软件系统分析与设计

**WeShare学术成果分享平台**

**数据库设计说明书**

|  |  |  |  |
| --- | --- | --- | --- |
| **项目组成员信息** | | | |
| **小组编号** | **15** | | |
| **小组名称** | **老师多给点分吧** | | |
| **小组联系人** | **何宇峰** | | |
| **学号** | **姓名** | **联系方式** | **本次实践中主要承担的工作内容** |
| 18182107 | 李睿楷 | [ricklee0624@gmail.com](mailto:ricklee0624@gmail.com) | 前端开发，完成数据库管理方案设计，参与讨论 |
| 18373041 | 王力 | [742650485@qq.com](mailto:742650485@qq.com) | 后端开发，完成数据库结构设计，参与讨论 |
| 18373056 | 罗驭 | [861392049@qq.com](mailto:861392049@qq.com) | 前端开发，完成数据库管理方案设计，参与讨论 |
| 18373082 | 罗志贤 | [1367678572@qq.com](mailto:1367678572@qq.com) | 前端开发，完成数据库管理方案设计，参与讨论 |
| 18373229 | 姚树基 | [3183903748@qq.com](mailto:3183903748@qq.com) | 整合修改小组成果，完善格式，参与讨论 |
| 18373355 | 张洋 | [zhangyang946@163.com](mailto:zhangyang946@163.com) | 前端开发，完成数据库管理方案设计，参与讨论 |
| 18373386 | 刘义浩 | lyh3024@163.com | 后端开发，完成数据库结构设计，参与讨论 |
| 18373669 | 赵致远 | [512063422@qq.com](mailto:512063422@qq.com) | 后端开发，完成数据库结构设计，参与讨论 |
| 18373672 | 何宇峰 | [18373672@buaa.edu.cn](mailto:18373672@buaa.edu.cn) | 前端开发，完成数据库管理方案设计，协调分工，主持讨论 |
| 18373761 | 吴涛 | [1422757727@qq.com](mailto:1422757727@qq.com) | 后端开发，完成数据库结构设计，参与讨论 |
| 78076003 | 金真率 | [jinsol000103@gmail.com](mailto:jinsol000103@gmail.com) | 前端开发，完成数据库管理方案设计，参与讨论 |

2020

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| V1.0 | 2020.11.28 | 姚树基 | 何宇峰 | 整合小组成员成果，完善格式 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[1. 引言 1](#_Toc57491378)

[1.1 项目背景 1](#_Toc57491379)

[1.2 编写目的 1](#_Toc57491380)

[1.3 参考资料 1](#_Toc57491381)

[1.4 相关文档 1](#_Toc57491382)

[2. 外部设计 2](#_Toc57491383)

[2.1 使用它的程序 2](#_Toc57491384)

[2.2 命名约定 2](#_Toc57491385)

[2.3 数据库平台 2](#_Toc57491386)

[3. 数据库结构设计 2](#_Toc57491387)

[3.1 概念结构设计 2](#_Toc57491388)

[3.2 逻辑结构设计 3](#_Toc57491389)

[3.3 物理结构设计 3](#_Toc57491390)

[3.4 数据库表设计 4](#_Toc57491391)

[3.4.1 数据实体汇总 4](#_Toc57491392)

[3.4.2 用户表 5](#_Toc57491393)

[3.4.3 管理员表 5](#_Toc57491394)

[3.4.4 门户表 5](#_Toc57491395)

[3.4.5 搜索记录表 6](#_Toc57491396)

[3.4.6 留言表 6](#_Toc57491397)

[3.4.7 论文表 6](#_Toc57491398)

[3.4.8 榜单表 6](#_Toc57491399)

[3.4.9 私信表 7](#_Toc57491400)

[3.4.10 模块表 7](#_Toc57491401)

[3.5 存储过程设计 7](#_Toc57491402)

[4. 数据部署方案 7](#_Toc57491403)

[5. 数据库管理系统 7](#_Toc57491404)

[5.1 数据库管理系统——MySQL 7](#_Toc57491405)

