Database Management 2021-2022 Midterm Project: LinkedinMoodle

05180000070 Doğukan Argüç

05180000108 Kılıçarslan Söyler

05190000110 Mustafa Ege Alanya

05190000113 Orkun Bilbaşar

Content

ANALYSIS

1.

LinkedIn: LinkedIn is the world's largest professional network on the internet. You can use LinkedIn tofind the right job or internship, connect and strengthen professional relationships, and learn the skills you need to succeed in your career. You can access LinkedIn from a desktop, LinkedIn mobile app, mobile web experience, or the LinkedIn Lite Android mobile app. A complete LinkedIn profile can help you connect with opportunities by showcasing your
unique professional story through experience, skills, and education. You can also use LinkedIn to organize offline events, join groups, write articles, post photos and videos, and more.

Moodle: Moodle stands for "Modular Object Oriented Dynamic Learning Environment" and the application is a free and open source learning management system providing a platform for e-learning and it helps the various educators considerably in conceptualizing the various courses, course structures and curriculum thus facilitating interaction with online students.

2.

2a)

LinkedIn: The mission of LinkedIn is connecting the world's professionals to make them more productive and successful

Moodle: Moodle is a learning platform designed to provide educators, administrators and learners with a single robust, secure and integrated system to create personalized learning environments.

2b)

LinkedIn: Member, address, connection, group, cv, profile, organization, message, event.

Moodle: Student, course, subject, assignment, message, course_enrolment, event, forum, teacher.

2c)

LinkedIn: Member entity has attributes "username, id, mail, password, joined date". Address entity has attributes "country, city". Profile entity has attributes "name, gender, birth date". Group entity has attributes "group id, group name, group description, creation date". Cv entity has attributes "cv id, creation and uptade date, cv section".Organization entity has attributes "org id, manager user id, web site, slogan, location, type". Message has attributes “sender and receiver user id”. Event has attributes “event id,event name, description, created date, event start date”.

Moodle: Student has attributes “student no, student email, reg date”. Teacher has attributes “teacher email”. Course has attributes “course name, id, semester, year”. Subject has attributes “texts, subject no, date”. Message has attributes “receiver and sender id, send date, content, message id”. Enrollment has attributes “student and course id”. Forum has attributes “course id”. Event has attributes “event id,event name, description, created date, event start date”.

2d) LinkedIn:

User creates organization.

User manages organization.

User works for organization.

User follows organization.

User has cv.

User joins to events.

User creates events.

User connects to user.

User follows user.

User messages to user.

User has profile.

User applies to job adv.

User joins group.

User likes post.

Group has group posts.

Organization comments to post.

Organization likes post.

Organization creates events.

Moodle:

Student has department.

Student enrolls to course.

Student posts to forum.

Student comments to forum.

Student submits assignment.

Teacher teaches course.

Teacher creates assignment.

Teacher posts to forum.

Teacher comments to forum.

Course has subject.

Course has forum.

Subject has assignment.

University admins department.

Department offers course.

2e)

LinkedIn:

A user manages only one organization.

A user can’t post to group he didn’t join.

A user can’t like post in grup he didn’t join.

Cv can’t be created without user.

Cv section can’t be created without cv.

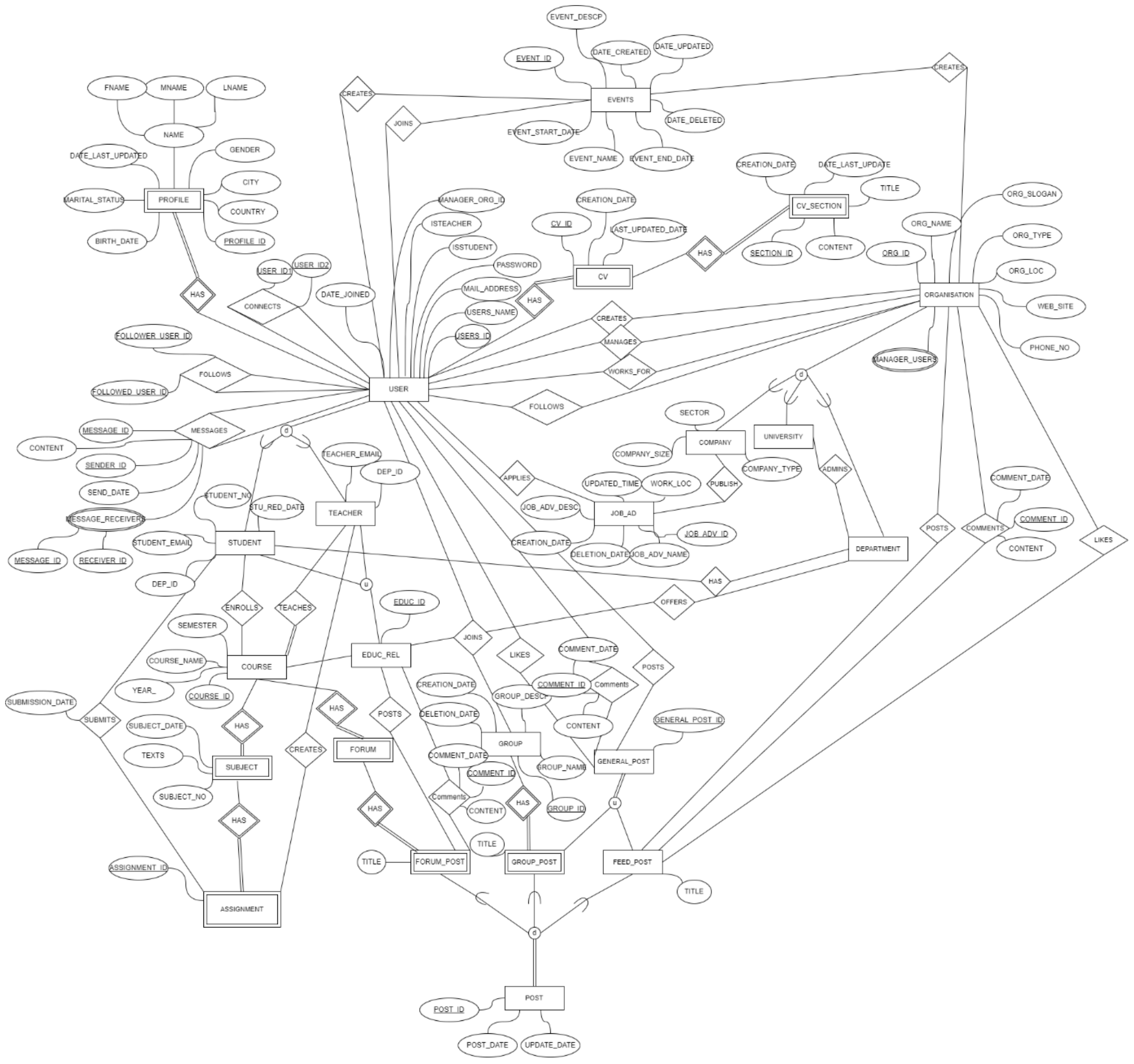
Moodle:

If the student is not in that department, he can’t enroll the course of that department.

