

# 逐梦校友圈

## 数据库设计说明书

组员：郑茂基，许渊博，谢语涵，蔡建斌，陈亮亮，翁炜华，  
邹普凡，黄海翔，党朝媚

逐梦校友圈组 编写

2021. 4. 23

# 目 录

一、 引言 .....	3
1.1 编写目的 .....	3
1.2 背景 .....	3
1.3 定义 .....	3
1.4 设计思路 .....	4
1.5 参考资料 .....	4
二、 外部设计 .....	4
2.1 标识符和状态 .....	4
2.1.1 数据库总体 .....	4
2.1.2 数据库表 .....	5
2.2 使用它的程序 .....	6
2.3 约定 .....	6
2.4 专门指导 .....	6
2.5 支持软件 .....	6
三、 结构设计 .....	6
3.1 概念结构设计 .....	7
3.1.1 数据流图 .....	7
3.1.2 实体属性图 .....	11
3.2 逻辑结构设计 .....	15
3.2.1 数据库逻辑结构图 .....	15
3.2.2 数据库表结构 .....	17
四、 运用设计 .....	23
4.1 权限设计 .....	23
4.1.1 权限设计概述 .....	23
4.1.2 角色清单 .....	23
4.1.3 页面权限 .....	23
4.1.4 操作权限 .....	24
4.1.5 数据权限 .....	24
4.2 安全设计 .....	24
4.2.1 总览 .....	24
4.2.3 设计细节 .....	25
五、 附录 .....	26
5.1 建表语句 .....	26

# 一、 引言

## 1.1 编写目的

本文档为筑梦校友圈团队开发人员关于系统数据库模型设计及相应的权限设计的详细文档，用于指导系统的开发。

具体使用人员有以下几类：系统设计人员、系统开发人员、系统测试人员、系统维护人员。

本数据库设计是根据前期《需求规格说明书》基础之上制定，作为系统开发的根基，数据库设计的好坏影响到整个系统开发的效率，以及系统的灵活性。好的数据库设计能够让方便我们的项目开发，加快我们的开发的进度，严格规定的格式也避免了数据交互时一些不必要的错误。数据库模型必须是能够满足用户的需求，支持所有的用户功能，并且拥有较高的范式，能够支持二次开发。

逐梦校友圈小程序的开发者应该最大程度地围绕这个数据库设计进行开发，依据文档进行项目设计，将它作为整个系统编码的依据。

## 1.2 背景

本次项目是完成的是一个微信小程序，名为“逐梦校友圈”。

随着我们迈入大学校园后，会见到许许多多校园的墙，的确给我们提供了很多我们需要的功能，例如拼车、拼单甚至可以找对象，关注人数较多的墙同时浏览数也都在 5000+，具有很强的信息传播能力。但是同时我们也不得不承认，在琳琅满目的墙面前，我们使用者对于墙的具体功能很不明确。墙的机制是人工审核，每天定点发，就导致了时效性很差。qq 又有着同类型设备只支持一台设备登录，也就需要大量的人力，成为墙的门槛低，墙的种类也就因此繁多起来，同种类型的墙甚至有多，例如福大表白墙我已知的就有 4 个。根据我们之前大家提供的需求分析就可以看到，我们的产品可以提供-一个很好的平台，组局功能可以消除人与人之间的距离感。最后一点就是可以增强我们对校园的归属感。

开发该应用是致力于解决大学生信息搜集，拼团组局，寻求帮助存在困难等问题，节约大学生在各个平台寻找需求信息的时间，也希望可以通过互联网这个窗口，让大家看到校园的生活，可以让大家感受到原来校园中还有其他的什么样的生活的存在，希望可以借此让大家拉近人与人之间的距离，感受一下大学生活更多的温暖。

## 1.3 定义

1、E-R 图：实体关系图

2、MySQL：一种关系型数据库管理系统

3、数据流图：简称 DFD，它从数据传递和加工角度，以图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程，是结构化系统分析方法的主要表达工具及用于表示软件模型的一种图示方法。

## 1.4 设计思路

数据库使用关系型数据库。数据库表结构设计在 ER 图的基础上，结合实体属性的存储方式，为各实体属性赋予相应的数据类型。

此外，为表示 ER 图中实体之间的一对多关系，数据库表中使用新增外键属性的方法对一对多的关系进行约束，例如帖子 `post` 与帖子的评论 `post_comment` 是一对多的关系，在 `post_comment` 中新增 `post_id` 属性与对应帖子 `post` 的 `id` 属性关联，建立外键约束，表示帖子评论所属的帖子，对帖子与帖子评论的一对多关系进行了刻画和约束。ER 图中还出现了实体与自身的一对多约束，同样，在数据库设计时在表中建立了外键与自身关联，例如，一条帖子评论下可能包含有多条评论（楼中楼），设计时新增 `pre_id` 外键关联自身表的 `id` 属性，建立外键，表示帖子评论所属的上一级评论，对帖子评论与自身的一对多的关系进行了刻画和约束。

ER 图中出现的多对多关系，考虑到要满足范式减少数据冗余，关系型数据库较难表示，与要对多对多关系进行转换（实际在 ER 图中已经转换），新建一个额外的表，存储多对多关联，将多对多关联分解为两个一对多关联，例如用户和组局之间的多对多关系，新增组局人员表对多对多进行转换，变为两个一对多的关系。

以上两种处理思路解决了系统的一对多和多对多约束。

为了符合一些行业规范，在各个表中新增加了 `gmt_create`, `gmt_modified` 对记录创建时间和最近一次修改时间进行了存储。还增加了 `deleted` 属性用于实现逻辑删除功能，方便后台管理查看记录。

## 1.5 参考资料

- 1、[阿里巴巴 Mysql 数据库规约](#)
- 2、[数据库设计说明书编写规范\(国标\)](#)
- 3、数据库系统概论（第四版） 王珊，萨师煊编著 高等教育出版社
- 4、构建之法（第三版） 邹欣编著 人民邮电出版社

# 二、 外部设计

## 2.1 标识符和状态

### 2.1.1 数据库总体

数据库软件的名称：MySQL  
数据库的名称：alumni\_circle  
用户名：root  
密码：123456  
权限：root 权限

有效时间：只在开发阶段

### 2.1.2 数据库表

在数据库设计中，我们将数据库分为用户、贴文和组局三部分分开设计，之后再行整合。这样设计更容易把握细节，且成员之间分工更加明确，效率更高。

表名	标识符	描述信息	状态
用户表	user	存放用户的基础信息	开发使用中
管理员表	admin	存放管理员的基础信息	开发使用中
树洞表	tree_hole	存放树洞模块信息	开发使用中
私聊表	private_chat	记录私聊信息	开发使用中
黑名单表	black_list	记录黑名单信息	开发使用中
组局表	party	记录组局信息	开发使用中
组局类别表	party_type	记录组局类别	开发使用中
组局评论表	party_comment	记录组局的评论记录	开发使用中
组局参与人员表	party_participants	记录组局的参与人员	开发使用中
贴文表	post	记录贴文信息	开发使用中
打赏表	post_reward	打赏人品值记录表	开发使用中
贴文类别表	post_type	记录贴文的类别	开发使用中
点赞表	post_like	记录贴文的点赞情况	开发使用中
举报表	report	记录贴文的举报信息	开发使用中
贴文评论表	post_comment	记录贴文的评论信息	开发使用中
贴文关注表	post_eye_on	记录贴文的关注情况	开发使用中

## 2.2 使用它的程序

- 1、“逐梦演艺圈”小程序 V1.0
- 2、“逐梦演艺圈”后台管理网页 V1.0

## 2.3 约定

- 1、表名、字段名必须使用小写字母或数字；禁止出现数字开头，禁止两个下划线中间只出现数字。
- 2、禁用保留字，如 desc、range、match、delayed 等，请参考 MySQL 官方保留字。
- 3、表必备三字段：id, gmt\_create, gmt\_modified。
- 4、表的命名最好是加上 “业务名称\_表的作用”。
- 5、字符集采用 “UTF-8”
- 6、不允许存储明文密码
- 7、如果修改字段含义或对字段表示的状态追加时，需要及时更新字段注释。

## 2.4 专门指导

- 1、编码人员在与数据库交互时，应注意表中的 id 字段是否为自增类型，规范使用，杜绝出错。
- 2、需留意字段是否可为空，是否有设置默认值。
- 3、数据库编码集为：“UTF-8”，请注意数据字符集规范操作。
- 4、本系统使用的 mysql 版本为 mysql5.0。
- 5、数据格式见数据库规格表。
- 6、使用过程中遵循 2.3 约定。

