

## CPSC 304 Project Cover Page

Milestone #: \_\_\_\_2\_\_\_\_

Date: \_\_\_\_2021/10/25\_\_\_\_

Group Number: \_\_\_\_70\_\_\_\_

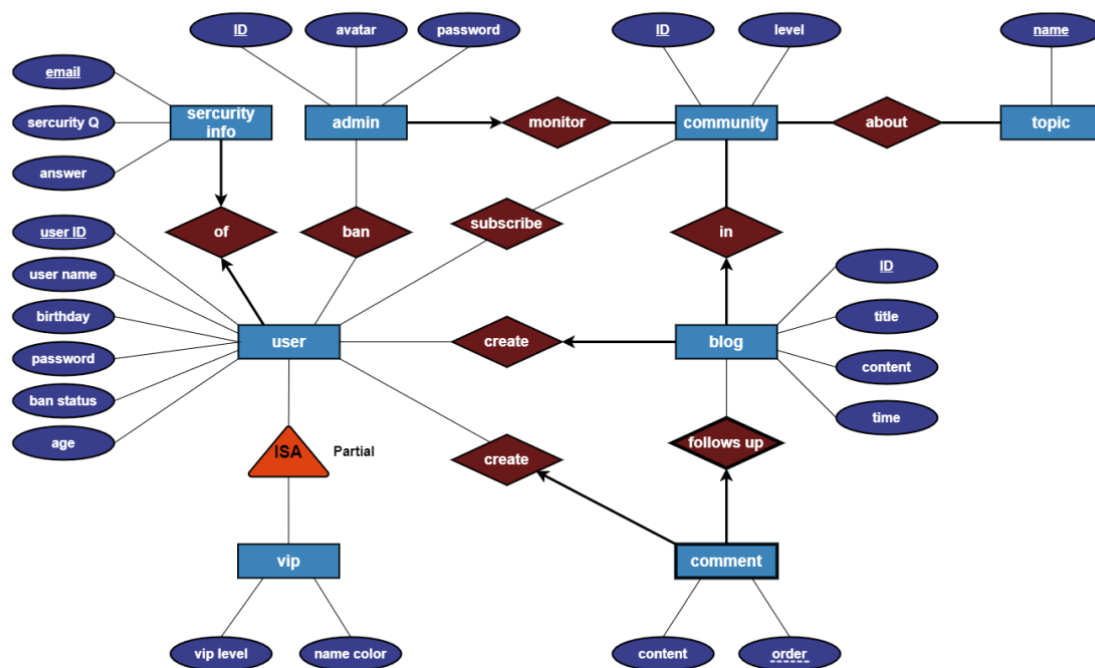
<b>Name</b>	<b>Student Number</b>	<b>CS Alias (Userid)</b>	<b>Preferred E-mail Address</b>
Annie Wang	28056638	h5l3b	1000wangyiqian@gmail.com
Cheryl Yu	88549647	e8j3b	opkisky@gmail.com
Ruinan Jin	61496774	k8g3b	jinruinan@163.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely By those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

## ER Diagram

Our new ER diagram:



Our changes: We basically followed all the feedback of MileStone 1 and made changes. First, the securityInfo is not a weak entity now, and our ISA only has vip as a subclass. All other users are included in the super class. Second, we have added topic as an entity. This let us categorize communities into different topics. In this case, we delete the old “favorites folder” entity. Third, we make comment a weak entity of blog because it depends on the blog, and different blogs have comments sorted by their order. Last, we added more attributes to admin. We also add age as an attribute to user, and time as an attribute in blog just to make them have more attributes.

### **Schema before normalization:**

We also have table that your can refer here

[https://github.students.cs.ubc.ca/CPSC304-2021W-T1/project\\_e8j3b\\_h5l3b\\_k8g3b/blob/master/SQL/sql-before\\_normalization.sql](https://github.students.cs.ubc.ca/CPSC304-2021W-T1/project_e8j3b_h5l3b_k8g3b/blob/master/SQL/sql-before_normalization.sql).

