

《数据库系统原理》大作业

系统设计报告

题目名称：学海伴航——教辅平台

学号及姓名：

22371212 郑广懿

一、需求分析

1.1 需求描述

1.1.1 背景调研

通过积极调研新生需求，了解学院教学需要，同时与辅导员老师沟通，我们发现传统的微信QQ群学习小组的答疑教辅模式已不再高效。建立功能更丰富，交互更便捷，上手更简单，连接更紧密的教辅平台需求迫在眉睫。

以下为传源书院计划开发的微信答疑小程序：

系统部



新设部门！
等你前来！
大展身手！



前端：

- 设计和实现小程序的界面
- 实现响应式布局
- 处理用户输入、动态效果、动画等

中端：

- 设计和开发小程序的API接口
- 与前端、后端对接数据

后端：

- 实现业务逻辑、数据验证、安全控制
- 管理数据库，保证数据安全和一致性
- 处理服务器部署与维护



助力全面发展

1.1.2 用户调研

对于学生与辅导员的诉求，我们通过联系传源书院的辅导员，获取了他们问卷调查的数据，初步总结需求如下：

学生	辅导员	管理人员
提出问题，并得到及时回答	避免过度同质化的提问	了解同学们的学习情况
获取各科目资料集合，进行自学自测	避免答疑活动对课下生活的打扰	了解同学们的学习诉求
看见其他同学的答疑内容	便于组织模拟考试并加以批改	统计辅导员的工作量
		开展模拟考试
		发布公告

1.1.3 需求总结

对于需要获取学业帮扶的学生

他们需要一款产品，能够：

- 提出课业上的问题，并获得及时准确的回答。
- 看见其它同学的答疑内容。
- 获取各科习题资料，包括往年资料。
- 有条理地获取自己的学习做题记录。
- 参与模拟测试，并及时获取答题情况。

对于辅导员

他们需要一款产品，能够：

- 避免对日常生活的过度打扰。
- 避免过度同质化的提问。
- 获得同学们对回答的反馈。
- 及时更新题库
- 发布模拟考试并加以批改

对于学业支持中心/导员

他们需要一款产品，能够：

- 直观了解辅导员工作情况，方便监督。
- 自动统计辅导员的工作量并换算为志愿时长。
- 开展模拟考试等集体学习活动，并发布公告。

1.1.4 项目优势

通过上述需求分析，可见，**学海伴航**具有庞大的需求空间，光明的应用前景，具有开发的价值，通过我们的平台，**三类目标用户**可以在多种场景下获得利好：

从学生的角度

- 共享学习资料：平台统一整合学习资料，方便同学查阅下载。
- 共享问题与解答：所有问题均公开共享，同学可以搜索自己困惑的相似的问题，从其他人的已解决问题和回答得到启发。
- 自主学习：平台组织统一模拟测试，学生获取模拟情况，指导自主学习。
- 激励学习：可视化自己解决的问题数量，同时在主页获取推荐题目，学习更有动力，更有积极性。
- 减少微信私聊数量：学生反馈群聊数量过多的问题，一个线下答疑活动需要建立很多的群聊，大群、小组群、互助群，影响日常的消息接收，带来不必要的麻烦。通过平台进行信息整合，交流与信息统一化。

从小导师角度

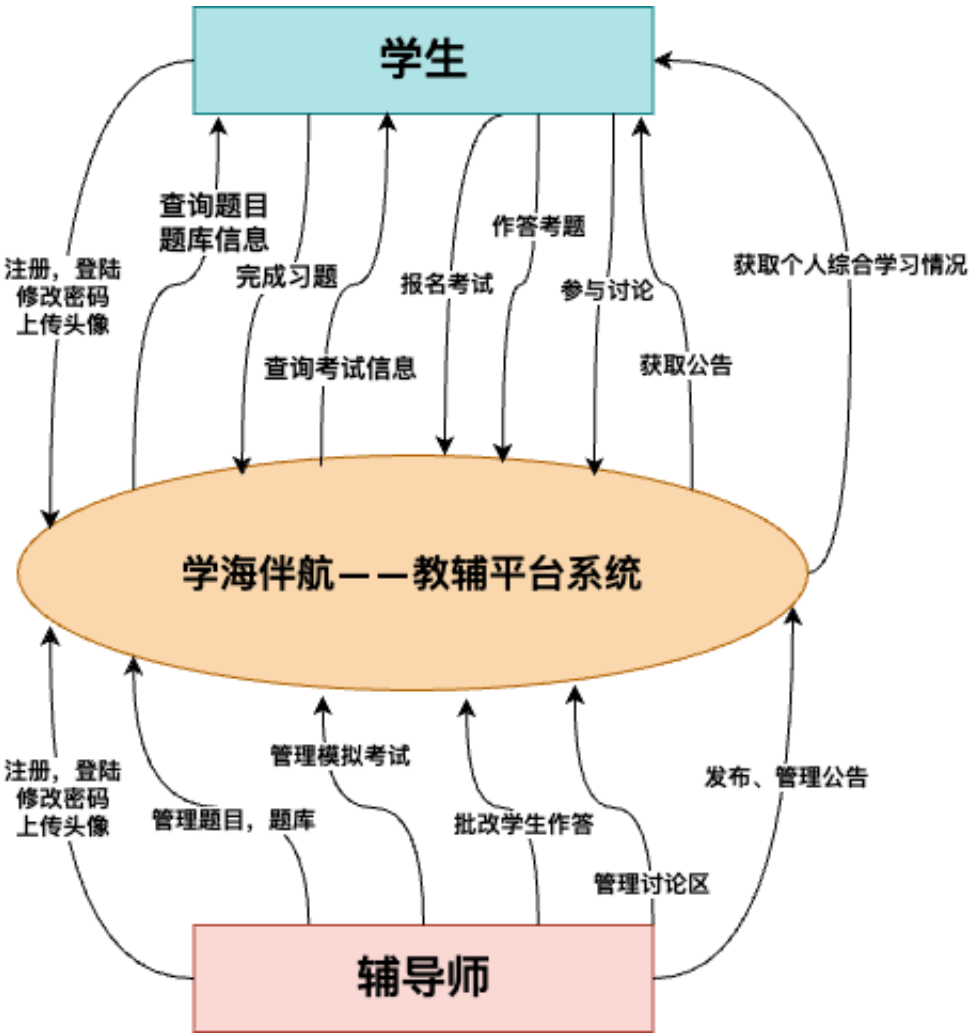
- 辅导积极性更高：基于回答问题，上传题目，组织考试，批改试题的次数分配志愿时长，奖励机制更加公平合理，增强辅导师的解答积极性。
- 在平台统一答疑，避免占用辅导师过多课下私人时间。

从学院组织者角度

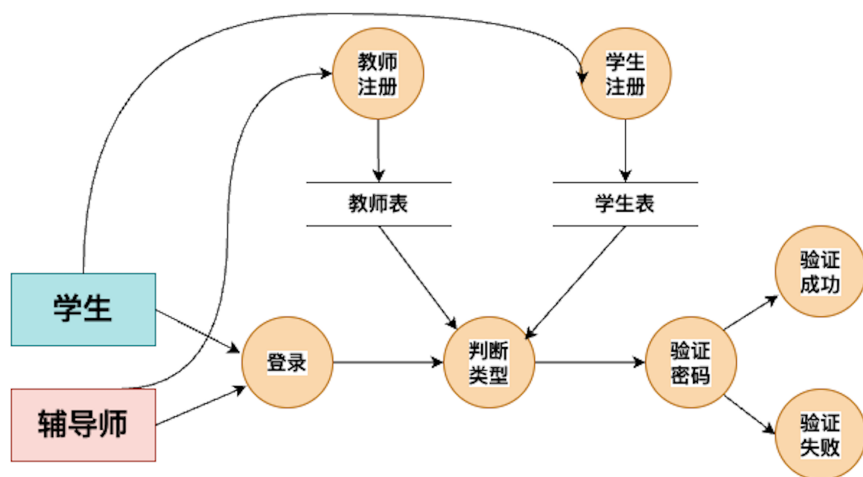
- 活动推进：可视化活动活跃度情况，学生参与度情况，有效推进活动开展。
- 氛围营造：通过平台统一管理活动，定期发布相关学习公告，营造良好的学习氛围。

