domain General\_number1 UNSIGNED integer maximum integer 65535

domain General\_number2 UNSIGNED integer maximum integer 255

domain General\_number3 UNSIGNED integer maximum integer 1.67e+7

domain General\_number4 UNSIGNED integer maximum integer 4.29e+9

domain User\_name variable length character string maximum length 20

domain General\_character1 variable length character string maximum length 255

domain General\_character2 variable length character string maximum length 100

domain General\_character3 variable length character string maximum length 30

domain General\_text1 variable length text string maximum length 65535

domain General\_text2 variable length text string maximum length 50

domain Date datetime format: 0000-00-00 00:00:00

user(

ID General\_number1 NOT NULL AUTO\_INCREMENT,

user\_login User\_name NOT NULL,

user\_pass General\_character1 NOT NULL,

user\_email General\_character2 NOT NULL,

user\_registered Date NOT NULL,

display\_name General\_character3 NOT NULL,

profile\_name General\_character2 NOT NULL)

Primary Key ID

usermeta(

umeta\_id General\_number1 NOT NULL AUTO\_INCREMENT,

user\_id General\_number1 NOT NULL,

meta\_key General\_character1 NOT NULL,

meta\_value General\_text1 NOT NULL)

Primary Key umeta\_id

Foreign Key user\_id References user(ID) ON DELETE CASCADE ON UPDATE CASCADE

association(

association\_ID General\_number1 NOT NULL AUTO\_INCREMENT,

association\_name General\_text2 NOT NULL,

memberNum General\_number1 NOT NULL DEFAULT 1,

association\_description General\_text1 NULL,

chatroom\_ID General\_number1 NULL,

is\_department General\_number2 NOT NULL DEFAULT 0)

Primary Key association\_ID

Foreign Key chatroom\_ID References user(ID) ON DELETE SET NULL ON UPDATE CASCADE

association\_touser(

ID General\_number1 NOT NULL AUTO\_INCREMENT,

association\_ID General\_number1 NOT NULL,

chatroom\_ID General\_number1 NULL,

user\_ID General\_number1 NOT NULL,

user\_level General\_number2 NOT NULL DEFAULT 1,

user\_sendLastTime Date NOT NULL DEFAULT: 0000-00-00 00:00:00)

Primary Key ID

Foreign Key association\_ID Reference association(association\_ID) ON DELETE CASCADE

Foreign Key user\_ID References user(ID) ON DELETE CASCADE

event(

event\_ID General\_number3 NOT NULL AUTO\_INCREMENT,

event\_name General\_text2 NOT NULL,

event\_date Date NOT NULL DEFAULT: 0000-00-00 00:00:00,

event\_description General\_text1 NULL,

association\_ID General\_number1 NOT NULL,

association\_name General\_text2 NOT NULL,

event\_status General\_number2 NOT NULL DEFAULT 0)

Primary Key event\_ID

Foreign Key association\_ID Reference association(association\_ID) ON DELETE CASCADE

eventmeta(

eventmeta\_ID General\_number3 NOT NULL AUTO\_INCREMENT,

event\_ID General\_number3 NOT NULL,

meta\_key General\_character1 NOT NULL,

meta\_value General\_text1 NOT NULL)

Primary Key eventmeta\_ID

Foreign Key event\_ID Reference event(event\_ID) ON DELETE CASCADE ON UPDATE CASCADE

comment(

comment\_ID General\_number4 NOT NULL AUTO\_INCREMENT,

comment\_association\_ID General\_number1 NOT NULL,

comment\_event\_ID General\_number3 NOT NULL,

comment\_display\_name General\_character3 NOT NULL,

comment\_author\_IP General\_character2 NULL,

comment\_date Date NOT NULL DEFAULT: 0000-00-00 00:00:00,

comment\_content General\_text1 NULL,

comment\_agent General\_character1 NULL

comment\_user\_ID General\_number1 NOT NULL)

Primary Key comment\_ID

Foreign Key comment\_association\_ID Reference association(association\_ID) ON DELETE NO ACTION ON UPDATE CASCADE

Foreign Key comment\_event\_ID Reference event(event\_ID) ON DELETE NO ACTION ON UPDATE CASCADE

Foreign Key comment\_user\_ID Reference user(ID) ON DELETE CASCADE ON UPDATE CASCADE

groupmsg(

msg\_ID General\_number4 NOT NULL AUTO\_INCREMENT,

chatroom\_ID General\_number1 NULL,

msg\_senderID General\_number1 NOT NULL,

msg\_sendername General\_character3 NOT NULL,

msg\_senderIP General\_character2 NULL,

msg\_date Date NOT NULL DEFAULT: 0000-00-00 00:00:00,

msg\_content General\_text1 NULL,

msg\_agent General\_character1 NULL)

Primary Key msg\_ID

Foreign Key msg\_senderID Reference user(ID) ON DELETE NO ACTION ON UPDATE CASCADE

About file organization and index

Usermeta, eventmeta, association\_touser table uses hash file organization, because the primary key of these tables has no other meaning except to distinguish between tuples. The most important of usermeta and eventmeta is to query the correspondence between all meta\_key and meta\_value according to user\_id or event\_ID. Therefore, using hash file organization can speed up query efficiency.

