<有课>

软件架构文档

版本 <0.2>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2019.9.9 | 1.0 |  | 郭志 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 参考资料 4

2. 用例视图 4

3. 逻辑视图 4

3.1 概述 4

3.2 在构架方面具有重要意义的设计包 4

4. 进程视图 4

5. 部署视图 4

6. 实现视图 5

7. 数据视图（可选） 5

8. 核心算法设计（可选） 5

软件架构文档 （简化版）

# 简介

## 目的

本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

## 参考资料

1.有课立项建议书，C18-有课小组

2.有课需求规约，C18-有课小组

# 用例视图

本章是对软件架构的用例视图的描述。

## 主要用例

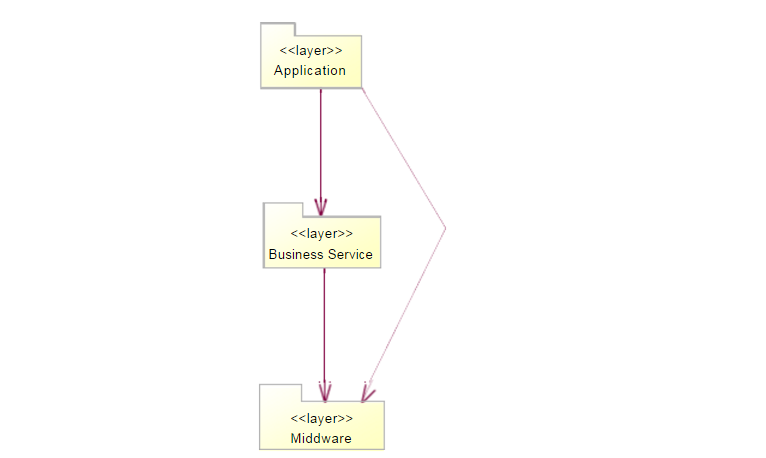
## IMG_256

图1 有课系统用例图

# 逻辑视图

本章是对软件架构的逻辑视图的描述。主要内容包括描述重要的类，类的分包，子系统以及子系统的分层等。

## 概述



有课系统的逻辑视图主要由三层组成，分别是Application层,Business Service层和Middle层。

Application层主要由相应用户请求的各种controller组成，它会调用Bussiness Service层中的函数进行业务逻辑处理，同时将处理结果以json形式返回给用户。

Bussiness Service层主要完成实际的业务逻辑，同时包括与数据库表OR映射的entiry类，以及访问数据库的repository层，和封装数据库的dao层。

Middle层为开发过程中用到的函数库。

## Application层

## 

Application主要 10 个子包组成，分别为UserController, CourseController, JaccountController， ReportController, TimeController, CourseCommentController, CourseEvaluateController, CourseGreatMomentController, CourseQuestionController，CourseRecommendController。

UserController包主要包含loginController,specificController，banController，unbanController,allController

CourseController包主要包含updateCourseController, searchCourseController, specificCourseController。

JaccountController包主要包含loginController,profileController

ReportController 包主要包含 reportController， getHandledController， getReportedController，handleController

TimeController 包主要包含getWeekController

CourseCommentController 包主要包含allController, findController, addController, deleteController, banController ,unbanController,praiseController , unpraiseController, replyController。

CourseEvaluateController 包主要包含 allController, findController, addController, praiseControoler, unpraiseController。