1.2 数据流图

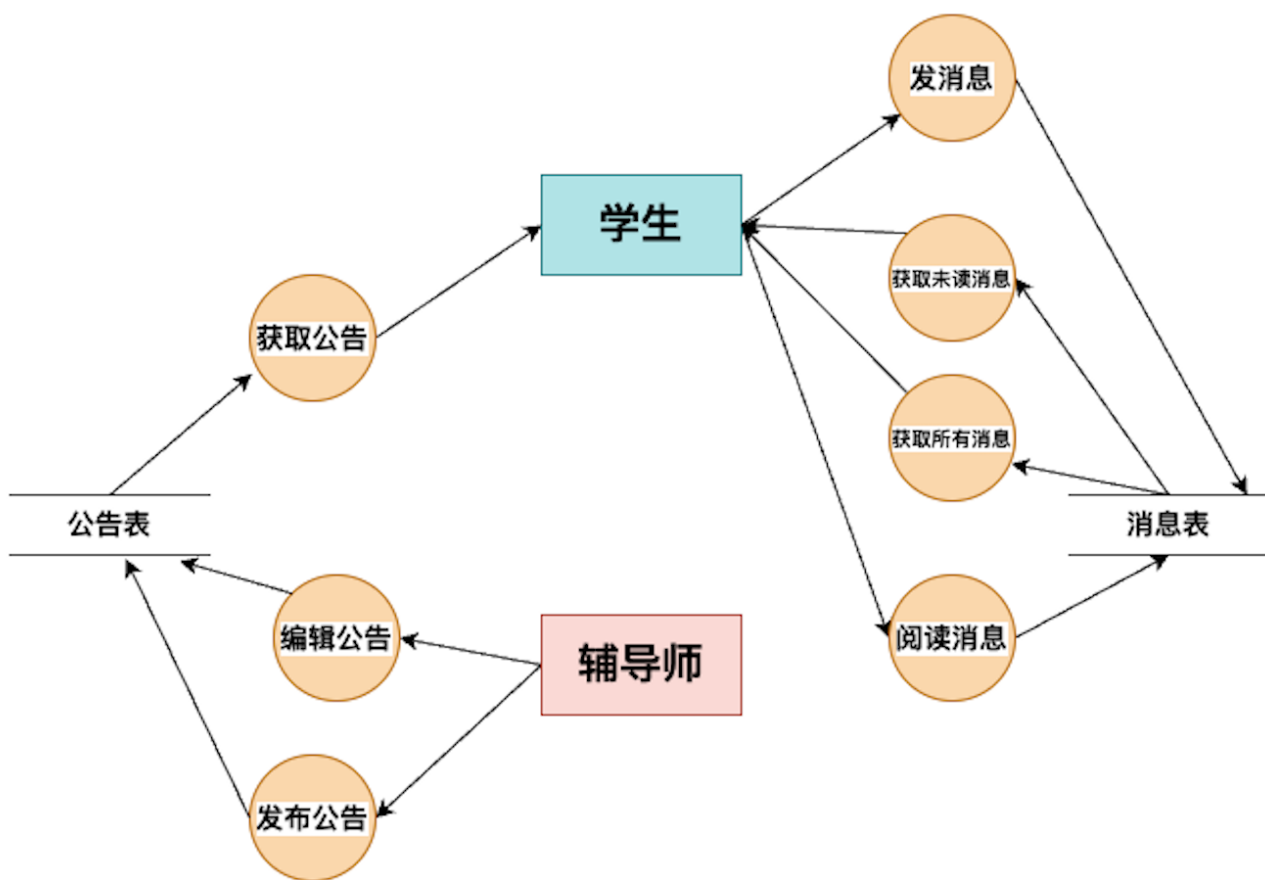
1.2.1 总体数据流图



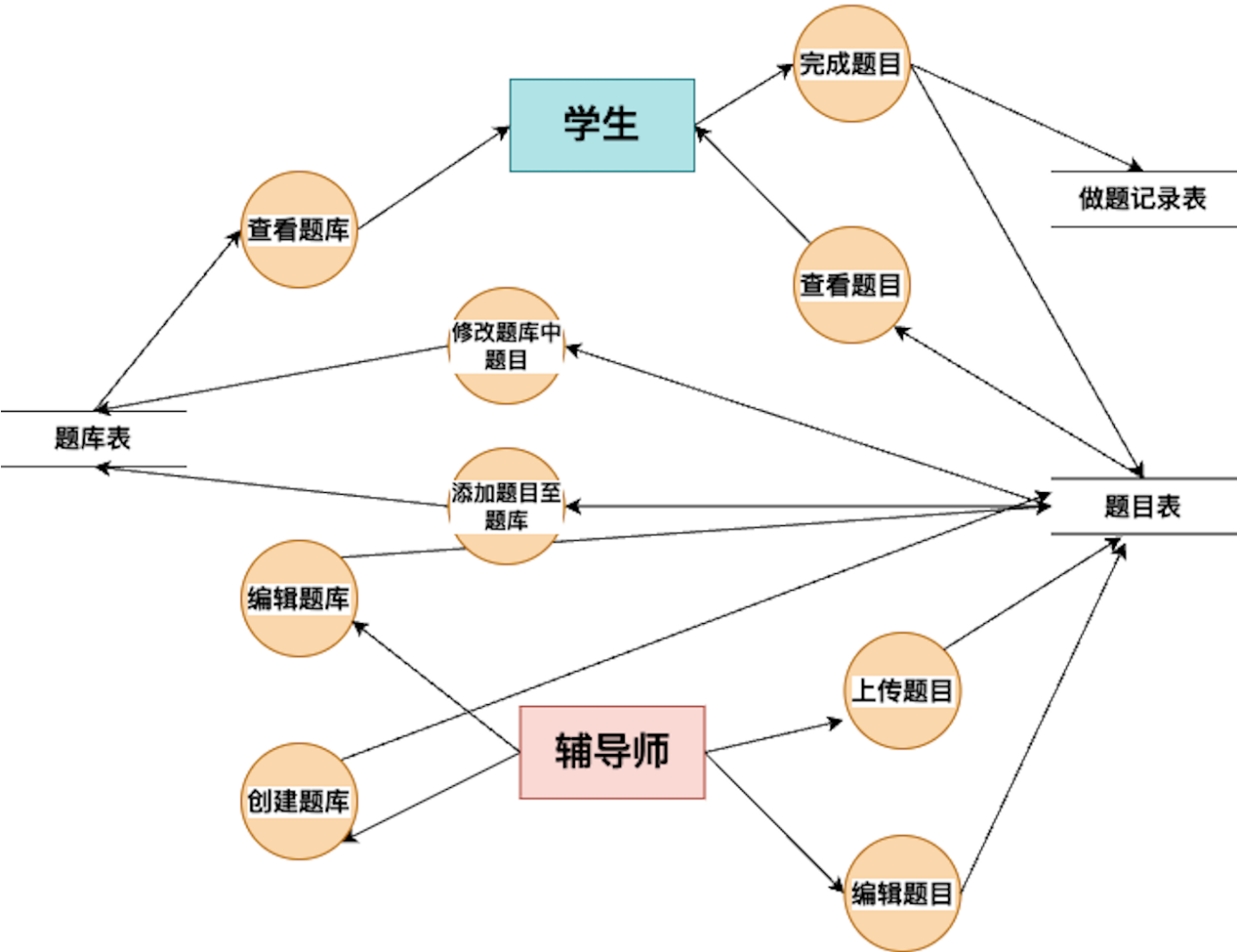
1.2.2 用户管理部分数据流图



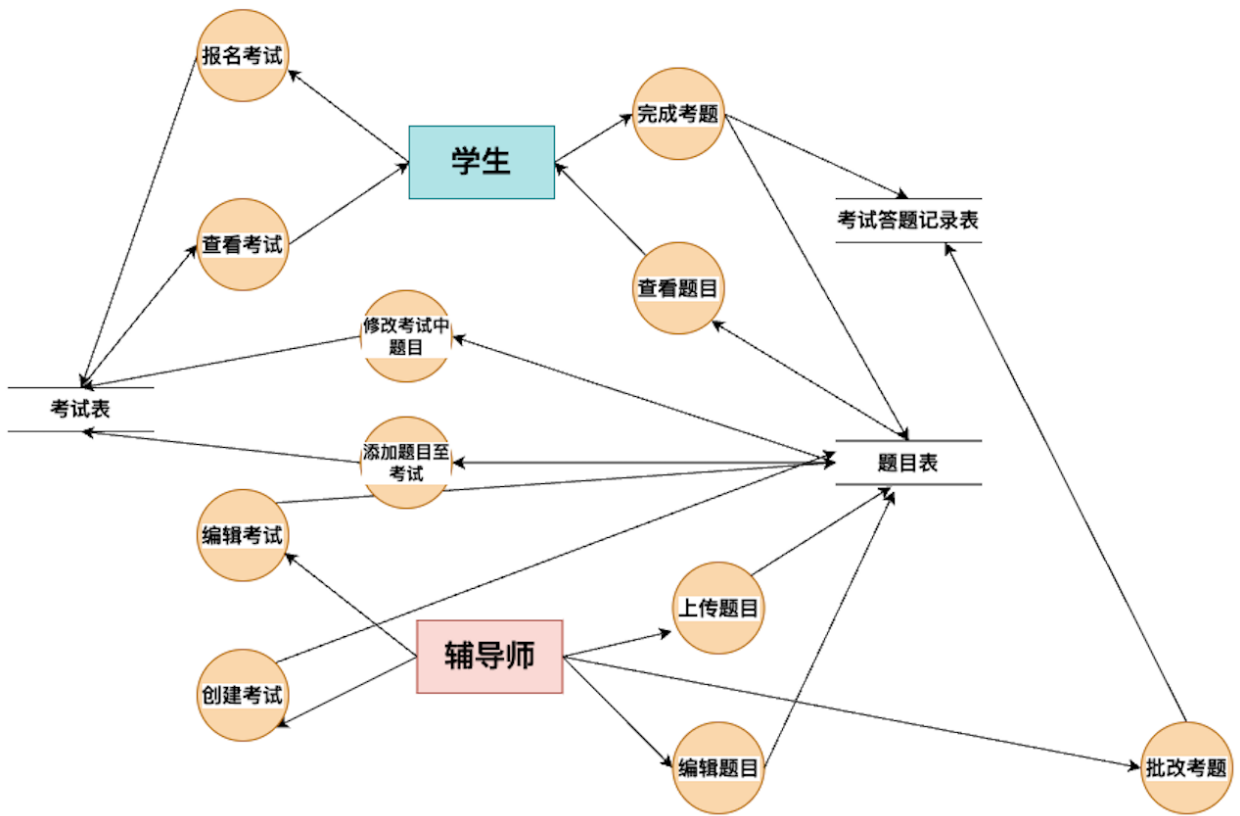
1.2.3 消息公告部分数据流图



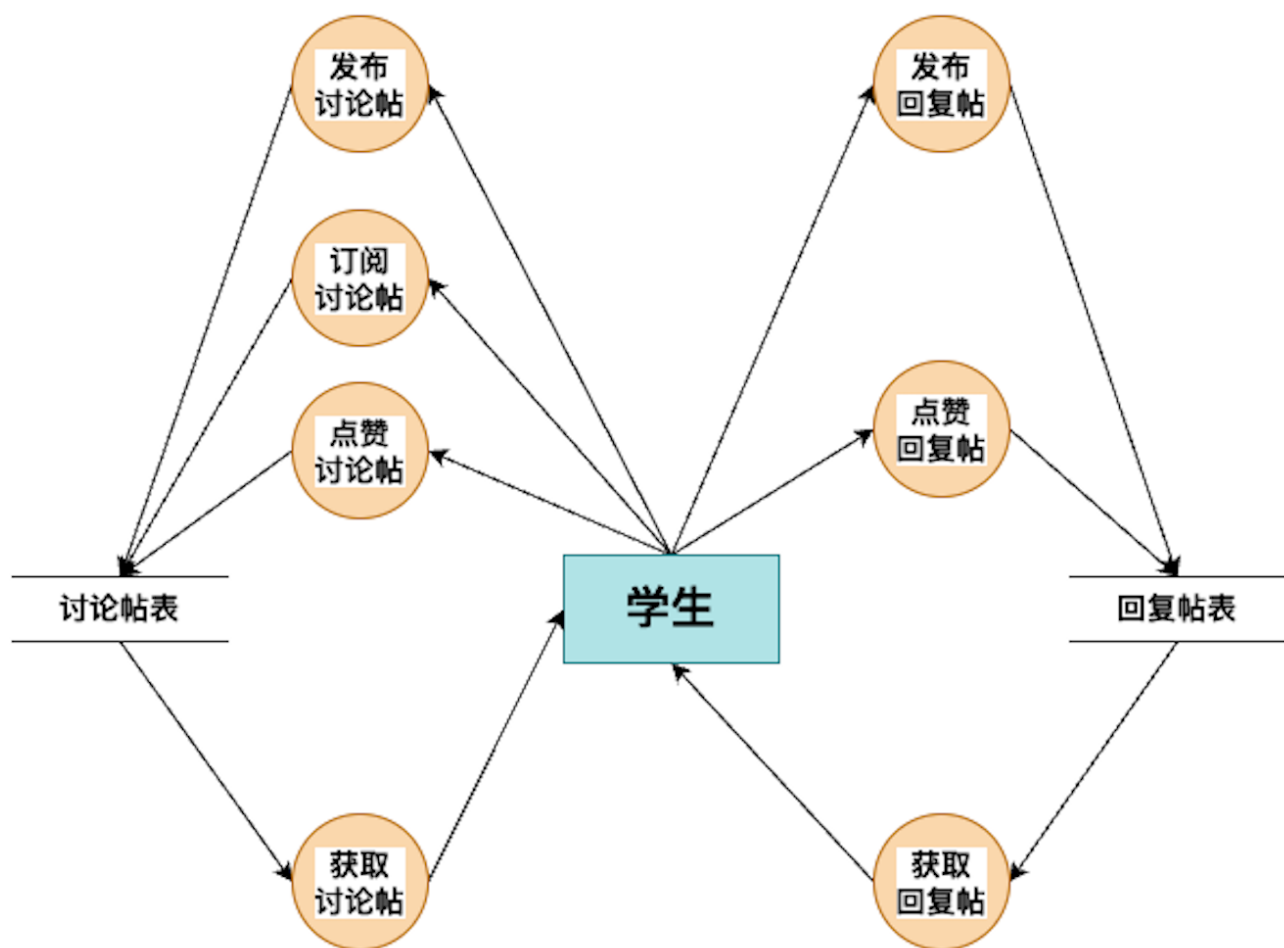
1.2.4 题目题库部分数据流图



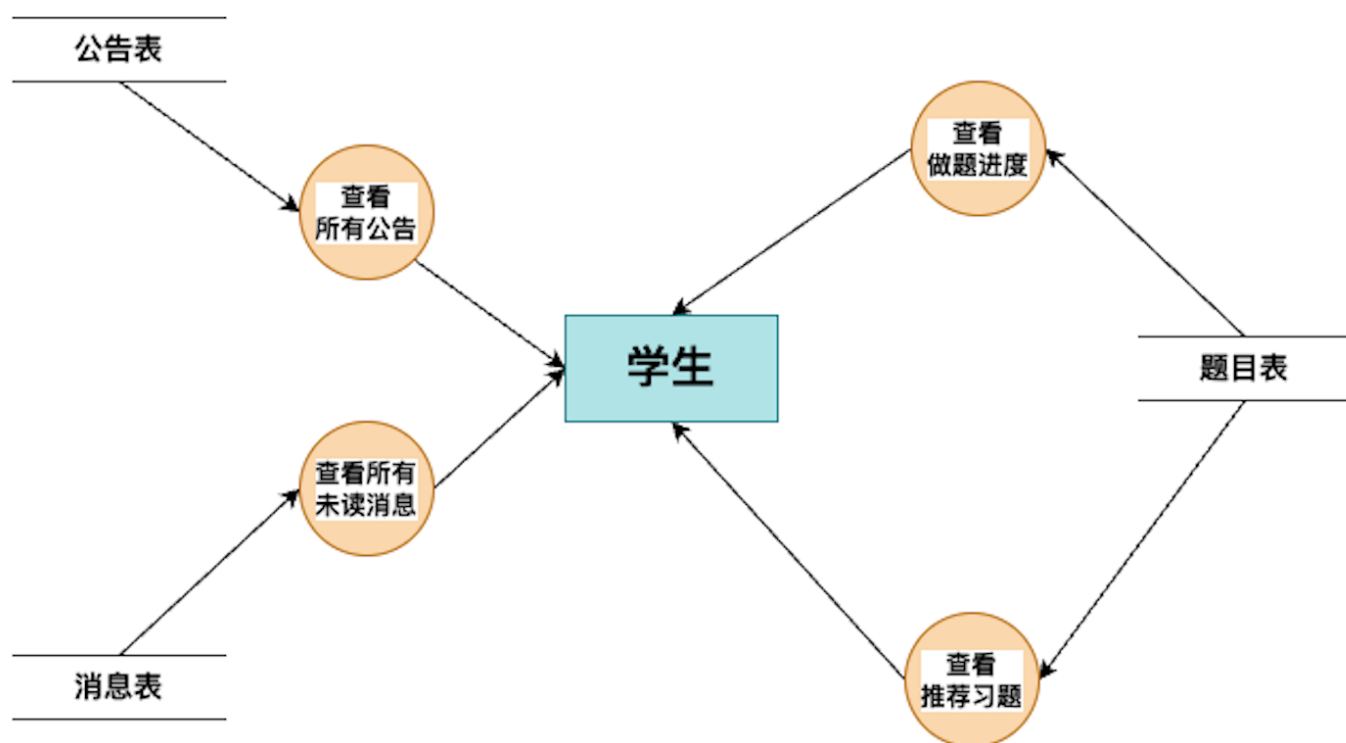
1.2.5 模拟考试部分数据流图



1.2.6 讨论区部分数据流图



1.2.7 用户主页部分数据流图



1.3 数据元素表

1.3.1 用户管理部分

用户表

字段名称	数据类型	可否为空	说明
id	int	NO	用户id，主键
user_id	varchar	NO	学号
pass_word	varchar	NO	密码
mail	varchar	NO	邮箱
college	varchar	NO	学院
entryYear	varchar	NO	入学年份
avatar	varchar	NO	头像
user_role	varchar	NO	用户身份（学生/辅导师）

