2 February 2022

LinkedInMoodle Database

1.

LinkedIn is a professional business application that people use to make connections in the international arena. Thanks to LinkedIn, people can transfer and develop their real-life connections to the virtual world, and in this way, LinkedIn becomes the gateway to the next step in people's careers. Linkedin gives people the opportunity to introduce themselves to the business world. It also provides a social platform environment where you can share your resume, education life, hobbies, professional competencies, social activities and future plans.

Moodle is a platform that provides an educational environment with various feedbacks by accessing the source files prepared or shared by competent instructors in the field, mostly used by universities, by moodle members who are trained in this field.

2.

A.

LinkedIn aims connecting users in their each respectable line of work. Allowing them to build a network with people who are relevant to their career. It aims to be a professional networking site with social overtones.

Moodle aims to be a learning platform which provides educators, administrators and learners with a secure, easy to access, integrated system to create learning environments.

B.

Main entities of each web application:

Main entities of LinkedIn are; Member, Group, Post, Company, Job Offer, Address, Office, Skill, University.

Main entities of Moodle are; Teacher, Student, Project, File, Faculty, Department, Course, University.

Main entities of LinkedInMoodle are; Member, Group, Post, Company, Job Offer, Address, Office, Skill, University, Teacher, Student, Project, File, Faculty, Department, Course, University, Post, Comment.

C.

Characteristics of each web application:

LinkedIn:

MEMBER

- member_id
- password

GROUP

- group_id
- name
- description

POST

- post_id
- media
- text

COMPANY

- org_id
- name

UNIVERSITY

- org_id
- name

JOB OFFER

- offer_id
- person_id
- job_title
- description
- address_id

ADDRESS

- person_id
- address_id
- country
- city
- street

OFFICE

- office_id
- name

SKILL

- person_id
- skill_id
- experience
- name
- category

MOODLE

TEACHER

- person_id
- branch
- dept_id

STUDENT

- person_id
- gpa

- grade

PROJECT

- project_id
- title
- description
- FILE
- file_id
- name
- privacy

FACULTY

- faculty_id
- name
- org_id

UNIVERSITY

- org_id
- name

DEPARTMENT

- dept_id
- name
- faculty_id

COURSE

- course_id
- name
- code

- credit

LinkedInMoodle

MEMBER

- member_id
- password

GROUP

- group_id
- name
- description

POST

- post_id
- media
- text

COMPANY

- org_id
- name

UNIVERSITY

- org_id
- name

JOB OFFER

- offer_id
- person_id

- job_title
- description
- address_id

ADDRESS

- person_id
- address_id
- country
- city
- street

OFFICE

- office_id
- name

SKILL

- person_id
- skill_id
- experience
- name
- category

TEACHER

- person_id
- branch
- dept_id

STUDENT

- person_id

- gpa
- grade

PROJECT

- project_id
- title
- description
- FILE
- file_id
- name
- privacy

FACULTY

- faculty_id
- name
- org_id

DEPARTMENT

- dept_id
- name
- faculty_id

COURSE

- course_id
- name
- code
- credit

D.

Each person must be any combination of teacher, student or member.

Each person can educate in any number of graduate levels.

Each person must live in exactly one address.

Each person can know any number of skills.

Each person can be referenced by any number of teachers.

Each teacher must work in a department.

Each teacher can upload any number of files.

Each teacher can teach any number of courses.

Each student can upload any number of files.

Each student can work on any number of projects.

Each student can enroll any number of courses.

Each member can work for any number of offices.

Each member can follow any number of companies.

Each member can apply to any number of job offers.

Each member can create any number of groups.

Each member can join any number of groups.

Each member can connect with or be connected with any number of members.

Each skill can be known by any number of people.

Each address can be a residential addres of any number of people.

Each address can be a work address of any number of job offers.

Each address can be location of any number of offices.

Each graduate level can educate any number of people.

Each graduate level must be given by a department.

Each department can give any number of graduate levels.

Each department must belong to a faculty.

Each department must be located in a address.

Each department can give any number of courses.

Each faculty can have any number of departments.

Each faculty must belong to a university.

Each course can give any number of projects.

Each course must belong to a department.

Each course can be taught by any number of teachers.

Each course can be enrolled by any number of students.

Each project must be given by a course.

Each project can be worked on any number of students.

Each file can be upladed by a teacher or a student.

Each organization must be either one of the university or company.

Each university can have any number of faculties belonging to it.

Each group must be created by exactly one member.

Each group can have any number of members who joined it.

Each job offer must be given by an office.

Each job offer can be applied by any number of members.

Each job offer must have a work address.

Each company can have any number of offices.

Each company can be followed by any number of members.

Each office must be controlled by a company.

Each offica can give any number of job offers.

Each office must be located in an address.

Each office employs any number of members as employees.

Each post are usable and reachable by any number of user.

Each post can contain any number of comment.

Each Member can messages each other.

Each Member can use post anytime.

Each Member can like posts once.

Each Member can share any number of comment to any post.

E.

For skill entity, skill's category must be either one of Software, Finearts, Science or Sports

For person entity, person must be at least one or more of teacher, student and member.

For student entity, the grade attribute must be one of 1, 2 ,3 or 4 representing which year the student is studying. For gpa attribute, gpa must be a float number in between 0 and 4.

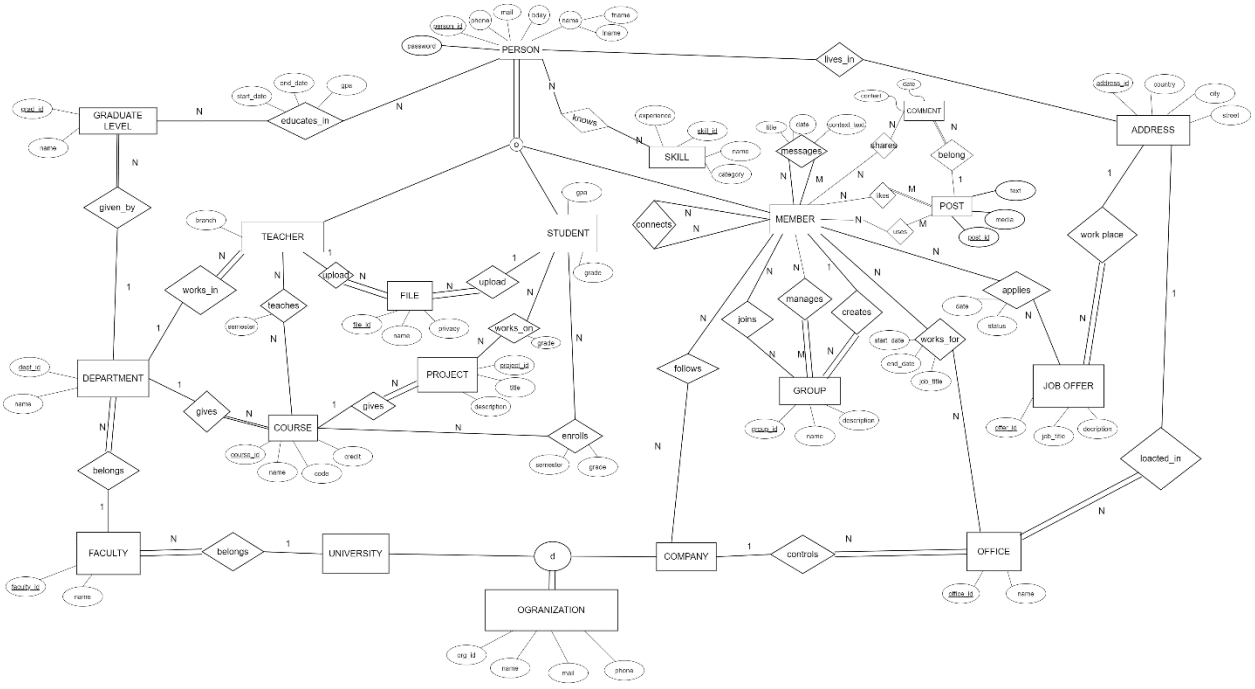
For member entity, the password attribute must be equal or more than 5 digits for security purposes.

