

《数据库系统原理》大作业

系统设计报告

题目名称：北航超算社团课程管理平台

学号及姓名：

20373067 张凯歌

20373110 李佳禄

20373159 李国玮

2022 年 12 月 22 日

组内同学承担任务说明

| 工作内容 | 主要负责学生 | 备注 | 工作占比 |
|--------------------|--------|--|------|
| 子任务 1：系统功能设计与数据库设计 | 张凯歌 | 张凯歌负责总体的需求和设计，部分前端文档撰写、部分前后端交互、部分前端页面完成和页面美化设计和部分页面实现 | 34% |
| 子任务 2：系统服务器端开发 | 李国玮 | 李国玮负责服务端开发，后端数据库表的定义和数据库开发和实现，API 定义和具体实现，系统安全性保障，后端文档撰写 | 33% |
| 子任务 3：系统客户端开发 | 李佳禄 | 李佳禄负责前端开发整体框架实现，主要页面实现和部分美化页面完成，部分前后端交互，部分前端文档撰写 | 33% |

一、需求分析

1. 需求描述

本次课程设计的超算课程实验平台是综合了选课系统、课程和作业管理系统、评价和交流系统、展示和公告系统，立足于北航超算社团面向全校师生的学习、交流、展示平台。

(1) 综合需求

超算是超级计算机的简称。超级计算机相比于普通的通用性计算机而言具有更高性能。一般使用每秒钟进行的浮点数运算（FLOPS）来衡量超级计算机的性能。目前现有的超级计算机的性能可达 10^{17} FLOPS；而普通的平板电脑的性能一般在 $10^{11} \sim 10^{13}$ FLOPS上下。超级计算机在计算科学的各个领域发挥着重要的作用，并用于各个领域的各种计算密集型任务，包括量子力学、天气预报、气候研究、石油和天然气勘探、分子建模（生物大分子、计算化合物）和物理模拟（分子动力学、模拟宇宙早期时刻，核武器爆炸和核聚变）。

北航超算队在为国家贡献超算人才上做出了卓越贡献：

- 2014年-2021年在世界大学生超算竞赛上，北航超算队获得总决赛**一等奖6次**、总决赛**亚军1次**、**最高性能奖1次**、**应用创新奖3次**
- 2017年欧洲大学生超算竞赛ISC上，获得总决赛**季军1次（北航首次进入该赛事）**、最高精度奖1次
- 2019年-2022年全国并行应用挑战赛PAC：优化组**银奖1项**、**铜奖3项**
- 2021年-2022年国产CPU并行应用挑战赛CPC上，获得**三等奖1项**、**并行基金奖1项**
- 2019年-2020年中国开源软件创意大赛上，获得**二等奖1项**、**三等奖1项**
- 2021年-2022年线性解法器算法与性能优化竞赛，获得**二等奖1项**、**三等奖1项**

在本学期，北航超算队成立了自己的社团，因此需要一个网络平台进行后续的招新宣传、社员培养练习并且逐步计划将超算成果和平台贡献给校内师生共同使用，推动超算与各个学科交叉融合。

为此，我们设计了超算课程实验平台，综合了选课系统、课程和作业管理系统、评价和交流系统、展示和公告系统以满足超算社团日常培训社员、社内团建交流和对外宣传招新的内容。此外，社团管理者可以通过评价查看同学们对于的学习状态并实时把控社团发展情况，方便开展超算训练、吸收优秀人才等。

(2) 课程和作业系统

普通的社团网页往往缺乏对于社团内部成员的训练，成员们也很难找到自己在团队中的定位，在进行组队比赛或训练的时候常常因为缺乏先前专业化的学习而导致任务效率低下、分工不明确的问题。

超算课程实验平台在社团管理中融入了选课系统和课程和作业系统。只需要课程相关负责人提前提供学习资料并布置好相关的学习任务，同学们就可以根据自己的爱好选择超算中不同领域的知识进行学习，从而充分发挥出个人的能力。超算在各个领域都发挥着巨大的作用，其知识面也非常广阔。因此选择合适的课程进行参加并精通是非常重要的。

(3) 评价和交流系统

目前对于超算的学习和讨论主要集中于微信，bilibili等视频网页下方评论区。但是这样评论的问题是问题讨论主题不明确，闲聊和讨论界线不明确，太过分散、不方便日后进行查找等问题。

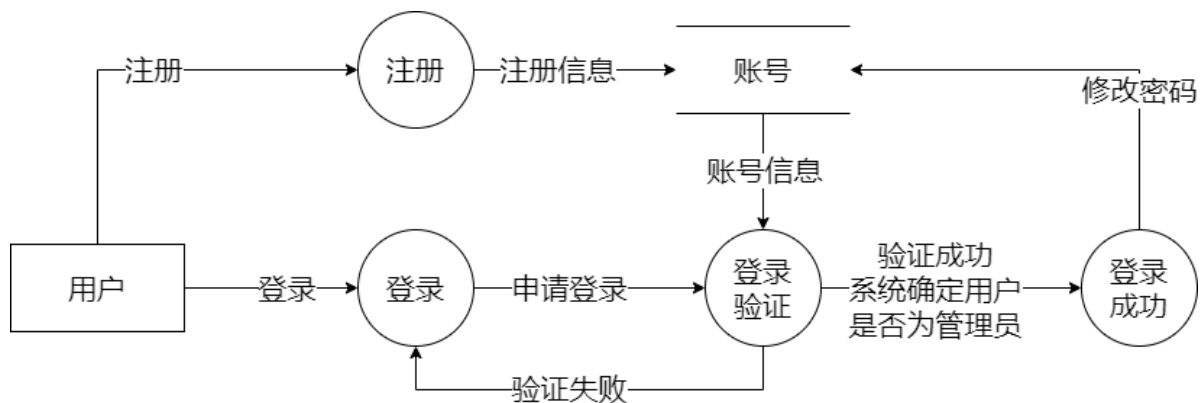
超算课程实验平台设置了评价和交流系统，社团成员可以创建讨论帖，发表学习看法、提出观点问题、或者仅仅评论有趣的新鲜事等，讨论结构分为主贴和跟帖回帖。主贴决定了讨论帖的主要内容，标题便于之后进行查找学习，跟帖回帖便于大家一起进行评论和讨论，畅所欲言。我们希望营造一个开心、欢快、自由、活泼的讨论环境。

