

IKNOW

数据库设计说明书

团队：刚下飞机

1. 引言	1
1.1 编写目的	1
1.2 项目背景和内容概要.....	1
1.3 相关资料、缩略语、定义.....	1
2. 约定	2
3. 数据库概念模型设计	2
3.1 数据实体图.....	2
3.2 数据实体描述.....	3
3.3 实体关系描述.....	6
4. 数据库逻辑模型设计	10
4.1 实体-关系图.....	10
4.2 关系模型概述.....	12
4.3 数据库一致性设计.....	40
5. 物理实现	63
5.1 数据库的安排.....	64
5.2 安全保密设计.....	64

引言

1.1 编写目的

- 1、本数据库设计说明书是关于“KKnow”管理系统数据库设计，主要包括数据库概念结构设计、逻辑结构设计、物理实现。
- 2、本数据库设计说明书根据“KKnow”项目的需求分析、构建原型，以及从原型中设计类图，再从类图中提取对应的数据库表来编写的。
- 3、本数据库说明书为现阶段开发人员的开发设计提供参考。
- 4、本数据库说明书也为使用者和需要二次开发的技术人员提供明确的使用、功能说明和数据库设计信息，以及供管理人员进行商讨和使用参考。

1.2 项目背景和内容概要

项目名： KKnow

开发团队： 刚下飞机

成员： 卓晓鑫 张庭博 朱凯文 张春翔 池政涛 郭秋中 郭福强 胡海江

内容概要： 为了解决大学生提出学习相关问题不方便的问题，让大学生的提出的问题能够被更多人看见、回答。同时也可以帮助那些性格内向的同学有机会解决自己所面临的问题。并且，教师的解答或高评分的优质回答可以帮助有类似疑问的同学。该应用提供了一个可供同学与老师、同学与同学之间交流的平台。

1.3 相关资料、缩略语、定义

缩略词和术语	解释
--------	----

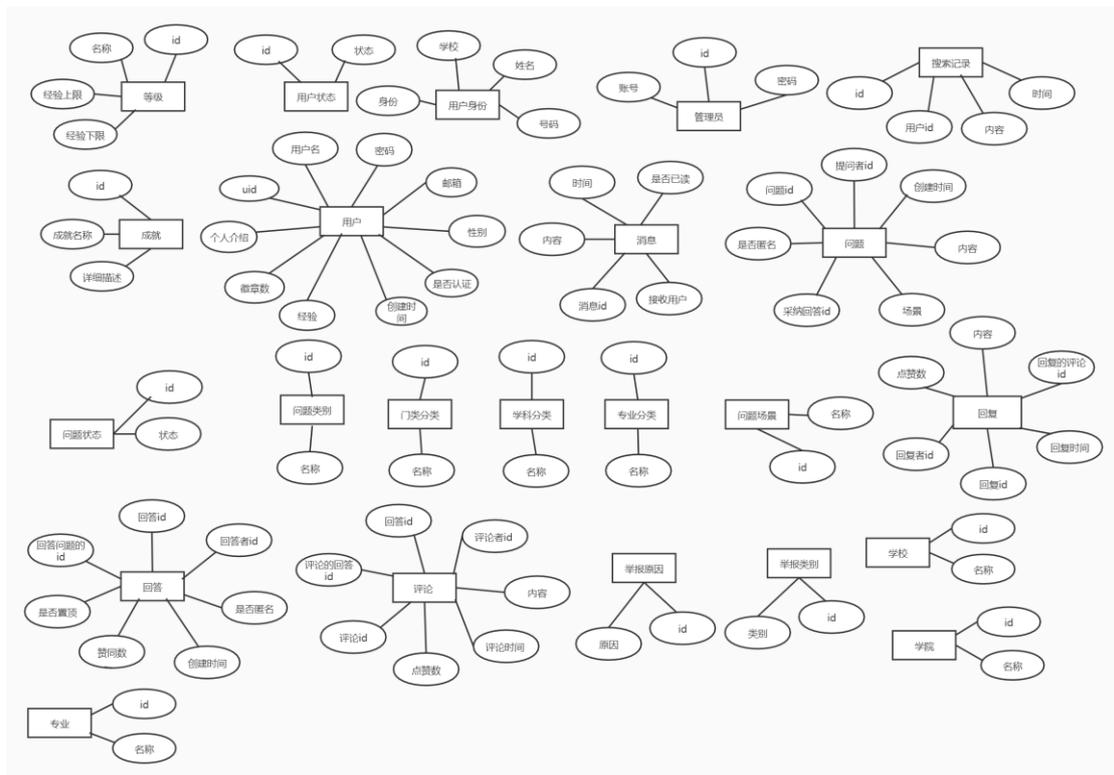
RDS	RDS 是关系型数据库服务 (Relational Database Service) 的简称, 是一种即开即用、稳定可靠、可弹性伸缩的在线数据库服务。具有多重安全防护措施和完善的性能监控体系, 并提供专业的数据库备份、恢复及优化方案, 使用户能专注于应用开发和业务发展。
-----	--

约定

- a. 表名小写字母开头遵循驼峰命名法则, 表名要能体现表内容。
- b. 字段名小写字母开头遵循驼峰命名法则, 字段名要能体现表内容。
- c. 各表必须要有唯一一个主键字段以 id 命名 (多对多关系连接表除外)。
- d. 外键删除约束统一使用 RESTRICT。

