Ege University

DATABASE MANAGEMENT

Project : LinkedinMoodle Database

Authors: Samet Çerezci, Eda Güneş, Ali Safarli

Due date: 2 February 2022

2
0 5180000045 / 5180000043 / 5190000757
2
2

Ege University

Introduction to the program

In a nutshell, in this project, we will design a database model for a mixture of two popular web applications' (LinkedIn and Moodle) database models (We will call our model as LinkedinMoodle).

Analysis Phase

1. A brief description for the each web application:

LinkedIn is a website for ensuring connection between people and businesses. It lets members create their own profile page where members can give information about their education, skills, interests, work experience and so on. Companies can advertise themselves and search for employees. Also LinkedIn provides a community to share their experience with other members.

Moodle is a website that is designed for universities.Each university has different web pages related to departments. Teachers can share course materials with students. Students can view their related courses content and there are submit areas where they can upload their homework. Also Moodle provides users to access their uploaded files anywhere at any time with cloud storage support.

2. An analysis report for each web application:

LinkedIn

a) What is the aim of the application:

Linkedin is a social network based on business purposes. In LinkedIn as other social networks we can share posts, comment, like and share it with each other. Also it allows members to connect with each other which we can call a relationship in real life. And the main purpose of Linkedin is that it is mostly used for career development and it allows job seekers to post their CVs and employers to post jobs. Users also can find pages of companies that users are interested in.

b) The main entities of application:

USER: This entity holds user information.
CONNECTION: This entity holds connection information between users.
GROUP: This entity holds group information.
COMPANY: This entity holds company information
POST: This entity holds post content (likes,comments,shares information).

c) The characteristics of each entity:

USER

- user_id
- first_name
- last_name
- DOB
- gender
- experience_company_id
- joined_date
- last_sign_date
- username
- cv

CONNECTION

- connection_id
- connected_member_id
- member_id

GROUP

- group_id
- name
- grade
- owner_id
- manager_id

COMPANY

- company_id
- company_name
- company_area
- employee_count
- created_date

POST

- post_id
- comment_count
- like_count
- share

- posted_member_id

d) Relationships that exist among the entities.

USER CONNECTS USER
USER CREATE GROUP
USER INVITES TO GROUP
USER HAS_EXP COMPANY
USER APPLY_JOB COMPANY
USER SHARE POST
USER LIKE POST
USER COMMENT POST
USER SHARE POST

e) The constraints that are related to entities, their characteristics and the relationships among them:

A USER can connect with multiple USERs.
A USER can be connected by multiple USERs.
A USER can create multiple GROUPs.
Each GROUP must be created by a USER.
A USER can participate in multiple GROUPs.
A USER can invite USERs to a GROUP.
A USER can have experience in multiple COMPANYs.
A USER can apply job in multiple COMPANYs.
Each job in the COMPANY can be applied by multiple USERs.
A USER can SHARE multiple POSTs.
Each POST must be shared by a USER.
A USER can like,comment on multiple POSTs .
A USER can share multiple POSTs with other USERs.
A POST can be shared by a USER.

Ege University

Moodle

a) What is the aim of the application:

Moodle aims to communicate easily between instructors and students in a university and to provide information transfer. Students create a password with a school number and they log into the system with that password. Then,they have to enter the instructor's password to register for the course. Once they have checked in, they can access the courses and notes.

b) The main entities of application:

UNIVERSITY: This entity holds what college the system belongs to.
DEPARTMENT: This entity holds what department the system belongs to.
INSTRUCTOR: This title holds which instructor teaches the course and instructor's information.
STUDENT: It is the entity that holds the characteristics of the student.
COURSE: This entity holds which courses are given by university (instructors) and the content of course (information).
ASSIGNMENT: This entity holds assignments which belong to the course.

c) The characteristics of each entity:

UNIVERSITY

- uni_id
- uni_name
- location

DEPARTMENT

- dept_id
- dept_name

INSTRUCTOR

- instructor_id
- instructor_name
- passwords
- login_date
- ins_type
- department_id

STUDENT

- student_id
- student_name
- passwords
- login_date
- grade

COURSE

- course_id
- course_name
- term
- student_count
- instructor

ASSIGNMENT

- assign_id
- assign_name
- deadline

d) Relationships that exist among the entities.

UNIVERSITY BELONG DEPARTMENT
INSTRUCTOR WORKS DEPARTMENT
INSTRUCTOR TEACHES COURSE
INSTRUCTOR ASSIGN ASSIGNMENT
COURSE ATTACHED DEPARTMENT
COURSE HAS ASSIGNMENT
STUDENT ENROLL COURSE
STUDENT DO ASSIGNMENT
STUDENT STUDY DEPARTMENT

e) The constraints that are related to entities, their characteristics and the relationships among them:

A DEPARTMENT can belong to one UNIVERSITY.
A UNIVERSITY must have one or more DEPARTMENTs.
AN INSTRUCTOR must work in one DEPARTMENT.
A DEPARTMENT can have an INSTRUCTOR.
AN INSTRUCTOR can teach one or more COURSEs.
Each COURSE must be given by an INSTRUCTOR.
AN INSTRUCTOR can assign ASSIGNMENTs for COURSE.
A COURSE can have multiple ASSIGNMENTs.
Each COURSE attached to DEPARTMENT.
A STUDENT can enroll in multiple COURSEs.
A COURSE can be given to multiple STUDENTs.
A STUDENT can do ASSIGNMENTs which are given to the related COURSE.
AN ASSIGNMENT can be done by multiple STUDENTs.
Each STUDENT studies in DEPARTMENT.
A DEPARTMENT may have multiple STUDENTs.

LinkedinMoodle

a) What is the aim of the application:

LinkedinModdle is an application that students and companies contact each other for internships or work. Members of this application can apply directly to job advertisements published by companies. In addition, just as students can view the company's pages, companies can also view the courses students have taken and the grades they have received from the assignments given by the instructor. Also, in our system, instructors can share assignments on the course page opened by their department

b) Restrictions in system:

- There must be a member to log into the system.

- The type of member can be student, instructor, employee. There may be more than one of these.

- A member must have ssn, first name, last name, birth date, sex and language.

- Users must have a different username. Also email that has not been registered before must be used when logging in.

- One member can follow multiple users and send messages. A user can also be followed by more than one person and can also receive messages from more than one person. A message must include sender and receiver with message date and context.

- A user can create a group of multiple users. Group must have a name and group member(s). A user can create multiple groups. A group can only be created by one user. A user can join more than one group. A group can have more than one user manager.