## 2.5 支持软件

操作系统：Linux/windows  
数据库系统：MySQL  
命令行工具：mysql

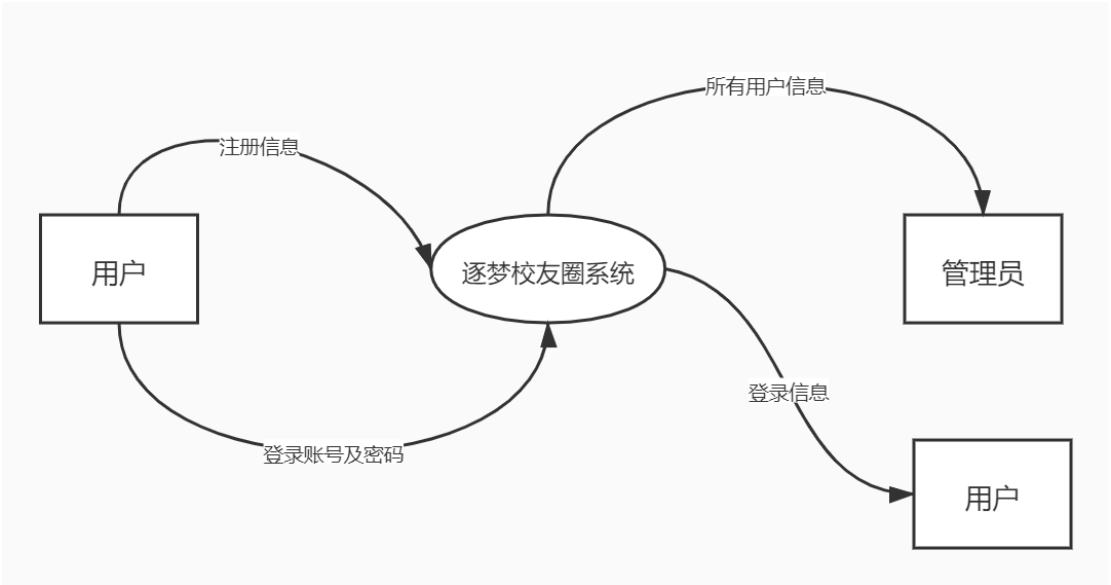
# 三、 结构设计

### 3.1 概念结构设计

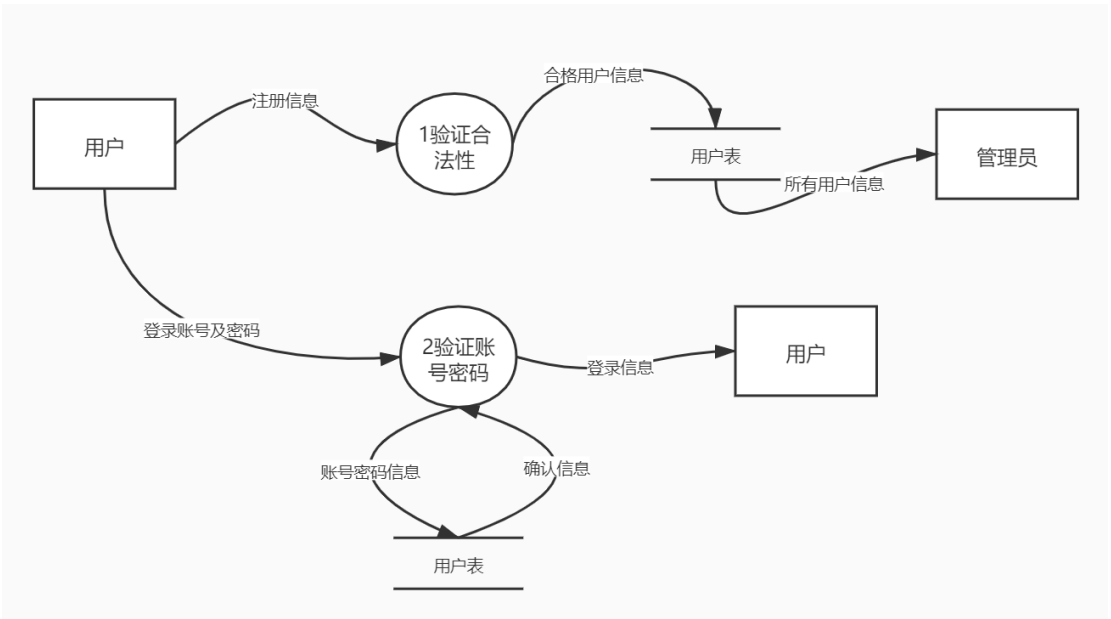
#### 3.1.1 数据流图

##### 3.1.1.1 授权验证部分

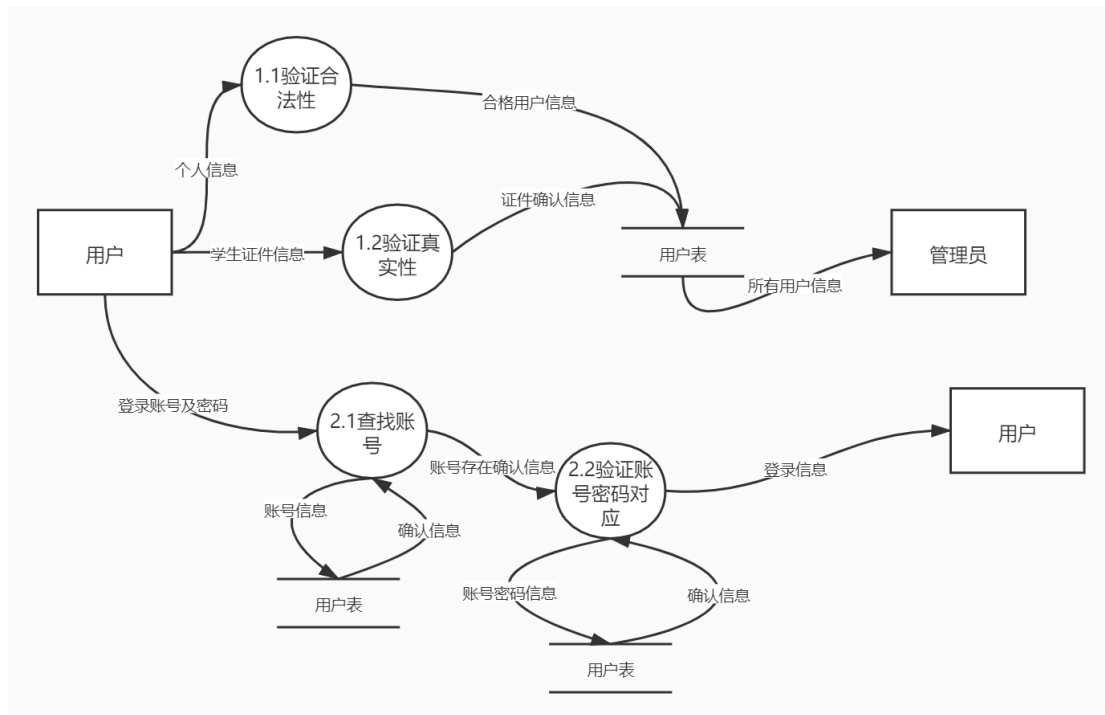
验证顶层图



授权验证第 0 层数据流图

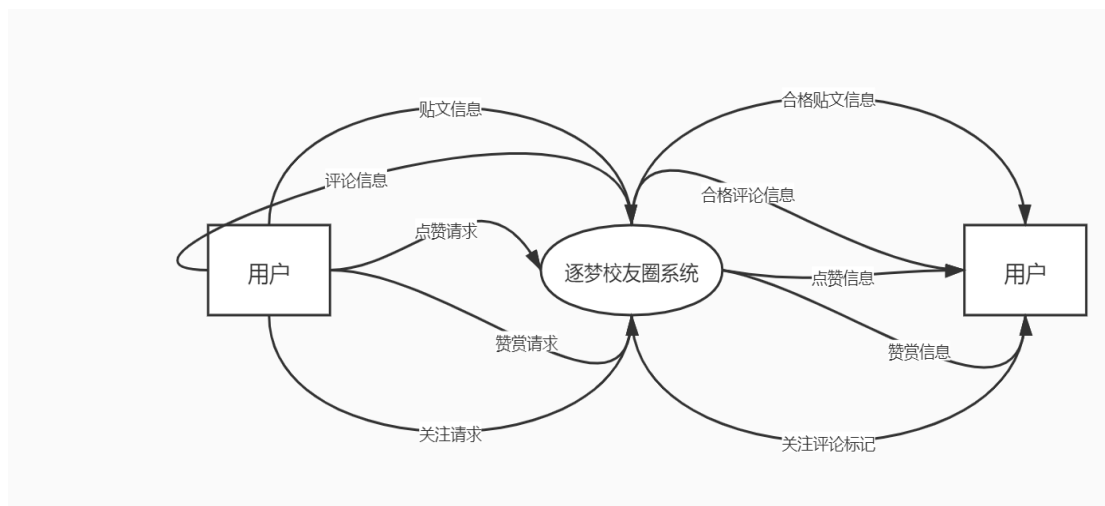


授权验证第 1 层数据流图



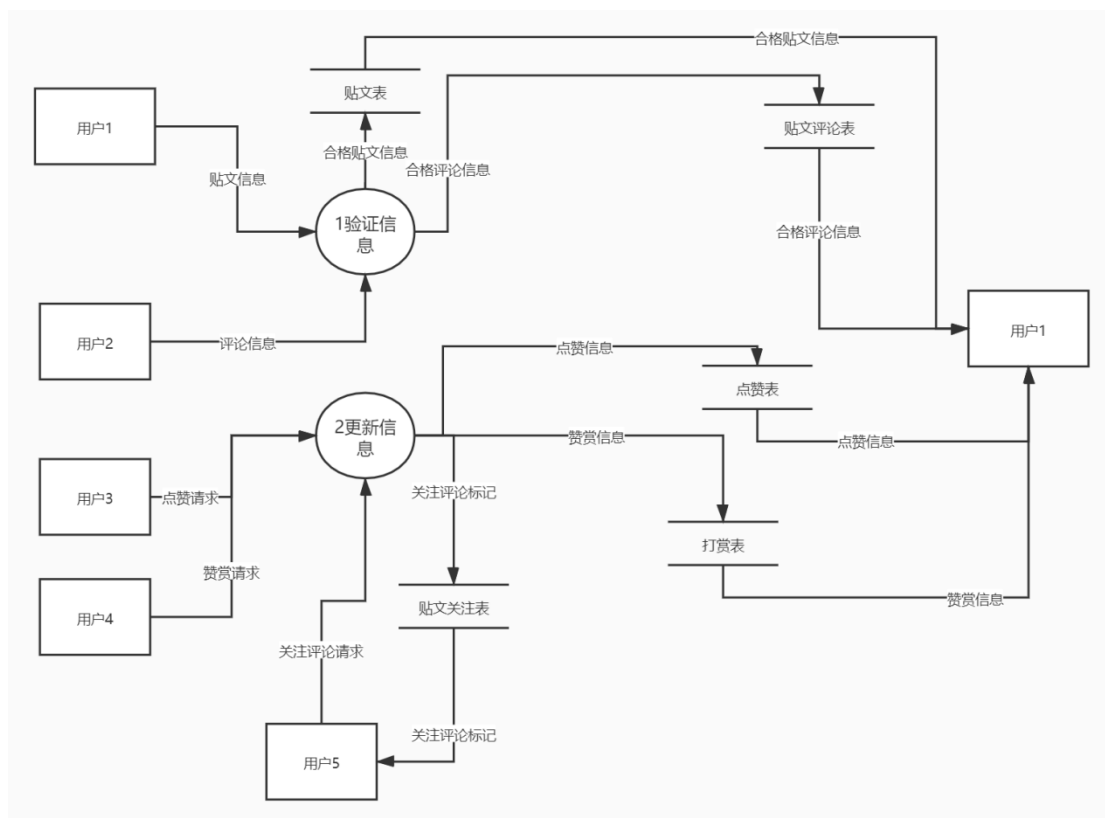
### 3.1.1.2 发帖部分

发帖顶层图

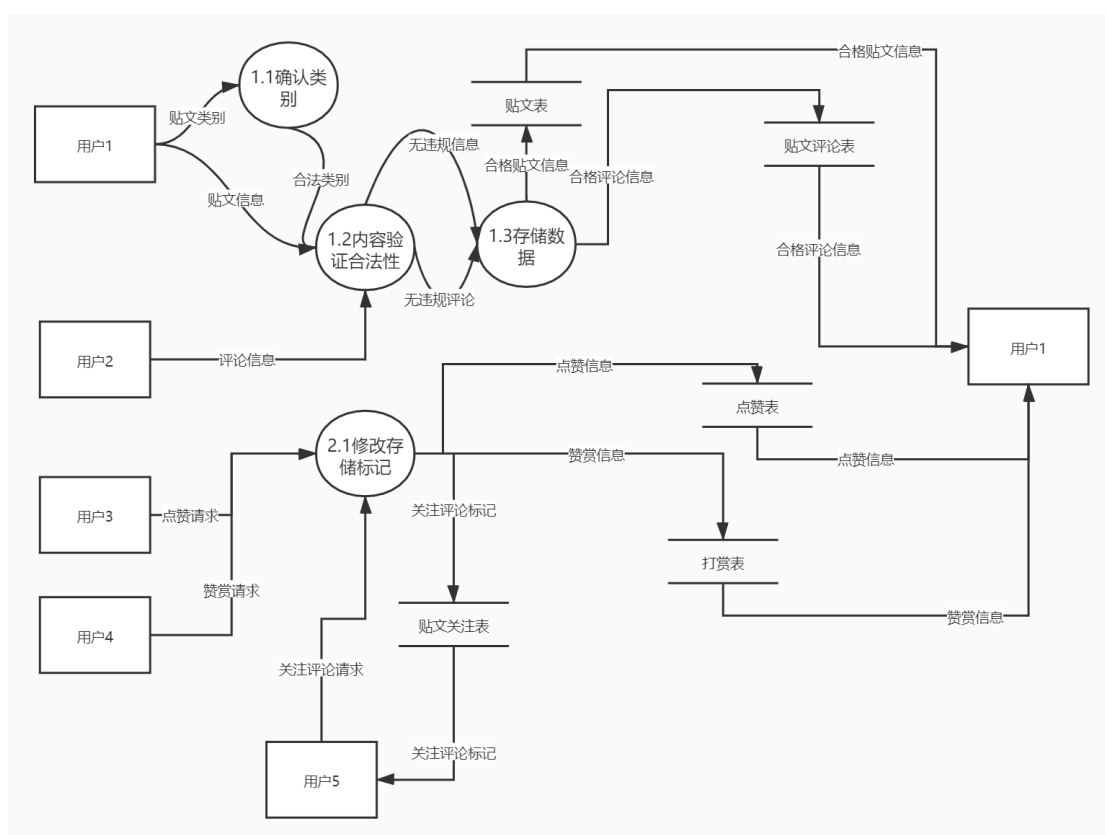


发帖第 0 层数据流图



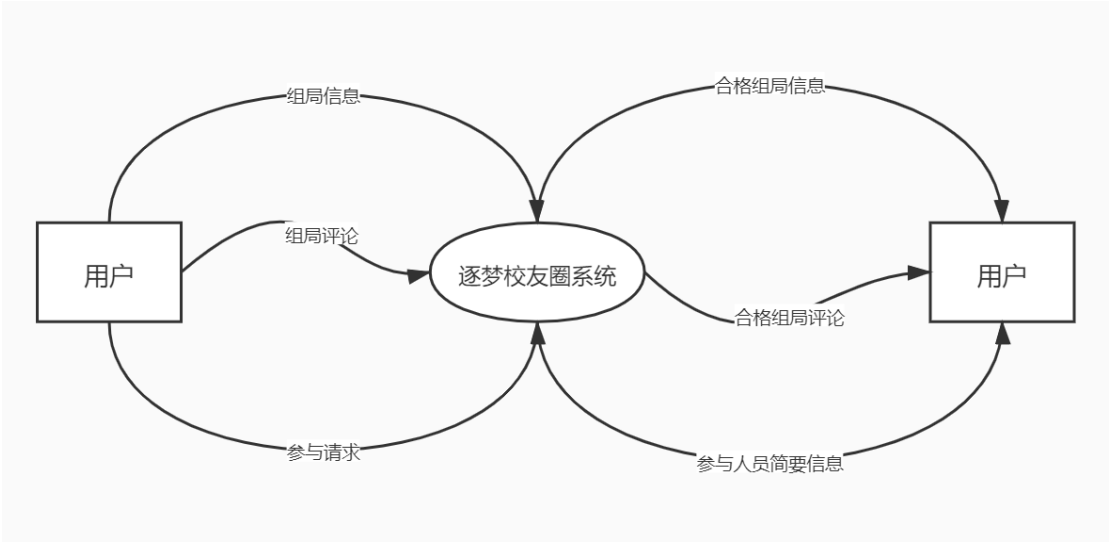


发帖第 1 层数据流图

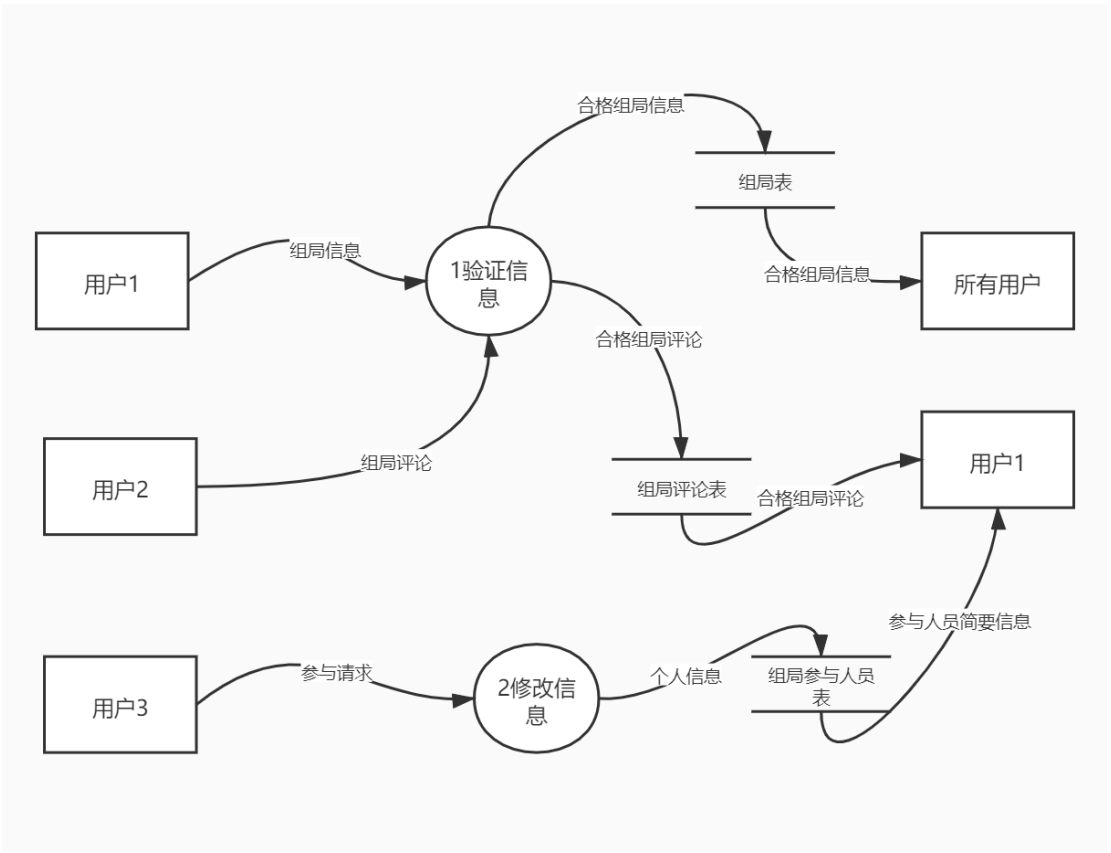


3.1.1.3 组局部分

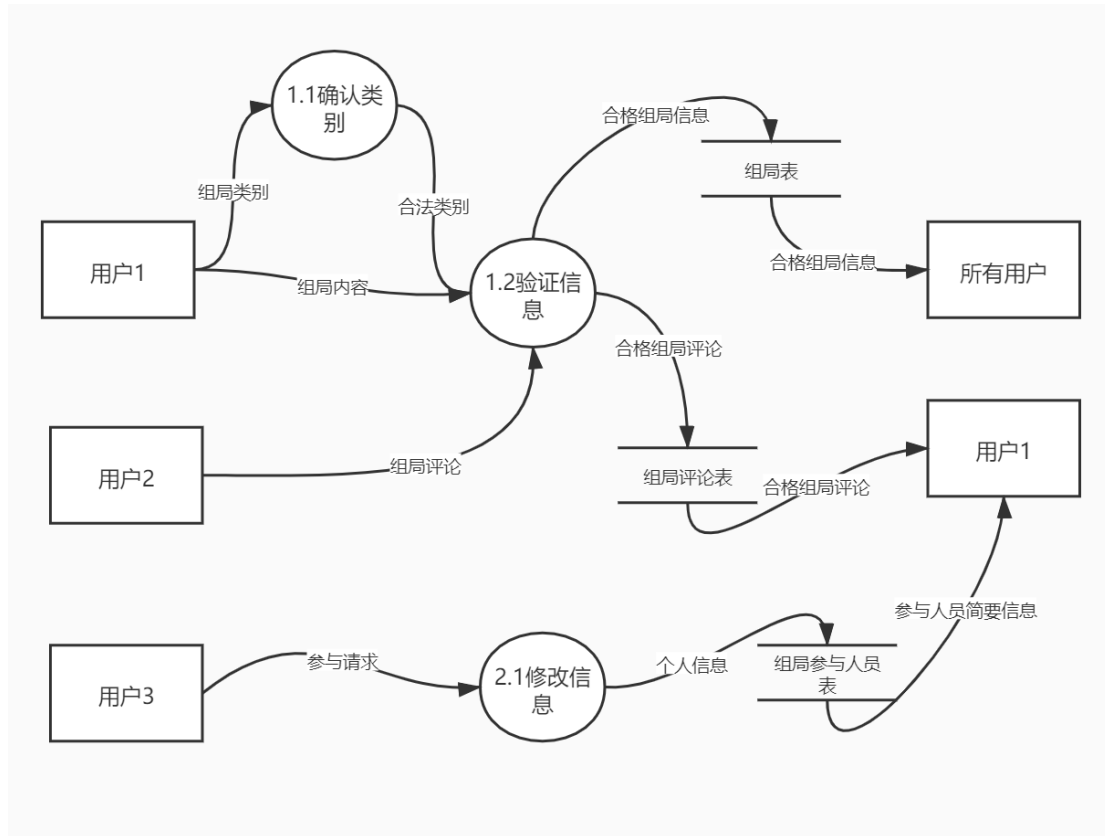
组局顶层图



组局第 0 层数据流图



组局第 1 层数据流图

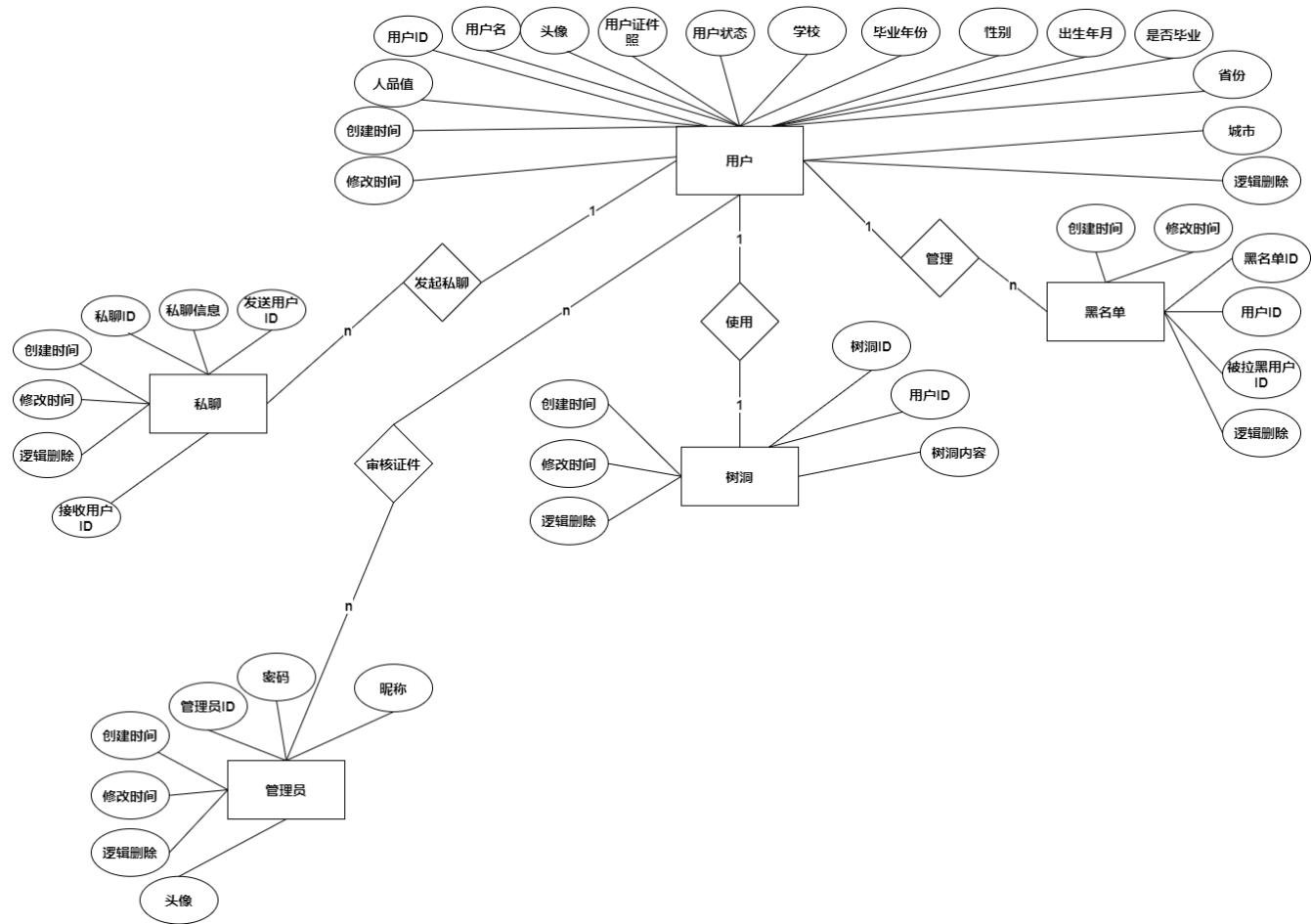


### 3.1.2 实体属性图

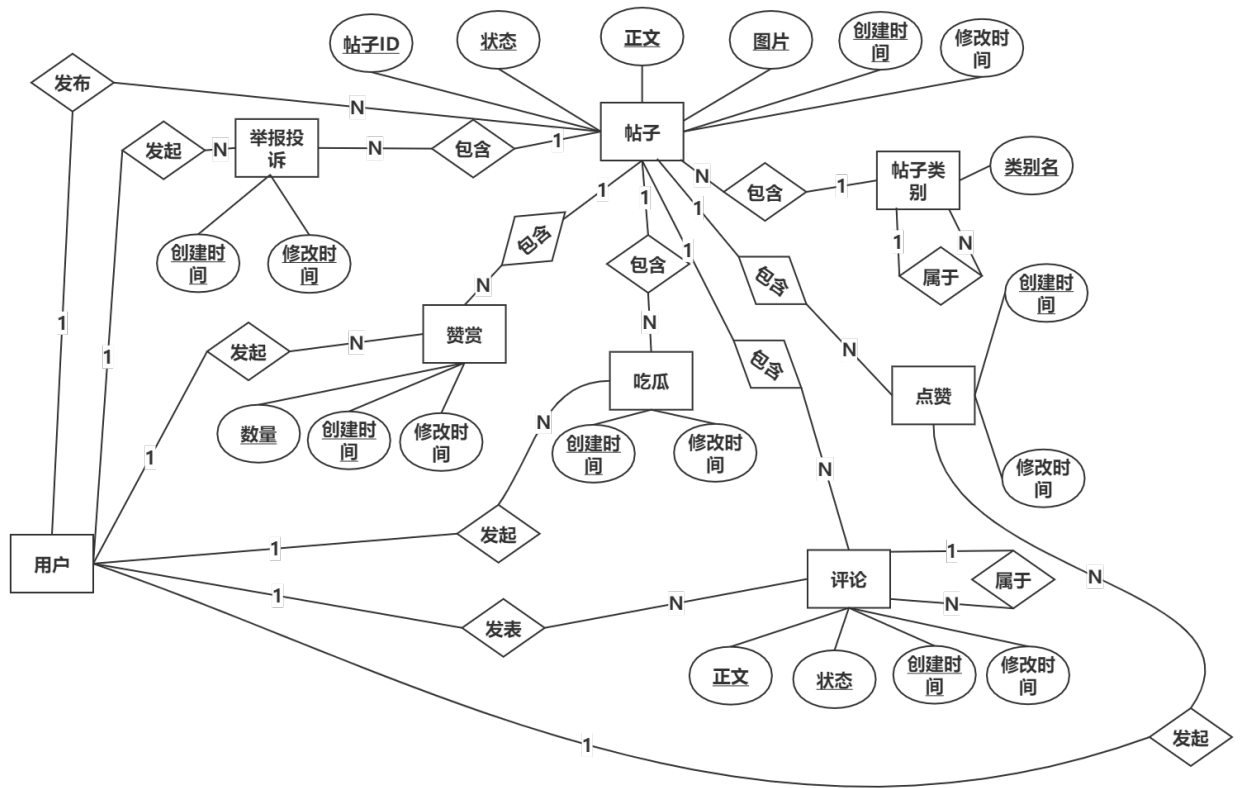
ER 图设计思路：

从用户角度出发，寻找用户拥有的属性，寻找用户与其他类（帖文、组局等）的关联，确定之间的关系：1 对多、1 对 1 等，再补充其他类的拥有的属性。

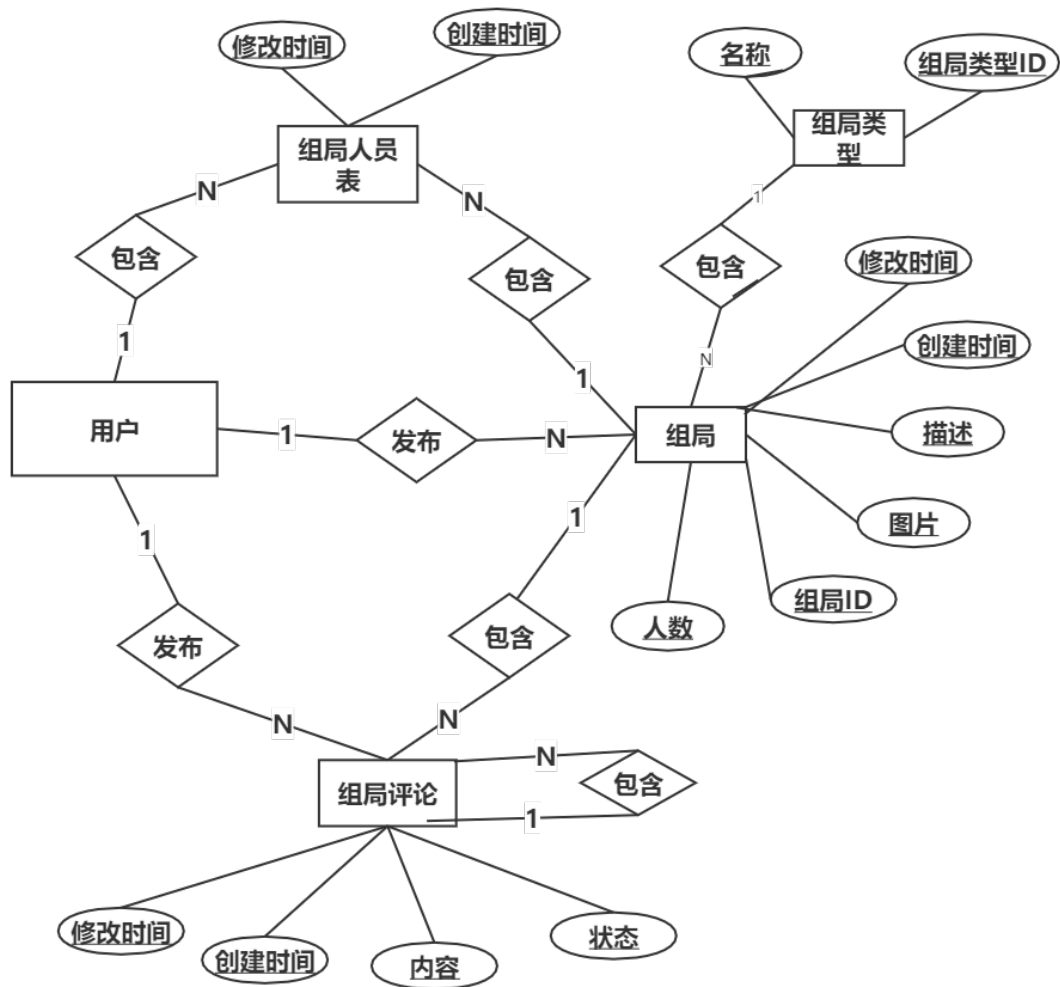
### 3.1.2.1 用户(user)和管理员(admin)部分



### 3.1.2.2 贴文（post）部分



### 3.1.2.3 贴文（post）部分



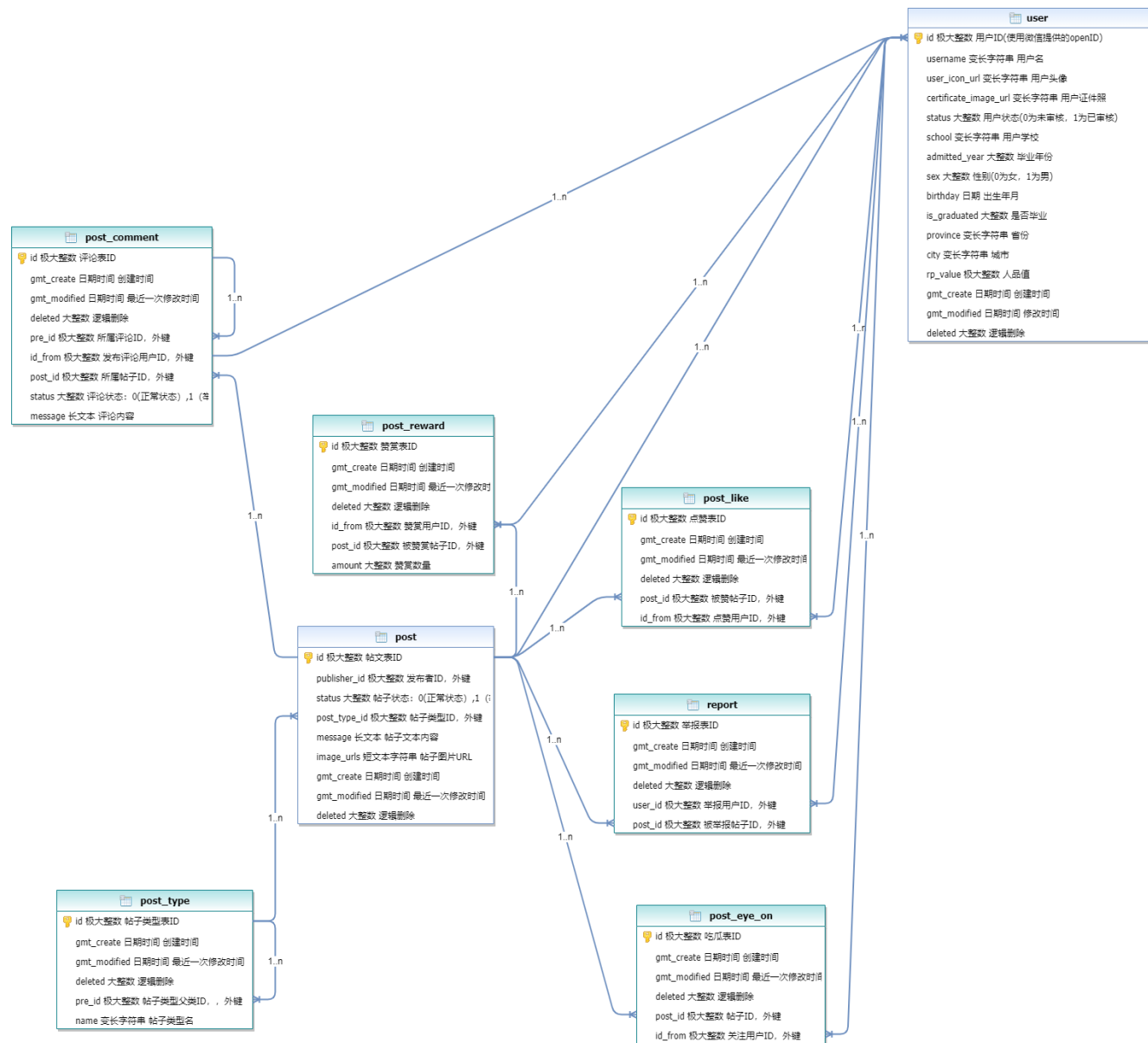
## 3.2 逻辑结构设计

### 3.2.1 数据库逻辑结构图

#### 3.2.1.1 用户（user）和管理员（admin）部分

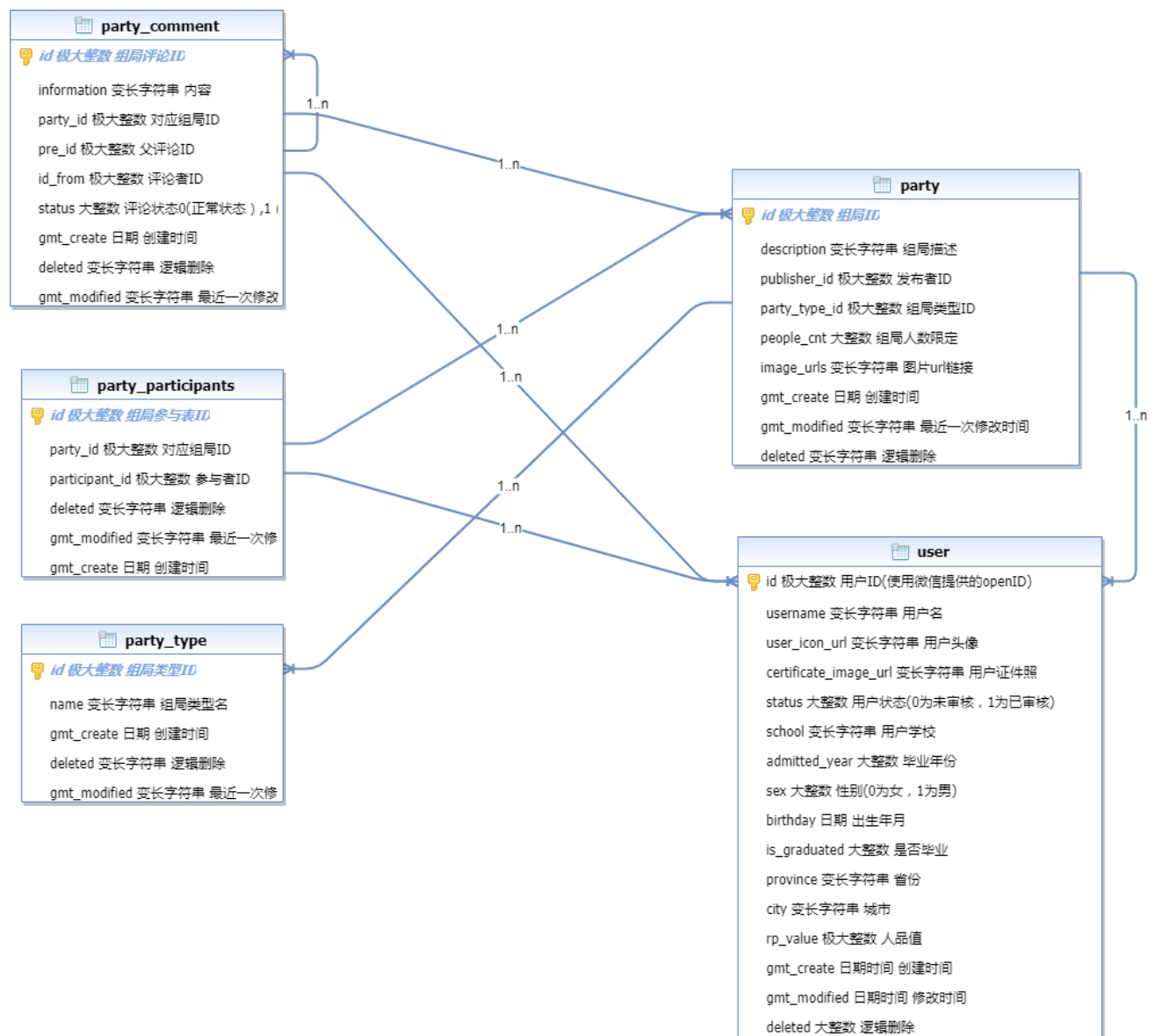


### 3.2.1.2 贴文（post）部分



### 3.2.1.3 组局（party）部分





## 3.2.2 数据库表结构

根据阿里巴巴 java 开发手册-数据库部分，每个表都有 id、gmt\_create、gmt\_modified 字段，以及增加了用于逻辑删除的 deleted 字段。

### 3.2.2.1 用户表 User

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	🔑 1
username	varchar	255	0	<input type="checkbox"/>	
user_icon_url	varchar	255	0	<input type="checkbox"/>	