数据库概念模型设计

3.1 数据实体图



3.2 数据实体描述

管理员：admin

描述：管理员用户的信息。

用户：user

描述：普通用户的信息。

问题：question

描述：用户提出的问题。

回答：answer

描述：用户在问题下做出的回答。

评论: comment

描述: 用户在回答下的评论。

回复: reply

描述: 用户在评论下的回复。

用户状态: userState

描述: 用户当前的状态。

问题状态: questionState

描述: 问题当前的状态。

等级: level

描述: 问题当前的等级。

成就: achievement

描述: 用户当前的成就。

搜索记录: searchHistory

描述: 用户的搜索记录。

消息: message

描述: 用户收到的消息。

问题场景: problemScenario

描述: 问题所属的问题场景。

学校: school

描述: 学校信息。

学院: college

描述: 学校的学院信息。

专业: major

描述: 学院的专业信息。

问题分类: questionCategory

描述: 问题的所有分类。

门类分类: categoriesType

描述: 问题的门类分类。

学科分类: subjectType

描述: 问题的学科分类。

专业分类: majorType

描述: 问题的专业分类。

举报原因: reportReason

描述: 评论、回答等的举报原因。

举报类别: reportCategory

描述: 举报所属的举报类别。

3.3 实体关系描述

用户提问关系

用户 1 : 问题 n

关系描述: 一个用户可以发表多个问题, 一个回问题只属于一个用户。

用户回答关系

用户 1 : 回答 n

关系描述: 一个用户可以发表多个回答, 一个回回答只属于一个用户。

用户评论关系

用户 1 : 评论 n

关系描述: 一个用户可以发表多个回评论, 一个评论只属于一个用户。

用户回复关系

用户 1 : 回复 n

关系描述: 一个用户可以发表多个回复, 一个回复只属于一个用户。

用户状态关系

用户 n : 用户状态 1

关系描述: 一个用户只能有一种状态, 一个状态可以多个用户拥有。

用户身份关系

用户 1 : 身份 1

关系描述: 一个用户只能有一个身份, 一个身份只属于一个用户。

用户消息关系

用户 1 : 消息 n

关系描述: 一个用户能收到多个消息, 一个消息只发给一个用户。

用户等级关系

用户 n : 等级 1

关系描述: 一个用户只能有一个等级, 一个等级下有多个用户。

用户成就关系

用户 n : 成就 m

关系描述: 一个用户能有多个成就, 一个成就被多个用户持有。

用户收藏问题关系

用户 n : 问题 m

关系描述: 一个用户能收藏多个问题, 一个问题能被多个用户收藏。

用户浏览问题关系

用户 n : 问题 m

关系描述: 一个用户能浏览多个问题, 一个问题能被多个用户浏览。

用户搜索记录关系

用户 1 : 搜索记录 n

关系描述: 一个用户能有多条搜索记录, 一条搜索记录属于一位用户

用户举报问题关系

用户 n : 问题 m

关系描述：一个用户能举报多个问题，一个问题能被多个用户举报。

用户举报回答关系

用户 n : 回答 m

关系描述：一个用户能举报多个问题，一个回答能被多个用户举报。

用户举报评论关系

用户 n : 评论 m

关系描述：一个用户能举报多个问题，一个评论能被多个用户举报。