(4) 展示和公告系统

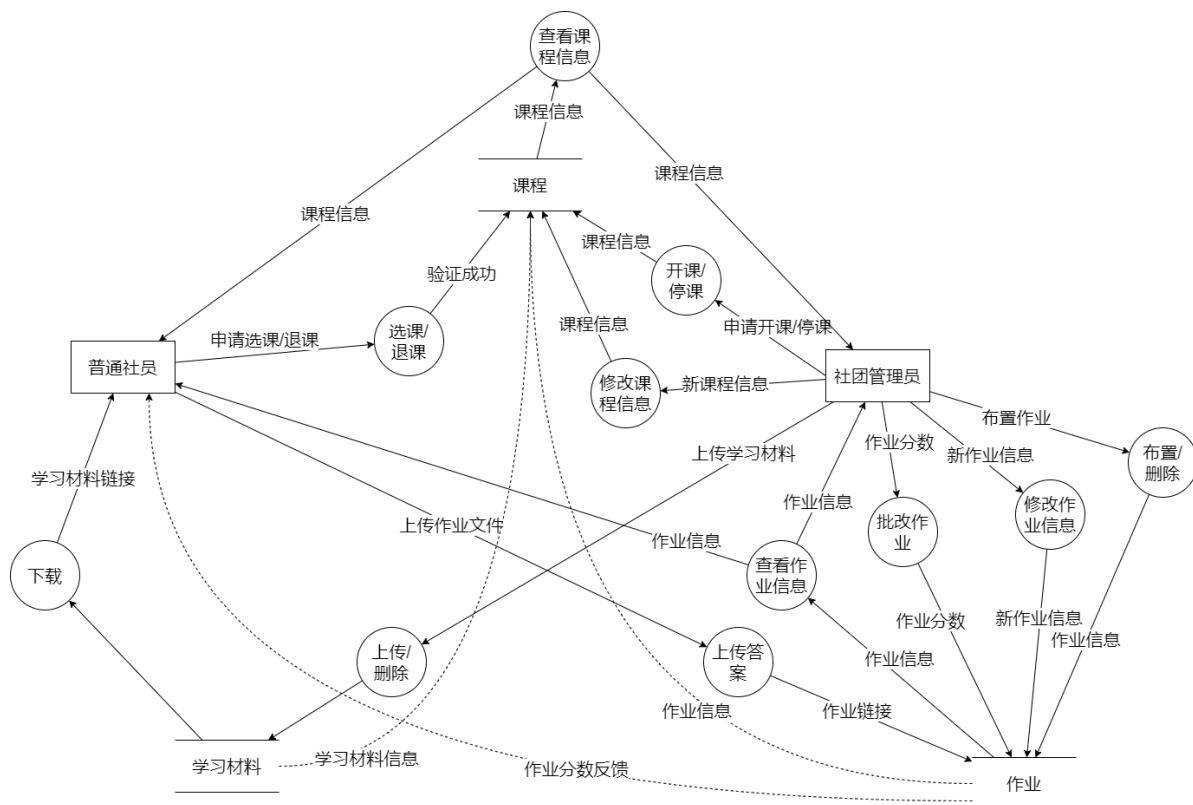
目前超算系统初步成立，需要展示和公告系统展示北航超算社团取得的成就。为此，超算课程平台添加了展示和公告系统，用时间轴的方式将每个时间段中决定北航超算发展的重要节点事件展示出来，便于其他同学了解超算社团。同时展示历届超算成员风采，让更多的同学通过往届队员经历了解超算。