1.3.2 消息部分

消息表

字段名称	数据类型	可否为空	说明
id	int	NO	消息id，主键
sender	int	NO	外键，发送消息学生编号
sender_avatar	varchar	NO	发送者头像
sent_at	date	NO	发送时间
receiver	int	NO	外键，接收消息学生编号
content	varchar	NO	消息内容
is_read	boolean	NO	是否已阅

1.3.3 公告部分

公告表

字段名称	数据类型	可否为空	说明
id	int	NO	公告id, 主键
sender	int	NO	外键, 发布者编号
sender_avatar	varchar	NO	发布者头像
sent_at	date	NO	发布时间
last_updated	date	NO	时间
content	varchar	NO	公告内容
title	varchar	NO	公告标题

1.3.4 题目部分

题目表

字段名称	数据类型	可否为空	说明
id	int	NO	题目id, 主键
type	varchar	NO	题目类型
content	varchar	NO	题目内容
subject	varchar	NO	题目所属科目
added_at	date	NO	上传时间
source	varchar	NO	题目来源
tags	json	YES	题目标签
difficulty	int	NO	题目难度
added_by	int	NO	题目添加者id
question_banks	json	YES	外键, 题目所属题库
option_count	int	NO	题目选项数量

题库表

字段名称	数据类型	可否为空	说明
id	int	NO	题库id, 主键
subject	varchar	NO	题库所属科目
estimated_time	date	NO	题库预计完成时间
created_at	date	NO	题库创建时间
creator	int	NO	题库创建者id
description	varchar	NO	题库描述
question_count	int	NO	题库题目数量

题目讨论区表

字段名称	数据类型	可否为空	说明
id	int	NO	题目讨论区id, 主键
question	int	NO	题目id, 外键
created_at	date	NO	题目创建时间
creator	int	NO	题目创建者id
content	varchar	NO	题目讨论区内容

题目讨论区讨论表

字段名称	数据类型	可否为空	说明
id	int	NO	题目讨论区id, 主键
question	int	NO	题目id, 外键
created_at	date	NO	题目创建时间
creator	int	NO	题目创建者id
content	varchar	NO	题目讨论区内容

做题记录表

字段名称	数据类型	可否为空	说明
id	int	NO	做题记录id，主键
user	int	NO	做题者用户id，外键
question	int	NO	题目id，外键
attempted_at	date	NO	做题时间
is_correct	boolean	NO	是否做对本题

1.3.5 模拟考试部分

考试表

字段名称	数据类型	可否为空	说明
id	int	NO	模拟考试id，主键
title	varchar	NO	模拟考试标题
subject	varchar	NO	题目所属科目
description	varchar	NO	模拟考试描述
created_at	date	NO	考试创建时间
start_time	date	NO	考试开始时间
duration	int	NO	考试持续时间
end_time	date	NO	考试结束时间
created_by	int	NO	考试创建者id，外键
students	json	YES	外键，模拟考试所有报名学生
questions	json	NO	外键，模拟考试所有题目

模拟考试作答表

字段名称	数据类型	可否为空	说明
id	int	NO	做题记录id，主键
exam	int	NO	作答所属考试id，外键
user	int	NO	作答用户id，外键
question	int	NO	作答题目id，外键
submitted_answer	varchar	NO	最后作答答案
submitted_at	date	NO	最后作答时间
is_correct	boolean	NO	是否做对本题

1.3.6 讨论区部分

讨论帖表

字段名称	数据类型	可否为空	说明
id	int	NO	题目讨论区id，主键
title	varchar	NO	讨论帖标题
content	varchar	NO	讨论帖内容
publisher	int	NO	发帖者id，外键
avatar	varchar	NO	发帖者头像
publish_time	date	NO	发帖时间
last_updated	date	NO	讨论帖最后更新时间
tag	varchar	NO	讨论帖标签
subscribers	json	YES	订阅者id集合
likes	json	YES	喜爱者id集合

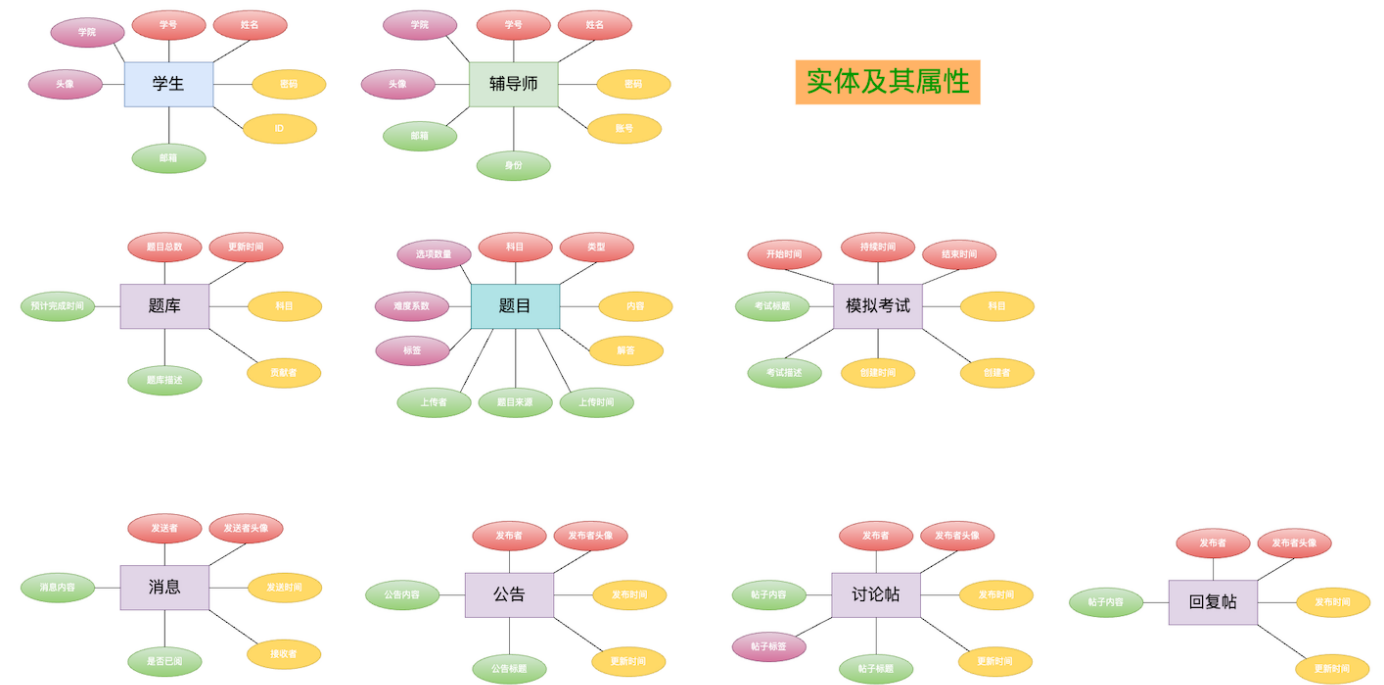
帖子回复表

字段名称	数据类型	可否为空	说明
id	int	NO	题目讨论区id，主键
discussion	int	NO	回帖所属讨论帖id，外键
content	varchar	NO	回复内容
publisher	int	NO	回帖者id，外键
avatar	varchar	NO	回帖者头像
publish_time	date	NO	回帖时间
last_updated	date	NO	回帖最后更新时间
likes	json	YES	回帖喜爱者id集合