certificate_image_url	varchar	255	0	<input type="checkbox"/>	
status	int	11	0	<input type="checkbox"/>	
school	varchar	255	0	<input type="checkbox"/>	
admitted_year	int	11	0	<input type="checkbox"/>	
sex	int	11	0	<input type="checkbox"/>	
birthday	date	0	0	<input type="checkbox"/>	
is_graduated	int	11	0	<input type="checkbox"/>	
province	varchar	255	0	<input type="checkbox"/>	
city	varchar	255	0	<input type="checkbox"/>	
rp_value	bigint	20	0	<input type="checkbox"/>	
gmt_create	datetime	0	0	<input type="checkbox"/>	
gmt_modified	datetime	0	0	<input type="checkbox"/>	
deleted	int	11	0	<input type="checkbox"/>	

### 3.2.2.2 管理员表 Admin

名	类型	长度	小数点	不是 null	
gmt_create	datetime	0	0	<input type="checkbox"/>	
gmt_modified	datetime	0	0	<input type="checkbox"/>	
deleted	int	11	0	<input type="checkbox"/>	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	🔑 1
password	varchar	255	0	<input type="checkbox"/>	
nickname	varchar	255	0	<input type="checkbox"/>	
admin_icon_url	varchar	255	0	<input type="checkbox"/>	

### 3.2.2.3 黑名单表 Black\_list

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	🔑 1
user_id	bigint	20	0	<input type="checkbox"/>	
be_user_id	bigint	20	0	<input type="checkbox"/>	
gmt_create	datetime	0	0	<input type="checkbox"/>	
gmt_modified	datetime	0	0	<input type="checkbox"/>	
deleted	int	11	0	<input type="checkbox"/>	

### 3.2.2.4 组局表 Party

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	🔑 1
description	varchar	50	0	<input type="checkbox"/>	
