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

关于file organization and index

usermeta, eventmeta, association\_touser表使用hash file organization，因为这些表的主键除了区分元组并没有其他的意义，usermeta和eventmeta最重要的是根据user\_id或者event\_ID来查询所有meta\_key和meta\_value之间的对应关系。因此使用hash file organization可以加快查询效率。

user, association, association\_touser, event, comment, groupmsg表则采用sequential (sort) files，这些表的主键除了区分元组以及作为其他表的外键并没有其他的意义。有些表像user里面存在列名user\_login，由于用户名是不能去相同的，因此user\_login可以用来区分不同的元组，但是为了增加管理员查阅数据库的可读性，因此保留了这些表的ID主键，并且决定将这些表规定为根据表的主键按照升序排序来增加管理员查阅数据库的可读性。如果未来由于数据量大导致查询效率变低。可以采用其他方法改善文件结构，虽然降低了可读性，但是可以提高查询效率

**本来有参考方法，但是个人觉得这里写出来似乎就等于废除了上面的方法。改动比较大，如果页数少可以往里面加类似上面写的那个占了两页半的东西。**

至于索引的话

user：user\_login作为索引，display\_name作为全文索引

usermeta: user\_id作为索引

association：association\_name作为全文索引

association\_touser：association\_ID和user\_ID分别作为索引

event：association\_ID作为索引，event\_name作为全文索引

eventmata: event\_ID作为索引

comment: comment\_association\_ID, comment\_event\_ID分别作为索引

groupmsg: chatroom\_ID 和msg\_date分别作为索引

关于user view

**貌似很多所以只写了一部分**

关于登录方面

CREATE VIEW web\_login as

SELECT ID

FROM user

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

WITH CHECK OPTION;

关于查看用户信息

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;

根据名字查找社团

CREATE VIEW web\_search\_association as

SELECT association\_name

FROM association

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

WITH CHECK OPTION;

根据名字查找活动

CREATE VIEW web\_search\_event as

SELECT event\_name

FROM event

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

WITH CHECK OPTION;

查看社团

CREATE VIEW web\_check\_association as

SELECT association\_name, memberNum, association\_description

FROM association

WHERE association\_id=’1’

WITH CHECK OPTION;

查看活动

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;

参加聊天室

CREATE VIEW web\_joinchat as

SELECT chatroom\_ID

FROM association

WHERE association\_id=’1’

WITH CHECK OPTION;

查看聊天室的消息以及用户

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;

关于Design security mechanisms

网站中，除了网站管理人员，所有用户都是集成在一个用户（这里暂时命名为test）进行操作，包括查询社团，查看信息，修改描述等功能。而那个数据库的用户拥有这个表的SELECT,INSERT,UPDATE,DELETE的权限，其余暂时没有。但是数据库在执行某个操作的时候网站后端会严格检查这个用户是否有这个权限执行这个操作。同时后端将会检查用户输入（如果有的话）防止用户恶意攻击对数据库造成破坏。在这之后如果有特殊要求，会在给这个用户执行其他权限。

这个用户将会被设置高强度的密码。

而作为网站开发人员和数据库设计人员将会直接使用root对网站设计的scheme库进行修改，需要用的表将会被提前创建，这样不需要test用户拥有相关修改数据库结构的权限了。

关于冗余度

考虑到查询效率等因素，因此数据库的表并没有符合第三范式，因为符合第三范式会导致更多的查询需要联接多个表来进行，这样将会导致查询速度变慢。并且数据量越多，影响效果就越明显

比如event的表中的association\_name是多余的，但是如果取消，在查询方面需要联合两个表才能进行。

又或者是在查询聊天室的消息记录的时候，因为需要显示发消息用户的display\_name，因此涉及到的groupmsg表中多了一个msg\_sendername字段，虽然增加了冗余度，但是这样就不需要联接user表。

关于Interface design

因为我们使用的php语言来编写网站后端以及数据库是MySQL，而php5.0之后的版本提供了mysqli接口来连接MySQL数据库。因此我们使用mysqli接口来连接数据库

在php语言中，连接数据库的语句为

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

其中$hn为数据库地址，$un为用户名，$pw为密码，$db为schema