用户举报回复关系

用户 n : 回复 m

关系描述：一个用户能举报多个问题，一个回复能被多个用户举报。

用户赞同回答关系

用户 n : 回答 m

关系描述：一个用户能赞同多个回答，一个回答能被多个用户赞同。

用户反对回答关系

用户 n : 回答 m

关系描述：一个用户能反对多个回答，一个回答能被多个用户反对。

用户点赞评论关系

用户 n : 回答 m

关系描述：一个用户能点赞多个评论，一个评论能被多个用户点赞。

用户点赞回复关系

用户 n : 回复 m

关系描述: 一个用户能点赞多个回复, 一个回复能被多个用户点赞。

问题回答关系

问题 1 : 回答 n

关系描述: 一个问题能有多个回答, 一个回答只属于一个问题。

回答评论关系

回答 1 : 评论 n

关系描述: 一个回答能有多条评论, 一条评论只属于一个回答。

评论回复关系

评论 1 : 回复 n

关系描述: 一条评论能有多个回复, 一条回复只属于一条评论。

问题状态关系

问题 n : 问题状态 1

关系描述: 一个问题只能有一种状态, 一种状态下能有多个问题。

问题场景关系

问题 n : 问题场景 1

关系描述: 一个问题只能有一种问题场景, 一种问题场景中能有多个问题。

问题类别关系

问题 n : 问题类别 1

关系描述: 一个问题只属于一种类别, 一种类别下能有多个问题。

问题类别门类分类关系

问题类别 n : 门类分类 1

关系描述: 一种问题类别只有一种门类分类, 一种门类分类能属于多个问题类别。

问题类别学科分类关系

问题类别 n : 学科分类 1

关系描述: 一种问题类别只有一种学科分类, 一种学科分类能属于多个问题类别。

问题类别专业分类关系

问题类别 n : 专业分类 1

关系描述: 一种问题类别只有一种专业分类, 一种专业分类能属于多个问题类别。

问题门类分类问题学科分类关系

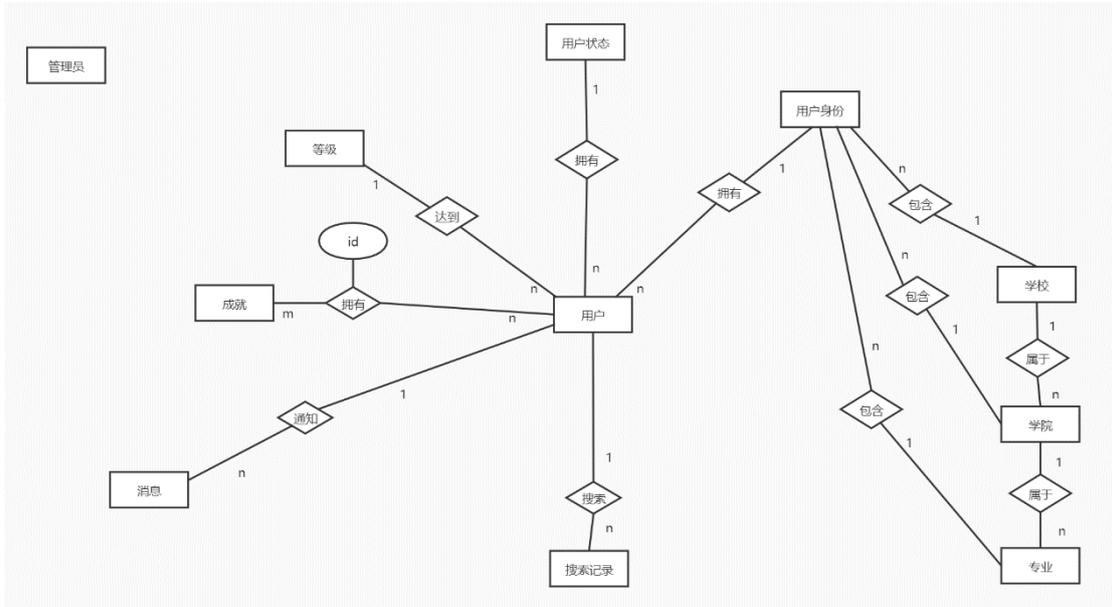
门类分类 1 : 学科分类 n

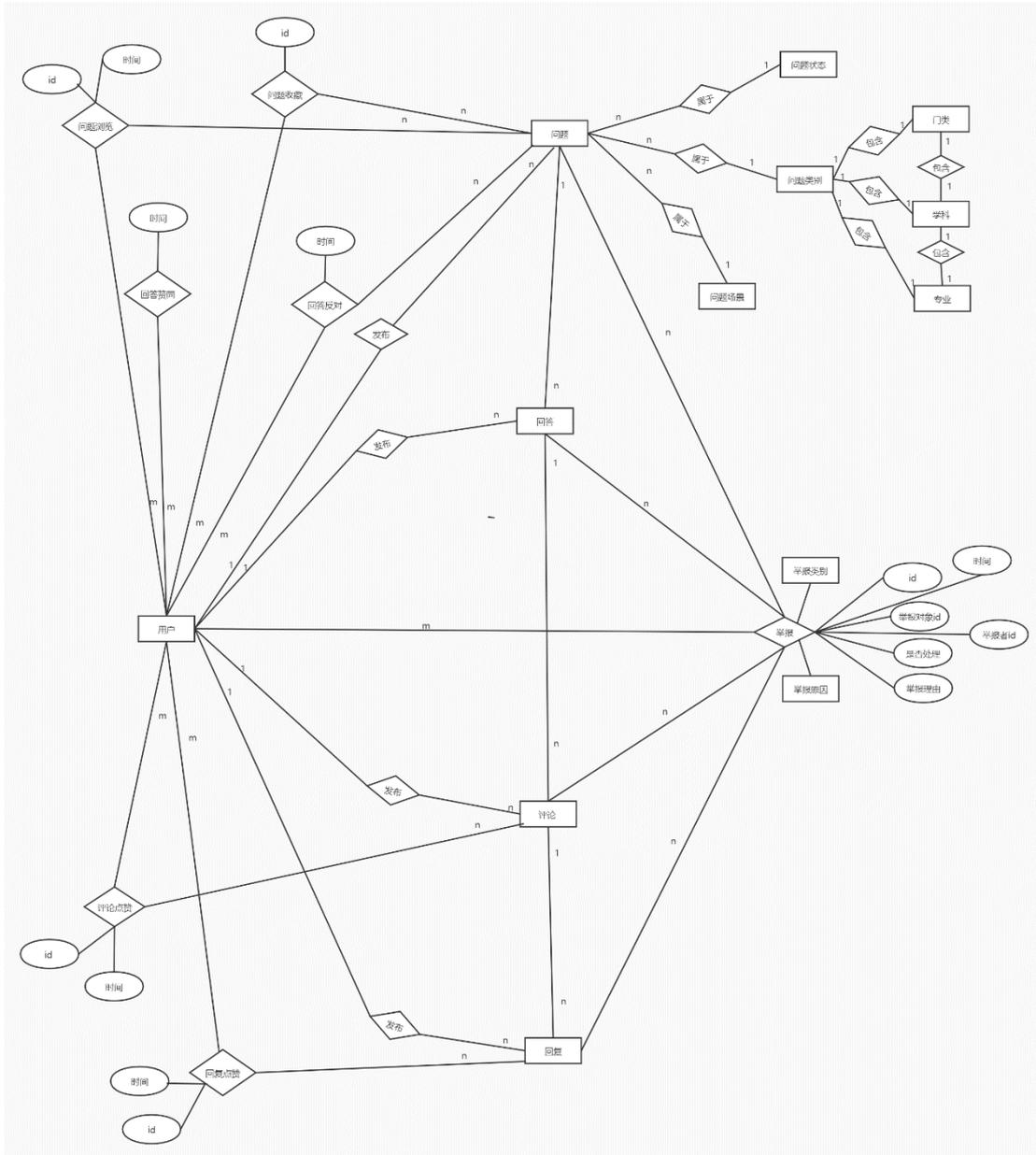
关系描述 “一种门类分类中能包含多种学科分类, 一种学科分类只属于一个门类。

问题学科分类问题专业分类关系

学科分类 1 : 专业分类 n

关系描述: 一种学科分类中能包含多种专业分类, 一种专业分类只属于一个学科。





4.2 关系模型描述

数据库表名：ADMIN

同义词（别名）：管理员表

主键：管理员 id

外键：无

索引：主键索引

约束: id:primary key,unique,not null;

account:unique,not null;

passwd:not null

中文名称	数据属性名	数据类型	数据长度	约束范围	是否空	注解
管理员id	id	int	11	0001-9999	否	主键。管理员唯一标识
管理员账号	account	int	4	0001-9999	否	
密码	passwd	varchar	255		否	管理员密码

数据库表名: USER

同义词(别名): 用户表

主键: 用户id

外键: 用户状态id

索引: 主键索引

约束: id:primary key,unique,not null;

email:unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
用户id	id	int	11	00000001-99999999	否	主键。用户唯