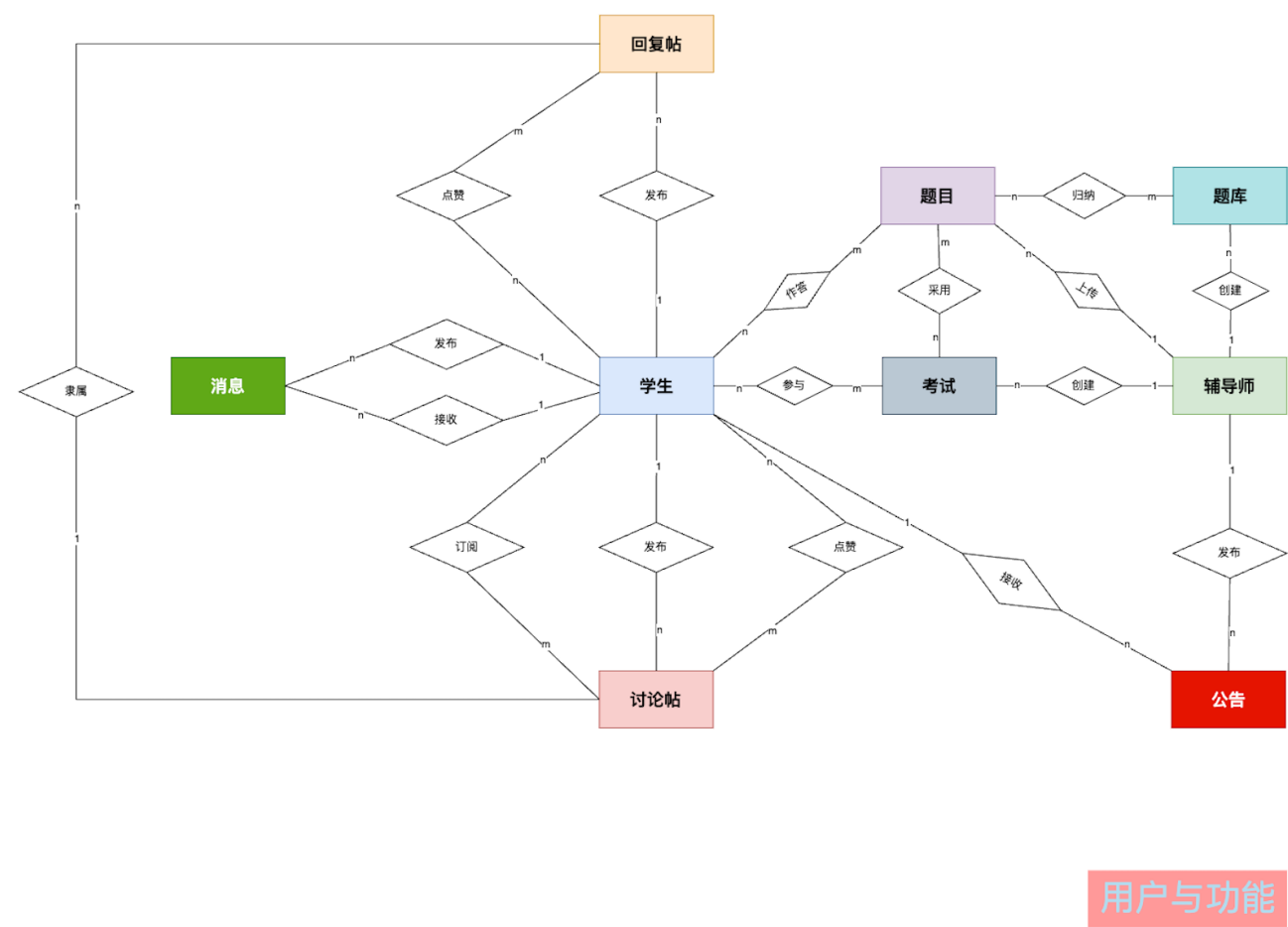
二、数据库概念模式设计

2.1 系统初步 ER 图

2.1.1 实体 ER 图



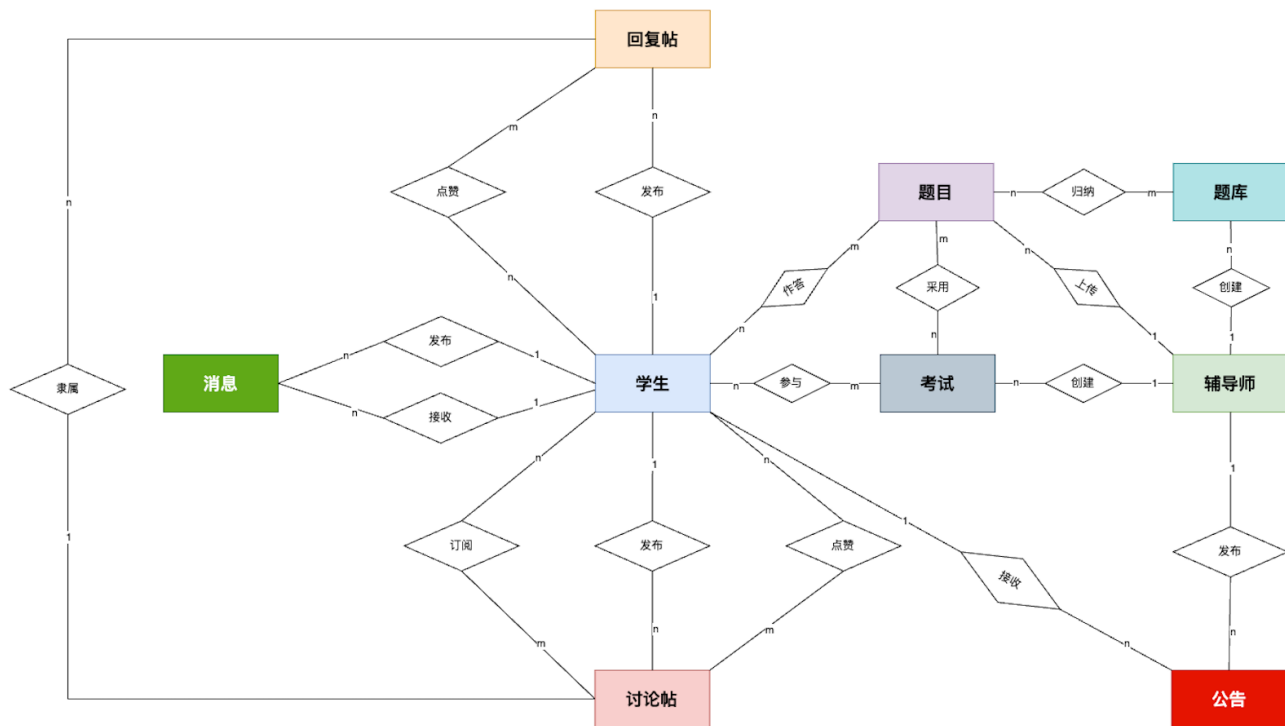
2.1.2 关系 ER 图



2.2 系统基本 ER 图

Important

由于实体属性过于复杂，将属性画入 ER图 内将大大影响图的美观性，故下面 ER图 内将不包含属性，只指示关系



用户与功能

三、数据库逻辑模式设计与优化

3.1 数据库关系模式

Important

关系模式可以形式化表示为 $R(U, D, DOM, F)$ 。U 为组成该关系的属性名，D 为 U 中属性所来自的域，DOM 指的是属性与域的映射，F 指的是属性间的依赖关系集合。以下约定 N 表示正整数，FLOAT 表示浮点数，S 为任意字符组成的字符，T 表示时间，B 表示布尔值，J 表示 json 类型。码以下划线标识。

以下是由 E-R 图得到的关系模式。

3.1.1 实体

1. 学生

students({id, 头像, 邮箱, 密码, 姓名, 学号, 年级, 学院, 身份}, D, DOM, F)

D = {STR, N}

DOM = {DOM(id) = DOM(学号) = DOM(年级) = DOM(身份) = N, DOM(头像) = DOM(邮箱) = DOM(密码) = DOM(姓名) = DOM(学号) = STR}

F = {id → 其他属性}

2. 辅导师

teachers({id, 头像, 邮箱, 密码, 姓名, 学号, 年级, 学院, 身份}, D, DOM, F)

$D = \{\text{STR}, N\}$

$DOM = \{DOM(\text{年级}) = DOM(\text{身份}) = N, DOM(\text{头像}) = DOM(\text{邮箱}) = DOM(\text{密码}) = DOM(\text{姓名}) = DOM(\text{学号}) = \text{STR}\}$

$F = \{id \rightarrow \text{其他属性}\}$

3. 消息

messages({id, 发送者, 发送者头像, 发送时间, 接收者, 消息内容, 是否已阅}, D, DOM, F)

$D = \{\text{STR}, N, B, T\}$

$DOM = \{DOM(id) = DOM(\text{发送者}) = DOM(\text{接收者}) = N, DOM(\text{头像}) = DOM(\text{内容}) = \text{STR}, DOM(\text{发送时间}) = T, DOM(\text{是否已阅}) = B\}$

