软件系统分析与设计

**IShare-学术成果分享平台**

**数据库设计文档**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **小组编号** | **3** | | | |
| **小组名称** | **就要整整齐齐队** | | | |
| **小组联系人** | **唐熙程** | | | |
| **学号** | **姓名** | **联系电话** | **本次实践中主要承担的工作内容** | **本次作业占比**  **(95%-105%之间)** |
| 20373809 | 周骁睿 | 18322377517 | 完善数据库表设计的细节 | 100% |
| 20373677 | 陈楚凡 | 18159211139 | 完善数据库表设计的细节 | 100% |
| 20373481 | 刘谕笑眉 | 18811596392 | 完善数据库表设计的细节 | 100% |
| 20373114 | 唐熙程 | 18873578949 | 完善数据库表设计的细节 | 100% |
| 20373571 | 吕新月 | 13305226276 | 完善数据库表设计的细节 | 100% |
| 20373565 | 黄瑞 | 15828398946 | 编写引言，数据设计概述，整理文档 | 100% |
| 20373661 | 徐亦佳 | 13857760431 | 查阅相关资料 | 100% |
| 20373721 | 洪秋訸 | 17398568289 | 完善数据库表设计的细节 | 100% |
| 20373420 | 张铭轩 | 18601000101 | 完善数据库表设计的细节 | 100% |
| 20373460 | 张博皓 | 17704295626 | 审核，修订文档 | 100% |
| 20372008 | 刘汉尧 | 18830788186 | 进行表格和图的绘制 | 100% |

2022

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| 1.0 | 2022/11/09 | 黄瑞 | 唐熙程 | 完成引言、数据概述 |
| 1.1 | 2022/11/10 | 徐亦佳 | 唐熙程 | 完成相关资料查找 |
| 1.2 | 2022/11/11 | 周骁睿，陈楚凡，刘谕笑眉，唐熙程，吕新月，洪秋訸，张铭轩 | 黄瑞 | 完成数据库的相关设计 |
| 1.3 | 2022/11/11 | 刘汉尧 | 张博皓 | 完成文档中相关表和图的绘制 |
| 1.4 | 2022/11/12 | 张博皓 | 唐熙程 | 审核、修订文档 |

目 录

[1. 引言 1](#_Toc29595)

[1.1 项目概述 1](#_Toc7542)

[1.2 文档概述 1](#_Toc29395)

[1.3 参考文献 1](#_Toc4124)

[2. 数据库架构设计 1](#_Toc23952)

[2.1 数据库管理系统 1](#_Toc19706)

[2.2 数据库容量规划 2](#_Toc24668)

[2.3 数据库架构 2](#_Toc1672)

[2.4 数据库部署方案 3](#_Toc9088)

[3. 数据模型 3](#_Toc19668)

[3.1 数据关系模型 3](#_Toc1043)

[3.2 数据库表设计详情 4](#_Toc26836)

[4. 数据库安全设计 9](#_Toc28294)

[5. 总结 10](#_Toc1855)

# 引言

## 项目概述

本数据设计说明书主要说明了关于IShare 学术成果分享平台的数据库设计。鉴于我们的项目还处于从0到1的初生阶段，我们不准备花费太多精力到额外的非必需特性上，待首轮开发结束后，我们会为系统加入更多的新特性（例如论文推荐功能，学术成果对接企业的推广功能）。在本阶段，我们计划以实现平台的核心功能和基础功能为工作重点，本文的数据库设计范围包括最为核心的**学术数据检索**和基础的**社交功能**所需要的数据。

## 文档概述

本文档旨在说明数据相关的设计方案，文档内容依次包括数据库架构、数据库物理模型和表设计、数据库安全设计。

数据库架构细致地说明了MySQL+Elastic Search的数据存储方案的交互方式以及该方案的优势，并且分析了我们预计使用的存储容量和部署方案，该数据库存储方案和部署方案则可以作为后续搭建数据库的过程参考和指导，便于项目上线。

