Daily6+1 系统

数据库设计说明书

文件状态:	文件标识:	
[ ] 草稿	当前版本:	1.0
[√] 正式发布	作　　者:	邵研、陈炎、陈静、马科学、潘晨宇
[ ]正在修改	完成日期:	2020/4/9

版 本 历 史

版本/状态	作者	参与者	起止日期	备注
1.0	陈炎、陈静、马科学、邵研		2020/4/7-4.8	开始编写
1.1	潘晨宇		4/7-4.9	修改排版

修 改 记 录

日期	修订版本	修改章节	修改描述	作者
2020/4/4	1.0	4	编写系统方案	潘晨宇
2020/4/6	1.1	5	完整系统后端接口细节	潘晨宇

目录

G.1 引言

G.1.1 编写目的

数据库的表结构设计是整个项目开发中一个非常重要的环节,一个良好的数据库设计，可以提高开发效率，方便系统维护，并且为以后项目功能的扩展留下余地。我们通过书写这份文档说明，从各方面进行对 Daily6+1 的数据库设计规划，用它指导该系统在数据库各方面的内容，为系统开发的程序员、系统分析员提供基准文档。我们也希望通过写数据设计说明书,规范数据名称、数据范围、数据代码等。这份文档是项目小组共同作战的基础，有了开发规范、程序模块之间和项目成员之间的接口规则、数据方式,大家就有了共同的工作语言、共同的工作平台，使整个软件开发工作可以协调有序地进行。

G.1.2 背景

1. 软件系统名称：Daily 6+1

2. 系统类型：WebApp

3. 系统从属:无从属关系。会关联到第三方系统,如百度地图、谷歌识别等。分为前台子系统与后台子系统

4. 项目提出者：潘晨宇

5. 开发项目组名称：Daily6+1

6. 软件使用硬件背景：支持 HTML5 网页的所有设备。

1.3 参考资料

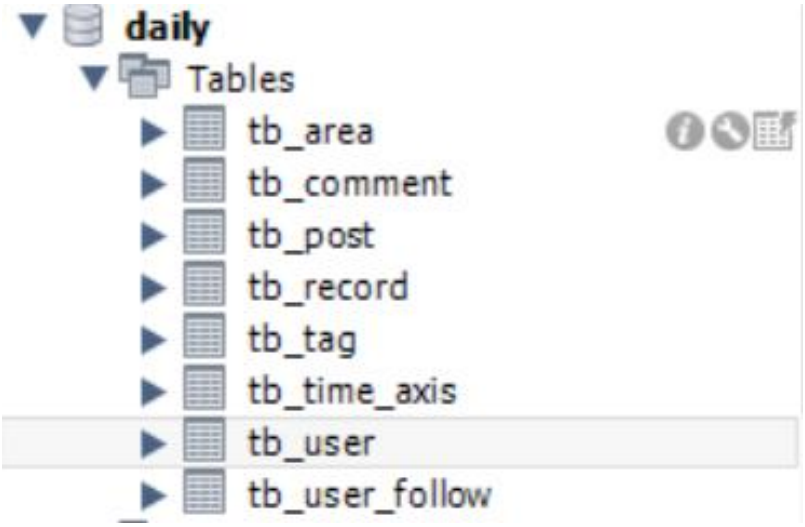
《数据库设计国标》 。

G.2 外部设计

G.2.1 标识符和状态

数据库软件：Mysql 5.7

数据库名：daily



表名	主键	描述
tb_area	area_id	地区表
tb_comment	comment_id	保存发布的评论
tb_post	post_id	保存发布的帖子

tb_record	record_id	保存发布的动态
tb_tag	tag_id	保存帖子的标签
tb_time_axis	time_axis_id	保存时间轴
tb_user	user_id	保存用户信息
tb_user_follow	user_follow_id	保存用户之间关注的关系

G.2.2 使用它的程序

应用程序	访问的数据表
登录	用户表
审核帖子	帖子表
冻结用户	用户表
广场显示帖子	帖子表
发布、点赞、转发、评论、举报、删除帖子	贴子表
搜索帖子	帖子表、地区表
为帖子添加标签	帖子表、标签表
关注发帖人	帖子表、用户关注表

