**CS425: Database Design and Applications**

**(Database Organization)**

**Team member:** Danna Liu, Shanshan Jiang, and Zhizheng Li

Edit by Danna Liu

**Introduction**

There are 3 parts in this document.

**Part I**

1. ER diagram according to the description of the class project
2. SQL schemas with relevant keys and functional dependencies of each schema, then create the tables with the specified attribute names and appropriate constraints such as primary keys, foreign keys, and unique attributes.

**Part II**

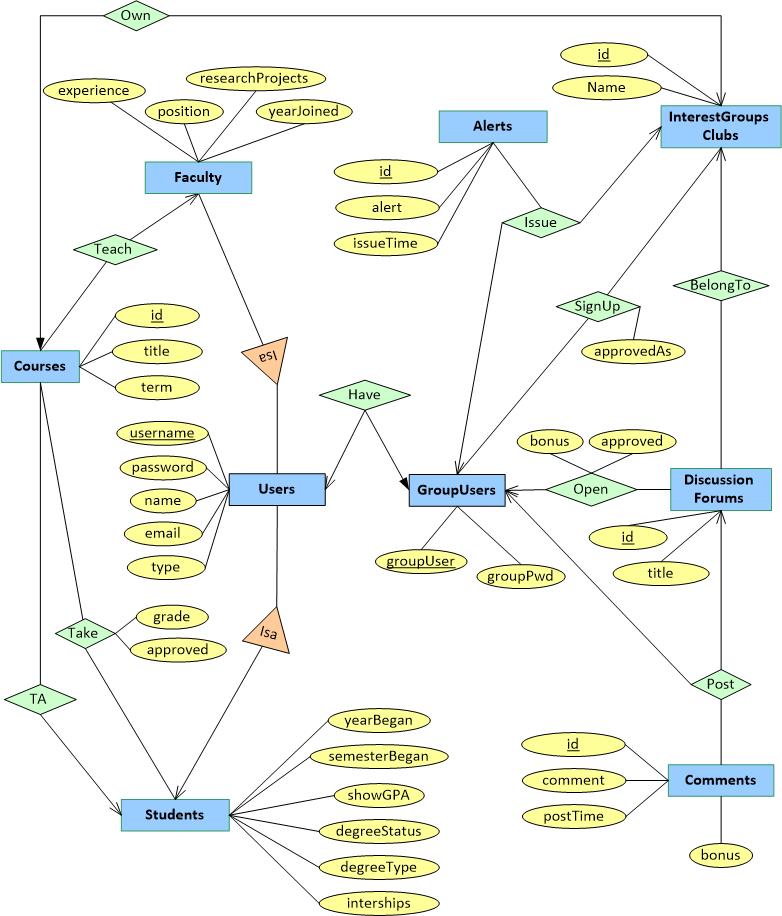
Show that the tables designed are at least 3NF.

**Part III**

Specify system requirements and features with user case diagrams and show ownerships among members with a features ownership form.

**Part I**

1. ER diagram



*\* GPA will be calculated only when required.*

*\* The site administrator is one type of users.*

2. SQL schemas

**All the primary keys are underlined. Other candidate keys (CK), foreign keys (FK), AND functional dependency (FD) are specified.**