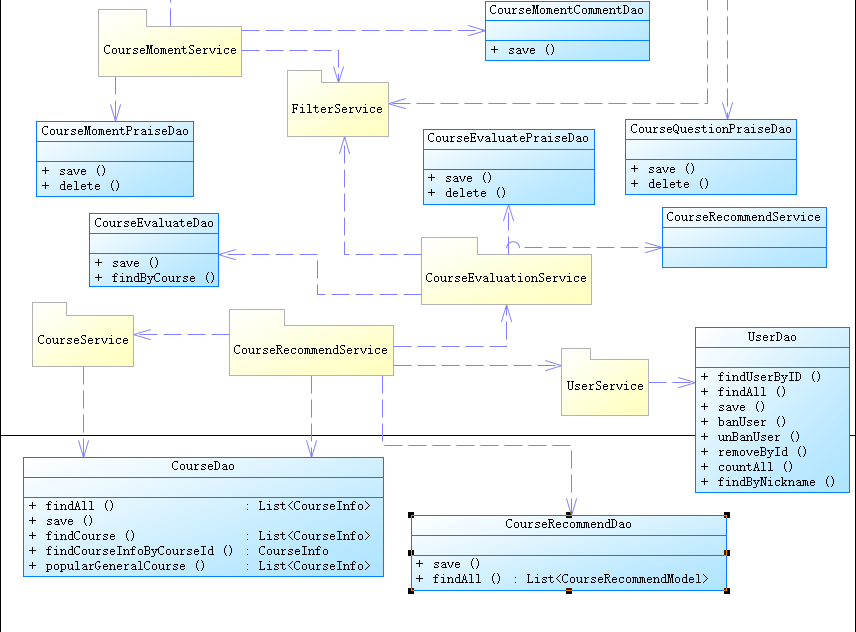
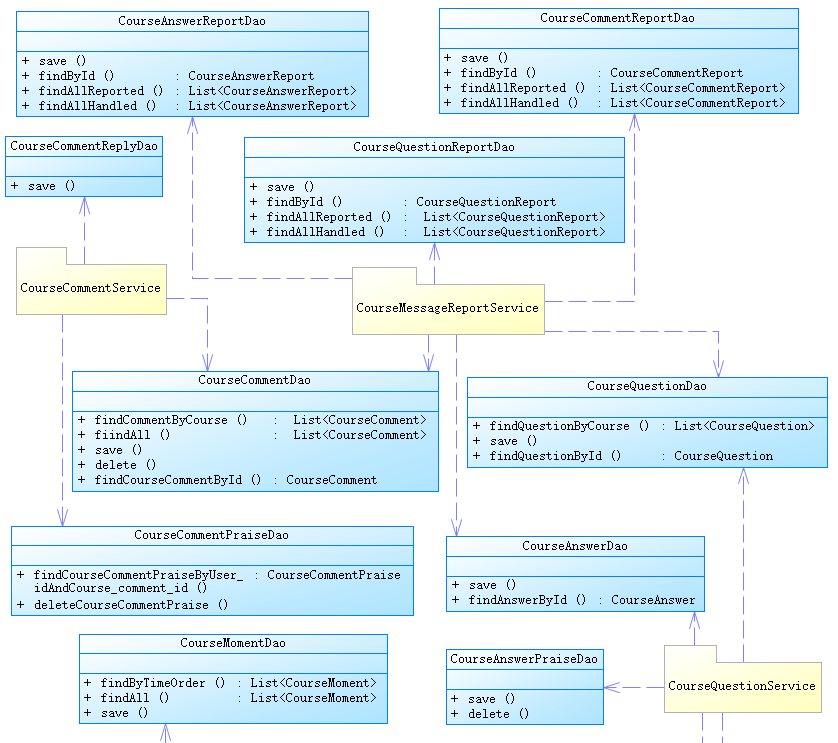
CourseGreatMomentController 包主要包含findAllController, findController, postController, praiseController ,unpraiseController，commentController，uploadContoller，

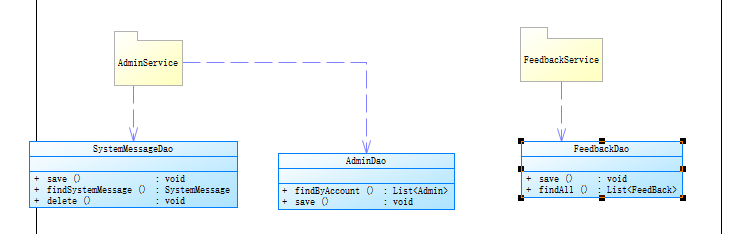
CourseQuestionController 包主要包含 findController, addController, addController, praiseController, unpraiseController。

CourseRecommendController 包主要包含findController, generateController, loadController