						一标识
邮箱	email	varchar	128		否	注册邮箱，用来登录，验证
密码	passwd	varchar	128		否	
用户名	name	varchar	255		否	
个人介绍	introduction	varchar	255		是	
性别	gender	varchar	2	"男"或"女" "或"保密"	否	
是否认证	isAttest	tinyint	1	0-1	否	
创建时间	date	datetime	0		否	精确到时分秒
徽章数	badgeNum	int	11	0-99999999	否	
经验	exp	int	11	0-99999999	否	
用户状态id	stateID	int	11	00000001-99999999	否	外键，关联用户状态表
用户身份	identityID	int	11	00000001-	是	外键，关联用

id				99999999		户身份 表
----	--	--	--	----------	--	----------

数据库表名：QUESTION

同义词（别名）：问题表

主键：问题 id

外键：用户 id, 问题类别 id, 问题状态 id, 问题场景 id

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否空	注解
问题 id	id	int	11	00000000001-99999999999	否	主键。问题唯一标识
用户 id	userID	int	11	00000000001-99999999999	否	该问题的提出者 id, 外键, 关联用户表
问题标题	title	varchar	30	00000000001-99999999999	否	问题标题
问题描述	content	tinytext	0		否	问题的内容
问题类别 id	typeID	int	11	00000000001-99999999999	否	外键, 关联问题类别表

问题状态 id	stateID	int	11	00000000001-99999999999	否	外键，关联到问题状态表
问题场景 id	scenarioID	int	11	00000000001-99999999999	否	外键，关联到问题场景表
问题提出时间	date	datetime	0		否	问题提出时间
是否删除	isDelete	tinyint	1		否	0 代表存在 1 代表已删除

数据库表名：ANSWER

同义词（别名）：回答表

主键：回答 id

外键：用户 id, 问题 id

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否空	注解
回答 id	id	int	11	00000000001-99999999999	否	主键。回答唯一标识

用户 id	userID	int	11	00000000001-99999999999	否	该回答的提出者 id, 外键, 关联用户表
回答内容	content	tinytext	0		否	回答的内容
问题 id	questionID	int	11	00000000001-99999999999	否	该回答对应的问题, 外键, 关联问题表
回答时间	date	Datetime	20		否	该回答提出的时间, 精确到时分秒
是否删除	isDelete	tinyint	1		否	0 代表存在 1 代表已删除

数据库表名：COMMENT

同义词（别名）：评论表

主键：评论 id

外键：用户 id, 回答 id

索引：主键索引

约束: id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否空	注解
评论id	id	int	11	00000000001-99999999999	否	主键。评论唯一标识
用户id	userID	int	11	00000000001-99999999999	否	该评论的提出者id, 外键, 关联用户表
评论内容	content	tinytext	0		否	问题的内容
回答id	answerID	int	11	00000000001-99999999999	否	该评论对应的回答, 外键, 关联回答表
评论时间	date	Datetime	20		否	该评论提出的时间, 精确到时分秒
是否删除	isDelete	tinyint	1		否	0 代表存在 1 代表已删除

数据库表名：REPLY

同义词（别名）：回复表

主键：回复 id

外键：用户 id, 评论 id

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否空	注解
回复 id	id	int	11	00000000001-99999999999	否	主键。回复唯一标识
用户 id	userID	int	11	00000000001-99999999999	否	该评论的提出者 id, 外键, 关联用户表
回复内容	content	tinytext	0		否	问题的内容
评论 id	commentID	int	11	00000000001-99999999999	否	该回复对应的评论, 外键, 关联
评论时间	date	Datetime	20		否	该评论提出的时间, 精确到

						时分秒
点赞数	count	int	11	>0	否	
是否删除	isDelete	tinyint	1		否	0 代表存在 1 代表已删除

数据库表名：REPORT

同义词（别名）：举报记录表

主键：举报 id

外键：用户 id，被举报信息 id(可为评论 id，回答 id，回复 id，问题 id)，
举报原因 id

索引：主键索引

约束：reportID:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否空	注解
举报 id	id	int	11	00000000001-99999999999	否	主键。回复唯一标识
用户 id	userID	int	11	00000000001-99999999999	否	该举报的提出者 id
举报类别	type	int	1	1-4	否	区分举报的是问题还是评论或回复

						等。
举报详情描述	int	tinytext	0		否	举报的内容描述
被举报信息 id	typeID	int	11	00000000001-99999999999	否	该举报对应的信息 id
举报时间	date	Datetime	20		否	该举报提出的时间，精确到时分秒
举报原因 id	reasonID	int	11	00000000001-99999999999	否	外键，关联到举报原因表

数据库表名：APPROVALCOMMENT

同义词（别名）：点赞评论表

主键：id

外键：用户 id，评论 id

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
------	-------	------	------	------	--------	----

点赞 评论 id	id	int	11	00000000001- 99999999999	否	主键， 唯一标识
日期	date	Datetime	0		否	
用户 id	userID	int	11	00000000001- 99999999999	否	外键， 连接用户表
评论 id	commentID	int	11	00000000001- 99999999999	否	外键， 连接评论表

数据库表名：APPROVALANSWER

同义词（别名）：支持回答表

主键：id

外键：用户 id, 回答 id

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
支持 回答 id	id	int	11	00000000001- 99999999999	否	主键， 唯一标识
日期	date	Datetime	20		否	

用户 id	userID	int	11	00000000001-99999999999	否	外键，连接用户表
回答 id	answerID	int	11	00000000001-99999999999	否	外键，连接回答表

数据库表名：OPPOSALANSWER

同义词（别名）：反对回答表

主键：id

外键：用户 id，回答 id

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
反对回答 id	id	int	11	00000000001-99999999999	否	主键，唯一标识
日期	date	Datetime	20		否	
用户 id	userID	int	11	00000000001-99999999999	否	外键，连接用户表
回答 id	answerID	Datetime	20		否	

数据库表名：MESSAGE

同义词（别名）：消息表

主键：id

外键：接受用户 id

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
消息 id	id	int	11	00000000001-99999999999	否	主键，唯一标识
日期	date	Datetime	20		否	
内容	content	text	0		否	
接受用户 id	userID	int	11	00000000001-99999999999	否	外键，连接用户表
是否已读	isRead	tinyint	1	0-1	否	

数据库表名：APPROVALREPLY

同义词（别名）：点赞回复表

主键：id

外键：用户 id, 回复 id

索引：主键索引

约束: id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
点赞 评论 id	id	int	11	00000000001- 99999999999	否	主键, 唯一标识
日期	date	Datetime	20		否	
用户 id	userID	int	11	00000000001- 99999999999	否	外键, 连接用 户表
评论 id	commentID	int	11	00000000001- 99999999999	否	外键, 连接评 论表