2. 数据流图

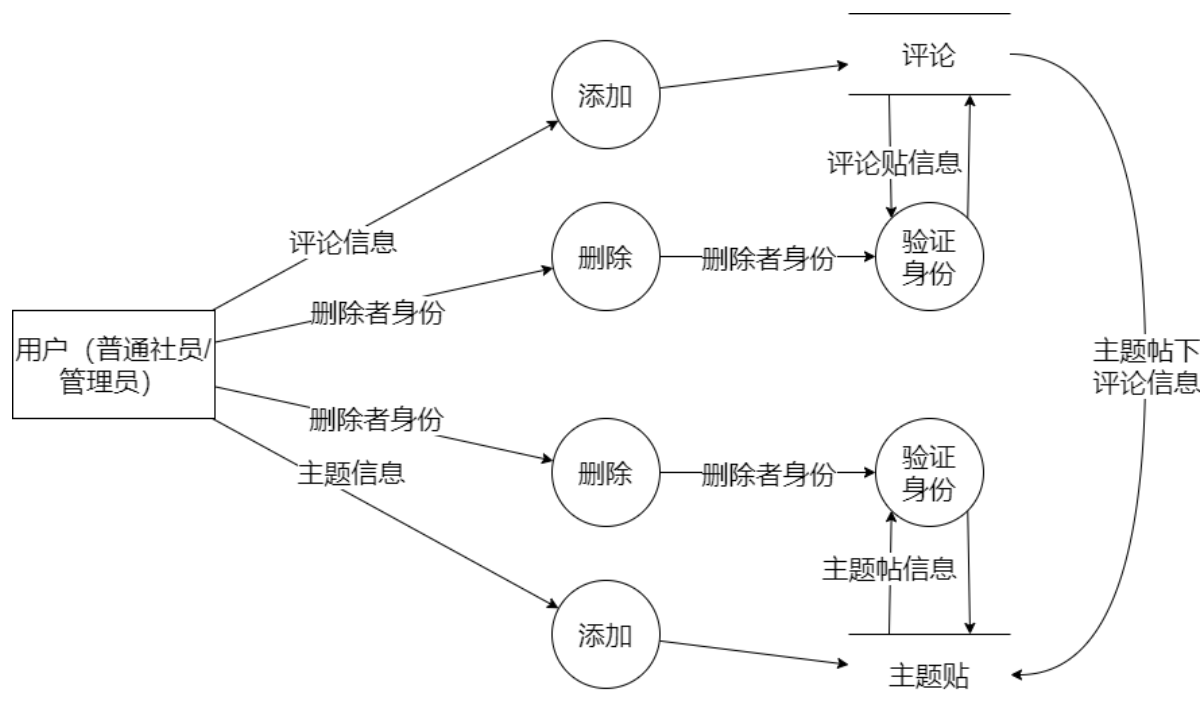
(1) 用户登录数据流图



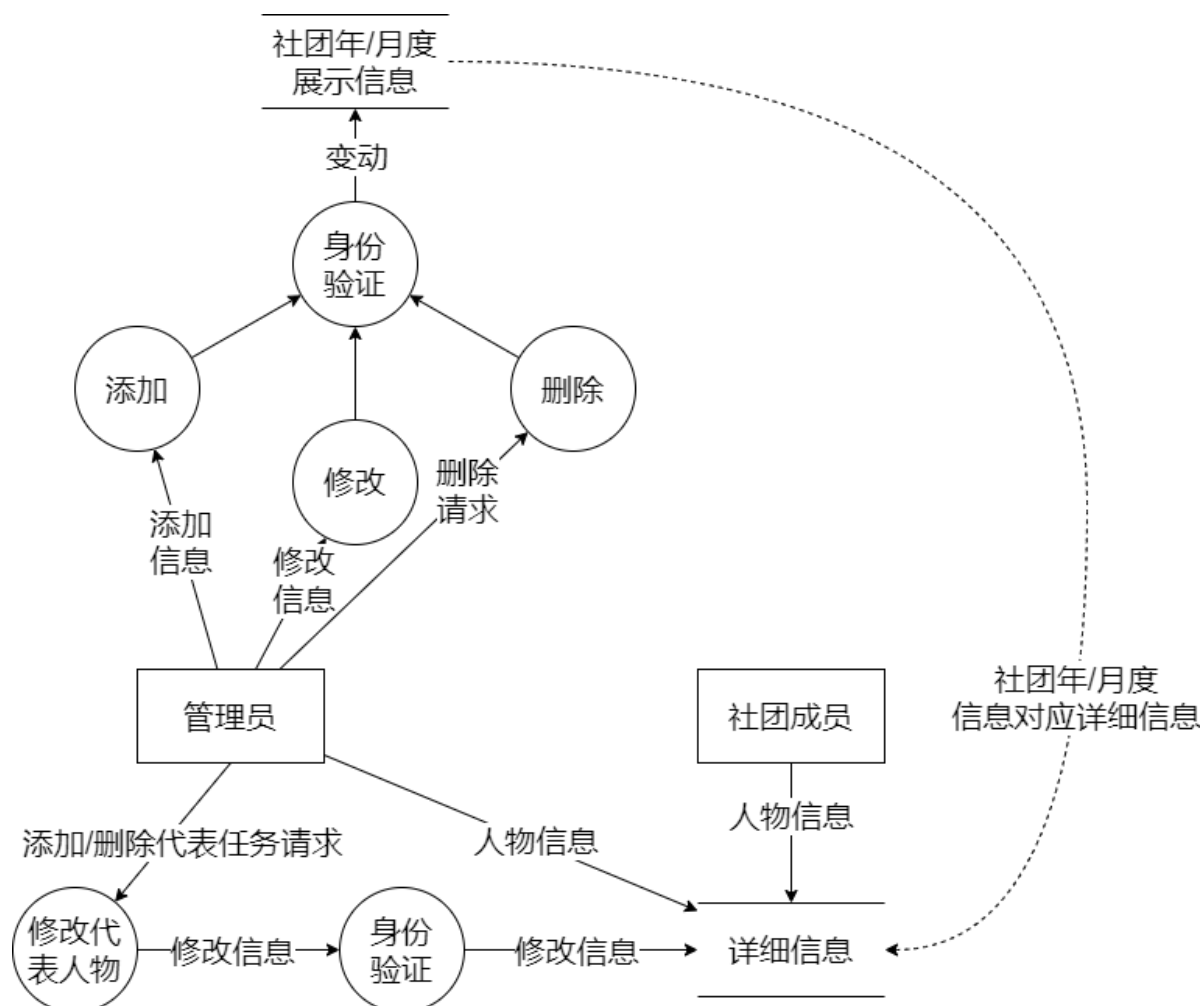
(2) 课程和作业系统数据流图



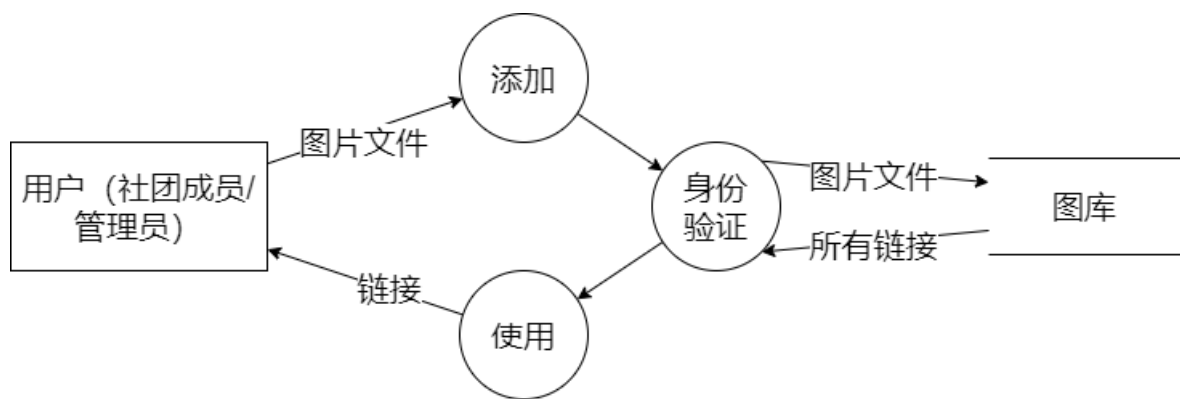
(3) 评价和交流数据流图



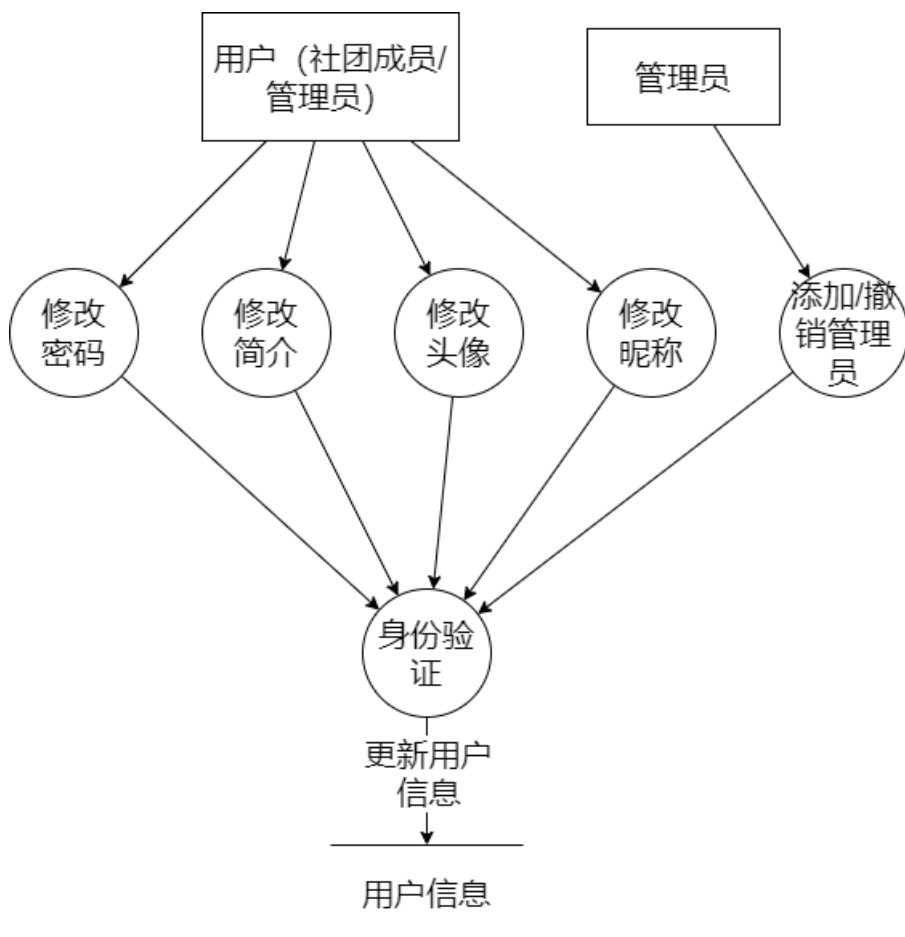
(4) 展示和公告数据流图



(5) 图库数据流图



(6) 用户区数据流图



3. 数据元素图

根据前文分析，在数据流中提取子数据项，得到8个实体表和7个联系表

(1) 实体表

1. 用户信息表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|----------|---------|------|------|----|----|---------|
| username | varchar | 100 | 是 | 是 | 否 | 表示用户的账号 |
| password | varchar | 100 | 是 | 否 | 否 | 表示用户的密码 |
| name | varchar | 50 | 是 | 否 | 否 | 表示用户名 |

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|-------------|---------|------|------|----|----|-------------|
| numOfTitles | int | 10 | 是 | 否 | 否 | 表示用户发帖数量 |
| photo | varchar | 100 | 否 | 否 | 否 | 表示用户头像的路径位置 |
| profile | varchar | 1000 | 否 | 否 | 否 | 表示用户资料 |
| isSuperUser | tinyint | 1 | 是 | 否 | 否 | 表示用户是否是管理员 |

2. 图床表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|----------|---------|------|------|----|----|---------------------------|
| id | int | 10 | 是 | 是 | 否 | 图片的id, 唯一标识图片 |