## Business Service层

## Business Service层包括Service，dao,repository,entity四个包。





### Service包

### Service包由9个包组成：

### UserService包 负责处理用户相关事务，

### CourseService包 负责处理课程相关事务。

CourseMomentService包 负责处理精彩瞬间相关事务

CourseCommentService包 负责处理课程评论相关事务

CourseQuestionService包 负责处理课程问题相关事务

CourseMessageService包 负责处理举报相关事务

CourseEvaluationService包 负责处理课程评测相关事务

CourseRecommendService包 负责处理课程推荐相关事宜

FilterService包 负责进行敏感词过滤相关事务

AdminService包 负责管理员的登陆与操作

FeedbackService包 负责反馈。

## Middleware层

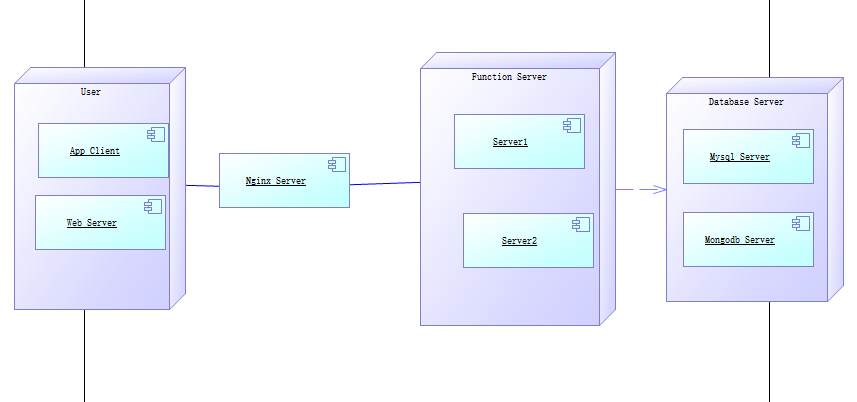
## Middle层为开发过程中用到的函数库。

# 进程视图

无

# 部署视图

本章描述部署和运行软件的物理网络配置。



## User Client

用户主要通过浏览器来访问系统，客户端与服务端的连接是互联网。

## Nginx

由Nginx分配任务给应用服务器，达到负载均衡的目的。

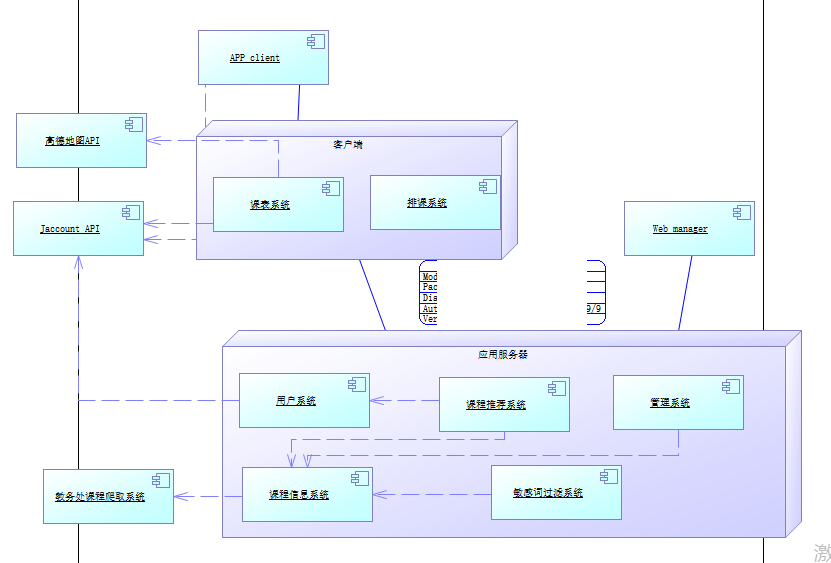
## Server

应用服务器运行有课系统，有课系统部署在Tomcat7.0容器上，它与数据服务器通过jdbc连接。

## DB Server

数据服务器运行MySQL、MongoDB数据库。

# 实现视图

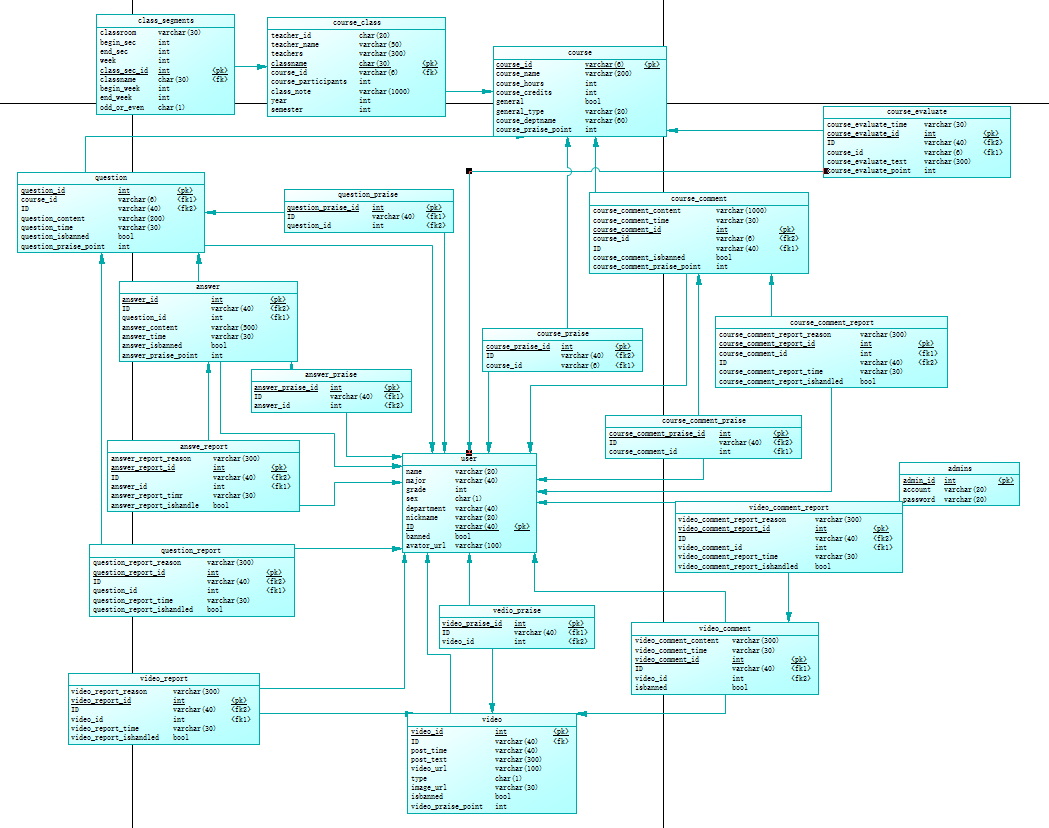


# 数据视图（可选）

Mongodb主要内容：

课程评测的用户自定义内容。

Mysql PDM:



## 附录

## Mysql建表语句

/\*==============================================================\*/

/\* DBMS name:      MySQL 5.0                                    \*/

/\* Created on:     2019/9/1 15:48:35                            \*/

/\*==============================================================\*/

drop table if exists admins;

drop table if exists answer;

drop table if exists answer\_praise;

drop table if exists answer\_report;

drop table if exists class\_segments;

drop table if exists course;

drop table if exists course\_class;

drop table if exists course\_comment;

drop table if exists course\_comment\_praise;

drop table if exists course\_comment\_report;

drop table if exists course\_evaluate;

drop table if exists course\_praise;

drop table if exists course\_recommend\_data\_model;

drop table if exists feedback;

drop table if exists question;

drop table if exists question\_praise;

drop table if exists question\_report;

drop table if exists system\_message;

drop table if exists user;

drop table if exists video;

drop table if exists video\_comment;

drop table if exists video\_comment\_report;

drop table if exists video\_praise;

drop table if exists video\_report;

/\*==============================================================\*/

/\* Table: admins                                                \*/

/\*==============================================================\*/

create table admins

(

   admin\_id             int not null auto\_increment,

   account              varchar(20),

   password             varchar(20),

   primary key (admin\_id)

);