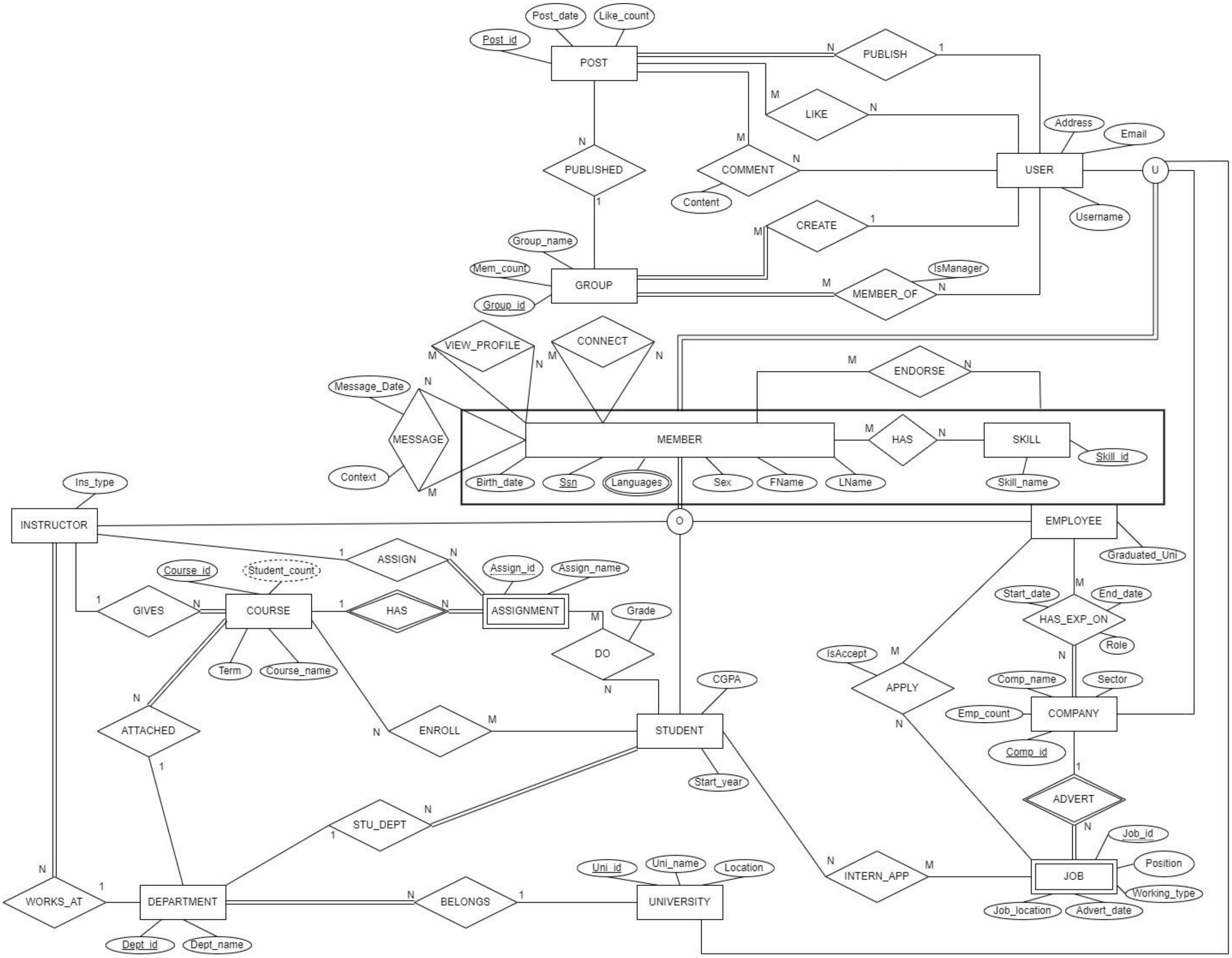
- A user can share multiple posts. A user can comment on the shared post and like the shared post. A comment has content. A post can be liked and commented on by multiple users. A post belongs to a user or a group.

- The instructor can only work in one department.

- An instructor can give more than one course. These courses may have assignments. These assignments may be done by the students. There is a department to which a course depends on. A department must contain at least one course.

- A company can have more than one job advertising. A job advertisement belongs to a company. A job has a working type(full time, part time), advert date, location and position. More than one student can apply for a job advertisement. A student can apply for more than one job. Students can apply for a job as an intern.

Design-conceptual design



Design-logical model

1.ITERATION

Step 1

POST(Post_id,Post_date,Like_count)
GROUP(Group_id,Mem_count,Group_name)
SKILL(Skill_id,Skill_name)
COURSE(Course_id,Course_name ,Student_count,Term)
DEPARTMENT(Dept_id,Dept_name)

Step 2

ASSIGNMENT(Assign_id,Course_id,Assign_name)

Step 3

-

Step 4

POST(...,Group_id)
COURSE(...,Dept_id)

Step 5

-

Step 6

-

Step 7

-

Step 8

-

Step 9

USER(User_id,Address,Email,Username)

MEMBER(Ssn,Birth_date,Sex,FName,LName,,User_id)
COMPANY(Comp_id,Emp_count,Comp_name,Sector,User_id)
UNIVERSITY(Uni_id,Uni_name,Location,User_id)

2.ITERATION

Step 1

-

Step 2

JOB(Job_id,Comp_id,Position,Working_type,Advert_date,Job_location)

Step 3

-

Step 4

POST(...,User_id)
GROUP(...,Owner_id)

Step 5

MESSAGE(Sender,Receiver,Context,Message_Date)
VIEW_PROFILE(Viewers,Viewing)
CONNECT(Following,Followers)
USER_GROUP(User_id,Group_id,IsManager)
COMMENT(User_id,Post_id,Content)
LIKE(User_id,Post_id)

Step 6

MEMBER_LANGUAGE(Ssn,Languages)

Step 7

MEMBER_ENDORSE_SKILL(Ssn,Skill_id,Endorsed_by)

Step 8

8A)

MEMBER(...,Mem_type)
INSTRUCTOR(Ssn,Ins_type)
STUDENT(Ssn,CGPA,Start_year)
EMPLOYEE(Ssn,Graduated_uni)

Step 9

-

3.ITERATION

Step 1

-

Step 2

-

Step 3

-

Step 4

INSTRUCTOR(…,Dept_id)
COURSE(…,Ins_type)
DEPARTMENT(…,Uni_id)
STUDENT(…,Dept_id)
ASSIGNMENT(…,Ssn)