数据库物理模型和表设计详细说明了数据库的数据存储结构，可以以此为依据确定后续业务逻辑的具体实现并作为后续开发的重要参照。

数据库安全设计从容灾和保密两个方面论述了我们项目对数据库安全方面的考虑，保证了数据库数据传输和保存的安全性，确保良好的用户体验，使系统的安全性、稳定性和有效性得到保障。

## 参考文献

1. 03-01-需求调研
2. 03-02-需求建模
3. 03-03-分析模型-01-分析模型总文档
4. 03-03-分析模型-04-架构设计文档
5. 《UML2面向对象分析与设计》 清华大学出版社

# 数据库架构设计



## 数据库管理系统

本项目中我们使用MySQL+Elastic Search作为数据库的管理系统。MySQL作为关系型数据库适用于对项目中的业务逻辑提供支持。一般而言，Elastic Search被视为搜索引擎，但实际上，使用JSON格式的Elastic Search已经成为了事实上的文档型数据库，Elasticsearch 更加擅长的是基于查询搜索的分析型数据库，倾向 OLAP。鉴于本项目中瓶颈性能在于大数据的查询检索，我们选择使用Elasticsearch 来完成此项任务，其余业务相关的任务交由MySQL处理。

## 数据库容量规划

根据前期调研、架构设计、以及现有数据的统计结果，我们对最终的数据规模进行了初步的估计。

参考当前用户量最多的谷歌学术，至少拥有一千万的每日访问量以及高达一亿的论文数量及一亿的学者主页数量。但以业界一流的用户量作为参考对我们的项目并不具有实际意义，一则目前项目并未推广，还没有到考虑如此大的高并发、多用户的时候，二则经费有限，服务器配置也无法做到同样的支持程度。

因此，我们做出了一个支持用户的预估性设计，在系统上线后我们的注册用户数量大致为万级，认证的入驻学者大致为千级，论文的数量与学者数量则是获取到了微软学术关系图的全部资源，论文数量约为2.3亿，学者数量为千万级。

考虑到了不同的数据类型需要这不同的存储空间，我们针对不同的数据做了估算。

对于用户来说，基本包含用户名、密码、头像路径、邮箱等基本信息字段，因此每个用户平均需要0.5KB的存储空间。而对于用户日志，我们设置一个存储上限，对于超出上限的日志我们会定时保存到离线的硬盘，超出一段时间（比如一年）后，这些日志信息被抛弃，以保证对存储空间的有效使用。我们考虑一个用户及其产生的日志大致需要2KB的存储空间。

对于文献来说，论文本身需要保存标题，摘要，引用书目，链接，作者等诸多信息。每一篇文献平均需要4KB的存储空间。对于论文的评论我们预估每篇文献的平均评论数量在10条左右。再考虑到评论对应的用户的标识以及文献的唯一标识等字段。每条大约需要0.5KB。

因此初步估算，用户数据大致需要20MB的空间，文献需要800GB的存储空间，文献评论大约需要1TB的存储空间。用户数据访问量以1万为基准，则产生的日志量大约为每天10MB左右，我们将日志信息的上限作为500M。综上结合各种数据，考虑到预留的弹性空间，总体需要2TB的存储空间。

鉴于经费问题，我们认为评论使用的空间可以在项目上线后弹性扩容，所以目前使用了1TB的存储容量。

## 数据库架构

我们使用了MySQL+Elastic Search作为项目的数据库管理系统。下面我们详细介绍我们对这两个系统的具体使用架构。

MySQL作为关系型数据库，对事务的支持非常好，支持原子性的事件操作，我们选择使用MySQL处理用户事务相关的业务逻辑，有利于保证数据的一致性和服务的可靠性。具体而言，我们将用户所使用的注册、登录、评论、笔记、申请认证、举报等功能需要的数据放到了MySQL数据库中进行管理。

Elastic Search作为搜索引擎，使用了倒排索引的方式加速各种查询的速度，做了充分的性能优化。因此我们选用Elastic Search存储海量的学术数据（论文、机构、作者、领域、期刊）。我们会定期（每月）向Elastic Search中插入新的学术数据更新最新的数据。

