

基于 js 的电影售票系统

说明：本文档参考国标“13 - 软件(结构)设计说明(SDD)”和案例中“四、网上购书系统概要设计说明书”等资料自定而成。

1. 引言.....	3
1.1 标识.....	3
1.2 系统概述.....	3
1.3 文档概述.....	3
1.4 基线.....	4
2. 引用文件.....	4
3. 可行性分析的前提.....	4
3.1 项目的要求.....	4
3.2 项目的目标.....	5
3.3 运行环境.....	6
3.4 约束条件.....	6
4. 总体设计.....	7
4.1 基本功能.....	7
4.2 附加功能.....	7
5. CSCI 体系结构设计.....	8
5.1 体系结构.....	8
5.2 全局数据结构说明.....	11
5.3 CSCI 部件.....	11
5.4 执行概念.....	16
6. 后台数据表.....	19
7. 尚未解决的问题.....	24
8. 注解.....	24
9. 故障树及其相应的割集树.....	26
10. 保证项目质量属性所采取的策略.....	30
11. 附录.....	31
A. 个人学习报告.....	31
B. SQL 核心数据.....	49

1. 引言

1.1 标识



系统标识符：

名称：基于 Html 的电影售票系统

畅影影业

版本号：1.0.0

保密级别：普通

1.2 系统概述

应课程要求，建立软件工程初步项目。

本团队实现基于 web 的电影购票网站。

提出方要求：(1) 支持用户在线浏览电影售票页面、热映电影信息、电影排行榜；

(2) 支持用户搜索附近影院；

(3) 支持用户在线购票、实时支付、退款请求等；

(4) 用户在观影结束后，可以对影院环境以及影片质量进行评价。

(5) 系统 UI 界面简洁大方，操作简单，实时更新电影影院数据；

项目目的：建立一个成规模的在线网络电影售票网址，实现上述功能。

实现环境为 html 环境，即完全架构在服务器上的浏览器运行模式。

限制条件大概限制为电脑用户，手机上运行此网页效果不太良好。

1.3 文档概述

本文档适用于 html 电影售票：畅影影业（网站）。

此项目为学生课程设计项目，目前处于开发阶段。

投资方：暂无，可以规划为学生自身；

需求方：目标项目组成员以及老师助教验收用；

用户：学生（开放性网址，均可以登录并进行合法操作）；

开发方：本组组员；

支持机构：暂无

1.4 基线

畅影影业可行性分析研究报告（FAR）（已完成）

2. 引用文件

《软件工程（第4版）》

《Architecture description template for use with ISO/IEC/IEEE
42010:2011》

《软件(结构)设计说明》(SDD, Software Design Document)

GB-T-8567-2006 计算机软件文档编制规范

山东大学软件工程课程课件

3. 可行性分析的前提

3.1 项目的要求

实现以下全部功能。

(1) 注册登入功能：

在登入页面（注册页面）登入账号（注册账号）即可。有管理员和普通用户两种登入方式，管理员登入可以对电影排版，哪个电影可以上映，在哪里上映等做出修改。

(2) 买票功能：

登入之后,在首页的正在热映页面可以看到如今正在热卖的电影,可能会快速找到自己喜爱的电影。

我们也可以点击首页的右方电影按钮,然后在里面可以查看如今所有在映的电影。我们也可以在影院页面查看是否有自己喜欢的电影,每个影院都有自己上映的电影的清单。

最后我们可以直接在首页上方的搜索界面搜索想要看的电影。

在上述所有操作中只要找到想要购买的电影票,都可以在当前页面购买。

(3) 搜索功能:

在首页上方直接搜索,可以搜索自己想看的电影,或者自己心仪的电影院。

(4) 预购功能:

在首页的即将上映页面又即将上映的电影,我们可以在里面进行预购。

(5) 查看订单功能:

在右下方“我的”界面可有“我的订单”功能,在这里我们可以看到自己的所有历史订单,清楚自己的每一笔消费。

(6) 想看的电影:

也是在右下方“我的”界面右边“想看的电影”可以在那里写入想看的电影。

(7) 看过的电影:

同是“我的”界面,我们可以查看自己所有看过的电影,留下自己的点点滴滴。

(8) 评论功能:

可以再电影下进行评论,谈谈自己对电影的感受等。

3.2 项目的目标

- a. 本次开发的目标是完成开发一个能在很多平台上运行的电影购票网站,是为解决多种平台下的用户购票和选票的问题。网站可以为多种平台下的用户提供电影信息和评价等信息,方便用户选择电影。同时对于电影的放映时间和地点会进行同步更新。能够使用户更加全方面地安排自己地购票安排

- b. 购票网站要求实现的功能有搜索电影，查看电影信息、对电影进行评分、对电影进行评价、购票、选座、点赞、收藏等功能
- c. 本次开发产品为独立产品，系统的层次结构图如下

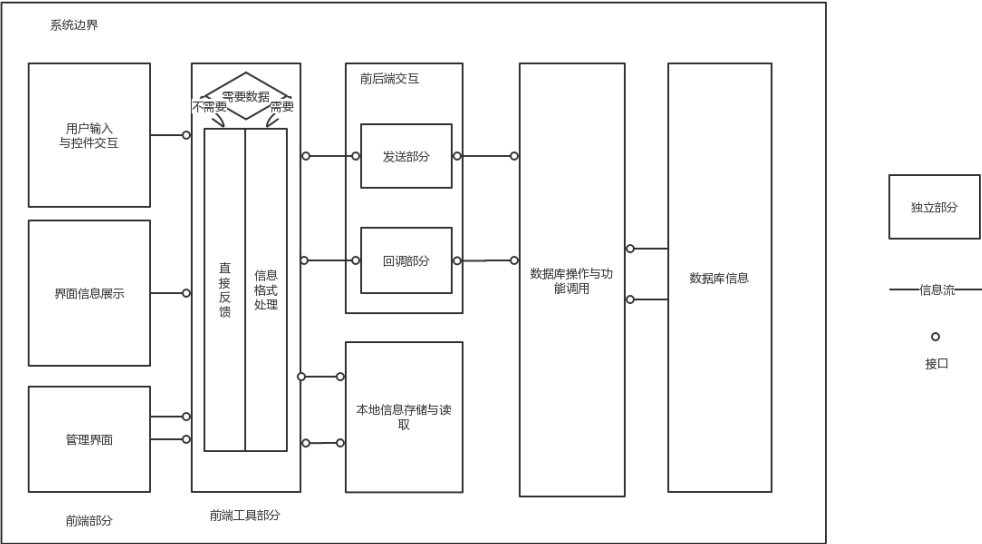


图 1 系统层次结构图

3.3 运行环境

此次开发是以 B/S 架构为模型构建，硬件环境包括前端程序运行环境为客户端的浏览器、后端程序的运行环境为服务器。程序语言支持环境包括前端的 HTML、CSS、JavaScript 环境，以及后端的 express 运行环境。储存支持环境需要在后端配置 Mysql 环境。

3.4 约束条件

本次开发的约束条件如下：经费限制不考虑、开发期限为 10~12 周、采用的方法和技术比较容易实现，不能投入过多的人员和时间。过滤掉政治导向不正确的曲目和言论等信息、遵循严禁盗用他人版权等法律限制、转播积极向上的正能量信息。

4. 总体设计

4.1 基本功能

- (a) 基于 web 的电影购票网站
- (b) 注册登录用户可在线浏览热映电影信息，电影排行榜，；
- (c) 注册登录用户查看附近影院，支持在线购票，实时支付等；
- (d) 用户在观影结束后，可以对影院环境以及影片质量进行评价
- (e) 系统 UI 界面简洁大方，操作简单，实时更新电影影院数据。
- (f) 具有良好的平台可移植性，支持多人同时在线访问。
- (g) 网页刷新时间等待时间不超过 2s；用户能够流畅使用各项功能。

4.2 附加功能

- (a) 安全与保密功能系统涉及用户私密信息，对于整个系统，需要完整的权限控制，每个用户分配特定的用户视图所应有的访问权限，防止某些人恶意的攻击系统，修改原始记录。设置必要的异常处理机制，比如当出现系统未知的操作时。也会弹出相应的错误提示等，以尽可能地保证用户体验。
- (b) 客户账号管理功能：管理人员能够列表显示用户的帐号信息、禁用或删除选定的用户帐号，在收到客户请求之后也可更改客户的账号信息。
- (c) 通知功能：管理人员可通过软件内消息向用户发送通知。
- (d) 电子邮件/短信服务：管理人员能够在客户账号信息出现异常时通过绑定的电子邮箱/手机号码向用户发送邮件/短信提醒。
- (e) 评论维护功能：对于违规评论，用户可向管理人员发送反馈，反馈人数达到一定程度即向管理人员显示该条消息，管理人员可以对其进行删除并发送消息通知该评论所属用户

5. CSCI 体系结构设计

5.1 体系结构

- 程序(模块)划分

模块名	子程序	功能
用户	查找电影、影院	根据关键词查找电影/影院
	查看正在热映/即将上映	显示最近时间上映热点电影
	用户订单	查看自己的订单
	想看/看过电影	查看收藏/已看过的电影
	评论电影	评论看过的电影/想看电影
管理员	用户管理	管理用户信息
	电影管理	管理电影信息
	影院管理	管理影院信息
	影厅管理	管理影厅信息
	评论管理	管理评论
	订单管理	管理订单
登陆系统	登录	显示用户登录信息

- 重要函数代码简要说明:

1、搜索影院相关

原理: 执行<template>部分构件 html 页面并调用<style>部分进行样式渲染。执行 js 代码, 经 `import {matchCinemaByName} from '../.../api'` 引入后端 router 内相关代码。

前端:

SearchCinema.vue

```
import {matchCinemaByName} from '../.../api'
export default {
  name: "SearchCinema",
  data(){
  },
  watch:{
    .....  },
  computed:{
    .....  }
```

后端:


```

router.get('/api/matchMovieByName',function (req,res)
{
    .....
});

```

2、订单组件

原理同上，多引入了 mintui（饿了么团队推出的前端 js 组件）

前端：

```

import {getOrderByUserId} from '../.../api'
import {Indicator} from 'mint-ui'
export default {
    name: "MyOrder",
    data(){
        .....
    },
    created() {
        Indicator.open('Loading...');
        this.loadInfo();
    },
    methods:{
        //加载信息
        async loadInfo(){
            .....
        },
        //格式座位
        formatSeat(num){
            略
        }
    }
}
</script>

```

后端：

```

router.get('/api/getOrderByUserId',function (req,res) {
    let userId = req.query.userId;
    let sqlStr = 'SELECT * FROM t_order INNER JOIN t_schedule ON t_order.schedule_id = t_schedule.schedule_id INNER JOIN t_movie ON t_movie.movie_id = t_schedule.movie_id INNER JOIN t_cinema ON t_cinema.cinema_id = t_schedule.cinema_id WHERE user_id = ?;';
    conn.query(sqlStr,[userId],(error,result,field)=>{

```

```

    if (error){
      console.log(error);
    } else{
      result = JSON.parse(JSON.stringify(result));
      res.json({success_code:200,data:result});
    }
  });
});

```

3、选座功能

前端:

```

import {getMovieDetail,getCurrentCinemaDetail,getScheduleById,updateScheduleSeat} from '../api/index'
export default {
  name: "SelectSeat",
  data(){
    return{
      .....
      seatIJ:[
        [0,0,0,0,0,0,0,0,0,0],
        [0,0,0,0,0,0,0,0,0,0],
        [0,0,0,0,0,0,0,0,0,0],
        [0,0,0,0,0,0,0,0,0,0],
        [0,0,0,0,0,0,0,0,0,0],
        [0,0,0,0,0,0,0,0,0,0]
      ],
    }
  },

  methods:{
    //加载信息
    .....
    //选择座位
    .....
    //取消选座
    cancelSelectedSeat(indexI,indexJ){
      .....
    },
    //确认选座
    async ensureSeatBtn(){
      .....
    }
  }
}

```

后端:

使用了

getMovieDetail, *getCurrentCinemaDetail*, *getScheduleById*, *updateScheduleSeat* 四个模块函数。

其中以*getMovieDetail*为例实现如下:

```
router.get('/api/getMovieDetail',function (req,res) {
  let movieId = req.query.movieId;
  let sqlStr = 'SELECT * FROM t_movie WHERE movie_id = ? LIMIT 1;';
  conn.query(sqlStr,[movieId],(error,result,field)=>{
    if (error){
      .....
    } else{
      .....
    }
  });
});
```

5.2 代码目录文件说明

目录/文件		说明
Build	项目构建(webpack)相关代码	
Config	配置目录，包括端口号等。我们初学可以使用默认的。	
Node_modules	Npm 加载的项目依赖模块	
Src	这里是我们要开发的目录，基本上要做的事情都在这个目录里。里面包含了几个目录及文件	
	assests	放置一些图片，如 logo 等。
	components	目录里面放了一个组件文件，可以不用。
	App. vue	项目入口文件，我们也可以直接将组件写这里，而不使用 components 目录。main. js:项目的核心文件。
Static	静态资源目录，如图片、字体等	
Test	初始测试目录，可删除	
.xxx 文件	这些是一些配置文件，包括语法配置、git 配置等	
Index. html	首页入口文件，可以添加一些 meta 信息或统计代码	
Package. json	项目配置文件	
READEME. md	项目的说明文档，markdown 格式	

5.3 CSCI 部件

A. 标识符

模块： 前端界面模块（用户、管理员）

样式主题模块

数据库模块（数据库）

用户逻辑树：

```
|-- .gitignore  #git 中忽略的文件
|-- index.html  #根文件
|-- package.json  #项目描述文件
|-- README.md  #项目介绍文件
|-- build  #webpack 开发和打包的配置文件
|-- config  #vue-cli 的配置文件
|-- src  #资源文件夹
|   |-- App.vue  #页面入口文件
|   |-- main.js  #程序入口文件
|   |-- api  #请求接口文件
|   |   |-- ajax.js  #基于 axios 封装 ajax 文件
|   |   |-- index.js  #调用接口文件
|   |-- common  #公共资源文件
|   |   |-- css
|   |   |   |-- style.css  #字体图标样式文件
|   |   |   |-- fonts  #字体图标文件
|   |   |-- util  #公共工具方法文件
|   |       |-- util.js
|   |-- components  #组件文件
|   |   |-- CinemaDetail  #影院详情组件
|   |   |-- DatePicker  #日期选择器组件
|   |   |-- Login  #登录组件
|   |   |-- MovieDetail  #电影详情组件
|   |   |   |-- MovieDetail.vue
|   |   |   |-- children
|   |   |       |-- CommentPanel.vue  #影评组件
|   |   |-- MovieItem  #单个电影组件
|   |   |-- Pay  #支付组件
|   |   |-- SelectCinema  #选择影院组件
|   |   |-- SelectSeat  #选择座位组件
|   |   |-- SubmitOrder  #提交订单组件
|   |   |-- TabBar  #底部导航组件
|   |-- pages  #页面文件
|   |   |-- Cinema  #影院页
|   |   |   |-- Cinema.vue
|   |   |   |-- children
|   |   |       |-- SearchCinema.vue  #搜索影院页
```

```

| | | |-- Home #主页
| | | | |-- Home.vue
| | | | |-- children
| | | | | |-- SearchAll.vue #全局搜索页
| | | |-- Movie #电影页
| | | | |-- Movie.vue
| | | | |-- children
| | | | | |-- SearchMovie.vue #搜索电影页
| | | |-- My #个人页
| | | | |-- My.vue
| | | | |-- children
| | | | | |-- ModifyUserName.vue #修改用户名页
| | | | | |-- ModifyUserSign.vue #修改性别页
| | | | | |-- MyInfo.vue #个人信息页
| | | | | |-- MyMovie.vue #收藏电影页
| | | | | |-- MyOrder.vue #订单页
| |-- router #路由
| | |-- index.js #路由配置
| |-- store #vuex 状态管理
|-- static #静态文件
    |-- css
        |-- adapter.css #屏幕自适应样式
        |-- reset.css #兼容性样式