Step 5

STUDENT_ASSIGNMENT(Ssn,Assign_id,Course_id,Grade)
STUDENT_COURSE(Ssn,Course_id)
STUDENT_JOB(Ssn,Job_id,Comp_id)
EMPLOYEE_JOB(Ssn,Job_id,Comp_id,IsAccept)
HAS_EXP_ON(Ssn,Comp_id,Start_date,End_date,Role)

Step 6

-

Step 7

-

Step 8

-

Step 9

-

Tables

MEMBER

Ssn	Birth_date	Sex	FName	LName	User_id	Mem_type

USER

User_id	Address	Email	Username

COMPANY

Comp_id	Emp_count	Comp_name	Sector	User_id

UNIVERSITY

Uni_id	Uni_name	Location	User_id

INSTRUCTOR

Ssn	Ins_type	Dept_id

STUDENT

Ssn	CGPA	Start_year	Dept_id

EMPLOYEE

Ssn	Graduated_uni

JOB

Job_id	Comp_id	Position	Working_type	Advert_date	Job_location

COURSE

Ege University

Course_id	Course_name	Student_count	Term	Dept_id	Ins_type

DEPARTMENT

Dept_id	Dept_name	Uni_id

POST

Post_id	Post_date	Like_count	Group_id	User_id

GROUP

Group_id	Mem_count	Group_name	Owner_id

ASSIGNMENT

Assign_id	Course_id	Assign_name	Ssn

SKILL

Skill_id	Skill_name

MESSAGE

Sender	Receiver	Context	Message_Date

VIEW_PROFILE

Viewers	Viewing

CONNECT

Following	Followers

USER_GROUP

User_id	Group_id	IsManager

COMMENT

User_id	Post_id	Content

LIKE

User_id	Post_id

MEMBER_LANGUAGE

Ssn	Languages

MEMBER_ENDORSE_SKILL

Ssn	Skill_id	Endorsed_by

STUDENT_ASSIGNMENT

Ssn	Assign_id	Course_id	Grade

STUDENT_COURSE

Ssn	Course_id

STUDENT_JOB

Ssn	Job_id	Comp_id

EMPLOYEE_JOB

Ssn	Job_id	Comp_id	IsAccept

HAS_EXP_ON

Ssn	Comp_id	Start_date	End_date	Role

Implementation-physical model

6. Creating the database

```
CREATE TABLE users(
        user_id CHAR(9) NOT NULL PRIMARY KEY,
        address VARCHAR(30) NOT NULL ,
        e_mail VARCHAR(50) NOT NULL ,
        username VARCHAR(20) NOT null,
        UNIQUE(e_mail,username)
);

CREATE TABLE members (
        ssn CHAR(9) PRIMARY KEY,
        birth_date Date NOT NULL,
        Sex CHAR NOT NULL,
        FName VARCHAR ( 50 ) NOT NULL ,
        LName VARCHAR ( 50 ) NOT NULL ,
        User_id CHAR(9) NOT NULL ,
        Mem_Type VARCHAR ( 10 ) NOT NULL ,
        foreign key(User_id) references users(user_id)
);

create table company(
        Comp_id CHAR(9) PRIMARY KEY,
        Emp_count INT ,
        Comp_Name VARCHAR ( 50 ) NOT NULL,
        Sector VARCHAR ( 50 ) NOT NULL,
        User_id CHAR(9) NOT NULL ,
        foreign key(User_id) references users(user_id),
        UNIQUE (Comp_Name)
);

create table university(
        Uni_id CHAR(5) PRIMARY KEY,
        Uni_name VARCHAR ( 50 ) NOT NULL,
        Location_name VARCHAR ( 50 ) NOT NULL,
        User_id CHAR(9) NOT NULL ,
        foreign key(User_id) references users(user_id),
        UNIQUE(Uni_name)
);

create table job(
        Job_id CHAR(5) PRIMARY KEY,
        Comp_id CHAR(9) NOT NULL,
        Position_name VARCHAR ( 50 ) NOT NULL,
        Working_Type VARCHAR ( 50 ) NOT NULL,
        Advert_date     DATE NOT NULL,
        Job_Location VARCHAR ( 50 ) NOT NULL,
        foreign key(Comp_id) references company (Comp_id)
);

create table department(
        Dept_id CHAR(5) PRIMARY KEY,
        Dept_name VARCHAR ( 20 ) NOT null,
        Uni_id CHAR(5)  NOT null,
        foreign key(Uni_id) references university (Uni_id)
);

create table Instructor(
        ssn CHAR(9) PRIMARY KEY,
        Ins_Type VARCHAR ( 20 ) NOT null,
        Dept_id CHAR(5),
        foreign key(Dept_id) references department(Dept_id)
);

create table Student(
        ssn CHAR(9) PRIMARY KEY,
        CPGA FLOAT NOT NULL ,
        Start_Year INT NOT NULL ,
        Dept_id CHAR(5),
        foreign key(Dept_id) references department(Dept_id)
);

create table employee(
        ssn CHAR(9) PRIMARY KEY,
        Graduated_Uni VARCHAR ( 20 ) NOT null
);

create table course(
        Course_id CHAR(5) PRIMARY KEY,
        Course_name VARCHAR ( 20 ) NOT NULL,
        Student_Count INT ,
        Term VARCHAR ( 10 ) NOT NULL,
        Dept_id CHAR(5) NOT NULL,
            Ins_Ssn CHAR ( 9 ) NOT null,
            foreign key(Dept_id) references department(Dept_id),
            foreign key(Ins_Ssn) references Instructor (ssn)
);

create table group_(
            Group_id CHAR(6) PRIMARY KEY,
            Group_Name VARCHAR ( 20 ) NOT NULL,
            Mem_Count INT,
            Owner_id CHAR(9) NOT NULL ,
            foreign key(Owner_id) references users(User_id)
);

create table post(
            Post_id CHAR(6) PRIMARY KEY,
            Post_date DATE NOT NULL,
            Like_Count INT default 0,
            Group_id  CHAR(6),
            User_id CHAR(9) NOT NULL ,
            foreign key(Group_id) references group_(Group_id),
            foreign key(User_id) references users(User_id)
);

create table assignment_(
            Assign_id CHAR(6) NOT NULL ,