$F = \{id \rightarrow \text{其他属性}\}$

4. 公告

broadcasts({id, 发布者, 发布者头像, 发布时间, 更新时间, 公告内容, 公告标题}, D, DOM, F)

$D = \{\text{STR}, N, T\}$

$DOM = \{DOM(id) = DOM(\text{发布者}) = N, DOM(\text{发布者头像}) = DOM(\text{内容}) = DOM(\text{标题}) = \text{STR}, DOM(\text{发布时间}) = DOM(\text{更新时间}) = T\}$

$F = \{id \rightarrow \text{其他属性}\}$

5. 题库

questionbanks({id, 科目, 预计完成时间, 创建时间, 创建者, 题库描述, 题目数量}, D, DOM, F)

$D = \{\text{STR}, N, T\}$

$DOM = \{DOM(id) = DOM(\text{创建者}) = DOM(\text{题目数量}) = N, DOM(\text{科目}) = DOM(\text{题库描述}) = \text{STR}, DOM(\text{创建时间}) = DOM(\text{预计完成时间}) = T\}$

$F = \{id \rightarrow \text{其他属性}\}$

6. 题目

questions({id, 类型, 内容, 科目, 上传时间, 来源, 标签, 难度, 上传者, 选项数量, 所属题库}, D, DOM, F)

$D = \{\text{STR}, N, T, J\}$

$DOM = \{DOM(id) = DOM(\text{上传者}) = DOM(\text{题目难度}) = DOM(\text{选项数量}) = DOM(\text{所属题库}) = N, DOM(\text{科目}) = DOM(\text{类型}) = DOM(\text{来源}) = DOM(\text{内容}) = \text{STR}, DOM(\text{标签}) = DOM(\text{所属题库}) = J, DOM(\text{上传时间}) = T\}$

$F = \{id \rightarrow \text{其他属性}\}$

7. 讨论帖

discussions({id, 发布者, 发布者头像, 发布时间, 更新时间, 帖子内容, 帖子标题, 帖子标签, 订阅者, 喜爱者}, D, DOM, F)

$D = \{\text{STR}, N, T, J\}$

$DOM = \{DOM(id) = DOM(\text{发布者}) = N, DOM(\text{发布者头像}) = DOM(\text{内容}) = DOM(\text{标题}) = DOM(\text{标签}) = \text{STR}, DOM(\text{订阅者}) = DOM(\text{喜爱者}) = J, DOM(\text{发布时间}) = DOM(\text{更新时间}) = T\}$

$F = \{id \rightarrow \text{其他属性}\}$

8. 回复帖

replies({id, 所属讨论帖, 发布者, 发布者头像, 发布时间, 更新时间, 帖子内容, 喜爱者}, D, DOM, F)

$D = \{STR, N, T, J\}$

$DOM = \{DOM(id) = DOM(所属讨论帖) = DOM(发布者) = N, DOM(发布者头像) = DOM(内容) = STR, DOM(喜爱者) = J, DOM(发布时间) = DOM(更新时间) = T\}$

$F = \{id \rightarrow \text{其他属性}\}$

9. 模拟考试

exams({id, 标题, 科目, 开始时间, 持续时间, 结束时间, 创建时间, 创建者, 考试描述, 考试题目, 参考学生}, D, DOM, F)

$D = \{STR, N, T, J\}$

$DOM = \{DOM(id) = DOM(创建者) = N, DOM(科目) = DOM(标题) = DOM(考试描述) = STR, DOM(考试题目) = DOM(参考学生) = J, DOM(创建时间) = DOM(持续时间) = DOM(创开始时间) = DOM(结束时间) = T\}$

$F = \{id \rightarrow \text{其他属性}\}$

3.1.2 联系

Important

约定以下函数依赖集合省略非主属性对码的完全函数依赖。

3.1.2.1 多对多联系

1. 学生——考试

studentExam({student, exam}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(student) = DOM(exam) = N\}$

外码：student 对应学生表的主码 student.id, course 对应考试表的主码 course.id

2. 考试——题目

questionExam({question, exam}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(question) = DOM(exam) = N\}$

外码：question 对应题目表的主码 question.id, course 对应考试表的主码 course.id

3. 题库——题目

questionQuestionbank({question, questionbank}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(question) = DOM(questionbank) = N\}$

外码：question 对应题目表的主码 question.id, questionbank 对应题库表的主码 questionbank.id

4. 帖子——订阅

subscribeDiscussion({student, discussion}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(student) = DOM(discussion) = N\}$

外码： student 对应学生表的主码 student.id, discussion 对应讨论帖表的主码 discussion.id

5. 帖子——喜爱

likeDiscussion({student, discussion}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(student) = DOM(discussion) = N\}$

外码： student 对应学生表的主码 student.id, discussion 对应讨论帖表的主码 discussion.id

6. 回复——喜爱

likeReply({student, reply}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(student) = DOM(reply) = N\}$

外码： student 对应学生表的主码 student.id, reply 对应回复帖表的主码 reply.id

7. 学生作答题目

studentSubmission({student, question, time, submission, is_correct}, D, DOM, F)

$D = \{STR, N, T, B\}$

$DOM = \{DOM(submission) = STR, DOM(student) = DOM(question) = N, DOM(time) = T, DOM(is_correct) = B\}$

外码： student 对应学生表的主码 student.id, question 对应题目表的主码 question.id

3.1.2.2 一对多联系

1. 学生——发送消息

sendMessage({message, student}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(student) = DOM(message) = N\}$

外码： student 对应学生表的主码 student.id, message 对应消息表的主码 message.id

2. 学生——接收消息

receiveMessage({message, student}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(student) = DOM(message) = N\}$

外码： student 对应学生表的主码 student.id, message 对应消息表的主码 message.id

3. 学生——接收公告

receiveBroadcast({message, student}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(student) = DOM(broadcast) = N\}$

外码： student 对应学生表的主码 student.id, broadcast 对应公告表的主码 broadcast.id

4. 教师——发布公告

publishBroadcast({message, teacher}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{teacher}) = DOM(\text{broadcast}) = N\}$

外码：teacher 对应教师表的主码 teacher.id, broadcast 对应公告表的主码 broadcast.id

5. 教师——题库

teacherQuestionbank({questionbank, teacher}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{teacher}) = DOM(\text{questionbank}) = N\}$

外码：teacher 对应教师表的主码 teacher.id, questionbank 对应题库表的主码 questionbank.id

6. 教师——题目

teacherQuestion({question, teacher}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{teacher}) = DOM(\text{question}) = N\}$

外码：teacher 对应教师表的主码 teacher.id, question 对应题库表的主码 question.id

7. 教师——考试

teacherExam({exam, teacher}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{teacher}) = DOM(\text{exam}) = N\}$

外码：teacher 对应教师表的主码 teacher.id, exam 对应考试表的主码 exam.id

8. 学生——讨论

studentDiscussion({discussion, student}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{student}) = DOM(\text{discussion}) = N\}$

外码：student 对应学生表的主码 student.id, discussion 对应讨论表的主码 discussion.id

9. 学生——回复

studentReply({reply, student}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{student}) = DOM(\text{reply}) = N\}$

外码：student 对应学生表的主码 student.id, reply 对应回复表的主码 reply.id

10. 讨论帖——回复

discussionReply({reply, discussion}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{reply}) = DOM(\text{discussion}) = N\}$

外码：discussion 对应讨论表的主码 discussion.id, reply 对应回复表的主码 reply.id

3.2 关系模式范式等级的判定与规范

Note

分别对上述表进行范式等级判断，然后将未达到 3NF 的表规范到 3NF。

3.2.1 实体

1. 学生

students({id, 头像, 邮箱, 密码, 姓名, 学号, 年级, 学院, 身份}, D, DOM, F)

$D = \{\text{STR}, N\}$

$DOM = \{DOM(id) = DOM(年级) = DOM(身份) = N, DOM(头像) = DOM(邮箱) = DOM(密码) = DOM(姓名) = DOM(学号) = \text{STR}\}$

$F = \{id \rightarrow \text{其他属性}\}$

对于学生表，id 是主码，非主属性仅仅依赖于唯一的主码，所以学生表 $\in BCNF$ ，学生表 $\in 3NF$

2. 辅导师

teachers({id, 头像, 邮箱, 密码, 姓名, 学号, 年级, 学院, 身份}, D, DOM, F)

$D = \{\text{STR}, N\}$

$DOM = \{DOM(年级) = DOM(身份) = N, DOM(头像) = DOM(邮箱) = DOM(密码) = DOM(姓名) = DOM(学号) = \text{STR}\}$