```

管理员逻辑树:

```

|-- .gitignore #git 中忽略的文件
|-- index.html #根文件
|-- package.json #项目描述文件
|-- README.md #项目介绍文件
|-- build #webpack 开发和打包的配置文件
|-- config #vue-cli 的配置文件
|-- src #资源文件夹
|   |-- App.vue #页面入口文件
|   |-- main.js #程序入口文件
|   |-- api #请求接口文件
|   |   |-- ajax.js #基于 axios 封装 ajax 文件
|   |   |-- index.js #调用接口文件
|   |-- common #公共资源文件
|   |   |-- css
|   |       |-- style.css #字体图标样式文件
|   |       |-- fonts #字体图标文件
|   |-- components #组件文件
|   |-- pages #页面文件
|   |-- Home #主页

```

```

|   |   |   |-- Home.vue
|   |   |   |-- children
|   |   |       |-- CinemaManage.vue  #影院管理页
|   |   |       |-- CommentManage.vue  #评论管理页
|   |   |       |-- HallManage.vue  #影厅管理页
|   |   |       |-- MovieManage.vue  #电影管理页
|   |   |       |-- MovieSchedule.vue  #电影排片页
|   |   |       |-- OrderManage.vue  #订单管理页
|   |   |       |-- UserManage.vue  #用户管理页
|   |   |-- Login  #登录页
|   |       |-- Login.vue
|   |-- router  #路由
|       |-- index.js  #路由配置
|-- static  #静态文件
    |-- css
        |-- reset.css  #兼容性样式

```

数据库模块：

db.js:	数据库连接文件
静态资源文件	admin 管理员图片文件、avatar 用户图片文件、movie 电影海报图片文件
Routes	路由
Index.js	方法实现

B. 静态关系

界面模块和主题模块以文件的形式组织，vue 自成一个完整的页面或者界面模块的 HTML 文档、样式表以及脚本组成一个完整的页面。HTML 文档通过连接的方式将样式表和脚本包括进来。所有的界面模块部件组成整个界面模块。

Index.js 部件模块中列表条目和表单组件是分别所属与列表和表单类的。其他 function 大多数是一个一个相对独立的部件，经过组合组装形成一个完整功能的大部件。所有的接口构成 index.js 部件模块供界面部分调用组装。交互部件模块的所有类相互独立

C. 用途

需求阶段

1、可行性分析报告

说明该软件开发项目的实现在技术上、经济上和社会因素上的可行性，评述为了合理地达到开发目标可供选择的各种可能实施方案，说明并论证所选定实施方案的理由。

2、项目开发计划

为软件项目实施方案制订出具体计划，应该包括各部分工作的负责人员、开发的进度、开发经费的预算、所需的硬件及软件资源等。

3、软件需求说明书（软件规格说明书）

对所开发软件的功能、性能、用户界面及运行环境等作出详细的说明。它是在用户与开发人员双方对软件需求取得共同理解并达成协议的条件下编写的，也是实施开发工作的基础。该说明书应给出数据逻辑和数据采集的各项要求，为生成和维护系统数据文件做好准备。

设计阶段

4、概要设计说明书

该说明书是概要实际阶段的工作成果，它应说明功能分配、模块划分、程序的总体结构、输入输出以及接口设计、运行设计、数据结构设计和出错处理设计等，为详细设计提供基础。

5、详细设计说明书

着重描述每一模块是怎样实现的，包括实现算法、逻辑流程等。

开发阶段

6、开发进度月报

该月报系软件人员按月向管理部门提交的项目进展情况报告，报告应包括进度计划与实际执行情况的比较、阶段成果、遇到的问题 and 解决的办法以及下个月的打算等。

测试阶段

7、测试计划

为做好集成测试和验收测试，需为如何组织测试制订实施计划。计划应包括测试的内容、进度、条件、人员、测试用例的选取原则、测试结果允许的偏差范围等。

8、测试分析报告

测试工作完成以后，应提交测试计划执行情况的说明，对测试结果加以分析，并提出测试的结论意见。

收尾阶段

9、用户操作手册

本手册详细描述软件的功能、性能和用户界面，使用户对如何使用该软件得到具体的了解，为操作人员提供该软件各种运行情况的有关知识，特别是操作方法的具体细节。

10、项目开发总结报告

软件项目开发完成以后，应与项目实施计划对照，总结实际执行的情况，如进度、成果、资源利用、成本和投入的人力，此外，还需对开发工作做出评价，

总结出经验和教训。

11、软件维护手册

主要包括软件系统说明、程序模块说明、操作环境、支持软件的说明、维护过程的说明，便于软件的维护。

维护阶段

12、软件问题报告

指出软件问题的登记情况，如日期、发现人、状态、问题所属模块等，为软件修改提供准备文档。

13、软件修改报告

软件产品投入运行以后，发现了需对其进行修正、更改等问题，应将存在的问题、修改的考虑以及修改的影响作出详细的描述，提交审批。

D. 开发状态

界面主题为新开发类型。交互部件模块部分为重用已有设计包括 AJAX 前后端交互类 ajax，工具类 util。后台数据操作部分功能分发以及 sql 语句执行为新开发，其他部分为重用已有设计。

E. 使用资源

Vue.js 移动端组件库 *mint-ui* 等

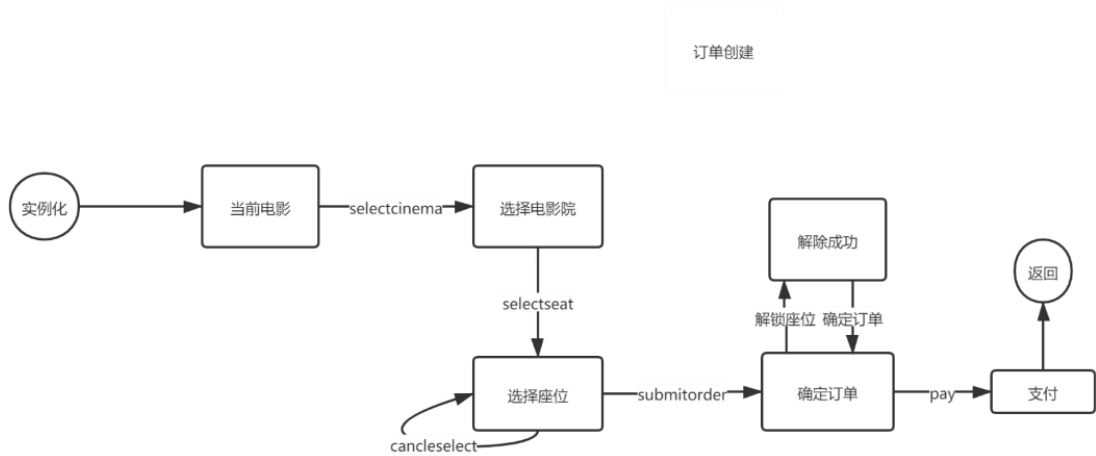
F. 程序库放置

前端后端的程序分开放置。

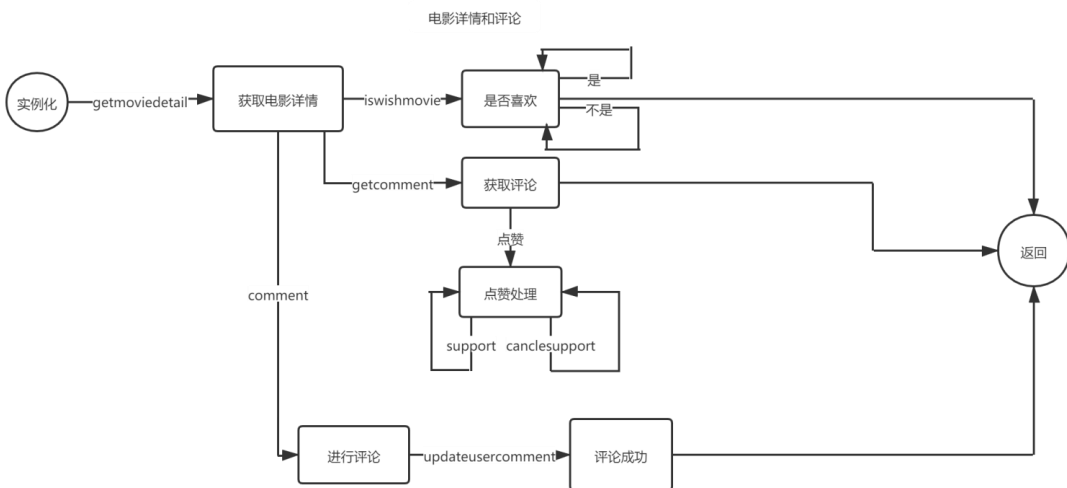
5.4 执行概念

状态转换图

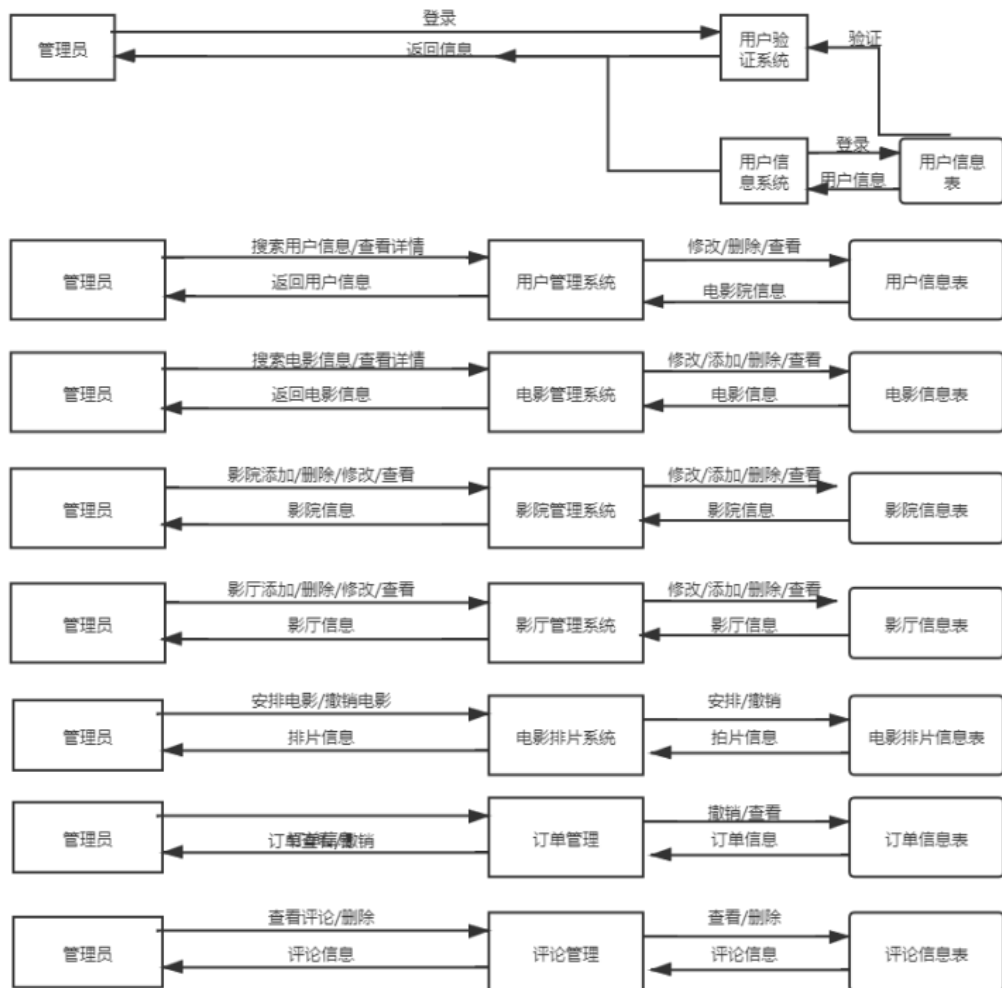
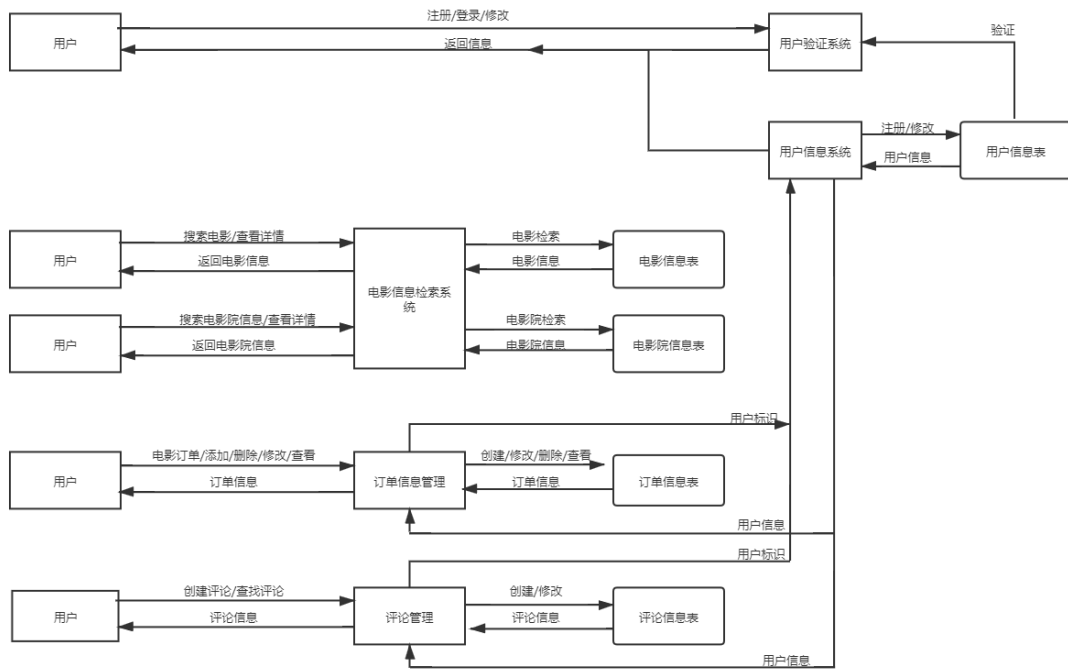
订单创建