***Users*** (username, password, name, email, type)

**CK:** email

**FD:** username → password name email type

**FD:** email → username password name type

***Faculty*** (username, position, yearJoined, experience, researchProjects)

**FK:** username references username in Users

**FD:** username → position yearJoined experience researchProjects

***Students*** (username, yearBegan, SemesterBegan, showGPA, degreeStatus, degreeType, interships)

**FK:** username references username in Users

**FD:** username → yearBegan, SemesterBegan, showGPA, degreeStatus, degreeType, interships

***InterestGroupsClubs*** (id, name)

**FD:** id → name

***Courses*** (id, title, term, instructor, TA, groupID)

**FK:** instructor references username in Faculty

**FK:** TA references username in Students

**FK:**groupID references id in InterestGroupsClubs

**CK:** groupID

**FD:** id → title term instructor TA groupID

**FD:** groupID → id title term instructor TA

***Take*** (student, courseID, grade, approved)

**FK:**student references username in Students

**FK:**courseID references id in Courses

**FD:** student courseID → grade approved

***GroupUsers*** (groupUser, username, groupPwd)

**FK:** username references username in Users

**CK:** username

**FD:** groupUser→ username groupPwd

**FD:** username →groupUser groupPwd

***Alerts*** (id, alert, issuer, issueTime, groupID)

**FK:** issuer references groupUser in GroupUsers

**FK:** groupID references id in InterestGroupsClubs

**FD:** id → alert issuer issueTime groupID

***SignUp*** (member, groupID, approvedAs)

**FK:** member references groupUser in GroupUsers

**FK:**groupID references id in InterestGroupsClubs

**FD:** member groupID → approvedAs

***DisscussionForums*** (id, title, groupID, creator, bonus, approved)

**FK:**groupID references id in InterestGroupsClubs

**FK:** creator references groupUser in GroupUsers

**FD:** id → title groupID creator bonus approved

***Comments*** (id, comment, author, forumID, postTime, bouns)

**FK:** author references groupUser in GroupUsers

**FK:**forumID references id in DiscussionForums

**FD:** id → comment, author, forumID, postTime, bouns

**Create all the tables with the specified attribute names AND appropriate constraints**

CREATE TABLE Users(

username VARCHAR(25) NOT NULL,

password NUMBER(10) NOT NULL ,

name VARCHAR(25) NOT NULL,

email VARCHAR(30) NOT NULL UNIQUE,

type VARCHAR(15) NOT NULL,

PRIMARY KEY (username)

);

CREATE TABLE Faculty(

username VARCHAR(25) NOT NULL,

position VARCHAR(25) NOT NULL ,

yearJoined NUMBER(4) NOT NULL,

experience VARCHAR(255) NOT NULL,

researchProjects VARCHAR(255) NOT NULL,

PRIMARY KEY (username),

FOREIGN KEY (username) REFERENCES Users (username)

);

CREATE TABLE Students(

username VARCHAR(25) NOT NULL,

yearBegan NUMBER(4) NOT NULL,

SemesterBegan VARCHAR(6)NOT NULL check(SemesterBegan in ('Fall', 'Winter', 'Spring', 'Summer')),

showGPA BOOLEAN NOT NULL,

degreeStatus VARCHAR(20) NOT NULL,

degreeType VARCHAR(20) NOT NULL,

internships VARCHAR(255) NOT NULL,

PRIMARY KEY (username),

FOREIGN KEY (username) REFERENCES Users (username)

);

CREATE TABLE InterestGroupsClubs(

id INT NOT NULL,

name VARCHAR(30) NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE Courses(

id INT NOT NULL,

title VARCHAR(50) NOT NULL,

term VARCHAR(10)NOT NULL,

instructor VARCHAR(25) NOT NULL,

TA VARCHAR(25) NOT NULL,

groupID INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (instructor) REFERENCES Faculty (username),

FOREIGN KEY (TA) REFERENCES Students (username),

FOREIGN KEY (groupID) REFERENCES InterestGroupsClubs (id)

);

CREATE TABLE Take(

student VARCHAR(25) NOT NULL,

courseID INT NOT NULL UNIQUE,

grade VARCHAR(5) NOT NULL,

approved VARCHAR(5) NOT NULL,

PRIMARY KEY (student,courseID),

FOREIGN KEY (student) REFERENCES Students (username),

FOREIGN KEY (courseID) REFERENCES Courses (id)

);

CREATE TABLE GroupUser(

groupUser VARCHAR(25) NOT NULL,

username VARCHAR(25) NOT NULL UNIQUE,

groupPwd NUMBER(10) NOT NULL,

PRIMARY KEY (groupUser)

);

CREATE TABLE Alerts(

id INT NOT NULL,

alert VARCHAR(255) NOT NULL,

issuer VARCHAR(25) NOT NULL,

issueTime DATE NOT NULL,

groupID INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (issuer) REFERENCES GroupUsers (groupUser),

FOREIGN KEY (groupID) REFERENCES InterestGroupsClubs (id)

);

CREATE TABLE SignUp(

member VARCHAR(25) NOT NULL,

groupID INT NOT NULL,

approvedAs VARCHAR(10) NOT NULL,

PRIMARY KEY (member, groupID),

FOREIGN KEY (member) REFERENCES GroupUsers (groupUser),

FOREIGN KEY (groupID) REFERENCES InterestGroupsClubs (id)

);