For file entity, the owner person of the file must belong to either student or teacher type.

For course entity, credit attribute must be greater than 0 and smaller than 9.

For any entites may have a start and end date attributes, start date must come before than end date.

3.



4.

We provide a useful design to web applications based on our design. Entities and relations are designed in an understandable way. We brought together Moodle, which is about sharing lessons between

Students and Teachers and LinkedIn, which hosts people such as Students, Graduates, Employees, and Employers.

5)

Iteration 1

1)

ADDRESS(address_id, country_id, city, street)

JOB_OFFER(offer_id, job_title, description)

OFFICE(Office_id, name)

FACULTY(faculty_id, name)

DEPARTMENT(dept_id, name)

GRADUATE_LEVEL(grad_id, name)

POST(post_id, text, media)

GROUP(group_id, name, description)

SKILL(skill_id, name, category, expreience)

COMMENT(comment_id, content)

COURSE(course_id, name, code, credit)

FILE(file_id, name, privacy)

PROJECT(Project_id, title, description)

2)

---

3)

---

4)

COURSE(course_id, name, code, credit, dept_id) FK departman -> dept_id

DEPARTMENT(dept_id, name, faculty_id) FK faculty -> faculty_id

PROJECT(Project_id, title, description, course_id) FK course -> course_id

JOB_OFFER(offer_id, job_title, description, address_id) FK address -> address_id

COMMENT(comment_id, content, post_id) FK post -> post_id

GRADUATE_LEVEL(grad_id, name, dept_id) FK department -> dept_id

5)

---

6)

---

7)

---

8)

8B UNIVERSITY(org_id, name, mail, phone)

8B COMPANY(org_id, name, mail, phone)

8D PERSON(person_id, password, phone, mail, bday, fname, lname, teacherFlag, branch, studentFlag, gpa, grade, member_flag)

9)

Iteration 2

1)

---

2)

---

3)

PERSON(person_id, password, phone, mail, bday, fname, lname, teacherFlag, branch, studentFlag, gpa, grade, member_flag, address_id) FK address -> address_id

4)

GROUP(group_id, name, description, person_id) FK person -> person_id

OFFICE(Office_id, name, org_id) FK university -> org_id

FILE(file_id, name, privacy, person_id) FK person -> person_id

PERSON(person_id, password, phone, mail, bday, fname, lname, teacherFlag, branch, dept_id, studentFlag, gpa, grade, member_flag, address_id) FK address -> address_id, department - > dept_id

5)

GROUP_MANAGE(group_id, person_id) FK person -> person_id, group -> group_İd

GROUP_JOIN(group_id, person_id) FK person -> person_id, group -> group_İd

MEMBER_CONNECT(person_id,connect_person_id) FK person -> person_id and connect_person_id

MEMBER_MESSAGE(person_id, connect_person_id, title, date, context_text)

MEMBER_COMMENT(person_id, comment_id) FK person -> person_id, comment -> comment_id

MEMBER_POST(person_id, post_id) FK person -> person_id, post -> post_id

MEMBER_USES(person_id, post_id) FK person -> person_id, post -> post_id

MEMBER_APPLIES_JOB_OFFER(offer_id, person_id)

MEMBER_FOLLOWS(person_id, company_id)

PERSON_SKILL(skill_id, person_id)

PERSON_EDUCATES_IN(person_id, grad_id)

COURSE_TEACHES(person_id, course_id, semesters)

STUDENTS_WORK_ON_PROJECT(person_id, Project_id, grade)

STUDENT_ENROLL(person_id, course_id, semester, grade)

6)

---

7)

---

8)

---

9)