            Course_id CHAR(6) NOT NULL ,
            Assign_Name VARCHAR (10) NOT NULL,
            Ins_ssn CHAR(9) NOT NULL,
            PRIMARY KEY(Assign_id,Course_id),
            foreign key(Course_id) references course(Course_id),
            foreign key(Ins_ssn) references instructor(ssn)
);

create table skill(
            Skill_id CHAR(3) PRIMARY KEY  ,
            Skill_name VARCHAR (20) NOT NULL
);
create table message(
            Sender_ssn CHAR(9) NOT null,
            Receiver_ssn CHAR(9) NOT null,
            Context VARCHAR (100) NOT null,
            Message_Date DATE NOT null,
            PRIMARY KEY(Sender_ssn,Receiver_ssn),
            foreign key(Sender_ssn) references members (ssn),
            foreign key(Receiver_ssn) references members (ssn)
);
create table view_profile(
            Viewers_ssn CHAR(9) NOT null,
            Viewing_ssn CHAR(9) NOT null,
            PRIMARY KEY(Viewers_ssn,Viewing_ssn),
            foreign key(Viewers_ssn) references members (ssn),
            foreign key(Viewing_ssn) references members (ssn)
);

create table Connect_(
            Following_ssn CHAR(9) NOT null,
            Followers_ssn CHAR(9) NOT null,
            PRIMARY KEY(Following_ssn,Followers_ssn),
            foreign key(Following_ssn) references members(ssn),
            foreign key(Followers_ssn) references members(ssn)
);

create table user_group(
            User_id CHAR(9) NOT null,
            Group_id CHAR(6) NOT null,
            IsManager BOOLEAN NOT NULL ,
            PRIMARY KEY(User_id,Group_id),
            foreign key(User_id) references users(User_id),
            foreign key(Group_id) references group_ (Group_id)
);

create table comment_(
            User_id CHAR(9) NOT null,
            Post_id CHAR(6) NOT null,
            Contents VARCHAR (50) NOT null,
            PRIMARY KEY (User_id,Post_id),
            foreign key(User_id) references users(User_id),
            foreign key(Post_id) references post(Post_id)
);

create table like_(
            User_id CHAR(9) NOT null,
            Post_id CHAR(6) NOT null,
            PRIMARY KEY (User_id,Post_id),
            foreign key(User_id) references users(User_id),
            foreign key(Post_id) references post(Post_id)
);

create table member_language(
            Member_Ssn CHAR(9) NOT null,
            Languages VARCHAR (15) NOT null,
            PRIMARY KEY(Member_Ssn,Languages),
            foreign key(Member_Ssn) references members (ssn)
);
```

```
create table member_endorse_skill(
        Member_Ssn CHAR(9) NOT null,
        Skill_id CHAR(3) NOT NULL,
        Endorsed_by CHAR(9) NOT null,
        PRIMARY KEY (Member_Ssn,Skill_id,Endorsed_by),
        foreign key(Member_Ssn) references members (ssn),
        foreign key(Skill_id) references skill(Skill_id)
);
create table student_assignment(
        Student_ssn CHAR(9) NOT null,
        Assign_id CHAR(6) NOT NULL ,
        Course_id CHAR(6) NOT NULL ,
        Grade INT NOT NULL ,
        PRIMARY KEY(Student_ssn,Assign_id,Course_id),
        foreign key(Assign_id,Course_id) references assignment_(Assign_id,Course_id),
        foreign key(Student_ssn) references Student(ssn)
);

create table student_course(
        Student_ssn CHAR(9) NOT null,
        Course_id CHAR(6) NOT NULL ,
        PRIMARY KEY(Student_ssn,Course_id),
        foreign key(Student_ssn) references Student(ssn),
        foreign key(Course_id) references course(Course_id)
);

create table student_job(
        Student_ssn CHAR(9) NOT null,
        Job_id CHAR(6) NOT NULL ,
        Comp_id CHAR(6) NOT NULL ,
        PRIMARY KEY(Student_ssn,Job_id,Comp_id),
        foreign key(Student_ssn) references Student(ssn),
        foreign key(Job_id) references job(Job_id),
        foreign key(Comp_id) references company (Comp_id )
);

create table employee_job(
        Employee_ssn CHAR(9) NOT null,
        Job_id CHAR(6) NOT NULL ,
        Comp_id CHAR(6) NOT NULL ,
        IsAccept BOOLEAN NOT NULL ,
        PRIMARY KEY(Employee_ssn ,Job_id,Comp_id),
        foreign key(Employee_ssn) references employee (ssn),
        foreign key(Job_id) references job(Job_id),
        foreign key(Comp_id) references company (Comp_id)
);
```

```
create table has_exp_on(
        Employee_ssn CHAR(9) NOT null,
        Comp_id CHAR(6) NOT NULL ,
        Start_Date DATE NOT NULL ,
        End_Date Date NOT NULL ,
        Roles VARCHAR(10) NOT NULL ,
        PRIMARY KEY(Employee_ssn,Comp_id),
        foreign key(Employee_ssn) references employee (ssn),
        foreign key(Comp_id) references company (Comp_id)
);

alter table has_exp_on alter column  Roles type VARCHAR(50);
```

