Database Management
2021-2022 SPRING
Ege University Computer Engineering Department
Term Project

Mehmet Ekşi 05180000102

Görkem Özalan 05190000055

Analysis

Scope of the applications

LinkedIn

LinkedIn is a social career platform where people can share their own career profiles and can also follow the other professionals on their area of expertise. Of course, LinkedIn isn't limiting the users to just their own profession. One can make connections with people from all around the world along with any kind of career field. It can also enable people to interact with companies or HR managers and create job opportunities for themselves.

Moodle

It is a place for educators and students to create their own customizable learning spaces. A learning platform that can operate all educators, including their teaching pages, in an integrated way. It provides ease of access for students and instructors. It is widely used in University and Colleges.

Analysis Report of the applications

2.0 – Aim

Linkedin

-Bringing employees and companies together and providing convenience for both of the parties. Users are able to create their profiles with all of their accomplishments and share it with the public.

-Enabling people to interact with each other and facilitate following the work of successful people and professionals in their field thus being a source of motivation.

Moodle

-Bringing educators and students together to provide easy access to information.

-Gathering desired courses for each student in one place.

-Creating a flexible area for learning and teaching with useful tools.

-Making interactions between both students among themselves and also students with instructors easier with with sections like surveys and forums

2.1 – Main Entities

Linkedin

User, Group, Company, Post, Event, User-Profile, Education

Moodle

Student, Instructor, Courses, Survey, Forums, Files

2.2 – Entity Characteristics

Linkedin

User: user-id, mail, password, Name, Surname

Group: group-id, group-name, rules, locations (multi value)

Company: company-id, company-name, ceo, industry, locations (multi value)

Post: post-id, url, content, like, comment

Event: event-id - type - event-name - timezone - start date - end date

User-Profile: profile-id, job, photo, skills (multi value), interests (multi value), about, activity, education

Message: message-id, sender-id, receiver-id, content, sendtime

Connection: connection-id, sender-id, receiver-id, connectiontime

Moodle

Student: student_id, name, surname, mail, password, city, country

Instructor: instructor_id, name, surname, mail, password, city, country, department

Courses: course_id, course_name, semester, instructor, course_description, course_content

Survey: survey_id, survey_name, question, answers (multi value), result

Forums: forum_id, forum_name, topic, subscribes (multi value)

Files: file_id, file_name, type, uploader_id (studentFile,instructorFile)

2.3 – Relationships among entities

Linkedin

User (1 to many) shares post.

User (1 to many) creates group.

User (many to many) joins group.

User (many to many) works at company.

User (1 to many) create company.

User (1 to many) creates event.

User (many to many) joins event.

Group (1 to many) shares post.

Group (1 to many) creates event.

Group (many to many) joins event.

Company (1 to many) shares post.

Company (1 to many) creates event.

Company (many to many) joins event.

User (1 to 1) has a user profile.

User (1 to many) sends message

User (1 to many) receives message

User (many to many) connects with people.

Moodle

Instructor (1 to many) teaches courses

Instructor (1 to many) creates forum

Student (many to many) joins forum

Instructor (1 to many) creates choices

Student (many to many) joins choices

Student (many to many) registers courses

Instructor (1 to many) creates instructorFile

Instructor (1 to many) grades studentFile

Student (1 to many) uploads studentFile

2.4 – Constraints

Linkedin

The company must have one manager.

Users has to have a user profile.

Username cannot contain anything other than letters.

Group cannot be created without a user or company.

A post can have a maximum length of 3000 characters.

Event start date cannot be set to a time that has already passed.

Event name cannot exceed 75 characters.

Post must be created by either 1 user, 1 group or 1 company.

Moodle

The course must have an Instructor.

Studentfile has to be uploaded by only one student (in order to prevent cheating)

Student and instructor must use a valid email.

Student password must contain a minimum of 6 characters, at least one number and at least one letter.

The semester has to be spring or fall.