电影详情和评论



数据流图



6. 后台数据表

```
-- MySQL dump 10.13 Distrib 8.0.23, for Win64 (x86_64)
--
-- Host: localhost  Database: db_film
-----
-- Server version  8.0.23
--
-- Table structure for table `t_admin`
--

DROP TABLE IF EXISTS `t_admin`;
CREATE TABLE `t_admin` (
  `admin_id` int unsigned NOT NULL AUTO_INCREMENT COMMENT '管理员 id',
  `name` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '管理员姓名',
  `password` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '管理员密码',
  `sex` varchar(10) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '管理员性别',
  `phone` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '管理员联系电话',
  `birthday` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '出生年月日',
  `avatar` varchar(255) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '管理员头像',
  PRIMARY KEY (`admin_id`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci ROW_FORMAT=DYNAMIC;
--
-- Table structure for table `t_cinema`
--

DROP TABLE IF EXISTS `t_cinema`;
CREATE TABLE `t_cinema` (
  `cinema_id` int unsigned NOT NULL AUTO_INCREMENT COMMENT '影院 id',
  `cinema_name` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '影院名称',
  `cinema_phone` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '影院电话',
  `province` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '影院所在省份',
```

```

`city` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'影院所在市',
`county` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMEN
T '影院所在区县',
`specified_address` varchar(255) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAUL
T NULL COMMENT '影院详细地址',
PRIMARY KEY (`cinema_id`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci ROW
_FORMAT=DYNAMIC;

```

```

--
-- Table structure for table `t_comment`
--

```

```

DROP TABLE IF EXISTS `t_comment`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
CREATE TABLE `t_comment` (
  `comment_id` int unsigned NOT NULL AUTO_INCREMENT COMMENT '评论 id',
  `user_id` int unsigned DEFAULT NULL COMMENT '用户 id',
  `movie_id` int unsigned DEFAULT NULL COMMENT '电影 id',
  `user_score` int unsigned DEFAULT NULL COMMENT '用户评分',
  `comment_content` text CHARACTER SET utf8 COLLATE utf8_unicode_ci COMMENT '用户评
论内容',
  `comment_date` datetime(6) DEFAULT NULL COMMENT '评论日期',
  `support_num` int unsigned DEFAULT NULL COMMENT '点赞数',
  `is_pass` int DEFAULT '0' COMMENT '评论是否通过审核',
  `support_user` text CHARACTER SET utf8 COLLATE utf8_unicode_ci COMMENT '点赞用户数
组',
  PRIMARY KEY (`comment_id`) USING BTREE,
  KEY `t_comment_ibfk_1` (`user_id`) USING BTREE,
  KEY `t_comment_ibfk_2` (`movie_id`) USING BTREE,
  CONSTRAINT `t_comment_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `t_user` (`user_i
d`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `t_comment_ibfk_2` FOREIGN KEY (`movie_id`) REFERENCES `t_movie` (`movi
e_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=59 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci ROW
_FORMAT=DYNAMIC;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Table structure for table `t_hall`
--

```

```

DROP TABLE IF EXISTS `t_hall`;

```

```

CREATE TABLE `t_hall` (
  `hall_id` int unsigned NOT NULL AUTO_INCREMENT COMMENT '影厅 id',
  `cinema_id` int unsigned DEFAULT NULL COMMENT '影院 id',
  `name` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'影厅名称',
  PRIMARY KEY (`hall_id`) USING BTREE,
  KEY `t_hall_ibfk_1` (`cinema_id`) USING BTREE,
  CONSTRAINT `t_hall_ibfk_1` FOREIGN KEY (`cinema_id`) REFERENCES `t_cinema` (`cinema_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=41 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci ROW
_FORMAT=DYNAMIC;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `t_movie`
--

DROP TABLE IF EXISTS `t_movie`;
CREATE TABLE `t_movie` (
  `movie_id` int unsigned NOT NULL AUTO_INCREMENT COMMENT '电影 id',
  `name` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'电影名称',
  `poster` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'电影海报',
  `director` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'导演',
  `actor` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'主演',
  `movie_long` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'电影片长',
  `language` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'电影语言版本',
  `intro` text CHARACTER SET utf8 COLLATE utf8_unicode_ci COMMENT '电影简介',
  `type` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'电影类型',
  `public_date` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'电影上映时间',
  `wish_num` int unsigned DEFAULT '0' COMMENT '想看人数',
  `score` decimal(4,1) DEFAULT NULL COMMENT '电影综合评分',
  PRIMARY KEY (`movie_id`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=39 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci ROW
_FORMAT=DYNAMIC;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Table structure for table `t_order`
--

DROP TABLE IF EXISTS `t_order`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `t_order` (
  `order_id` int unsigned NOT NULL AUTO_INCREMENT COMMENT '订单 id',
  `user_id` int unsigned DEFAULT NULL COMMENT '用户 id',
  `schedule_id` int unsigned DEFAULT NULL COMMENT '影院 id',
  `order_phone` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '下单手机',
  `order_date` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '用户下单时间',
  `ticket_num` int unsigned DEFAULT NULL COMMENT '购买电影票数',
  `ticket_total_price` decimal(10,2) unsigned DEFAULT NULL COMMENT '电影票单价',
  `order_seat_info` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '座位信息',
  `pay_type` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '支付方式',
  `phone_code` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '手机取票码',
  PRIMARY KEY (`order_id`) USING BTREE,
  KEY `t_order_ibfk_1` (`user_id`) USING BTREE,
  KEY `t_order_ibfk_2` (`schedule_id`) USING BTREE,
  CONSTRAINT `t_order_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `t_user` (`user_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `t_order_ibfk_2` FOREIGN KEY (`schedule_id`) REFERENCES `t_schedule` (`schedule_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=102 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci ROW_FORMAT=DYNAMIC;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Table structure for table `t_schedule`
--

```

```

DROP TABLE IF EXISTS `t_schedule`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `t_schedule` (
  `schedule_id` int unsigned NOT NULL AUTO_INCREMENT COMMENT '排片 id',
  `movie_id` int unsigned DEFAULT NULL COMMENT '电影 id',

```

```

`cinema_id` int unsigned DEFAULT NULL COMMENT '影院 id',
`hall_name` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COM
MENT '影厅名字',
`show_date` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COM
MENT '放映日期',
`show_time` varchar(10) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COM
MENT '放映时间',
`price` decimal(10,2) DEFAULT NULL COMMENT '电影售价',
`seat_info` text CHARACTER SET utf8 COLLATE utf8_unicode_ci COMMENT '座位信息',
PRIMARY KEY (`schedule_id`) USING BTREE,
KEY `hall_id` (`hall_name`) USING BTREE,
KEY `t_schedule_ibfk_1` (`movie_id`) USING BTREE,
KEY `t_schedule_ibfk_2` (`cinema_id`) USING BTREE,
CONSTRAINT `t_schedule_ibfk_1` FOREIGN KEY (`movie_id`) REFERENCES `t_movie` (`mov
ie_id`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `t_schedule_ibfk_2` FOREIGN KEY (`cinema_id`) REFERENCES `t_cinema` (`c
inema_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=102 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci RO
W_FORMAT=DYNAMIC;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `t_user`
--

DROP TABLE IF EXISTS `t_user`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `t_user` (
  `user_id` int unsigned NOT NULL AUTO_INCREMENT COMMENT '用户 id',
  `user_name` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COM
MENT '用户名',
  `avatar` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMME
NT '用户头像',
  `password` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COM
MENT '用户密码',
  `phone` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'用户手机号码',
  `sex` varchar(10) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT '
用户性别',
  `birthday` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMM
ENT '用户出生年月日',
  `sign` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'用户签名',

```

```

PRIMARY KEY (`user_id`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=55 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci ROW
_FORMAT=DYNAMIC;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `t_wishmovie`
--

DROP TABLE IF EXISTS `t_wishmovie`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `t_wishmovie` (
  `wishMovie_id` int unsigned NOT NULL AUTO_INCREMENT COMMENT '想看电影 id',
  `user_id` int unsigned DEFAULT NULL COMMENT '用户 id',
  `movie_id` int unsigned DEFAULT NULL COMMENT '电影 id',
  PRIMARY KEY (`wishMovie_id`) USING BTREE,
  KEY `t_wishmovie_ibfk_1` (`user_id`) USING BTREE,
  KEY `t_wishmovie_ibfk_2` (`movie_id`) USING BTREE,
  CONSTRAINT `t_wishmovie_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `t_user` (`user
_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `t_wishmovie_ibfk_2` FOREIGN KEY (`movie_id`) REFERENCES `t_movie` (`mo
vie_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=119 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci RO
W_FORMAT=DYNAMIC;
-- Dump completed on 2021-05-26 19:37:39

```

7. 尚未解决的问题

- 真正意义上的支付
- 验证等子系统

8. 注解

B/S 结构

B/S 结构 (Browser/Server, 浏览器/服务器模式), 是 WEB 兴起后的一种网络结构模式, WEB 浏览器是客户端最主要的应用软件。这种模式统一了客户端, 将系

统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器，如 Netscape Navigator 或 Internet Explorer，服务器安装 SQL Server、Oracle、MYSQL 等数据库。浏览器通过 Web Server 同数据库进行数据交互。

MySQL

MySQL 是一个精巧的 SQL 数据库管理系统，而且是开源的数据管理系统，是一个真正的多用户、多线程 SQL 数据库服务器。MySQL 主要目标是快速、健壮和易用。

Apache: Apache 是世界使用排名第一的 Web 服务器软件。它可以运行在几乎所有广泛使用的计算机平台上，由于其跨平台和安全性被广泛使用，是最流行的 Web 服务器端软件之一。它快速、可靠并且可通过简单的 API 扩充，将 Perl/Python 等解释器编译到服务器中。同时 Apache 音译为阿帕奇，是北美印第安人的一个部落，叫阿帕奇族，在美国的西南部。也是一个基金会的名称、一种武装直升机等等。

Node. js

Node. js 发布于 2009 年 5 月，由 Ryan Dahl 开发，是一个基于 Chrome V8 引擎的 JavaScript 运行环境，使用了一个事件驱动、非阻塞式 I/O 模型，让 JavaScript 运行在服务端的开发平台，它让 JavaScript 成为与 PHP、Python、Perl、Ruby 等服务端语言平起平坐的脚本语言。

Vue

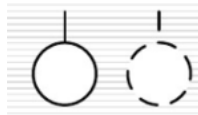
Vue 是一套用于构建用户界面的渐进式 JavaScript 框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用（SPA）提供驱动。