结合前文的数据库存储容量分析，我们发现Elastic Search存储了绝大部分的数据（800GB以上），而MySQL存储的数据相对较少。我们认为这是一个合理的设计，因为海量的学术数据中需要的操作是读操作，使用Elastic Search可以很好完成此任务。而涉及读写的任务交给MySQL处理，可以保证数据的一致性，进而保障系统的稳定可靠。

## 数据库部署方案

* 初期部署架构

考虑到我们目前只有一台高性能服务器，我们决定初期不采用主从数据库的方案。直接在高性能服务器部署项目应用的docker即可（docker中包含了数据库的配置）。

* 升级部署架构

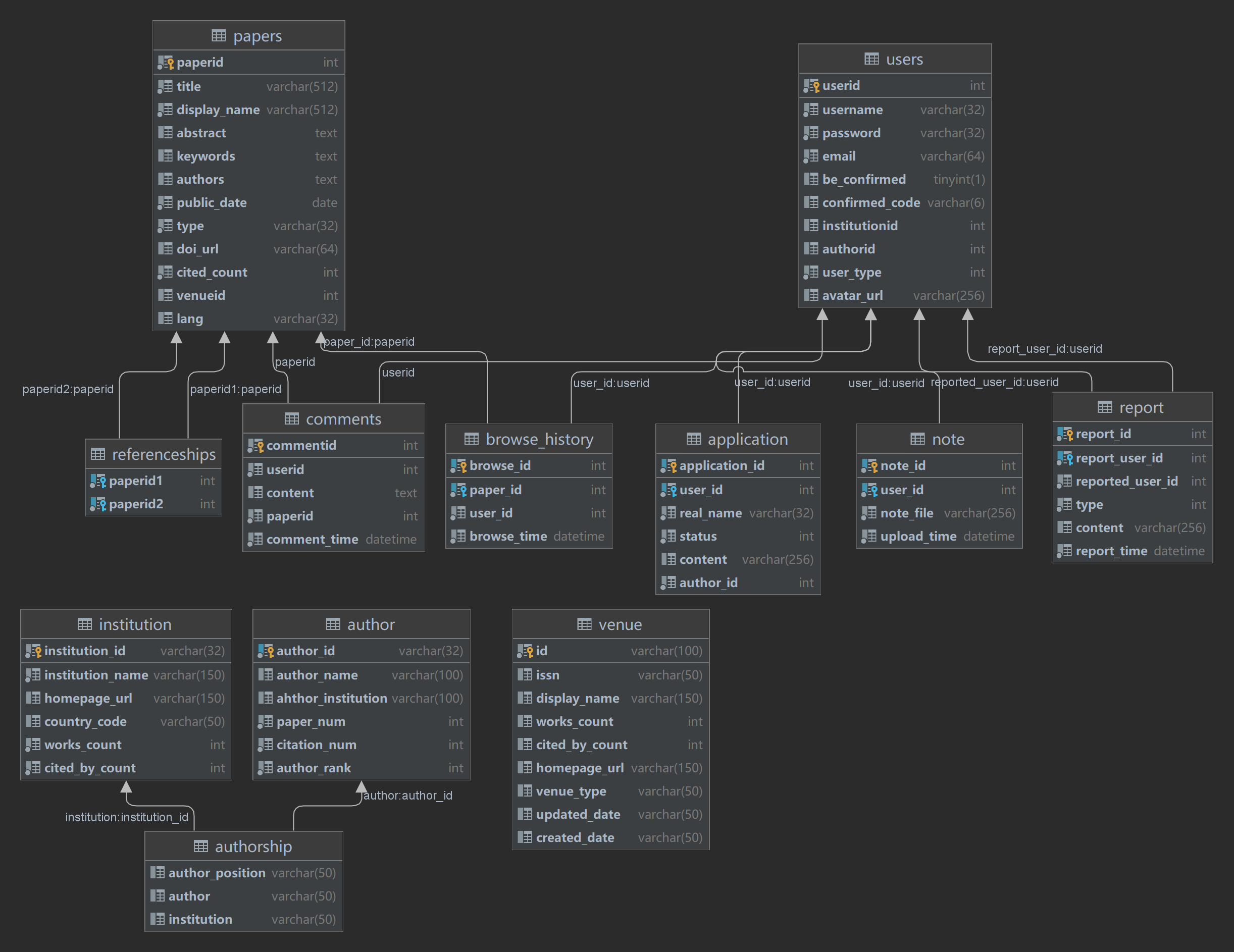
根据前文的描述，Elastic Search所需要的空间较大，我们其余的服务器并没有足够的空间。但是MySQL数据库需要的存储空间较少，可以做在其他服务器上做主从数据库的设计，提高数据库的性能和安全性。