User (userID: INTEGER, userName: CHAR(250), birthday: CHAR(8), userPassword: CHAR(250), ban\_status: INTEGER, age:INTEGER)

**Candidate Key:** {userID}

**FDs:** userID  $\rightarrow$  userName, birthday, userPassword, ban\_status, age

birthday  $\rightarrow$  age

### **Constraints:**

userName is not null.

userPassword is not null.

ban\_status is not null.

age is less than 100.

ban\_status can only be 0 or 1.

VIP (**vip\_ID**: INTEGER, vip\_level: INTEGER, preferred\_colour: CHAR(250))

**Candidate Key:** {vip\_ID}

**FD:** vip\_ID  $\rightarrow$  vip\_level, preferred\_color

### **Constraints:**

Participation constraints for vip\_ID.

vip\_level is not null.

VIP is in partial participation with User.

SecurityInfo\_Of(email: CHAR(250), **User\_ID**: INTEGER,  
security\_question: CHAR(250), answer: CHAR(250))

**Candidate Key:** {email}

**FD:** email  $\rightarrow$  User\_ID, security\_question, answer

**Constraints:**

Participation constraints for User\_ID.

security\_question is not null.

answer is not null.

User\_ID is not null.

Admin\_Monitor(adminID: INTEGER, avatar: CHAR(10), password:  
CHAR(250), **CommunityID**: INTEGER)

**Candidate Key:** {adminID}

**FD:** adminID  $\rightarrow$  avatar, password, CommunityID

**Constraints:**

Participation constraints (total participation) for CommunityID.

avatar is not null.

password is not null.

CommunityID is not null.

Ban(User\_ID: INTEGER, Admin\_ID: INTEGER)

**Candidate Key:** {User\_ID, Admin\_ID}

**FD:** User\_ID, Admin\_ID  $\rightarrow$  User\_ID, Admin\_ID

Community(ID:INTEGER, level:INTEGER)

**Candidate Key:** {ID}

**FD:** ID  $\rightarrow$  level

**Constraints:**

level is not null.

Topic(topic\_name: CHAR(250))

**Candidate Key:** {topic\_name}

About(CommunityID: INTEGER, Topic\_name: CHAR(250))

**Candidate Key:** {CommunityID, Topic\_name}

**Constraints:**

Participation constraints (total participation) for CommunityID, Topic\_name. (Assertion needed)

Subscribe(User\_ID: INTEGER, CommunityID: INTEGER)

**Candidate Key:** {User\_ID, CommunityID}

Blog\_Create\_In(blogId: INTEGER, **User\_ID**: INTEGER, blogTime: DATETIME, **CommunityID**: INTEGER, title: CHAR(250), content CHAR(250))

**Candidate Key:** {blogID}

**FDs:**

blogID → title, content, User\_ID, DATETIME\_record, CommunityID

CommunityID, DATETIME\_record, title → content

User\_ID, DATETIME\_record, CommunityID → title, content

**Constraints:**

Participation constraints (total participation) for User\_ID, CommunityID.

User\_ID is not null.

DATETIME\_record is not null.

CommunityID is not null.

title is not null.

content is not null.

Comment\_Create\_Follows(content: CHAR(250), order: INTEGER, **Blog\_ID**: INTEGER, **User\_ID**: INTEGER)

**Candidate Key:** {Blog\_ID, order}

**FD:** Blog\_ID, order  $\rightarrow$  User\_ID, content

**Constraints:**

Participation constraints for Blog\_ID, User\_ID.

content is not null.

User\_ID is not null.

**Normalization:**

User (UserID: INTEGER, userName: CHAR(250), birthday: CHAR(8), userPassword CHAR(250), ban\_status INTEGER, age: INTEGER)

**FDs:**

User\_ID  $\rightarrow$  userName, birthday, userpassword, ban\_status, age

birthday  $\rightarrow$  age

**Closures:**

User\_ID<sup>+</sup> = {User\_ID, name, birthday, password, ban\_status, age}

birthday<sup>+</sup> = {birthday, age}

The only key is User\_ID.

In the FD  $\text{birthday} \rightarrow \text{age}$ , birthday is not a superkey, so this relation is not in BCNF. Also, age is not part of a key, so this relation is also not in 3NF.

So, we need to decompose.