publisher_id	bigint	20	0	<input type="checkbox"/>	
party_type_id	bigint	20	0	<input type="checkbox"/>	
people_cnt	int	50	0	<input type="checkbox"/>	
image_urls	varchar	50	0	<input type="checkbox"/>	
gmt_create	datetime	0	0	<input type="checkbox"/>	
gmt_modified	datetime	0	0	<input type="checkbox"/>	
deleted	int	11	0	<input type="checkbox"/>	

### 3.2.2.5 组局评论表 Party\_comment

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	🔑 1
information	varchar	50	0	<input type="checkbox"/>	
party_id	bigint	20	0	<input type="checkbox"/>	
pre_id	bigint	20	0	<input type="checkbox"/>	
id_from	bigint	20	0	<input type="checkbox"/>	
status	int	11	0	<input type="checkbox"/>	
gmt_create	datetime	0	0	<input type="checkbox"/>	
deleted	int	11	0	<input type="checkbox"/>	
gmt_modified	datetime	0	0	<input type="checkbox"/>	

### 3.2.2.6 组局成员表 Party\_participants

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	🔑 1
party_id	bigint	20	0	<input type="checkbox"/>	
participant_id	bigint	20	0	<input type="checkbox"/>	
deleted	int	11	0	<input type="checkbox"/>	
gmt_modified	datetime	0	0	<input type="checkbox"/>	
gmt_create	datetime	0	0	<input type="checkbox"/>	

### 3.2.2.7 组局类型表 Party\_type

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	🔑 1
name	varchar	10	0	<input type="checkbox"/>	
gmt_create	datetime	0	0	<input type="checkbox"/>	
deleted	int	11	0	<input type="checkbox"/>	
gmt_modified	datetime	0	0	<input type="checkbox"/>	

### 3.2.2.8 帖文表 Post

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	🔑 1
