept_id INT,
    PRIMARY KEY (course_id , dept_id),
    FOREIGN KEY (course_id)
        REFERENCES COURSE (course_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (dept_id)
        REFERENCES DEPARTMENT (dept_id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```



);

## 7. Populating the database

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(1,'aygunonur@protonmail.com','123456','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(2,'arcelik@gmail.com','12345','ORGANISATION');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(3,'turktelekom@gmail.com','12345','ORGANISATION');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(4,'samitugal1907@gmail.com','123456','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(5,'bildoga7@gmail.com','123456','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(6,'turkiyefinans@gmail.com','1234','ORGANISATION');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(7,'onuraygun1999@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(8,'ciceksepeti@gmail.com','12345','ORGANISATION');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(9,'buraktahapeksen@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(10,'ozancoskun@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(11,'anilozsoy@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(12,'andres@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(13,'mert@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(14,'yagmurdurukan@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(15,'zeynepduru@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(16,'egeyazici@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(17,'alierdemsafak@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(18,'berfinerdogan@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(19,'meyrasahin@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(20,'ulassenturk@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(21,'egebilgisayar@gmail.com','12345','ORGANISATION');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(22,'rizacenkerdur@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(23,'murasmanunalir@gmail.com','12345','PERSON');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(24,'egeuniversitesi@gmail.com','12345','ORGANISATION');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(25,'necdetbudak@gmail.com','12345','PERSON');
```

```
INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (1,'2020-02-  
23',NULL,1);
```

```
INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (4,'2021-03-  
22',NULL,2);
```

```
INSERT INTO ORGANISATION (member_id,start_date,Umember_id) VALUES (2,'2020-02-  
23',1);
```

```
INSERT INTO ORGANISATION (member_id,start_date,Umember_id) VALUES (3,'2021-03-  
22',4);
```

```
INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (5,'2021-06-  
01',NULL,3);
```

```
INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (7,'2022-01-  
01',NULL,5);
```

```
INSERT INTO ORGANISATION (member_id,start_date,Umember_id) VALUES (6,'2022-01-  
01',7);
```

```
INSERT INTO ORGANISATION (member_id,start_date,Umember_id) VALUES (8,'2022-02-  
01',9);
```

```

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (9,'2020-02-01',NULL,6);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (10,'2020-02-01',NULL,7);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (11,'2020-02-01',NULL,8);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (12,'2020-02-01',NULL,9);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (13,'2020-02-01',NULL,10);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (14,'2020-02-01',NULL,11);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (15,NULL,NULL,NULL);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (16,NULL,NULL,NULL);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (17,NULL,NULL,NULL);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (18,NULL,NULL,NULL);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (19,NULL,NULL,NULL);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (20,NULL,NULL,NULL);

INSERT INTO ORGANISATION (member_id,start_date,Umember_id) VALUES (21,'2021-09-23',23);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (22,'1985-02-02',NULL,12);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (23,'2021-09-23',NULL,13);

INSERT INTO ORGANISATION (member_id,start_date,Umember_id) VALUES (24,'2018-02-02',25);

INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (25,'2018-02-02',NULL,14);

```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)
VALUES (1,'Onur','Aygün','+905318559999','1999-07-11','Erkek','Samsun',1);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)
VALUES (2,'Sami','Tuğal','+905399959999','1999-08-10','Erkek','Rize',4);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)
VALUES (3,'Doğa','Bilgin','+905319669999','2000-07-11','Kadın','İzmir',5);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)
VALUES (4,'Burak','Peksen','+90532833333','1999-08-11','Erkek','İzmir',9);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)
VALUES (5,'Ozan','Coskun','+905355559999','1999-09-11','Erkek','İzmir',10);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)
VALUES (6,'Anıl','Özsoy','+9053431259999','1999-01-03','Erkek','İzmir',11);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)
VALUES (7,'Andres','Calderon','+905318457997','1999-07-11','Erkek','İzmir',12);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)
VALUES (8,'Mert','Aydın','+905318559999','1997-07-11','Erkek','İzmir',13);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)
VALUES (9,'Yağmur','Durukan','+905318879999','1999-07-11','Kadın','İzmir',14);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)
```

```
VALUES (10,'Zeynep','Duru','+905352879999','1999-07-11','Kadın','İzmir',15);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)  
VALUES (11,'Ege','Yazıcı','+905318556999','1999-12-11','Erkek','Ankara',16);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)  
VALUES (12,'Ali Erdem','Şafak','+905318599999','1999-08-11','Erkek','Bursa',17);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)  
VALUES (13,'Berfin','Erdogan','+905318559999','1999-07-11','Kadın','Gebze',18);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)  
VALUES (14,'Meyra','Sahin','+905318559999','1999-07-11','Kadın','Ordu',19);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)  
VALUES (15,'Ulaş','Şentürk','+905418559999','1999-06-11','Erkek','Ankara',20);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)  
VALUES (16,'Rıza Cenk','Erdur','+905418559999','1974-06-11','Erkek','İzmir',22);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)  
VALUES (17,'Murat Osman','Ünalır','+905418559899','1975-06-11','Erkek','İzmir',23);
```

```
INSERT INTO PROFILE (profile_id,Fname,Lname,phone,birthday,sex,address,Umember_id)  
VALUES (18,'Necdet','Budak','+905418559899','1975-06-11','Erkek','İzmir',25);
```

```
INSERT INTO ORG_PROFILE  
(org_id,org_name,org_phone,founder,sector,website,no_of_employee,found_date,Omemb  
er_id)
```

```
VALUES (1,'Arcelik','02122553212','onur aygun','Beyaz Esya','www.arcelik.com','5000','1970-02-02',2);
```

```
INSERT INTO ORG_PROFILE
```

```
(org_id,org_name,org_phone,founder,sector,website,no_of_employee,found_date,Omemb  
er_id)
```

```
VALUES (2,'Turk Telekom','02122231212','sami  
tugal','Telekomünikasyon','www.turktelekom.com','10000','1950-10-21',3);
```

```
INSERT INTO ORG_PROFILE
```

```
(org_id,org_name,org_phone,founder,sector,website,no_of_employee,found_date,Omemb  
er_id)
```

```
VALUES (3,'Türkiye Finans','02124753892','onur aygun','Beyaz  
Esya','www.arcelik.com','5000','1970-02-02',6);
```

```
INSERT INTO ORG_PROFILE
```

```
(org_id,org_name,org_phone,founder,sector,website,no_of_employee,found_date,Omemb  
er_id)
```

```
VALUES (4,'Cicek Sepeti','02129953912','onur aygun','Beyaz  
Esya','www.arcelik.com','5000','1970-02-02',8);
```

```
INSERT INTO ORG_PROFILE
```

```
(org_id,org_name,org_phone,founder,sector,website,no_of_employee,found_date,Omemb  
er_id)
```

```
VALUES (5,'Ege Bilgisayar','02129953882','Murat Osman  
Ünalır','Eğitim','www.egebilmuh.com','30','1970-02-02',21);
```

```
INSERT INTO ORG_PROFILE
```

```
(org_id,org_name,org_phone,founder,sector,website,no_of_employee,found_date,Omemb  
er_id)
```

```
VALUES (6,'Ege Üniversitesi','02129953882','Necdet  
Budak','Eğitim','www.egeuniversitesi.com','1000','1970-02-02',24);
```

```
INSERT INTO ORG_LOCATIONS (org_id, Olocation) VALUES (1,'İstanbul');
```

```
INSERT INTO ORG_LOCATIONS (org_id, Olocation) VALUES (2,'Ankara');
```

INSERT INTO ORG\_LOCATIONS (org\_id, Olocation) VALUES (3,'Istanbul');

INSERT INTO ORG\_LOCATIONS (org\_id, Olocation) VALUES (4,'Izmir');

INSERT INTO ORG\_LOCATIONS (org\_id, Olocation) VALUES (5,'Izmir');

INSERT INTO ORG\_LOCATIONS (org\_id, Olocation) VALUES (6,'Izmir');

INSERT INTO STUDENT (member\_id,student\_no) VALUES (15,'123450');

INSERT INTO STUDENT (member\_id,student\_no) VALUES (16,'123451');