CREATE TABLE DisscussionForums(

id INT NOT NULL,

title VARCHAR(30) NOT NULL,

groupID INT NOT NULL,

creator VARCHAR(25) NOT NULL,

bonus NUMBER(3) NOT NULL,

approved VARCHAR(5) NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (groupID) REFERENCES InterestGroupsClubs (id),

FOREIGN KEY (creator) REFERENCES GroupUsers (groupUser)

);

CREATE TABLE Comments(

id INT NOT NULL,

comment VARCHAR(255) NOT NULL,

author VARCHAR(25) NOT NULL,

forumID INT NOT NULL,

postTime DATE NOT NULL,

bonus NUMBER(3) NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (author) REFERENCES GroupUsers (groupUser),

FOREIGN KEY (forumID) REFERENCES DisscussionForums (id)

);

**Part II**

**Show that the tables designed above are at least 3NF.**

***Users*** (username, password, name, email, type)

**FD:** username → password name email type

**FD:** email → username password name type

* The key of Relation *Users* is {username} and {email}
* There are two nontrivial FD for Relation *Users*
* Each left side of the nontrivial FD is superkey

Thus, we could say that Relation *Users* is in **BCNF**.

***Faculty*** (username, position, yearJoined, experience, researchProjects)

**FD:** username → position yearJoined experience researchProjects

* The key of Relation *Faculty* is {username}
* There is only one nontrivial FD for Relation *Faculty*
* The left side of the nontrivial FD is superkey

Thus, we could say that Relation *Faculty* is in **BCNF**.

***Students*** (username, yearBegan, SemesterBegan, showGPA, degreeStatus, degreeType, interships)

**FD:** username → yearBegan, SemesterBegan, showGPA, degreeStatus, degreeType, interships

* The key of Relation *Students* is {username}
* There is only one nontrivial FD for Relation *Students*
* The left side of the nontrivial FD is superkey

Thus, we could say that Relation *Students* is in **BCNF**.

***InterestGroupsClubs*** (id, name)

**FD:** id → name

* There is two-attribute relation
* Our textbook has already been proved that ***any two-attribute relation is in BCNF.***

Thus, we could say that Relation *InterestGroupsClubs* is in **BCNF**.

***Courses*** (id, title, term, instructor, TA, groupID)

**FD:** id → title term instructor TA groupID

**FD:** groupID → id title term instructor TA

* The key of Relation *Courses* is {id} and {groupID}
* There are two nontrivial FD for Relation *Courses*
* Each left side of the nontrivial FD is superkey

Thus, we could say that Relation *Courses* is in **BCNF**.

***Take*** (student, courseID, grade, approved)

**FD:** student courseID → grade approved

* The key of Relation *Take* is {student, courseID}
* There is only one nontrivial FD for Relation *Take*
* The left side of the nontrivial FD is superkey

Thus, we could say that Relation *Take* is in **BCNF**.

***GroupUsers*** (groupUser, username, groupPwd)

**FD:** groupUser→ username groupPwd

**FD:** username → groupUser groupPwd

* The key of Relation *GroupUsers* is {groupUser} and {username}
* There are two nontrivial FD for Relation *GroupUsers*
* Each left side of the nontrivial FD is superkey

Thus, we could say that Relation *GroupUsers* is in **BCNF**.

***Alerts*** (id, alert, issuer, issueTime, groupID)

**FD:** id → alert issuer issueTime groupID

* The key of Relation *Alerts* is {id}
* There is only one nontrivial FD for Relation *Alerts*
* The left side of the nontrivial FD is superkey

Thus, we could say that Relation *Alerts* is in **BCNF**.

***SignUp*** (member, groupID, approvedAs)

**FD:** member groupID → approvedAs

* The key of Relation *SignUp* is {member, groupID}
* There is only one nontrivial FD for Relation *SignUp*
* The left side of the nontrivial FD is superkey

Thus, we could say that Relation *SignUp* is in **BCNF**.

***DisscussionForums*** (id, title, groupID, creator, bonus, approved)

**FD:** id → title groupID creator bonus approved

* The key of Relation *DisscussionForums* is {id}
* There is only one nontrivial FD for Relation *DisscussionForums*
* The left side of the nontrivial FD is superkey

Thus, we could say that Relation *DisscussionForums* is in **BCNF**.