If the student is not taking that course, he can’t submit an assignment.

Teacher can not teaches course if he is not working its department.

DESIGN-CONCEPTUAL DESIGN



DESIGN-LOGICAL MODEL

5.

USER(<u>USER_ID</u>,USER_NAME,MAIL_ADDRESS,PASSWORD,ISSTUDENT,ISTEACHER,PROFILE_ID,CV_ID,DATE_JOINED,MANAGER_ORG_ID)

USER_TYPE 1 İSE TEACHER, 0 İSE STUDENT, NULL İSE HER İKİSİ DE DEĞİL.

STUDENT(<u>USER_ID</u>,STUDENT_NO,STUDENT_EMAIL,DEP_ID,STU_REG_DATE)

TEACHER(<u>USER_ID</u>,TEACHER_EMAIL,DEP_ID)

PROFILE(USER_ID, PROFILE_ID, CURRENT_ORGANIZATION_ID, FNAME, MNAME, LNAME, COUNTRY, CITY, GENDER, DATE_LAST_UPDATED, MARITAL_STATUS, BIRTH_DATE)

CV(USER_ID, CV_ID, CREATION_DATE, LAST_UPDATED_DATE)

CV_SECTION(CV_ID,SECTION_ID,CREATION_DATE,DATE_LAST_UPDATED,CV_SECTION_TITLE, CV_SECTION_CONTENT)

ORGANIZATION(ORGANIZATION_ID,SUPER_MANAGER_USER, MANAGER_USERS, OGRANIZATION_NAME,WEB_SITE,ORGANIZATION_SLOGAN,PHONE_NO,LOCATION, ORG_TYPE)

ORG_TYPE 0 İSE COMPANY, 1 İSE UNIVERSITY, 2 İSE DEPARTMENT.

COMPANY(ORGANIZATION_ID,SECTOR,COMPANY_SIZE,COMPANY_TYPE)

UNIVERSITY (ORGANIZATION_ID,)

DEPARTMENT(ORGANIZATION_ID,SUPER_MANAGER_USER, MANAGER_USERS, OGRANIZATION_NAME,WEB_SITE,ORGANIZATION_SLOGAN,PHONE_NO,LOCATION,ADMIN_UNI)

EVENT(EVENT_ID,CREATOR_USER_ID,EVENT_NAME,EVENT_DESCRIPTION,EVENT_START_DATE,EVENT_END_DATE,DATE_CREATED,TIME_CREATED,TIME_CREATED,DATE_UPDATED,TIME_UPDATED,DATE_DELETED,CREATOR_ORG_ID,CREATOR_TYPE)

JOB_AD(JOB_ADV_ID,ADVERTISING_ORGANIZATION_ID,ADV_CREATOR_USER_ID,APPLICANTS,JOB_ADV_NAME,JOB_ADV_DESCRIPTION,WORK_LOCATION,CREATION_DATE,CREATION_TIME,DELETION_DATE,DELETION_TIME,UPDATED_DATE,UPDATED_TIME)

COURSE(COURSE_ID,COURSE_NAME,SEMESTER,YEAR,DEP_ID)

GROUP(GROUP_ID,CREATOR_USER_ID,GROUP_NAME,GROUP_DESCRIPTION,CREATION_DATE,DELETION_DATE)

SUBJECT(COURSE_ID, SUBJECT_NO, SUBJECT_DATE, TEXT)

FORUM(COURSE_ID)

GROUP_POST(POST_ID,USER_ID,GROUP_ID,POST_DATE,POST_TIME,UPTADE_DATE,UPDATE_TIME,TITLE,_GENERAL_POST_ID)

FORUM_POST(POST_ID,USER_ID,COURSE_ID,POST_DATE,POST_TIME,UPTADE_DATE,UPDATE_TIME,TITLE)

FEED_POST(POST_ID,USER_ID,ORG_ID,POST_DATE,POST_TIME,UPTADE_DATE,UPDATE_TIME,TITLE, GENERAL_POST_ID)

GENERAL_POST(GENERAL_POST_ID,USER_ID)

EDUC_REL(EDUC_ID)

ASSIGNMENT(COURSE_ID,SUBJECT_NO,ASSIGNMENT_ID,DEADLINE_DATE,DEADLINE_TIME, CREATOR_TEACHER_ID)

MANAGER_USERS(ORGANIZATION_ID, MANAGER_USER_IDS)

MESSAGE_RECEIVERS(MESSAGE_ID, SENDER_ID, RECEIVER_ID)

GENERAL_COMMENTS(USER_ID,GENERAL_POST_ID,COMMENT_ID,COMMENT_DATE,COMMENT_TIME,CONTENT)

FEED_POST_COMMENTS(USER_ID,FEED_POST_ID,COMMENT_ID,COMMENT_DATE,COMMENT_TIME,CONTENT)

GENERAL_LIKES(USER_ID,POST_ID)

FEED_POST_LIKES(USER_ID,POST_ID)

FORUM_COMMENTS(EDUC_ID,FORUM_POST_ID,COMMENT_ID,COMMENT_DATE,COMMENT_TIME,CONTENT)

SUBMITS(STUDENT_ID,ASSIGNMENT_ID,SUBMISSION_DATE,SUBMISSION_TIME)

WORKS_FOR(USER_ID, ORG_ID)

MESSAGES(MESSAGE_ID, SENDER_ID, RECEIVER_ID, SEND_DATE, SEND_TIME, CONTENT)

CONNECTS(USER_ID1, USER_ID2)

FOLLOWS_USER(FOLLOWER_USER_ID, FOLLOWED_USER_ID)

FOLLOWS_ORG(FOLLOWER_USER_ID, FOLLOWED_ORGANIZATION_ID)
JOINS_EVENT(EVENT_ID, USER_ID)

JOINS_GROUP(GROUP_ID, USER_ID)

APPLIES(USER_ID, JOB_ADV_ID)

ENROLLS(STUDENT_ID, COURSE_ID)