$F = \{id \rightarrow \text{其他属性}\}$

对于辅导师表，id 是主码，非主属性仅仅依赖于唯一的主码，所以辅导师表 $\in BCNF$ ，辅导师表 $\in 3NF$

3. 消息

messages({id, 发送者, 发送者头像, 发送时间, 接收者, 消息内容, 是否已阅}, D, DOM, F)

$D = \{\text{STR}, N, B, T\}$

$DOM = \{DOM(id) = DOM(发送者) = DOM(接收者) = N, DOM(头像) = DOM(内容) = \text{STR}, DOM(发送时间) = T, DOM(是否已阅) = B\}$

$F = \{id \rightarrow \text{其他属性}\}$

对于消息表，id 是主码，非主属性仅仅依赖于唯一的主码，所以消息表 $\in BCNF$ ，消息表 $\in 3NF$

4. 公告

broadcasts({id, 发布者, 发布者头像, 发布时间, 更新时间, 公告内容, 公告标题}, D, DOM, F)

$D = \{\text{STR}, N, T\}$

$DOM = \{DOM(id) = DOM(发布者) = N, DOM(发布者头像) = DOM(内容) = DOM(标题) = \text{STR}, DOM(发布时间) = DOM(更新时间) = T\}$

$F = \{id \rightarrow \text{其他属性}\}$

对于公告表，id 是主码，非主属性仅仅依赖于唯一的主码，所以公告表 $\in BCNF$ ，公告表 $\in 3NF$

5. 题库

questionbanks({id, 科目, 预计完成时间, 创建时间, 创建者, 题库描述, 题目数量}, D, DOM, F)

$D = \{\text{STR}, N, T\}$

$DOM = \{DOM(id) = DOM(创建者) = DOM(题目数量) = N, \quad DOM(科目) = DOM(题库描述) = STR, \quad DOM(创建时间) = DOM(预计完成时间) = T\}$

$F = \{id \rightarrow \text{其他属性}\}$

对于题库表, id 是主码, 非主属性仅仅依赖于唯一的主码, 所以题库表 $\in BCNF$, 题库表 $\in 3NF$

6. 题目

questions(id , 类型, 内容, 科目, 上传时间, 来源, 标签, 难度, 上传者, 选项数量, 所属题库, D , DOM , F)

$D = \{STR, N, T, J\}$

$DOM = \{DOM(id) = DOM(上传者) = DOM(题目难度) = DOM(选项数量) = DOM(所属题库) = N, \quad DOM(科目) = DOM(类型) = DOM(来源) = DOM(内容) = STR, \quad DOM(标签) = DOM(所属题库) = J, \quad DOM(上传时间) = T\}$

$F = \{id \rightarrow \text{其他属性}\}$

对于题目表, id 是主码, 非主属性仅仅依赖于唯一的主码, 所以题目表 $\in BCNF$, 题目表 $\in 3NF$

7. 讨论帖

discussions(id , 发布者, 发布者头像, 发布时间, 更新时间, 帖子内容, 帖子标题, 帖子标签, 订阅者, 喜爱者}, D , DOM , F)

$D = \{STR, N, T, J\}$

$DOM = \{DOM(id) = DOM(发布者) = N, \quad DOM(发布者头像) = DOM(内容) = DOM(标题) = DOM(标签) = STR, \quad DOM(订阅者) = DOM(喜爱者) = J, \quad DOM(发布时间) = DOM(更新时间) = T\}$

$F = \{id \rightarrow \text{其他属性}\}$

对于讨论帖表, id 是主码, 非主属性仅仅依赖于唯一的主码, 所以讨论帖表 $\in BCNF$, 讨论帖表 $\in 3NF$

8. 回复帖

replies(id , 所属讨论帖, 发布者, 发布者头像, 发布时间, 更新时间, 帖子内容, 喜爱者}, D , DOM , F)

$D = \{STR, N, T, J\}$

$DOM = \{DOM(id) = DOM(所属讨论帖) = DOM(发布者) = N, \quad DOM(发布者头像) = DOM(内容) = STR, \quad DOM(喜爱者) = J, \quad DOM(发布时间) = DOM(更新时间) = T\}$

$F = \{id \rightarrow \text{其他属性}\}$

对于回复帖表, id 是主码, 非主属性仅仅依赖于唯一的主码, 所以回复帖表 $\in BCNF$, 回复帖表 $\in 3NF$

9. 模拟考试

exams(id , 标题, 科目, 开始时间, 持续时间, 结束时间, 创建时间, 创建者, 考试描述, 考试题目, 参考学生}, D , DOM , F)

$D = \{STR, N, T, J\}$

$DOM = \{DOM(id) = DOM(创建者) = N, \quad DOM(科目) = DOM(标题) = DOM(考试描述) = STR, \quad DOM(考试题目) = DOM(参考学生) = J, \quad DOM(创建时间) = DOM(持续时间) = DOM(创开始时间) = DOM(结束时间) = T\}$

$F = \{id \rightarrow \text{其他属性}\}$

对于模拟考试表, id 是主码, 非主属性仅仅依赖于唯一的主码, 所以模拟考试表 $\in BCNF$, 模拟考试表 $\in 3NF$

3.2.2 联系

Important

约定以下函数依赖集合省略非主属性对码的完全函数依赖。

3.2.2.1 多对多联系

1. 学生——考试

$\text{studentExam}(\{\text{student}, \text{exam}\}, D, \text{DOM}, F)$

$D = \{N\}$

$\text{DOM} = \{\text{DOM}(\text{student}) = \text{DOM}(\text{exam}) = N\}$

外码：student 对应学生表的主码 student.id, exam 对应考试表的主码 exam.id

对于该表，所有函数依赖左端都是主码，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

2. 考试——题目

$\text{questionExam}(\{\text{question}, \text{exam}\}, D, \text{DOM}, F)$

$D = \{N\}$

$\text{DOM} = \{\text{DOM}(\text{question}) = \text{DOM}(\text{exam}) = N\}$

外码：question 对应题目表的主码 question.id, exam 对应考试表的主码 exam.id

对于该表，所有函数依赖左端都是主码，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

3. 题库——题目

$\text{questionQuestionbank}(\{\text{question}, \text{questionbank}\}, D, \text{DOM}, F)$

$D = \{N\}$

$\text{DOM} = \{\text{DOM}(\text{question}) = \text{DOM}(\text{questionbank}) = N\}$

外码：question 对应题目表的主码 question.id, questionbank 对应题库表的主码 questionbank.id

对于该表，所有函数依赖左端都是主码，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

4. 帖子——订阅

$\text{subscribeDiscussion}(\{\text{student}, \text{discussion}\}, D, \text{DOM}, F)$

$D = \{N\}$

$\text{DOM} = \{\text{DOM}(\text{student}) = \text{DOM}(\text{discussion}) = N\}$

外码：student 对应学生表的主码 student.id, discussion 对应讨论帖表的主码 discussion.id

$F = \{(\text{student}, \text{discussion}) \rightarrow \text{subscribed}\}$

对于该表，所有函数依赖左端都是主码，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

5. 帖子——喜爱

$\text{likeDiscussion}(\{\text{student}, \text{discussion}\}, D, \text{DOM}, F)$

$D = \{N\}$

$\text{DOM} = \{\text{DOM}(\text{student}) = \text{DOM}(\text{discussion}) = N\}$

外码：student 对应学生表的主码 student.id, discussion 对应讨论帖表的主码 discussion.id

$F = \{(student, discussion) \rightarrow liked\}$

对于该表，所有函数依赖左端都是主码，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

6. 回复——喜爱

likeReply({student, reply}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(student) = DOM(reply) = N\}$

外码：student 对应学生表的主码 student.id, reply 对应回复帖表的主码 reply.id

$F = \{(student, reply) \rightarrow liked\}$

对于该表，所有函数依赖左端都是主码，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

7. 学生作答题目

studentSubmission({student, question, time, submission, is_correct}, D, DOM, F)

$D = \{STR, N, T, B\}$

$DOM = \{DOM(submission) = STR, DOM(student) = DOM(question) = N, DOM(time) = T, DOM(is_correct) = B\}$

外码：student 对应学生表的主码 student.id, question 对应题目表的主码 question.id

$F = \{(student, question) \rightarrow is_correct, (student, question) \rightarrow time\}$

对于该表，所有函数依赖左端都是主码，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

3.2.2.2 一对多联系

1. 学生——发送消息

sendMessage({message, student}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(student) = DOM(message) = N\}$

外码：student 对应学生表的主码 student.id, message 对应消息表的主码 message.id

$F = \{message \rightarrow student\}$

函数依赖左端仅包含主码，不存在传递函数依赖和部分函数依赖，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