INSERT INTO STUDENT (member\_id,student\_no) VALUES (17,'123452');

INSERT INTO STUDENT (member\_id,student\_no) VALUES (18,'123453');

INSERT INTO STUDENT (member\_id,student\_no) VALUES (19,'123454');

INSERT INTO STUDENT (member\_id,student\_no) VALUES (20,'123455');

INSERT INTO INSTRUCTOR (member\_id,instr\_name,title) VALUES (22,'Rıza Cenk Erdur','Doç');

INSERT INTO INSTRUCTOR (member\_id,instr\_name,title) VALUES (23,'Murat Osman Ünalır','Prof');

INSERT INTO INSTRUCTOR (member\_id,instr\_name,title) VALUES (25,'Necdet Budak','Prof');

INSERT INTO UNIVERSITY (member\_id,uni\_name) VALUES (24,'Ege Üniversitesi');

INSERT INTO DEPARTMENT (dept\_id,dept\_name,Cstart\_date,Umember\_id) VALUES (1,'Ege Bilgisayar Mühendisliği','2021-09-23',23);

INSERT INTO COMPANY(member\_id) VALUES (2);

INSERT INTO COMPANY(member\_id) VALUES (3);

INSERT INTO COMPANY(member\_id) VALUES (6);

INSERT INTO COMPANY(member\_id) VALUES (8);

INSERT INTO COURSE (course\_id,Cname,instr,Cdept\_id) VALUES (1,'Nesneye Yönelik Programlama','Rıza Cenk Erdur',1);

```
INSERT INTO COURSE (course_id,Cname,instr,Cdept_id) VALUES (2,'Database Management','Murat Osman Ünalır',1);
```

```
INSERT INTO OFFEREDCOURSE(course_id,dept_id) VALUES (1,1);
```

```
INSERT INTO OFFEREDCOURSE(course_id,dept_id) VALUES (2,1);
```

```
INSERT INTO TEACHCOURSE (course_id,member_id) VALUES (1,22);
```

```
INSERT INTO TEACHCOURSE (course_id,member_id) VALUES (2,23);
```

```
INSERT INTO ENROLLCOURSE(course_id,member_id) VALUES (1,15);
```

```
INSERT INTO ENROLLCOURSE(course_id,member_id) VALUES (2,15);
```

```
INSERT INTO ENROLLCOURSE(course_id,member_id) VALUES (1,16);
```

```
INSERT INTO ENROLLCOURSE(course_id,member_id) VALUES (2,16);
```

```
INSERT INTO ENROLLCOURSE(course_id,member_id) VALUES (1,17);
```

```
INSERT INTO ENROLLCOURSE(course_id,member_id) VALUES (2,17);
```

```
INSERT INTO ENROLLCOURSE(course_id,member_id) VALUES (1,18);
```

```
INSERT INTO ENROLLCOURSE(course_id,member_id) VALUES (2,18);
```

```
INSERT INTO ENROLLCOURSE(course_id,member_id) VALUES (1,19);
```

```
INSERT INTO ENROLLCOURSE(course_id,member_id) VALUES (2,19);
```

```
INSERT INTO ENROLLCOURSE(course_id,member_id) VALUES (1,20);
```

```
INSERT INTO ENROLLCOURSE(course_id,member_id) VALUES (2,20);
```

```
INSERT INTO PROJECT (project_id,project_name,course_id) VALUES (1,'Design Patterns Project',1);
```

```
INSERT INTO PROJECT (project_id,project_name,course_id) VALUES (2,'LinkedInMoodle Project',2);
```

```
INSERT INTO UPLOADPROJECT (project_id,member_id) VALUES (1,22);
```

```
INSERT INTO UPLOADPROJECT (project_id,member_id) VALUES (2,23);
```



```

INSERT INTO DELIVERPROJECT (project_id,member_id) VALUES (1,15);
INSERT INTO DELIVERPROJECT (project_id,member_id) VALUES (1,16);
INSERT INTO DELIVERPROJECT (project_id,member_id) VALUES (1,17);
INSERT INTO DELIVERPROJECT (project_id,member_id) VALUES (1,18);
INSERT INTO DELIVERPROJECT (project_id,member_id) VALUES (1,19);
INSERT INTO DELIVERPROJECT (project_id,member_id) VALUES (1,20);
INSERT INTO DELIVERPROJECT (project_id,member_id) VALUES (2,15);
INSERT INTO DELIVERPROJECT (project_id,member_id) VALUES (2,16);
INSERT INTO DELIVERPROJECT (project_id,member_id) VALUES (2,17);
INSERT INTO DELIVERPROJECT (project_id,member_id) VALUES (2,18);
INSERT INTO DELIVERPROJECT (project_id,member_id) VALUES (2,19);
INSERT INTO DELIVERPROJECT (project_id,member_id) VALUES (2,20);