TEACHES(TEACHER_ID, COURSE_ID)

IMPLEMENTATION-PHYSICAL MODEL

6. Database: microsotf sql

```
CREATE TABLE ORGANIZATION(

ORGANIZATION_ID INT IDENTITY(1,1) PRIMARY KEY,

SUPER_MANAGER_USER_ID INT NOT NULL,

ORGANIZATION_NAME VARCHAR(40) NOT NULL,

WEB_SITE VARCHAR(100),

ORGANIZATION_SLOGAN VARCHAR(60),

PHONE_NO VARCHAR(12) NOT NULL,

ORG_LOCATION VARCHAR(100) NOT NULL,

ORG_TYPE TINYINT NOT NULL

)


CREATE TABLE COMPANY(

ORGANIZATION_ID INT PRIMARY KEY FOREIGN KEY REFERENCES
ORGANIZATION(ORGANIZATION_ID),

SECTOR VARCHAR(40),

COMPANY_SIZE VARCHAR(20),

COMPANY_TYPE VARCHAR(30)

)


CREATE TABLE UNIVERSITY(

ORGANIZATION_ID INT PRIMARY KEY FOREIGN KEY REFERENCES ORGANIZATION(ORGANIZATION_ID)

)


CREATE TABLE DEPARTMENT(

ORGANIZATION_ID INT PRIMARY KEY FOREIGN KEY REFERENCES
ORGANIZATION(ORGANIZATION_ID),

ADMIN_UNI_ID INT FOREIGN KEY REFERENCES UNIVERSITY(ORGANIZATION_ID) NOT NULL

)
```

```
CREATE TABLE USERS
(USERS_ID   INT IDENTITY(1,1)    NOT NULL,
USERS_NAME  VARCHAR(20) NOT NULL,
MAIL_ADDRESS  VARCHAR(30) NOT NULL,
PASSWORD      VARCHAR(22) NOT NULL,
ISSTUDENT    BIT     NOT NULL DEFAULT 0,
ISTEACHER    BIT     NOT NULL DEFAULT 0,
DATE_JOINED  DATE,
MANAGER_ORG_ID INT,
PRIMARY KEY(USERS_ID),
FOREIGN KEY(MANAGER_ORG_ID) REFERENCES ORGANIZATION(ORGANIZATION_ID) ,
UNIQUE (USERS_NAME),
UNIQUE(MAIL_ADDRESS)

)

ALTER TABLE ORGANIZATION
ADD FOREIGN KEY (SUPER_MANAGER_USER_ID) REFERENCES USERS(USERS_ID)


CREATE UNIQUE INDEX USERNAME ON USERS (USERS_NAME)

CREATE TABLE STUDENT(
USERS_ID INT PRIMARY KEY FOREIGN KEY REFERENCES USERS(USERS_ID),
STUDENT_NO INT IDENTITY(1000,1) UNIQUE NOT NULL,
STUDENT_EMAIL VARCHAR(30) NOT NULL ,
DEP_ID INT FOREIGN KEY REFERENCES DEPARTMENT(ORGANIZATION_ID),
STU_REG_DATE DATE NOT NULL,
)
```

```
CREATE TABLE TEACHER(
USERS_ID INT PRIMARY KEY FOREIGN KEY REFERENCES USERS(USERS_ID),
TEACHER_EMAIL VARCHAR(30) NOT NULL ,
DEP_ID INT FOREIGN KEY REFERENCES DEPARTMENT(ORGANIZATION_ID)
)



CREATE TABLE PROFILES(
USERS_ID INT  FOREIGN KEY REFERENCES USERS(USERS_ID),
PROFILE_ID INT IDENTITY(1,1),
CURRENT_ORGANIZATION_ID INT FOREIGN KEY REFERENCES ORGANIZATION(ORGANIZATION_ID) ,
FNAME VARCHAR(15) NOT NULL,
MNAME VARCHAR(15),
LNAME VARCHAR(30) NOT NULL,
COUNTRY VARCHAR(40),
CITY VARCHAR(20),
GENDER BIT,
DATE_LAST_UPDATED DATE,
MARITAL_STATUS BIT,
BIRTH_DATE DATE NOT NULL,
PRIMARY KEY(USERS_ID,PROFILE_ID)
)

CREATE TABLE CV(
USERS_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID),
CV_ID INT IDENTITY(1,1) NOT NULL,
CREATION_DATE DATE,
LAST_UPDATED_DATE DATE,
PRIMARY KEY(USERS_ID,CV_ID)
)
```

```
CREATE TABLE CV_SECTION(

CV_ID INT ,

USERS_ID INT,

SECTION_ID INT IDENTITY(1,1) NOT NULL,

CREATION_DATE DATE,

DATE_LAST_UPDATED DATE,

CV_SECTION_TITLE VARCHAR(30),

CV_SECTION_CONTENT VARCHAR(1000),

PRIMARY KEY(CV_ID,SECTION_ID),

FOREIGN KEY (USERS_ID, CV_ID) REFERENCES CV(USERS_ID, CV_ID)

)
```

```
CREATE TABLE EVENT_(

EVENT_ID INT IDENTITY(1,1) PRIMARY KEY,

CREATOR_USER_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID),

EVENT_NAME VARCHAR(40) NOT NULL,

EVENT_DESCRIPTION VARCHAR(200) NOT NULL,

EVENT_START_DATE DATE NOT NULL,

EVENT_END_DATE DATE,

DATE_CREATED DATETIME NOT NULL,

DATE_UPDATED DATETIME,

DATE_DELETED DATETIME,

CREATOR_ORG_ID INT FOREIGN KEY REFERENCES ORGANIZATION(ORGANIZATION_ID),

CREATOR_TYPE VARCHAR(30)

)

CREATE TABLE JOB_AD(

JOB_ADV_ID INT IDENTITY(1,1) PRIMARY KEY,
ADVERTISING_ORGANIZATION_ID INT FOREIGN KEY REFERENCES
ORGANIZATION(ORGANIZATION_ID),

ADV_CREATOR_USER_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID),

JOB_ADV_NAME VARCHAR(100) NOT NULL,

JOB_ADV_DESCRIPTION VARCHAR(1000) NOT NULL,

WORK_LOCATION VARCHAR(100) NOT NULL,

CREATION_DATE DATETIME NOT NULL ,

DELETION_DATE DATETIME,

UPDATED_DATE DATETIME

)


CREATE TABLE COURSE(

COURSE_ID INT IDENTITY(1,1) PRIMARY KEY,

COURSE_NAME VARCHAR(20) UNIQUE,

SEMESTER INT NOT NULL,

YEAR_ INT NOT NULL,

DEP_ID INT FOREIGN KEY REFERENCES DEPARTMENT(ORGANIZATION_ID) NOT NULL

)


CREATE TABLE GROUP_ (

GROUP_ID INT IDENTITY (1,1) PRIMARY KEY,

CREATOR_USER_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID),

GROUP_NAME VARCHAR(40) NOT NULL,

GROUP_DESCRIPTION VARCHAR(1000),

CREATION_DATE DATETIME NOT NULL,

DELETION_DATE DATETIME

)


CREATE TABLE SUBJECT_ (

COURSE_ID INT IDENTITY(1,1) PRIMARY KEY,

SUBJECT_NO INT UNIQUE,
SUBJECT_DATE DATE NOT NULL,
TEXTS VARCHAR(2000)
)

CREATE TABLE FORUM(
COURSE_ID INT IDENTITY(1,1) PRIMARY KEY FOREIGN KEY REFERENCES COURSE(COURSE_ID)
)

CREATE TABLE GENERAL_POST(
GENERAL_POST_ID INT IDENTITY(1,1) PRIMARY KEY,
USERS_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID) NOT NULL
)

CREATE TABLE GROUP_POST(
POST_ID INT,
GROUP_ID INT FOREIGN KEY REFERENCES GROUP_(GROUP_ID) NOT NULL,
USERS_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID) NOT NULL,
TITLE VARCHAR(50) NOT NULL,
POST_DATE DATETIME NOT NULL,
UPDATE_DATE DATETIME ,
GENERAL_POST_ID INT FOREIGN KEY REFERENCES GENERAL_POST(GENERAL_POST_ID),
PRIMARY KEY(POST_ID,GROUP_ID)
)

CREATE TABLE FORUM_POST(
POST_ID INT,