### **Decompose into BCNF:**

For this FD:  $\text{birthday} \rightarrow \text{age}$

User (UserID, userName, birthday, userPassword, ban\_status)

Birthday\_Age(birthday, age)

Now, both User and Birthday\_Age are in BCNF, done.

Second table:

Blog\_Create\_In (blogID, title, content, **User\_ID**,  
DATETIME\_record, **CommunityID**)

**FDs:**

$\text{blogID} \rightarrow \text{title}, \text{content}, \text{User\_ID}, \text{DATETIME\_record}, \text{CommunityID}$

$\text{CommunityID}, \text{DATETIME\_record}, \text{title} \rightarrow \text{content}$

$\text{User\_ID}, \text{DATETIME\_record}, \text{CommunityID} \rightarrow \text{title}, \text{content}$

(Note: in this blog system, two blogs with the same title cannot be posted in the same community at the same DATETIME\_record. One user cannot post two blogs in the same community at the same DATETIME\_record.)



### Closure:

1.  $\text{blogID}^+ = \{\text{blogID}, \text{title}, \text{content}, \text{User\_ID}, \text{DATETIME\_record}, \text{CommunityID}\}$
2.  $\text{CommunityID}, \text{DATETIME\_record}, \text{title}^+ = \{\text{CommunityID}, \text{DATETIME\_record}, \text{title}, \text{content}\}$
3.  $\text{User\_ID}, \text{DATETIME\_record}, \text{CommunityID}^+ = \{\text{User\_ID}, \text{DATETIME\_record}, \text{CommunityID}, \text{title}, \text{content}\}$

The only key is blogID.

For both the second and third FDs, their LHS are not a superkey, and their RHS are not part of a key, so this relationship is not in BCNF or 3NF.

Decomposition into BCNF:

Apply in Blog\_Create\_In (blogID, title, content, **User\_ID**, DATETIME\_record, **CommunityID**):

$\text{CommunityID}, \text{DATETIME\_record}, \text{title} \rightarrow \text{content}$

CommunityID, DATETIME\_record, title determines content but not user\_ID or ID, thus decompose.

R1(**CommunityID**, DATETIME\_record, title, content)

R2(blogID, title, **User\_ID**, DATETIME\_record, **CommunityID**)

Apply in R2:

User\_ID, DATETIME\_record, CommunityID  $\rightarrow$  title, content

This can be divided into:

User\_ID, DATETIME\_record, CommunityID  $\rightarrow$  title

User\_ID, DATETIME\_record, CommunityID  $\rightarrow$  content (content is not in R2)

User\_ID, DATETIME\_record, CommunityID determines title, but not ID thus violates BCNF.

Decompose:

R3(User\_ID, CommunityID, DATETIME\_record, title)

R4(blogID, DATETIME\_record, **User\_ID, CommunityID**)

The decomposition result:

CID\_time\_title(**CommunityID**, DATETIME\_record, title, content)

UID\_CID\_time(**User\_ID, CommunityID**, DATETIME\_record, title)

BID(blogID, **User\_ID, CommunityID**, DATETIME\_record)

### **Schema after normalization:**

User (userID: INTEGER, userName: CHAR(250), birthday: CHAR(8), userPassword: CHAR(250), ban\_status: INTEGER)

**Candidate Key:** {userID}

**FD:** userID  $\rightarrow$  userName, birthday, userPassword, ban\_status

**Constraints:**

userName is not null.

userPassword is not null.

ban\_status is not null.

ban\_status can only be 0 or 1.

Birthday\_Age(birthday: CHAR(8), age: INTEGER)

**Candidate Key:** {birthday}

**FD:** birthday  $\rightarrow$  age

**Constraints:**

age should be less than 100.

VIP (**vip\_ID**: INTEGER, vip\_level: INTEGER, preferred\_colour: CHAR(250))

**Candidate Key:** {vip\_ID}

**FD:** vip\_ID  $\rightarrow$  vip\_level, preferred\_color

**Constraints:**

Participation constraints for vip\_ID.

vip\_level is not null.

VIP is in partial participation with the User.

SecurityInfo\_Of(email: CHAR(250), **User\_ID**: INTEGER,  
security\_question: CHAR(250), answer: CHAR(250))

**Candidate Key:** {email}

**FD:** email  $\rightarrow$  User\_ID, security\_question, answer

**Constraints:**

Participation constraints for UserID.

security\_question is not null.

answer is not null.