publisher_id	bigint	20	0	<input type="checkbox"/>	
status	int	11	0	<input checked="" type="checkbox"/>	
post_type_id	bigint	20	0	<input checked="" type="checkbox"/>	
message	text	0	0	<input checked="" type="checkbox"/>	
image_urls	tinytext	0	0	<input checked="" type="checkbox"/>	
gmt_create	datetime	0	0	<input checked="" type="checkbox"/>	
gmt_modified	datetime	0	0	<input checked="" type="checkbox"/>	
deleted	int	11	0	<input checked="" type="checkbox"/>	

### 3.2.2.9 帖文评论表 Post\_comment

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	🔑 1
gmt_create	datetime	0	0	<input checked="" type="checkbox"/>	
gmt_modified	datetime	0	0	<input checked="" type="checkbox"/>	
deleted	int	11	0	<input checked="" type="checkbox"/>	
pre_id	bigint	20	0	<input checked="" type="checkbox"/>	
post_id	bigint	20	0	<input checked="" type="checkbox"/>	
id_from	bigint	20	0	<input checked="" type="checkbox"/>	
status	int	10	0	<input checked="" type="checkbox"/>	
message	text	0	0	<input type="checkbox"/>	


3.2.2.10 帖文审核表 Post\_eye\_on

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	 1
gmt_create	datetime	0	0	<input checked="" type="checkbox"/>	
gmt_modified	datetime	0	0	<input checked="" type="checkbox"/>	
deleted	int	11	0	<input checked="" type="checkbox"/>	
post_id	bigint	20	0	<input checked="" type="checkbox"/>	
id_from	bigint	20	0	<input checked="" type="checkbox"/>	

3.2.2.11 帖文收藏表 Post\_like

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	 1