The user, association, association\_touser, event, comment, groupmsg tables use sequential (sort) files. The primary keys of these tables have no other meaning except to distinguish between tuples and foreign keys as other tables. Some tables have a column name user\_login in user. Since the user name cannot be the same, user\_login can be used to distinguish different tuples, but in order to increase the readability of the database by the administrator, the ID primary keys of these tables are retained. And decided to specify these tables to increase the readability of the administrator's lookup database according to the table's primary key sorted in ascending order. If the amount of data is large in the future, the query efficiency will be low. Other methods can be used to improve the file structure, although the readability is reduced, but the query efficiency can be improved.

Index

User: user\_login as the index and display\_name as the full-text index

Usermeta: user\_id as an index

Association:association\_name as a full-text index

Association\_touser: association\_ID and user\_ID as indexes respectively

Event: association\_ID as index, event\_name as full-text index

Eventmata: event\_ID as an index

Comment: comment\_association\_ID, comment\_event\_ID as an index

Groupmsg: chatroom\_ID and msg\_date as indexes respectively

user view

About login

CREATE VIEW web\_login as

SELECT ID

FROM user

WHERE user\_login=’admin’ and user\_pass=’123456’

WITH CHECK OPTION;

About viewing user information

CREATE VIEW web\_checkuserinfo as

SELECT a.user\_email, a.display\_name, a.profile\_name, b.meta\_key, b.meta\_value

FROM user a, usermeta b

WHERE a.user\_id=1 and a.user\_id=b.user\_id;

Find a society by name

CREATE VIEW web\_search\_association as

SELECT association\_name

FROM association

WHERE MATCH(association\_name) AGAINST("computer")

WITH CHECK OPTION;

Find activities by name

CREATE VIEW web\_search\_event as

SELECT event\_name

FROM event

WHERE MATCH(event\_name) AGAINST("computer competition 2019")

WITH CHECK OPTION;

View society

CREATE VIEW web\_check\_association as

SELECT association\_name, memberNum, association\_description

FROM association

WHERE association\_id=’1’

WITH CHECK OPTION;

View event

CREATE VIEW web\_check\_event as

SELECT a.user\_email, a.display\_name, a.profile\_name, b.meta\_key, b.meta\_value

FROM association a, event b

WHERE a.user\_id=1 and a.user\_id=b.user\_id;

Join the chat room

CREATE VIEW web\_joinchat as

SELECT chatroom\_ID

FROM association

WHERE association\_id=’1’

WITH CHECK OPTION;

View chat room messages and users

CREATE VIEW web\_check\_event as

SELECT a.user\_ID,a.user\_level, b.msg\_sendername,b.msg\_date,b.msg\_content

FROM association\_touser a, groupmsg b

WHERE a.chatroom\_ID=1 and a.chatroom\_ID=b. chatroom\_ID and TIMESTAMPDIFF(DAY, NOW(),a)<=3;

About Design security mechanisms

In the website, in addition to the website administrators, all users are integrated into a user (here named test), including querying the community, viewing information, and modifying the description. The user of that database has the SELECT, INSERT, UPDATE, and DELETE permissions for this table, and the rest are not. However, when the database performs an operation, the backend of the website will strictly check whether the user has this permission to perform this operation. At the same time, the backend will check for user input (if any) to prevent malicious attacks on the database from causing damage to the database. After this, if there are special requirements, other permissions will be given to this user.

This user will be given a strong password.

As a website developer and database designer, you will directly use root to modify the schema library designed by the website. The required tables will be created in advance, so that the test user does not have the permission to modify the database structure.

About redundancy

Considering the efficiency of the query and other factors, the database table does not conform to the third normal form, because conforming to the third normal form will result in more queries needing to join multiple tables, which will result in slower query speed. And the more the amount of data, the more obvious the effect is.

For example, the association\_name in the event table is redundant, but if it is canceled, it is necessary to combine two tables in the query.

Or when querying the message record of the chat room, because the display\_name of the message user needs to be displayed, the msm\_sendername field in the groupmsg table involved is added, although the redundancy is increased, but the user table is not required to be connected. .

About Interface design

Because we use the php language to write the website backend and the database is MySQL, and the version after php5.0 provides the mysqli interface to connect to the MySQL database. So we use the mysqli interface to connect to the database

In the php language, the statement that connects to the database is

$conn = new mysqli($hn,$un,$pw,$db);

Where $hn is the database address, $un is the username, $pw is the password, and $db is the schema.