7. Inserting the data into the database

```
INSERT INTO users (user_id,address,e_mail,username)
VALUES
        ('000000001', 'Baku', 'alisafarli@hotmail.com','xEliaze'),
        ('000000002', 'Denizli', 'sametcerezci@gmail.com','MrCODER'),
        ('000000003', 'Antalya', 'edagunes@gmail.com','Edaaguness'),
        ('000000004', 'Istanbul','arceliktr@support.com','arceliktr'),
        ('000000005', 'Istanbul ','samsung@support.com','samsungtr'),
        ('000000006', 'Izmir','webadmin@ege.edu.tr','egeuniversity'),
        ('000000007', 'Bursa','uludag.rektorluk@hs03.kep.tr','uludaguniversity' ),
        ('000000008', 'Manisa', 'vestel@support.com','Vestel'),
        ('000000009', 'Denizli', 'kmlisler@gmail.com','KML'),
  ('000000010', 'Ankara', 'mehmetkaptan@gmail.com','Memedo'),
  ('000000011', 'Mugla', 'sibelcan@gmail.com','sibelcann'),
  ('000000012', 'Izmir', 'kazimdirik@gmail.com','kazimdirik'),
  ('000000013', 'Aydın', 'muratosman@gmail.com','MOsmanUnalir'),
  ('000000014', 'Istanbul', 'anilhoca@gmail.com','anilhocaniz'),
        ('000000015', 'Yozgat', 'kadireren@gmail.com','kadireren'),
        ('000000016', 'Denizli', 'yasaragar@gmail.com','yasaragar'),
        ('000000017', 'Antalya', 'microsoft@support.com','microsofttr'),
        ('000000018', 'Mugla', 'msku@gmail.com','mskuuni'),
        ('000000019', 'Kırklareli', 'hakandemirel@gmail.com','hakooo'),
        ('000000020', 'Aydın', 'eminepolat@gmail.com','polatemine'),
        ('000000021', 'Izmir', 'mettehan@gmail.com','meteh'),
        ('000000022', 'Istanbul', 'anilaladag@gmail.com','anilaladag'),
        ('000000023', 'Istanbul', 'xholding@gmail.com','Xholding'),
        ('000000024', 'Ankara', 'apple@support.com','Appletr'),
        ('000000025', 'Istanbul', 'anilaladag@gmail.com','anilaladag_1'),
        ('000000026', 'Denizli', 'melihatarlaci@gmail.com','melihatarlaci'),
        ('000000027', 'Denizli', 'ozgeduman@gmail.com','ozgeduman'),
        ('000000028', 'Baku', 'ahmetalici@gmail.com','ahmetalici'),
        ('000000029', 'Ankara', 'kazimduman@gmail.com','kazimduman'),
        ('000000030', 'Izmir', 'eminekalayci@gmail.com','eminekalayci')

;

INSERT  into members(ssn,birth_date,sex,fname,lname,user_id,mem_type)
VALUES
        ('113620442','23.11.2002','M','Ali','Safarli','000000001','student'),
        ('251936178','26.09.1999','M','Samet','Cerezci','000000002','student'),
        ('991529451','18.06.1999','F','Eda','Gunes','000000003','instructor'),
        ('895031363','17.09.1999','M','Kamil','Isler','000000009','employee'),
        ('975070373','14.02.1987','M','Mehmet','Kaptan','000000010','student'),
        ('332344725','25.04.1972','F','Sibel','Can','000000011','employee'),
        ('427030309','15.12.1935','M','Kazim','Dirik','000000012','instructor'),
        ('526493374','15.12.1965','M','Murat','Unalir','000000013','instructor'),
        ('242052516','21.04.1992','M','Anil','Guven','000000014','employee'),
        ('129935662','21.04.1992','M','Kadir','Eren','000000015','employee'),
        ('327741590','12.01.1999','M','Yasar','Agar','000000016','student'),
        ('396933008','02.12.2000','M','Hakan','Demirel','000000019','student'),
        ('595726029','09.06.1999','F','Emine','Polat','000000020','student'),
        ('652176327','12.02.1998','M','Metehan','Akmese','000000021','employee'),
        ('144656579','27.04.2000','M','Anil','Aladag','000000022','student'),
        ('059670493','21.10.1984','F','Meliha','Tarlacı','000000022','employee'),
        ('025517954','14.04.1993','F','Ozge','Duman','000000022','employee'),
        ('084064654','24.11.2001','M','Ahmet','Alıcı','000000022','employee'),
        ('788333451','02.09.1975','M','Kazım','Duman','000000022','employee'),
        ('906303067','17.05.1996','M','Emine','Kalaycı','000000022','student')

;

INSERT INTO company
VALUES
        ('10001',1,'Arcelik Turkiye','Beyaz eşya üreticisi','000000004'),
  ('10002',1,'Samsung','Elektronik','000000005'),
  ('10003',1,'Vestel','Elektronik','000000008'),
        ('10004',1,'Microsoft','Yazılım','000000017'),
        ('10005',1,'Xholding','Finans','000000023'),
        ('10006',1,'Apple','Yazılım','000000024')
;

INSERT INTO university
VALUES
        ('20001','Ege Universitesi','Izmir','000000006'),
        ('20002','Uludag Universites','Bursa','000000007'),
        ('20003','MSK Universitesi','Mugla','000000018')
;
```

```
INSERT INTO job
VALUES
        ('11001','10001','Manager','Full time','28.12.2021','Istanbul'),
        ('11002','10001','Software Engineer','Full time','28.12.2021','Istanbul'),
        ('11003','10002','Data Analyst','Part time','28.12.2021','Istanbul'),
        ('11004','10002','Chief','Full time','24.02.2021','Istanbul'),
        ('11005','10003','Worker','Full time','15.04.2021','Istanbul'),
        ('11006','10003','CEO','Full time','23.01.2022','Manisa'),
        ('11007','10003','Worker','Part time','07.12.2021','Manisa'),
        ('11008','10004','Head Assistant','Full time','17.10.2021','Antalya'),
        ('11009','10004','Office Manager','Full time','03.01.2021','Antalya'),
        ('11010','10004','Electronic Engineer','Full time','03.01.2021','Antalya'),
        ('11011','10005','Financer','Full time','23.01.2022','Istanbul'),
        ('11012','10005','stock market manager','Part time','23.01.2022','Istanbul'),
        ('11013','10005','Computer Engineer','Full time','24.02.2021','Istanbul'),
        ('11014','10006','Industrial Engineer','Full time','15.04.2021','Ankara'),
        ('11015','10006', 'Office Manager','Part time','23.01.2022','Ankara')
;

insert INTO department
VALUES
        ('21001','Computer Eng.','20001'),
        ('21002','Food Eng.','20001'),
        ('21003','Math','20001'),
        ('21004','Computer Eng.','20002'),
        ('21005','Physic','20002'),
        ('21006','Computer Eng.','20003'),
        ('21007','Food Eng.','20003'),
        ('21008','Math','20003'),
        ('21009','Mechanical Eng','20001'),
        ('21010','Industrial Eng.','20002')
;

INSERT INTO instructor
VALUES
        ('991529451','Prof','21003'),
        ('427030309','Dr','21001'),
        ('526493374','Dr','21007')
;

INSERT INTO student
VALUES
        ('113620442',3.50,2019,'21001'),
        ('975070373',3.80,2019,'21002'),
        ('327741590',3.20,2017,'21003'),
        ('396933008',2.20,2017,'21003'),
        ('595726029',3.38,2018,'21004'),
        ('144656579',3.20,2018,'21005'),
        ('906303067',3.20,2018,'21006')
;
INSERT INTO employee
VALUES
        ('895031363','Ege University'),
        ('332344725','Ankara Universitesi'),
        ('242052516','Ankara Universitesi'),
        ('129935662','Ege Universitesi'),
        ('652176327','MSK Universitesi'),
        ('059670493','Ege Universitesi'),