gmt_create	datetime	0	0	<input checked="" type="checkbox"/>	
gmt_modified	datetime	0	0	<input checked="" type="checkbox"/>	
deleted	int	11	0	<input checked="" type="checkbox"/>	
post_id	bigint	20	0	<input checked="" type="checkbox"/>	
id_from	bigint	20	0	<input checked="" type="checkbox"/>	

3.2.2.12 帖文赞赏表 Post\_reward

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	 1
gmt_create	datetime	0	0	<input checked="" type="checkbox"/>	
gmt_modified	datetime	0	0	<input checked="" type="checkbox"/>	
deleted	int	11	0	<input checked="" type="checkbox"/>	
post_id	bigint	20	0	<input checked="" type="checkbox"/>	
id_from	bigint	20	0	<input checked="" type="checkbox"/>	
amount	int	10	0	<input checked="" type="checkbox"/>	

### 3.2.2.13 帖文类型表 Post\_type

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	 1
gmt_create	datetime	0	0	<input checked="" type="checkbox"/>	
gmt_modified	datetime	0	0	<input checked="" type="checkbox"/>	
deleted	int	11	0	<input checked="" type="checkbox"/>	
pre_id	bigint	20	0	<input checked="" type="checkbox"/>	
name	varchar	255	0	<input checked="" type="checkbox"/>	

### 3.2.2.14 私聊表 Private\_chat

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	 1
message	text	0	0	<input type="checkbox"/>	
from_id	bigint	20	0	<input type="checkbox"/>	
to_id	bigint	20	0	<input type="checkbox"/>	
gmt_create	datetime	0	0	<input type="checkbox"/>	
gmt_modified	datetime	0	0	<input type="checkbox"/>	
deleted	int	11	0	<input type="checkbox"/>	