User\_ID is not null.

Admin\_Monitor(adminID: INTEGER, avatar: CHAR(10), password:  
CHAR(250), **CommunityID**: INTEGER)

**Candidate Key:** {ID}

**FD:** adminID  $\rightarrow$  avatar, password, CommunityID

**Constraints:**

Participation constraints for CommunityID.

avatar is not null.

password is not null.

CommunityID is not null.

Ban(**User\_ID**: INTEGER, **Admin\_ID**: INTEGER)

**Candidate Key:** {User\_ID, Admin\_ID}

**Constraints:**

Participation constraints for User\_ID, Admin\_ID.

Community(ID:INTEGER, level:INTEGER)

**Candidate Key:** {ID}

**FD:** ID  $\rightarrow$  level

**Constraints:**

level is not null.

Topic(topic\_name: CHAR(250))

**Candidate Key:** {topic\_name}

About(CommunityID: INTEGER, Topic\_name: CHAR(250))

**Candidate Key:** {CommunityID, Topic\_name}

**FD:** CommunityID, Topic\_name  $\rightarrow$  CommunityID, Topic\_name

**Constraints:**

Participation constraints for CommunityID, Topic\_name. (Assertion needed)

Subscribe(User\_ID: INTEGER, CommunityID: INTEGER)

**Candidate Key:** {User\_ID, CommunityID}

**FD:** User\_ID, CommunityID → User\_ID, CommunityID

**Constraints:**

Participation constraints for User\_ID, CommunityID.

CID\_DATETIME\_Title(**CommunityID**: INTEGER,  
DATETIME\_record: DATETIME, title: CHAR(250), content:  
CHAR(250))

**Candidate Key:** {CommunityID, DATETIME\_record, title}

**FD:** CommunityID, DATETIME\_record, title → content

**Constraints:**

Participation constraints for CommunityID.

content is not null.

UID\_CID\_DATETIME(**User\_ID**: INTEGER, **CommunityID**:  
INTEGER, DATETIME\_record: DATETIME, title: CHAR(250))

**Candidate Key:** {User\_ID, CommunityID, DATETIME\_record}

**FD:** User\_ID, DATETIME\_record, CommunityID → title

**Constraints:**

Participation constraints for User\_ID, CommunityID.

DATETIME is not null.

BID(blogID: INTEGER, **User\_ID**: INTEGER, **communityID**:  
INTEGER ,DATETIME\_record: DATETIME)

**Candidate Key:** {blogID}

**FD:** blogID  $\rightarrow$  DATETIME\_record, User\_ID, communityID

**Constraints:**

Participation constraints for User\_ID, CommunityID.

User\_ID is not null.

CommunityID is not null.

title is not null.

Comment\_Create\_Follows(content: CHAR(250), order: INTEGER,  
**Blog\_ID**: INTEGER,**User\_ID**: INTEGER)

**Candidate Key:** {Blog\_ID, order}

**FD:** Blog\_ID, order  $\rightarrow$  User\_ID, content

**Constraints:**

Participation constraints for Blog\_ID, User\_ID.

content is not null.

User\_ID is not null.

SQL DDL (

[https://github.students.cs.ubc.ca/CPSC304-2021W-T1/project\\_e8j3b\\_h5l3b\\_k8g3b/tree/master/SQL](https://github.students.cs.ubc.ca/CPSC304-2021W-T1/project_e8j3b_h5l3b_k8g3b/tree/master/SQL)):

```
CREATE TABLE User (
```

```

        userID INTEGER,

        userName CHAR(250) NOT NULL,

        birthday CHAR(8),

        userPassword CHAR(250) NOT NULL,

        ban_status INTEGER NOT NULL,

        PRIMARY KEY(userID)
    )

CREATE TABLE Birthday_Age(

    birthday CHAR(8),

    age INTEGER,

    PRIMARY KEY(birthday)
)

CREATE TABLE VIP(

    vipID INTEGER,

    vip_level INTEGER NOT NULL,

    preferred_color CHAR(250),

    PRIMARY KEY(vipID),

    FOREIGN KEY(vipID) REFERENCES User(userID) ON DELETE CASCADE ON
UPDATE CASCADE
)

CREATE TABLE SecurityInfo_Of(

    userID INTEGER,