Grade must be between 0 and 100

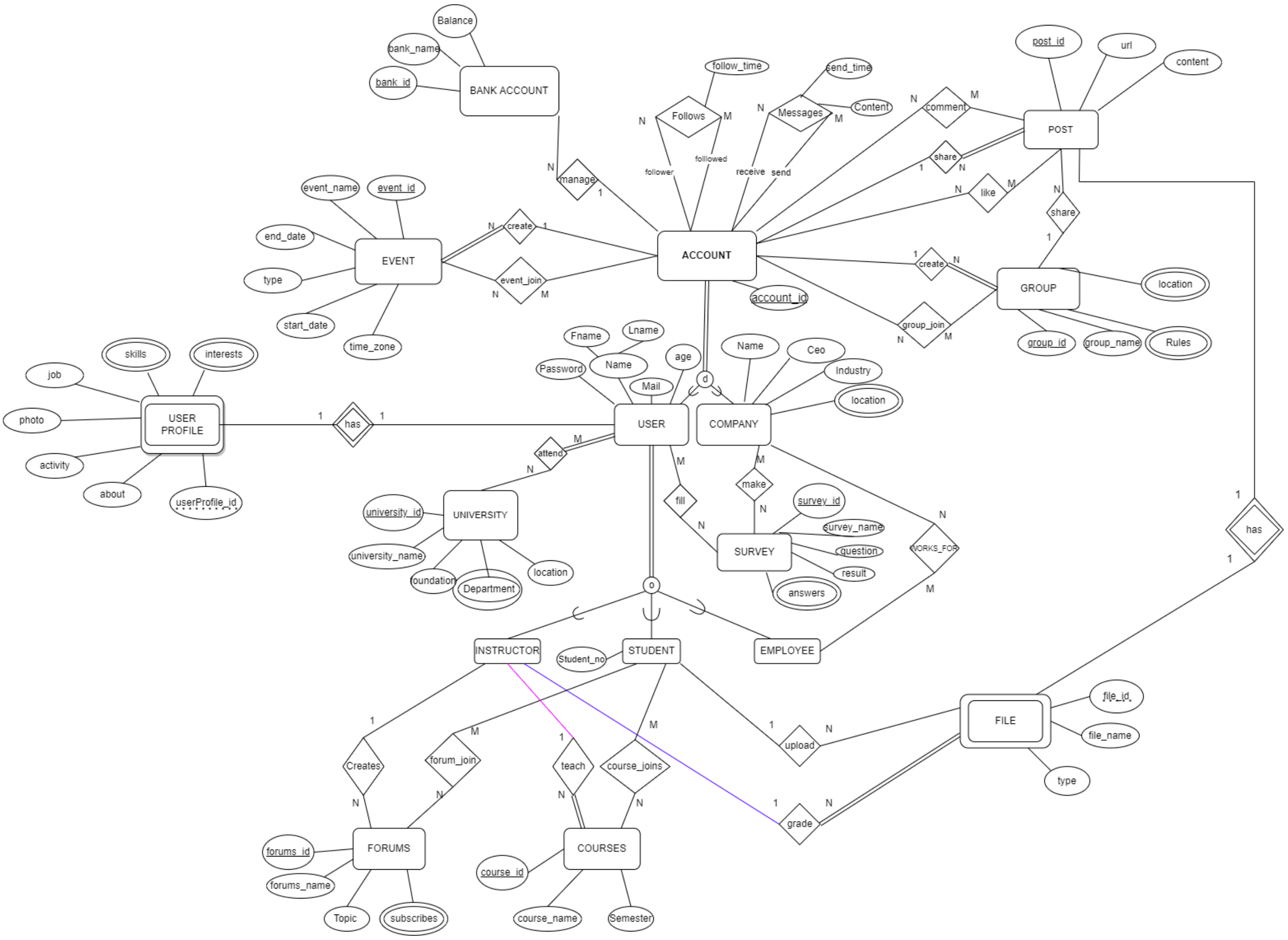
File has to be uploaded in pdf or word document

Instructor must teach at least one courses

StudentFile must be graded by Instructor

Design-Conceptual Design

Virtually integrated ER



Design-Logical Model

Iteration 1

STEP 1

BANK_ACCOUNT (bank_id , bank_name , balance)

EVENT (event_id , event_name , start_date , end_date, type , time_zone)

POST (post_id , url , content)

GROUP (group_id , group_name)

UNIVERSITY (university_id , university_name , foundation , location)

SURVEY (survey_id , survey_name , question , result)

FORUMS (forums_id , forums_name , topic)

COURSES (course_id , course_name , semester)

STEP 2

FILE (post_id , file_id , file_name , type)

STEP 3

-

STEP 4

POST (post_id , url , content, group_id) // share group-post

STEP 5

-

STEP 6

GROUP_LOCATION (group_id , glocation)

RULES (group_id , grules)

ANSWERS (survey_id , sanswers)

SUBSCRIBES (forums_id , ssubscribes)

DEPARTMENT(university_id, udepartment)