COURSE_ID INT FOREIGN KEY REFERENCES FORUM(COURSE_ID) DEFAULT 0,
USERS_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID) NOT NULL,
TITLE VARCHAR(50) NOT NULL,
POST_DATE DATETIME NOT NULL,
UPDATE_DATE DATETIME ,
PRIMARY KEY (POST_ID,COURSE_ID)
)

ALTER TABLE FORUM_POST ADD CONSTRAINT FORUM_FORUMPOST_FK
FOREIGN KEY (COURSE_ID) REFERENCES FORUM(COURSE_ID) ON DELETE SET DEFAULT

CREATE TABLE FEED_POST(
POST_ID INT PRIMARY KEY,
USERS_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID) NOT NULL,
ORG_ID INT FOREIGN KEY REFERENCES ORGANIZATION(ORGANIZATION_ID),
TITLE VARCHAR(50) NOT NULL,
POST_DATE DATETIME NOT NULL,
UPDATE_DATE DATETIME ,
GENERAL_POST_ID INT FOREIGN KEY REFERENCES GENERAL_POST(GENERAL_POST_ID)
)


CREATE TABLE EDUC_REL
(
EDUC_ID INT PRIMARY KEY
)

CREATE TABLE ASSIGNMENT(
COURSE_ID INT FOREIGN KEY REFERENCES COURSE(COURSE_ID),
SUBJECT_NO INT FOREIGN KEY REFERENCES SUBJECT_(SUBJECT_NO),
ASSIGNMENT_ID INT IDENTITY(1,1),
DEADLINE_DATE DATETIME NOT NULL,
CREATOR_TEACHER_ID INT FOREIGN KEY REFERENCES TEACHER(USERS_ID),
PRIMARY KEY(COURSE_ID,SUBJECT_NO,ASSIGNMENT_ID)
)
```

```
CREATE TABLE MANAGER_USERS(
ORGANIZATION_ID INT FOREIGN KEY REFERENCES ORGANIZATION(ORGANIZATION_ID),
MANAGER_USER_IDS INT FOREIGN KEY REFERENCES USERS(USERS_ID)
)

CREATE TABLE MESSAGE_RECEIVERS(
MESSAGE_ID INT IDENTITY(1,1),
RECEIVER_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID)
PRIMARY KEY(MESSAGE_ID, RECEIVER_ID)
)


CREATE TABLE GENERAL_COMMENTS(
USERS_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID),
GENERAL_POST_ID INT FOREIGN KEY REFERENCES GENERAL_POST(GENERAL_POST_ID),
COMMENT_ID INT IDENTITY(1,1),
COMMENT_DATE DATETIME NOT NULL,
CONTENT VARCHAR(1000) NOT NULL,
PRIMARY KEY(USERS_ID,GENERAL_POST_ID,COMMENT_ID)
)


CREATE TABLE FEED_POST_COMMENTS(
USERS_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID) ,
FEED_POST_ID INT FOREIGN KEY REFERENCES FEED_POST(POST_ID),
COMMENT_ID INT IDENTITY(1,1),
COMMENT_DATE DATETIME NOT NULL,
CONTENT VARCHAR(1000) NOT NULL,
PRIMARY KEY(USERS_ID,FEED_POST_ID,COMMENT_ID)
)
```

```
CREATE TABLE GENERAL_LIKES(
USERS_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID) ,
GENERAL_POST_ID INT FOREIGN KEY REFERENCES GENERAL_POST(GENERAL_POST_ID),
PRIMARY KEY (USERS_ID,GENERAL_POST_ID)
)

CREATE TABLE FEED_POST_LIKES(
USERS_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID) ,
FEED_POST_ID INT FOREIGN KEY REFERENCES FEED_POST(POST_ID),
PRIMARY KEY(USERS_ID,FEED_POST_ID)
)

CREATE TABLE FORUM_COMMENTS(
EDUC_ID INT FOREIGN KEY REFERENCES EDUC_REL(EDUC_ID),
FORUM_POST_ID INT ,
COURSE_ID INT,
COMMENT_ID INT IDENTITY(1,1),
COMMENT_DATE DATETIME NOT NULL,
CONTENT VARCHAR(1000) NOT NULL,
FOREIGN KEY (FORUM_POST_ID,COURSE_ID) REFERENCES FORUM_POST(POST_ID,COURSE_ID)
)


CREATE TABLE SUBMITS(
STUDENT_ID INT FOREIGN KEY REFERENCES STUDENT(USERS_ID),
COURSE_ID INT,
SUBJECT_NO INT ,
ASSIGNMENT_ID INT,
SUBMISSION_DATE DATETIME NOT NULL,
GRADE TINYINT,
PRIMARY KEY(STUDENT_ID, COURSE_ID, SUBJECT_NO, ASSIGNMENT_ID),
FOREIGN KEY (COURSE_ID,SUBJECT_NO,ASSIGNMENT_ID) REFERENCES
ASSIGNMENT(COURSE_ID,SUBJECT_NO,ASSIGNMENT_ID)
)

CREATE TABLE WORKS_FOR(
USERS_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID),
ORG_ID INT FOREIGN KEY REFERENCES ORGANIZATION(ORGANIZATION_ID),
PRIMARY KEY (USERS_ID,ORG_ID)
)

CREATE TABLE MESSAGES(
MESSAGE_ID INT IDENTITY(1,1),
SENDER_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID),
SEND_DATE DATETIME NOT NULL,
CONTENT VARCHAR(1000) NOT NULL,
PRIMARY KEY(MESSAGE_ID,SENDER_ID)
)


CREATE TABLE CONNECTS(
USER_ID1 INT FOREIGN KEY REFERENCES USERS(USERS_ID),
USER_ID2 INT FOREIGN KEY REFERENCES USERS(USERS_ID),
PRIMARY KEY (USER_ID1, USER_ID2)
)

CREATE TABLE FOLLOWS_USER(
FOLLOWER_USER_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID),
FOLLOWED_USER_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID),
PRIMARY KEY (FOLLOWER_USER_ID, FOLLOWED_USER_ID)
```

```
)


CREATE TABLE FOLLOWS_ORG(
FOLLOWER_USER_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID),
FOLLOWED_ORGANIZATION_ID INT FOREIGN KEY REFERENCES ORGANIZATION(ORGANIZATION_ID),
PRIMARY KEY (FOLLOWER_USER_ID, FOLLOWED_ORGANIZATION_ID)
)




CREATE TABLE JOINS_EVENT(
EVENT_ID INT FOREIGN KEY REFERENCES EVENT_(EVENT_ID),
USERS_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID),
PRIMARY KEY(EVENT_ID, USERS_ID)
)


CREATE TABLE JOINS_GROUP(
GROUP_ID INT FOREIGN KEY REFERENCES GROUP_(GROUP_ID),
USERS_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID),
PRIMARY KEY(GROUP_ID, USERS_ID)
)

CREATE TABLE APPLIES(
USERS_ID INT FOREIGN KEY REFERENCES USERS(USERS_ID),
JOB_ADV_ID INT FOREIGN KEY REFERENCES JOB_AD(JOB_ADV_ID),
PRIMARY KEY(USERS_ID, JOB_ADV_ID)
)
```

```
CREATE TABLE ENROLLS(
STUDENT_ID INT FOREIGN KEY REFERENCES STUDENT(USERS_ID),
COURSE_ID INT FOREIGN KEY REFERENCES COURSE(COURSE_ID),
PRIMARY KEY (STUDENT_ID, COURSE_ID)
)

CREATE TABLE TEACHES(
TEACHER_ID INT FOREIGN KEY REFERENCES TEACHER(USERS_ID),
COURSE_ID INT FOREIGN KEY REFERENCES COURSE(COURSE_ID),
PRIMARY KEY(TEACHER_ID, COURSE_ID)
)
```