```

```

INSERT INTO DGROUP (group_id,Gname,no_of_members,Umember_id) VALUES (1,'Lazy
Programmers',233,1);
INSERT INTO DGROUP (group_id,Gname,no_of_members,Umember_id) VALUES (2,'Strong
Programmers',573,4);

```

```

INSERT INTO GROUPMANAGER(group_id,member_id) VALUES (1,1);
INSERT INTO GROUPMANAGER(group_id,member_id) VALUES (2,4);

```

```

INSERT INTO VIEWGROUP(group_id,member_id) VALUES (1,15);
INSERT INTO VIEWGROUP(group_id,member_id) VALUES (2,15);

```

```

INSERT INTO JOINGROUP(group_id,member_id) VALUES(1,9);
INSERT INTO JOINGROUP(group_id,member_id) VALUES(2,13);

```

```

INSERT INTO POST (post_id,member_id) VALUES(1,15);
INSERT INTO POST (post_id,member_id) VALUES(2,16);
INSERT INTO POST (post_id,member_id) VALUES(3,17);

```

```

INSERT INTO POST (post_id,member_id) VALUES(4,18);
INSERT INTO POST (post_id,member_id) VALUES(5,19);
INSERT INTO POST (post_id,member_id) VALUES(6,20);
INSERT INTO POST (post_id,member_id) VALUES(7,5);
INSERT INTO POST (post_id,member_id) VALUES(8,12);
INSERT INTO POST (post_id,member_id) VALUES(9,24);
INSERT INTO POST (post_id,member_id) VALUES(10,1);
INSERT INTO POST (post_id,member_id) VALUES(11,15);
INSERT INTO POST (post_id,member_id) VALUES(12,16);
INSERT INTO POST (post_id,member_id) VALUES(13,17);
INSERT INTO POST (post_id,member_id) VALUES(14,18);
INSERT INTO POST (post_id,member_id) VALUES(15,19);
INSERT INTO POST (post_id,member_id) VALUES(16,20);
INSERT INTO POST (post_id,member_id) VALUES(17,5);
INSERT INTO POST (post_id,member_id) VALUES(18,5);

```

```

INSERT INTO GRADE (post_id,project_id,student_no,letter_grade,grade_point,member_id)
VALUES (1,1,'123450','AA',90,15);

INSERT INTO GRADE (post_id,project_id,student_no,letter_grade,grade_point,member_id)
VALUES (2,1,'123451','BA',80,16);

INSERT INTO GRADE (post_id,project_id,student_no,letter_grade,grade_point,member_id)
VALUES (3,1,'123452','BB',70,17);

INSERT INTO GRADE (post_id,project_id,student_no,letter_grade,grade_point,member_id)
VALUES (4,1,'123453','CC',60,18);

INSERT INTO GRADE (post_id,project_id,student_no,letter_grade,grade_point,member_id)
VALUES (5,1,'123454','BB',70,19);

INSERT INTO GRADE (post_id,project_id,student_no,letter_grade,grade_point,member_id)
VALUES (6,1,'123455','CC',60,20);

INSERT INTO GRADE (post_id,project_id,student_no,letter_grade,grade_point,member_id)
VALUES (11,2,'123450','AA',90,15);