        ('025517954','Ankara Universitesi'),
        ('084064654','MSK Universitesi'),
        ('788333451','MSK Universitesi')
;

INSERT INTO course
VALUES
        ('21101','DBMS',1,'Autumn','21001', '427030309'),
        ('21102','Calculus 2',1,'Spring','21003', '991529451'),
        ('21103','Algorithm',1,'Autumn','21009', '526493374'),
        ('21104','Algorithm 2',1,'Spring','21005', '526493374'),
        ('21105','Calculus 1',1,'Autumn','21010', '991529451'),
        ('21106','DBMS2',1,'Spring','21008', '427030309')
;
INSERT INTO course
VALUES
        ('21107','Biyology',1,'Autumn','21001', '991529451');
INSERT INTO group_
VALUES
        ('300001','Muhendises',2,'000000002'),
        ('300002','Girisimciler',3,'000000006'),
        ('300003','Yatırımcılar',3,'000000007'),
        ('300004','Matematik sorular',3,'000000013'),
        ('300005','Fizik deney',3,'000000006')
;

INSERT INTO post
VALUES
        ('400001','19.01.2022',5,'300001','000000005'),
        ('400002','21.01.2022',1,NULL,'000000002'),
        ('400003','25.01.2022',0,'300002','000000009'),
        ('400004','25.01.2022',0,'300002','000000009'),
        ('400005','19.01.2022',0,NULL,'000000021')
;
```

```
INSERT INTO assignment_ ------------------------------------------------------------
VALUES
            ('211101','21101','ER-EER', '427030309'),
            ('211102','21102','Basic ops', '991529451'),
            ('211103','21102','HW-1', '991529451'),
            ('211104','21103','C ', '526493374'),
            ('211105','21104','C++ ', '526493374'),
            ('211106','21105','Integral ', '991529451'),
            ('211107','21106','Sql', '427030309'),
            ('211108','21103','Normal', '427030309')
;


INSERT INTO skill
VALUES
            ('501','C#'),
            ('502','Python'),
            ('503','Integral'),
            ('504','Swimming'),
            ('505','Art'),
            ('506','Analysis'),
            ('507','Hardware'),
            ('508','Coding')
;

INSERT INTO message
VALUES
            ('084064654','975070373','Merhaba','20.01.2022'),
            ('113620442', '895031363','Sağlıklı günler','01.01.2022'),
            ('129935662', '327741590','Nasıl yardımcı olabilirim?','02.12.2021'),
            ('144656579','059670493','İşe alındım mı?','02.12.2021')
;

INSERT INTO view_profile
VALUES
            ('059670493', '652176327'),
            ('129935662', '242052516'),
            ('427030309', '975070373'),
            ('991529451', '906303067'),
            ('025517954','144656579'),
            ('788333451', '113620442'),
            ('113620442', '788333451'),
            ('327741590', '059670493')
;

INSERT INTO connect_
VALUES
```

```
          ('113620442', '991529451'),
          ('396933008', '113620442'),
          ('144656579', '025517954'),
          ('788333451', '652176327'),
          ('906303067', '427030309'),
          ('332344725', '906303067')
;

INSERT INTO user_group
VALUES
  ('000000002','300001',FALSE),
          ('000000007','300001',TRUE),
          ('000000011','300002',FALSE),
          ('000000022','300002',FALSE),
          ('000000006','300002',FALSE),
          ('000000015','300003',TRUE),
          ('000000016','300003',FALSE),
          ('000000016','300004',TRUE),
          ('000000009','300004',FALSE),
          ('000000019','300005',FALSE),
          ('000000002','300005',TRUE),
          ('000000007','300005',FALSE)
;

INSERT INTO comment_
VALUES
          ('000000001','400002','Eline sağlık'),
          ('000000005','400001','Basarılar'),
          ('000000003','400003','Guzel is'),
          ('000000011','400001','Tebrikler'),
          ('000000012','400002','Mükemmel paylasım'),
          ('000000012','400004','Cok kullanışlı bir robot'),
          ('000000012','400005','Dolar yine yükselişte')
;

INSERT INTO like_
VALUES
  ('000000001','400001'),
  ('000000001','400003'),
  ('000000002','400002'),
  ('000000002','400001'),
  ('000000003','400001'),
  ('000000003','400003'),
  ('000000009','400001'),
  ('000000001','400002'),
  ('000000014','400002'),
  ('000000010','400001'),
  ('000000010','400003'),
  ('000000009','400002'),
  ('000000014','400004'),
  ('000000012','400004'),
  ('000000012','400005'),
  ('000000002','400005')
;

INSERT INTO member_language
VALUES
          ('113620442','English'),
          ('251936178','French'),
          ('427030309','English'),
          ('113620442','German'),
          ('595726029','English'),
          ('084064654','Italian'),
          ('652176327','French'),
          ('332344725','Turkish'),
          ('251936178','Italian')
;

INSERT INTO member_endorse_skill
VALUES
          ('251936178','501', '427030309'),
          ('991529451','504', '242052516'),
          ('113620442','502', '396933008'),
          ('595726029','503', '144656579'),
          ('251936178','505', '526493374'),
          ('025517954','506', '895031363'),
          ('788333451','507', '526493374'),
          ('906303067','508', '242052516')
;
INSERT INTO student_assignment --------------------------------
VALUES
          ('113620442','211101','21101',80),
          ('975070373','211102','21102',75),
          ('327741590','211103','21102',60),
          ('396933008','211104','21103',100),
          ('595726029','211105','21104',60),
          ('906303067','211106','21105',85),
          ('113620442','211107','21106',95),
          ('396933008','211108','21103',70)
;

INSERT INTO student_course  -------------------------
VALUES
          ('113620442', '21101'),
            ('975070373','21102'),
            ('327741590','21101'),
            ('396933008','21103'),
            ('595726029', '21104'),
            ('906303067','21105'),
            ('113620442','21106'),
            ('396933008', '21106')
;

INSERT INTO student_job
VALUES
            ('113620442','11002','10001'),
            ('975070373','11003','10002'),
            ('595726029','11002','10004')
;

INSERT INTO employee_job
VALUES
            ('332344725','11001','10001',TRUE),
            ('129935662','11004','10002',FALSE),
            ('788333451','11005','10003',TRUE)
;

INSERT INTO has_exp_on
VALUES
  ('895031363','10002','20.01.2016','12.10.2017','Head Assistant'),
  ('332344725','10003','10.05.2018','10.09.2019','Computer Engineer'),
  ('242052516','10004','11.11.2014','20.12.2016','CEO'),
  ('129935662','10006','12.07.2018','12.10.2017' ,'Worker'),
  ('652176327','10006','10.05.2018','10.09.2018','Computer Engineer'),
  ('059670493','10001','21.01.2021','12.10.2017','Mechanical Engineer'),
  ('025517954','10001','23.06.2020','23.01.2022','Software Engineer'),
  ('084064654','10001','20.01.2016','10.09.2018','Computer Engineer'),
  ('788333451','10005','17.05.2017','12.10.2017','Financer')
;
```