| position | varchar | 100 | 是 | 否 | 否 | 图片路径, 用于后续找到对应的图片 |
| username | varchar | 100 | 是 | 否 | 是 | 图片对应的用户的账号, 图片将出现在该账号的图床中 |

3. 主题帖表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|-----------|----------|-------|------|----|----|---------------------|
| id | int | 10 | 是 | 是 | 否 | 主题帖id, 唯一标识主题帖 |
| title | varchar | 100 | 是 | 否 | 否 | 主题帖标题, 方便后续查找主题帖 |
| content | varchar | 10000 | 否 | 否 | 否 | 主题帖内容 |
| time | datetime | 30 | 是 | 否 | 否 | 主题帖发布时间, 以后端记录时间为准 |
| username | varchar | 100 | 是 | 否 | 是 | 主题贴发布人, 只有作者才能删除主题帖 |
| numofcoms | int | 10 | 是 | 否 | 否 | 主题帖下方评论数量 |

4. 评论贴表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|---------|---------|-------|------|----|----|----------------|
| id | int | 10 | 是 | 是 | 否 | 评论贴id, 唯一标识评论贴 |
| content | varchar | 10000 | 否 | 否 | 否 | 评论贴内容 |

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|------------------|----------|------|------|----|----|-------------------|
| time | datetime | 30 | 是 | 否 | 否 | 评论贴发布时间，以后端记录时间为准 |
| username | varchar | 100 | 是 | 否 | 是 | 发帖人账号名，这是外码 |
| commentatee_name | varchar | 50 | 否 | 否 | 否 | 被评论人用户名，这不是外码 |