---

6)

```
CREATE TABLE public."ADRESS"
(
    address_id integer NOT NULL,
    country_id integer,
    city character varying(100),
    street character varying(100),
    PRIMARY KEY (address_id)
);

CREATE TABLE public."JOB_OFFER"
(
    offer_id integer NOT NULL,
    job_title character varying(100),
    description character varying(100),
    address_id integer,
    PRIMARY KEY (offer_id),
    CONSTRAINT "ADRESS" FOREIGN KEY (address_id)
        REFERENCES public."ADRESS" (address_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);
```

```
CREATE TABLE public."POST"
(
    post_id integer NOT NULL,
    text character varying(100),
    media path,
    date date,
    PRIMARY KEY (post_id)
);

CREATE TABLE public."SKILL"
(
    skill_id integer NOT NULL,
    name character varying(100),
    category character varying(100),
    expreience character varying(100),
    PRIMARY KEY (skill_id),
        CONSTRAINT category_check CHECK(category IN ('Software', 'FineArts',
'Science','Sports','language', 'programming language' ))
);

CREATE TABLE public."FACULTY"
(
    faculty_id integer NOT NULL,
    name character varying(100),
    PRIMARY KEY (faculty_id)
);

CREATE TABLE public."DEPARTMENT"
```

```
(
    dept_id integer NOT NULL,
    name character varying(100),
    faculty_id integer,
    PRIMARY KEY (dept_id),
    CONSTRAINT faculty_id FOREIGN KEY (faculty_id)
        REFERENCES public."FACULTY" (faculty_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."GRADUATE_LEVEL"
(
    grad_id integer NOT NULL,
    name character varying(100),
    dept_id integer,
    PRIMARY KEY (grad_id),
    CONSTRAINT "DEPARTMENT" FOREIGN KEY (dept_id)
        REFERENCES public."DEPARTMENT" (dept_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."COMMENT"
(
    comment_id integer NOT NULL,
    content character varying(100),
```

```
    post_id integer,
    PRIMARY KEY (comment_id),
    CONSTRAINT post FOREIGN KEY (post_id)
        REFERENCES public."POST" (post_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);



CREATE TABLE public."PERSON"
(
    person_id integer NOT NULL,
    password character varying(100),
    phone character varying(100),
    mail character varying(100),
    bday date,
    fname character varying(100),
    lname character varying(100),
    "teacherFlag" bit(1),
    branch character varying(100),
    dept_id integer,
    "studentFlag" bit(1),
    gpa character varying(100),
    grade character varying(100),
    "memberFlag" bit(1),
    address_id integer,
    PRIMARY KEY (person_id),
    FOREIGN KEY (dept_id)
```

```
    REFERENCES public."DEPARTMENT" (dept_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID,
        CONSTRAINT password_check CHECK( char_length(password)>=4 )
);

CREATE TABLE public."FILE"
(
  file_id integer NOT NULL,
  name character varying(100),
  privacy character varying(100),
  person_id integer,
  PRIMARY KEY (file_id),
  FOREIGN KEY (person_id)
    REFERENCES public."PERSON" (person_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID,
        CONSTRAINT privacy_check CHECK (privacy in ('0', '1', '2'))
);

CREATE TABLE public."GROUP"
(
  group_id integer NOT NULL,
  name character varying,
  description character varying,
  person_id integer,
  PRIMARY KEY (group_id),
```

```
    FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."UNIVERSITY"
(
    org_id integer NOT NULL,
    name character varying(100),
    mail character varying(100),
    phone character varying(100),
    PRIMARY KEY (org_id)
);

CREATE TABLE public."COMPANY"
(
    org_id integer NOT NULL,
    name character varying(100),
    mail character varying(100),
    phone character varying(100),
    PRIMARY KEY (org_id)
);

CREATE TABLE public."OFFICE"
(
    office_id integer NOT NULL,
    name character varying,
```

```
    org_id integer,
    PRIMARY KEY (office_id),
    FOREIGN KEY (org_id)
        REFERENCES public."COMPANY" (org_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);



CREATE TABLE public."COURSE"
(
    course_id integer NOT NULL,
    name character varying(100),
    code character varying(100),
    credit character varying(100),
    dept_id integer,
    PRIMARY KEY (course_id),
    FOREIGN KEY (dept_id)
        REFERENCES public."DEPARTMENT" (dept_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."PROJECT"
(
    project_id integer NOT NULL,
    title character varying(100),
```

```
    description character varying(100),
    course_id integer,
    PRIMARY KEY (project_id),
    FOREIGN KEY (course_id)
        REFERENCES public."COURSE" (course_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);



CREATE TABLE public."GROUP_MANAGE"
(
    group_id integer NOT NULL,
    person_id integer NOT NULL,
    PRIMARY KEY (group_id, person_id),
    CONSTRAINT "GROUP" FOREIGN KEY (group_id)
        REFERENCES public."GROUP" (group_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."GROUP_JOIN"
```

```
(
    group_id integer NOT NULL,
    person_id integer NOT NULL,
    PRIMARY KEY (group_id, person_id),
    CONSTRAINT "GROUP" FOREIGN KEY (group_id)
        REFERENCES public."GROUP" (group_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."MEMBER_CONNECT"
(
    person_id integer NOT NULL,
    connect_person_id integer NOT NULL,
    PRIMARY KEY (person_id, connect_person_id),
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT coonect FOREIGN KEY (connect_person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
```

```
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."MEMBER_MESSAGE"
(
    person_id integer NOT NULL,
    connect_person_id integer NOT NULL,
    title character varying(100),
    date date,
    context_text character varying(100),
    PRIMARY KEY (person_id, connect_person_id),
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "connect" FOREIGN KEY (connect_person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);


CREATE TABLE public."MEMBER_POST"
(
    person_id integer NOT NULL,
    post_id integer NOT NULL,
```

```
    PRIMARY KEY (person_id, post_id),
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "POST" FOREIGN KEY (post_id)
        REFERENCES public."POST" (post_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."MEMBER_USES"
(
    person_id integer NOT NULL,
    post_id integer NOT NULL,
    PRIMARY KEY (person_id, post_id),
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "POST" FOREIGN KEY (post_id)
        REFERENCES public."POST" (post_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);
```

```
CREATE TABLE public."MEMBER_APPLIES_JOB_OFFER"
(
    offer_id integer NOT NULL,
    person_id integer NOT NULL,
    PRIMARY KEY (offer_id, person_id),
    CONSTRAINT offer FOREIGN KEY (offer_id)
        REFERENCES public."JOB_OFFER" (offer_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."MEMBER_FOLLOWS"
(
    person_id integer NOT NULL,
    company_id integer NOT NULL,
    PRIMARY KEY (person_id, company_id),
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT company FOREIGN KEY (company_id)
```

```
        REFERENCES public."COMPANY" (org_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."PERSON_SKILL"
(
    skill_id integer NOT NULL,
    person_id integer NOT NULL,
    PRIMARY KEY (skill_id, person_id),
    CONSTRAINT "SKILL" FOREIGN KEY (skill_id)
        REFERENCES public."SKILL" (skill_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."PERSON_EDUCATES_IN"
(
    person_id integer NOT NULL,
    grad_id integer NOT NULL,
    PRIMARY KEY (person_id, grad_id),