```



```
email CHAR(250),

security_question CHAR(250) NOT NULL,

answer CHAR(250) NOT NULL,

PRIMARY KEY(email),

FOREIGN KEY(userID) REFERENCES User(userID) ON DELETE CASCADE ON
UPDATE CASCADE
)
```

```
CREATE TABLE Admin_Monitor(

adminID INTEGER,

avatar CHAR(10) NOT NULL,

adminPassword CHAR(250) NOT NULL,

communityID INTEGER NOT NULL,

PRIMARY KEY(adminID),

FOREIGN KEY(communityID) REFERENCES Community(communityID) ON
DELETE CASCADE ON UPDATE CASCADE
)
```

```
CREATE TABLE Ban(

userID INTEGER,

adminID INTEGER,

PRIMARY KEY(userID,adminID),

FOREIGN KEY(userID) REFERENCES User(userID) ON DELETE CASCADE ON
UPDATE CASCADE,

FOREIGN KEY(adminID) REFERENCES Admin_Monitor(adminID) ON DELETE
CASCADE ON UPDATE CASCADE
)
```

```
CREATE TABLE Community(
```

```
    communityID INTEGER,
```

```
    community_level INTEGER NOT NULL,
```

```
    PRIMARY KEY (communityID)
```

```
)
```

```
CREATE TABLE Topic(
```

```
    topic_name CHAR(250),
```

```
    PRIMARY KEY (topic_name)
```

```
)
```

```
CREATE TABLE About(
```

```
    communityID INTEGER,
```

```
    topic_name CHAR(250),
```

```
    PRIMARY KEY (communityID, topic_name),
```

```
    FOREIGN KEY (communityID) REFERENCES Community (communityID) ON  
DELETE CASCADE ON UPDATE CASCADE,
```

```
    FOREIGN KEY (topic_name) REFERENCES Topic (topic_name) ON DELETE  
CASCADE ON UPDATE CASCADE
```

```
)
```

```
CREATE TABLE Subscribe(
```

```
    userID INTEGER,
```

```
    communityID INTEGER,
```

```

PRIMARY KEY(userID,communityID),

FOREIGN KEY(userID) REFERENCES User(userID) ON DELETE CASCADE ON
UPDATE CASCADE,

FOREIGN KEY(communityID) REFERENCES Community(communityID) ON
DELETE CASCADE ON UPDATE CASCADE
)

CREATE TABLE CID_DATETIME_Title(

communityID INTEGER,

DATETIME_record DATETIME,

title CHAR(250),

content CHAR(250) NOT NULL,

PRIMARY KEY(communityID,DATETIME_record,title),

FOREIGN KEY(communityID) REFERENCES Community(communityID) ON
DELETE CASCADE ON UPDATE CASCADE
)

CREATE TABLE UID_CID_DATETIME(

userID INTEGER,

communityID INTEGER,

DATETIME_record DATETIME,

title CHAR(250) NOT NULL,

PRIMARY KEY(userID,communityID,DATETIME_record),

FOREIGN KEY(userID) REFERENCES User(userID) ON DELETE CASCADE ON
UPDATE CASCADE,

FOREIGN KEY(communityID) REFERENCES Community(communityID) ON
DELETE CASCADE ON UPDATE CASCADE

```

```

)

CREATE TABLE BID(

    blogId INTEGER,

    userID INTEGER NOT NULL,

    communityID INTEGER NOT NULL,

    DATETIME_record DATETIME NOT NULL,

    PRIMARY KEY(blogId) ,

    FOREIGN KEY(userID) REFERENCES User(userID) ON DELETE CASCADE ON
UPDATE CASCADE,

    FOREIGN KEY(communityID) REFERENCES Community(communityID) ON
DELETE CASCADE ON UPDATE CASCADE

)

CREATE TABLE Comment_Create_Follows(

    order INTEGER,

    blogID INTEGER,

    userID INTEGER NOT NULL,

    content CHAR(250) NOT NULL,

    PRIMARY KEY(order, blogID) ,

    FOREIGN KEY(blogID) REFERENCES BID(blogID) ON DELETE CASCADE ON
UPDATE CASCADE,

    FOREIGN KEY(userID) REFERENCES User(userID) ON DELETE CASCADE ON
UPDATE CASCADE

)

-- populate this table --