5. 课程信息表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|-------------|---------|-------|------|----|----|---------------------------------------|
| id | int | 10 | 是 | 是 | 否 | 课程id，唯一标识一门课程 |
| name | varchar | 100 | 是 | 否 | 否 | 课程名称 |
| description | varchar | 10000 | 否 | 否 | 否 | 课程描述，管理在定义课程后可以编写课程描述，记录课程地点、时间、内容等信息 |
| exam | int | 10 | 否 | 否 | 否 | 表示课程评分的占比统计，字段表示考试占比 |
| time | varchar | 100 | 否 | 否 | 否 | 表示上课时间 |
| position | varchar | 100 | 否 | 否 | 否 | 表示上课地点 |

6. 作业信息表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|------------|----------|-------|------|----|----|---|
| id | int | 10 | 是 | 是 | 否 | 作业id，唯一标识一次作业 |
| name | varchar | 100 | 是 | 否 | 否 | 作业名称 |
| content | varchar | 10000 | 否 | 否 | 否 | 作业内容，管理员在布置作业时 可以填写作业描述，描述作业的主要内容、功能、完成作业时可以参考的资料、和注意事项等 |
| begin_time | datetime | 30 | 否 | 否 | 否 | 这次作业的开始时间 |
| end_time | datetime | 30 | 否 | 否 | 否 | 这次作业的结束时间 |

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|--------------|------|------|------|----|----|---------|
| averageScore | int | 10 | 否 | 否 | 否 | 作业的平均分数 |

7. 课程附件表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|------|----------|------|------|----|----|------------------|
| id | int | 10 | 是 | 是 | 否 | 附件id, 唯一标识一个课程附件 |
| name | varchar | 100 | 是 | 否 | 否 | 附件名称 |
| time | datetime | 30 | 是 | 否 | 否 | 附件提交时间 |

8. 社团发展历史信息表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|----------|---------|-------|------|----|----|--------------------------|
| id | int | 10 | 是 | 是 | 否 | 社团发展历史信息id, 唯一标识一个社团发展信息 |
| time | varchar | 100 | 是 | 否 | 否 | 社团发展历史信息时间段 |
| overview | varchar | 10000 | 否 | 否 | 否 | 社团发展概况 |

(2) 联系表

1. 主题帖-评论贴关系表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|------------|------|------|------|----|----|-------------------|
| title_id | int | 10 | 是 | 否 | 是 | 标识一个讨论贴对应的一个主题帖id |
| comment_id | int | 10 | 是 | 是 | 是 | 标识讨论帖 |

2. 课程-作业关系表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|-------------|------|------|------|----|----|-----------------|
| class_id | int | 10 | 是 | 否 | 是 | 标识一次作业对应的一个课程id |
| homework_id | int | 10 | 是 | 是 | 是 | 标识一次作业的作业id |

3. 课程-用户关系表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|----------|---------|------|------|----|----|-------------------|
| username | varchar | 100 | 是 | 是 | 是 | 用户账号，表示一条选课关系中的用户 |
| class_id | int | 10 | 是 | 是 | 是 | 课程id，表示一条选课关系中的课程 |

4. 作业-用户关系表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|-------------|----------|------|------|----|----|---------------------------|
| id | int | 10 | 是 | 是 | 否 | 一次作业提交关系的id，唯一表示了一次作业提交关系 |
| username | varchar | 100 | 是 | 否 | 是 | 用户账号，作业提交关系中的标识作业提交的用户 |
| homework_id | int | 10 | 是 | 否 | 是 | 作业id，作业提交关系中标识用户提交的作业的id |
| score | int | 10 | 否 | 否 | 否 | 成绩，标识用户此次作业提交的成绩 |
| last_time | datetime | 30 | 是 | 否 | 否 | 上传时间，以后端记录的时间为准 |
| filename | varchar | 500 | 是 | 否 | 否 | 上传的作业 |

5. 课程-课程附件关系表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|-------------|------|------|------|----|----|--------------------|
| class_id | int | 10 | 是 | 否 | 是 | 课程id，表示课程-附件关系中的课程 |
| material_id | int | 10 | 是 | 是 | 是 | 附件id，表示课程-附件关系中的附件 |

6. 社团发展历史信息-成员关系表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|------|------|------|------|----|----|----|
|------|------|------|------|----|----|----|

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|------------|---------|------|------|----|----|--------------------------------------|
| develop_id | int | 10 | 是 | 是 | 是 | 社团发展历史信息id，在社团发展信息-成员关系中标识一个社团发展阶段 |
| username | varchar | 100 | 是 | 是 | 是 | 用户账号，在社团发展信息-成员关系中标识一个用户，表示用户在此发展阶段中 |