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
```

```
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT grad FOREIGN KEY (grad_id)
        REFERENCES public."GRADUATE_LEVEL" (grad_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);
CREATE TABLE public."COURSE_TEACHES"
(
    person_id integer NOT NULL,
    course_id integer NOT NULL,
    semester character varying(100) NOT NULL,
    PRIMARY KEY (course_id, person_id, semester),
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "COURSE" FOREIGN KEY (course_id)
        REFERENCES public."COURSE" (course_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."STUDENTS_WORK_ON_PROJECT"
```

```
(
    person_id integer NOT NULL,
    project_id integer NOT NULL,
    grade integer NOT NULL,
    PRIMARY KEY (person_id, project_id, grade),
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "PROJECT" FOREIGN KEY (project_id)
        REFERENCES public."PROJECT" (project_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "GRADUATE_LEVEL" FOREIGN KEY (grade)
        REFERENCES public."GRADUATE_LEVEL" (grad_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);



CREATE TABLE public."STUDENT_ENROLL"
(
    person_id integer NOT NULL,
    course_id integer NOT NULL,
    semester character varying(100) NOT NULL,
    grade character varying(100) NOT NULL,
```

```
    PRIMARY KEY (person_id, course_id, semester, grade),
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "COURSE" FOREIGN KEY (course_id)
        REFERENCES public."COURSE" (course_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."MEMBER_COMMENT"
(
    person_id integer NOT NULL,
    comment_id integer NOT NULL,
    PRIMARY KEY (person_id, comment_id),
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "COMMENT" FOREIGN KEY (comment_id)
        REFERENCES public."COMMENT" (comment_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);
```

```
CREATE TABLE public."MEMBER_WORKS_FOR_OFFICE"
(
    person_id integer NOT NULL,
    office_id integer NOT NULL,
    start_date date,
    end_date date,
    job_title character varying(100),
    PRIMARY KEY (person_id, office_id),
    CONSTRAINT "PERSON" FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "OFFICE" FOREIGN KEY (office_id)
        REFERENCES public."OFFICE" (office_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

CREATE TABLE public."MEMBER_LIKES_POST"
(
    person_id integer NOT NULL,
    post_id integer NOT NULL,
    CONSTRAINT person PRIMARY KEY (person_id, post_id),
    FOREIGN KEY (person_id)
        REFERENCES public."PERSON" (person_id) MATCH SIMPLE
        ON UPDATE NO ACTION
```

```
    ON DELETE NO ACTION
    NOT VALID,
  CONSTRAINT "POST" FOREIGN KEY (post_id)
    REFERENCES public."POST" (post_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID
);

7)
INSERT INTO public."ADRESS" (address_id, country_id, city, street)
VALUES (1, 90, 'İzmir', 'Bornova');

INSERT INTO public."ADRESS" (address_id, country_id, city, street)
VALUES (2, 90, 'İstanbul', 'Kadıköy');

INSERT INTO public."ADRESS" (address_id, country_id, city, street)
VALUES (3, 90, 'İzmir', 'Karşıyaka');

INSERT INTO public."ADRESS" (address_id, country_id, city, street)
VALUES (4, 90, 'İzmir', 'Bayraklı');

INSERT INTO public."ADRESS" (address_id, country_id, city, street)
VALUES (5, 90, 'İzmir', 'Konak');

INSERT INTO public."ADRESS" (address_id, country_id, city, street)
VALUES (6, 90, 'İzmir', 'Göztepe');

INSERT INTO public."ADRESS" (address_id, country_id, city, street)
```

```
VALUES (7, 90, 'İstanbul', 'Beşiktaş');

INSERT INTO public."ADRESS" (address_id, country_id, city, street)
VALUES (8, 90, 'İstanbul', 'Bebek');

INSERT INTO public."ADRESS" (address_id, country_id, city, street)
VALUES (9, 90, 'İstanbul', 'Maltepe');

INSERT INTO public."ADRESS" (address_id, country_id, city, street)
VALUES (10, 90, 'Kütahya', 'Tavşanlı');



INSERT INTO public."JOB_OFFER" (offer_id, job_title, description , address_id )
VALUES (1, 'software developer', 'mac or windows?', '1');

INSERT INTO public."JOB_OFFER" (offer_id, job_title, description , address_id )
VALUES (2, 'game developer', 'do you like play game?', '1');

INSERT INTO public."JOB_OFFER" (offer_id, job_title, description , address_id )
VALUES (3, 'embedded system', 'looking for an embedded system specialist', '2');

INSERT INTO public."JOB_OFFER" (offer_id, job_title, description , address_id )
VALUES (4, 'mobile developer', 'looking for mobile developer specialist', '2');
```

```
INSERT INTO public."POST" (post_id, text, media)
VALUES (1, 'yıldızları yönetmek maharet isteyen bir iş değil.', NULL);

INSERT INTO public."POST" (post_id, text, media)
VALUES (2, 'tüm mesele bu olmak ya da olmamak...', NULL);

INSERT INTO public."POST" (post_id, text, media)
VALUES (3, 'hiçbir arzum yok, yeryüzünde masum kalmak dışında', NULL);

INSERT INTO public."SKILL" (skill_id, name, category, expreience)
VALUES (1, 'english', 'language', 'B1');

INSERT INTO public."SKILL" (skill_id, name, category, expreience)
VALUES (2, 'english', 'language', 'B2');

INSERT INTO public."SKILL" (skill_id, name, category, expreience)
VALUES (3, 'english', 'language', 'C1');

INSERT INTO public."SKILL" (skill_id, name, category, expreience)
VALUES (4, 'java', 'programming language', 'junior');

INSERT INTO public."SKILL" (skill_id, name, category, expreience)
VALUES (5, 'java', 'programming language', 'senior');
```

```
INSERT INTO public."FACULTY" (faculty_id, name)
VALUES (1, 'Faculty of Engineering');

INSERT INTO public."FACULTY" (faculty_id, name)
VALUES (2, 'Faculty of Humanities');

INSERT INTO public."FACULTY" (faculty_id, name)
VALUES (3, 'Faculty of Law');


INSERT INTO public."DEPARTMENT" (dept_id, name, faculty_id)
VALUES (1, 'computer engineering', 1);

INSERT INTO public."DEPARTMENT" (dept_id, name, faculty_id)
VALUES (2, 'mechanical engineering', 1);

INSERT INTO public."DEPARTMENT" (dept_id, name, faculty_id)
VALUES (3, 'electrical engineering', 1);


INSERT INTO public."GRADUATE_LEVEL" (grad_id, name, dept_id)
VALUES (1, 'master', 1);


INSERT INTO public."GRADUATE_LEVEL" (grad_id, name, dept_id)
VALUES (2, 'bachelor', 1);
```

```
INSERT INTO public."COMMENT" (comment_id, content, post_id)

VALUES (1, 'katılıyorum', 1);


INSERT INTO public."COMMENT" (comment_id, content, post_id)

VALUES (2, 'gerçekten öyle', 1);


INSERT INTO public."COMMENT" (comment_id, content, post_id)

VALUES (3, 'bu ifadeyi biraz abartılı buldum...', 1);
```

```
INSERT INTO public."PERSON" (person_id, password, phone, mail, bday, fname, lname, "teacherFlag",
branch, dept_id, "studentFlag", gpa, grade, "memberFlag", address_id )

VALUES (1, '123456', '+905557891597', 'alikoc@gmail.com', '15-7-1980', 'Ali', 'Koç', '1', 'turkish', 1, '0',
null, null, '0', 3);


INSERT INTO public."PERSON" (person_id, password, phone, mail, bday, fname, lname, "teacherFlag",
branch, dept_id, "studentFlag", gpa, grade, "memberFlag", address_id )

VALUES (2, '123456', '+905437891498', 'cemsakin@gmail.com', '19-1-1985', 'Cem', 'Sakin', '0', null, null,
'1', 3.45, 'master', '0', 4);


INSERT INTO public."PERSON" (person_id, password, phone, mail, bday, fname, lname, "teacherFlag",
branch, dept_id, "studentFlag", gpa, grade, "memberFlag", address_id )

VALUES (3, '123456', '+905317894573', 'cemregecer@gmail.com', '19-4-1995', 'Cemre', 'Geçer', '0', null,
null, '1', 3.31, 'master', '0', 5);


INSERT INTO public."PERSON" (person_id, password, phone, mail, bday, fname, lname, "teacherFlag",
branch, dept_id, "studentFlag", gpa, grade, "memberFlag", address_id )
```