2. 学生——接收消息

receiveMessage({message, student}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(student) = DOM(message) = N\}$

外码：student 对应学生表的主码 student.id, message 对应消息表的主码 message.id

$F = \{message \rightarrow student\}$

函数依赖左端仅包含主码，不存在传递函数依赖和部分函数依赖，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

3. 学生——接收公告

receiveBroadcast({message, student}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(student) = DOM(broadcast) = N\}$

外码：student 对应学生表的主码 student.id, broadcast 对应公告表的主码 broadcast.id

$F = \{\text{broadcast} \rightarrow \text{student}\}$

函数依赖左端仅包含主码，不存在传递函数依赖和部分函数依赖，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

4. 教师——发布公告

publishBroadcast({message, teacher}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{teacher}) = DOM(\text{broadcast}) = N\}$

外码：teacher 对应教师表的主码 teacher.id, broadcast 对应公告表的主码 broadcast.id

$F = \{\text{broadcast} \rightarrow \text{teacher}\}$

函数依赖左端仅包含主码，不存在传递函数依赖和部分函数依赖，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

5. 教师——题库

teacherQuestionbank({questionbank, teacher}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{teacher}) = DOM(\text{questionbank}) = N\}$

外码：teacher 对应教师表的主码 teacher.id, questionbank 对应题库表的主码 questionbank.id

$F = \{\text{questionbank} \rightarrow \text{teacher}\}$

函数依赖左端仅包含主码，不存在传递函数依赖和部分函数依赖，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

6. 教师——题目

teacherQuestion({question, teacher}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{teacher}) = DOM(\text{question}) = N\}$

外码：teacher 对应教师表的主码 teacher.id, question 对应题库表的主码 question.id

$F = \{\text{question} \rightarrow \text{teacher}\}$

函数依赖左端仅包含主码，不存在传递函数依赖和部分函数依赖，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

7. 教师——考试

teacherExam({exam, teacher}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{teacher}) = DOM(\text{exam}) = N\}$

外码：teacher 对应教师表的主码 teacher.id, exam 对应考试表的主码 exam.id

$F = \{\text{exam} \rightarrow \text{teacher}\}$

函数依赖左端仅包含主码，不存在传递函数依赖和部分函数依赖，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

8. 学生——讨论

studentDiscussion({discussion, student}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{student}) = DOM(\text{discussion}) = N\}$

外码：student 对应学生表的主码 student.id, discussion 对应讨论表的主码 discussion.id

$F = \{\text{discussion} \rightarrow \text{student}\}$

函数依赖左端仅包含主码，不存在传递函数依赖和部分函数依赖，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

9. 学生——回复

studentReply({reply, student}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{student}) = DOM(\text{reply}) = N\}$

外码：student 对应学生表的主码 student.id, reply 对应回复表的主码 reply.id

$F = \{\text{reply} \rightarrow \text{student}\}$

函数依赖左端仅包含主码，不存在传递函数依赖和部分函数依赖，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

10. 讨论帖——回复

discussionReply({reply, discussion}, D, DOM, F)

$D = \{N\}$

$DOM = \{DOM(\text{reply}) = DOM(\text{discussion}) = N\}$

外码：discussion 对应讨论表的主码 discussion.id, reply 对应回复表的主码 reply.id

$F = \{\text{reply} \rightarrow \text{discussion}\}$

函数依赖左端仅包含主码，不存在传递函数依赖和部分函数依赖，所以 $\in BCNF$ ，因此也 $\in 3NF$ 。

3.3 数据库关系模式优化

3.3.1 使用外键

使用外键来保证数据的关联性并提高访问速度，同时在使用过程中保证数据的参照完整性。在本项目的后端数据库中，各表具有完备的外键联系，通过遵循这种外键关系进行实际数据库的建立，将从逻辑的角度保证数据库的设计功能能够完全正确执行到位。

3.3.2 优化查询语句

可以优化的地方有如下几点：

- 能使用联表查询的地方尽可能使用联表查询比较高效
- 尽可能不将查询语句放在循环内
- 使用高效的 ORM 框架

3.3.3 合并相同主码的表

对于主码相同的表，可以在设计的时候将其合并，例如将题目创建表和题目表合并，直接将题目的上传者和上传时间存入题目表，这样做的好处是在同时需要查询两个表中的数据时，减少了一次连接操作，提高效率。

3.3.4 严格约束加入数据库的数据

通过建表时添加约束、用触发器实现约束等方法，严格约束对数据库的增删操作，避免将异常数据插入数据表。

建表时可以添加的约束有主键约束、UNIQUE 约束、外码约束等。

四、数据库物理设计

4.1 存取方法

存储引擎：

- 选择支持事务和索引优化的存储引擎——支持MySQL 的 GAUSSDB。
- 理由：GAUSSDB 支持事务和外键约束，可以确保数据一致性，同时 MySQL 针对复杂查询性能较优。

分区策略：

- 垂直分区
 - 将核心表（如 User、Question 和 QuestionBank）与非核心表（如 QuestionComment 和 QuestionDiscussion）分开存储。
 - 理由：核心表频繁被查询，可以优化存储和查询效率；非核心表如评论和讨论，属于高并发写入的表，可独立优化。
- 水平分区
 - 针对大数据量表（如 Question 和 UserQuestionRecord），根据字段（如 added_by 或 user_id）进行分片。
 - 理由：便于高并发环境下的数据分布和扩展性。

数据类型选择：

- 使用合适的数据类型，如 INTEGER、VARCHAR、DATE 等。
- 对于 JSON 格式数据（如 tags），选择利用字符串存储。

4.2 索引定义

为提高查询效率，我们设计索引如下：

1. User 表

- 主键索引：id（自动创建）
- 唯一索引：student_id
 - 优点：学生学号唯一，且常用于查询，加快查询速度
- 索引：user_role
 - 优点：辅导师角色（user_role = 1）在权限检查中频繁使用，加快查询速度

2. Question 表

- 主键索引: `id` (自动创建)
- 索引: `added_by`
 - 优点: 便于根据创建者快速查询
- 索引: `type`和 `subject`
 - 优点: 按题目类型和科目分类查询时提升效率
- 索引: `added_at`
 - 优点: 时间排序查询优化
- 索引: `tags`
 - 优点: 支持基于关键词的搜索 (如 `工科数学分析`)

3. QuestionBank 表

- 主键索引: `id` (自动创建)
- 索引: `subject`
 - 优点: 按学科分类查询优化
- 索引: `creator`
 - 优点: 便于按创建者筛选

4. QuestionDiscussion 表

- 主键索引: `id` (自动创建)
- 索引: `question_id`
 - 优点: 按题目快速查询讨论区。
- 索引: `created_by`
 - 优点: 便于按创建者筛选讨论。

5. QuestionComment 表

- 主键索引: `id` (自动创建)
- 索引: `discussion_id`
 - 优点: 便于按讨论区快速查询评论
- 索引: `created_by`
 - 优点: 便于按创建者筛选评论

6. UserQuestionRecord 表

- 主键索引: `id` (自动创建)
- 唯一复合索引: `(user_id, question_id)`
 - 优点: 确保每个用户对每道题目只有一个做题记录

- 索引: `user_id`
 - 优点: 按用户快速筛选做题记录
- 索引: `question_id`
 - 优点: 按题目快速筛选做题记录

7. ExamRecordRecord 表

- 主键索引: `id` (自动创建)
- 唯一复合索引: `(student_id, question_id)`
 - 优点: 确保每个用户对每道题目只有一个做题记录
- 索引: `student_id`
 - 优点: 按用户快速筛选做题记录
- 索引: `question_id`
 - 优点: 按题目快速筛选做题记录

8. Discussion 表

- 主键索引: `id` (自动创建)
- 索引: `publisher`
 - 优点: 便于按创建者筛选讨论

9. Reply 表

- 主键索引: `id` (自动创建)
- 索引: `publisher`
 - 优点: 便于按创建者筛选讨论
- 索引: `discussion`
 - 优点: 便于按讨论帖筛选回复

10. Message 表

- 主键索引: `id` (自动创建)
- 索引: `sender`
 - 优点: 便于按发送者筛选消息
- 索引: `receiver`
 - 优点: 便于按接收者筛选消息

11. Broadcast 表

- 主键索引: `id` (自动创建)
- 索引: `sender`
 - 优点: 便于按发布者筛选公告

12. 外键约束

- 为外键字段自动创建索引：
 - `QuestionBank.questions` → `Question.id`
 - `QuestionDiscussion.question` → `Question.id`
 - `QuestionComment.discussion` → `QuestionDiscussion.id`
 - `UserQuestionRecord.question` → `Question.id`
 - `UserQuestionRecord.user` → `User.id`