合肥工业大学

知识共享平台

数据结构设计说明书

项目小组：喜欢我对不队

小组成员：2022213386 徐力行

2022213384 陈远飞

2022213381 郭灵杰

2021215157 解正伟

2021215176 阎佳明

版　　本：第一版

发布日期：2025 年 1 月 5 日

版本记录

版本号	时间	记录人	变更描述
1	2024 年 1 月 8 日	徐力行	完成数据结构设计说明书初稿

目　录

1 引言

1.1 编写目的

传递设计意图：通过说明书，可以将数据结构的设计意图传递给其他人员，包括数据结构的目的、应用场景、使用方式等。这有助于提高沟通效率，减少误解和不必要的沟通成本。

统一规范：通过说明书，可以规范数据结构的命名、属性、操作等方面的设计，避免出现不一致或混淆的情况。这有助于提高代码的可读性和可维护性。

提高开发效率：通过说明书，可以让开发人员更快地理解和掌握数据结构的设计方案，避免重复开发和不必要的修改。这有助于提高开发效率和质量。

促进团队协作：通过说明书，可以让团队成员更加清晰地了解数据结构的设计细节，从而更好地协作完成开发任务。这有助于促进团队合作和提高工作效率。

保证系统稳定性：通过说明书，可以让维护人员更好地理解数据结构的设计和使用方式，从而更好地维护系统的稳定性和可靠性。

面向读者：数据结构设计人员

1.2 背景

知识共享是现代社会推动创新与协作的重要方式，对于知识的传播与价值的实现具有深远意义。随着信息技术的飞速发展，知识获取的渠道和形式变得多样化，但同时也伴随着知识碎片化和获取效率低下的问题。如何实现知识的高效共享和利用，成为社会关注的热点。

传统的知识传播方式通常依赖于单向的传播或个体间的线下互动，存在信息孤岛现象，难以满足现代社会对高效协作和共享的需求。为了解决这些问题，促进知识流通，知识共享平台应运而生。

该平台旨在通过数字化手段，打造一个开放、互动、高效的知识共享生态，为用户提供涵盖创建、分享、学习和合作的全面解决方案。平台支持多种形式的知识内容，包括文档、视频、课程和社区讨论等，满足不同领域用户的需求。

通过知识共享平台，用户可以方便地上传和分享自己的知识成果，系统提供标签化管理和智能推荐功能，帮助用户快速找到所需内容。同时，平台支持在线协作与实时互动，促进团队成员间的高效沟通与知识共享。

平台还具备数据分析和用户行为洞察功能，可以生成知识传播效果的统计分析报告，帮助用户了解知识的影响力和传播路径。此外，平台支持与第三方工具的无缝对接，实现知识的跨平台共享和沉淀。

知识共享平台的使用，将有助于个体和组织快速获取所需知识资源，提升创新能力和协作效率，推动知识价值的最大化，共建共享共赢的知识经济新时代。 🌐✨

1.3 定义

英文缩写	全称	释义
KSP	Knowledge Sharing Platform	知识共享平台
CMM	Content Management Module	内容管理模块
KA	Knowledge Analytics	知识分析
UC	User Collaboration	用户协作
NT	Networked Technology	网络化技术