STEP 7

-

STEP 8 (8A because we need account as an entity in our database model otherwise 8B is also a good implementation since this is a mandatory disjoint relationship)

ACCOUNT (account_id)

USER (user_id, password , Fname , Lname , mail, age)

COMPANY (company_id , company_name , ceo , ındusty)

STEP 9

-

Iteration 2

STEP 1

-

STEP 2

USER_PROFILE (userProfile_id , user_id , about , activity, photo, job)

STEP 3

-

STEP 4

BANK_ACCOUNT (bank_id , bank_name ,balance , account_id) //manage bankaccount-account

EVENT (event_id , event_name , start_date , end_date, type , time_zone , account_id)   //create event-account

POST (post_id , url , content , group_id, account_id) //share post-account

GROUP (group_id , group_name , account_id) //create group-account

STEP 5

FOLLOWS (followed_account_id , follower_account_id , follow_time)

MESSAGES (receive_account_id , send_account_id , send_time , content)

COMMENT (post_id , account_id)

LIKE (post_id , account_id)

GROUP_JOIN (group_id , account_id)

EVENT_JOIN (event_id , account_id)

ATTEND (university_id , user_id)

FILL(survey_id, user_id)

MAKE (survey_id , user_id)

STEP 6

SKILLS (userProfile_id , user_id , uskills)

INTERESTS (userProfile_id , user_id , uinterests)

COMP_LOCATION (company_id , clocation)

STEP 7

-

STEP 8 (8D BECAUSE OVERLAP)

USER (user_id , password , Fname , Lname , mail ,age, Iflag , Sflag, student_no , Eflag)

STEP 9

İTERASYON 3

STEP 1

-

STEP 2

-

STEP 3

-

STEP 4

FORUMS (forums_id , forums_name , topic , instructor_id)

COURSES (course_id , course_name , semester , instructor_id)   //teach

FILE (post_id , file_id , file_name , type , student_id) //upload

FILE (post_id , file_id , file_name , type , student_id , instructor_id) //grade

STEP 5

WORKS_FOR (company_id , employee_id)

FORUM_JOIN (forums_id , student_id)

COURSE_JOINS (course_id , student_id)

STEP 6

-

STEP 7

-

STEP 8

-

STEP 9

Implementation-Physical Model

SQL Scripts

```
CREATE TABLE `account` (
  `idaccount` int NOT NULL,
  PRIMARY KEY (`idaccount`),
  UNIQUE KEY `idaccount_UNIQUE` (`idaccount`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `answers` (
  `idsurvey` int NOT NULL,
  `sanswers` varchar(45) NOT NULL,
  PRIMARY KEY (`idsurvey`,`sanswers`),
  CONSTRAINT `idsurveyanswers` FOREIGN KEY (`idsurvey`) REFERENCES `survey` (`idsurvey`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `attend` (
  `iduniversity` int NOT NULL,
  `iduser` int NOT NULL,
  PRIMARY KEY (`iduniversity`,`iduser`),
  KEY `iduserattend_idx` (`iduser`),
  CONSTRAINT `iduniversityattend` FOREIGN KEY (`iduniversity`) REFERENCES `university` (`iduniversity`),
  CONSTRAINT `iduserattend` FOREIGN KEY (`iduser`) REFERENCES `user` (`iduser`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `bank_account` (
  `idbank_account` int NOT NULL,
  `balance` int NOT NULL,
  `idaccount` int NOT NULL,
  `bank_name` varchar(45) NOT NULL,
  PRIMARY KEY (`idbank_account`),
  KEY `idaccountbank_account_idx` (`idaccount`),
  CONSTRAINT `idaccountbank_account` FOREIGN KEY (`idaccount`) REFERENCES `account` (`idaccount`),
  CONSTRAINT `balance` CHECK ((`balance` >= 0))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `comment` (
  `idpost` int NOT NULL,
  `idaccount` int NOT NULL,
  PRIMARY KEY (`idpost`,`idaccount`),
  KEY `idaccountcomment_idx` (`idaccount`),
  CONSTRAINT `idaccountcomment` FOREIGN KEY (`idaccount`) REFERENCES `account` (`idaccount`),
  CONSTRAINT `idpostcomment` FOREIGN KEY (`idpost`) REFERENCES `post` (`idpost`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `comp_location` (
  `idcompany` int NOT NULL,
  `clocation` varchar(100) NOT NULL,
  PRIMARY KEY (`idcompany`,`clocation`),
  CONSTRAINT `idaccountcomp_location` FOREIGN KEY (`idcompany`) REFERENCES `account` (`idaccount`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `company` (
  `idcompany` int NOT NULL,
  `company_name` varchar(45) NOT NULL,
  `ceo` varchar(45) NOT NULL,
  `industry` varchar(45) NOT NULL,
  PRIMARY KEY (`idcompany`),
  CONSTRAINT `idcompany` FOREIGN KEY (`idcompany`) REFERENCES `account` (`idaccount`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `course` (
  `idcourse` int NOT NULL,
  `course_name` varchar(45) NOT NULL,
  `semester` varchar(45) NOT NULL,
  `idinstructor` int NOT NULL,
  PRIMARY KEY (`idcourse`),
  KEY `idinstructorcourse_idx` (`idinstructor`),
  CONSTRAINT `idinstructorcourse` FOREIGN KEY (`idinstructor`) REFERENCES `user` (`iduser`),
  CONSTRAINT `semester` CHECK ((`semester` in (_utf8mb4'fall',_utf8mb4'FALL',_utf8mb4'SPRING',_utf8mb4'spring')))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `course_join` (
  `idcourse` int NOT NULL,
  `idstudent` int NOT NULL,
  PRIMARY KEY (`idcourse`,`idstudent`),
  KEY `idusercourse_join_idx` (`idstudent`),
  CONSTRAINT `idcourse_join` FOREIGN KEY (`idcourse`) REFERENCES `course` (`idcourse`),
  CONSTRAINT `idstudentcourse_join` FOREIGN KEY (`idstudent`) REFERENCES `user` (`iduser`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