7. POPULATING

```
INSERT INTO USERS (USERS_NAME,MAIL_ADDRESS,ISSTUDENT,ISTEACHER, PASSWORD)
VALUES('EGEALANYA','EGEALANYA@GMAIL.COM',1,0, 'EGE22')

INSERT INTO USERS (USERS_NAME,MAIL_ADDRESS,ISSTUDENT,ISTEACHER, PASSWORD)
VALUES('OSMANUNALIR','MURATUNALIR@GMAIL.COM',0,1, 'OSMAN22')

INSERT INTO USERS (USERS_NAME,MAIL_ADDRESS,ISSTUDENT,ISTEACHER,PASSWORD)
VALUES('ORKUNBIL','ORKUNBILBASAR@GMAIL.COM',1,0, 'ORKUN22')

INSERT INTO USERS (USERS_NAME,MAIL_ADDRESS,ISSTUDENT,ISTEACHER,PASSWORD)
VALUES('KILICSOY','KILICSOYLER@GMAIL.COM',1,0, 'KILIC22')

INSERT INTO USERS (USERS_NAME,MAIL_ADDRESS,ISSTUDENT,ISTEACHER,PASSWORD)
VALUES('DOGUKANARGUC','DODO0101@GMAIL.COM',1,0, 'DOGUKAN22')

INSERT INTO USERS (USERS_NAME,MAIL_ADDRESS,ISSTUDENT,ISTEACHER,PASSWORD)
VALUES('SEZERCANTANISMAN','SEZERCAN@GMAIL.COM',0,1,'SEZERCAN22')

INSERT INTO USERS (USERS_NAME,MAIL_ADDRESS,ISSTUDENT,ISTEACHER, PASSWORD)
```

```
VALUES('ÖMER FARUK ÖZCAN','OFARUKOZCAN@GMAIL.COM',0,0, 'OMERFARUK35')


INSERT INTO USERS (USERS_NAME,MAIL_ADDRESS,ISSTUDENT,ISTEACHER, PASSWORD)

VALUES('KEMAL KILIÇ','KEMALKILIC@GMAIL.COM',1,0, '19992000')


INSERT INTO USERS (USERS_NAME,MAIL_ADDRESS,ISSTUDENT,ISTEACHER, PASSWORD)

VALUES('Necdet BUDAK','NECDETBUDAK@GMAIL.COM',0,1, 'EGEUNIKALP')



INSERT INTO ORGANIZATION(SUPER_MANAGER_USER_ID, ORGANIZATION_NAME, WEB_SITE,
ORGANIZATION_SLOGAN, PHONE_NO, ORG_LOCATION, ORG_TYPE)

VALUES(7,CRYSTAL LAMPS,www.crystallamps.com,CRYSTAL LAMPS FOR A BRIGHT FUTURE, 0280 599
99 23, IZMIR/TURKEY,0)


UPDATE USERS SET MANAGER_ORG_ID=1

WHERE USERS_ID=7


INSERT INTO COMPANY(ORGANIZATION_ID,SECTOR,COMPANY_SIZE,COMPANY_TYPE)

VALUES(1,'ELECTRONICS','SMALL','LTD')


INSERT INTO
JOB_AD(ADVERTISING_ORGANIZATION_ID,ADV_CREATOR_USER_ID,JOB_ADV_NAME,JOB_ADV_DES
CRIPTION,WORK_LOCATION,CREATION_DATE)

VALUES(1,7,'ELEKTRİK ELEKTRONİK MÜHENDİSİ ARANIYOR','Firmamızda çalışmak üzere 5 adet
elektrik elektronik mühendisi arıyoruz. Bu ilana başvurup bize CV'nizi
gönderebilirsiniz','İZMİR/TÜRKİYE','3.02.2022')


INSERT INTO PROFILES
(USERS_ID,FNAME,MNAME,LNAME,COUNTRY,CITY,GENDER,MARITAL_STATUS,BIRTH_DATE)
```

```
VALUES(1,'MUSTAFA','EGE','ALANYA','FRANCE','LYON',1,0,'2001-04-02')

INSERT INTO CV(USERS_ID,CREATION_DATE)
VALUES(1,'2020-08-15')

INSERT INTO CV_SECTION(CV_ID,USERS_ID,CV_SECTION_TITLE,CV_SECTION_CONTENT)
VALUES(1,1,'SERTİFİKALAR','GLOBAL BUSINESS FORUM, AI 101')

INSERT INTO STUDENT(USERS_ID,STUDENT_EMAIL,STU_REG_DATE)
VALUES(1,'1001@STUDENT.EDU','2019-08-15')



INSERT INTO PROFILES
(USERS_ID,FNAME,MNAME,LNAME,COUNTRY,CITY,GENDER,MARITAL_STATUS,BIRTH_DATE)
VALUES(2,'MURAT','OSMAN','UNALIR','BELGIUM','BRUGGE',1,0,'1971-01-01')

INSERT INTO CV(USERS_ID,CREATION_DATE)
VALUES(2,'1991-10-12')

INSERT INTO CV_SECTION(CV_ID,USERS_ID,CV_SECTION_TITLE,CV_SECTION_CONTENT)
VALUES(2,2,'YETENEKLER','CRITICAL THINKING, TIME MANAGEMENT, TEAMWORK')

INSERT INTO TEACHER(USERS_ID,TEACHER_EMAIL)
VALUES(2,'OSMANUNALIR@TEACHER.EDU')



INSERT INTO PROFILES
(USERS_ID,FNAME,LNAME,COUNTRY,CITY,GENDER,MARITAL_STATUS,BIRTH_DATE)
VALUES(3,'ORKUN','BILBASAR','CYPRUS','NICOSIA',1,0,'1999-04-13')
```

```
INSERT INTO CV(USERS_ID,CREATION_DATE)

VALUES(3,'2018-06-19')


INSERT INTO CV_SECTION(CV_ID,USERS_ID,CV_SECTION_TITLE,CV_SECTION_CONTENT)