1.4 参考资料

1、知识共享平台项目计划书

2、数据库设计基础课件

2 系统数据结构设计

2.1 逻辑结构设计要点

实体识别和定义：识别和定义数据库中的实体，即需要存储和管理的对象或概念。每个实体应该具有唯一的标识符，以便在数据库中进行区分和查询。

属性定义和数据类型选择：为每个实体定义属性，即实体具有的特性或描述性信息。选择适当的数据类型来存储属性值，并确定每个属性是否允许为空值。

关系建立和定义：根据实体之间的关系，建立适当的关系模型。常见的关系包括一对一、一对多和多对多关系。使用外键来定义实体之间的关联。

键的选择和定义：为每个实体选择主键，用于唯一标识实体。主键可以是单个属性或多个属性的组合。另外，还可以定义备选键和外键来支持数据的完整性和一致性。

约束条件定义：定义适当的约束条件来保证数据的完整性和一致性。常见的约束包括主键约束、唯一约束、默认值约束、检查约束和外键约束等。

视图定义：根据用户的需求，定义适当的视图来呈现数据库中的数据。视图可以提供数据的不同组合、筛选和计算等功能。

安全性和权限管理：定义适当的安全性和权限控制策略，限制用户对数据库的访问和操作。这包括用户身份验证、授权和审计等功能。

性能优化考虑：在设计过程中考虑数据库的性能优化问题，如索引的选择和创建、查询的优化、范式设计等。

数据库标准化：遵循数据库设计的标准化原则，如第一范式、第二范式和第三范式等，以减少数据冗余和提高数据的一致性。

文档化和维护：对数据库的逻辑结构设计进行文档化记录，包括实体关系图、属性定义、关系定义、约束条件等。并定期进行维护和更新，以适应业务需求的变化。

2.2 物理结构设计要点

引用数据结构：

•存储要求：包括引用编号（citation_id）、文章编号（article_id）、引用内容（content）、引用时间（timestamp）等。

•访问方法：可以通过引用编号、文章编号等方式查询引用信息。

•存取单位：每条引用记录作为一条独立的数据记录存储。

•存取的物理关系：采用关系型数据库存储，每个字段均具有唯一的数据类型和长度。

•设计考虑：需要确保引用编号的唯一性，并优化文章编号和引用时间的索引以提升查询效率。

•保密条件：引用数据中可能包含敏感信息，需对访问权限进行严格控制，避免未经授权的修改和泄露。

用户数据结构：

•存储要求：包括用户编号（user_id）、用户名（username）、邮箱（email）、注册时间（registration_date）、用户角色（role）等。

•访问方法：可以通过用户编号、用户名或邮箱查询用户信息。

•存取单位：每个用户信息作为一条独立的记录存储。

•存取的物理关系：采用关系型数据库存储，每个字段均具有唯一的数据类型和长度。

•设计考虑：确保用户编号和邮箱的唯一性，优化用户名字段以便快速检索。

•保密条件：对用户敏感信息（如邮箱、密码等）进行加密存储，限制访问权限以确保隐私。

文章数据结构：

•存储要求：包括文章编号（article_id）、标题（title）、内容（content）、作者编号（author_id）、创建时间（created_at）等。

•访问方法：可以通过文章编号、标题、作者编号等方式查询文章信息。

•存取单位：每篇文章作为一条独立的记录存储。

•存取的物理关系：采用关系型数据库存储，字段内容支持大文本类型（如TEXT 或 BLOB）。

•设计考虑：优化文章编号和标题字段的索引，确保查询和排序的效率。

•保密条件：对未公开的文章设置访问权限，仅允许特定用户或管理员查看。

评论数据结构：

•存储要求：包括评论编号（comment_id）、文章编号（article_id）、用户编号（user_id）、评论内容（content）、评论时间（timestamp）等。

•访问方法：可以通过评论编号、文章编号、用户编号查询评论信息。

•存取单位：每条评论作为一条独立的记录存储。

•存取的物理关系：采用关系型数据库存储，字段内容支持大文本类型以记录详细评论内容。

•设计考虑：确保评论编号的唯一性，并对文章编号和用户编号字段建立索引，以提升关联查询效率。

•保密条件：评论内容中可能包含敏感信息，对用户身份和评论内容进行保护，防止滥用和泄露。

标签数据结构：

•存储要求：包括标签编号（tag_id）、标签名称（tag_name）、标签描述（description）等。

•访问方法：可以通过标签编号或标签名称查询标签信息。

•存取单位：每个标签作为一条独立的记录存储。

•存取的物理关系：采用关系型数据库存储，每个字段均具有唯一的数据类型和长度。

•设计考虑：优化标签名称字段的索引，确保查询效率；对于高频标签需提供快速匹配能力。

•保密条件：对敏感标签信息设置访问权限，仅允许特定用户进行编辑或查看。

以上为知识共享平台的物理结构设计要点，旨在确保数据存储的高效性、查询的快速性以及敏感信息的安全性。

2.3 数据结构与程序的关系

数据结构	知识录入	权限管理	搜索功能	个人中心	推荐功能
知识数据结构	√	√			
评论数据结构	√	√			
收藏数据结构	√	√			
点赞数据结构	√				
用户数据结构	√	√			
搜索数据结构	√	√	√		
权限管理数据结构	√		√		
富文本编辑数据结构	√				
推荐算法数据结构	√		√	√	√