[5.2 数据库管理工具——Navicat 8](#_Toc57491406)

[6. 优化 9](#_Toc57491407)

[7. 数据库安全 9](#_Toc57491408)

[7.1 数据库安全问题 10](#_Toc57491409)

[7.1.1 部署前无安全测试 10](#_Toc57491410)

[7.1.2 功能中的缺陷成为了数据库安全问题 10](#_Toc57491411)

[7.1.3 弱而复杂的数据库基础架构 10](#_Toc57491412)

[7.1.4 无限的管理访问=糟糕的数据保护 10](#_Toc57491413)

[7.1.5 密钥管理不足 10](#_Toc57491414)

[7.1.6 数据库中的不规范 10](#_Toc57491415)

[7.1.7 数据量过大 10](#_Toc57491416)

[7.2 数据库安全措施 10](#_Toc57491417)

[7.2.1 防止用户直接操作数据库的方法 10](#_Toc57491418)

[7.2.2 用户帐号密码的加密方法 10](#_Toc57491419)

[7.2.3 角色与权限 10](#_Toc57491420)

[7.3 备份 11](#_Toc57491421)

[7.3.1 备份策略 11](#_Toc57491422)

[7.3.2 备份方式 11](#_Toc57491423)

[7.3.3 恢复策略 11](#_Toc57491424)

# 引言

## 项目背景

随着国家对科技创新的日益重视，学术成果正变成日益宝贵的企业和社会资源，如何有效地实现学术成果分享并进一步推动科研合作正成为科学技术领域的热门话题。现如今的互联网上存在着很多学术成果分享平台，它们各有各的特点，能够满足很多人的需要。但是我们经过需求调研，发现这些平台并没有较多的温情，这并不能完全满足“分享”这一精神的要求。

因此，在我们进行充足的需求调研之后，我们计划开发WeShare平台，着重体现share这一精神，确定了“FEEL”这一主题风格，即：

Free（免费）：更为普通人考虑，让每一个人都能分享科学的喜悦。

Efficient（高效）：更快更实用的高级搜索，让好文章不再难以寻找。

Easy（简单）：更简洁的页面，解放你的眼睛，更解放你的心情。

Legal（合法）：更好的版权政策，给每一位科研人员我们最大的敬意。

我们力求给每一位用户带来最好的体验！

## 编写目的

在完成架构设计之后，我们小组便着手准备编写代码和设计数据库。在组内开会讨论之后，我们确定了前后端的分工，后端同学完成数据库的结构设计，前端同学完成数据库的管理方案设计，整合工作后撰写了这份数据库设计说明书，向整个开发期提供后台数据库的设计，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。

本文档介绍了数据库的外部设计规则，建立了三个结构模型，详细描述了数据库表、字段信息，进而描述了数据库的部署和管理方案，最后介绍了数据库的安全性问题。

## 参考资料

1. 谭火彬. UML2面向对象分析与设计[M].第二版.清华大学出版社,2019

## 相关文档

1. 《学术成果分享平台-背景陈述》
2. 《15-01-需求调研-01-需求调研报告》
3. 《15-02-需求模型-01-系统需求规格说明书》
4. 《15-03-分析模型-01-分析模型文档》
5. 《15-04-架构设计-02-系统架构设计说明书》