9、故障树分析

本项目故障树所用基本符号说明如下：

故障树事件符号说明：

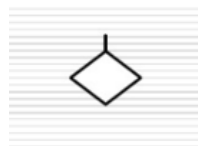
底事件：元、部件在设计的运行条件下发生的随机故障事件，故障分布已知。



• 实线○：硬件故障

• 虚线○：人为故障

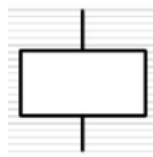
未探明事件：表面该事件可能发生，但是概率较小，可以勿需再进一步分析的故障事件。



顶事件：人们不希望发生的显著影响系统技术性能、经济性、可靠性和安全性的故障事件。



中间事件：包括故障树中除底事件及顶事件之外的所有事件。

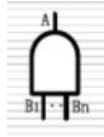


开关事件：描述逻辑门起作用的具体限制的特殊事件。

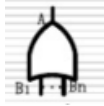


故障树逻辑门符号

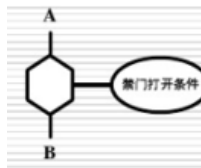
与门：当输入事件同时发生时，输出事件必然发生，这种逻辑关系称为事件交。



或门:当输入事件至少有一个发生时,输出事件发生,这种逻辑关系称为事件并。

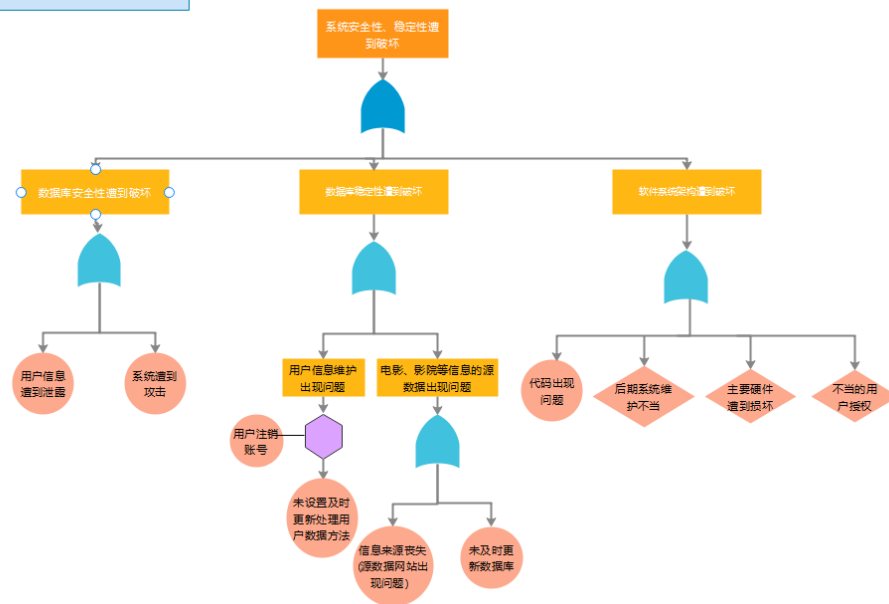


禁门: 仅当“禁门打开条件”发生时,输入事件的发生才导致输出事件的发生。

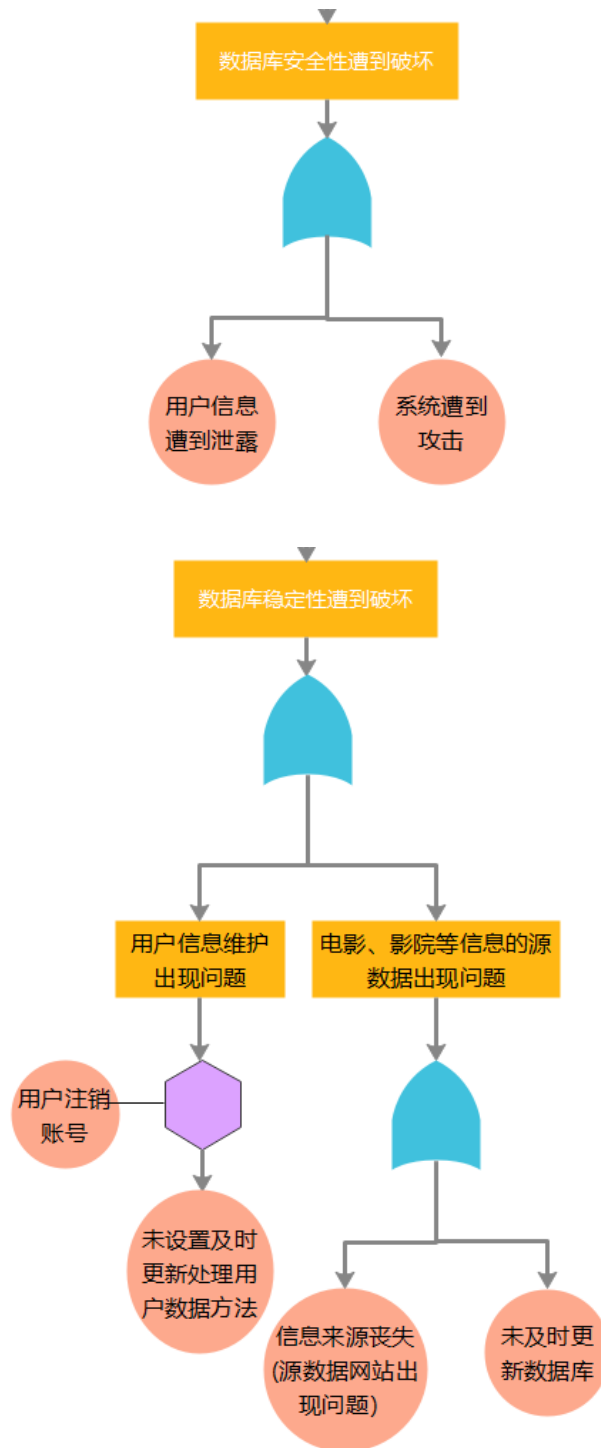


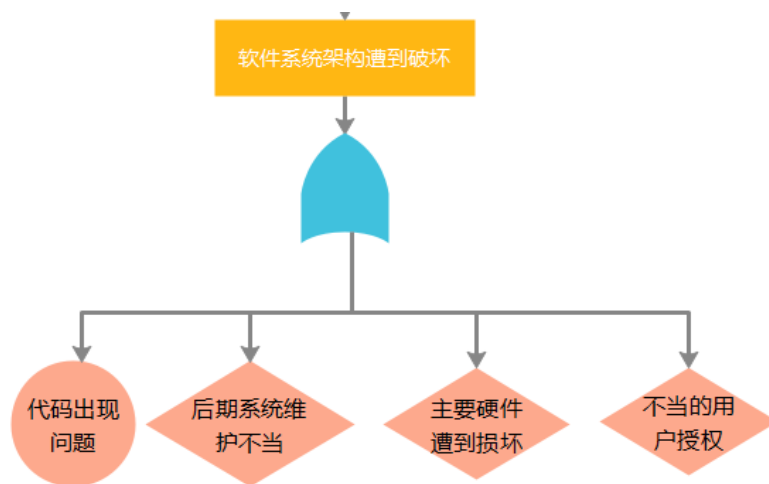
故障树

网络电影售票系统故障树模型

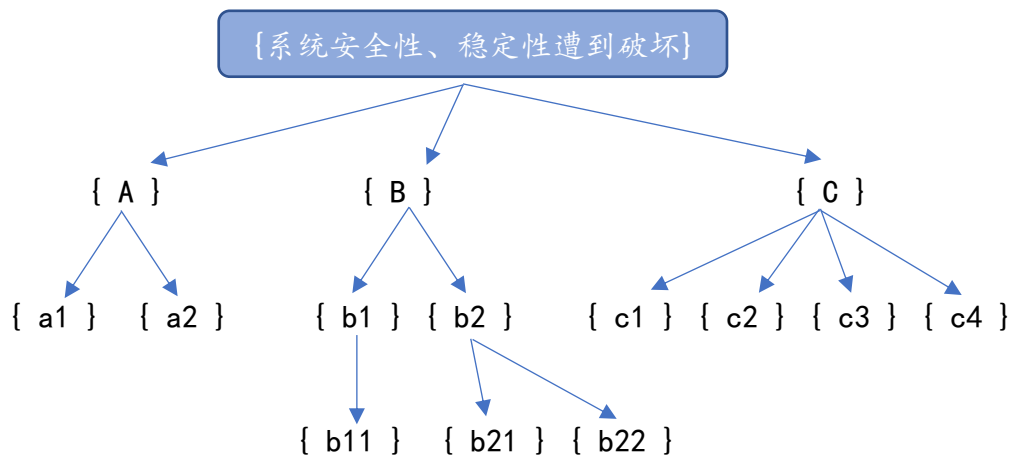


局部细节展示





相应的割集树



A : 数据库安全性遭到破坏

a1 : 用户信息遭到泄露

a2 : 系统遭到攻击

B : 数据库稳定性遭到破坏

b1 : 用户信息维护出现问题

b11 : 未设置及时更新处理用户数据方法

b2 : 电影、影院等信息的源数据出现问题

b21 : 信息来源丧失 (源数据网站出现问题)

b22 : 未及时更新数据库

C : 软件系统架构遭到破坏

c1 : 代码出现问题

c2 : 后期系统维护不当

c3 : 主要硬件遭到破坏

c4 : 不当的用户授权

10、保证项目质量属性所采取的策略

软件质量属性包括：功能性、性能、可用性、可靠性、健壮性、安全性、可修改性、可变性、易用性、可测试性、可复用性等…

六种质量属性策略：

1. 可用性

错误检测：命令/响应，异常监控

错误恢复：表决（裁决表），主动冗余，被动冗余，备件，状态再同步，检查点/回滚

错误预防：从服务中删除，事物（要么全成功，要么全失败），定期重置，进程监视器

2. 可修改性

局部化修改：维持语义的一致性，预期期望的变更，泛化该模块，限制可能的选择

防止连锁反应：信息隐藏，维持现有的接口，限制通信路径，仲裁者的使用

推迟绑定时间：运行时注册，配置文件，多态，构件更换

3. 性能

资源需求：减少处理时间所需的资源，减少所处理时间的数量，控制资源使用，限制执行时间

资源管理：引入并发，维持数据或计算的多个副本，增加可用资源

资源仲裁：先进/先出，固定优先级，动态优先级调度，静态调度

4. 安全性

抵抗攻击：对用户进行身份验证，对用户进行授权，维护数据的机密性，维护完整性，限制暴露的信息，限制访问

检测攻击：部署入侵检测系统

从攻击中恢复：恢复，识别攻击者

5. 可测试性

输入/输出：记录/回放，将接口-实现分离，优化访问路线/接口

内部监控：当监视器处于激活状态时，记录事件

6. 易用性

运行时：任务模型，用户模型，系统模型

设计时：将用户接口与应用的其余部分分离

支持用户主动：支持用户主动操作

我们的项目采取了：

可修改性：所有功能实现基本都是使用 sql 语句从数据库中调取信息，所以维持了语义的一致性，泛化了该模块，限制可能的选择。

性能：采用 express 中的 Router 作用就是为了方便我们更好的根据路由去分模块。避免将所有路由都写在入口文件中，在很大程度上提升了系统性能

安全性：采用用户登录模式对用户进行身份验证，管理员身份对用户进行授权，维护数据的机密性，维护完整性，限制暴露的信息，限制访问。

可测试性：mysql 数据库内置 Log 会记录数据查看和修改情况，可实现记录/回放功能。

易用性：前后端分离模型，将用户接口与应用的其余部分分离。

11、附录

附录 A：个人学习报告

软件体系结构学习报告：

“软件体系结构包括有关软件系统组织的一系列重要决策，包括对构成系统的结构要素及其接口的选择；这些要素之间协作规定的行为；将这些结构和行为要素组成更大的子系统；以及指导该组织的建筑风格。软件体系结构还涉及功能，可用性，弹性，性能，重用性，可理解性，经济和技术约束，折衷和美学方面的考虑。”

学习报告 朱可欣：

软件体系结构的发展方向

● 各种软件体系结构语言之间的信息互换

1. 现有的各种软件体系结构描述语言大多是与领域相关的，所以不利于对不同领域体系结构的说明。但这些针对不同领域的软件体系结构描述语言在某些方面又大同小异，造成资源的冗余。
2. 其实，大多数软件体系结构描述语言具有一系列共同概念。
3. 如何用一种公共形式把各种语言综合起来，使得能够交换各种体系结构描述信息，将是今后软件体系结构研究和实践的重点之一。

● 设计工具和环境

1. 软件体系结构设计作为软件工程的一部分，它的计算机辅助实现手段是相当重要的。
2. 应当开发出一些软件工具来实现体系结构的描述和分析，开发阶段转换工具可以实现阶段成果的自动转换，例如，把需求规格说明自动转换为构件等。
3. 目前关于这方面的研究成果很少，特别是可以应用到实际项目开发中的工具和环境就更少。

● 体系结构再工程问题

1. 现在软件系统的规模变得越来越大，结构也越来越复杂，同时从头开始构建的大系统数量在急剧地减少，因而很多遗留系统正在被逐步地利用。
2. 从遗留系统软件代码和系统中抽取结构信息，经过描述、统一、抽象、一般化与实例化等处理，可总结出系统的体系结构。

软件体系结构的风格与模式

软件体系结构风格和模式的概念

● 软件体系结构风格 Architectural Style

- 一种体系结构风格以结构组织模式定义了一个系统家族
- 关于构件和连接件类型的术语；一组约束对它们组合方式的规定；一个或多个语义模型，规定了如何从各成分的特性决定系统整体特性
- 概括地说，一种软件体系结构风格刻画一个具有共享结构和语义的系统家族

● 软件体系结构模式 Architectural Pattern

- 一种软件体系结构模式是对某具体环境下问题的结构性解决方法

● 软件体系结构的构建风格

体系结构风格不是对软件进行分类的标准它仅仅是表示描述软件的不同角度而已例如，一个系统采用了分层风格，但这并不妨碍它用面向对象的方法来实现。同一个系统采用多种风格造成了所谓体系结构风格的异构组合。

• 管道-过滤器风格

在管道-过滤器风格下，每个功能模块都有一组输入和输出功能模块称作过滤器（filters）功能模块间的连接可以看作输入、输出数据流之间的通路，所以称作管道（pipes）。

- 。 管道-过滤器风格的特性之一在于过滤器的相对独立性，即过滤器独立完成自身功能，相互之间无需进行状态交互。

过滤器是独立运行的构件

- 。 非临近的过滤器之间不共享状态

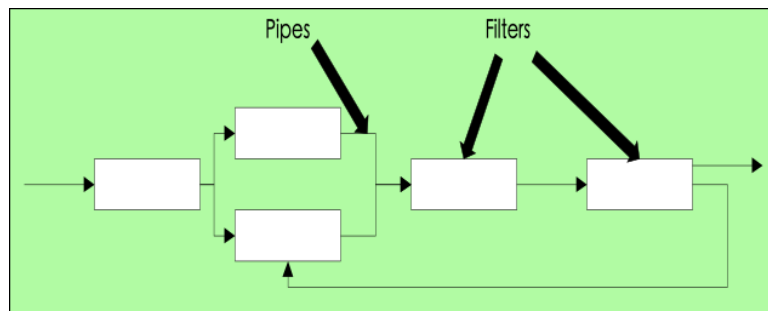
过滤器自身无状态

- 。 过滤器对其处理上下连接的过滤器“无知”

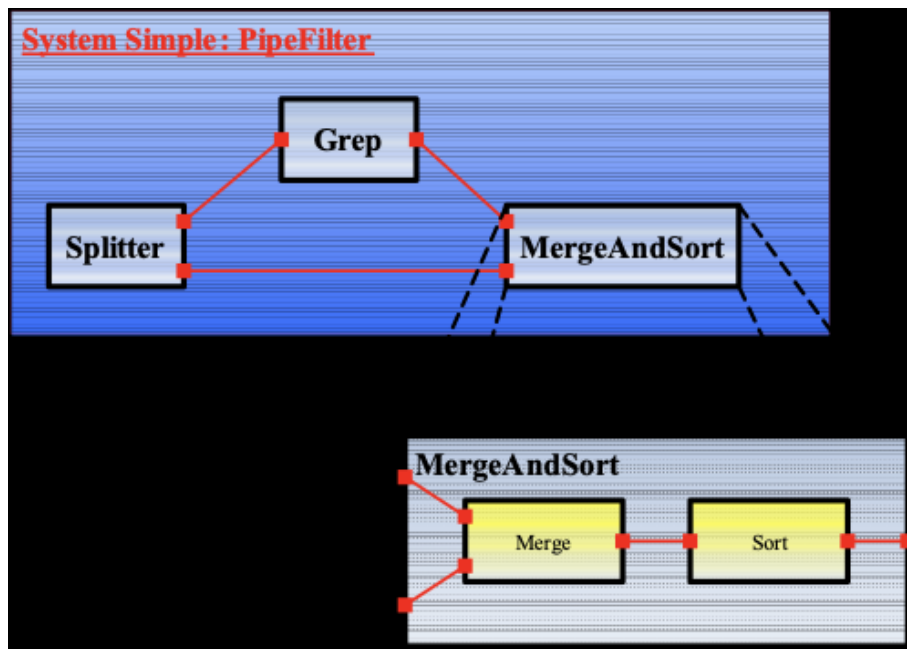
对相邻的过滤器不施加任何限制

- 。 结果的正确性不依赖于各个过滤器运行的先后次序
- 。 各过滤器在输入具备后完成自己的计算。
- 。 完整的计算过程包含在过滤器之间的拓扑结构中。

- 。 一个管道-过滤器风格的图示



- 。 一个采用了嵌套的管道过滤器的系统示例



○ 实例

Unix 系统中的管道过滤器结构

```
ls -al | grep my
```

DOS 中的管道命令：

DOS 允许在命令中出现用竖线字符“|”分开的多个命令，将符号“|”之前的命令的输出，作为“|”之后命令的输入，这就是“管道功能”，竖线字符“|”是管道操作符。

例如命令 `dir | more` 使得当前目录列表在屏幕上逐屏显示。`dir` 的输出是整个目录列表，它不出现在屏幕上而是由于符号“|”的规定，成为下一个命令 `more` 的输入，`more` 命令则将其输入，`more` 命令则将其输入一屏一屏地显示，成为命令行的输出。

○ 优点

设计者可以将整个系统的输入、输出特性简单的理解为各个过滤器功能的合成

- 设计人员将整个系统的输入输出行为理解为单个过滤器行为的叠加与组合。

- 这样可以将问题分解，化繁为简。
- 将系统抽象成一个“黑箱”，其输入是系统中第一个过滤器的输入管道，输出是系统中最后一个过滤器的输出管道，而其内部各功能模块的具体实现对用户完全透明。