VALUES(3,3,'SERTİFİKALAR','MACHINE LEARNING 101, ADVANCED LEVEL DATABASE MANAGEMENT')


INSERT INTO STUDENT(USERS_ID,STUDENT_EMAIL,STU_REG_DATE)

VALUES(3,'1002@STUDENT.EDU','2019-09-08')



INSERT INTO PROFILES
(USERS_ID,FNAME,LNAME,COUNTRY,CITY,GENDER,MARITAL_STATUS,BIRTH_DATE)

VALUES(4,'KILICARSLAN','SOYLER','TURKEY','BALIKESIR',1,0,'2000-08-05')


INSERT INTO CV(USERS_ID,CREATION_DATE)

VALUES(4,'2020-09-08')


INSERT INTO CV_SECTION(CV_ID,USERS_ID,CV_SECTION_TITLE,CV_SECTION_CONTENT)

VALUES(4,4,'YETENEKLER','SOFTWARE DEVELOPMENT,CRISIS MANAGEMENT')


INSERT INTO STUDENT(USERS_ID,STUDENT_EMAIL,STU_REG_DATE)

VALUES(4,'1003@STUDENT.EDU','2019-09-08')




INSERT INTO PROFILES
(USERS_ID,FNAME,LNAME,COUNTRY,CITY,GENDER,MARITAL_STATUS,BIRTH_DATE)

VALUES(5,'DOGUKAN','ARGUC','GERMANY','BERLIN',1,0,'2000-08-05')


INSERT INTO CV(USERS_ID,CREATION_DATE)

VALUES(5,'2020-08-15')
```

```
INSERT INTO CV_SECTION(CV_ID,USERS_ID,CV_SECTION_TITLE,CV_SECTION_CONTENT)

VALUES(5,5,'SERTİFİKALAR','IOT 101, AI 101')


INSERT INTO STUDENT(USERS_ID,STUDENT_EMAIL,STU_REG_DATE)

VALUES(5,'1004@STUDENT.EDU','2018-09-21')




INSERT INTO PROFILES
(USERS_ID,FNAME,LNAME,COUNTRY,CITY,GENDER,MARITAL_STATUS,BIRTH_DATE)

VALUES(6,'SEZERCAN','TANISMAN','ENGLAND','LONDON',1,0,'1992-05-10')


INSERT INTO CV(USERS_ID,CREATION_DATE)

VALUES(6,'2012-08-15')


INSERT INTO CV_SECTION(CV_ID,USERS_ID,CV_SECTION_TITLE,CV_SECTION_CONTENT)

VALUES(6,6,'SERTİFİKALAR','GLOBAL BUSINESS FORUM , ADVANCED IOT')


INSERT INTO TEACHER(USERS_ID,TEACHER_EMAIL)

VALUES(6,'SEZERCANTANISMAN@TEACHER.EDU')



INSERT INTO CV(USERS_ID,CREATION_DATE)

VALUES(8,'03.12.2021')


INSERT INTO
CV_SECTION(CV_ID,USERS_ID,CREATION_DATE,CV_SECTION_TITLE,CV_SECTION_CONTENT)

VALUES(7,8,'02.02.2022','OKULLAR','İZMİR FRANSIZ LİSESİ - EGE ÜNİVERSİTESİ ELEKTRONİK
ELEKTRONİK MÜHENDİSLİĞİ')



INSERT INTO APPLIES(USERS_ID,JOB_ADV_ID)
```

```
VALUES(8,1)

INSERT INTO ORGANIZATION(SUPER_MANAGER_USER_ID, ORGANIZATION_NAME, WEB_SITE,
PHONE_NO, ORG_LOCATION, ORG_TYPE)
VALUES(9,'EGE ÜNİVERSİTESİ','www.ege.edu.tr', '0232 311 10 10', IZMIR/TURKEY,1)

UPDATE USERS SET MANAGER_ORG_ID=2
WHERE USERS_ID=9

INSERT INTO UNIVERSITY(ORGANIZATION_ID)
VALUES(2)


INSERT INTO ORGANIZATION(SUPER_MANAGER_USER_ID, ORGANIZATION_NAME, WEB_SITE,
PHONE_NO, ORG_LOCATION, ORG_TYPE)
VALUES(2,'ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ FAKÜLTESİ','www.electronics.ege.edu.tr',
'02323111817', IZMIR/TURKEY,2)

INSERT INTO DEPARTMENT(ORGANIZATION_ID,ADMIN_UNI_ID)
VALUES(3,2)

INSERT INTO STUDENT(USERS_ID,STUDENT_EMAIL,DEP_ID,STU_REG_DATE)
VALUES(8,'1001@student.ege.edu.tr',3,'05.04.2017')

INSERT INTO COURSE(COURSE_NAME,SEMESTER,YEAR_,DEP_ID)
VALUES('DCD',4,2021,3)

UPDATE TEACHER
SET DEP_ID = 8 WHERE USERS_ID = 2

INSERT INTO TEACHES(TEACHER_ID,COURSE_ID)
VALUES(2,3)
```

```
INSERT INTO USERS(USERS_NAME, MAIL_ADDRESS, PASSWORD, ISSTUDENT, ISTEACHER,
MANAGER_ORG_ID)

VALUES('LEVENTTOKER', 'LEVENTTOKER@GMAIL.COM', 'LEVENT22', 0, 1, 8)


INSERT INTO MANAGER_USERS

VALUES(8, 10)


INSERT INTO PROFILES(USERS_ID, CURRENT_ORGANIZATION_ID, FNAME, LNAME, COUNTRY, CITY,
GENDER, MARITAL_STATUS, BIRTH_DATE)

VALUES(10, 8, 'LEVENT', 'TOKER', 'TURKEY', 'IZMIR', 1, 1, '1960-04-03')
```