# 外部设计

## 使用它的程序

访问此数据库的所有应用程序：WeShare学术成果分享平台

## 命名约定

为了编写方便，数据库的表空间，数、视图、列名存储过程等数据库对象均采用英文全称、端写、简写或者拼音来命名，必要时添加分隔符，以能够理解阅读为标准。

## 数据库平台

数据库软件：MySQL 8.0

操作系统：Windows 10

数据库模型设计：Power Designer 16

浏览器：Google Chrome

# 数据库结构设计

## 概念结构设计

下图为Weshare数据库概念结构图（可放大观看）



**图3.1 Weshare数据库概念结构图**

## 逻辑结构设计

下图为Weshare数据库逻辑结构图（可放大观看）



**图3.2 Weshare数据库逻辑结构图**

## 物理结构设计

下图为Weshare数据库物理结构图（可放大观看）



**图3.3 Weshare数据库物理结构图**

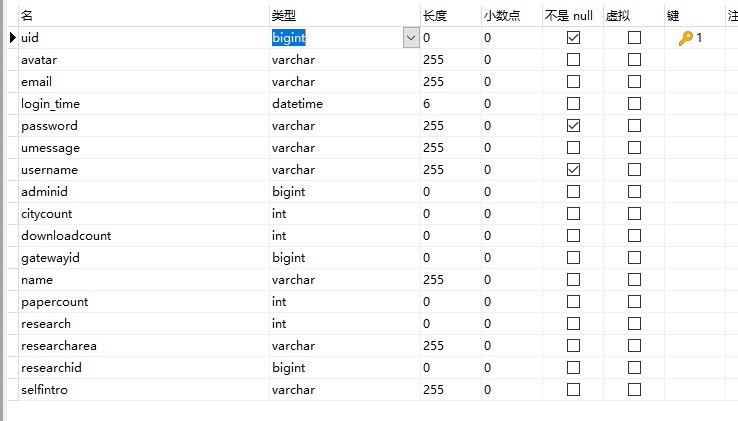
## 数据库表设计

### 数据实体汇总

|  |  |  |
| --- | --- | --- |
| 英文表名 | 中文表名 | 功能说明 |
| user | 用户 | 用户是在本平台注册了的人，是平台的主要使用者，包括平台的科研人员 |
| administrator | 管理员 | 管理员是管理平台的人，可以管理平台的用户和资源 |
| gateway | 门户 | 论文作者的门户界面，显示对应作者的科研信息 |
| search | 搜索记录 | 用户在平台搜索过的内容的历史记录 |
| comment | 留言 | 用户在论文介绍页面留言区发表的留言 |
| paper | 论文 | 系统爬取或科研人员上传的论文 |
| bangdan | 榜单 | 平台对论文库统计和分析之后按一定标准生成的论文排行榜 |
| privatemessage | 私信 | 私信是用户之间发送的消息 |
| module | 模块 | 对论文进行分类后得到各大论文模块 |

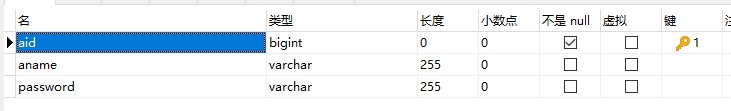
**表3.1 Weshare数据实体汇总表**

### 用户表



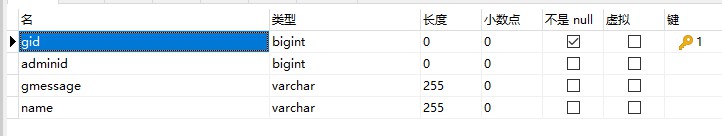
**图3.4 用户表**

### 管理员表



**图3.5 管理员表**

### 门户表

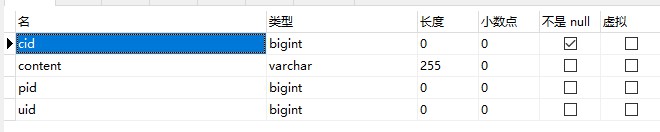
**图3.6 门户表**

### 搜索记录表



**图3.7 搜索记录表**

### 留言表



**图3.8 留言表**

### 论文表



**图3.9 论文表**

### 榜单表



**图3.10 榜单表**

### 私信表



**图3.11 私信表**

### 模块表



**图3.12 模块表**

## 存储过程设计

存储过程是一组预先定义并编译好的SQL语句，可以接受参数、返回状态值和参数值，并可以嵌套调用。使用存储过程可以改变SQL语句的运行性能，提高其执行效率。存储过程还可用做一种安全机制，使用户通过它访问未授权的表或视图。