### 3.2.2.15 举报表 Report

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	 1
gmt_create	datetime	0	0	<input checked="" type="checkbox"/>	
gmt_modified	datetime	0	0	<input checked="" type="checkbox"/>	
deleted	int	11	0	<input checked="" type="checkbox"/>	
user_id	bigint	20	0	<input checked="" type="checkbox"/>	
post_id	bigint	20	0	<input checked="" type="checkbox"/>	

### 3.2.2.16 树洞表 Tree\_hole

名	类型	长度	小数点	不是 null	
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	🔑 1
from_id	bigint	20	0	<input type="checkbox"/>	
message	text	0	0	<input type="checkbox"/>	
gmt_create	datetime	0	0	<input type="checkbox"/>	
gmt_modified	datetime	0	0	<input type="checkbox"/>	
deleted	int	11	0	<input type="checkbox"/>	

## 四、 运用设计

### 4.1 权限设计

#### 4.1.1 权限设计概述

权限管理是所有后台系统的都会涉及的一个重要组成部分,主要目的是对不同的人访问资源进行权限的控制,避免因权限控制缺失或操作不当引发的风险问题,如操作错误,隐私数据泄露等问题。迄今为止最为普及的权限设计模型是 RBAC 模型,基于角色的访问控制。由于我们的记账系统是为已注册登录的用户提供服务,故用户对应的角色即为用户本身,只要有登录到我们的系统,就能使用我们软件所包含的所有服务,即能行使用户角色本身所对应到的所有权限。权限是用户可以访问的资源,包括页面权限,操作权限,数据权限。

#### 4.1.2 角色清单

本次共有以下角色:

- 1、游客用户
- 2、普通用户
- 3、超级管理员

#### 4.1.3 页面权限

游客用户: 认证页面, 校友圈页面, 个人页面

普通用户：认证页面，校友圈页面，个人页面，消息页面，发帖页面，拼局页面

超级管理员：认证页面，校友圈页面，个人页面，消息页面，发帖页面，拼局页面，应用后台页面

#### 4.1.4 操作权限

游客用户：修改个人资料，查看校友圈内容，参与个人认证

普通用户：修改个人资料，查看校友圈内容，参与个人认证，查看消息，发送帖子，发起拼局，参与签到，

超级管理员：拥有最高权限

#### 4.1.5 数据权限

游客用户：能查看并修改自己的基础信息，能查看贴文数据，贴文的点赞数据，贴文的评论信息，贴文的关注情况

普通用户：能查看并修改自己的基础信息，自己的贴文数据，自己的私聊数据，自己的黑名单数据，能删除自己贴文的评论信息，能查看贴文数据，贴文的点赞数据，贴文的评论信息，贴文的关注情况，树洞的数据，组局的信息，组局的类别，组局的评论记录，组局的参与人员

校管理员：可以查看并修改本校所有用户的树洞，组局，贴文，举报，评论数据

超级管理员：可以访问一切数据

### 4.2 安全设计

#### 4.2.1 总览

不同需求对安全产生不同的定义，安全对我们来说就是数据安全，即数据不被窃取，数据不被篡改，数据不被伪造。安全设计的核心是信任域的划分，而基于安全域的原子权限则是数据的坚实保护。原子权限正是这次安全设计的核心思路。



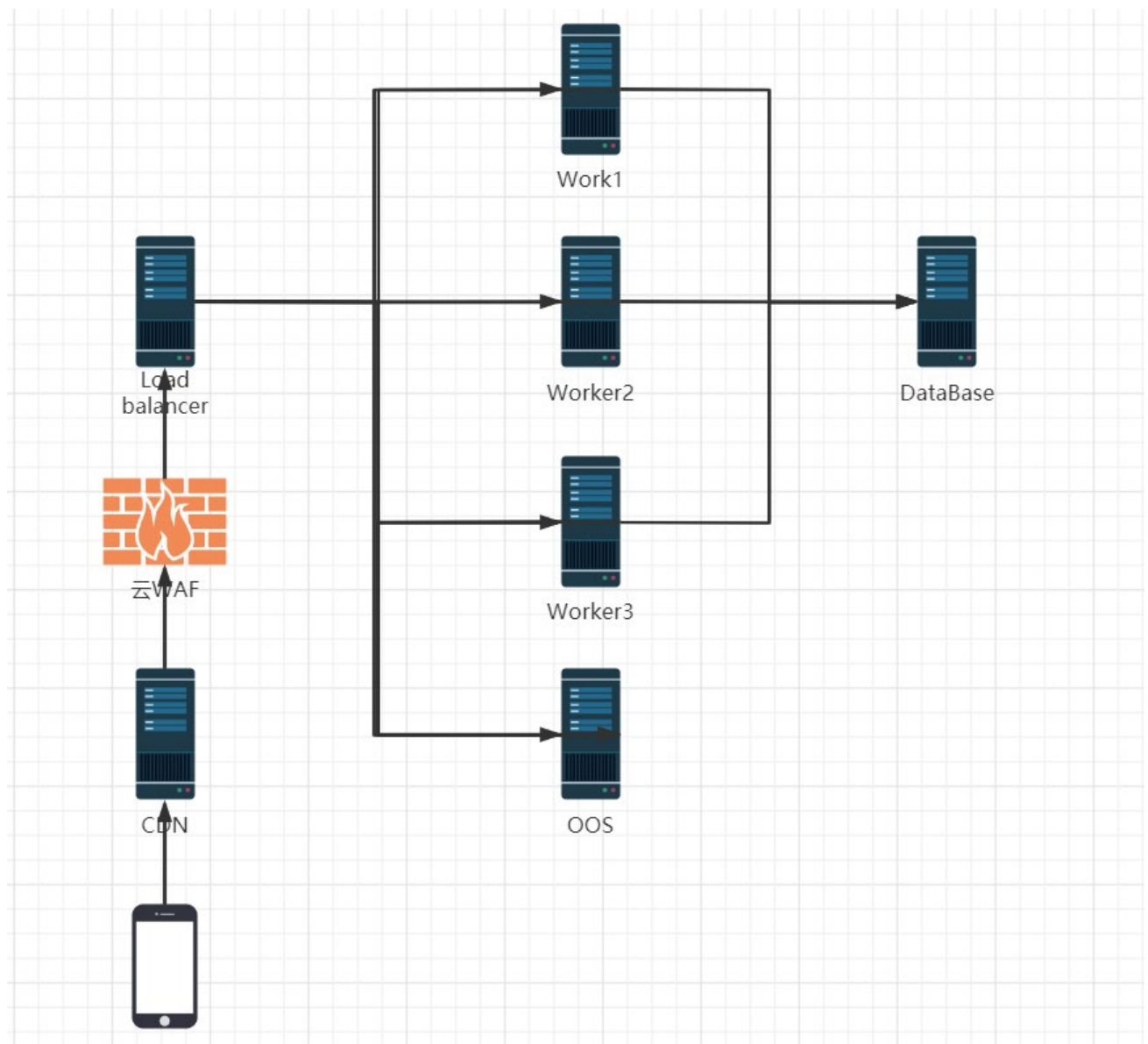
### 4.2.3 设计细节

1、密码等关键敏感信息经过 md5+salt 来保存在数据库中，对于数据较大且需要能还原的数据使用 AES 加密后存储在数据库

2、严格限制权限

3、前端对敏感数据比如密码使用 ECDH 密钥交换+AES 加密

4、数据库与代码服务器硬件隔离



## 五、附录

### 5.1 建表语句

```
CREATE TABLE IF NOT EXISTS post (
    id BIGINT AUTO_INCREMENT PRIMARY KEY comment '帖文 ID',
    publisher_id BIGINT comment '发布者 ID，外键',
    status INT NOT NULL comment '帖子状态: 0(正常状态), 1(举报过多被挂起), 2(已被删除)',
    post_type_id BIGINT NOT NULL comment '帖子类型 ID，外键',
    message TEXT NOT NULL comment '帖子文本内容',
    image_urls TINYTEXT NOT NULL comment '帖子图片 URL',
    gmt_create DATETIME NOT NULL default CURRENT_TIMESTAMP comment '创建时间',
    gmt_modified DATETIME NOT NULL default CURRENT_TIMESTAMP comment '最近一次修改时间',
    deleted INT NOT NULL default 0 comment '逻辑删除'
);
```

```
CREATE TABLE IF NOT EXISTS post_comment (
    id BIGINT AUTO_INCREMENT PRIMARY KEY comment 'ID',
    gmt_create DATETIME NOT NULL default CURRENT_TIMESTAMP comment '创建时间',
    gmt_modified DATETIME NOT NULL default CURRENT_TIMESTAMP comment '最近一次修改时间',
    deleted INT NOT NULL default 0 comment '逻辑删除',
    pre_id BIGINT NOT NULL comment '所属评论 ID，外键',
    post_id BIGINT NOT NULL comment '所属帖子 ID，外键',
    id_from BIGINT NOT NULL comment '发布评论用户 ID，外键',
    status INT UNSIGNED NOT NULL comment '评论状态: 0(正常状态), 1(举报过多被挂起), 2(已被删除)'
);
```

```
CREATE TABLE IF NOT EXISTS report (
    id BIGINT AUTO_INCREMENT PRIMARY KEY comment 'ID',
    gmt_create DATETIME NOT NULL default CURRENT_TIMESTAMP comment '创建时间',
    gmt_modified DATETIME NOT NULL default CURRENT_TIMESTAMP comment '最近一次修改时间',
    deleted INT NOT NULL default 0 comment '逻辑删除',
    user_id BIGINT NOT NULL comment '举报用户 ID，外键',
    post_id BIGINT NOT NULL comment '被举报帖子 ID，外键'
);
```

```
CREATE TABLE IF NOT EXISTS post_type (
    id BIGINT AUTO_INCREMENT PRIMARY KEY comment 'ID',
    gmt_create DATETIME NOT NULL default CURRENT_TIMESTAMP comment '创建时间',
```

```

        gmt_modified DATETIME NOT NULL default CURRENT_TIMESTAMP comment '最近一次修改时间',
        deleted INT NOT NULL default 0 comment '逻辑删除',
        pre_id BIGINT NOT NULL comment '帖子类型父类 ID, 外键',
        name VARCHAR(255) NOT NULL comment '帖子类型名'
    );