3 数据库设计

3.1 数据库环境说明

数据库开发使用 Mysql 8.3 版本，使用 navicate 结合 VScode 共同开发

```
1.    datasource:
2.        url:  jdbc:mysql://localhost:3306/ksp2?useSSL=false
3.        username:  root
4.        password:  123456
```

3.2 数据库的命名规则

1、表名：全部使用小写字母，中间用_分开，前缀加模块标识；

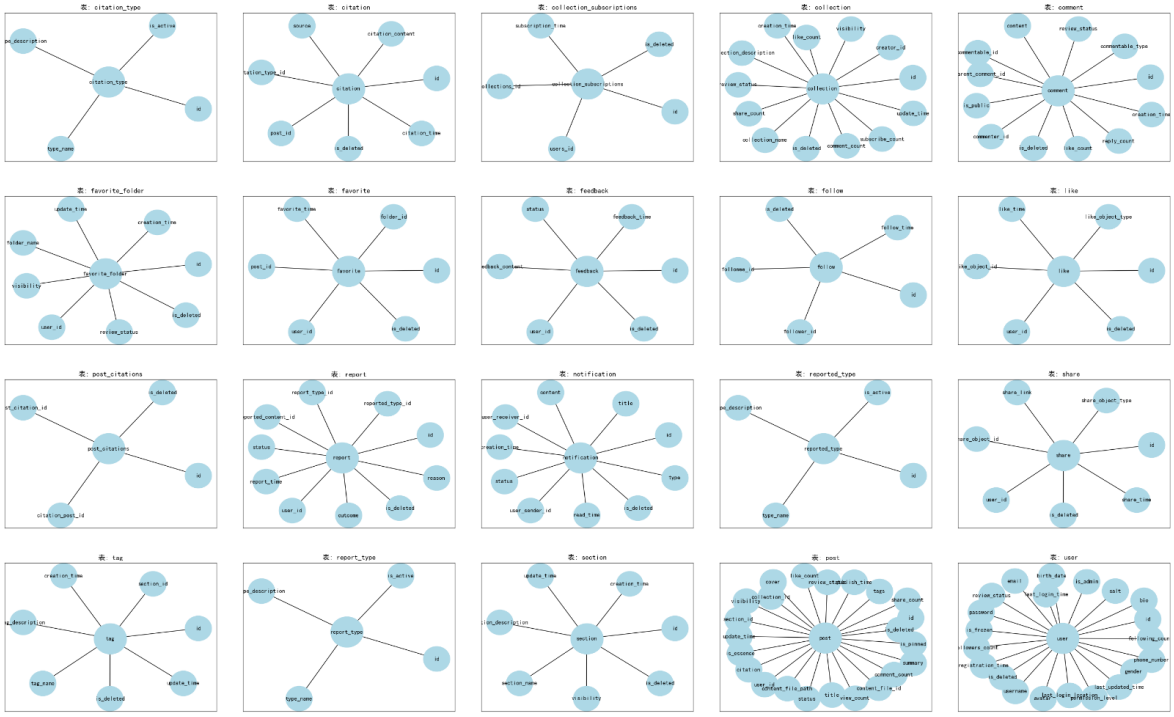
2、字段名：全部使用 id 作为主键，字段名全部由小写字母组成；

3、对象名称使用有意义的名称，做到见名知义；

4、表名设计清晰简短；

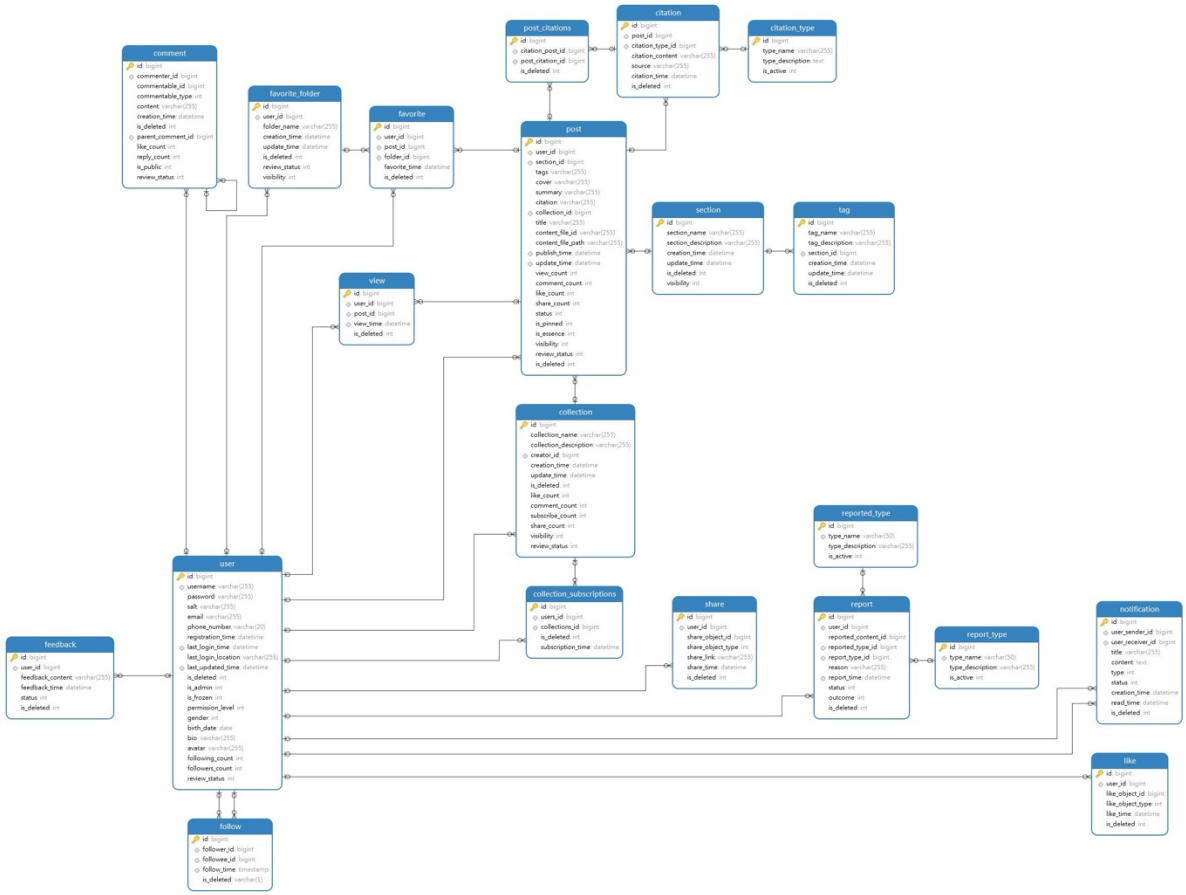
5、保证对象名称没有和保留词、数据库系统或者常用访问方法冲突。

3.3 概念设计



3.4 逻辑设计

3.5 物理设计



citation 表：

0	1	2	3	4	5
id	bigint	引用 ID，唯一标识一条引用记录，主键，自增		√	
post_id	bigint	帖子 ID，使用引用的帖子 ID，外键			关联帖子表 id
citation_type_id	bigint	引用类型 ID			关联引用类型表 id
citation_content	varchar	引用内容	255		
source	varchar	引用来源	255		
citation_time	datetime	引用时间			
is_deleted	int	是否删除			0 未删除，1 已删除

citation_type 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	引用类型 ID，唯一标识一个引用类型，主键，自增		√	
type_name	varchar	引用类型名称	255		
type_description	text	引用类型描述			
is_active	int	是否启用			1 表示启用