/\*==============================================================\*/

/\* Table: answer                                                \*/

/\*==============================================================\*/

create table answer

(

   answer\_id            int not null auto\_increment,

   ID                   varchar(40),

   question\_id          int,

   answer\_content       varchar(500),

   answer\_time          varchar(30),

   answer\_isbanned      bool,

   answer\_praise\_point  int,

   primary key (answer\_id)

);

/\*==============================================================\*/

/\* Table: answer\_praise                                         \*/

/\*==============================================================\*/

create table answer\_praise

(

   answer\_praise\_id     int not null auto\_increment,

   ID                   varchar(40),

   answer\_id            int,

   primary key (answer\_praise\_id)

);

/\*==============================================================\*/

/\* Table: answer\_report                                         \*/

/\*==============================================================\*/

create table answer\_report

(

   answer\_report\_reason varchar(300),

   answer\_report\_id     int not null auto\_increment,

   ID                   varchar(40),

   answer\_id            int,

   answer\_report\_time   varchar(30),

   answer\_report\_ishandled bool,

   primary key (answer\_report\_id)

);

/\*==============================================================\*/

/\* Table: class\_segments                                        \*/

/\*==============================================================\*/

create table class\_segments

(

   classroom            varchar(30),

   begin\_sec            int,

   end\_sec              int,

   week                 int,

   class\_sec\_id         int not null auto\_increment,

   classname            char(30),

   begin\_week           int,

   end\_week             int,

   odd\_or\_even          char(1),

   primary key (class\_sec\_id)

);

/\*==============================================================\*/

/\* Table: course                                                \*/

/\*==============================================================\*/

create table course

(

   course\_id            varchar(6) not null,

   course\_name          varchar(200),

   course\_hours         int,

   course\_credits       int,

   general              bool,

   general\_type         varchar(20),

   course\_deptname      varchar(60),

   course\_praise\_point  int,

   primary key (course\_id)

);

/\*==============================================================\*/

/\* Table: course\_class                                          \*/

/\*==============================================================\*/

create table course\_class

(

   teacher\_id           char(20),

   teacher\_name         varchar(50),

   teachers             varchar(300),

   classname            char(30) not null,

   course\_id            varchar(6),

   course\_participants  int,

   class\_note           varchar(1000),

   year                 int,

   semester             int,

   primary key (classname)

);

/\*==============================================================\*/

/\* Table: course\_comment                                        \*/

/\*==============================================================\*/

create table course\_comment

(

   course\_comment\_content varchar(1000),

   course\_comment\_time  varchar(30),

   course\_comment\_id    int not null auto\_increment,

   course\_id            varchar(6),

   ID                   varchar(40),

   course\_comment\_isbanned bool,

   course\_comment\_praise\_point int,

   primary key (course\_comment\_id)

);

/\*==============================================================\*/

/\* Table: course\_comment\_praise                                 \*/

/\*==============================================================\*/

create table course\_comment\_praise

(

   course\_comment\_praise\_id int not null auto\_increment,

   ID                   varchar(40),

   course\_comment\_id    int,

   primary key (course\_comment\_praise\_id)

);

/\*==============================================================\*/

/\* Table: course\_comment\_report                                 \*/

/\*==============================================================\*/

create table course\_comment\_report

(

   course\_comment\_report\_reason varchar(300),

   course\_comment\_report\_id int not null auto\_increment,

   course\_comment\_id    int,

   ID                   varchar(40),

   course\_comment\_report\_time varchar(30),

   course\_comment\_report\_ishandled bool,

   primary key (course\_comment\_report\_id)

);

/\*==============================================================\*/

/\* Table: course\_evaluate                                       \*/

/\*==============================================================\*/

create table course\_evaluate

(

   course\_evaluate\_time varchar(30),

   course\_evaluate\_id   int not null auto\_increment,

   ID                   varchar(40),

   course\_id            varchar(6),

   course\_evaluate\_praise\_point int,

   course\_evaluate\_point int,

   primary key (course\_evaluate\_id)

);

/\*==============================================================\*/

/\* Table: course\_praise                                         \*/

/\*==============================================================\*/

create table course\_praise

(

   course\_praise\_id     int not null auto\_increment,

   ID                   varchar(40),

   course\_id            varchar(6),

   primary key (course\_praise\_id)