HATALI VERI ORNEKLERI

```
/*INSERT  into members(ssn,birth_date,sex,fname,lname,user_id,mem_type)
VALUES
            ('113620442','23.11.2002','X','Ali','Safarli','000000001','student');*/

/*INSERT INTO company
VALUES
            ('10008',NULL,'Arcelik turkiye','Beyaz eşya üreticisi','000000014');*/
```

```
/*INSERT INTO job
VALUES
          ('11001','10004','Manager','Fulltime','28.12.2021','Istanbul');*/
```

8. 3 Triggers for 3 different tables (1 extra for assertion):

When new data is added about a member who is joining the group, in the group table, to the related group, group member count is increasing 1.

```
CREATE OR REPLACE FUNCTION log_group_mem()
RETURNS TRIGGER
LANGUAGE PLPGSQL
AS
$$
BEGIN
          UPDATE group_ SET mem_count= mem_count+1
          WHERE group_id=(SELECT group_id
                                        FROM group_
                                        WHERE group_id=NEW.group_id);

          RETURN NEW;
END;
$$
CREATE TRIGGER log_group_mem
AFTER INSERT ON user_group
FOR EACH ROW EXECUTE PROCEDURE log_group_mem();
```

VERI ORNEK:

```
INSERT INTO user_group
VALUES
  ('000000031','300001',FALSE);
```

When a user is trying to change a username, trigger is checking the users table to find out if that username is being used by someone else. If it's used, the user can't change his username to that one. And notice is rising.

```
CREATE OR REPLACE FUNCTION username_changes()
    RETURNS TRIGGER
    LANGUAGE PLPGSQL
    AS
$$
BEGIN
    IF (SELECT U.username
        FROM users AS U
        WHERE U.username = NEW.username) IS NOT NULL THEN
        NEW.USERNAME = OLD.USERNAME ;
        raise notice 'Already user exists with this username!!';
    END IF;
    RETURN NEW;
END;
$$

CREATE TRIGGER username_changes
BEFORE UPDATE
ON USERS
FOR EACH ROW
EXECUTE PROCEDURE username_changes();
```

HATALI VERI ORNEK:

```
UPDATE USERS SET  USERNAME = 'MOsmanUnalir' WHERE USER_ID = '000000015'
```

When a user is liking some post, the related post in the post table, like the count section, is increasing 1.*/

```
CREATE OR REPLACE FUNCTION like_count_update()
RETURNS TRIGGER
LANGUAGE PLPGSQL
AS
$$
BEGIN
    UPDATE post SET like_count= like_count+1
    WHERE post_id=(SELECT post_id
                                 FROM post
                                 WHERE post_id=NEW.post_id);

    RETURN NEW;
END;
$$
CREATE TRIGGER like_count_update
AFTER INSERT ON like_
FOR EACH ROW EXECUTE PROCEDURE like_count_update();
```

VERI ORNEK:

```
/*INSERT INTO like_
```

```
VALUES
  ('000000011','400005'),
  ('000000030','400005')
;*/
```

When the instructor is grading students for assignment, the trigger is checking if the grade is less than zero (grade should be >=0).

```
CREATE OR REPLACE FUNCTION grade_control()
RETURNS TRIGGER
LANGUAGE PLPGSQL
AS
$$
BEGIN
        IF NEW.grade < 0 THEN
                RAISE EXCEPTION 'Grade can not be a negative value grade: %',
NEW.grade  USING HINT = 'Please check your grade value';
        END IF;
        RETURN NEW;
END;
$$

CREATE TRIGGER grade_control
BEFORE INSERT
ON student_assignment
FOR EACH ROW
EXECUTE PROCEDURE grade_control();
```

HATALI VERI ORNEK:

```
/*INSERT INTO student_assignment
VALUES
        ('906303067','211101','21101',-30);*/
```

9. 3 Constraints (2 extra for assertions)

```
ALTER TABLE members ADD CONSTRAINT SexControl CHECK (sex='F' or sex='M');
ALTER TABLE company  ADD CONSTRAINT CompanyEmployeeControl CHECK
(company.emp_count>=0);
```

```
ALTER TABLE job  ADD CONSTRAINT WorkingTypeControl CHECK (job.working_type='Full
time' or job.working_type='Part time');
ALTER TABLE instructor ADD CONSTRAINT ins_types CHECK (ins_type = 'Prof' OR
ins_type = 'Dr' OR ins_type='Prof Dr' or ins_type='Asst');
ALTER TABLE employee_job  ADD CONSTRAINT isaccept CHECK (isaccept =TRUE or
isaccept=FALSE);
```

10. SQL Statements:

a) INSERT, DELETE and UPDATE

INSERT

```
INSERT INTO member_language
VALUES
        ('788333451','Turkish');
INSERT INTO users (user_id,address,e_mail,username)
VALUES
        ('000000031', 'Hakkari', 'memduh@hotmail.com','memduh');
INSERT  into members(ssn,birth_date,sex,fname,lname,user_id,mem_type)
VALUES  ('159357246','23.11.1986','M','Memduh','Hisar','000000031','student');
```

DELETE

```
DELETE FROM connect_  WHERE Following_ssn='144656579';
DELETE FROM message  WHERE Sender_ssn='129935662';
DELETE FROM like_  WHERE user_id='000000002';
```

UPDATE

```
UPDATE comment_  SET contents ='Dolar düşüşte' WHERE user_id ='000000012'
and post_id='400005';
UPDATE instructor  SET ins_type ='Prof Dr' WHERE  ssn ='526493374';
UPDATE users  SET e_mail ='smtcrzc@gmail.com'  where user_id ='000000002';
```

b) 10 SELECT statements

i) 3 statements for one table

Courses which are enrolled by at least 3 students in the Autumn term.

```
SELECT course_name, student_count
FROM course
WHERE term='Autumn' and student_count > 3
ORDER BY course_name ASC;
```

Ordering provinces where at least 1 company is searching for a Software

Engineer.

```
SELECT job_location, COUNT(*)
FROM job
WHERE position_name='Software Engineer'
GROUP BY job_location
HAVING COUNT(*) > 0;
```