8. TRIGGERS

```
CREATE TRIGGER COURSE_FORUM ON COURSE

AFTER INSERT

AS

BEGIN

        INSERT INTO FORUM(COURSE_ID)

        SELECT COURSE_ID

        FROM inserted

END


------------------------------------------

CREATE TRIGGER CV_INSERT ON CV

AFTER INSERT

AS

BEGIN

        UPDATE CV

        SET CREATION_DATE = GETDATE()

        FROM CV

        INNER JOIN Inserted i ON CV.CV_ID = i.CV_ID AND CV.USERS_ID = i.USERS_ID

END

---------------------------------------------
```

```
CREATE TRIGGER CV_UPDATE ON CV
AFTER UPDATE
AS
BEGIN
	UPDATE CV
	SET LAST_UPDATED_DATE = CURRENT_TIMESTAMP
	FROM CV
	INNER JOIN Inserted i ON CV.CV_ID = i.CV_ID AND CV.USERS_ID = i.USERS_ID
END


---------------------------------------------
CREATE TRIGGER CV_SECTION_UPDATE ON CV_SECTION
AFTER UPDATE
AS
BEGIN
	UPDATE CV_SECTION
	SET DATE_LAST_UPDATED = CURRENT_TIMESTAMP
	FROM CV_SECTION
	INNER JOIN Inserted i ON CV_SECTION.CV_ID = i.CV_ID AND CV_SECTION.USERS_ID =
i.USERS_ID AND CV_SECTION.SECTION_ID = i.SECTION_ID
END
--------------------------------------------------
CREATE TRIGGER EVENT_UPDATE ON EVENT_
AFTER UPDATE
AS
BEGIN
	UPDATE EVENT_
	SET LAST_UPDATED_DATE = CURRENT_TIMESTAMP
	FROM EVENT_
	INNER JOIN Inserted i ON EVENT.EVENT_ID = i.EVENT_ID
```

```
END

---------------------------------------------

CREATE TRIGGER PROFILE_UPDATE ON PROFILES

AFTER UPDATE

AS

BEGIN

        UPDATE PROFILES

        SET DATE_LAST_UPDATED = CURRENT_TIMESTAMP

        FROM PROFILES

        INNER JOIN Inserted i ON PROFILES.USERS_ID = i.USERS_ID AND PROFILES.PROFILE_ID =
i.PROFILE_ID

END

---------------------------------------------------

CREATE TRIGGER USER_DATE_JOINED ON USERS

AFTER INSERT

AS

BEGIN

        UPDATE USERS

        SET DATE_JOINED = GETDATE()

        FROM USERS

        INNER JOIN Inserted i ON USERS.USERS_ID = i.USERS_ID

END

------------------------------------------------------

CREATE TRIGGER CV_SECTION_INSERT ON CV_SECTION

AFTER INSERT

AS

BEGIN

        UPDATE CV_SECTION

        SET CREATION_DATE = GETDATE()

        FROM CV_SECTION

        INNER JOIN Inserted i ON CV_SECTION.CV_ID = i.CV_ID AND CV_SECTION.USERS_ID =
i.USERS_ID AND CV_SECTION.SECTION_ID = i.SECTION_ID
```

```
END

------------------------------------------------------------

CREATE TRIGGER INSERT_EDUC_REL ON STUDENT

AFTER INSERT

AS

BEGIN

          INSERT INTO EDUC_REL

          SELECT USERS_ID

          FROM inserted

END

----------------------------------------------------------

CREATE TRIGGER INSERT_EDUC_REL_TEACHER ON TEACHER

AFTER INSERT

AS

BEGIN

          INSERT INTO EDUC_REL

          SELECT USERS_ID

          FROM inserted

END
```