collection 表：

0	1	2	3	4	5
id	bigint	主键，唯一标识一个合集，自增		√	
collection_name	varchar	合集名称	255		
collection_description	varchar	合集描述	255		
creator_id	bigint	创建者 ID			关联用户表 id
creation_time	datetime	创建时间			
update_time	datetime	更新时间			
is_deleted	int	是否删除			1 表示删除
like_count	int	点赞数量			
comment_count	int	评论数量			
subscribe_count	int	订阅次数			
share_count	int	分享次数			
visibility	int	可见性			0 私密，1 公开
review_status	int	审核状态			0 待审核，1 通过，2 拒绝

collection_subscriptions 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，自增且唯一标识一条订阅记录		√	
users_id	bigint	订阅合集的用户 ID			关联用户表 id
collections_id	bigint	被订阅的合集 ID			关联合集表 id
is_deleted	int	是否删除			1 表示删除，0 表示未删除
subscription_time	datetime	订阅时间			默认为当前时间

comment 表：

0	1	2	3	4	5
id	bigint	主键，唯一标识一条评论记录，自增		√	
commenter_id	bigint	发表评论的用户 ID			关联用户表 id
commentable_id	bigint	被评论的对象 ID			可能是帖子、合集或其他评论的 ID
commentable_type	int	被评论对象的类型			1 表示帖子，2 表示合集，3 表示评论
content	varchar	评论的具体文本内容	255		
creation_time	datetime	评论创建的时间			
is_deleted	int	是否删除			0 表示未删除，1 表示已删除
parent_comment_id	bigint	回复的评论 ID			自关联到评论表 id
like_count	int	点赞数量			
reply_count	int	回复数量			
is_public	int	是否公开			0 表示否，1 表示是
review_status	int	审核状态			0 表示待审核，1 表示通过，2 表示拒绝

favorite 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一个收藏记录，自增		√	
user_id	bigint	进行收藏的用户 ID			关联用户表 id
post_id	bigint	被收藏的帖子 ID			关联帖子表 id
folder_id	bigint	收藏所属的收藏夹 ID			关联收藏夹表 id
favorite_time	datetime	用户收藏的时间			
is_deleted	int	是否删除			0 表示未删除，1 表示已删除

favorite_folder 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一个收藏夹，自增		√	
user_id	bigint	创建收藏夹的用户 ID			关联用户表 id
folder_name	varchar	收藏夹名称	255		
creation_time	datetime	收藏夹创建的时间			
update_time	datetime	收藏夹信息最后一次更新的时间			
is_deleted	int	是否删除			0 表示未删除，1 表示已删除
review_status	int	审核状态			0 表示待审核，1 表示通过，2 表示拒绝
visibility	int	可见性			0 表示私密，1 表示公开

feedback 表：

0	1	2	3	4	5
id	bigint	主键，唯一标识一条反馈记录，自增		√	
user_id	bigint	提供反馈的用户 ID			关联用户表 id
feedback_content	varchar	反馈内容	255		
feedback_time	datetime	提交反馈的时间			
status	int	反馈处理状态			0 未处理，1 已处理，2 已回复
is_deleted	int	是否删除			0 表示未删除，1 表示已删除

follow 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一条关注记录，自增		√	
follower_id	bigint	关注者 ID			关联用户表 id
followee_id	bigint	被关注者 ID			关联用户表 id
follow_time	timestamp	关注时间			
is_deleted	varchar	是否删除	1		0 表示未删除，1 表示已删除

like 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一个点赞记录，自增		√	
user_id	bigint	进行点赞的用户 ID			关联用户表 id
like_object_id	bigint	被点赞的对象 ID			关联帖子、合集或评论等表 id
like_object_type	int	对象类型			1 表示帖子，2 表示合集，3 表示评论
like_time	datetime	点赞时间			
is_deleted	int	是否删除			0 表示未删除，1 表示已删除

notification 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一条消息通知记录		√	
user_sender_id	bigint	发送通知的用户或系统 ID			关联用户表 id，系统通知可为空
user_receiver_id	bigint	接收通知的用户 ID			关联用户表 id
title	varchar	通知标题	255		
content	text	通知内容			
type	int	通知类型			1 评论回复，2 点赞，3 系统通知
status	int	通知状态			0 未读，1 已读
creation_time	datetime	创建时间			
read_time	datetime	阅读时间			未读则为空
is_deleted	int	是否删除			0 表示未删除，1 表示已删除