VALUES (4, '123456', '+905387896472', 'mehmetpiskin@gmail.com', '8-4-1996', 'Mehmet', 'Pişkin', '0', null, null, '0', null, null, '1', 6);

INSERT INTO public."PERSON" (person_id, password, phone, mail, bday, fname, lname, "teacherFlag", branch, dept_id, "studentFlag", gpa, grade, "memberFlag", address_id)

VALUES (5, '123456', '+905456149437', 'alpkorkar@gmail.com', '18-1-1995', 'Alp', 'Korkar', '0', null, null, '0', null, null, '1', 7);

INSERT INTO public."PERSON" (person_id, password, phone, mail, bday, fname, lname, "teacherFlag", branch, dept_id, "studentFlag", gpa, grade, "memberFlag", address_id)

VALUES (6, '123456', '+905456149437', 'cebrail@gmail.com', '18-12-1990', 'Cebrail', 'Korkmaz', '0', null, null, '0', null, null, '1', 8);

INSERT INTO public."PERSON" (person_id, password, phone, mail, bday, fname, lname, "teacherFlag", branch, dept_id, "studentFlag", gpa, grade, "memberFlag", address_id)

VALUES (7, '123456', '+905456743417', 'berke.durmaz@gmail.com', '25-04-1989', 'Berke','Durmaz', '0', null, null, '0', null, null, '1', 1);

INSERT INTO public."FILE" (file_id, name, privacy, person_id)

VALUES (1, 'dönem projesi', '1', 1);

INSERT INTO public."FILE" (file_id, name, privacy, person_id)

VALUES (2, 'dönem projesi teslim', '0', 2);

INSERT INTO public."FILE" (file_id, name, privacy, person_id)

VALUES (3, 'dönem projesi teslim', '0', 3);

INSERT INTO public."UNIVERSITY" (org_id, name, mail, phone)

VALUES (1, 'Ege Üniversitesi', 'ege@university.com', '+905557894565');

```
INSERT INTO public."UNIVERSITY" (org_id, name, mail, phone)
VALUES (2, 'Boğaziçi Ünersitesi', 'bogazici@university.com', '+905317894578');


INSERT INTO public."COMPANY" (org_id, name, mail, phone)
VALUES (1, 'İşBank', 'isbank@kurumsal.com', '+905557891357');

INSERT INTO public."COMPANY" (org_id, name, mail, phone)
VALUES (2, 'LCW', 'lcw@kurumsal.com', '+905479651578');


INSERT INTO public."COURSE" (course_id, name, code, credit, dept_id)
VALUES (1, 'Turkish1', 'TR201', 6, 1);

INSERT INTO public."COURSE" (course_id, name, code, credit, dept_id)
VALUES (2, 'Turkish2', 'TR202', 6, 1);

INSERT INTO public."COURSE" (course_id, name, code, credit, dept_id)
VALUES (3, 'Matematik1', 'MT361', 8, 1);

INSERT INTO public."COURSE" (course_id, name, code, credit, dept_id)
VALUES (4, 'Matematik2', 'MT362', 8, 1);


INSERT INTO public."PROJECT" (project_id, title, description, course_id)
```

VALUES (1, 'kompozisyon', 'kompozisyon yazınız.', 1);

INSERT INTO public."PROJECT" (project_id, title, description, course_id)
VALUES (2, 'araştırma', 'ünlü matematikçileri araştırınız.', 3);

INSERT INTO public."PROJECT" (project_id, title, description, course_id)
VALUES (3, 'kanıtla', 'bir matematik formülünün kanıtını yapınız.', 3);

INSERT INTO public."PROJECT" (project_id, title, description, course_id)
VALUES (4, '2=2', '2=2 nin doğruluğunu kontrol ediniz.', 4);

INSERT INTO public."MEMBER_CONNECT" (person_id, connect_person_id)
VALUES (1, '2');

INSERT INTO public."MEMBER_CONNECT" (person_id, connect_person_id)
VALUES (1, '3');

INSERT INTO public."MEMBER_CONNECT" (person_id, connect_person_id)
VALUES (1, '4');

INSERT INTO public."MEMBER_CONNECT" (person_id, connect_person_id)
VALUES (1, '5');

