数据库设计说明书

魔幻现实处理加工队

目录

1. 引言

本数据库设计旨在支持系统的功能需求，为系统提供高效可靠的数据存储和管理。数据库设计应考虑到系统的性能、安全和可扩展性，以满足用户需求并支持系统未来的发展。

2. 外部设计

外部设计主要包括数据库的接口设计和数据模型设计。

2.1 接口设计

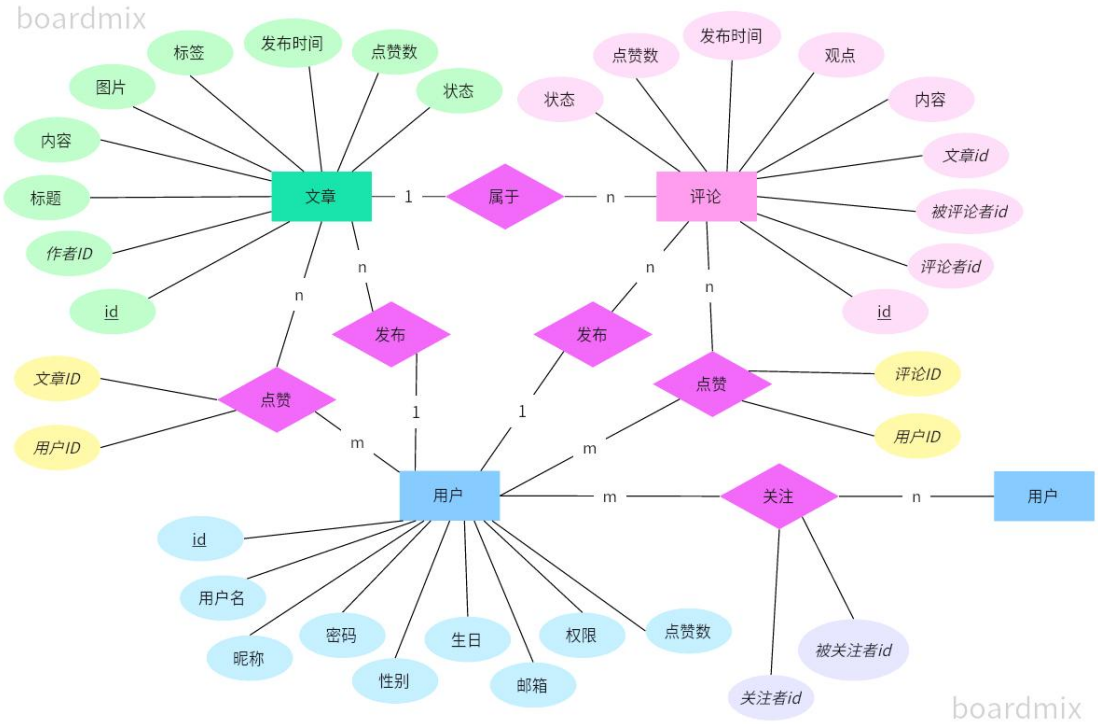
系统对外提供了一系列接口，包括用户注册、登录、发布文章、评论等功能。这些接口通过 SQL 语句实现对数据库的操作，保证数据的一致性和完整性。

2.2 数据模型设计

数据库采用关系型数据库 MySQL，通过实体-关系模型设计数据库

表结构，包括用户表、文章表、评论表、点赞表和好友表等，保证数据之间的关联性和有效性。

2.3 整体总览



3 结构设计

3.1 数据库表设计

- 用户表（User）：存储用户信息，如用户名、密码等。
- 文章表（Post）：存储用户发布的文章信息。
- 文章点赞表（PostLike）：存储用户对文章的点赞信息。
- 评论表（Comment）：存储用户对文章的评论信息。

- 评论点赞表（CommentLike）：存储用户对评论的点赞信息。
- 好友表（Friendship）：存储用户之间的好友关系信息。

3.2 关系建立

- 用户表与文章表之间为一对多关系，一个用户可以发布多篇文章。
- 文章表与文章点赞表之间为一对多关系，一篇文章可以被多个用户点赞。
- 用户表与评论表之间为一对多关系，一个用户可以发表多条评论。
- 文章表与评论表之间为一对多关系，一篇文章可以有多条评论。
- 评论表与评论点赞表之间为一对多关系，一条评论可以被多个用户点赞。
- 用户表与好友表之间为多对多关系，一个用户可以有多个好友，一个好友可以是多个用户的好友。