# 数据部署方案

考虑到WeShare学术成果分享平台对安全性、高可用性、高并发性、大数据量的要求，由单台Mysql作为独立的数据库是不能满足实际需求的。我们将采用主从复制（Master-Slave）的方式来同步数据，采用读写分离（MySQL-Proxy）来提升数据库的并发负载能力。

# 数据库管理系统

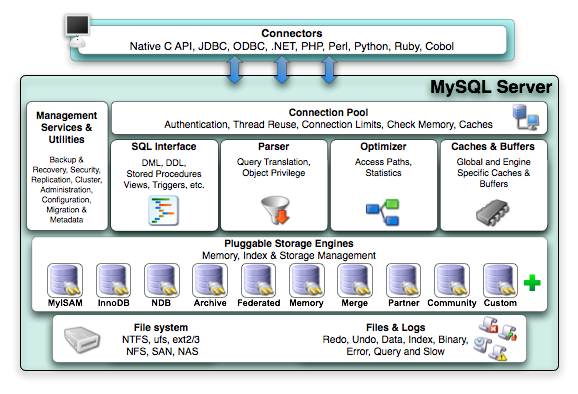
## 数据库管理系统——MySQL

经过小组讨论，我们决定使用MySQL数据库来管理数据MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。

MySQL是一种关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

下面是MySQL数据库管理系统的架构：



**图5.1 MySQL管理系统结构图**

## 数据库管理工具——Navicat

Navicat是一个可多重连接的数据库管理工具，它可以连接到MySQL、Oracle、PostgreSQL、SQLite、SQL Server和/或MariaDB数据库，让数据库管理更加方便。[Navicat](https://www.formysql.com/" \t "_blank)的功能可以满足专业开发人员的所有需求，对数据库服务器的新手来说学习起来也不难。Navicat同时拥有极完备的图形用户界面（GUI），可以让用户简便、安全地创建、组织、访问和共享信息。Navicat还具有很多优点，例如：

* 1. 智能对象设计

Navicat提供一个直观和设计完善的用户界面，用于创建、修改和管理资料库的所有对象，例如表、视图、函数或过程、索引、触发器和序列。通过表设计器帮助用户创建和修改数据库的表，让设置高级选项，如关系、限制、触发器。

* 1. 简化数据编辑

使用Navicat浏览和修改数据，插入、编辑、删除数据或复制和粘贴记录到数据表形式的数据编辑器，Navicat将运行相应的命令（例如INSERT 或UPDATE），免除写复杂的SQL。广泛的数据编辑工具令编辑工作更为方便，例如外键查找、set/enum选择器和记录筛选。

* 1. 先进的报表创建工具

报表创建工具能从用户的数据源转换数据成一个精美的报表。使用其交互式设计界面和高效的工作流程，可在几分钟内创建专业和高质量的报表。 丰富的图表和可视化的选项能提高用户的报表水准。一套全面的报表组件，如区域、子报表和交叉制表，让用户有个快速的开端来创建报表。

* 1. 智能模型设计

运用精密的数据库设计和模型工具，用户可以用图形表达数据库。使用实体关系图表来显现数据库结构及关系，这样就可以更容易塑造，建立和理解复杂的数据库。

* 1. 强大的用户管理

用户管理功能提升和管理每个用户的管理权限，不需输入任何命令，在数分钟内就能创建和编辑用户角色，借助这个精确控制的层面，可以在不影响数据库的安全性下，创建规则并让用户访问数据库。

* 1. 确保数据安全

Navicat提供本机备份解决方案，当发生灾难时确保复原数据库，使用Navicat计划功能自动运行备份，并保存到本机硬盘或网络硬盘。

# 优化

首先数据库从设计的角度进行优化，在进行表设计时充分考虑关系型数据库的特点，减少冗余，明确表之间的关联关系，另外从查询效率的角度去考虑，为减少过多关联而适当的增加冗余。

从数据库应用的角度进行优化，在认真分析业务特点的基础上，主要包括索引的建立、数据查询时的SQL调优等，以及数据库参数调优等。

|  |  |  |
| --- | --- | --- |
| 优先级 | 优化对象（目标） | 措施 |
| 1 | 数据库设计 | 尽量避免冗余，减少不必要的表之间关联 |
| 2 | 数据库参数 | 根据应用的特点，合理设置数据:库参数 |
| 3 | 应用级调优 | 合理使用索引，SQL调优 |