支持功能模块的复用

- 任何两个过滤器，只要它们之间传送的数据遵守共同的规约，就可以相连接。
- 每个过滤器都有自己独立的输入输出接口，如果过滤器间传输的数据遵守其规约，只要用管道将它们连接就可以正常工作。

系统具有较强的可维护性和可扩展性

- 旧的过滤器可以被替代，新的过滤器可以添加到已有的系统上。
- 软件的易于维护和升级是衡量软件系统质量的重要指标之一，在管道-过滤器模型中，只要遵守输入输出数据规约，任何一个过滤器都可以被另一个新的过滤器代替，同时为增强程序功能，可以添加新的过滤器。
- 这样，系统的可维护性和可升级性得到了保证。

支持一些特定的分析

- 如吞吐量计算和死锁检测等
- 利用管道-过滤器风格的视图，可以很容易的得到系统的资源使用和请求的状态图。
- 然后，根据操作系统原理等相关理论中的死锁检测方法就可以分析出系统目前所处的状态，是否存在死锁可能及如何消除死锁等问题。

具有并发性

- 每个过滤器作为一个单独的执行任务，可以与其它过滤器并发执行。
- 过滤器的执行是独立的，不依赖于其它过滤器的。在实际运行时，可以将存在并发可能的多个过滤器看作多个并发的任务并行执行，从而大大提高系统的整体效率，加快处理速度。

○ 缺点

交互式处理能力弱

- 管道-过滤器模型适于数据流的处理和变换，不适合为与用户交互频繁的系统建模。
- 在这种模型中，每个过滤器都有自己的数据，这些数据或者是从磁盘存储器中读取来，或者是由另一个过滤器的输出导入进来，整个系统没有一个共享的数据区。
- 这样，当用户要操作某一项数据时，要涉及到多个过滤器对相应数据的操作，其实现较为复杂。由以上的缺点，可以对每个过滤器增加相应的用户控制接口，使得外部可以对过滤器的执行进行控制。

导致系统处理过程的成批操作。

- 设计者也许不得不花费精力协调两个相对独立但又存在某种关系的数据流之间的关系
例如多过滤器并发执行时数据流之间的同步问题等。
- 根据实际设计的需要，设计者也需要对数据传输进行特定的处理（如为了防止数据泄漏而采取加密等手段），导致过滤器必须对输

入、输出管道中的数据流进行解析或反解析，增加了过滤器具体实现的复杂性。

- 面向对象风格

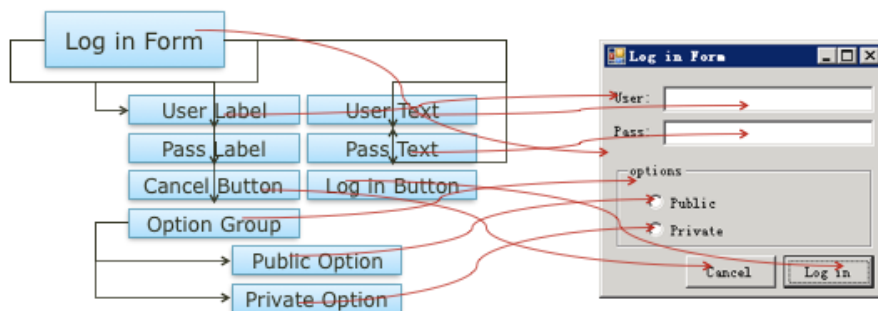
- 概述

面相对象模式集数据抽象、抽象数据类型、类继承为一体，使软件工程公认的模块化、信息隐藏、抽象、重用性等原则在面向对象风格下得以充分实现。

- 应用场合

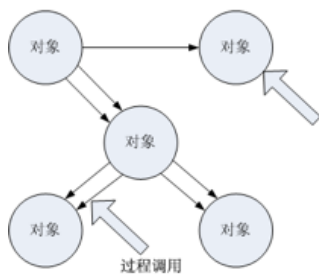
面向对象的体系结构模式适用于数据和功能分离的系统中，同样也适合于问题域模型比较明显，或需要人机交互界面的系统。大多数应用事件驱动风格的系统也常常应用了面向对象风格

图示

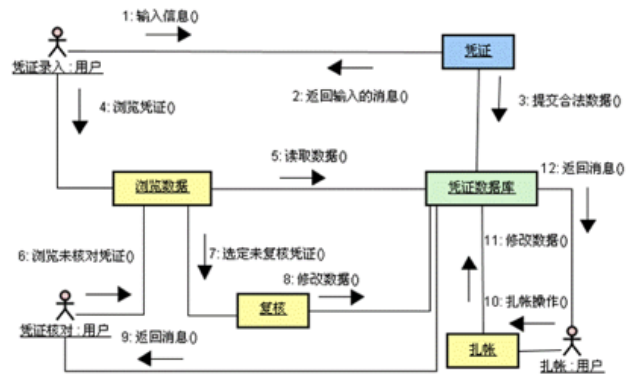


- 基本原则

- 将逻辑上的实体映射为对象，实体之间的关系映射为对象之间的应用关系。
 - 对象利用应用关系来访问对方公开的接口，完成某个特定任务；一组对象之间相互协作，完成总体目标。
 - 图示



面向对象风格的抽象描述



一个银行系统对象之间的协作实例

。 优点

高度模块性

数据与其相关操作被组织为对象， 成为模块组织的基本单位

封装功能

一组功能和其实现细节被封装在一个对象中，具有功能的接口被暴露出来

代码共享

对象的相对独立性可被反复重用，通过拼装形成不同的软件系统

灵活性

对象在组织过程中，相互关系可以任意变化，只要接口兼容

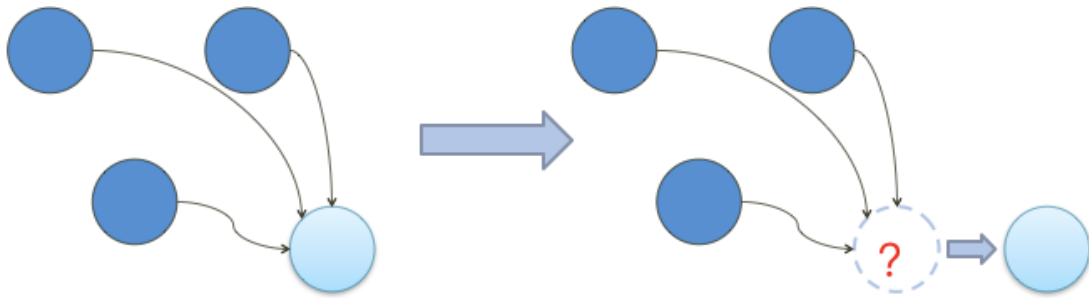
易维护性

对象接近于人对问题和解决方案模型的思维方式，易于理解和修改

。 缺点

- 。 面向对象风格最大的不足在于如果一个对象需要调用另一个对象，它就必须知道那个对象的标识（对象名或对象引用），这样就无形之中增强了对对象之间的依赖关系。
- 。 如果一个对象改变了自己的标识，就必须通知系统中所有和它有调用关系的对象，否则系统就无法正常运行。

。 图示



软件体系结构学习报告_张天祥:

软件体系结构风格:

一种体系结构风格以结构组织模式定义了一个系统家族。

关于构件和连接件类型的术语;一组约束对它们组合方式的规定;一个或多个语义模型,规定了如何从各成分的特性决定系统整体特性

概括地说,一种软件体系结构风格刻画一个具有共享结构和语义的系统家族.

架构风格:

客户端-服务器:

将系统分为两个应用,其中客户端向服务器发送服务请求。

基于组件的架构:

把应用设计分解为可重用的功能、逻辑组件,这些组件的位置相互透明,只暴露明确定义的通信接口。

分层架构:

把应用的关注点分割为堆栈组(层)。

消息总线:

指接收、发送消息的软件系统,消息基于一组已知格式,以便系统无需知道实际接收者就能互相通信。

N层/三层架构:

用与分层风格差不多一样的方式将功能划分为独立的部分,每个部分是一个层,处于完全独立的计算机上。

面向对象:

该架构风格是将应用或系统任务分割成单独、可重用、可自给的对象,每个对象包含数据,以及与对象相关的行为。

分离表现层:

将处理用户界面的逻辑从用户界面(UI)视图和用户操作的数据中分离出来。

面向服务架构(SOA):

是指那些利用契约和消息将功能暴露为服务、消费功能服务的应用。

这些架构风格分别适用于特定领域:

通信:

SOA，消息总线，管道和过滤器

部署：

客户端/服务器，三层架构，N 层架构

领域：

领域模型，网关

交互：

分离表现层

结构：

基于组件的架构，面向对象，分层架构

下面介绍几种常见的架构风格：

管道和过滤器风格：

在管道/过滤器风格的软件体系结构中，每个构件都有一组输入和输出，构件读输入的数据流，经过内部处理，然后产生输出数据流。这个过程通常通过对输入流的变换及增量计算来完成，所以在输入被完全消费之前，输出便产生了。因此，这里的构件被称为过滤器，这种风格的连接件就像是数据流传输的管道，将一个过滤器的输出传到另一过滤器的输入。此风格特别重要的 过滤器必须是独立的实体，它不能与其它过滤器共享数据，而且一个过滤器不知道它上游和下游的标识。一个管道/过滤器网络输出的正确性并不依赖于过滤器进行增量计算过程的顺序。

图 1 是管道/过滤器风格的示意图。一个典型的管道/过滤器体系结构的例子是以 Unix shell 编写的程序。Unix 既提供一种符号，以连接各组成部分 (Unix 的进程)，又提供某种进程运行时机制以实现管道。另一个著名的例子是传统的编译器。传统的编译器一直被认为是一种管道系统，在该系统中，一个阶段 (包括词法分析、语法分析、语义分析和代码生成) 的输出是另一个阶段的输入。

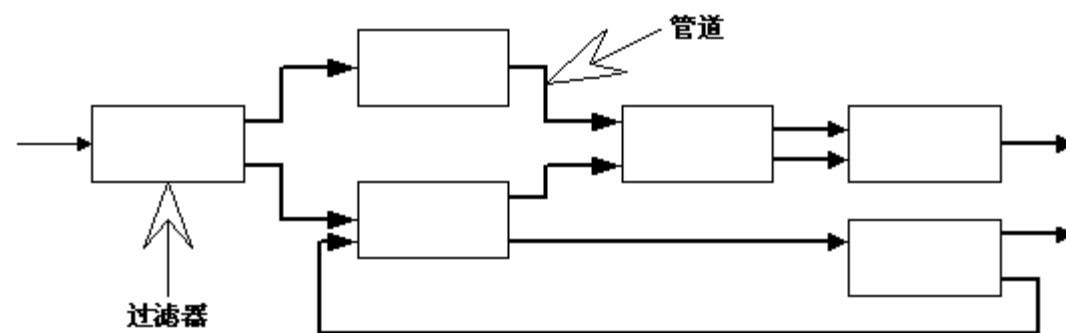


图 1 管道/过滤器风格的体系结构

管道/过滤器风格的软件体系结构具有许多很好的特点：

- (1) 使得软构件具有良好的隐蔽性和高内聚、低耦合的特点；
- (2) 允许设计者将整个系统的输入/输出行为看成是多个过滤器的行为的简单合成；
- (3) 支持软件重用。重要提供适合在两个过滤器之间传送的数据，任何两个过滤器都可被连接起来；
- (4) 系统维护和增强系统性能简单。新的过滤器可以添加到现有系统中来；旧的可以被改进的过滤器替换掉；
- (5) 允许对一些如吞吐量、死锁等属性的分析；

个事件被触发，系统自动调用在这个事件中注册的所有过程，这样，一个事件的触发就导致了另一模块中的过程的调用。

从体系结构上说，这种风格的构件是一些模块，这些模块既可以是一些过程，又可以是一些事件的集合。过程可以用通用的方式调用，也可以在系统事件中注册一些过程，当发生这些事件时，过程被调用。

基于事件的隐式调用风格的主要特点是事件的触发者并不知道哪些构件会被这些事件影响。这样不能假定构件的处理顺序，甚至不知道哪些过程会被调用，因此，许多隐式调用的系统也包含显式调用作为构件交互的补充形式。

支持基于事件的隐式调用的应用系统很多。例如，在编程环境中用于集成各种工具，在数据库管理系统中确保数据的一致性约束，在用户界面系统中管理数据，以及在编辑器中支持语法检查。例如在某系统中，编辑器和变量监视器可以登记相应 Debugger 的断点事件。当 Debugger 在断点处停下时，它声明该事件，由系统自动调用处理程序，如编辑程序可以卷屏到断点，变量监视器刷新变量数值。而 Debugger 本身只声明事件，并不关心哪些过程会启动，也不关心这些过程做什么处理。

隐式调用系统的主要优点有：

(1) 为软件重用提供了强大的支持。当需要将一个构件加入现存系统中时，只需将它注册到系统的事件中。

(2) 为改进系统带来了方便。当用一个构件代替另一个构件时，不会影响到其它构件的接口。

隐式调用系统的主要缺点有：

(1) 构件放弃了对系统计算的控制。一个构件触发一个事件时，不能确定其它构件是否会响应它。而且即使它知道事件注册了哪些构件的构成，它也不能保证这些过程被调用的顺序。

(2) 数据交换的问题。有时数据可被一个事件传递，但另一些情况下，基于事件的系统必须依靠一个共享的仓库进行交互。在这些情况下，全局性能和资源管理便成了问题。

(3) 既然过程的语义必须依赖于被触发事件的上下文约束，关于正确性的推理存在问题。

层次系统风格：

层次系统组织成一个层次结构，每一层为上层服务，并作为下层客户。在一些层次系统中，除了一些精心挑选的输出函数外，内部的层只对相邻的层可见。这样的系统中构件在一些层实现了虚拟机(在另一些层次系统中层是部分不透明的)。连接件通过决定层间如何交互的协议来定义，拓扑约束包括对相邻层间交互的约束。

这种风格支持基于可增加抽象层的设计。这样，允许将一个复杂问题分解成一个增量步骤序列的实现。由于每一层最多只影响两层，同时只要给相邻层提供相同的接口，允许每层用不同的方法实现，同样为软件重用提供了强大的支持。

图 3 是层次系统风格的示意图。层次系统最广泛的应用是分层通信协议。在这一应用领域中，每一层提供一个抽象的功能，作为上层通信的基础。较低的层次定义低层的交互，最低层通常只定义硬件物理连接。