3.3 数据表字段设计

用户表（User）

- id (主键，自增长): 用户 ID
- username (唯一): 用户名
- password: 密码（加密存储）

- nickname : 昵称

- gender: 性别

- isOnline：是否在线

- description: 个人描述

- email: 邮箱

- picture: 头像（默认头像）

- birthday: 生日

- lastLogin: 最后登录时间

- privilege: 用户权限（例如：管理员、普通用户等）

```
mysql> desc user;
+-------------+--------------+------+-----+-------------------+-------------------+
| Field       | Type         | Null | Key | Default           | Extra             |
+-------------+--------------+------+-----+-------------------+-------------------+
| id          | int          | NO   | PRI | NULL              | auto_increment    |
| username    | varchar(20)  | NO   | UNI | NULL              |                   |
| password    | varchar(16)  | NO   |     | NULL              |                   |
| nickname    | varchar(20)  | NO   |     | NULL              |                   |
| gender      | tinyint(1)   | YES  |     | NULL              |                   |
| isOnline    | tinyint(1)   | NO   |     | 0                 |                   |
| thumbs      | int          | NO   |     | 0                 |                   |
| discription | text         | YES  |     | NULL              |                   |
| email       | varchar(30)  | YES  |     | NULL              |                   |
| picture     | varchar(100) | YES  |     | NULL              |                   |
| birthday    | date         | YES  |     | NULL              |                   |
| lastlogin   | datetime     | NO   |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| privilege   | tinyint(1)   | YES  |     | NULL              |                   |
+-------------+--------------+------+-----+-------------------+-------------------+
```

文章表（Post）

- id (主键，自增长): 文章 ID

- authorId (外键): 作者 ID

- title: 文章标题

- content: 文章内容

- thumbs: 点赞数

- postTime: 创建时间

- tags: 标签名称

- picture: 文章图片

- status: 关系状态

```
mysql> desc post;
+----------+-------------+------+-----+-------------------+-------------------+
| Field    | Type        | Null | Key | Default           | Extra             |
+----------+-------------+------+-----+-------------------+-------------------+
| id       | int         | NO   | PRI | NULL              | auto_increment    |
| authorId | int         | NO   | MUL | NULL              |                   |
| title    | varchar(50) | NO   |     | NULL              |                   |
| content  | text        | NO   |     | NULL              |                   |
| thumbs   | int         | NO   |     | 0                 |                   |
| postTime | datetime    | NO   |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| tags     | int         | NO   |     | 0                 |                   |
| picture  | text        | YES  |     | NULL              |                   |
| status   | tinyint(1)  | YES  |     | NULL              |                   |
+----------+-------------+------+-----+-------------------+-------------------+
```

文章点赞表（PostLike）

- postId (外键): 文章 ID

- userId (外键): 点赞用户 ID

```
mysql> desc thumbsPost;
+--------+------+------+-----+---------+-------+
| Field  | Type | Null | Key | Default | Extra |
+--------+------+------+-----+---------+-------+
| userId | int  | NO   | MUL | NULL    |       |
| postId | int  | NO   | MUL | NULL    |       |
+--------+------+------+-----+---------+-------+
```

评论表（Comment）

- id (主键，自增长): 评论 ID

- postId (外键): 文章 ID

- fromUser(外键): 一级评论用户 ID

- toUser(外键): 二级评论用户 ID

- content: 评论内容

- thumbs: 点赞数

- side: 边缘

- time: 创建时间

- status: 关系状态

```
mysql> desc comment;
+---------+------------+------+-----+-------------------+-------------------+
| Field   | Type       | Null | Key | Default           | Extra             |
+---------+------------+------+-----+-------------------+-------------------+
| id      | int        | NO   | PRI | NULL              | auto_increment    |
| postId  | int        | NO   | MUL | NULL              |                   |
| fromUser| int        | NO   | MUL | NULL              |                   |
| toUser  | int        | YES  | MUL | NULL              |                   |
| content | text       | NO   |     | NULL              |                   |
| thumbs  | int        | YES  |     | 0                 |                   |
| side    | tinyint(1) | YES  |     | NULL              |                   |
| time    | datetime   | NO   |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| status  | tinyint(1) | YES  |     | NULL              |                   |
+---------+------------+------+-----+-------------------+-------------------+
```