```

```
-- format

-- up to 999 users

-- UserID1: 3040001

-- vipID: 3041001

-- blogID: 3042001

-- commentID: 3043001

-- adminID: 3044001

-- communityID: 3045001

-- avatar: autobot1

-- adminPwd: adminpassword1

-- vip_level: 1

-- community_level: 1

-- Birthday: 2000-01-01

-- ban_status: 1 or 0

-- password: password3040001
```

```
INSERT INTO User
VALUES (3040001, 'Amy', '2000-01-01', 'password3040001', 0)
```

```
INSERT INTO User
VALUES (3040002, 'Bob', '2000-01-02', 'password3040002', 0)
```

```
INSERT INTO User
VALUES (3040003, 'Cindy', '2000-01-03', 'password3040003', 0)
```

```
INSERT INTO User
VALUES (3040004, 'Darwin', '2000-01-04', 'password3040004', 0)
```

```
INSERT INTO User
VALUES (3040005, 'Grace', '2000-01-05', 'password3040005', 1)
```

```
INSERT INTO Birthday_Age VALUES('2000-01-01',21)
```

```
INSERT INTO Birthday_Age VALUES('2000-01-02',21)
```

```
INSERT INTO Birthday_Age VALUES('2000-01-03',21)
```

```
INSERT INTO Birthday_Age VALUES('2000-01-04',21)
```

```
INSERT INTO Birthday_Age VALUES('2000-01-05',21)
```

```
INSERT INTO VIP VALUES(3041001,3,'blue')
```

```
INSERT INTO VIP VALUES(3041002,8,NULL)
```

```
INSERT INTO VIP VALUES(3041003,7,'green')
```

```
INSERT INTO VIP VALUES(3041004,5,'red')
```

```
INSERT INTO VIP VALUES(3041005,4,'purple')
```

```
INSERT INTO SecurityInfo_Of VALUES(3040001,'jim1@gmail.com','What is  
the favorite food?','apple')
```

```
INSERT INTO SecurityInfo_Of VALUES(3040002,'jim2@gmail.com','What is  
the favorite color?','blue')
```

```
INSERT INTO SecurityInfo_Of VALUES(3040003,'jim3@gmail.com','What is  
the favorite animal?','dog')
```

```
INSERT INTO SecurityInfo_Of VALUES(3040004,'jim4@gmail.com','What is  
the favorite singer?','Jay Chou')
```

```
INSERT INTO SecurityInfo_Of VALUES(3040005,'jim5@gmail.com','What is  
the favorite subject?','computer science')
```

```
INSERT INTO Admin_Monitor  
VALUES(3044001,'autobot1','adminpassword1',3043001)
```

```
INSERT INTO Admin_Monitor
VALUES (3044002, 'autobot2', 'adminpassword2', 3043002)
```

```
INSERT INTO Admin_Monitor
VALUES (3044003, 'autobot3', 'adminpassword3', 3043003)
```

```
INSERT INTO Admin_Monitor
VALUES (3044004, 'autobot4', 'adminpassword4', 3043004)
```

```
INSERT INTO Admin_Monitor
VALUES (3044005, 'autobot5', 'adminpassword5', 3043005)
```

```
INSERT INTO Ban      VALUES (3040001, 3044001)
```

```
INSERT INTO Ban      VALUES (3040002, 3044002)
```

```
INSERT INTO Ban      VALUES (3040003, 3044003)
```

```
INSERT INTO Ban      VALUES (3040004, 3044004)
```

```
INSERT INTO Ban      VALUES (3040005, 3044005)
```

```
INSERT INTO Community VALUES (3045001, 1)
```

```
INSERT INTO Community VALUES (3045002, 2)
```

```
INSERT INTO Community VALUES (3045003, 3)
```

```
INSERT INTO Community VALUES (3045004, 4)
```

```
INSERT INTO Community VALUES (3045005, 5)
```

```
INSERT INTO Topic VALUES ("Sports")
```

```
INSERT INTO Topic VALUES ("Music")
```

```
INSERT INTO Topic VALUES ("Cook")
```

```
INSERT INTO Topic VALUES ("Anime")
```

```
INSERT INTO Topic VALUES ("IT")
```

```
-- many to many

INSERT INTO About      VALUES(3045001, "Sports")