```

```
INSERT INTO GRADE (post_id,project_id,student_no,letter_grade,grade_point,member_id)
VALUES (12,2,'123451','BA',80,16);
```

```
INSERT INTO GRADE (post_id,project_id,student_no,letter_grade,grade_point,member_id)
VALUES (13,2,'123452','BB',70,17);
```

```
INSERT INTO GRADE (post_id,project_id,student_no,letter_grade,grade_point,member_id)
VALUES (14,2,'123453','CC',60,18);
```

```
INSERT INTO GRADE (post_id,project_id,student_no,letter_grade,grade_point,member_id)
VALUES (15,2,'123454','BB',70,19);
```

```
INSERT INTO GRADE (post_id,project_id,student_no,letter_grade,grade_point,member_id)
VALUES (16,2,'123455','CC',60,20);
```

```
INSERT INTO VIDEO (post_id,thumbnail,member_id) VALUES (7,'video1',5);
```

```
INSERT INTO PHOTO (post_id,tag,member_id) VALUES (17,NULL,5);
```

```
INSERT INTO TEXT (post_id,text,member_id) VALUES (18,'Yeni projeme şuradan göz
atabilirsiniz. ',5);
```

```
INSERT INTO PHOTO (post_id,tag,member_id) VALUES (8,NULL,12);
```

```
INSERT INTO EVENT (post_id,event_type,event_name,event_date,member_id) VALUES
(9,'Yüz Yüze','Programcılar Buluşuyor','2022-03-02',24);
```

```
INSERT INTO TEXT (post_id,text,member_id) VALUES (10,'Tüm bağlantılarımın dikkatine iş
arayışım devam etmektedir.',1);
```

```
INSERT INTO
JOB_ADV(member_id,Jorg_name,no_of_appliers,Jorg_location,last_apply_date) VALUES
(8,'Software Developer',49,'İzmir','2022-03-02');
```

```
INSERT INTO
JOB_ADV(member_id,Jorg_name,no_of_appliers,Jorg_location,last_apply_date) VALUES
(6,'Software Developer',49,'İzmir','2022-03-02');
```

```
INSERT INTO
JOB_ADV(member_id,Jorg_name,no_of_appliers,Jorg_location,last_apply_date) VALUES
(2,'Software Developer',49,'İzmir','2022-03-02');
```

```
INSERT INTO
JOB_ADV(member_id,Jorg_name,no_of_appliers,Jorg_location,last_apply_date) VALUES
(6,'Data Engineer',102,'İzmir','2022-04-03');
```

```
INSERT INTO
JOB_ADV(member_id,Jorg_name,no_of_appliers,Jorg_location,last_apply_date) VALUES
(24,'Araştırma Görevlisi',12,'İzmir','2022-02-03');
```

```
INSERT INTO VIEWJOB(member_id,Jorg_name) VALUES (1,'Çiçek Sepeti İş');
INSERT INTO VIEWJOB(member_id,Jorg_name) VALUES (2,'Türkiye Finans İş');
```

```
INSERT INTO VIEWPROFILE(member_id,Umember_id) VALUES (1,4);
```

```
INSERT INTO APPLYJOB(member_id,Jorg_name) VALUES (17,'Çiçek Sepeti İş');
INSERT INTO APPLYJOB(member_id,Jorg_name) VALUES (18,'Türkiye Finans İş');
```

```
INSERT INTO CHATMEM(member_id,CDate,chat_text,UMember_id) VALUES (1,'2022-01-
29','Selam!',20);
```

```
INSERT INTO CONNECTMEM(member_id,UMember_id) VALUES (1,20);
```

```
INSERT INTO LIKEPOST(member_id,post_id,Ldate) VALUES (1,1,'2022-01-29');
```

```
INSERT INTO SHAREPOST(member_id,post_id,Sdate) VALUES (16,2,'2022-01-29');
```

```
INSERT INTO COMMENTPOST(member_id,post_id,Ltext,Ldate) VALUES (16,2,'Notum çok
iyi','2022-01-29');
```

## 8. Triggers

-Organisation Start Date Control:

```
DELIMITER $$
```

```
CREATE TRIGGER start_Date_error BEFORE INSERT ON ORGANISATION
```

```
FOR EACH ROW
```

```
BEGIN
```

```
IF NEW.start_date>=DATE(NOW()) THEN
```

```
SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT = 'Başlama tarihi bugünden büyük olamaz';
```

```
END IF;
```

```
END$$
```