```
CREATE TABLE `department` (
  `iduniversity` int NOT NULL,
  `udepartment` varchar(45) NOT NULL,
  PRIMARY KEY (`iduniversity`,`udepartment`),
  CONSTRAINT `iduniversitydepartment` FOREIGN KEY (`iduniversity`) REFERENCES `university` (`iduniversity`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `event` (
  `idevent` int NOT NULL,
  `event_name` varchar(45) NOT NULL,
  `start_date` date NOT NULL,
  `end_date` date NOT NULL,
  `time_zone` varchar(45) NOT NULL,
  `idaccount` int DEFAULT NULL,
  `type` varchar(45) NOT NULL,
  PRIMARY KEY (`idevent`),
  KEY `idaccountevent_idx` (`idaccount`),
  CONSTRAINT `idaccountevent` FOREIGN KEY (`idaccount`) REFERENCES `account` (`idaccount`),
  CONSTRAINT `time_zone` CHECK ((`time_zone` in (_utf8mb4'UTC',_utf8mb4'utc',_utf8mb4'GMT',_utf8mb4'gmt')))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `event_join` (
  `idevent` int NOT NULL,
  `idaccount` int NOT NULL,
  PRIMARY KEY (`idevent`,`idaccount`),
  KEY `ideventaccount_idx` (`idaccount`),
  CONSTRAINT `ideventaccount` FOREIGN KEY (`idaccount`) REFERENCES `account` (`idaccount`),
  CONSTRAINT `ideventjoin` FOREIGN KEY (`idevent`) REFERENCES `event` (`idevent`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci


CREATE TABLE `file` (
  `idfile` int NOT NULL,
  `idpost` int NOT NULL,
  `type` varchar(45) DEFAULT NULL,
  `idstudent` int NOT NULL,
  `idinstructor` int NOT NULL,
  PRIMARY KEY (`idfile`,`idpost`),
  KEY `idpost_idx` (`idpost`),
  KEY `idstudent_idx` (`idstudent`),
  KEY `idinstructorfile_idx` (`idinstructor`),
  CONSTRAINT `idinstructorfile` FOREIGN KEY (`idinstructor`) REFERENCES `user` (`iduser`),
  CONSTRAINT `idpost` FOREIGN KEY (`idpost`) REFERENCES `post` (`idpost`),
  CONSTRAINT `idstudentfile` FOREIGN KEY (`idstudent`) REFERENCES `user` (`iduser`),
  CONSTRAINT `type` CHECK ((`type` in (_utf8mb4'word',_utf8mb4'pdf')))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci


CREATE TABLE `fill` (
  `idsurvey` int NOT NULL,
  `idaccount` int NOT NULL,
  PRIMARY KEY (`idsurvey`,`idaccount`),
  KEY `iduserfill_idx` (`idaccount`),
  CONSTRAINT `idsurveyfill` FOREIGN KEY (`idsurvey`) REFERENCES `survey` (`idsurvey`),
  CONSTRAINT `iduserfill` FOREIGN KEY (`idaccount`) REFERENCES `user` (`iduser`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci


CREATE TABLE `follows` (
  `idfollowedaccount` int NOT NULL,
  `idfolloweraccount` int NOT NULL,
  `follow_time` date NOT NULL,
  PRIMARY KEY (`idfollowedaccount`,`idfolloweraccount`),
  KEY `idfolloweraccount_idx` (`idfolloweraccount`),
  CONSTRAINT `idfollowedaccount` FOREIGN KEY (`idfollowedaccount`) REFERENCES `account` (`idaccount`),
  CONSTRAINT `idfolloweraccount` FOREIGN KEY (`idfolloweraccount`) REFERENCES `account` (`idaccount`),
  CONSTRAINT `follewer` CHECK ((`idfollowedaccount` <> `idfolloweraccount`))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci


CREATE TABLE `forum` (
  `idforum` int NOT NULL,
  `forums_name` varchar(45) NOT NULL,
  `topic` varchar(45) NOT NULL,
  `idinstructer` int DEFAULT NULL,
  PRIMARY KEY (`idforum`),
  KEY `idaccountforum_idx` (`idinstructer`),
  CONSTRAINT `idaccountforum` FOREIGN KEY (`idinstructer`) REFERENCES `account` (`idaccount`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci


CREATE TABLE `forum_join` (
  `idforum` int NOT NULL,
  `idstudent` int NOT NULL,
  PRIMARY KEY (`idforum`,`idstudent`),
  KEY `idstudentforum_join_idx` (`idstudent`),
  CONSTRAINT `idforum_join` FOREIGN KEY (`idforum`) REFERENCES `forum` (`idforum`),
  CONSTRAINT `idstudentforum_join` FOREIGN KEY (`idstudent`) REFERENCES `user` (`iduser`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci


CREATE TABLE `group` (
  `idgroup` int NOT NULL,
  `group_name` varchar(45) NOT NULL,
  `idaccount` int DEFAULT NULL,
  PRIMARY KEY (`idgroup`),
  KEY `idaccountgroup_idx` (`idaccount`),
  CONSTRAINT `idaccountgroup` FOREIGN KEY (`idaccount`) REFERENCES `account` (`idaccount`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci


CREATE TABLE `group_join` (
  `idgroup` int NOT NULL,
  `idaccount` int NOT NULL,
  PRIMARY KEY (`idgroup`,`idaccount`),
  KEY `idaccountgroup_join_idx` (`idaccount`),
  CONSTRAINT `idaccountgroup_join` FOREIGN KEY (`idaccount`) REFERENCES `account` (`idaccount`),
  CONSTRAINT `idgroup_join` FOREIGN KEY (`idgroup`) REFERENCES `group` (`idgroup`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

```
CREATE TABLE `group_location` (
  `idgroup` int NOT NULL,
  `glocation` varchar(45) NOT NULL,
  PRIMARY KEY (`idgroup`,`glocation`),
  CONSTRAINT `idgroup` FOREIGN KEY (`idgroup`) REFERENCES `group` (`idgroup`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `interests` (
  `iduser_profile` int NOT NULL,
  `iduser` int NOT NULL,
  `uinterests` varchar(100) NOT NULL,
  PRIMARY KEY (`iduser_profile`,`iduser`,`uinterests`),
  KEY `iduserinterets_idx` (`iduser`),
  CONSTRAINT `iduser_profileinterests` FOREIGN KEY (`iduser_profile`) REFERENCES `user_profile` (`iduser_profile`),
  CONSTRAINT `iduserinterets` FOREIGN KEY (`iduser`) REFERENCES `user` (`iduser`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `like` (
  `idpost` int NOT NULL,
  `idaccount` int NOT NULL,
  PRIMARY KEY (`idpost`,`idaccount`),
  KEY `idaccountlike_idx` (`idaccount`),
  CONSTRAINT `idaccountlike` FOREIGN KEY (`idaccount`) REFERENCES `account` (`idaccount`),
  CONSTRAINT `idpostlike` FOREIGN KEY (`idpost`) REFERENCES `post` (`idpost`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