post 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一个帖子，自增		√	
user_id	bigint	发布帖子的用户 ID			关联用户表 id
section_id	bigint	帖子所属的分区 ID			关联分区表 id
tags	varchar	帖子标签	255		用于分类和搜索，可多个标签以逗号分隔
cover	varchar	帖子封面图片	255		路径或 URL
summary	varchar	帖子简要描述	255		
citation	varchar	引用信息	255		详细引用内容
collection_id	bigint	帖子所属的合集 ID			关联合集表 id
title	varchar	帖子标题	255		
content_file_id	varchar	富文本文件 ID	255		关联 MinIO 文件 ID
content_file_path	varchar	富文本文件路径	255		文件存储路径
publish_time	datetime	发布时间			
update_time	datetime	最后更新时间			
view_count	int	浏览次数			

comment_count	int	评论数量			
like_count	int	点赞数量			
share_count	int	分享次数			
status	int	帖子状态			1 已发布，0 草稿
is_pinned	int	是否置顶			1 是，0 否
is_essence	int	是否精华			1 是，0 否
visibility	int	可见性			0 私密，1 公开
review_status	int	审核状态			0 待审核，1 通过，2 拒绝
is_deleted	int	是否删除			0 表示未删除，1 表示已删除

post_citations 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，自增		√	
citation_post_id	bigint	关联引用表的帖子 ID			关联引用表 id
post_citation_id	bigint	关联帖子表的引用 ID			关联帖子表 id
is_deleted	int	是否删除			0 表示未删除，1 表示已删除

report 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一条举报记录，自增		√	
user_id	bigint	进行举报的用户 ID			关联用户表 id
reported_content_id	bigint	被举报的内容 ID			具体内容根据类型决定
reported_type_id	bigint	举报对象类型 ID			关联举报对象类型表 id
report_type_id	bigint	举报类型 ID			关联举报类型表 id
reason	varchar	举报理由	255		
report_time	datetime	举报时间			
status	int	处理状态			0 待处理，1 正在处理，2 已处理，3 无需处理
outcome	int	处理结果			0 无行动，1 删除内容，2 警告用户，3 禁言，4 封禁
is_deleted	int	是否删除			0 表示未删除，1 表示已删除

report_type 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一个举报类型，自增		√	
type_name	varchar	举报类型名称	50		如不当言论、色情内容等
type_description	varchar	举报类型描述	255		详细描述举报类型
is_active	int	是否启用			1 表示启用，0 表示未启用

reported_type 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一个举报对象类型，自增		√	
type_name	varchar	举报对象类型名称	50		如帖子、评论、用户等
type_description	varchar	举报对象类型描述	255		详细描述对象类型范围和特点
is_active	int	是否启用			1 表示启用，0 表示未启用

review_log 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一条审核记录，自增		√	
content_id	bigint	被审核内容的 ID			可能是用户、帖子、合集或收藏夹的 ID
content_type	int	被审核内容的类型			0 用户，1 帖子，2 合集，3 收藏夹
reviewer_id	bigint	审核人的 ID			关联用户表 id
review_status	int	审核状态			0 待审核，1 通过，2 拒绝
review_time	datetime	审核时间			默认当前时间
review_notes	text	审核备注			

section 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一个分区，自增		√	
section_name	varchar	分区名称	255		用于用户识别和展示
section_description	varchar	分区描述	255		分区的主题、目的等概述
creation_time	datetime	创建时间			
update_time	datetime	更新时间			
is_deleted	int	是否删除			0 表示未删除，1 表示已删除
visibility	int	可见性			0 表示隐藏，1 表示公开

share 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一个分享记录，自增		√	
user_id	bigint	分享的用户 ID			关联用户表 id
share_object_id	bigint	被分享的对象 ID			关联帖子、合集等对象表 id
share_object_type	int	对象类型			1 表示帖子，2 表示合集
share_link	varchar	分享链接	255		用于追踪分享内容
share_time	datetime	分享时间			
is_deleted	int	是否删除			0 表示未删除，1 表示已删除