数据库表名: USERSTATE

同义词(别名): 用户状态表

主键: id

外键: 无

索引: 主键索引

约束: id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
id	id	int	11	00000000001- 99999999999	否	主键, 唯一标识

						识
状态	state	varchar	255		空	

数据库表名：SCHOOL

同义词（别名）：学校表

主键：id

外键：无

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
学校id	id	int	11	00000000001-999999999999	否	主键，唯一标识
学校名称	name	varchar	20		否	学校名称

数据库表名：COLLEGE

同义词（别名）：学院表

主键：id

外键：学校id

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
学院 id	id	int	11	000000000001-999999999999	否	主键, 唯一标识
学院名称	name	varchar	20		否	学院名称
学校 id	schoolID	int	11	000000000001-999999999999	否	外键, 学校 id

数据库表名：MAJOR

同义词（别名）：专业表

主键：id

外键：学院 id

索引：主键索引

约束：id:primary key, unique, not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
专业 id	id	int	11	000000000001-999999999999	否	主键, 唯一标识
专业名称	name	varchar	20		否	专业名称

学院 id	collegeID	int	11	000000000001- 999999999999	否	外键， 学院 id
----------	-----------	-----	----	-------------------------------	---	--------------

数据库表明：CATEGORIESTYPE

同义词（别名）：门类分类表

主键：id

外键：无

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
门类分类 id	id	int	11	000000000001- 999999999999	否	主键， 唯一标识
名称	name	varchar	20		否	门类分类名称

数据库表明：SUBJECTTYPE

同义词（别名）：学科分类表

主键：id

外键：无

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
学科分类 id	id	int	11	00000000001-99999999999	否	主键，唯一标识
名称	name	varchar	20		否	学科分类名称
所属门类分类 id	categoryID	int	11	00000000001-99999999999	否	外键，关联门类分类表

数据库表明：MAJORTYPE

同义词（别名）：专业分类表

主键：id

外键：无

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
专业分类 id	id	int	11	00000000001-99999999999	否	主键，唯一标识
名称	name	varchar	20		否	专业类别名称

所属学科分类id	subjectID	int	11	00000000001-99999999999	否	外键，关联学科分类表
----------	-----------	-----	----	-------------------------	---	------------

数据库表名：LEVEL

同义词（别名）：等级表

主键：id

外键：无

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
等级id	id	int	11	00000000001-99999999999	否	主键，唯一标识
等级	level	int	11	01-16	否	等级
经验值上限	expTopLimit	int	11	00000000001-99999999999	否	经验值上限
经验值下限	expBotLimit	int	11	00000000001-99999999999	否	经验值下限

数据库表名：ACHIEVEMENT

同义词（别名）：成就表

主键：id

外键：无

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
成就 id	id	int	11	00000000001-99999999999	否	主键，唯一标识
成就名字	name	varchar	20		否	成就名字

数据库表名 ACHIEVEMENTRECORD

同义词（别名）：成就记录表

主键：id

外键：用户 id, 成就 id

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
成就记录 id	id	int	11	00000000001-99999999999	否	主键, 唯一标识
日期	date	Datetime	20		否	
用户 id	userID	int	11	00000000001-99999999999	否	外键, 关联用户表
成就 id	achievementID	int	11	00000000001-99999999999	否	外键, 关联成就表

数据库表名 COLLECTIONPROBLEM

同义词（别名）：收藏记录表

主键：id

外键：用户 id, 问题 id

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
------	-------	------	------	------	--------	----

收藏记录 id	id	int	11	00000000001-99999999999	否	主键, 唯一标识
日期	date	Datetime	20		否	
用户 id	userID	int	11	00000000001-99999999999	否	外键, 关联用户表
问题 id	questionID	int	11	00000000001-99999999999	否	外键, 关联问题表

数据库表名 SEARCHHISTORY

同义词 (别名) : 搜索记录表

主键: id

外键: 用户 id

索引: 主键索引

约束: id:primary key, unique, not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
搜索记录 id	id	int	11	00000000001-99999999999	否	主键, 唯一标识
日期	date	Datetime	20		否	

用户 id	userID	int	11	00000000001-99999999999	否	外键，关联用户表
内容	content	tinytext	0		否	

数据库表名 BROWSINGHISTORY

同义词（别名）：浏览记录表

主键：id

外键：用户 id, 问题 id

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
浏览记录 id	id	int	11	00000000001-99999999999	否	主键，唯一标识
日期	date	Datetime	20		否	
用户 id	userID	int	11	00000000001-99999999999	否	外键，关联用户表
问题	questionID	int	11	00000000001-	否	外键，关联问

id				999999999999		题表
----	--	--	--	--------------	--	----

数据库表名 REPORTREASON

同义词（别名）：举报原因表

主键：id

外键：无

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
举报原因id	id	int	11	00000000001-999999999999	否	主键，唯一标识
原因内容	content	varchar	255		否	

数据库表名 REPORTTYPE

同义词（别名）：举报类别表

主键：id

外键：无

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
举报类别 id	id	int	11	000000000001-999999999999	否	主键, 唯一标识
类别	type	varchar	255		否	

数据库表名 QUESTIONTYPE

同义词（别名）：问题类别表

主键：id

外键：无

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
问题类别 id	id	int	11	000000000001-999999999999	否	主键, 唯一标识
门类分类	categoryID	int	11	000000000001-999999999999	否	外键, 关联门类分类

id						表的 id 属性
学科分类 id	subjectID	int	11	00000000001-99999999999	否	外键，关联学科分类表的 id 属性
专业分类 id	majorID	int	11	00000000001-99999999999	否	外键，关联专业分类表的 id 属性

数据库表名 QUESTIONSTATE

同义词（别名）：问题状态表

主键：id

外键：无

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
问题状态 id	id	int	11	00000000001-99999999999	否	主键，唯一标识
状态	state	varchar	10		否	问题状