INSERT INTO public."MEMBER_CONNECT" (person_id, connect_person_id)

```
VALUES (2, '3');

INSERT INTO public."MEMBER_CONNECT" (person_id, connect_person_id)
VALUES (2, '4');

INSERT INTO public."MEMBER_CONNECT" (person_id, connect_person_id)
VALUES (3, '5');

INSERT INTO public."MEMBER_CONNECT" (person_id, connect_person_id)
VALUES (3, '4');



INSERT INTO public."MEMBER_MESSAGE" (person_id, connect_person_id, title, date, context_text)
VALUES (1, 2, 'önemli', '2-2-2022', 'proje teslimini unutmayın.');

INSERT INTO public."MEMBER_MESSAGE" (person_id, connect_person_id, title, date, context_text)
VALUES (2, 3, 'projee!!', '2-2-2022', 'projemiz varmış?.');



INSERT INTO public."MEMBER_USES" (person_id, post_id)
VALUES (1,1);
INSERT INTO public."MEMBER_USES" (person_id, post_id)
VALUES (2,1);
INSERT INTO public."MEMBER_USES" (person_id, post_id)
```

```
VALUES (3,1);
INSERT INTO public."MEMBER_USES" (person_id, post_id)
VALUES (1,2);
INSERT INTO public."MEMBER_USES" (person_id, post_id)
VALUES (5,2);
INSERT INTO public."MEMBER_USES" (person_id, post_id)
VALUES (4,2);
INSERT INTO public."MEMBER_USES" (person_id, post_id)
VALUES (3,3);

INSERT INTO public."MEMBER_APPLIES_JOB_OFFER" (offer_id, person_id)
VALUES (1, 4);
INSERT INTO public."MEMBER_APPLIES_JOB_OFFER" (offer_id, person_id)
VALUES (2, 5);
INSERT INTO public."MEMBER_APPLIES_JOB_OFFER" (offer_id, person_id)
VALUES (4, 5);

INSERT INTO public."MEMBER_FOLLOWS" (person_id, company_id)
VALUES (4, 1);
INSERT INTO public."MEMBER_FOLLOWS" (person_id, company_id)
VALUES (5, 2);
```

```
INSERT INTO public."PERSON_SKILL" (skill_id, person_id)
VALUES (4, 2);
INSERT INTO public."PERSON_SKILL" (skill_id, person_id)
VALUES (4, 4);
INSERT INTO public."PERSON_SKILL" (skill_id, person_id)
VALUES (5, 3);
INSERT INTO public."PERSON_SKILL" (skill_id, person_id)
VALUES (5, 4);
INSERT INTO public."PERSON_SKILL" (skill_id, person_id)
VALUES (5, 5);
```

```
INSERT INTO public."PERSON_EDUCATES_IN" (person_id, grad_id)
VALUES (4, 1);
INSERT INTO public."PERSON_EDUCATES_IN" (person_id, grad_id)
VALUES (5, 2);
```

```
INSERT INTO public."COURSE_TEACHES" (person_id, course_id, semester)
VALUES (1, 1, 'güz');
INSERT INTO public."COURSE_TEACHES" (person_id, course_id, semester)
VALUES (1, 2, 'bahar');
```

```
INSERT INTO public."STUDENTS_WORK_ON_PROJECT" (person_id, project_id, grade)
```

```
VALUES (2,1,1);
INSERT INTO public."STUDENTS_WORK_ON_PROJECT" (person_id, project_id, grade)
VALUES (3,1,1);
INSERT INTO public."STUDENTS_WORK_ON_PROJECT" (person_id, project_id, grade)
VALUES (3,2,1);
INSERT INTO public."STUDENTS_WORK_ON_PROJECT" (person_id, project_id, grade)
VALUES (3,3,1);
INSERT INTO public."STUDENTS_WORK_ON_PROJECT" (person_id, project_id, grade)
VALUES (3,4,1);
INSERT INTO public."STUDENTS_WORK_ON_PROJECT" (person_id, project_id, grade)
VALUES (2,4,1);



INSERT INTO public."STUDENT_ENROLL" (person_id, course_id, semester, grade)
VALUES (2,1,'güz','1');
INSERT INTO public."STUDENT_ENROLL" (person_id, course_id, semester, grade)
VALUES (2,2,'bahar','1');
INSERT INTO public."STUDENT_ENROLL" (person_id, course_id, semester, grade)
VALUES (3,2,'güz','1');
INSERT INTO public."STUDENT_ENROLL" (person_id, course_id, semester, grade)
VALUES (3,4,'bahar','1');



INSERT INTO public."MEMBER_COMMENT" (person_id, comment_id)
VALUES (4,1);
INSERT INTO public."MEMBER_COMMENT" (person_id, comment_id)
```

```
VALUES (4,2);
INSERT INTO public."MEMBER_COMMENT" (person_id, comment_id)
VALUES (5,3);



INSERT INTO public."OFFICE" (office_id, name, org_id)
VALUES (1, 'istanbul şubesi', 1);
INSERT INTO public."OFFICE" (office_id, name, org_id)
VALUES (2, 'izmir şubesi', 1);
INSERT INTO public."OFFICE" (office_id, name, org_id)
VALUES (3, 'kocaeli şubesi', 1);
INSERT INTO public."OFFICE" (office_id, name, org_id)
VALUES (4, 'muğla şubesi', 2);
INSERT INTO public."OFFICE" (office_id, name, org_id)
VALUES (5, 'istanbul şubesi', 2);
INSERT INTO public."OFFICE" (office_id, name, org_id)
VALUES (6, 'izmir şubesi', 2);



INSERT INTO public."MEMBER_POST" (person_id, post_id)
VALUES (6,1);
INSERT INTO public."MEMBER_POST" (person_id, post_id)
VALUES (7,2);
INSERT INTO public."MEMBER_POST" (person_id, post_id)
VALUES (6,3);
```

```
INSERT INTO public."GROUP" (group_id, name, description, person_id)
VALUES (1, 'Güvercin severler derneği', 'gerçek güvercin severler için toplanma yeri.', 4);

INSERT INTO public."GROUP" (group_id, name, description, person_id)
VALUES (2, 'floresan mağdurları', 'hepimizin floresanı patladı.', 4);

INSERT INTO public."GROUP" (group_id, name, description, person_id)
VALUES (3, 'Kupon tartışma grubu', '%100 tutacak kuponlar burada.', 5);

INSERT INTO public."GROUP" (group_id, name, description, person_id)
VALUES (4, 'kitap okuma grubu', 'her hafta bir kitap', 6);



INSERT INTO public."GROUP_JOIN" (group_id, person_id)
VALUES (1, 4);

INSERT INTO public."GROUP_JOIN" (group_id, person_id)
VALUES (2, 4);

INSERT INTO public."GROUP_JOIN" (group_id, person_id)
VALUES (1, 5);

INSERT INTO public."GROUP_JOIN" (group_id, person_id)
VALUES (1, 6);

INSERT INTO public."GROUP_JOIN" (group_id, person_id)
VALUES (1, 7);
```

```
INSERT INTO public."GROUP_JOIN" (group_id, person_id)
VALUES (2, 7);

INSERT INTO public."GROUP_JOIN" (group_id, person_id)
VALUES (3, 5);

INSERT INTO public."GROUP_JOIN" (group_id, person_id)
VALUES (3, 6);



INSERT INTO public."GROUP_MANAGE" (group_id, person_id)
VALUES (1, 4);

INSERT INTO public."GROUP_MANAGE" (group_id, person_id)
VALUES (1, 5);

INSERT INTO public."GROUP_MANAGE" (group_id, person_id)
VALUES (1, 6);

INSERT INTO public."GROUP_MANAGE" (group_id, person_id)
VALUES (2, 4);

INSERT INTO public."GROUP_MANAGE" (group_id, person_id)
VALUES (3, 5);

INSERT INTO public."GROUP_MANAGE" (group_id, person_id)
VALUES (4, 6);
```

```
INSERT INTO public."MEMBER_WORKS_FOR_OFFICE" (person_id, office_id, start_date, end_date, job_title)

VALUES (4, 2, null, null, null);

INSERT INTO public."MEMBER_WORKS_FOR_OFFICE" (person_id, office_id, start_date, end_date, job_title)

VALUES (5, 2, null, null, null);

INSERT INTO public."MEMBER_WORKS_FOR_OFFICE" (person_id, office_id, start_date, end_date, job_title)

VALUES (6, 4, null, null, null);

INSERT INTO public."MEMBER_WORKS_FOR_OFFICE" (person_id, office_id, start_date, end_date, job_title)

VALUES (7, 5, null, null, null);


INSERT INTO public."MEMBER_LIKES_POST" (person_id, post_id)

VALUES (4,1);

INSERT INTO public."MEMBER_LIKES_POST" (person_id, post_id)

VALUES (5,1);

INSERT INTO public."MEMBER_LIKES_POST" (person_id, post_id)

VALUES (6,1);

INSERT INTO public."MEMBER_LIKES_POST" (person_id, post_id)

VALUES (7,1);

INSERT INTO public."MEMBER_LIKES_POST" (person_id, post_id)

VALUES (1,1);

INSERT INTO public."MEMBER_LIKES_POST" (person_id, post_id)

VALUES (2,1);

INSERT INTO public."MEMBER_LIKES_POST" (person_id, post_id)

VALUES (3,1);

INSERT INTO public."MEMBER_LIKES_POST" (person_id, post_id)
```

```
VALUES (4,2);
INSERT INTO public."MEMBER_LIKES_POST" (person_id, post_id)
VALUES (5,2);
INSERT INTO public."MEMBER_LIKES_POST" (person_id, post_id)
VALUES (1,3);
INSERT INTO public."MEMBER_LIKES_POST" (person_id, post_id)
VALUES (2,3);
INSERT INTO public."MEMBER_LIKES_POST" (person_id, post_id)
VALUES (3,3);

8)
CREATE OR REPLACE FUNCTION add_active_student()
  RETURNS trigger AS
$$
BEGIN
IF NEW.end_date = Null  THEN
        INSERT INTO STUDENT(student_id,gpa,grade)
        VALUES(NEW.person_id,NEW.gpa,'1');

END IF;
END;
$$
LANGUAGE 'plpgsql';

CREATE TRIGGER add_active_student_trigger
  AFTER INSERT
  ON PERSON_EDUCATES_IN
  FOR EACH ROW
  EXECUTE PROCEDURE add_active_student();
```

```
CREATE OR REPLACE FUNCTION remove_student()
  RETURNS trigger AS
$$
BEGIN
IF NEW.end_date != Null THEN
        DELETE FROM STUDENT WHERE student_id = OLD.student_id;

END IF;
END;
$$
LANGUAGE 'plpgsql';

CREATE TRIGGER remove_student_trigger
  AFTER UPDATE
  ON PERSON_EDUCATES_IN
  FOR EACH ROW
  EXECUTE PROCEDURE remove_student();

CREATE TRIGGER commenttime
BEFORE INSERT ON COMMENT
FOR EACH ROW
BEGIN
SET NEW.date= CURRENT_TIMESTAMP;
END
```