**表6.1 优化方案**

# 数据库安全

数据库作为管理系统的基础，通常保存着重要的商店经营信息和客户信息。由于涉及到财务管理，数据的完整性和安全性显得尤为重要。系统中的数据一旦丢失，将需要很长时间进行恢复，有时甚至使信息系统不得不从系统初始化阶段重新开始运行。因此，系统安全保护措施是否有效是数据库系统的主要技术指标之一。

## 数据库安全问题

### 部署前无安全测试

数据库被攻击最常见的原因之一就是在开发过程中部署阶段的疏忽。虽然为了确保高性能，我们可能进行了功能测试，但是这种类型的测试无法显示数据库是否正在执行不应该执行的操作。因此，在完全部署之前，使用不同类型的测试来测试网站安全性是非常重要的。

### 功能中的缺陷成为了数据库安全问题

黑客可以利用数据库的功能缺陷进行攻击，通过破解合法凭据并强制系统运行任意代码。所以我们可以通过安全测试保护数据库免受第三方访问。此外，其功能结构越简单，确保对每个数据库功能进行良好保护的机会就越多。

### 弱而复杂的数据库基础架构

不要让整个数据库基础结构过于复杂。当它很复杂时，有可能忘记或忽视检查和修复它的弱点。

### 无限的管理访问=糟糕的数据保护

管理员和用户之间应该有明确的分工，确保团队是有限制性的访问，这样如果有用户试图窃取任何数据，那么也会因未参与数据库管理的过程而遇到更多困难。同时也还可以限制用户帐户的数量。

### 密钥管理不足

对敏感数据进行加密是很重要的，注意在网页任何地方都不能出现敏感数据的明文形式。

### 数据库中的不规范

导致数据库漏洞的原因多种多样，因为需要测试网站安全性并定期进行数据保护。如果发现有任何差异，一定要尽快修复。

### 数据量过大

数据库存储数据过多，超过数据库容量会导致数据库出错。

## 数据库安全措施

### 防止用户直接操作数据库的方法

用户只能用帐号登陆到应用软件，通过应用软件访问数据库，而没有其它途径操作数据库。

### 用户帐号密码的加密方法

对用户帐号的密码进行加密处理，确保在任何地方都不会出现密码的明文。

### 角色与权限

使用者在使用此数据库时，由于不同的访问者的登陆号码不同，故相当于对于数据的访问设置了权限，所以数据可以说是安全的。

通过区分不同的访问者、不同的访问类型和不同的数据对象，进行分别对待而获得的数据库安全保密设计考虑。确定每个角色对数据库表的操作权限，如创建、检索、更新、删除等。每个角色拥有刚好能够完成任务的权限，不多也不少。在应用时再为用户分配角色，则每个用户的权限等于他所兼角色的权限之和。

## 备份

进行数据备份是保障系统安全的重要手段。数据备份需要严格按照事先制定的备份与故障恢复策略进行，并落实备份登记和检查措施。

### 备份策略

根据Weshare平台运行的情况，采用定期冷备份的方式对数据库进行全备份,然后在全备份的基础上,每天对数据库进行逻辑备份。

### 备份方式

全备份采用冷备份的方式，根据平台运行情况和备份数据规模，确定一个合适的周期，定期对数据文件、日志文件、控制文件、参数文件、口令文件等（包括路径）进行备份。

逻辑备份方式：每周一采用完全逻辑备份，周五采用累计增量逻辑备份，每周其它时间采用增量逻辑备份的方式。

备份方案根据实际情况可做进一步调整。

### 恢复策略

数据库恢复时，首先将最近一次冷备份的全库备份恢复。

然后对全席备份时间点之后的逻辑备份进行恢复，恢复时依次执行：

1. 完全增量导入；
2. 累计增量导入；
3. 最近的增量导入。