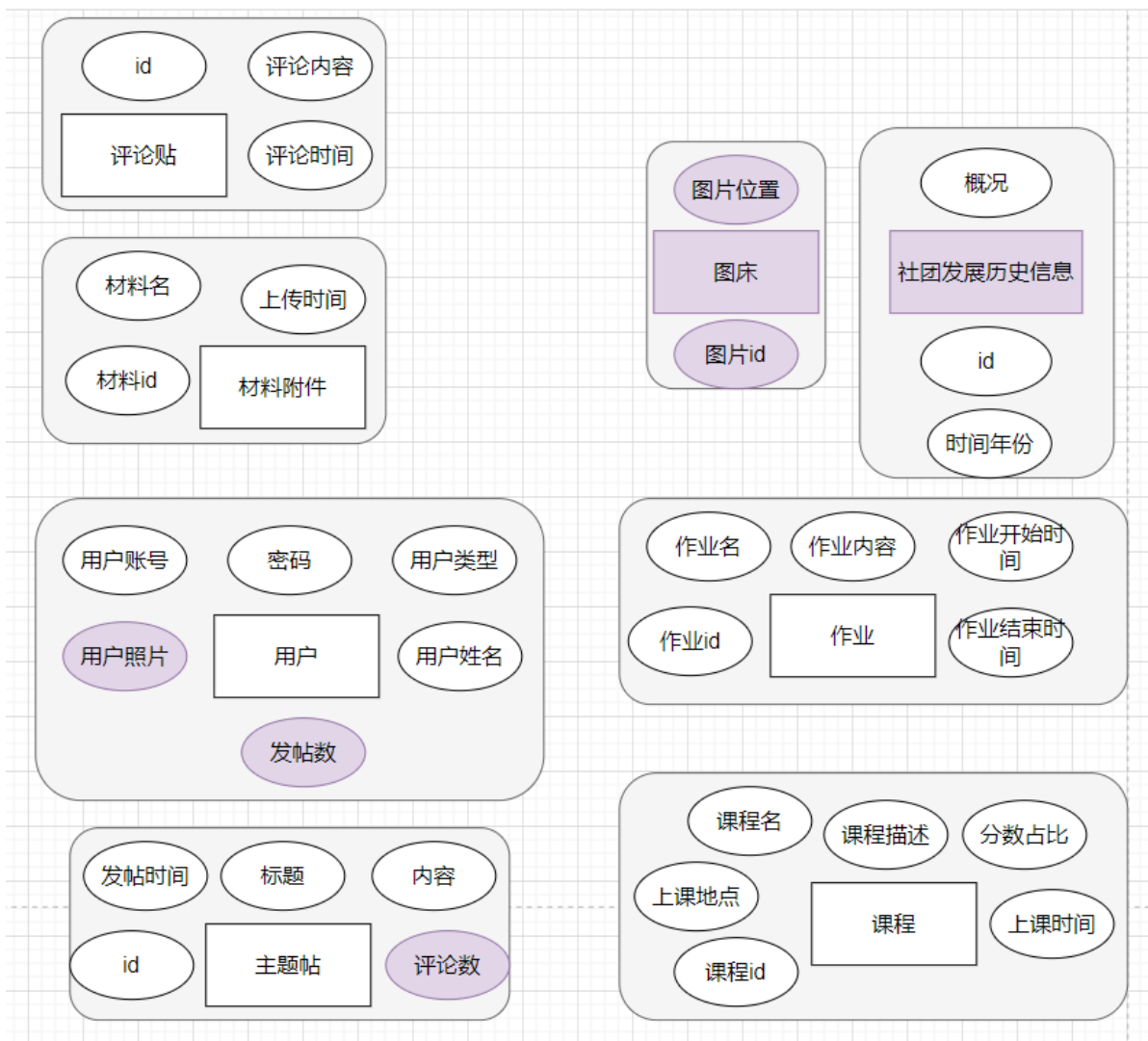
7. 社团发展信息历史信息-图片关系表

| 字段名称 | 数据类型 | 字段大小 | 必填字段 | 主键 | 外键 | 说明 |
|------------|---------|------|------|----|----|--|
| pic_url | varchar | 100 | 是 | 是 | 否 | 图片url，表示一个照片的链接。在社团发展历史阶段-图片关系中表示一个发展阶段中的图片。 |
| develop_id | int | 10 | 是 | 是 | 是 | 社团发展历史信息id，在社团发展阶段-图片关系中表示一个发展阶段 |

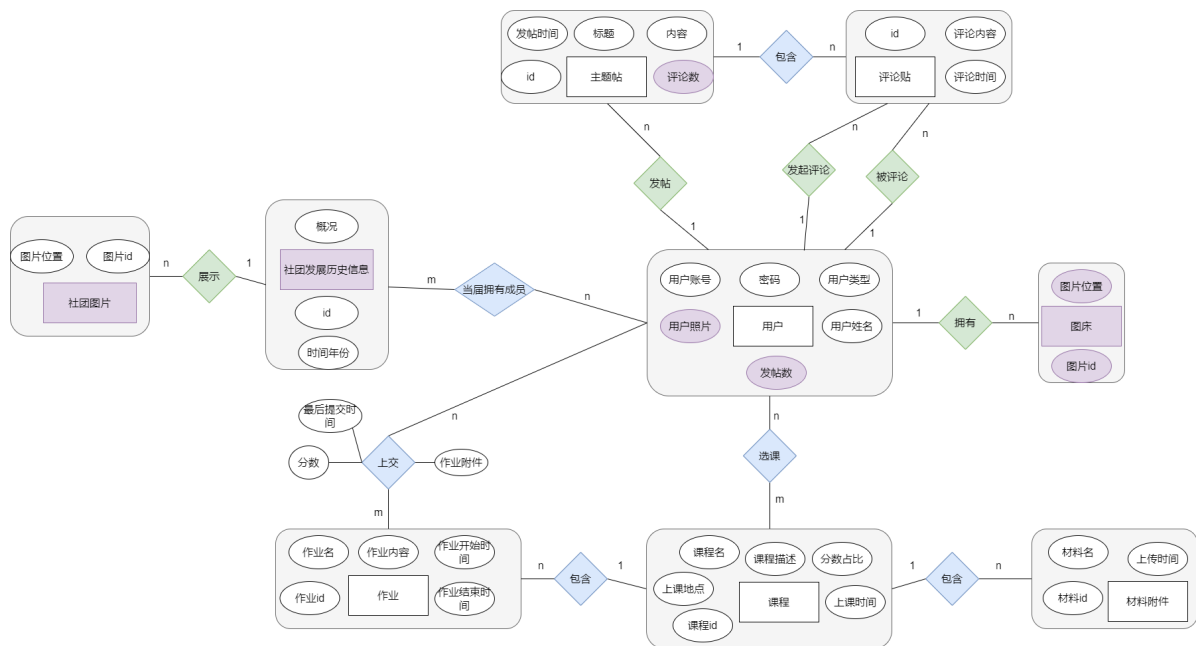
二、数据库概念模式设计

1. 系统初步 E-R 图

(1) 实体 E-R 图



2. 系统基本 E-R 图



三、数据库逻辑模式设计

1. 数据库关系模式

根据系统全局ER图，我们设计了15个关系模式，每个关系模式都是一个五元组：

$$R(U, D, DOM, F)$$

这里：

- 关系名 R 是符号化的元组语义
- U 是组成该关系的属性集合
- D 为属性组 U 中的属性所来自的域
- DOM 为属性到域的映射
- F 为属性组 U 的一组函数依赖