9. CHECK CONSTRAINTS

```
ALTER TABLE SUBMITS

ADD CONSTRAINT CHK_GRADE CHECK (GRADE < 101);

-----------------------------------------------

ALTER TABLE ASSIGNMENT

ADD CONSTRAINT CHK_DEADLINE CHECK (DEADLINE_DATE > CURRENT_TIMESTAMP);

------------------------------------------------------------------------

ALTER TABLE PROFILES

ADD CONSTRAINT CHK_BIRTHDATE CHECK (BIRTH_DATE < GETDATE());
```

ASSERTIONS

```
--- SQLSTATE '45000' unhandled user-defined exception

CREATE TRIGGER ORGANIZATION_ASSERTION ON ORGANIZATION
```

```
BEFORE INSERT

AS

FOR EACH ROW BEGIN

            IF (NEW.SUPER_MANAGER_USER_ID NOT IN (SELECT TEACHER.USERS_ID FROM TEACHER))

            THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT='HATA: EKLENMEK İSTENEN YÖNETİCİ,
EĞİTMEN OLMALIDIR.'

END

------------------------------------------------------------




CREATE TRIGGER COURSE_ASSERTION ON COURSE

BEFORE INSERT

AS

FOR EACH ROW BEGIN

            IF(NEW.DEP_ID NOT IN (SELECT COURSE.DEP_ID FROM COURSE,STUDENT WHERE
STUDENT.DEP_ID=COURSE.DEP_ID))

            THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT='HATA: ÖĞRENCİ KURSUN OLDUĞU
BÖLÜMDE KAYITLI OLMALIDIR.'

END


------------------------------------------------------------


CREATE TRIGGER GROUP_ASSERTION ON GROUP_POST

BEFORE INSERT

AS

FOR EACH ROW BEGIN

            IF(NEW.GROUP_ID NOT IN (SELECT GROUP_ID FROM JOINS_GROUP WHERE
NEW.USERS_ID=JOINS_GROUP.USERS_ID ))

            THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT='HATA: GRUPTA OLMAYAN KULLANICI
GRUP POST ATAMAZ.'

END
```

10. SELECT STATEMENTS

```
SELECT * FROM JOB_AD

SELECT * FROM UNIVERSITY

SELECT * FROM USERS

SELECT O.ORGANIZATION_NAME, J.JOB_ADV_NAME, J.JOB_ADV_DESCRIPTION
FROM ORGANIZATION O, JOB_AD J
WHERE O.ORGANIZATION_ID = J.ADVERTISING_ORGANIZATION_ID

SELECT O.ORGANIZATION_NAME, C.COURSE_NAME
FROM DEPARTMENT D, COURSE C, ORGANIZATION O
WHERE C.DEP_ID = D.ORGANIZATION_ID AND D.ORGANIZATION_ID = O.ORGANIZATION_ID

SELECT P.FNAME, P.LNAME, C.COURSE_NAME
FROM PROFILES P, COURSE C, TEACHER T, TEACHES TS
WHERE P.USERS_ID = T.USERS_ID AND T.USERS_ID = TS.TEACHER_ID AND C.COURSE_ID =
TS.COURSE_ID

SELECT P.FNAME, P.LNAME, T.USERS_ID, O.ORGANIZATION_NAME
FROM TEACHER T, ORGANIZATION O, DEPARTMENT D, UNIVERSITY U, PROFILES P
WHERE T.DEP_ID = D.ORGANIZATION_ID AND D.ADMIN_UNI_ID = U.ORGANIZATION_ID AND
U.ORGANIZATION_ID = O.ORGANIZATION_ID AND P.USERS_ID = T.USERS_ID

SELECT S.STUDENT_NO, P.FNAME, P.LNAME, C.COURSE_NAME, AVG(SB.GRADE) AS
AVERAGE_GRADES
FROM STUDENT S, PROFILES P, COURSE C, ASSIGNMENT A, SUBMITS SB
WHERE S.USERS_ID = P.USERS_ID AND C.COURSE_ID = SB.COURSE_ID AND A.ASSIGNMENT_ID =
SB.ASSIGNMENT_ID
GROUP BY S.STUDENT_NO, P.FNAME, P.LNAME, C.COURSE_NAME
ORDER BY AVERAGE_GRADES DESC
```

```
SELECT P.FNAME, P.LNAME, P.CITY, E.EVENT_NAME

FROM PROFILES P, JOINS_EVENT J, EVENT_ E

WHERE J.USERS_ID = P.USERS_ID AND J.EVENT_ID = E.EVENT_ID


SELECT U.USERS_ID, S.STUDENT_NO, O.ORGANIZATION_NAME

FROM USERS U, STUDENT S, ORGANIZATION O

WHERE U.USERS_ID = S.USERS_ID AND S.DEP_ID = O.ORGANIZATION_ID


SELECT P.FNAME, P.LNAME, P.COUNTRY, P.CITY, J.JOB_ADV_ID, J.JOB_ADV_NAME,
CVS.CV_SECTION_TITLE, CVS.CV_SECTION_CONTENT

FROM PROFILES P, JOB_AD J, CV_SECTION CVS, CV C, APPLIES A

WHERE P.USERS_ID = A.USERS_ID AND J.JOB_ADV_ID = A.JOB_ADV_ID AND A.USERS_ID =
C.USERS_ID AND C.USERS_ID = CVS.USERS_ID AND C.CV_ID = CVS.CV_ID


SELECT O.ORGANIZATION_NAME, FP.TITLE, COUNT(*) LIKES

FROM ORGANIZATION O, FEED_POST FP, FEED_POST_LIKES FPL

WHERE FP.ORG_ID = O.ORGANIZATION_ID AND FPL.FEED_POST_ID = FP.POST_ID

GROUP BY O.ORGANIZATION_NAME, FP.TITLE


SELECT O.ORGANIZATION_NAME AS UNIVERSITY, OD.ORGANIZATION_NAME AS DEPARTMENT

FROM ORGANIZATION O, DEPARTMENT D, ORGANIZATION OD

WHERE D.ADMIN_UNI_ID = O.ORGANIZATION_ID AND D.ORGANIZATION_ID =
OD.ORGANIZATION_ID


SELECT P.FNAME AS CHAIR_NAME, P.LNAME AS CHAIR_LAST_NAME, O.ORGANIZATION_NAME AS
DEPARTMENT_NAME, COUNT(*) AS NUMBER_OF_STUDENTS

FROM PROFILES P, ORGANIZATION O, DEPARTMENT D, STUDENT S

WHERE D.ORGANIZATION_ID = O.ORGANIZATION_ID AND O.SUPER_MANAGER_USER_ID =
P.USERS_ID AND S.DEP_ID = D.ORGANIZATION_ID

GROUP BY P.FNAME, P.LNAME, O.ORGANIZATION_NAME


SELECT S.STUDENT_NO, P.FNAME, P.LNAME, C.COURSE_NAME, MAX(SB.GRADE) AS MAX_GRADE

FROM PROFILES P, COURSE C, SUBMITS SB, ASSIGNMENT A, STUDENT S
```

```
WHERE P.USERS_ID = SB.STUDENT_ID AND C.COURSE_ID = SB.COURSE_ID AND A.ASSIGNMENT_ID =
SB.ASSIGNMENT_ID AND S.USERS_ID = SB.STUDENT_ID

GROUP BY S.STUDENT_NO, P.FNAME, P.LNAME, C.COURSE_NAME
```