显示高亮地图	帖子表、地区表
时间轴显示动态	动态表、时间轴表
添加动态	动态表
删除动态	动态表
修改动态	动态表
修改个人信息	用户表

G.2.3 约定

表名	记录	数据项命名的约定
tb_area	Area	无
tb_comment	Comment	无
tb_post	Post	无
tb_record	Record	无
tb_tag	Tag	无
tb_time_axis	TimeAxis	无
tb_user	User	无
tb_user_follow	UserFollow	无

G.2.4 专门指导

准备从事此数据库的生成、从事此数据库的测试、维护人员提供专门的指导。

G.2.5 支持软件

支持软件名称	版本号	主要功能
MySql	5.7	建立数据库并提供数据库维护与管理功能
Java	1.8	代码编写
Javascript		脚本语言，提供 HTTP 的技术支持
navicat		管理员工具
SpringBoot		后端框架
Mybatis		持久层框架

G.3 结构设计

G.3.1 概念结构设计

说明本数据库将反映的现实世界中的实体，属性和它们之间的关系等的原始数据形式，包括各数据项、记录、系、文卷的标识符、定

义、类型、度量单位和值域，建立本数据库的每一幅用户视图。

用户模块

实体：用户

主要功能：显示帖子和动态，搜索帖子和动态，修改个人信息，登录。


帖子
用户id
显示帖子和动态
动态


用户id
帖子
搜索帖子和动态
搜索内容
动态

帖子模块

实体：帖子

主要功能：发布帖子，赞转举报帖子，发布帖子。


发布帖子
帖子
标签
用户id
地区
是否匿名
帖子内容
tag
图片url


帖子
赞转举报帖子
点赞数
举报数
转发数


评论帖子
评论
帖子
用户id
评论内容
评论数
是否匿名

地图模块

主要功能：显示地图，搜索帖子。

时间轴模块

实体：动态，时间轴，用户时间轴。

主要功能：编辑时间轴

广场模块

主要功能：展示关注列表帖子，展示所有帖子，查看帖子。

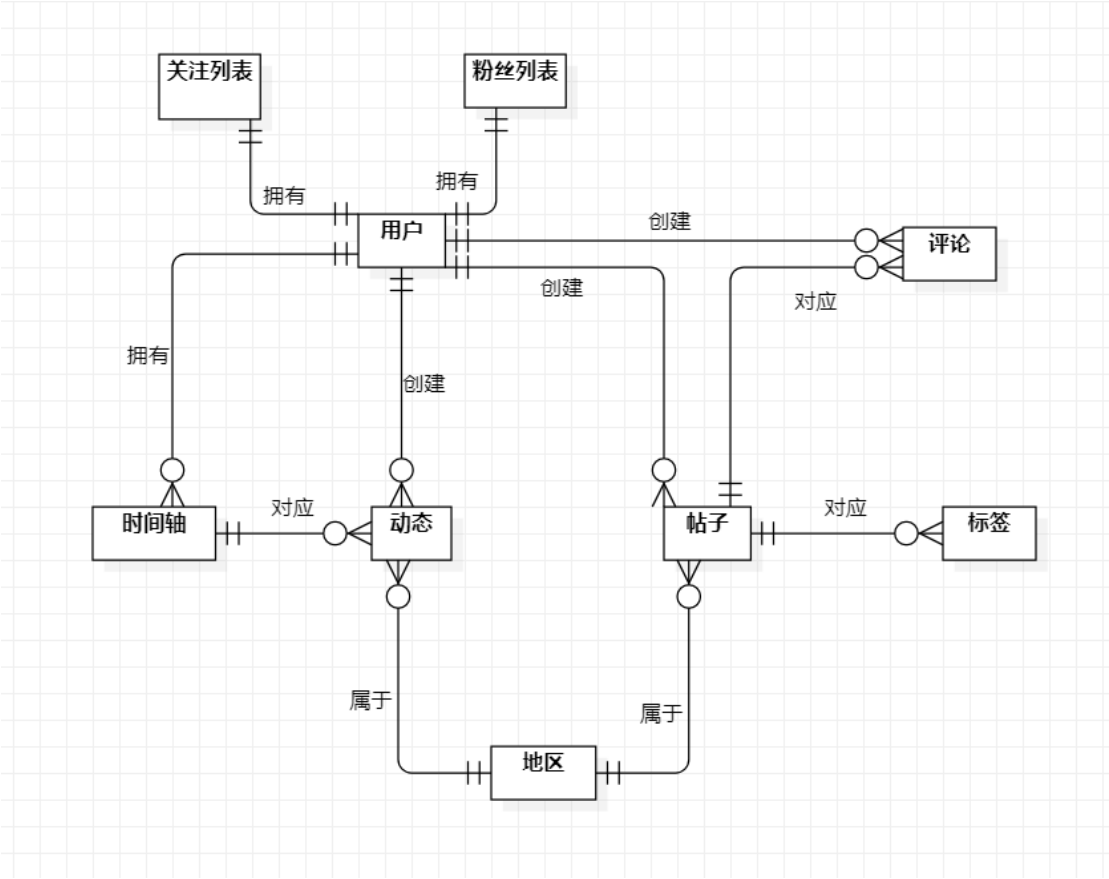

展示关注列表帖子
用户id
关注
帖子


查看帖子
帖子
用户

管理员模块

主要功能：删除帖子，冻结用户。

全局的 ER 图：



G.3.2 逻辑结构设计

表 tb_area 的结构（用于记录地区 ID 和名称）

字段名	数据类型	长度	主键	非空	描述
area_id	整数	/	是	是	自增
area_name	字符串	45	否	是	地区具体名称

表 tb_comment 的结构（用于保存评论的相关信息）

字段名	数据类型	长度	主键	非空	描述
comment_id	整数	/	是	是	自增
comment_content	字符串	500	否	是	评论内容
comment_create_time	时间格式	/	否	是	评论的时间

comment_update_time	时间格式	/	否	是	评论修改时间
post_id	整数	/	否	是	被评论帖子ID
user_id	整数	/	否	是	评论用户 ID
anonym	整数	/	否	是	是否匿名评论（是 1，否 0）