```

```

CREATE TABLE IF NOT EXISTS post_like (id BIGINT AUTO_INCREMENT PRIMARY KEY comment 'ID',
    gmt_create DATETIME NOT NULL default CURRENT_TIMESTAMP comment '创建时间',
    gmt_modified DATETIME NOT NULL default CURRENT_TIMESTAMP comment '最近一次修改时间',
    deleted INT NOT NULL default 0 comment '逻辑删除',
    post_id BIGINT NOT NULL comment '被赞帖子 ID, 外键',
    id_from BIGINT NOT NULL comment '点赞用户 ID, 外键'
);

```

```

CREATE TABLE IF NOT EXISTS post_reward ( id BIGINT AUTO_INCREMENT PRIMARY KEY comment 'ID',
    gmt_create DATETIME NOT NULL default CURRENT_TIMESTAMP comment '创建时间',
    gmt_modified DATETIME NOT NULL default CURRENT_TIMESTAMP comment '最近一次修改时间',
    deleted INT NOT NULL default 0 comment '逻辑删除',
    post_id BIGINT NOT NULL comment '被赞赏帖子 ID, 外键',
    id_from BIGINT NOT NULL comment '赞赏用户 ID, 外键',
    amount INT UNSIGNED NOT NULL comment '赞赏数量'
);

```

```

CREATE TABLE IF NOT EXISTS post_eye_on ( id BIGINT AUTO_INCREMENT PRIMARY KEY comment 'ID',
    gmt_create DATETIME NOT NULL default CURRENT_TIMESTAMP comment '创建时间',
    gmt_modified DATETIME NOT NULL default CURRENT_TIMESTAMP comment '最近一次修改时间',
    deleted INT NOT NULL default 0 comment '逻辑删除',
    post_id BIGINT NOT NULL comment '帖子 ID, 外键',
    id_from BIGINT NOT NULL comment '关注用户 ID, 外键'
);

```

```

CREATE TABLE IF NOT EXISTS user ( id BIGINT PRIMARY KEY comment '用户 ID(使用微信提供的 openID)',
    username VARCHAR(255) comment '用户名',
    user_icon_url VARCHAR(255) comment '用户头像',
    certificate_image_url VARCHAR(255) comment '用户证件照',
    status INT default 0 comment '用户状态(0 为未审核, 1 为已审核)',
    school VARCHAR(255) comment '用户学校',

```

```

    admitted_year INT comment '毕业年份',
    sex INT comment '性别(0 为女, 1 为男)',
    birthday DATE comment '出生年月',
    is_graduated INT comment '是否毕业',
    province VARCHAR(255) comment '省份',
    city VARCHAR(255) comment '城市',
    rp_value BIGINT default 0 comment '人品值',
    gmt_create DATETIME default CURRENT_TIMESTAMP comment '创建时间',
    gmt_modified DATETIME default CURRENT_TIMESTAMP comment '修改时间',
    deleted INT default 0 comment '逻辑删除'
);

CREATE TABLE IF NOT EXISTS black_list ( id BIGINT AUTO_INCREMENT PRIMARY KEY
comment '黑名单 ID',
    user_id BIGINT comment '用户 ID',
    be_user_id BIGINT comment '被拉黑用户 ID',
    gmt_create DATETIME default CURRENT_TIMESTAMP comment '创建时间',
    gmt_modified DATETIME default CURRENT_TIMESTAMP comment '修改时间',
    deleted INT default 0 comment '逻辑删除'
);

CREATE TABLE IF NOT EXISTS private_chat ( id BIGINT AUTO_INCREMENT PRIMARY KEY
comment '私聊 ID',
    message TEXT comment '私聊信息',
    from_id BIGINT comment '发送用户 ID',
    to_id BIGINT comment '接收用户 ID',
    gmt_create DATETIME default CURRENT_TIMESTAMP comment '创建时间',
    gmt_modified DATETIME default CURRENT_TIMESTAMP comment '修改时间',
    deleted INT default 0 comment '逻辑删除'
);

CREATE TABLE IF NOT EXISTS admin ( gmt_create DATETIME default CURRENT_TIMESTAMP
comment '创建时间',
    gmt_modified DATETIME default CURRENT_TIMESTAMP comment '修改时间',
    deleted INT default 0 comment '逻辑删除',
    id BIGINT AUTO_INCREMENT PRIMARY KEY comment '管理员 ID',
    password VARCHAR(255) comment '密码',
    nickname VARCHAR(255) comment '昵称',
    admin_icon_url VARCHAR(255) comment '头像'
);

CREATE TABLE IF NOT EXISTS tree_hole ( id BIGINT AUTO_INCREMENT PRIMARY KEY
comment '树洞 ID',
    from_id BIGINT comment '用户 ID',
    message TEXT comment '树洞内容',
    gmt_create DATETIME default CURRENT_TIMESTAMP comment '创建时间',

```

```

gmt_modified DATETIME default CURRENT_TIMESTAMP comment '修改时间',
deleted INT default 0 comment '逻辑删除'
);

CREATE TABLE IF NOT EXISTS party_type ( id BIGINT PRIMARY KEY comment '组局类型 ID',
name VARCHAR(10) comment '组局类型名',
gmt_create DATE comment '创建时间',
deleted VARCHAR(50) default 0 comment '逻辑删除',
gmt_modified VARCHAR(50) comment '最近一次修改时间'
);

CREATE TABLE IF NOT EXISTS party_participants ( id BIGINT PRIMARY KEY comment '组局参与表 ID',
party_id BIGINT comment '对应组局 ID',
participant_id BIGINT comment '参与者 ID',
deleted VARCHAR(50) default 0 comment '逻辑删除',
gmt_modified VARCHAR(50) comment '最近一次修改时间',
gmt_create DATE comment '创建时间'
);

CREATE TABLE IF NOT EXISTS party_comment (
id BIGINT PRIMARY KEY comment '组局评论 ID',
information VARCHAR(50) comment '内容',
party_id BIGINT comment '对应组局 ID',
pre_id BIGINT comment '父评论 ID',
id_from BIGINT comment '评论者 ID',
status INT comment '评论状态 0(正常状态) ,1（举报过多被挂起），2（已被删除）',
gmt_create DATE comment '创建时间',
deleted VARCHAR(50) default 0 comment '逻辑删除',
gmt_modified VARCHAR(50) comment '最近一次修改时间'
);

CREATE TABLE IF NOT EXISTS party ( id BIGINT PRIMARY KEY comment '组局 ID',
description VARCHAR(50) comment '组局描述',
publisher_id BIGINT comment '发布者 ID',
party_type_id BIGINT comment '组局类型 ID',
people_cnt INT(50) comment '组局人数限定',
image_urls VARCHAR(50) comment '图片 url 链接',
gmt_create DATE comment '创建时间',
gmt_modified VARCHAR(50) comment '最近一次修改时间',
deleted VARCHAR(50) default 0 comment '逻辑删除'
);

alter table post add constraint post_publisher_id_fk0 foreign key (publisher_id) references user
(id);

alter table post add constraint post_post_type_id_fk0 foreign key (post_type_id) references

```

```

post_type (id);
alter table post_comment add constraint post_comment_pre_id_fk0 foreign key (pre_id)
references post_comment (id);
alter table post_comment add constraint post_comment_post_id_fk0 foreign key (post_id)
references post (id);
alter table post_comment add constraint post_comment_id_from_fk0 foreign key (id_from)
references user (id);
alter table report add constraint report_user_id_fk0 foreign key (user_id) references user (id);
alter table report add constraint report_post_id_fk0 foreign key (post_id) references post (id);
alter table post_type add constraint post_type_pre_id_fk0 foreign key (pre_id) references
post_type (id);
alter table post_like add constraint post_like_post_id_fk0 foreign key (post_id) references post
(id);
alter table post_like add constraint post_like_id_from_fk0 foreign key (id_from) references user
(id);
alter table post_reward add constraint post_reward_post_id_fk0 foreign key (post_id)
references post (id);
alter table post_eye_on add constraint post_eye_on_post_id_fk0 foreign key (post_id)
references post (id);
alter table post_eye_on add constraint post_eye_on_id_from_fk0 foreign key (id_from)
references user (id);
alter table black_list add constraint black_list_user_id_fk0 foreign key (user_id) references user
(id);
alter table black_list add constraint black_list_be_user_id_fk0 foreign key (be_user_id)
references user (id);
alter table private_chat add constraint private_chat_from_id_fk0 foreign key (from_id)
references user (id);
alter table private_chat add constraint private_chat_to_id_fk0 foreign key (to_id) references
user (id);
alter table tree_hole add constraint tree_hole_from_id_fk0 foreign key (from_id) references
user (id);
alter table party_participants add constraint party_participants_participant_id_fk0 foreign key
(participant_id) references user (id);
alter table party_participants add constraint party_participants_party_id_fk0 foreign key
(party_id) references party (id);
alter table party_comment add constraint party_comment_pre_id_fk0 foreign key (pre_id)
references party_comment (id);
alter table party_comment add constraint party_comment_id_from_fk0 foreign key (id_from)
references user (id);

```