);

/\*==============================================================\*/

/\* Table: course\_recommend\_data\_model                           \*/

/\*==============================================================\*/

create table course\_recommend\_data\_model

(

   recommend\_id         int not null,

   user\_id              bigint,

   lcourse\_id           bigint,

   evaluate\_point       int,

   evaluate\_time        bigint,

   primary key (recommend\_id)

);

/\*==============================================================\*/

/\* Table: feedback                                              \*/

/\*==============================================================\*/

create table feedback

(

   ID                   varchar(40),

   content              varchar(3000),

   time                 varchar(40),

   feedback\_id          int not null auto\_increment,

   primary key (feedback\_id)

);

/\*==============================================================\*/

/\* Table: question                                              \*/

/\*==============================================================\*/

create table question

(

   question\_id          int not null auto\_increment,

   course\_id            varchar(6),

   ID                   varchar(40),

   question\_content     varchar(200),

   question\_time        varchar(30),

   question\_isbanned    bool,

   question\_praise\_point int,

   primary key (question\_id)

);

/\*==============================================================\*/

/\* Table: question\_praise                                       \*/

/\*==============================================================\*/

create table question\_praise

(

   question\_praise\_id   int not null auto\_increment,

   ID                   varchar(40),

   question\_id          int,

   primary key (question\_praise\_id)

);

/\*==============================================================\*/

/\* Table: question\_report                                       \*/

/\*==============================================================\*/

create table question\_report

(

   question\_report\_reason varchar(300),

   question\_report\_id   int not null auto\_increment,

   ID                   varchar(40),

   question\_id          int,

   question\_report\_time varchar(30),

   question\_report\_ishandled bool,

   primary key (question\_report\_id)

);

/\*==============================================================\*/

/\* Table: system\_message                                        \*/

/\*==============================================================\*/

create table system\_message

(

   admin\_id             int,

   content              varchar(3000),

   image\_url            varchar(30),

   time                 varchar(40),

   message\_id           int not null auto\_increment,

   primary key (message\_id)

);

/\*==============================================================\*/

/\* Table: user                                                  \*/

/\*==============================================================\*/

create table user

(

   name                 varchar(20),

   major                varchar(40),

   grade                int,

   sex                  char(1),

   department           varchar(40),

   nickname             varchar(20),

   ID                   varchar(40) not null,

   banned               bool,

   avatar\_url           varchar(100),

   avator\_url           varchar(100),

   primary key (ID)

);

/\*==============================================================\*/

/\* Table: video                                                 \*/

/\*==============================================================\*/

create table video

(

   video\_id             int not null auto\_increment,

   ID                   varchar(40),

   post\_time            varchar(40),

   post\_text            varchar(300),

   video\_url            varchar(100),

   type                 char(1),

   image\_url            varchar(30),

   isbanned             bool,

   video\_praise\_point   int,

   primary key (video\_id)

);

/\*==============================================================\*/

/\* Table: video\_comment                                         \*/

/\*==============================================================\*/

create table video\_comment

(

   video\_comment\_content varchar(300),

   video\_comment\_time   varchar(30),

   video\_comment\_id     int not null auto\_increment,

   ID                   varchar(40),

   video\_id             int,

   isbanned             bool,

   primary key (video\_comment\_id)

);

/\*==============================================================\*/

/\* Table: video\_comment\_report                                  \*/

/\*==============================================================\*/

create table video\_comment\_report

(

   video\_comment\_report\_reason varchar(300),

   video\_comment\_report\_id int not null auto\_increment,

   ID                   varchar(40),

   video\_comment\_id     int,

   video\_comment\_report\_time varchar(30),

   video\_comment\_report\_ishandled bool,

   primary key (video\_comment\_report\_id)

);

/\*==============================================================\*/

/\* Table: video\_praise                                          \*/

/\*==============================================================\*/

create table video\_praise

(

   video\_praise\_id      int not null auto\_increment,

   ID                   varchar(40),

   video\_id             int,

   primary key (video\_praise\_id)

);