tag 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一个标签，自增		√	

tag_name	varchar	标签名称	255		用于用户识别和展示
tag_description	varchar	标签描述	255		包含标签的主题或分类的额外信息
section_id	bigint	所属分区 ID			关联分区表 id
creation_time	datetime	创建时间			
update_time	datetime	更新时间			
is_deleted	int	是否删除			0 表示未删除，1 表示已删除

user 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一个用户，自增		√	
username	varchar	用户名	255		用户登录的名称
password	varchar	密码	255		加密后的密码
salt	varchar	密码加密的盐值	255		
phone_number	varchar	手机号	20		
registration_time	datetime	注册时间			
last_login_time	datetime	最后登录时间			
last_login_location	varchar	最后登录地点	255		
last_updated_time	datetime	最后更新时间			
is_deleted	int	是否删除			0 表示未删除，1 表示已删除
is_admin	int	是否管理员			1 表示是，0 表示否
is_frozen	int	是否被冻结			1 表示是，0 表示否
permission_level	int	权限等级			0 正常，1 禁止评论，2 禁止发帖
gender	int	性别			0 男，1 女，2 其他
birth_date	date	出生日期			
bio	varchar	个人简介	255		自我描述
avatar	varchar	头像路径	255		
following_count	int	关注数			
followers_count	int	粉丝数			
review_status	int	审核状态			0 待审核，1 通过，2 拒绝

view 表：

字段名	类型	注释	长度	是否必填	备注
id	bigint	主键，唯一标识一条浏览记录，自增		√	
user_id	bigint	浏览的用户 ID			关联用户表 id
post_id	bigint	被浏览的帖子 ID			关联帖子表 id
view_time	datetime	浏览时间			
is_deleted	int	是否删除			0 表示未删除，1 表示已删除