表 tb_post 的结构（用于保存分享帖的相关信息）

字段名	数据类型	长度	主键	非空	描述
post_id	整数	/	是	是	自增
post_create_time	时间格式	/	否	是	帖子发表时间
post_update_time	时间格式	/	否	是	帖子修改时间
like_num	整数	/	否	是	点赞数
forward_num	整数	/	否	是	转发数
comment_num	整数	/	否	是	评论数
tipoff_num	整数	/	否	是	举报数
post_img	URL	1024	否	否	帖子包含的图片，上传到数据库转换成对应的 URL 格式
post_content	字符串	9999	否	是	帖子的内容
anonym	整数	/	否	是	是否匿名发帖（是 1，否 0）
area_id	整数	/	否	是	发帖地区 ID
user_id	整数	/	否	是	发帖用户 ID，如果是匿名发帖则通过某种加密算法加密
forward	整数	/	否	是	该帖子是否是转发的帖子（是 1 否 0）

表 tb_record 的结构（用于保存时间轴中动态的相关信息）

字段名	数据类型	长度	主键	非空	描述
record_id	整数	/	是	是	自增
record_img	URL	1024	否	是	动态包含的图片，上传到数据库转换成对应的 URL 格式
record_content	字符串	3000	否	是	动态文字内容
record_create_time			否	是	动态创建时间
record_update_time			否	是	动态修改时间
time_axis_id			否	是	动态所属时间轴 ID
area_id			否	是	动态所属地区 ID

表 tb_tag 的结构（用于保存分享帖的 tag 信息）

字段名	数据类型	长度	主键	非空	描述
tag_id	整数	/	是	是	自增
tag_content	字符串	45	否	是	标签内容
post_id	整数	/	否	是	标签对应的帖子 ID

表 tb_time_axis 的结构（用于保存时间轴的相关信息）

字段名	数据类型	长度	主键	非空	描述
time_axis_id	整数	/	是	是	自增
area_name	字符串	45	否	是	时间轴类别
user_id	整数	/	否	是	所属用户 ID

表 tb_user 的结构（用于保存用户的相关信息）

字段名	数据类型	长度	主键	非空	描述

user_id	整数	/	是	是	自增
user_type	整数	/	否	是	用户类型（普通用户0,管理员1）
user_name	字符串	45	否	是	用户名
user_pwd	字符串	45	否	是	用户密码,采用某种算法加密后存储。
user_img	URL	1024	否	是	用户头像，上传到数据库转换成对应的URL格式
area_id	整数	/	否	是	用户所属地区ID
user_date	时间格式	/	否	是	用户出生日期
gender	字符串	2	否	是	用户性别
profile	字符串	1024	否	是	用户简介
state	整数	/	否	是	是否可用（可用1，不可用0）
fans_num	整数	/	否	是	粉丝数
follow_num	整数	/	否	是	关注数