***Comments*** (id, comment, author, forumID, postTime, bouns)

**FD:** id → comment, author, forumID, postTime, bouns

* The key of Relation*Comments* is {id}
* There is only one nontrivial FD for Relation *Comments*
* The left side of the nontrivial FD is superkey

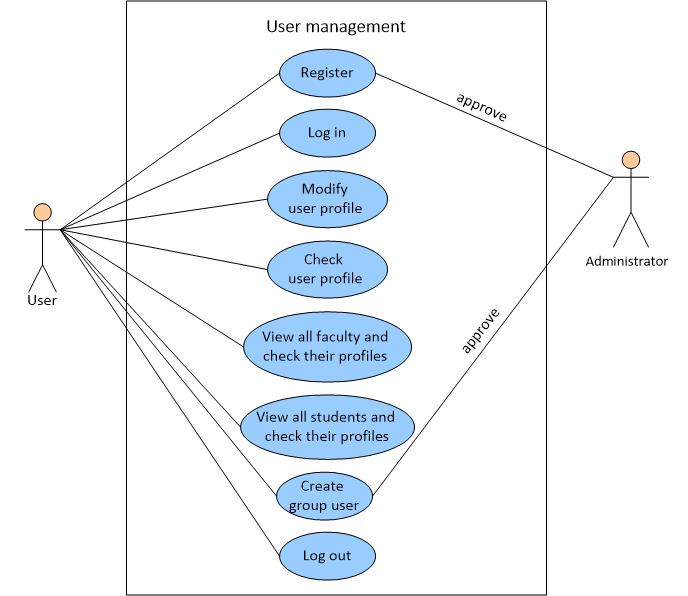
Thus, we could say that Relation *Comments* is in **BCNF**.

**Part III**

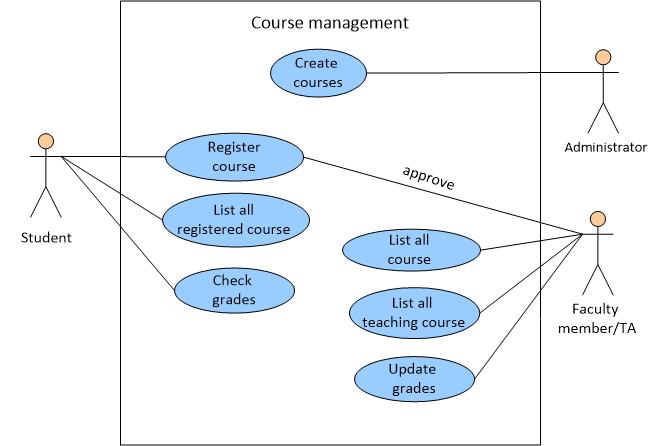
**The specified feature ownership among team members**

To identify and organize system requirements, the class project *Interactive Students/Faculties Networking system* has been divided into three large parts – **user management, course management, and discussion group management**, as following:

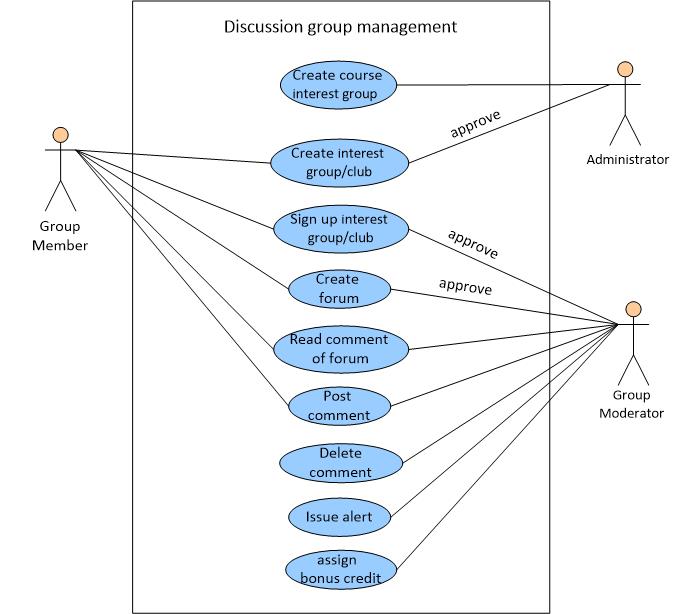
1. **User management**



**2. Course management**

****

**3. Discussion group management**



**Feature ownership form**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **User management** | **Course management** | **Discussion group management** |
| Danna Liu | √ |  |  |
| Shanshan Jiang |  | √ |  |
| Zhizheng Li |  |  | √ |