评论点赞表（CommentLike）

- id (主键，自增长)：点赞 ID

- commentId (外键)：评论 ID

- userId (外键)：点赞用户 ID

```
mysql> desc thumbsComment;
+-----------+------+------+-----+---------+-------+
| Field     | Type | Null | Key | Default | Extra |
+-----------+------+------+-----+---------+-------+
| userId    | int  | NO   | MUL | NULL    |       |
| commentId | int  | NO   | MUL | NULL    |       |
+-----------+------+------+-----+---------+-------+
```

好友表（Friendship）

- id (主键，自增长)：关系 ID

- userId (外键)：用户 1 ID

- userId2 (外键): 用户 2 ID

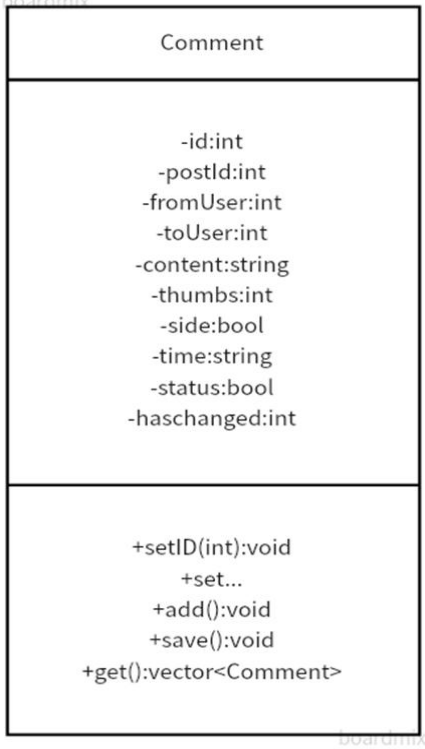
```
mysql> desc follows;
+---------+------+------+-----+---------+-------+
| Field   | Type | Null | Key | Default | Extra |
+---------+------+------+-----+---------+-------+
| userid  | int  | NO   | MUL | NULL    |       |
| userid2 | int  | NO   | MUL | NULL    |       |
+---------+------+------+-----+---------+-------+
```

通过以上设计，可以明确区分文章点赞和评论点赞，保证了数据库表结构的合理性和数据操作的准确性。

3.4 数据库结构设计


User
-id:int
-username:string
-password:string
-nickname:string
-gender:bool
-isOnline:bool
-thumbs:int
-description:string
-email:string
-picture:string
-birthday:string
-privilege:bool
-haschanged:int
+setID(int):void
+set...
+add():void
+save():void
+get():vector<User>


Post
-id:int
-authorId:int
-title:string
-content:string
-picture:string
-tags:int
-postTime:string
-thumbs:int
-status:bool
-haschanged:int
+setID(int):void
+set...
+add():void
+save():void
+get():vector<Post>


Comment
-id:int
-postId:int
-fromUser:int
-toUser:int
-content:string
-thumbs:int
-side:bool
-time:string
-status:bool
-haschanged:int
+setID(int):void
+set...
+add():void
+save():void
+get():vector<Comment>

❤ 为用户、文章、评论分别创建了一个类，类中的属性与数据库中的字段一一对应。

❤ 通过调用 set 方法来设置对象中的属性

❤ add 方法往数据库中新增一条数据

❤ save 方法对数据库中已有的数据进行修改

❤ get 方法用于根据已经设置的值在数据库中查找并返回

4. 运用设计

运用设计主要包括数据库的备份、恢复和性能优化。

4.1 数据库备份

定期对数据库进行备份，保证数据的安全性和完整性，防止数据丢失或遭受攻击后能够及时恢复。

4.2 数据库恢复

在数据丢失或遭受攻击后，能够通过备份数据进行恢复，保证系统的稳定性和可靠性。

4.3 性能优化

通过缓存技术、负载均衡等手段进行性能优化，提高系统的响应速度和并发处理能力，提升用户体验。