表tb_user_follow的结构（用于保存用户关注人的相关信息）

字段名	数据类型	长度	主键	非空	描述
time_axis_id	整数	/	是	是	自增
user_id	整数	/	否	是	用户ID
follow_id	整数	/	否	是	关注用户的ID
follow_time	时间格式	/	否	是	关注时间

G.3.3 物理结构设计

数据库名称：daily

创建tb_area表

```
CREATE TABLE `tb_area` (
  `area_id` int(0) NOT NULL AUTO_INCREMENT COMMENT '地区 id',
  `area_name` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL COMMENT '地区名',
  PRIMARY KEY (`area_id`) USING BTREE
)
```

创建 tb_comment 表

```
CREATE TABLE `tb_comment` (
  `comment_id` int(0) NOT NULL AUTO_INCREMENT COMMENT '评论 id',
  `comment_content` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL COMMENT '评论内容',
  `comment_create_time` datetime(0) NOT NULL COMMENT '评论时间',
  `commnet_update_time` datetime(0) NOT NULL COMMENT '评论更新时间',
  `post_id` int(0) NOT NULL COMMENT '被评论帖子 id',
  `user_id` int(0) NOT NULL COMMENT '用户 id',
  `anonym` int(0) NOT NULL COMMENT '是否匿名评论',
  PRIMARY KEY (`comment_id`) USING BTREE
)
```

创建 tb_post 表

```
CREATE TABLE `tb_post` (
  `post_id` int(0) NOT NULL AUTO_INCREMENT COMMENT '帖子 id',
  `post_create_time` datetime(0) NOT NULL COMMENT '发帖时间',
```

```
  `post_update_time` datetime(0) NULL DEFAULT NULL COMMENT '更新时间',
  `like_num` int(0) NOT NULL COMMENT '点赞数',
  `forward_num` int(0) NULL DEFAULT NULL COMMENT '转发数',
  `comment_num` int(0) NULL DEFAULT NULL COMMENT '评论数',
  `tipoff_num` int(0) NULL DEFAULT NULL COMMENT '举报数',
  `post_img` varchar(1024) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NULL DEFAULT NULL COMMENT '图片',
  `post_content` varchar(9999) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL COMMENT '内容',
  `anonym` int(0) NOT NULL COMMENT '匿名 1 非匿名 0',
  `area_id` int(0) NOT NULL COMMENT '帖子所属地区的 id',
  `user_id` int(0) NOT NULL COMMENT '发布者 id',
  `forward` int(0) NOT NULL COMMENT '该帖子是否是转发的帖子; 是 1 否 0',
  PRIMARY KEY (`post_id`) USING BTREE
)
```

创建 tb_record 表

```
CREATE TABLE `tb_record` (
  `record_id` int(0) NOT NULL AUTO_INCREMENT COMMENT '动态 id',
  `record_img` varchar(1024) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL COMMENT '动态的图片',
```

```
  `record_content` varchar(3000) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL COMMENT '动态文字内容',

  `record_create_time` datetime(0) NOT NULL COMMENT '动态创建时间',

  `record_update_time` datetime(0) NOT NULL COMMENT '动态更新时间',

  `time_axis_id` int(0) NOT NULL COMMENT '动态所属时间轴 id',

  `area_id` int(0) NOT NULL COMMENT '动态所属的地区 id',

  PRIMARY KEY (`record_id`) USING BTREE
)
```

创建 tb_tag 表

```
CREATE TABLE `tb_tag` (

  `tag_id` int(0) NOT NULL AUTO_INCREMENT COMMENT '标签 id',

  `tag_content` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL COMMENT '标签内容',

  `post_id` int(0) NOT NULL COMMENT '标签对应的帖子 id',

  PRIMARY KEY (`tag_id`) USING BTREE
)
```