9)

CONSTRAINT privacy_check CHECK (privacy in '0', '1', '2')

CONSTRAINT password_check CHECK(char_length(password)>=4)

CONSTRAINT category_check CHECK(category IN ('Software', 'FineArts', 'Science','Sports', 'language', 'programming language'))

10)

A)

```
INSERT INTO public."ADRESS" (address_id, country_id, city, street)

VALUES (11, 90, 'Muğla', 'Bodrum');
```

```
UPDATE public."ADRESS"

SET street ='Fethiye'

WHERE public."ADRESS".address_id = 11;
```

```
DELETE

FROM public."ADRESS"

WHERE public."ADRESS".address_id = 11;
```

```
INSERT INTO public."POST" (post_id, text)

VALUES (4, 'fenerbahçe bu sene şampiyon');
```

```
UPDATE public."POST"

SET text ='fener bildiğimiz gibi (edit)'

WHERE public."POST".post_id = 4;
```

```
DELETE

FROM public."POST"

WHERE public."POST".post_id = 4;
```

```
INSERT INTO public."GROUP" (group_id, name, description, person_id)
VALUES (5, 'tofaşk', 'gerçek araba sevdalıları', 7);

UPDATE public."GROUP"
SET description ='gerçek araba sevdalıları 2022 devaaaam'
WHERE public."GROUP".group_id = 5;

DELETE
FROM public."GROUP"
WHERE public."GROUP".group_id = 5;
```

B)

Single table

```
//person name list.
SELECT public."PERSON".fname
FROM public."PERSON";
```

	fname character varying (100)
1	Ali
2	Cem
3	Cemre
4	Mehmet
5	Alp
6	Cebrail
7	Berke

//group names and descriptions.

```
SELECT public."GROUP".name, public."GROUP".description
FROM public."GROUP";
```

	name character varying	description character varying
1	Güvercin severler derneği	gerçek güvercin severler için toplanma yeri.
2	floresan mağdurları	hepimizin floresanı patladı.
3	Kupon tartışma grubu	%100 tutacak kuponlar burada.
4	kitap okuma grubu	her hafta bir kitap

//job postings.

```
SELECT public."JOB_OFFER".job_title, public."JOB_OFFER".description
FROM public."JOB_OFFER";
```

	job_title character varying (100)	description character varying (100)
1	software developer	mac or windows?
2	embedded system	looking for an embedded system specialist
3	game developer	do you like play game?
4	mobile developer	looking for mobile developer specialist

Double table

// Members of group with id 1 .

```
SELECT public."PERSON".fname
FROM public."PERSON"
INNER JOIN public."GROUP_JOIN"
ON public."GROUP_JOIN".person_id =  public."PERSON".person_id
WHERE public."GROUP_JOIN".group_id = 1;
```

	fname character varying (100)
1	Mehmet
2	Alp
3	Cebrail
4	Berke

// Posted comments with id 1 .

```
SELECT public."POST".text , public."COMMENT".content
FROM public."POST"
INNER JOIN public."COMMENT"
ON public."POST".post_id =  public."COMMENT".post_id
WHERE public."POST".post_id = 1;
```

	text character varying (100)	content character varying (100)
1	yıldızları yönetmek maharet isteyen bir iş değil.	katılıyorum
2	yıldızları yönetmek maharet isteyen bir iş değil.	gerçekten öyle
3	yıldızları yönetmek maharet isteyen bir iş değil.	bu ifadeyi biraz abartılı buldum...

// job postings applied by person with personid 5.

```
SELECT public."JOB_OFFER".job_title, public."JOB_OFFER".description
FROM public."JOB_OFFER"
INNER JOIN public."MEMBER_APPLIES_JOB_OFFER"
ON public."JOB_OFFER".offer_id =  public."MEMBER_APPLIES_JOB_OFFER".offer_id
WHERE public."MEMBER_APPLIES_JOB_OFFER".person_id = 5;
```

	job_title character varying (100)	description character varying (100)
1	game developer	do you like play game?
2	mobile developer	looking for mobile developer specialist

// company offices with id 1.

```
SELECT office_id, public."OFFICE".name, public."COMPANY".name

FROM public."OFFICE"

INNER JOIN public."COMPANY"

ON public."OFFICE".org_id =  public."COMPANY".org_id

WHERE public."OFFICE".org_id = 1;
```

	office_id integer	name character varying	name character varying (100)
1	1	istanbul şubesi	İşBank
2	2	izmir şubesi	İşBank
3	3	kocaeli şubesi	İşBank