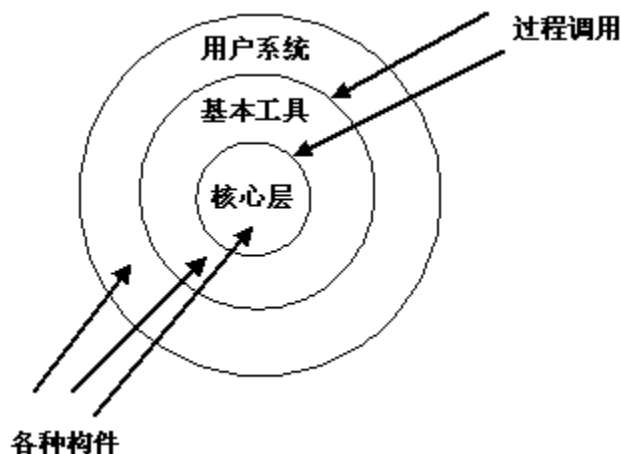


图 3 层次系统风格的体系结构

层次系统有许多可取的属性：

(1) 支持基于抽象程度递增的系统设计，使设计者可以把一个复杂系统按递增的步骤进行分解；

(2) 支持功能增强，因为每一层至多和相邻的上下层交互，因此功能的改变最多影响相邻的上下层；

(3) 支持重用。只要提供的服务接口定义不变，同一层的不同实现可以交换使用。这样，就可以定义一组标准的接口，而允许各种不同的实现方法。

但是，层次系统也有其不足之处：

(1) 并不是每个系统都可以很容易地划分为分层的模式，甚至即使一个系统的逻辑结构是层次化的，出于对系统性能的考虑，系统设计师不得不把一些低级或高级的功能综合起来；

(2) 很难找到一个合适的、正确的层次抽象方法。

仓库风格：

在仓库风格中，有两种不同的构件：中央数据结构说明当前状态，独立构件在中央数据存贮上执行，仓库与外构件间的相互作用在系统中会有大的变化。

控制原则的选取产生两个主要的子类。若输入流中某类时间触发进程执行的选择，则仓库是一传统型数据库；另一方面，若中央数据结构的当前状态触发进程执行的选择，则仓库是一黑板系统。

图 4 是黑板系统的组成。黑板系统的传统应用是信号处理领域，如语音和模式识别。另一应用是松耦合代理数据共享存取。

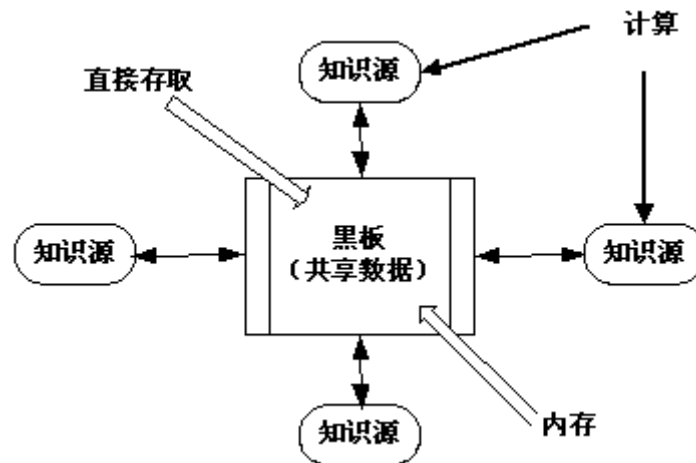


图 4 黑板系统的组成

我们从图 4 中可以看出，黑板系统主要由三部分组成：

- (1) 知识源。知识源中包含独立的、与应用程序相关的知识，知识源之间不直接进行通讯，它们之间的交互只通过黑板来完成。
- (2) 黑板数据结构。黑板数据是按照与应用程序相关的层次来组织的解决问题的数据，知识源通过不断地改变黑板数据来解决问题。
- (3) 控制。控制完全由黑板的状态驱动，黑板状态的改变决定使用的特定知识。

学习报告庞守任：

通过这学期的软件工程的课程的学习，我清楚地认知到了软件就是程序加软件工程，软件工程的基本含义。一个项目的好坏不仅仅决定于程序员的代码水平，还有很多的其他方面的因素。同时我了解到针对不同类型的大型项目有不同的模型。常见的体系架构模式有以下几种：

一. 分层模式

这种模式也称为多层体系架构模式。它可以用来构造可以分解为子任务组的程序，每个子任务都处于一个特定的抽象级别。每个层都为下一个提供更高层次服务。一般信息系统中最常见的是如下所列的 4 层：

- 表示层(也称为 UI 层)
- 应用层(也称为服务层)
- 业务逻辑层(也称为领域层)
- 数据访问层(也称为持久化层)

二. 客户端-服务器模式

这种模式由两部分组成：一个服务器和多个客户端。服务器组件将为多个客户端组件提供服务。客户端从服务器请求服务，服务器为这些客户端提供相关服务。此外，服务器持续侦听客户机请求。

三. 主从设备模式

这种模式由两方组成；主设备和从设备。主设备组件在相同的从设备组件中分配工作，并计算最终结果，这些结果是由从设备返回的结果。

四. 管道-过滤器模式

此模式可用于构造生成和处理数据流的系统。每个处理步骤都封装在一个过滤器组件内。要处理的数据是通过管道传递的。这些管道可以用于缓冲或用于同步。

五. 代理模式

此模式用于构造具有解耦组件的分布式系统。这些组件可以通过远程服务调用彼此交互。代理组件负责组件之间的通信协调。

服务器将其功能(服务和特征)发布给代理。客户端从代理请求服务,然后代理将客户端重定向到其注册中心的适当服务。

六. 点对点模式

在这种模式中,单个组件被称为对等点。对等点可以作为客户端,从其他对等点请求服务,作为服务器,为其他对等点提供服务。对等点可以充当客户端或服务或两者的角色,并且可以随时间动态地更改其角色。

七. 事件总线模式

这种模式主要是处理事件,包括4个主要组件:事件源、事件监听器、通道和事件总线。消息源将消息发布到事件总线上的特定通道上。侦听器订阅特定的通道。侦听器会被通知消息,这些消息被发布到它们之前订阅的一个通道上。

八. 模型-视图-控制器模式

这种模式,也称为MVC模式,把一个交互式应用程序划分为3个部分,

模型:包含核心功能和数据

视图:将信息显示给用户(可以定义多个视图)

控制器:处理用户输入的信息

这样做是为了将信息的内部表示与信息的呈现方式分离开来,并接受用户的请求。它分离了组件,并允许有效的代码重用。

九. 黑板模式

这种模式对于没有确定解决方案策略的问题是有用的。黑板模式由3个主要组成部分组成。

黑板——包含来自解决方案空间的对象的结构化全局内存

知识源——专门的模块和它们自己的表示

控制组件——选择、配置和执行模块

所有的组件都可以访问黑板。组件可以生成添加到黑板上的新数据对象。组件在黑板上查找特定类型的数据,并通过与现有知识源的模式匹配来查找这些数据。

十. 解释器模式

这个模式用于设计一个解释用专用语言编写的程序的组件。它主要指定如何评估程序的行数,即以特定的语言编写的句子或表达式。其基本思想是为每种语言的符号都有一个分类。

学习报告 王凯:

ATAM 体系结构权衡分析方法

一、 基本概念

ATAM: Architecture Tradeoff Analysis Method(构架权衡分析方法),它是评价 软件构架 的一种综合全面的方法。这种方法不仅可以揭示出构架满足特定质量目标的情况,而且(因为它认识到了构架决策会影响多个质量属性)可以使更清楚地认识到质量目标之间的联系——即如何权衡诸多质量目标。

二、参与成员

评估小组：小组是所评估构架的项目外部的小组。它通常由 3~5 个人组成。在评估期间，该小组的每个成员都要扮演大量的特定角色。评估小组可能是一个常设小组，其中要定期执行构架评估，其成员也可能是为了应对某次评估，从了解构架的人中挑选出来的。他们可能与开发小组（其构架是公开的）为相同的组织工作，也可能是外部的咨询人员。在任何情况下，他们都应该是有能力、没有偏见且私下都没有其他工作要做的外部人员。

项目决策者：这些人对开发项目具有发言权，并有权要求进行某些改变。他们通常包括项目管理人员，如果有一个承担开发费用的可以确认的客户，他（她）或其代表也应该列入其中。设计师肯定要参与评估——构架评估的一个基本准则是设计师必须愿意参与评估。最后，委托进行评估的人通常有权就开发项目发言，如果他（她）没有权利代表项目发言的话，他（她）也必须是小组的一个成员。

构架涉众：涉众在构架中有一个既得利益（正如所宣称的那样）。他们完成工作的能力与支持可修改性、安全性、高可靠性等特性的构架密切相关。涉众包括开发，测试，集成，维护，性能，构架工程师和用户、与正在分析的系统交互的系统的构建人员以及其他人员。在评估期间，他们的工作职责是清晰地阐述构架应该满足的具体质量属性目标，以使锁开发的系统能够取得成功。根据经验，应该有 12~15 个涉众参与评估。

三、输出

能在 1 小时内表述的构架描述

清楚的业务目标

用场景捕获的质量需求

敏感点、权衡点、有风险决策、无风险决策…

四、过程

0 阶段，合作关系与准备，确定细节：人员名单，时间，地点；评估小组获取资料并进行初步了解分析；

1 阶段，评估阶段，决策者参与，小组开始信息收集与分析；耗时约 1 周

1~2 周中断期，评估小组进一步以非正式方式了解构架；

2 阶段，评估阶段，涉众参与，分析继续；约 2 天；

3 阶段，后续阶段，生成最终报告，进行评估活动总结；1 周；

评估阶段的细分：

1) 评估负责人向决策者表述 ATAM 方法，使大家理解其过程，了解角色布局；

2) 决策者介绍系统商业动机、重要功能、各种限制、商业目标、驱动因素等；

3) 设计师介绍构架，技术限制、所用模式等；

4) 评估小组利用所有已知信息对构架方法进行分类；

5) 生成质量属性效用树，捕获详细的需求信息，为每个场景分配一个级别，如（高，中），前者为重要度，后者为实现难易度，重点放在（高，高）的场景；此处场景具备刺激、环境、响应三要素就可以了；

6) 评估小组分析所有重要场景，设计师解释如何支持该场景，检查所用构架方法，分析风险点、权衡点、敏感点

效用树：效用为根，性能等质量属性组成二级节点，继续细分直到场景，场景为叶经过一段中断期，第2阶段开始，此时涉众开始参与；首先仍然需要一个对ATAM方法的介绍，并使涉众了解已有的成果；

7) 集体讨论并分析场景的优先级，以了解更广泛的涉众的想法；该过程可能产生新的场景；使用“有限票数法”投票确定每个场景的优先级——此处不考虑实现难度了吧；

8) 分析新的高优先级的场景，构架师解释构架是怎么满足各场景的；

9) 总结评估结果，评估负责人展示该结果；

注意，看起来第1阶段与第2阶段活动类似，但它们的关注点与具体操作并不相同。

五、 限制

ATAM不是一个准确的方法，且风险不可被量化；

软件体系结构学习报告常滋厚：

通过参考文献以及网上搜索得到的内容，对软件体系结构有了比较深层次的了解，学习了有关软件体系结构的相关内容，软件体系结构为我们所使用的软件提供了结构和行为的抽象，不仅指定了系统的结构，也表明了系统的需求，以及系统元素之间的关系。我认为规范化，系统化的软件体系结构描述，对理解软件体系结构有很大的帮助作用，但这个过程不是很简单就能完成的，我们都必须经过非形式化的过程，在这个过程中不断积累知识，最后将这个非形式化的过程变得形式化，标准化。

通过短暂的学习了解到了软件体系结构的模型及其作用：

结构模型：用体系结构的构件，连接件来刻画结构，并试图通过结构来反映系统的重要语义内容，包括了系统的配置，约束等。

框架模型：侧重于描写的是整体结构，主要为目标建立只针对该目标的架构。

动态模型：对结构模型，框架模型的补充。

过程模型：顾名思义，构造了系统的过程和主要步骤。

功能模型：一种比较特殊的框架，认为整体架构是按照功能按层次分开的。

4+1 试图模型及其作用：

逻辑视图：主要支持系统的功能需求，即系统提供给最终用户的服务。在逻辑视图中，系统分解成一系列的功能抽象，这些抽象主要来自问题领域。这种分解不但可以用来进行功能分析，而且可用作标识在整个系统的各个不同部分的通用机制和设计元素。在面向对象技术中，通过抽象、封装和继承，可以用对象模型来代表逻辑视图，用类图来描述逻辑视图。

开发视图：也称模块视图，主要侧重于软件模块的组织和管理。开发视图要考虑软件内部的需求，如软件开发的容易性、软件的重用和软件的通用性，要充分考虑到由于具体开发工具的不同而带来的局限性。开发视图通过系统输入输出关系的模型图和子系统图来描述。

进程视图：侧重于系统的运行特性，主要关注一些非功能性的需求。进程视图强调并发性、分布性、系统集成性和容错能力，以及从逻辑视图中的主要抽象如何适合进程结构。它也定义逻辑视图中的各个类的操作具体是在哪一个线程中被执行的。进程视图可以描述成多层抽象，每个级别分别关注不同的方面。在最高层抽象中，进程结构可以看作是构成一个执行单元的一组任务。它可看成一系

列独立的，通过逻辑网络相互通信的程序。它们是分布的，通过总线或局域网、广域网等硬件资源连接起来。

物理视图：主要考虑如何把软件映射到硬件上，它通常要考虑到系统性能、规模、可靠性等。解决系统拓扑结构、系统安装、通讯等问题。当软件运行于不同的节点上时，各视图中的构件都直接或间接地对应于系统的不同节点上。因此，从软件到节点的映射要有较高的灵活性，当环境改变时，对系统其他视图的影响最小。

场景视图：可以看作是那些重要系统活动的抽象，它使四个视图有机联系起来，从某种意义上说场景是最重要的需求抽象。在开发体系结构时，它可以帮助设计者找到体系结构的构件和它们之间的作用关系。同时，也可以用场景来分析一个特定的视图，或描述不同视图构件间是如何相互作用的。场景可以用文本表示，也可以用图形表示

C/S（客户-服务器）风格定义：C/S 软件体系结构是基于资源不对等，且为实现共享而提出来的。它定义了工作站如何与服务器相连，以实现数据和应用分布到多个处理机上。有三个主要组成部分：数据库服务器、客户应用程序和网络。

C/S 风格优点：

具有强大的数据操作和事务处理能力，模型思想简单，易于人们理解和接受系统的客户应用程序和服务器构件分别运行在不同的计算机上，系统中每台服务器都可以适合各构件的要求，这对于硬件和软件的变化显示出极大的适应性和灵活性，而且易于对系统进行扩充和缩小

在 C/S 体系结构中，客户应用程序的开发集中于数据的显示和分析，而数据库服务器的开发则集中于数据的管理，不必在每一个新的应用程序中都要对一个 DBMS 进行编码。将大的应用处理任务分布到许多通过网络连接的低成本计算机上，以节约大量费用