						态
--	--	--	--	--	--	---

数据库表名 QUESTIONSCENARIO

同义词（别名）：问题场景表

主键：id

外键：无

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
问题场景id	id	int	11	000000000001-999999999999	否	主键，唯一标识
名称	name	varchar	255		否	

数据库表名 USERIDENTITY

同义词（别名）：用户身份表

主键：id

外键：学校分类 id, 学院分类 id, 专业分类 id

索引：主键索引

约束：id:primary key,unique,not null;

中文名称	数据属性名	数据类型	数据长度	约束范围	是否允许为空	注解
身份id	id	int	11	00000000001-99999999999	否	主键，唯一标识
学校分类id	schoolID	int	11	00000000001-99999999999	否	外键，关联学校分类表的id属性
学院分类id	collegeID	int	11	00000000001-99999999999	否	外键，关联学科分类表的id属性
专业分类id	majorID	int	11	00000000001-99999999999	是	外键，关联专业分类表的id属性
学号	studentNum	int	11	00000000001-99999999999	是	
工号	jobNum	int	11	00000000001-99999999999	是	
姓名	name	varchar	32		否	用户真实姓名
身份类型	type	varchar	0	"教师"或"学生"	否	区别该记录是学生还

						是教师
--	--	--	--	--	--	-----

4.3 数据库一致性设计

ADMIN 表:

```
CREATE TABLE `admin` (  
  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  
  `account` int(4) NOT NULL,  
  
  `passwd` varchar(255) CHARACTER SET utf8 COLLATE  
utf8_general_ci NOT NULL,  
  
  PRIMARY KEY (`id`) USING BTREE  
  
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8  
COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;
```

USER 表:

```
CREATE TABLE `user` (  
  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  
  `email` varchar(128) NOT NULL,  
  
  `passwd` varchar(128) NOT NULL,  
  
  `name` varchar(255) NOT NULL,
```

```

`introduction` varchar(255) DEFAULT NULL,

`gender` varchar(10) NOT NULL,

`isAttest` tinyint(1) NOT NULL,

`date` datetime NOT NULL,

`badgeNum` int(11) NOT NULL DEFAULT '0',

`exp` int(11) DEFAULT NULL,

`stateID` int(11) NOT NULL,

`identityID` int(11) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `stateID` (`stateID`),

KEY `identityID` (`identityID`),

CONSTRAINT `user_ibfk_1` FOREIGN KEY (`stateID`)
REFERENCES `userstate` (`id`),

CONSTRAINT `user_ibfk_2` FOREIGN KEY (`identityID`)
REFERENCES `useridentity` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

QUESTION 表:

```
CREATE TABLE `question` (  
  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  
  `userID` int(11) NOT NULL,  
  
  `title` varchar(30) NOT NULL,  
  
  `content` text NOT NULL,  
  
  `typeID` int(11) NOT NULL,  
  
  `stateID` int(11) NOT NULL,  
  
  `scenarioID` int(11) NOT NULL,  
  
  `date` datetime NOT NULL,  
  
  `isDelete` tinyint(1) NOT NULL,  
  
  PRIMARY KEY (`id`),  
  
  KEY `typeID` (`typeID`),  
  
  KEY `stateID` (`stateID`),  
  
  KEY `scenarioID` (`scenarioID`),  
  
  KEY `userID` (`userID`),
```

```

        CONSTRAINT `question_ibfk_1` FOREIGN KEY (`typeID`)
REFERENCES `questiontype` (`id`),

        CONSTRAINT `question_ibfk_2` FOREIGN KEY (`stateID`)
REFERENCES `questionstate` (`id`),

        CONSTRAINT `question_ibfk_3` FOREIGN KEY (`scenarioID`)
REFERENCES `questionscenario` (`id`),

        CONSTRAINT `question_ibfk_4` FOREIGN KEY (`userID`)
REFERENCES `user` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

ANSWER 表:

```

CREATE TABLE `answer` (

    `id` int(11) NOT NULL AUTO_INCREMENT ,

    `userID` int(11) NOT NULL,

    `content` text NOT NULL,

    `questionID` int(11) NOT NULL COMMENT ,

    `date` datetime NOT NULL,

    `isDelete` tinyint(1) NOT NULL,

```

```

PRIMARY KEY (`id`),

KEY `userID` (`userID`),

KEY `questionID` (`questionID`),

CONSTRAINT `answer_ibfk_1` FOREIGN KEY (`userID`)
REFERENCES `user` (`id`),

CONSTRAINT `answer_ibfk_2` FOREIGN KEY (`questionID`)
REFERENCES `question` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

COMMENT 表:

```

CREATE TABLE `comment` (

  `id` int(11) NOT NULL AUTO_INCREMENT,

  `userID` int(11) NOT NULL,

  `content` text NOT NULL,

  `answerID` int(11) NOT NULL,

  `date` datetime NOT NULL,

  `isDelete` tinyint(1) NOT NULL,

  PRIMARY KEY (`id`),

```

```

KEY `answerID` (`answerID`),

KEY `userID` (`userID`),

CONSTRAINT `comment_ibfk_1` FOREIGN KEY (`answerID`)
REFERENCES `user` (`id`),

CONSTRAINT `comment_ibfk_2` FOREIGN KEY (`userID`)
REFERENCES `user` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

REPLY 表:

```

CREATE TABLE `reply` (

  `id` int(11) NOT NULL AUTO_INCREMENT ,

  `userID` int(11) NOT NULL,

  `content` text NOT NULL,

  `commentID` int(11) NOT NULL,

  `date` datetime NOT NULL,

  `count` int(11) NOT NULL,

  `isDelete` tinyint(1) NOT NULL,

  PRIMARY KEY (`id`),

```

```

KEY `userID` (`userID`),

KEY `commentID` (`commentID`),

CONSTRAINT `reply_ibfk_1` FOREIGN KEY (`userID`)
REFERENCES `user` (`id`),

CONSTRAINT `reply_ibfk_2` FOREIGN KEY (`commentID`)
REFERENCES `comment` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

REPORT 表:

```

CREATE TABLE `report` (

  `id` int(11) NOT NULL AUTO_INCREMENT,

  `userID` int(11) NOT NULL,

  `type` int(1) NOT NULL,

  `int` tinytext NOT NULL,

  `typeID` int(11) NOT NULL,

  `date` datetime NOT NULL,

  `reasonID` int(11) NOT NULL,

```

```

PRIMARY KEY (`id`),

KEY `userID` (`userID`),

KEY `typeID` (`typeID`),

KEY `reasonID` (`reasonID`),

CONSTRAINT `report_ibfk_1` FOREIGN KEY (`userID`)
REFERENCES `user` (`id`),

CONSTRAINT `report_ibfk_2` FOREIGN KEY (`typeID`)
REFERENCES `reporttype` (`id`),

CONSTRAINT `report_ibfk_3` FOREIGN KEY (`reasonID`)
REFERENCES `reportreason` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

APPROVALCOMMENT 表:

```

CREATE TABLE `approvalcomment` (

  `id` int(11) NOT NULL AUTO_INCREMENT,

  `date` datetime NOT NULL,

  `userID` int(11) NOT NULL,

```

```

`commentID` int(11) NOT NULL,

PRIMARY KEY (`id`),

KEY `userID` (`userID`),

KEY `commentID` (`commentID`),

CONSTRAINT `approvalcomment_ibfk_1` FOREIGN KEY
(`userID`) REFERENCES `user` (`id`),

CONSTRAINT `approvalcomment_ibfk_2` FOREIGN KEY
(`commentID`) REFERENCES `comment` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

APPROVALANSWER 表:

```

CREATE TABLE `approvalanswer` (

`id` int(11) NOT NULL AUTO_INCREMENT,

`date` datetime NOT NULL,

`userID` int(11) NOT NULL,

`answerID` int(11) NOT NULL,

PRIMARY KEY (`id`),

```

```

KEY `userID` (`userID`),

KEY `answerID` (`answerID`),

CONSTRAINT `approvalanswer_ibfk_1` FOREIGN KEY (`userID`)
REFERENCES `user` (`id`),

CONSTRAINT `approvalanswer_ibfk_2` FOREIGN KEY
(`answerID`) REFERENCES `answer` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

OPPOSALANSWER 表:

```

CREATE TABLE `opposalanswer` (

  `id` int(11) NOT NULL AUTO_INCREMENT,

  `date` datetime NOT NULL,

  `userID` int(11) NOT NULL,

  `answerID` int(11) NOT NULL,

  PRIMARY KEY (`id`),

  KEY `userID` (`userID`),

  KEY `answerID` (`answerID`),

```

```

        CONSTRAINT `opposalanswer_ibfk_1` FOREIGN KEY (`userID`)
REFERENCES `user` (`id`),

        CONSTRAINT `opposalanswer_ibfk_2` FOREIGN KEY
(`answerID`) REFERENCES `answer` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

MESSAGE 表:

```

CREATE TABLE `message` (

    `id` int(11) NOT NULL AUTO_INCREMENT,

    `date` datetime NOT NULL,

    `userID` int(11) NOT NULL,

    `isRead` tinyint(1) NOT NULL,

    `content` text,

    PRIMARY KEY (`id`),

    KEY `userID` (`userID`),

    CONSTRAINT `userID` FOREIGN KEY (`userID`) REFERENCES
`user` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

APPROVALREPLY 表:

```
CREATE TABLE `approvalreply` (  
  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  
  `date` datetime NOT NULL,  
  
  `userID` int(11) NOT NULL,  
  
  `commentID` int(11) NOT NULL,  
  
  PRIMARY KEY (`id`),  
  
  KEY `userID` (`userID`),  
  
  KEY `commentID` (`commentID`),  
  
  CONSTRAINT `approvalreply_ibfk_1` FOREIGN KEY (`userID`)  
REFERENCES `user` (`id`),  
  
  CONSTRAINT `approvalreply_ibfk_2` FOREIGN KEY  
(`commentID`) REFERENCES `comment` (`id`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

USERSTATE 表:

```
CREATE TABLE `userstate` (  

```

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
  
`state` varchar(255) CHARACTER SET utf8 COLLATE  
utf8_general_ci NOT NULL,  
  
PRIMARY KEY (`id`) USING BTREE  
  
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8  
COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;
```

SCHOOL 表:

```
CREATE TABLE `school` (  
  
`id` int(11) NOT NULL AUTO_INCREMENT,  
  
`name` varchar(20) NOT NULL,  
  
PRIMARY KEY (`id`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

COLLEGE 表:

```
CREATE TABLE `college` (  
  
`id` int(11) NOT NULL AUTO_INCREMENT,  
  
`name` varchar(20) NOT NULL,  
  
`schoolID` int(11) NOT NULL,
```

```
PRIMARY KEY (`id`),  
  
KEY `schoolID` (`schoolID`),  
  
CONSTRAINT `college_ibfk_1` FOREIGN KEY (`schoolID`)  
REFERENCES `school` (`id`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

MAJOR 表:

```
CREATE TABLE `major` (  
  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  
  `name` varchar(20) NOT NULL,  
  
  `collegeID` int(11) NOT NULL,  
  
  PRIMARY KEY (`id`),  
  
  KEY `collegeID` (`collegeID`),  
  
  CONSTRAINT `major_ibfk_1` FOREIGN KEY (`collegeID`)  
REFERENCES `college` (`id`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

CATEGORIESTYPE 表:

```
CREATE TABLE `categoriestype` (  
  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  
  `name` varchar(20) NOT NULL,  
  
  PRIMARY KEY (`id`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

subjectType 表:

```
CREATE TABLE `subjecttype` (  
  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  
  `name` varchar(20) NOT NULL,  
  
  `categoryID` int(11) NOT NULL,  
  
  PRIMARY KEY (`id`),  
  
  KEY `categoryID` (`categoryID`),  
  
  CONSTRAINT `subjecttype_ibfk_1` FOREIGN KEY  
  (`categoryID`) REFERENCES `categoriestype` (`id`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

MAJORTYPE 表:

```
CREATE TABLE `majortype` (  
  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  
  `name` varchar(20) NOT NULL,  
  