Triple table

```
// senior java developers name list.

SELECT public."PERSON".fname, public."PERSON".lname, public."SKILL".name, public."SKILL".category, public."SKILL".expreience

FROM public."PERSON_SKILL"

INNER JOIN public."PERSON"

ON public."PERSON_SKILL".person_id =  public."PERSON".person_id

INNER JOIN public."SKILL"

ON public."SKILL".skill_id =  public."PERSON_SKILL".skill_id

WHERE public."PERSON_SKILL".skill_id = 5;
```

	fname character varying (100)	lname character varying (100)	name character varying (100)	category character varying (100)	expreience character varying (100)
1	Cemre	Geçer	java	programming language	senior
2	Mehmet	Pişkin	java	programming language	senior
3	Alp	Korkar	java	programming language	senior

```
// The name of the group with id 1 and the name of its members.

SELECT public."GROUP".name, public."PERSON".fname, public."PERSON".Lname

FROM public."PERSON"

INNER JOIN public."GROUP_JOIN"
```

ON public."GROUP_JOIN".person_id = public."PERSON".person_id

INNER JOIN public."GROUP"

ON public."GROUP".group_id = public."GROUP_JOIN".group_id

WHERE public."GROUP_JOIN".group_id = 1;

	name character varying	fname character varying (100)	lname character varying (100)
1	Güvercin severler derneği	Mehmet	Pişkin
2	Güvercin severler derneği	Alp	Korkar
3	Güvercin severler derneği	Cebrail	Korkmaz
4	Güvercin severler derneği	Berke	Durmaz

// Projects worked by cem calm with person_id of 2.

SELECT public."PERSON".fname, public."PERSON".lname, public."PROJECT".title, public."PROJECT".description

FROM public."PROJECT"

INNER JOIN public."STUDENTS_WORK_ON_PROJECT"

ON public."PROJECT".project_id = public."STUDENTS_WORK_ON_PROJECT".project_id

INNER JOIN public."PERSON"

ON public."PERSON".person_id = public."STUDENTS_WORK_ON_PROJECT".person_id

WHERE public."PERSON".person_id = 2;

	fname character varying (100)	lname character varying (100)	title character varying (100)	description character varying (100)
1	Cem	Sakin	kompozisyon	kompozisyon yazınız.
2	Cem	Sakin	2=2	2=2 nin doğruluğunu kontrol ediniz.

C) Special query

// Comments and sender names to the post with id 1.

SELECT public."POST".text, public."PERSON".fname, public."PERSON".lname, public."COMMENT".content

FROM public."POST"

```
INNER JOIN public."COMMENT"
ON public."POST".post_id =  public."COMMENT".post_id
INNER JOIN public."MEMBER_COMMENT"
ON public."MEMBER_COMMENT".comment_id =  public."COMMENT".comment_id
INNER JOIN public."PERSON"
ON public."MEMBER_COMMENT".person_id =  public."PERSON".person_id
WHERE public."POST".post_id = 1;
```

	text character varying (100)	fname character varying (100)	lname character varying (100)	content character varying (100)
1	yıldızları yönetmek maharet isteyen bir iş değil.	Mehmet	Pişkin	katılıyorum
2	yıldızları yönetmek maharet isteyen bir iş değil.	Mehmet	Pişkin	gerçekten öyle
3	yıldızları yönetmek maharet isteyen bir iş değil.	Alp	Korkar	bu ifadeyi biraz abartılı buldum...

```
// course_id 2 registered student names.
SELECT public."PERSON".fname, public."PERSON".lname, public."COURSE".name, public."COURSE".code
FROM public."STUDENT_ENROLL"
INNER JOIN public."PERSON"
ON public."STUDENT_ENROLL".person_id =  public."PERSON".person_id
INNER JOIN public."ADRESS"
ON public."ADRESS".address_id =  public."PERSON".address_id
INNER JOIN public."COURSE"
ON public."COURSE".course_id =  public."STUDENT_ENROLL".course_id
WHERE public."STUDENT_ENROLL".course_id = 2;
```

	fname character varying (100)	lname character varying (100)	name character varying (100)	code character varying (100)
1	Cem	Sakin	Turkish2	TR202
2	Cemre	Geçer	Turkish2	TR202

```
// people living in izmir.
SELECT  public."PERSON".fname,  public."PERSON".lname,  public."PERSON".phone,
public."PERSON".mail
```

```
FROM  public."PERSON"
WHERE  address_id IN
  (SELECT address_id
        FROM public."ADRESS"
        WHERE (public."ADRESS".city) = 'İzmir');
```

	fname character varying (100)	lname character varying (100)	phone character varying (100)	mail character varying (100)
1	Ali	Koç	+905557891597	alikoc@gmail.com
2	Cem	Sakin	+905437891498	cemsakin@gmail.com
3	Cemre	Geçer	+905317894573	cemregecer@gmail.com
4	Mehmet	Pişkin	+905387896472	mehmetpiskin@gmail.com
5	Berke	Durmaz	+905456743417	berke.durmaz@gmail.com

```
//those who are currently working in an office and applying for a job.
SELECT  public."PERSON".fname, public."PERSON".lname, public."PERSON".phone
FROM  public."MEMBER_WORKS_FOR_OFFICE"
INNER JOIN public."PERSON"
ON public."PERSON".person_id =  public."MEMBER_WORKS_FOR_OFFICE".person_id
WHERE public."MEMBER_WORKS_FOR_OFFICE".person_id IN
  (SELECT person_id
        FROM public."MEMBER_APPLIES_JOB_OFFER");
```

	fname character varying (100)	lname character varying (100)	phone character varying (100)
1	Alp	Korkar	+905456149437
2	Mehmet	Pişkin	+905387896472

```
//In order of likes, the post and the sender's name, surname.
SELECT count(public."MEMBER_LIKES_POST".person_id) AS "LIKE", public."POST".text,
public."PERSON".fname, public."PERSON".lname
FROM public."MEMBER_LIKES_POST"
INNER JOIN public."MEMBER_POST"
ON public."MEMBER_POST".post_id =  public."MEMBER_LIKES_POST".post_id
INNER JOIN public."PERSON"
```

ON public."PERSON".person_id = public."MEMBER_POST".person_id

INNER JOIN public."POST"

ON public."POST".post_id = public."MEMBER_LIKES_POST".post_id

GROUP BY public."MEMBER_LIKES_POST".post_id, public."PERSON".fname, public."PERSON".lname, public."POST".text

ORDER BY count(public."MEMBER_LIKES_POST".person_id) DESC;

	LIKE bigint	text character varying (100)	fname character varying (100)	lname character varying (100)
1	7	yıldızları yönetmek maharet isteyen bir iş değil.	Cebrail	Korkmaz
2	3	hiçbir arzum yok, yeryüzünde masum kalmak dışında	Cebrail	Korkmaz
3	2	tüm mesele bu olmak ya da olmamak...	Berke	Durmaz