INSERT INTO About      VALUES(3045001, "Music")

INSERT INTO About      VALUES(3045002, "Music")

INSERT INTO About      VALUES(3045003, "Cook")

INSERT INTO About      VALUES(3045004, "Anime")

INSERT INTO About      VALUES(3045005, "IT")

INSERT INTO About      VALUES(3045006, "IT")


INSERT INTO Subscribe   VALUES(3040001,3045001)

INSERT INTO Subscribe   VALUES(3040002,3045002)

INSERT INTO Subscribe   VALUES(3040003,3045003)

INSERT INTO Subscribe   VALUES(3040004,3045004)

INSERT INTO Subscribe   VALUES(3040005,3045005)


INSERT INTO CID_DATETIME_Title  VALUES(3045001,0001-01-01
00:00:00.0000001,"testing_title","testing_content")

INSERT INTO CID_DATETIME_Title  VALUES(3045002,0001-01-01
00:00:00.0000002,"testing_title","testing_content")

INSERT INTO CID_DATETIME_Title  VALUES(3045003,0001-01-01
00:00:00.0000003,"testing_title","testing_content")

INSERT INTO CID_DATETIME_Title  VALUES(3045004,0001-01-01
00:00:00.0000004,"testing_title","testing_content")

INSERT INTO CID_DATETIME_Title  VALUES(3045005,0001-01-01
00:00:00.9999999,"testing_title","testing_content")
```



```
INSERT INTO UID_CID_DATETIME VALUES (3040001,3045001,0001-01-01
00:00:00.0000001,"testing_title")
```

```
INSERT INTO UID_CID_DATETIME VALUES (3040002,3045002,0001-01-01
00:00:00.0000002,"testing_title")
```

```
INSERT INTO UID_CID_DATETIME VALUES (3040003,3045003,0001-01-01
00:00:00.0000003,"testing_title")
```

```
INSERT INTO UID_CID_DATETIME VALUES (3040004,3045004,0001-01-01
00:00:00.0000004,"testing_title")
```

```
INSERT INTO UID_CID_DATETIME VALUES (3040005,3045005,0001-01-01
00:00:00.9999999,"testing_title")
```

```
INSERT INTO BID VALUES (3042001,3040001,3045001,0001-01-01
00:00:00.0000001)
```

```
INSERT INTO BID VALUES (3042002,3040002,3045002,0001-01-01
00:00:00.0000002)
```

```
INSERT INTO BID VALUES (3042003,3040003,3045003,0001-01-01
00:00:00.0000003)
```

```
INSERT INTO BID VALUES (3042004,3040004,3045004,0001-01-01
00:00:00.0000004)
```

```
INSERT INTO BID VALUES (3042005,3040005,3045005,0001-01-01
00:00:00.9999999)
```

```
INSERT INTO Comment_Create_Follows
VALUES (1,3042001,3040001,"testing_content")
```

```
INSERT INTO Comment_Create_Follows
VALUES (2,3042002,3040002,"testing_content")
```

```
INSERT INTO Comment_Create_Follows
VALUES (3,3042003,3040003,"testing_content")
```

```
INSERT INTO Comment_Create_Follows
VALUES (4,3042004,3040004,"testing_content")
```

```
INSERT INTO Comment_Create_Follows  
VALUES (5,3042005,3040005,"testing_content")
```

8.

INSERTION:

Add a VIP user into User and VIP.

Add the corresponding security info into SecurityInfo\_Of.

Add a topic into Topic.

Add a community and a relationship of the topic and the community into About.

Add an admin that monitors the community into Admin\_Monitor.

Add a blog posted by the user in the community.

Add a comment following up the blog by the same user.

UPDATE:

Update the level of the community.

Update the password of the user.

Update the name color of the vip user.

Update the password of the admin.

Update the answer to the securityInfo.

DELETE:

Drop the blog, also drop its comments.

Drop the vip user and the corresponding security info.

Drop the admin.