```
CREATE TABLE `like` (
  `idpost` int NOT NULL,
  `idaccount` int NOT NULL,
  PRIMARY KEY (`idpost`,`idaccount`),
  KEY `idaccountlike_idx` (`idaccount`),
  CONSTRAINT `idaccountlike` FOREIGN KEY (`idaccount`) REFERENCES `account` 
(`idaccount`),
  CONSTRAINT `idpostlike` FOREIGN KEY (`idpost`) REFERENCES `post` (`idpost`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `make` (
  `idsurvey` int NOT NULL,
  `idaccount` int NOT NULL,
  PRIMARY KEY (`idsurvey`,`idaccount`),
  KEY `idcompanymake` (`idaccount`),
  CONSTRAINT `idcompanymake` FOREIGN KEY (`idaccount`) REFERENCES `company` 
(`idcompany`),
  CONSTRAINT `idsurveymake` FOREIGN KEY (`idsurvey`) REFERENCES `survey` (`idsurvey`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `messages` (
  `idreceiveaccount` int NOT NULL,
  `idsendaccount` int NOT NULL,
  `send_time` date NOT NULL,
  `content` varchar(500) NOT NULL,
  PRIMARY KEY (`idreceiveaccount`,`idsendaccount`),
  KEY `idsendaccount_idx` (`idsendaccount`),
  CONSTRAINT `idreceiveaccount` FOREIGN KEY (`idreceiveaccount`) REFERENCES `account` 
(`idaccount`),
  CONSTRAINT `idsendaccount` FOREIGN KEY (`idsendaccount`) REFERENCES `account`
(`idaccount`),
  CONSTRAINT `message` CHECK ((`idreceiveaccount` <> `idsendaccount`))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `post` (
  `idpost` int NOT NULL,
  `url` varchar(45) DEFAULT NULL,
  `content` varchar(3000) DEFAULT NULL,
  `idgroup` int DEFAULT NULL,
  `idaccount` int DEFAULT NULL,
  PRIMARY KEY (`idpost`),
  UNIQUE KEY `url_UNIQUE` (`url`),
  KEY `idgrouppost_idx` (`idgroup`),
  KEY `idaccountpost_idx` (`idaccount`),
  CONSTRAINT `idaccountpost` FOREIGN KEY (`idaccount`) REFERENCES `account`
(`idaccount`),
  CONSTRAINT `idgrouppost` FOREIGN KEY (`idgroup`) REFERENCES `group` (`idgroup`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci



CREATE TABLE `rules` (
  `idgroup` int NOT NULL,
  `grules` varchar(400) NOT NULL,
  PRIMARY KEY (`idgroup`,`grules`),
  CONSTRAINT `idgrouprule` FOREIGN KEY (`idgroup`) REFERENCES `group` (`idgroup`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci



CREATE TABLE `skills` (
  `iduser_profile` int NOT NULL,
  `iduser` int NOT NULL,
  `uskills` varchar(45) NOT NULL,
  PRIMARY KEY (`iduser_profile`,`iduser`,`uskills`),
  KEY `iduserskills_idx` (`iduser`),
  CONSTRAINT `iduser_profileskills` FOREIGN KEY (`iduser_profile`) REFERENCES
`user_profile` (`iduser_profile`),
  CONSTRAINT `iduserskills` FOREIGN KEY (`iduser`) REFERENCES `user` (`iduser`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci


CREATE TABLE `subscribes` (
  `idforum` int NOT NULL,
  `ssubscribes` varchar(45) NOT NULL,
  PRIMARY KEY (`idforum`,`ssubscribes`),
  CONSTRAINT `idforumsubscribes` FOREIGN KEY (`idforum`) REFERENCES `forum`
(`idforum`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `survey` (
  `idsurvey` int NOT NULL,
  `survey_name` varchar(45) NOT NULL,
  `question` varchar(45) NOT NULL,
  `result` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`idsurvey`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci


CREATE TABLE `university` (
  `iduniversity` int NOT NULL,
  `uni_name` varchar(45) NOT NULL,
  `foundation` date NOT NULL,
  `location` varchar(45) NOT NULL,
  PRIMARY KEY (`iduniversity`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `user` (
  `iduser` int NOT NULL,
  `password` varchar(45) NOT NULL,
  `fname` varchar(45) NOT NULL,
  `lname` varchar(45) NOT NULL,
  `mail` varchar(45) NOT NULL,
  `iflag` tinyint NOT NULL,
  `sflag` tinyint NOT NULL,
  `student_no` varchar(11) DEFAULT NULL,
  `eflag` tinyint NOT NULL,
  `age` int NOT NULL,
  PRIMARY KEY (`iduser`),
  UNIQUE KEY `student_no_UNIQUE` (`student_no`),
  CONSTRAINT `iduser` FOREIGN KEY (`iduser`) REFERENCES `account` (`idaccount`),
  CONSTRAINT `age` CHECK ((`age` >= 18))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci


CREATE TABLE `user_profile` (
  `iduser_profile` int NOT NULL,
  `iduser` int NOT NULL,
  `about` varchar(300) DEFAULT NULL,
  `activity` varchar(45) DEFAULT NULL,
  `photo` blob,
  `job` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`iduser_profile`,`iduser`),
  KEY `iduser_idx` (`iduser`),
  CONSTRAINT `iduser_profile` FOREIGN KEY (`iduser`) REFERENCES `user` (`iduser`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

```
CREATE TABLE `works_for` (
  `idcompany` int NOT NULL,
  `idemployee` int NOT NULL,
  PRIMARY KEY (`idcompany`,`idemployee`),
  KEY `idemployeeworks_for_idx` (`idemployee`),
  CONSTRAINT `idcompanyworks_for` FOREIGN KEY (`idcompany`) REFERENCES `company` (`idcompany`),
  CONSTRAINT `idemployeeworks_for` FOREIGN KEY (`idemployee`) REFERENCES `user` (`iduser`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf
8mb4_0900_ai_ci
```

Sample Insert, Delete and Update

```
INSERT INTO `linkedinmoodle`.`account` (`idaccount`) VALUES ('6');
UPDATE `linkedinmoodle`.`account` SET `idaccount` = '7' WHERE (`idaccount` = '6');
DELETE FROM `linkedinmoodle`.`account` WHERE (`idaccount` = '7');
```

```
INSERT INTO `linkedinmoodle`.`user` (`iduser`, `password`, `fname`, `lname`, `mail`, `iflag`, `sflag`, `student_no`, `eflag`, `age`) VALUES ('6', '1453', 'osman', 'ünalır', 'unalir@gmail.com', '1', '1', '00000000001', '1', '35');
UPDATE `linkedinmoodle`.`user` SET `iduser` = '6', `password` = '123osman', `age` = '36' WHERE (`iduser` = '6');
```

```
DELETE FROM `linkedinmoodle`.`user` WHERE (`iduser` = '6')
```

```
INSERT INTO `linkedinmoodle`.`university` (`iduniversity`, `uni_name`, `foundation`, `location`) VALUES ('4', 'ankara uni', '11.11.11', 'ankara');

UPDATE `linkedinmoodle`.`university` SET `uni_name` = 'bogaz içi', `location` = 'istanbul' WHERE (`iduniversity` = '4');

DELETE FROM `linkedinmoodle`.`university` WHERE (`iduniversity` = '4');
```

Triggers

```
CREATE DEFINER=`root`@`%` TRIGGER `company_BEFORE_INSERT` BEFORE INSERT ON `company` FOR EACH ROW BEGIN


IF NEW.idcompany IN (

        SELECT iduser

   FROM user

        )

   then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'HATA:aynı account id kullanılamaz.';


END IF;




END




CREATE DEFINER=`root`@`%` TRIGGER `company_BEFORE_UPDATE` BEFORE UPDATE ON `company` FOR EACH ROW BEGIN
```

```
IF NEW.idcompany IN (
        SELECT iduser
   FROM user
        )
   then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'HATA:güncellemede aynı account id kullanılamaz.';

END IF;


END



CREATE DEFINER=`root`@`%` TRIGGER `course_BEFORE_INSERT` BEFORE INSERT ON `course` FOR EACH ROW BEGIN

IF NEW.idinstructor IN (
        SELECT iduser
   FROM user
   where iflag=0
        )
   then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'HATA: insturctor olmayanlar kurs veremez';
END IF;
```

```
CREATE DEFINER=`root`@`%` TRIGGER `event_BEFORE_INSERT` BEFORE
INSERT ON `event` FOR EACH ROW BEGIN
if
DATEDIFF(NEW.end_date, NEW.start_date) < 0
then  SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'HATA: bitiş tarihi
başlangıç tarihinden önce olamaz';

end if;
END

CREATE DEFINER=`root`@`%` TRIGGER `event_BEFORE_UPDATE` BEFORE
UPDATE ON `event` FOR EACH ROW BEGIN

if
DATEDIFF(NEW.end_date, NEW.start_date) < 0
then  SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'HATA: bitiş tarihi
başlangıç tarihinden önce olamaz';

end if;

END

CREATE DEFINER=`root`@`%` TRIGGER `user_BEFORE_INSERT` BEFORE INSERT
ON `user` FOR EACH ROW BEGIN

IF NEW.iduser IN (
        SELECT idcompany
   FROM company
```

```
        )
   then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error: Can’t use th .';
END IF;




if NEW.iflag=0 and new.sflag=0 and new.eflag=0
then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error: either has to be student employee or instructor';
END IF;
END



CREATE DEFINER=`root`@`%` TRIGGER `user_AFTER_INSERT` AFTER INSERT ON `user` FOR EACH ROW BEGIN
insert into user_profile  (`iduser`, `iduser_profile`) VALUES (new.iduser,new.iduser);


END


CREATE DEFINER=`root`@`%` TRIGGER `user_BEFORE_UPDATE` BEFORE UPDATE ON `user` FOR EACH ROW BEGIN
IF NEW.iduser IN (
        SELECT idcompany
   FROM company
        )
   then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error:güncellemede aynı account id kullanılamaz.';
```

```
END IF;

if NEW.iflag=0 and new.sflag=0 and new.eflag=0
then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error:Either has to be instructor student or employee;
END IF;

END

END
```