我们在其他服务器上安装MySQL作为高性能服务器的从数据库（slave），以高性能服务器的MySQL作为主库，配置读写分离方案，写操作只在主库上进行，在从服务器上向外提供读功能。

# 数据模型



## 数据关系模型



**图3-1 数据关系模型图**

## 数据库表设计详情

评论表（comments）

**表3-1** comments**表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键/外键 | 能否为空 | 备注 |
| commentid | int | PRI | NO | 评论ID |
| userid | int |  | NO | 评论人的ID |
| content | text |  | YES | 评论内容 |
| paperid | int |  | NO | 被评论论文的ID |
| comment\_time | datetime |  | NO | 评论时间 |

论文表（papers）

**表3-2** papers**表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键/外键 | 能否为空 | 备注 |
| paperid | int | PRI | NO | 论文ID |
| title | varchar(512) |  | NO | 论文标题 |
| authors | text |  | NO | 论文作者 |
| display\_name | varchar(512) |  | NO | 论文显示的名称 |
| abstract | text |  | YES | 摘要 |
| keywords | text |  | YES | 关键词，由逗号分隔 |
| public\_date | date |  | NO | 发表时间 |
| type | varchar(32) |  | NO | 类型 |
| doi\_url | varchar(64) |  | YES | Doi号 |
| cited\_count | int |  | NO | 被引量 |
| venueid | int |  | YES | 所属期刊ID |
| lang | varchar(32) |  | YES | 论文使用的语言 |

论文引用关系表（referenceships）

**表3-3** referenceships**表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键/外键 | 能否为空 | 备注 |
| paperid1 | int | FORIGN | NO | 引用者的ID |
| paperid2 | int | FORIGN | NO | 被引者的ID |

用户表（users）

**表3-4 users表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键/外键 | 能否为空 | 备注 |
| userid | int | PRI | NO | 用户ID |
| username | varchar(32) |  | NO | 用户名 |
| password | varchar(32) |  | NO | 密码 |
| email | varchar(64) |  | NO | 邮箱 |
| be\_confirmed | tinyint(1) |  | YES | 是否验证有效用户，默认为0 |
| confirmed\_code | varchar(6) |  | YES | 验证码 |
| institutionid | int |  | YES | 所属机构ID |
| authorid | int |  | YES | 如果认证了学者，则等于作者ID |
| user\_type | int |  | NO | 用户类型 |
| avatar\_url | varchar(256) |  | YES | 头像url |

|  |  |
| --- | --- |
| User\_type可选值 | 含义 |
| 0 | 游客 |
| 1 | 普通用户 |
| 2 | 管理员 |

作者信息表(Author)：

**表3-5 Author表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键/外键 | 能否为空 | 备注 |
| author\_id | varchar(32) | PRI | NO | 作者id |
| author\_name | varchar(100) |  | NO | 作者名称 |
| author\_institution | varchar(100) |  | NO | 作者所属机构 |
| paper\_num | int |  | NO | 作者发表论文数量 |
| citation\_num | int |  | NO | 论文被引次数 |
| author\_rank | int |  | NO | 作者等级 |

著述表(Author)：

**表3-6 Authorship表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键/外键 | 能否为空 | 备注 |
| author\_position | varchar(50) | PRI | NO | 作者id |
| author | varchar(50) |  | NO | 作者 |
| institution | varchar(50) |  | NO | 作者所属机构 |

机构表(Institution):