3.6 安全性设计

（1）访问控制：

设置严格的用户权限：为每个用户分配适当的权限，限制其对数据库的访问和操作。

强密码策略：要求用户使用复杂的密码，并定期更改密码以增加安全性。

多因素身份验证：使用多因素身份验证提供额外的安全层，例如使用密码和令牌或生物识别技术。

（2）数据加密：

数据库加密：对数据库中的敏感数据进行加密，确保数据在存储和传输过程中的安全性。

传输加密：使用安全的通信协议（如 HTTPS）来加密数据库与应用程序之间的数据传输。

（3）审计和日志记录：

审计跟踪：启用数据库的审计功能，记录对数据库的访问和操作，并存储审计日志以便监测和调查安全事件。

日志记录和监控：实时监控数据库活动和日志文件，及时检测异常行为和安全漏洞。

（4）异常检测和防御：

安全漏洞扫描：定期进行数据库安全漏洞扫描，及时修复发现的漏洞。

异常行为检测：使用入侵检测系统（IDS）或安全信息和事件管理系统（SIEM）监测数据库活动，检测并阻止异常行为。

（5）定期备份和恢复：

定期备份数据库：定期对数据库进行备份，以防止数据丢失，并确保备份数据的安全存储。

灾难恢复计划：制定灾难恢复计划，包括备份恢复测试、数据恢复策略和紧急情况下的应急响应措施。

（6）更新和补丁管理：

及时安装更新和补丁：定期更新数据库软件和相关组件，以修复已知的安全漏洞和问题。

3.6.1 防止用户直接操作数据库的方法

（1）应用程序层面的访问控制：

使用应用程序作为数据库的中间层：通过应用程序作为数据库的接口，用户只能通过应用程序进行数据的访问和操作，而无法直接连接到数据库。

限制用户权限：在应用程序中实施严格的用户权限管理，确保用户只能执行其授权范围内的操作。例如，通过身份验证和授权机制来限制用户的访问权限。

（2）数据库层面的访问控制：

禁止直接访问数据库端口：配置数据库服务器防火墙，仅允许来自应用程序服务器的连接，禁止外部直接访问数据库端口。

限制用户访问权限：在数据库层面设置用户账户，并为每个用户分配适当的权限。只允许授权的用户通过应用程序进行数据库访问。

（3）输入验证和过滤：

实施有效的输入验证：在应用程序层面对用户输入进行验证和过滤，以预防 SQL 注入、跨站脚本攻击等安全威胁。

使用参数化查询：通过使用参数化查询或预编译语句，可以防止用户直接将不受信任的数据传递给数据库。

（4）加密传输：

使用安全的通信协议：通过使用 HTTPS 等安全的通信协议，确保数据库与应用程序之间的数据传输过程中的机密性和完整性。

3.6.2 用户账号密码的加密方法

（1）哈希函数：

使用密码哈希函数：将用户密码作为输入，通过哈希算法生成一个固定长度的哈希值。常用的密码哈希函数包括 MD5、SHA-1、SHA-256 等。

盐值加密：为了增加密码的安全性，可以使用盐值来对密码进行加密。盐值是一个随机的字符串，与用户密码组合后再进行哈希运算。

（2）密钥派生函数（Key Derivation Function，KDF）：

KDF 是一种专门用于从密码生成密钥的函数。

常见的 KDF 算法包括 PBKDF2、bcrypt 和 scrypt。这些算法通过迭代和盐值的方式增加密码的安全性。

（3）公钥加密：

网络通信过程中的密码传输，用公钥加密算法，如 RSA 或椭圆曲线加密（ECC）。

用户的密码首先使用公钥加密，然后只有私钥持有者才能解密密码。

3.6.3 角色与权限

角色	可以访问的表与列	操作权限
系统管理员	所有表和所有字段	查询&修改
数据分析员	用户表、文章表、评论表	查询
内容管理员	文章表、评论表	查询&修改
普通用户	文章表（公开字段）、评论表（公开字段）	查询

3.7 优化

优先级	优化对象	措施
高	数据库性能	定期监控数据库服务器的性能指标，如 CPU 利用率、内存利用率、磁盘 IO 等，及时发现并解决性能问题。
中	索引	分析查询语句的执行计划，评估索引的使用情况，合理创建、删除或修改索引，以提高查询性能。
低	SQL 语句	通过分析和调整 SQL 语句，减少数据库的访问次数和数据传输量，提高查询效率。

3.8 数据库管理与维护说明

数据库备份与恢复：

定期进行数据库备份：根据业务需求和数据变更频率，制定合理的备份策略，并定期对数据库进行备份。备份可以包括完全备份和增量备份。

确保备份数据的安全存储：备份数据应存储在安全的介质中，如离线存储或加密存储，以防止意外数据丢失或未经授权的访问。

定期测试备份恢复：定期测试备份文件的完整性，并进行恢复测试，确保备份数据可以成功还原。

数据库性能优化：

监控数据库性能：定期监控数据库服务器的性能指标，如 CPU 利用率、内存利用率、磁盘 IO 等，及时发现并解决性能问题。

索引优化：分析查询语句的执行计划，评估索引的使用情况，合理创建、删除或修改索引，以提高查询性能。

SQL 语句优化：通过分析和调整 SQL 语句，减少数据库的访问次数和数据传输量，提高查询效率。

数据库安全管理：

用户权限管理：设置用户账户，并为每个用户分配适当的权限，确保用户只能执行其授权范围内的操作。

强化访问控制：限制数据库的直接访问，仅允许来自应用程序服务器的连接，并使用防火墙等措施阻止未经授权的访问。

定期审计和监测：启用数据库的审计功能，记录和监测数据库的访问和操作活动，发现潜在的安全风险和异常行为。

定期维护和更新：

数据库版本更新：定期评估数据库厂商发布的安全补丁和更新，及时应用以修复已知的漏洞和安全问题。

统计信息更新：定期更新数据库的统计信息，以确保优化查询计划的准确性和性能。

容灾与恢复：

制定灾难恢复计划：建立有效的灾难恢复计划，包括备份的恢复测试、数据恢复策略和紧急情况下的应急响应措施。

高可用性配置：根据业务需求，采用冗余机制如主备复制、集群等，确保数据库系统的高可用性和容错性。