在下面的设计分析中，为了简洁起见，只讨论三元组 $R(U, F)$ 。

1. tb_user

- 关系名 R : *tb_user*
- 属性集合 U :

| 属性名 | 中文含义 |
|-----------------|--------|
| <u>username</u> | 用户账号 |
| password | 用户密码 |
| name | 用户姓名 |
| numOfTitles | 用户发帖数 |
| photo | 用户照片 |
| profile | 用户简介 |
| isSuperUser | 是否为管理员 |

- 函数依赖集合 F :
 - $username \rightarrow password$
 - $username \rightarrow name$
 - $username \rightarrow numOfTitles$
 - $username \rightarrow photo$
 - $username \rightarrow profile$
 - $username \rightarrow isSuperUser$

2. tb_pics

- 关系名 R : *tb_pics*
- 属性集合 U :

| 属性名 | 中文含义 |
|-----------|------|
| <u>id</u> | 图片id |
| position | 图片位置 |

| 属性名 | 中文含义 |
|----------|--------|
| username | 图片所属用户 |

- 函数依赖集合 F :
 - $id \rightarrow position$
 - $id \rightarrow username$

3. dc_title

- 关系名 R : dc_title
- 属性集合 U :

| 属性名 | 中文含义 |
|-----------|-------|
| <u>id</u> | 主题帖id |
| title | 标题 |
| content | 内容 |
| time | 发帖时间 |
| username | 发帖用户 |
| numofcoms | 贴内评论数 |

- 函数依赖集合 F :
 - $id \rightarrow title$
 - $id \rightarrow content$
 - $id \rightarrow time$
 - $id \rightarrow username$
 - $id \rightarrow numofcoms$

4. dc_comment

- 关系名 R : $dc_comment$
- 属性集合 U :

| 属性名 | 中文含义 |
|------------------|--------|
| <u>id</u> | 评论贴id |
| content | 内容 |
| time | 发帖时间 |
| username | 发帖用户 |
| commentatee_name | 被评论者名称 |

- 函数依赖集合 F :
 - $id \rightarrow content$
 - $id \rightarrow time$

- $id \rightarrow username$
- $id \rightarrow commentatee_name$

5. dc_com2title

- 关系名 R : $dc_com2title$
- 属性集合 U :

| 属性名 | 中文含义 |
|------------|-------|
| title_id | 主题帖id |
| comment_id | 评论贴id |

- 函数依赖集合 F :
 - $comment_id \rightarrow title_id$

6. cl_class

- 关系名 R : cl_class
- 属性集合 U :

| 属性名 | 中文含义 |
|-------------|--------|
| <u>id</u> | 课程id |
| name | 课程名称 |
| description | 课程描述 |
| pingshi | 课程平时占比 |
| exam | 课程考试占比 |
| time | 上课时间 |
| position | 上课地点 |

- 函数依赖集合 F :
 - $id \rightarrow name$
 - $id \rightarrow description$
 - $id \rightarrow pingshi$
 - $id \rightarrow exam$
 - $id \rightarrow time$
 - $id \rightarrow position$

7. cl_homework

- 关系名 R : $cl_homework$
- 属性集合 U :

| 属性名 | 中文含义 |
|-----------|------|
| <u>id</u> | 作业id |

| 属性名 | 中文含义 |
|--------------|--------|
| name | 作业名称 |
| content | 作业内容 |
| begin_time | 作业开始时间 |
| end_time | 作业截止时间 |
| averageScore | 作业平均分 |

- 函数依赖集合 F :
 - $id \rightarrow name$
 - $id \rightarrow content$
 - $id \rightarrow begin_time$
 - $id \rightarrow end_time$
 - $id \rightarrow averageScore$

8. cl_class_homework

- 关系名 R : *cl_class_homework*
- 属性集合 U :

| 属性名 | 中文含义 |
|--------------------|------|
| class_id | 课程id |
| <u>homework_id</u> | 作业id |

- 函数依赖集合 F :
 - $homework_id \rightarrow class_id$

9. cl_class_user

- 关系名 R : *cl_class_user*
- 属性集合 U :

| 属性名 | 中文含义 |
|----------|------|
| username | 用户账号 |
| class_id | 课程id |

- 函数依赖集合 F : 码为全码，省略依赖关系 F 中的主属性对码的完全函数依赖。

10. cl_homework_user

- 关系名 R : *cl_homework_user*
- 属性集合 U :

| 属性名 | 中文含义 |
|-----------|--------|
| <u>id</u> | 作业文件id |

| 属性名 | 中文含义 |
|-------------|---------|
| username | 提交者 |
| homework_id | 对应的作业id |
| score | 作业得分 |
| last_time | 最后提交时间 |
| filename | 作业文件名 |

- 函数依赖集合 F :
 - $id \rightarrow username$
 - $id \rightarrow homework_id$
 - $id \rightarrow score$
 - $id \rightarrow last_time$
 - $id \rightarrow filename$

11. cl_material

- 关系名 R : *cl_material*
- 属性集合 U :

| 属性名 | 中文含义 |
|-----------|--------|
| <u>id</u> | 课件附件id |
| name | 附件文件名 |
| time | 附件上传时间 |

- 函数依赖集合 F :
 - $id \rightarrow name$
 - $id \rightarrow time$

12. cl_class_material

- 关系名 R : *cl_class_material*
- 属性集合 U :

| 属性名 | 中文含义 |
|--------------------|--------|
| class_id | 课程id |
| <u>material_id</u> | 课程附件id |