**表3-7** Institution**表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键/外键 | 能否为空 | 备注 |
| institution \_id | varchar(32) | PRI | NO | 机构id |
| institution \_name | varchar(150) |  | NO | 机构名称 |
| homepage\_url | varchar(150) |  | YES | 机构官网链接 |
| country\_code | varchar(50) |  | YES | 地区码 |
| works\_count | int |  | NO | 文章数目 |
| cited\_by\_count | int |  | NO | 被引用次数 |

刊物表(Venue):

**表3-8** Venue**表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键/外键 | 能否为空 | 备注 |
| id | varchar(100) | PRI | NO | 刊物id |
| issn | varchar(50) |  | NO | 刊物的issn |
| display\_name | varchar(150) |  | NO | 刊物名 |
| homepage\_url | varchar(150) |  | YES | 官网链接 |
| works\_count | int |  | NO | 文章数目 |
| cited\_by\_count | int |  | NO | 被引用次数 |
| venue\_type | varchar(50) |  | NO | 刊物类型 |
| updated\_date | varchar(50) |  | YES | 更新时间 |
| created\_date | varchar(50) |  | YES | 创建时间 |

申请表(Application)：其中申请状态status=0时表示申请未处理，status=1时表示申请通过，status=2表示申请被拒绝。

**表3-9 Application表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键/外键 | 能否为空 | 备注 |
| application\_id | int | PRI | NO | 申请表id |
| user\_id | int | FORIGN | NO | 用户id |
| real\_name | varchar(32) |  | NO | 真实姓名 |
| status | int |  | NO | 申请状态 |
| content | varchar(256) |  | YES | 详细信息 |
| author\_id | int |  | NO | 被认证作者id |

举报表(Report)：其中举报类型type=1时表示用户所属文献内容违规，type=2时表示文献内容涉嫌抄袭……type = 0时为其他（默认）

**表3-10 Report表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键/外键 | 能否为空 | 备注 |
| report\_id | int | PRI | NO | 文献id |
| report\_user\_id | int | FORIGN | NO | 举报用户id |
| reported\_user\_id | int |  | NO | 被举报用户id |
| type | int |  | NO | 举报类型 |
| content | varchar(256) |  | NO | 举报内容 |
| report\_time | datetime |  | NO | 举报时间 |

笔记表(Note)：

**表3-11 Note表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键/外键 | 能否为空 | 备注 |
| note\_id | int | PRI | NO | 笔记id |
| user\_id | int | FORIGN | NO | 上传用户id |
| note\_file | varchar(256) |  | NO | 笔记文件地址 |
| upload\_time | datetime |  | NO | 上传时间 |

浏览记录表(Browse\_History)：

**表3-12 Browse\_History表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键/外键 | 能否为空 | 备注 |
| browse\_id | int | PRI | NO | 浏览记录id |
| paper\_id | int | FORIGN | NO | 文献id |
| user\_id | int |  | NO | 用户id |
| browse\_time | datetime |  | NO | 浏览时间 |

# 数据库安全设计

4.1容灾设计

数据库作为管理系统的基础，通常保存着重要的文献信息与用户信息。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。

我们主要使用的主从数据库可以对数据完成备份，保证即使在某些服务器宕机后依然可以为用户提供稳定的服务。

当然不排除所有服务器同时出现故障的情况。我们采用定期数据备份的措施保障系统安全。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。

4.2安全设计

另外，系统设置用户的标识以鉴定是否是合法用户，并要求合法用户设置其密码，保证用户身份不被盗用；系统对不同的数据设置不同的访问级别，限制访问用户可查询的处理数据类别和内容；系统对不同用户设置不同的权限，区分不同的用户，如区分管理员和普通用户。具体的系统配置应当根据系统实际运行情况做进一步的调整。

# 总结

本文中小组采用了MySQL和Elastic Search作为数据库管理系统，并进行了数据库的设计和管理。本文档中详细阐述了选择MySQL和Elastic Search的原因，并且对于数据库的架构设计做出了详尽的说明。文档中还简要介绍了部署方案，方便后续部署时进行参考。并且文档中包含了数据库表的详细设计，阐述了数据库的物理模型，为后续开发提供了重要的指导。随着开发的推进，在本文基础上会随时对数据库进行一定调整，以便用户获得最好的使用体验。