缺点：开发成本较高。客户端程序设计复杂。信息内容和形式单一。用户界面风格不一，使用繁杂，不利于推广使用。软件移植困难。软件维护和升级困难。新技术不能轻易应用。

12. 三层 C/S 风格：三层：表示层，功能层，数据层。

优点：1 允许合理地划分三层结构的功能，使之在逻辑上保持相对独立性，能提高系统和软件的可维护性和可扩展性 2 允许更灵活有效地选用相应的平台和硬件系统，使之在处理负荷能力上与处理特性上分别适应于结构清晰的三层；并且这些平台和各个组成部分可以具有良好的可升级性和开放性 3 应用的各层可以并行开发，可以选择各自最适合的开发语言 4 利用功能层有效地隔离开表示层与数据层，未授权的用户难以绕过功能层而利用数据库工具或黑客手段去非法地访问数据层，为严格的安全管理奠定了坚实的基础

缺点：1 三层 C/S 结构各层间的通信效率若不高，即使分配给各层的硬件能力很强，其作为整体来说也达不到所要求的性能 2 设计时必须慎重考虑三层间的通信方法、通信频度及数据量。这和提高各层的独立性一样是三层 C/S 结构的关键问题

B/S 风格（浏览/服务器）：就是三层 C/S 应用结构的一种实现方式，其具体结构为：浏览器/Web 服务器/数据库服务器

优点：1 基于 B/S 体系结构的软件，系统安装、修改和维护全在服务器端解决。用户在使用系统时，仅仅需要一个浏览器就可运行全部的模块，真正达到了

“零客户端”的功能，很容易在运行时自动升级 2 还提供了异种机、异种网、异种应用服务的联机、联网、统一服务的最现实的开放性基础

缺点：1 B/S 体系结构缺乏对动态页面的支持能力，没有集成有效的数据库处理功能 2 系统扩展能力差，安全性难以控制 3 在数据查询等响应速度上，要远远地低于 C/S 体系结构 4 数据提交一般以页面为单位，数据的动态交互性不强，不利于在线事务处理(OLTP)应用

HMB 风格（基于层次消息总线的体系结构风格）： HMB 风格基于层次消息总线/支持构件的分布和并发，构件之间通过消息总线进行通信。消息总线是系统的连接件，负责消息的分派、传递和过滤以及处理结果的返回。各个构件挂接在消息总线上，向总线登记感兴趣的消息类型。构件根据需要发出消息，由消息总线负责把该消息分派到系统中所有对此消息感兴趣的构件，消息是构件之间通信的唯一方式。构件接收到消息后，根据自身状态对消息进行响应，并通过总线返回处理结果。由于构件通过总线进行连接，并不要求各个构件具有相同的地址空间或局限在一台机器上。

附录 B：SQL 核心数据

```
/*
Navicat MySQL Data Transfer

Source Server          : Jlong
Source Server Type     : MySQL
Source Server Version  : 80013
Source Host            : localhost:3306
Source Schema          : db_film

Target Server Type     : MySQL
Target Server Version  : 80013
File Encoding          : 65001

Date: 03/09/2019 17:18:36
*/

SET NAMES utf8mb4;
SET FOREIGN_KEY_CHECKS = 0;

--
-- Table structure for t_admin
--
DROP TABLE IF EXISTS `t_admin`;
CREATE TABLE `t_admin` (
  `admin_id` int(4) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '管理员 id',
```

```

    `name` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DE
FAULT NULL COMMENT '管理员姓名',
    `password` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci NU
LL DEFAULT NULL COMMENT '管理员密码',
    `sex` varchar(10) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEF
AULT NULL COMMENT '管理员性别',
    `phone` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL D
EFAULT NULL COMMENT '管理员联系电话',
    `birthday` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NUL
L DEFAULT NULL COMMENT '出生年月日',
    `avatar` varchar(255) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL
DEFAULT NULL COMMENT '管理员头像',
    PRIMARY KEY (`admin_id`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8 COLLATE = utf
8_unicode_ci ROW_FORMAT = Dynamic;

```

```

-- -----
-- Records of t_admin
-- -----

```

```

INSERT INTO `t_admin` VALUES (1, 'admin', 'admin', '男
', '13414850215', '1999-9-9', '/images/admin/admin.gif');

```

```

-- -----
-- Table structure for t_cinema
-- -----

```

```

DROP TABLE IF EXISTS `t_cinema`;
CREATE TABLE `t_cinema` (
    `cinema_id` int(4) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '影院 id',
    `cinema_name` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci
NULL DEFAULT NULL COMMENT '影院名称',
    `cinema_phone` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci
NULL DEFAULT NULL COMMENT '影院电话',
    `province` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci NUL
L DEFAULT NULL COMMENT '影院所在省份',
    `city` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DE
FAULT NULL COMMENT '影院所在市',
    `county` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL
DEFAULT NULL COMMENT '影院所在区县',
    `specified_address` varchar(255) CHARACTER SET utf8 COLLATE utf8_unic
ode_ci NULL DEFAULT NULL COMMENT '影院详细地址',
    PRIMARY KEY (`cinema_id`) USING BTREE
) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8_unicode_ci ROW_FO
RMAT = Dynamic;

```

```

-- -----
-- Records of t_cinema
-- -----

INSERT INTO `t_cinema` VALUES (1, '横店电影城(庆丰店)', '13498476301', '广东', '广州', '白云区', '白云区庆丰广场路财智广场 2 楼');
INSERT INTO `t_cinema` VALUES (2, '尚影影院', '13745454545', '广东', '广州', '白云区', '白云区石夏路 288 号金铂广场 7 楼');
INSERT INTO `t_cinema` VALUES (4, '烽禾影城(祈福新邨店)', '13409989898', '广东', '广州', '番禺区', '番禺区钟屏路钟福广场 3 层(祈福新村祈福医院旁)');
INSERT INTO `t_cinema` VALUES (5, '沙湾 3D 数字电影院', '13465102832', '广东', '广州', '番禺区', '番禺区沙湾镇大巷涌路沙湾文化中心 2 楼(沁芳园对面)');
INSERT INTO `t_cinema` VALUES (6, '大地影院(奥体高德美居店)', '13454729283', '广东', '广州', '天河区', '天河区奥体南路 12 号高德美居家居馆 3 楼(近天河百货)');
INSERT INTO `t_cinema` VALUES (7, '金逸国际影城(维家思店)', '13483743433', '广东', '广州', '天河区', '天河区黄埔大道西 188 号维家思广场 3 楼(富力盈隆广场对面)');
INSERT INTO `t_cinema` VALUES (8, 'FUN 范影城(喜洋时代东圃店)', '13418226112', '广东', '广州', '天河区', '天河区东圃大马路 4 号四季时尚荟 A 栋 1 楼(近东圃购物中心)');
INSERT INTO `t_cinema` VALUES (14, '万达影院', '13498746534', NULL, NULL, NULL, '天河区长湴北街 34 号 3 楼');

-- -----
-- Table structure for t_comment
-- -----

DROP TABLE IF EXISTS `t_comment`;
CREATE TABLE `t_comment` (
  `comment_id` int(4) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '评论 id',
  `user_id` int(4) UNSIGNED NULL DEFAULT NULL COMMENT '用户 id',
  `movie_id` int(4) UNSIGNED NULL DEFAULT NULL COMMENT '电影 id',
  `user_score` int(4) UNSIGNED NULL DEFAULT NULL COMMENT '用户评分',
  `comment_content` text CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL COMMENT '用户评论内容',
  `comment_date` datetime(6) NULL DEFAULT NULL COMMENT '评论日期',
  `support_num` int(4) UNSIGNED NULL DEFAULT NULL COMMENT '点赞数',
  `is_pass` int(1) NULL DEFAULT 0 COMMENT '评论是否通过审核',
  `support_user` text CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL COMMENT '点赞用户数组',
  PRIMARY KEY (`comment_id`) USING BTREE,
  INDEX `t_comment_ibfk_1` (`user_id`) USING BTREE,
  INDEX `t_comment_ibfk_2` (`movie_id`) USING BTREE,

```

```

    CONSTRAINT `t_comment_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `t_u
ser` (`user_id`) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT `t_comment_ibfk_2` FOREIGN KEY (`movie_id`) REFERENCES `t_
movie` (`movie_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE = InnoDB AUTO_INCREMENT = 58 CHARACTER SET = utf8 COLLATE = ut
f8_unicode_ci ROW_FORMAT = Dynamic;

```

```

-- -----
-- Records of t_comment
-- -----

```

```

INSERT INTO `t_comment` VALUES (57, 1, 4, 9, '很好看，挺悲伤的，让我哭一
会!', '2019-08-08 15:59:38.000000', 2, 1, '[1,53]');
INSERT INTO `t_comment` VALUES (58, 1, 7, 8, '很好看，有一起的吗?
', '2019-08-08 16:05:46.000000', 2, 1, '[53,1]');

```

```

-- -----
-- Table structure for t_hall
-- -----

```

```

DROP TABLE IF EXISTS `t_hall`;
CREATE TABLE `t_hall` (
  `hall_id` int(4) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '影厅 id',
  `cinema_id` int(4) UNSIGNED NULL DEFAULT NULL COMMENT '影院 id',
  `name` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DE
FAULT NULL COMMENT '影厅名称',
  PRIMARY KEY (`hall_id`) USING BTREE,
  INDEX `t_hall_ibfk_1` (`cinema_id`) USING BTREE,
  CONSTRAINT `t_hall_ibfk_1` FOREIGN KEY (`cinema_id`) REFERENCES `t_ci
nema` (`cinema_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE = InnoDB AUTO_INCREMENT = 40 CHARACTER SET = utf8 COLLATE = ut
f8_unicode_ci ROW_FORMAT = Dynamic;

```

```

-- -----
-- Records of t_hall
-- -----

```

```

INSERT INTO `t_hall` VALUES (30, 1, '1 号激光厅');
INSERT INTO `t_hall` VALUES (32, 2, '1 号激光厅');
INSERT INTO `t_hall` VALUES (34, 4, '1 号激光厅');
INSERT INTO `t_hall` VALUES (35, 5, '1 号激光厅');
INSERT INTO `t_hall` VALUES (36, 6, '1 号激光厅');
INSERT INTO `t_hall` VALUES (37, 7, '1 号激光厅');
INSERT INTO `t_hall` VALUES (38, 8, '1 号激光厅');
INSERT INTO `t_hall` VALUES (40, 5, '2 号激光厅');

```

```

-- -----

```

```
-- Table structure for t_movie
-----
DROP TABLE IF EXISTS `t_movie`;
CREATE TABLE `t_movie` (
  `movie_id` int(4) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '电影 id',
  `name` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DE
FAULT NULL COMMENT '电影名称',
  `poster` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL
DEFAULT NULL COMMENT '电影海报',
  `director` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NUL
L DEFAULT NULL COMMENT '导演',
  `actor` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL
DEFAULT NULL COMMENT '主演',
  `movie_long` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci N
ULL DEFAULT NULL COMMENT '电影片长',
  `language` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci NUL
L DEFAULT NULL COMMENT '电影语言版本',
  `intro` text CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL COMMENT
'电影简介',
  `type` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DE
FAULT NULL COMMENT '电影类型',
  `public_date` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci
NULL DEFAULT NULL COMMENT '电影上映时间',
  `wish_num` int(4) UNSIGNED NULL DEFAULT 0 COMMENT '想看人数',
  `score` decimal(4, 1) NULL DEFAULT NULL COMMENT '电影综合评分',
  PRIMARY KEY (`movie_id`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 39 CHARACTER SET = utf8 COLLATE = ut
f8_unicode_ci ROW_FORMAT = Dynamic;
```

```
-- -----
-- Records of t_movie
-- -----
```

```
INSERT INTO `t_movie` VALUES (1, '驯龙高手
3', '/images/movie/hot1.jpg', '迪恩·德布洛斯', '刘昊然', '130 分钟', '英语
', '统领伯克岛的酋长嗝嗝（刘昊然 配音），与阿丝翠德（亚美莉卡·费雷拉 配音）共
同打造了一个奇妙而热闹的飞龙乌托邦。一只雌性光煞飞龙的意外出现，加上一个前所未
有的威胁的到来，令嗝嗝和没牙仔不得不离开自己唯一的家园，前往他们本以为只存在于
神话之中的隐秘之境。在发现自己真正的命运之后，飞龙与骑士将携手殊死奋战，保护他
们所珍爱的一切。', '动漫', '2018-12-11', 0, 0.0);
INSERT INTO `t_movie` VALUES (3, '夏目友人帐
', '/images/movie/hot3.jpg', '大森贵弘', '村濑步', '105 分钟', '日语', '夏
目（神谷浩史 配音）在一次归还妖怪名字的过程中，结识了外祖母玲子（小林沙苗 配
音）的旧识津村容莉枝岛本须美 配音）和她的儿子棕雄（高良健吾 配音）。在与其接触
```

后，夏目的“保镖”猫咪老师（井上和彦 配音）竟第一次意外分裂成了三只！夏目能否解决危机？这对母子又与妖怪有何关系？', '言情', '2018-11-13', 1, NULL);

INSERT INTO `t_movie` VALUES (4, '比悲伤更悲伤的故事', '/images/movie/hot4.jpg', '林孝谦', '陈庭妮', '105 分钟', '国语', '唱片制作人张哲凯（刘以豪 饰）和王牌作词人宋媛媛（陈意涵 饰）相依为命，两人自幼身世坎坷只有彼此为伴，他们是亲人、是朋友，也仿佛是命中注定的另一半。父亲罹患遗传重症而被母亲抛弃的哲凯，深怕自己随时会发病不久人世，始终没有跨出友谊的界线对媛媛展露爱意。眼见哲凯的病情加重，他暗自决定用剩余的生命完成他们之间的终曲，再为媛媛找个可以托付一生的好男人。这时，事业有成温柔体贴的医生（张书豪 饰）适时的出现让他成为照顾媛媛的最佳人选，二人按部就班发展着关系。一切看似都在哲凯的计划下进行。然而，故事远比这里所写更要悲伤', '言情', '2018-9-1', 0, 9.0);

INSERT INTO `t_movie` VALUES (5, '惊奇队长', '/images/movie/hot5.jpg', '瑞安·弗雷克', '本·门德尔森', '108 分钟', '英语', '卡罗尔·丹弗斯（布丽·拉尔森 饰）是克里人的精英部队一员，但她挣扎着回忆过去，对于怎么获得超能力、以及之前的人生等超能力之前的人生，她只在梦中有片段不成章的记忆，这层困扰引起的情绪波动也进而影响她对超能力的掌控。所幸，在星际战队指挥官（裘德·洛 饰）的调教之下，卡罗尔·丹弗斯学会如何控制自己的超能力。一次外星任务中，卡罗尔·丹弗斯因故来到了地球，并遇到还是低阶探员的尼克·弗瑞（塞缪尔·杰克逊 饰），卡罗尔·丹弗斯与尼克·弗瑞一起追捕在地球惹事的变形人塔罗斯（本·门德尔森 饰），同时，也发现自己如何获得超能力。', '爱情', '2018-7-3', 1, 0.0);