- Connection Error:

DELIMITER \$\$

CREATE TRIGGER cant\_connect\_yourself BEFORE INSERT ON CONNECTMEM

FOR EACH ROW

BEGIN

IF NEW.member\_id=NEW.Umember\_id THEN

SIGNAL SQLSTATE '50001' SET MESSAGE\_TEXT = 'Kişi kendisiyle bağlantı kuramaz';

END IF;

END\$\$

-Change the Date of the Liked Post to the Current Date:

DELIMITER \$\$

CREATE TRIGGER liked\_date

BEFORE INSERT ON likepost

FOR EACH ROW

BEGIN

Set new.Ldate=date(now());

END

\$\$ DELIMITER ;

-Change the Date of the Shared Post to the Current Date:

DELIMITER \$\$

CREATE TRIGGER share\_date

BEFORE INSERT ON sharepost

FOR EACH ROW

BEGIN

SET new.sdate=date(now());

END

\$\$ DELIMITER ;

- Change the Date of the Commented Post to the Current Date:

```
DELIMITER $$  
  
CREATE TRIGGER comment_date  
BEFORE INSERT ON commentpost  
FOR EACH ROW  
  
BEGIN  
  
    SET NEW.ldate = date(now());  
  
END  
  
$$ DELIMITER ;
```

-Student no Control:

```
DELIMITER //  
  
CREATE TRIGGER student_no BEFORE INSERT  
ON student  
FOR EACH ROW  
  
BEGIN  
  
    IF new.student_no < 123450 OR new.student_no > 123999 THEN  
  
        SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT = 'Öğrenci numarası 123450 ve 123999  
arasında olmalıdır.';  
  
    END IF;  
  
END //
```

## 9. Check Constraints

```
ALTER TABLE profile  
  
ADD CONSTRAINT check_profile_sex CHECK (sex in ('erkek','kadın','belirtmiyor'));
```

```
ALTER TABLE person  
  
ADD CONSTRAINT date_check CHECK (start_date <= finish_date);
```

```
ALTER TABLE grade
```

```
ADD CONSTRAINT grade_check CHECK (grade_point >= 0 AND grade_point <= 100);
```

```
ALTER TABLE job_adv
```

```
ADD CONSTRAINT employee_check CHECK (no_of_appliers >= 0);
```

```
ALTER TABLE dgroup
```

```
ADD CONSTRAINT droup_check CHECK (no_of_members > 0);
```

```
ALTER TABLE grade
```

```
ADD CONSTRAINT grade_check2 CHECK (letter_grade='AA' OR letter_grade='BA' OR  
letter_grade='BB' OR letter_grade='CB' OR letter_grade='CC' OR letter_grade='DC' OR  
letter_grade='DD' OR letter_grade='FF' );
```

```
ALTER TABLE profile
```

```
ADD CONSTRAINT age_check CHECK (birthday <= '2008-01-01');
```

```
ALTER TABLE member
```

```
ADD CONSTRAINT type_check CHECK (member_type='PERSON' OR  
member_type='ORGANISATION');
```

## 10.

### a.

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(26,'odtu@gmail.com','12345','ORGANISATION');
```

```
INSERT INTO MEMBER (member_id,mail,Mpassword,member_type) VALUES  
(27,'odtuKurucusu@gmail.com','12345','PERSON');
```

```
INSERT INTO ORGANISATION (member_id,start_date,Umember_id) VALUES  
(26,'2018-02-02',27);
```

```
INSERT INTO PERSON (member_id,start_date,finish_date,job_id) VALUES (27,'2018-  
02-02',NULL,15);
```

```
INSERT INTO UNIVERSITY (member_id,uni_name) VALUES (26,'Orta Doğu Teknik  
Üniversitesi');
```

```
INSERT INTO INSTRUCTOR (member_id,instr_name,title) VALUES (27,'Aybars Ugur','Prof');
```

```
INSERT INTO COURSE (course_id,Cname,instr,Cdept_id) VALUES (3,'Data Structures','Aybars Uğur',1);
```