- 函数依赖集合 F :
 - $material_id \rightarrow class_id$

13. an_develops

- 关系名 R : $an_develops$
- 属性集合 U :

| 属性名 | 中文含义 |
|-----------|---------|
| <u>id</u> | 社团信息id |
| time | 信息所属时间段 |
| overview | 社团发展概况 |

- 函数依赖集合 F :
 - $id \rightarrow time$
 - $id \rightarrow overview$

14. an_develop_member

- 关系名 R : $an_develop_member$
- 属性集合 U :

| 属性名 | 中文含义 |
|------------|--------|
| develop_id | 社团信息id |
| username | 成员用户账号 |

- 函数依赖集合 F : 码为全码，省略依赖关系 F 中的主属性对码的完全函数依赖。

15. an_pics

- 关系名 R : an_pics
- 属性集合 U :

| 属性名 | 中文含义 |
|------------|--------|
| pic_url | 图片存储位置 |
| develop_id | 社团信息id |

- 函数依赖集合 F : 码为全码，省略依赖关系 F 中的主属性对码的完全函数依赖。

2. 关系模式范式等级的判定与规范化

1. tb_user

在关系模式 `tb_user` 中，满足：

- 每一个非主属性都完全函数依赖于所有候选码（`username`），因此符合2NF。
- 不存在非主属性对码（`username`）的传递函数依赖，因此符合3NF。

2. tb_pics

在关系模式 `tb_pics` 中，满足：

- 每一个非主属性都完全函数依赖于所有候选码（`id`），因此符合2NF。
- 不存在非主属性对码（`id`）的传递函数依赖，因此符合3NF。

3. dc_title

在关系模式 `dc_title` 中，满足：

- 每一个非主属性都完全函数依赖于所有候选码（`id`），因此符合2NF。
- 不存在非主属性对码（`id`）的传递函数依赖，因此符合3NF。

4. dc_comment

在关系模式 `dc_comment` 中，满足：

- 每一个非主属性都完全函数依赖于所有候选码（`id`），因此符合2NF。
- 不存在非主属性对码（`id`）的传递函数依赖，因此符合3NF。

5. dc_com2title

在关系模式 `dc_com2title` 中，满足：

- 每一个非主属性都完全函数依赖于所有候选码（`comment_id`），因此符合2NF。
- 不存在非主属性对码（`comment_id`）的传递函数依赖，因此符合3NF。

6. cl_class

在关系模式 `cl_class` 中，满足：

- 每一个非主属性都完全函数依赖于所有候选码（`id`），因此符合2NF。
- 不存在非主属性对码（`id`）的传递函数依赖，因此符合3NF。

7. cl_homework

在关系模式 `cl_homework` 中，满足：

- 每一个非主属性都完全函数依赖于所有候选码（`id`），因此符合2NF。
- 不存在非主属性对码（`id`）的传递函数依赖，因此符合3NF。

8. cl_class_homework

在关系模式 `cl_class_homework` 中，满足：

- 每一个非主属性都完全函数依赖于所有候选码（`homework_id`），因此符合2NF。
- 不存在非主属性对码（`homework_id`）的传递函数依赖，因此符合3NF。

9. cl_class_user

在关系模式 `cl_class_user` 中，由于是全码，因此没有非主属性，因此一定不存在非主属性对码的部分和传递函数依赖，因此符合3NF。

10. cl_homework_user

在关系模式 `cl_homework_user` 中, 满足:

- 每一个非主属性都完全函数依赖于所有候选码 (`id`), 因此符合2NF。
- 不存在非主属性对码 (`id`) 的传递函数依赖, 因此符合3NF。

11. cl_material

在关系模式 `cl_material` 中, 满足:

- 每一个非主属性都完全函数依赖于所有候选码 (`id`), 因此符合2NF。
- 不存在非主属性对码 (`id`) 的传递函数依赖, 因此符合3NF。

12. cl_class_material

在关系模式 `cl_class_material` 中, 满足:

- 每一个非主属性都完全函数依赖于所有候选码 (`material_id`), 因此符合2NF。
- 不存在非主属性对码 (`material_id`) 的传递函数依赖, 因此符合3NF。

13. an_develops

在关系模式 `an_develops` 中, 满足:

- 每一个非主属性都完全函数依赖于所有候选码 (`id`), 因此符合2NF。
- 不存在非主属性对码 (`id`) 的传递函数依赖, 因此符合3NF。

14. an_develop_member

在关系模式 `an_develop_member` 中, 由于是全码, 因此没有非主属性, 因此一定不存在非主属性对码的部分和传递函数依赖, 因此符合3NF。

15. an_pics

在关系模式 `an_pics` 中, 由于是全码, 因此没有非主属性, 因此一定不存在非主属性对码的部分和传递函数依赖, 因此符合3NF。

3. 数据库设计优化

(1) 增加索引

在数据库系统中, 增加索引可以实现对数据的快速检索。因此, 我们对经常进行查询操作的字段增加索引信息, 提高存取效率。对于主键和外键, Mysql会自动创建主键和外键索引, 因此不在下方列出。

```
1 create index ti on dc_title(title);
2 create index pos on tb_pics(position);
```

(2) 增加外键

为了保证数据的一致性, 以及关系模型的参照完整性, 为每个联系的属性增加外键约束。同时增加级联删除功能, 当主表的记录被删除时, 联系表会自动删除与主表删除记录相关联的记录。以主题帖表为例, 在创建表的语句中, 增加:

```
1 | foreign key(username) references tb_user(username) on delete cascade
```

(3) 模式分解与合并

对于码相同的关系，可以将其合并。例如两个表存在1:n的联系时，就可以省去联系表，将1端关系的id存在n端关系中，节省存储空间。例如在社团图片表中，一张图片对应唯一一个社团信息，因此将社团信息id存入该表。

但为了提高存取效率和存储空间的利用率，对某些1:n的联系不进行合并，仍单独建立联系表，例如主题帖与评论贴属于1:n的联系，由于主题帖id和评论贴id经常被一起使用，评论贴中一个元组占用空间较大，因此有必要单独建立联系表。