  `subjectID` int(11) NOT NULL,  
  
  PRIMARY KEY (`id`),  
  
  KEY `subjectID` (`subjectID`),  
  
  CONSTRAINT `majortype_ibfk_1` FOREIGN KEY (`subjectID`)  
REFERENCES `subjecttype` (`id`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

LEVEL 表:

```
CREATE TABLE `level` (  
  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  
  `level` int(11) NOT NULL,  
  
  `expTopLimit` int(11) NOT NULL,
```

```
    `expBotLimit` int(11) NOT NULL,  
  
    PRIMARY KEY (`id`)  
)  
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

ACHIEVEMENT 表:

```
CREATE TABLE `achievement` (  
  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
  
    `name` varchar(20) NOT NULL,  
  
    PRIMARY KEY (`id`)  
)  
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

ACHIEVEMENTRECORD 表:

```
CREATE TABLE `achievementrecord` (  
  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
  
    `date` datetime(0) NOT NULL,  
  
    `userID` int(11) NOT NULL,
```

```

`achievementID` int(11) NOT NULL,

PRIMARY KEY (`id`) USING BTREE,

INDEX `userID` (`userID`) USING BTREE,

INDEX `achievementID` (`achievementID`) USING BTREE,

CONSTRAINT `achievementrecord_ibfk_1` FOREIGN KEY
(`userID`) REFERENCES `user` (`id`) ON DELETE RESTRICT ON
UPDATE RESTRICT,

CONSTRAINT `achievementrecord_ibfk_2` FOREIGN KEY
(`achievementID`) REFERENCES `achievement` (`id`) ON DELETE
RESTRICT ON UPDATE RESTRICT

) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;

```

COLLECTIONPROBLEM 表:

```

CREATE TABLE `collectionproblem` (

`id` int(11) NOT NULL AUTO_INCREMENT,

`date` datetime NOT NULL,

`userID` int(11) NOT NULL,

```

```

`questionID` int(11) NOT NULL,

PRIMARY KEY (`id`),

KEY `userID` (`userID`),

KEY `questionID` (`questionID`),

CONSTRAINT `collectionproblem_ibfk_1` FOREIGN KEY
(`userID`) REFERENCES `user` (`id`),

CONSTRAINT `collectionproblem_ibfk_2` FOREIGN KEY
(`questionID`) REFERENCES `question` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

SEARCHHISTORY 表:

```

CREATE TABLE `searchhistory` (

  `id` int(11) NOT NULL AUTO_INCREMENT,

  `date` datetime NOT NULL,

  `userID` int(11) NOT NULL,

  `content` varchar(255) NOT NULL,

  PRIMARY KEY (`id`),

```

```

KEY `userID` (`userID`),

CONSTRAINT `searchhistory_ibfk_1` FOREIGN KEY (`userID`)
REFERENCES `user` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

BROWSINGHISTORY 表:

```

CREATE TABLE `browsinghistory` (

  `id` int(11) NOT NULL AUTO_INCREMENT,

  `date` datetime NOT NULL,

  `userID` int(11) NOT NULL,

  `questionID` int(11) NOT NULL,

  PRIMARY KEY (`id`),

  KEY `questionID` (`questionID`),

  KEY `browsinghistory_ikbf_1` (`userID`),

  CONSTRAINT `browsinghistory_ibfk_2` FOREIGN KEY
(`questionID`) REFERENCES `question` (`id`),

```

```
CONSTRAINT `browsinghistory_ikbf_1` FOREIGN KEY  
(`userID`) REFERENCES `user` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

REPORTTYPE 表:

```
CREATE TABLE `reporttype` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `type` varchar(255) NOT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

QUESTIONTYPE 表:

```
CREATE TABLE `questiontype` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `categoryID` int(11) NOT NULL,  
    `subjectID` int(11) NOT NULL,
```

```

`majorID` int(11) NOT NULL,

PRIMARY KEY (`id`),

KEY `categoryID` (`categoryID`),

KEY `subjectID` (`subjectID`),

KEY `majorID` (`majorID`),

CONSTRAINT `questiontype_ibfk_1` FOREIGN KEY
(`categoryID`) REFERENCES `categoriestype` (`id`),

CONSTRAINT `questiontype_ibfk_2` FOREIGN KEY
(`subjectID`) REFERENCES `subjecttype` (`id`),

CONSTRAINT `questiontype_ibfk_3` FOREIGN KEY (`majorID`)
REFERENCES `majortype` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

QUESTIONSTATE 表:

```

CREATE TABLE `questionstate` (

`id` int(11) NOT NULL AUTO_INCREMENT,

`state` varchar(11) NOT NULL,

```

```
PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

QUESTIONSCENARIO 表:

```
CREATE TABLE `questionscenario` (  
  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  
  `name` varchar(255) NOT NULL,  
  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

USERIDENTITY 表:

```
CREATE TABLE `useridentity` (  
  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  
  `schoolID` int(11) NOT NULL,  
  
  `collegeID` int(11) NOT NULL,  
  
  `majorID` int(11) NULL DEFAULT NULL,
```

```

`studentNum` int(11) NULL DEFAULT NULL,

`jobNum` int(11) NULL DEFAULT NULL,

`name` varchar(32) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL,

`type` varchar(255) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL,

PRIMARY KEY (`id`),

CONSTRAINT `useridentity_ibfk_1` FOREIGN KEY (`schoolID`)
REFERENCES `school` (`id`) ON DELETE RESTRICT ON UPDATE
RESTRICT,

CONSTRAINT `useridentity_ibfk_2` FOREIGN KEY
(`collegeID`) REFERENCES `college` (`id`) ON DELETE
RESTRICT ON UPDATE RESTRICT,

CONSTRAINT `useridentity_ibfk_3` FOREIGN KEY (`majorID`)
REFERENCES `major` (`id`) ON DELETE RESTRICT ON UPDATE
RESTRICT

) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;

```

物理实现

5.1 数据库的安排

使用阿里云 RDS，MySQL 版本 5.7，存储引擎 InnoDB。

5.2 安全保密设计

- a. 使用阿里云安全策略进行数据库防护。
- b. 密码等需要保密的字段使用密文存储。