```
UPDATE TEXT
```

```
SET text = 'Teşekkürler her şey için..'
```

```
WHERE post_id= 10;
```

```
UPDATE JOB_ADV
```

```
SET Jorg_location = 'Ankara'
```

```
WHERE member_id = 8;
```

```
UPDATE GRADE
```

```
SET letter_grade = 'BA'
```

```
WHERE post_id = 5 ;
```

```
DELETE FROM MEMBER WHERE member_id = 11;
```

```
DELETE FROM POST WHERE post_id = 16;
```

```
DELETE FROM DELIVERPROJECT WHERE member_id = 18;
```

b.

i)

```
SELECT Cname, COUNT(E.member_id)
```

```
FROM ENROLLCOURSE AS E
```

```
JOIN COURSE AS C ON E.course_id = C.course_id;
```

```
SELECT org_name, org_phone , sector , website , no_of_employee , found_date
```

```
FROM ORG_PROFILE
```

```
ORDER BY no_of_employee ASC;
```



```

SELECT Jorg_name , no_of_appliers, COUNT(*)
FROM JOB_ADV
GROUP BY Jorg_name
HAVING no_of_appliers > 11

```

ii)

```

SELECT FName, LName , Gname , no_of_members
FROM DGROUP
JOIN PROFILE ON PROFILE.Umember_id = DGROUP.Umember_id;

```

```

SELECT org_name, sector
FROM ORG_PROFILE
LEFT JOIN JOB ON JOB.member_id = ORG_PROFILE.Omember_id
WHERE no_of_employee > 1000

```

```

SELECT member_id, letter_grade, grade_point, project_name
FROM GRADE AS G, PROJECT AS P
WHERE G.project_id = p.project_id AND grade_point > 70 AND project_name =
('LinkedInMoodle Project')
ORDER BY grade_point DESC;

```

```

SELECT dept_name, Cname, instr
FROM DEPARTMENT, COURSE
WHERE dept_id = Cdept_id

```

iii)

```

SELECT PR.Fname , PR.Lname , T.text
FROM POST AS P, TEXT AS T, PROFILE AS PR
WHERE P.post_id = T.post_id AND P.member_id = PR.Umember_id

```

```

SELECT S.student_no
FROM STUDENT AS S,POST AS P,GRADE AS G
WHERE S.student_no = G.student_no AND G.post_id = P.post_id AND G.letter_grade
< 'CC';

```

```

SELECT member_type, FName, LName
FROM LIKEPOST AS L, MEMBER AS M, PROFILE AS P
WHERE L.member_id = M.member_id AND L.member_id = P.Umember_id

```

c.

```

SELECT org_name,sector,website
FROM ORG_PROFILE AS OP,ORGANISATION AS O
WHERE OP.Omember_id = O.member_id
AND Omember_id IN
        ( SELECT member_id
          FROM JOB_ADV
          WHERE Jorg_name = 'Software Developer');

```

```

SELECT DISTINCT M.mail , P.project_id, P.project_name, C.Cname, C.instr
FROM DELIVERPROJECT AS D , PROJECT AS P , MEMBER AS M, COURSE AS C
WHERE D.project_id = P.project_id AND D.member_id = M.member_id
AND C.course_id = P.course_id;

```

```

SELECT S.student_no , PR.Fname , PR.Lname , PRJ.project_name , G.letter_grade
FROM STUDENT AS S,POST AS P,GRADE AS G , PROFILE PR , PROJECT AS PRJ
WHERE S.student_no = G.student_no AND G.post_id = P.post_id
AND G.letter_grade < 'CC' AND PR.Umember_id = S.member_id
AND PRJ.project_id = G.project_id;

```

```
SELECT member_id,mail,member_type
FROM MEMBER AS M
WHERE member_id IN ( SELECT member_id
                     FROM POST AS P
                     WHERE M.member_id = P.member_id)
ORDER BY member_id ASC;
```

```
SELECT CName, project_name, COUNT(G.student_no), AVG(grade_point)
FROM GRADE AS G, PROJECT AS P, COURSE AS C
WHERE G.project_id = P.project_id AND C.course_id = P.course_id
GROUP BY G.project_id;
```