/\*==============================================================\*/

/\* Table: video\_report                                          \*/

/\*==============================================================\*/

create table video\_report

(

   video\_report\_reason  varchar(300),

   video\_report\_id      int not null auto\_increment,

   ID                   varchar(40),

   video\_id             int,

   video\_report\_time    varchar(30),

   video\_report\_ishandled bool,

   primary key (video\_report\_id)

);

alter table answer add constraint FK\_Relationship\_34 foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table answer add constraint FK\_Relationship\_5 foreign key (question\_id)

      references question (question\_id) on delete restrict on update restrict;

alter table answer\_praise add constraint FK\_Relationship\_29 foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table answer\_praise add constraint FK\_Relationship\_30 foreign key (answer\_id)

      references answer (answer\_id) on delete restrict on update restrict;

alter table answer\_report add constraint FK\_Relationship\_11 foreign key (answer\_id)

      references answer (answer\_id) on delete restrict on update restrict;

alter table answer\_report add constraint FK\_report\_answer foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table class\_segments add constraint FK\_segment foreign key (classname)

      references course\_class (classname) on delete restrict on update restrict;

alter table course\_class add constraint FK\_Relationship\_10 foreign key (course\_id)

      references course (course\_id) on delete restrict on update restrict;

alter table course\_comment add constraint FK\_Relationship\_3 foreign key (course\_id)

      references course (course\_id) on delete restrict on update restrict;

alter table course\_comment add constraint FK\_post\_course\_comment foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table course\_comment\_praise add constraint FK\_Relationship\_23 foreign key (course\_comment\_id)

      references course\_comment (course\_comment\_id) on delete restrict on update restrict;

alter table course\_comment\_praise add constraint FK\_Relationship\_24 foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table course\_comment\_report add constraint FK\_Relationship\_15 foreign key (course\_comment\_id)

      references course\_comment (course\_comment\_id) on delete restrict on update restrict;

alter table course\_comment\_report add constraint FK\_reprot\_course\_comment foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table course\_evaluate add constraint FK\_Relationship\_19 foreign key (course\_id)

      references course (course\_id) on delete restrict on update restrict;

alter table course\_evaluate add constraint FK\_evaluate\_course foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table course\_praise add constraint FK\_Relationship\_25 foreign key (course\_id)

      references course (course\_id) on delete restrict on update restrict;

alter table course\_praise add constraint FK\_Relationship\_26 foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table feedback add constraint FK\_Relationship\_36 foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table question add constraint FK\_Relationship\_33 foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table question add constraint FK\_ask foreign key (course\_id)

      references course (course\_id) on delete restrict on update restrict;

alter table question\_praise add constraint FK\_Relationship\_27 foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table question\_praise add constraint FK\_Relationship\_28 foreign key (question\_id)

      references question (question\_id) on delete restrict on update restrict;

alter table question\_report add constraint FK\_Relationship\_13 foreign key (question\_id)

      references question (question\_id) on delete restrict on update restrict;

alter table question\_report add constraint FK\_Relationship\_14 foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table system\_message add constraint FK\_Relationship\_35 foreign key (admin\_id)

      references admins (admin\_id) on delete restrict on update restrict;

alter table video add constraint FK\_Relationship\_22 foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table video\_comment add constraint FK\_Relationship\_7 foreign key (video\_id)

      references video (video\_id) on delete restrict on update restrict;

alter table video\_comment add constraint FK\_post\_video\_comment foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table video\_comment\_report add constraint FK\_Relationship\_17 foreign key (video\_comment\_id)

      references video\_comment (video\_comment\_id) on delete restrict on update restrict;

alter table video\_comment\_report add constraint FK\_Relationship\_18 foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table video\_praise add constraint FK\_Relationship\_31 foreign key (ID)

      references user (ID) on delete restrict on update restrict;

alter table video\_praise add constraint FK\_Relationship\_32 foreign key (video\_id)

      references video (video\_id) on delete restrict on update restrict;

alter table video\_report add constraint FK\_Relationship\_8 foreign key (video\_id)

      references video (video\_id) on delete restrict on update restrict;

alter table video\_report add constraint FK\_Relationship\_9 foreign key (ID)

      references user (ID) on delete restrict on update restrict;