Check Constraints

```
  CONSTRAINT `balance` CHECK ((`balance` >= 0))

  CONSTRAINT `semester` CHECK ((`semester` in (_utf8mb4'fall',_utf8mb4'FALL',_utf8mb4'SPRING',_utf8mb4'spring')))

  CONSTRAINT `time_zone` CHECK ((`time_zone` in (_utf8mb4'UTC',_utf8mb4'utc',_utf8mb4'GMT',_utf8mb4'gmt')))

  CONSTRAINT `age` CHECK ((`age` >= 18))
```

SELECT STATEMENTS

1 table

Lists all instructors

```
SELECT * FROM user WHERE iflag=1
```

Lists the group that were created by the user/company with the account id of 2

```
SELECT * FROM linkedinmoodle.group WHERE idaccount=2;
```

Lists the id numbers of all followers of the user with the id number of 1

```
SELECT idfolloweraccount FROM follows WHERE idfollowedaccount=1;
```

2 table

Lists the names of all followers of the user with the id number of 3

```
SELECT distinct fname,lname

FROM follows,user

WHERE idfollowedaccount=3;
```

Lists all the users that are both instructor student and employee at the same time along with a bank balance of greater than 1500$

```
Select distinct fname, lname

From user, bank_account

WHERE iflag = 1 and sflag = 1 and eflag and 1 and balance > 1500
```

Lists all the employess that have at least one follower

```
Select *

from user, follows

where iduser=idfollowedaccount and eflag=1
```

Lists all users and their id’s that takes courses

```
SELECT distinct idcourse,fname,lname

FROM course_join,user

WHERE course_join.idstudent=iduser;
```

3 table

Lists all companies that makes surveys

```
SELECT distinct idcompany,company_name, survey_name

FROM company,survey,make

WHERE idcompany=idaccount and survey.idsurvey=make.idsurvey and
idcompany=make.idaccount
```

Lists all students that take the course “database”

```
SELECT distinct fname,lname,course_name

FROM user,course,course_join

where course_name="database" and user.iduser=course_join.idstudent and
course.idcourse=course_join.idcourse
```

Lists all employess that are connected to the “ziarat” bank

```
SELECT distinct fname,lname,iduser

FROM user,bank_account,works_for

where user.iduser=bank_account.idaccount and eflag=1 and
bank_name="ziraat" and works_for.idemployee=user.iduser
```

CRITICAL

List all employess ordered by their age but only those older than 25

```
SELECT *
FROM user
group by age
having age>25
order by age
```

Lists all users that have shared at least one post along with their contents

```
SELECT distinct fname,lname,content
FROM user,account,post
where account.idaccount=post.idaccount and user.iduser=account.idaccount
```

Lists all instructors that uploaded a file

```
SELECT distinct fname,lname,idfile
FROM user,file
where user.iduser=file.idstudent and iflag=1
```

List all users that have ever send a message that contains “hi” keyword in it

```
SELECT distinct fname,lname
FROM user,messages
where messages.content like "%hi%" and user.iduser=messages.idsendaccount
```

Lists all employess that have less bank balance than 500$

```
SELECT distinct fname,lname
FROM user,bank_account
where user.iduser=bank_account.idaccount and eflag=1 and balance<500
```