创建 tb_time_axis 表

```
CREATE TABLE `tb_time_axis` (

  `time_axis_id` int(0) NOT NULL AUTO_INCREMENT COMMENT '时间轴 id',

  `time_axis_type` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL COMMENT '时间轴类别',

  `user_id` int(0) NULL DEFAULT NULL,
```

```
  PRIMARY KEY (`time_axis_id`) USING BTREE
)
```

创建 tb_user 表

```
CREATE TABLE `tb_user`  (
  `user_id` int(0) NOT NULL AUTO_INCREMENT COMMENT '用户 id',
  `user_type` int(0) NOT NULL COMMENT '普通用户 0, 管理员 1',
  `user_name` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL COMMENT '用户名',
  `user_pwd` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL COMMENT '用户密码',
  `user_img` varchar(1024) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL COMMENT '用户头像',
  `area_id` int(0) NOT NULL COMMENT '用户所属地区 id',
  `user_date` datetime(0) NOT NULL COMMENT '用户出生日期',
  `gender` varchar(2) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL COMMENT '用户性别',
  `profile` varchar(1024) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL COMMENT '用户简介',
  `state` int(0) NOT NULL COMMENT '是否可用 (可用 1, 不可用 0)',
  `fans_num` int(0) NOT NULL COMMENT '粉丝数',
  `follow_num` int(0) NOT NULL COMMENT '关注数',
  PRIMARY KEY (`user_id`) USING BTREE
```

)

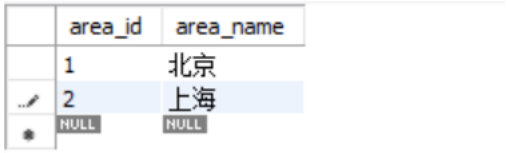
创建 tb_user_follow 表

```
CREATE TABLE `tb_user_follow` (
  `tb_user_follow_id` int(0) NOT NULL AUTO_INCREMENT COMMENT '用户关注表 id',
  `user_id` int(0) NOT NULL COMMENT '用户 id',
  `follow_id` int(0) NOT NULL COMMENT '关注用户的 id\n',
  `follow_time` datetime(0) NOT NULL COMMENT '关注时间',
  PRIMARY KEY (`tb_user_follow_id`) USING BTREE
)
```

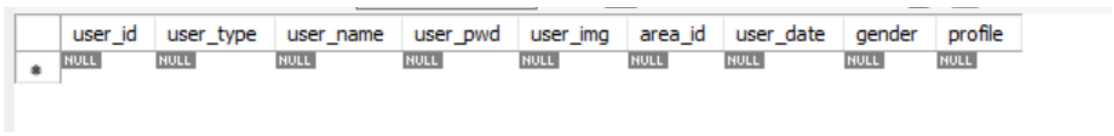
G.4 运用设计

G.4.1 数据字典设计

把主体的属性代码化放入独立的表中，不是和主体放在一起，主体中只保留属性的代码。这里属性的数量是不变的，而属性取值的数量可以是变化的.



area_id	area_name
1	北京
2	上海
NULL	NULL



user_id	user_type	user_name	user_pwd	user_img	area_id	user_date	gender	profile
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

G.4.2 安全保密设计

(1) 后台过滤：

后端设置过滤机制，使用过滤器对没有注册登录用户的请求进行拦截，不予放行，防止非法用户恶意操作，只有经过常规途径注册并登录的用户才能使用系统。

访问者	权限提供
用户	评论、转发、点赞、举报帖子、查看时间轴等
管理员	审核帖子、冻结用户

(2) 数据加密

对存储和传输的数据进行加密处理，从而使得不知道解密算法的人无法获知数据的内容。

(3) 防 SQL 语言注入

前端对输入进行验证，防止诸如“SELECT USER”等语句的输入导致后台执行数据库操作泄露数据库中数据。

(4) 存取控制

通过用户权限定义和合法权检查确保只有合法权限的用户访问数据库，所有未被授权的人员无法存取数据。 例如 C2 级中的自主存取控制(I)AC)，B1 级中的强制存取控制(M. AC)。

(5)　视图控制

为不同的用户定义视图，通过视图机制把要保密的数据对无权存取的用户隐藏起来，从而自动地对数据提供一定程度的安全保护。

(6)　安全可靠的存储

将数据库配置在含有物理安全防护以及互联网防护的云服务器端上，避免物理损坏等意外事故。