Listing the companies' names, their sector and how many employees are working there (employee count), which includes holding at their names.

```
SELECT DISTINCT ON (sector) sector, comp_name, emp_count
FROM company
WHERE comp_name LIKE '%holding%';
```

ii) 4 statements for at least 2 tables

Listing the groups' names and member counts created by the owner of the "Girişimciler" group.

```
SELECT Group_name, Mem_count
FROM (users AS u INNER JOIN group_ AS gr ON u.user_id = gr.owner_id)
WHERE owner_id IN
((SELECT owner_id
FROM group_
WHERE group_name = 'Girisimciler')) AND group_name<>'Girisimciler'
ORDER BY Group_name;
```

Listing of lecturers of courses without homework.

```
SELECT DISTINCT ON (course.ins_ssn) course.ins_ssn AS Instructor, course_name AS
Course
FROM course
WHERE NOT EXISTS(
                        SELECT course.course_id
                        FROM assignment_ AS assign
                        WHERE assign.course_id = course.course_id);
```

Grouping the instructors working in the departments of the university with 'Uni_id=20001' according to their instructor types and listing the number of instructors.

```
SELECT dept_name, ins_type, COUNT(*)
FROM (department AS D INNER JOIN instructor AS I ON D.dept_id=I.dept_id)
WHERE D.dept_id= I.dept_id AND D.uni_id='20001' AND
        EXISTS (        SELECT ins_type
                        FROM instructor as I,department as D
                        WHERE D.dept_id=I.dept_id)
GROUP BY ins_type, dept_name;
```

Listing the information of the members who speak more than 1 language, born after 01.01.1999.

```
SELECT *
FROM members AS mem
WHERE ssn IN
(       SELECT ssn
        FROM member_language AS mem_lan
        WHERE mem.ssn=mem_lan.member_ssn
        GROUP BY mem.Ssn
HAVING COUNT(*) > 1)  AND mem.birth_date>'01.01.1999’;
```

iii) 3 statements for at least 3 tables

Listing the company name, employee name and last name, position of the job applications that shared job advertisements between February and May 2021 and where workers were accepted.

```
SELECT fname AS Adı,lname AS Soyadı, co.comp_name, j.position_name, j.working_type,
j.advert_date
FROM employee_job AS emp, job AS j, company AS co, members AS mem
WHERE mem.ssn=emp.employee_ssn AND j.job_id=emp.job_id AND
co.comp_id= emp.comp_id AND emp.isaccept= true AND (j.advert_date >= '2021-02-01'
AND j.advert_date < '2021-05-01');
```

Listing 2 different comments from the same post of the company whose post got the most likes.

```
SELECT DISTINCT ON (contents) contents, post_id, userid
FROM COMMENT
WHERE POST_ID = (SELECT POST_ID
FROM ((POST NATURAL JOIN USERS) NATURAL JOIN COMPANY)
ORDER BY LIKE_COUNT
LIMIT 1)
LIMIT 2
```

Listing the average grades of the students who took the 'DBMS' course from 'Kazim dirik' from the given assignments.

```
SELECT ass.assign_name, AVG(grade)
FROM (student_assignment AS sa JOIN assignment_ AS ass ON
sa.assign_id=ass.assign_id)
WHERE sa.assign_id IN
             (SELECT assign_id
             FROM assignment_ AS ass
             WHERE (ass.ins_ssn, ass.course_id) IN
                    (SELECT ssn, course_id
                    FROM (members AS mem INNER JOIN course AS co ON
                     co.ins_ssn = mem.ssn)
                    WHERE fname='Kazim' AND lname='Dirik' AND
                     course_name='DBMS'))
GROUP BY ass.assign_name
;
```

c) 5 Original SELECT statements

Listing the internships of the students who took the 'DBMS' course from 'Kazim Dirik' and whose position is "Data Analyst".

```
SELECT fname, lname, sex, position_name, working_type
FROM ((student_job NATURAL JOIN job) NATURAL JOIN members)
WHERE members.ssn IN
        (SELECT student_ssn
        FROM (course NATURAL JOIN student_course)
        WHERE (ins_ssn,course_id) IN
         (SELECT ssn, course_id
          FROM (members AS mem INNER JOIN course AS co ON co.ins_ssn = mem.ssn)
          WHERE fname='Kazim' AND lname='Dirik' AND course_name='DBMS'))
AND position_name = 'Data Analyst'
ORDER BY fname DESC
;
```

Listing the first name, last name and type of instructors of the "university" users of the groups that include "deney", who are instructors in the "math" department.

```
SELECT fname, lname, ins_type
FROM (members AS mem INNER JOIN instructor AS i ON i.ssn= mem.ssn)
WHERE dept_id =
        (SELECT dept_id
        FROM (department AS dept JOIN university AS uni ON dept.uni_id=uni.uni_id)
        WHERE uni.user_id =
         (      SELECT owner_id
                FROM group_
                WHERE group_name LIKE '%deney%') AND dept_name='Math')
;
```

Lists the first name, last name and cgpa in descending order of the students who are messaged by the employee who works 'Computer Engineer' position.

```
SELECT fname, lname, cpga
FROM (members AS mem JOIN student AS s ON mem.ssn= s.ssn)
WHERE mem.ssn IN
        (SELECT s.ssn
         FROM (message AS me JOIN student ON me.receiver_ssn= s.ssn)
         WHERE me.sender_ssn IN
         (SELECT employee_ssn
                FROM has_exp_on
                WHERE roles = 'Computer Engineer'))
ORDER BY cpga
;
```

Lists the first name and last name of students who speak Italian or English with an average grade of 2.5 and started university after 2000.

```
SELECT DISTINCT mem.fname, mem.lname
FROM member_language AS ml, student AS s, members AS mem
WHERE (ml.languages='Italian' OR  ml.languages='English') AND s.cpga > 2.5 AND
s.Start_year > 2000
        AND ml.member_ssn = s.ssn AND mem.ssn=ml.member_ssn
;
```

List the skill name,first name,last name of members who endorse ssn number '251936178's skills.

```
SELECT s.skill_name,fname,lname
FROM members,skill as s
WHERE (members.ssn,s.skill_id)  IN (SELECT members.ssn ,skill_id
                                   FROM member_endorse_skill as mes,members
                                   WHERE member_ssn ='251936178' and
                                           endorsed_by=members.ssn)
;
```

It lists how many people have endorsed their SKILLs for users who have a CPGA higher than the average GPA and who is female

```
SELECT mes.member_ssn, COUNT(*)
FROM member_endorse_skill AS mes
WHERE mes.member_ssn IN
   (SELECT mem.ssn
    FROM (student AS s JOIN members AS mem ON s.ssn=mem.ssn)
    WHERE s.cpga > (SELECT AVG(s.cpga)
            FROM student AS s) AND mem.sex='F')
GROUP BY mes.member_ssn;
```