INSERT INTO `t_movie` VALUES (6, '我的英雄学院', '/images/movie/hot6.jpg', '长崎健司', '生濑胜久', '96 分钟', '日语', '讲述在绝大多数人都有超能力的未来时代，没有天赋的异能少年，如何凭努力与坚持打破绝境，成长为最强英雄，守卫世界和平的故事。 剧场版以移动科研都市“我之岛”为舞台，描绘正反两派的殊死激战。在第一英雄被困之际，少年与英雄科同学拼死克敌，却在无意中探得隐秘真相，剧情随之反转', '科幻', '2018-8-13', 1, NULL);

INSERT INTO `t_movie` VALUES (7, '我叫为何', '/images/movie/1.jpg', '李小军', '乔新峰', '86 分钟', '国语', '讲述了两个技术宅男为何和吕相立志设计一款以圣朝名将无束为主角的火爆游戏。作为战神无束的死忠粉丝，为何在设计游戏的过程中，意外穿越到了圣朝，有机会亲眼目睹无束驱除西土的宏大场面。', '喜剧', '2019-8-20', 1, 8.0);

INSERT INTO `t_movie` VALUES (8, '如影随心', '/images/movie/2.jpg', '霍建起', '马苏', '96 分钟', '国语', '小提琴家陆松（陈晓 饰）与室内设计师文罂（杜鹃 饰）巴黎邂逅，不打不相识，彼此迷恋越陷越深。爱情之名怂恿着他们为爱再大胆一次，走出各自问题满满的情感关系。仿佛只要现任变前任，幸福就唾手可得。然而前任们也都是战士，文罂的前任郑可（华少 饰）是个理性至极的医生，陆松的前任刘娟（马苏 饰）则在情感社区开启了喋喋不休的骂战，是继续坚持对爱的信仰，还是选择放下与原谅重新开始？现代都市情感危机在北京与巴黎街头上演。越是深爱的，往往也正是痛苦的源头，爱情，像是影子，抓不住却也离不开，如影随心', '爱情', '2019-9-11', 2, NULL);

INSERT INTO `t_movie` VALUES (9, '转型团伙', '/images/movie/3.jpg', '吴镇宇', '吴镇宇', '90 分钟', '国语', '过气影帝宇哥（吴镇宇 饰）与当红童星费曼（费曼 饰）因一场拍摄意外而陷入黑帮纷争的故事。失忆的宇哥意外“入戏”，以为自己就是真正的社团大哥，为了赚钱照顾儿子决定重出江湖，殊不知时代已经改变，引发一连

串令人啼笑皆非的乌龙事件，更上演了险象环生的黑帮纷争', '剧情', '2019-7-28', 2, NULL);

INSERT INTO `t_movie` VALUES (10, '境界', '/images/movie/4.jpg', '佐藤信介', '真野惠里菜', '108 分钟', '日语', '从小异于常人的黑崎一护（福士苍汰 饰），在一次机缘巧合下遭遇女死神“朽木露琪亚”（杉咲花 饰）。露琪亚是将灵魂从现实世界引领到尸魂界的收魂者之一，他们为保护人类不被“虚”侵害，一直在与这些怪物战斗。在与一只异常强大的“虚”战斗时，露琪亚身负重伤，无奈只能将死神的力量暂时转给黑崎一护，让其代为战斗。两人经过一段时间的磨合后并肩战斗，终于战胜了当年害死一护母亲的“虚”，但在这期间，露琪亚的兄长朽木白哉（石原贵雅 饰）强迫她不顾黑崎一护的生命安危收回法力。露琪亚为保护一护，拖延到与“虚”的战斗结束，并清除了一护的记忆', '动作', '2019-11-12', 1, NULL);

INSERT INTO `t_movie` VALUES (11, '神奇乐园历险记', '/images/movie/5.jpg', 'Dylan Brown', '基南·汤普森', '95 分钟', '英语', '古灵精怪的十二岁女孩朱妮（Sofia Mali 配音），偶然间她发现了一座神奇的乐园。小朱妮在乐园里遇到了憨厚可掬的蓝熊布默（肯·哈德森·坎贝尔 配音），善良的疣猪格里塔（米拉·库尼斯 配音），贫嘴的豪猪史蒂夫（约翰·奥利弗 配音），滑稽搞笑的海狸兄弟库珀（郑肯 配音）和格斯（基南·汤普森 配音），充满奇幻想象力的首领和乐园里炙手可热的动物明星猴子花生，并很快和它们成为朋友。这座隐藏在森林中的神奇乐园不仅有着形形色色的动物，还有很多惊险刺激的游乐设施。但从动物朋友口中，小朱妮得知乐园已经衰败无法运营。很快，小朱妮就意识到这座神奇乐园和她之前想象中的乐园一模一样。只有她能拯救这座已经没落的乐园。于是，她和动物朋友们组成了一只小分队，希望帮助乐园重现昔日繁华。', '冒险', '2019-12-1', 0, NULL);

INSERT INTO `t_movie` VALUES (12, '毕业旅行之逍遥骑士', '/images/movie/6.jpg', '郭奎永', '田雨', '90 分钟', '国语', '高考落榜的文艺少年杨风（张朔祯 饰）梦想成为一个“嬉皮士”浪子，并誓要在毕业旅行中完成自己的浪子清单：打架、泡妞、吸烟喝酒、打坐冥想、和摇滚歌手交朋友.....他在好友陈火（赵一凡 饰）的相助下伪装成“逍遥骑士”，想借助漂亮师姐思思（安紫依 饰）完成自己的“破处之旅”，却没想到思思也暗藏自己的“少女成人计划”，“逍遥路上”演绎啼笑皆非的动人故事。', '青春', '2019-11-11', 2, 0.0);

INSERT INTO `t_movie` VALUES (13, '复仇者联盟 4', '/images/movie/7.jpg', '乔·罗素', '小罗伯特·唐尼', '181 分钟', '英语', '改编自漫威漫画，也是漫威电影宇宙第 22 部影片。复仇者联盟的一众超级英雄，必须抱着牺牲一切的信念，与史上最强大反派灭霸殊死一搏，阻止其摧毁宇宙的邪恶计划。', '动作', '2019-4-24', 3, 0.0);

INSERT INTO `t_movie` VALUES (14, '撞死了一只羊', '/images/movie/8.jpg', '万玛才旦', '金巴', '87 分钟', '国语', '司机金巴（金巴 饰）在路上撞死了一只羊，决意超度此羊；杀手金巴（更登彭措 饰）即将找到杀父仇人，准备报仇雪恨。阴差阳错，杀手金巴搭上了司机金巴的卡车。于是，两个叫金巴的男人的命运便神秘地联系在了一起，一段惊心动魄的旅程开始.....', '剧情', '2019-8-12', 0, NULL);

INSERT INTO `t_movie` VALUES (15, '云雾笼罩的山峰', '/images/movie/9.jpg', '左志国', '王海涛', '92 分钟', '国语', '一位年轻的父亲为了筹钱医治自己病重的爱女，又要不到拖欠的赔款，被迫铤而走险绑架了老板的女儿。然而他不知道的是，在这座幽静的山里，还有其他人带着不同的目的和欲望参与了

这件事情，一座山上、一日之内、六组人马，各自的贪念在错综复杂的因果之下，逐渐失控，转向不可预知的方向。罪恶之中，谁能逃脱命运的审判？', '悬疑', '2019-7-14', 1, NULL);

INSERT INTO `t_movie` VALUES (16, '九克拉战栗', '/images/movie/10.jpg', '林峻兆', '付然', '86 分钟', '国语', '珠宝店店员包小萱鬼迷心窍捡走车祸死者一女老板虞娜的钻戒，这是枚被恶魔诅咒的钻戒。瞬间，包小萱被虞娜灵魂附身，同时，这场车祸似乎也从没发生过。随之各种恐怖事件纷纷降临在被虞娜灵魂附身的包小萱身上.....', '恐怖', '2019-7-1', 0, NULL);

INSERT INTO `t_movie` VALUES (17, '绿皮书', '/images/movie/hot7.jpg', '皮特·法雷里', '维果·莫腾森', '130 分钟', '英语', '一名黑人钢琴家，为前往种族歧视严重的南方巡演，找了一个粗暴的白人混混做司机。在一路开车南下的过程里，截然不同的两人矛盾不断，引发了不少争吵和笑料。但又在彼此最需要的时候，一起共渡难关。行程临近结束，两人也慢慢放下了偏见..... 绿皮书，是一本专为黑人而设的旅行指南，标注了各城市中允许黑人进入的旅店、餐馆。电影由真实故事改编。', '剧情', '2019-3-1', 1, 0.0);

INSERT INTO `t_movie` VALUES (18, '阿丽塔', '/images/movie/hot8.jpg', '罗伯特·罗德里', '罗莎·萨拉查', '123 分钟', '英语', '未来 26 世纪，科技发展，人类与机械改造人共存，弱肉强食是钢铁城唯一的生存法则。依德（克里斯托夫·沃尔兹 饰）是钢铁城著名的改造人医生，他在垃圾场发现了一个半机械少女残躯，依德医生将其拯救后为她取名阿丽塔（罗莎·萨拉扎尔 饰）。阿丽塔虽然重获生命却失去了记忆，如一个新生儿一样对这个世界充满新鲜感。在依德医生与好友雨果（基恩·约翰逊 饰）的帮助下，她逐步适应着新生活 and 街头险恶。一次偶然的机会，阿丽塔发现自己竟有着惊人的战斗天赋。一次次猎杀激发着她的觉醒，阿丽塔逐渐明白自己注定为战斗而生，为正义而战。一场揭开自己身世之谜，并打破宇宙旧秩序的史诗级冒险之旅就这样展开。', '动作', '2019-2-22', 0, NULL);

INSERT INTO `t_movie` VALUES (19, '狗眼看人心', '/images/movie/hot9.jpg', '吴楠', '黄磊', '91 分钟', '国语', '本片讲述了一桩因“狗咬狗”引发的连锁故事。余峰（黄磊 饰）、亮亮（闫妮 饰）、余一诺（安吉 饰）家的小狗妮蔻，为了救主人被隔壁藏獒咬成重伤，有权有势的藏獒主人却对余峰一家百般刁难。面对种种不公，余峰不再忍气吞声，发誓要为小狗讨回公道，但柴女士（汤唯 饰）等人的出现，却让矛盾进一步激化.....狗生不易，小狗愿意用生命守护我们，我们又能为它做什么呢？', '剧情', '2019-4-20', 0, 0.0);

INSERT INTO `t_movie` VALUES (20, '祈祷落幕时', '/images/movie/hot10.jpg', '福泽克雄', '阿部宽', '119 分钟', '日语', '在东京都葛饰区小菅的某座公寓里，发现了一具被勒死的女性遗体。被害者是在清洁公司上班、住在滋贺县的押谷道子（中岛博子 饰）。而杀人现场的住户越川睦夫也下落不明。而以松宫（沟端淳平 饰）等警视厅搜查一科的刑警们查不出越川与押谷道子有何交集，案件调查陷入困境。尽管查明押谷道子是为了拜访她的同学，即导演浅居博美（松岛菜菜子 饰）才来东京的，但博美与越川之间也找不出什么交集。这时，案发地点附近发现了被烧死的尸体，松宫怀疑两者有关联。从遗物中，发现了写有日本桥周围 12 座桥名的文字。知道此事后加贺恭一郎（阿部宽 饰）非常激动，因为那东西与他的亡母大有关联。', '剧情', '2019-4-12', 2, 0.0);

INSERT INTO `t_movie` VALUES (29, '大侦探皮卡丘', '/images/movie/1557131178228.jpg', '卡梅隆', '皮卡丘', '104 分钟', '英

语', '蒂姆·古德曼（贾斯提斯·史密斯 饰） 为寻找下落不明的父亲来到莱姆市，意外与父亲的前宝可梦搭档大侦探皮卡丘（瑞安·雷诺兹 配音）相遇，并惊讶地发现自己是唯一能听懂皮卡丘说话的人类，他们决定组队踏上揭开真相的刺激冒险之路。探案过程中他们邂逅了各式各样的宝可梦，并意外发现了一个足以毁灭整个宝可梦宇宙的惊天阴谋。', '冒险', '2019-5-20', 3, 0.0);

```
INSERT INTO `t_movie` VALUES (33, '罗马', '/images/movie/1557394982584.jpg', '张三', '李四', '1200 分钟', '粤语', '嘻嘻嘻嘻嘻嘻', '言情', '2019-5-23', 1, 0.0);
```

```
INSERT INTO `t_movie` VALUES (34, '来电狂响', '/images/movie/1557457509601.jpg', '张三', '董大伟', '130 分钟', '国语', '七个好友聚餐，有人提议大家玩一个将手机公开的游戏：与在场所有人分享当晚收到的每一条短信、每一个电话，每一个弹窗，由此掀开了一场轩然大波。 一条短信，拆散一对恩爱夫妻；一个电话，推翻多年发小感情.....数字时代，小小手机藏了无数秘密，当手机变成锋利的利刃，当聚餐局变成“揭 秘局”，信任危机过后他们能否跨越信任危机，重新思考彼此之间的亲密关系.....', '悬疑', '2019-5-31', 2, 0.0);
```

```
INSERT INTO `t_movie` VALUES (35, '雪暴', '/images/movie/1557555972902.jpg', '张三', '李四', '120 分钟', '国语', '无', '喜剧', '2019-5-17', 1, 0.0);
```

```
INSERT INTO `t_movie` VALUES (36, '一个母亲的复仇', '/images/movie/1563173524205.jpg', '王明', '张三', '118 分钟', '国语', '一个母亲的复仇', '悬疑', '2019-11-14', 0, 0.0);
```

```
INSERT INTO `t_movie` VALUES (37, '蜘蛛侠：英雄远征', '/images/movie/1563173216299.jpg', '大奔', '蜘蛛', '180 分钟', '英语', '故事全面延续“复联 4”，蜘蛛侠志承钢铁侠远征欧洲，独挑大梁对抗群敌！新角色“神秘客”穿越多元宇宙霸气登场！神盾局局长尼克·弗瑞回归领军！“漫威新铁三角”组合强势出击！全新蜘蛛战衣酷炫升级！史诗对决燃爆今夏突破想象极限！', '科幻', '2019-7-11', 0, NULL);
```

```
INSERT INTO `t_movie` VALUES (38, '银河补习班', '/images/movie/1563180596034.jpg', '李四', '王五', '156 分钟', '英语', '银河补习班', '喜剧', '2019-7-11', 0, NULL);
```

```
-- -----  
-- Table structure for t_order  
-- -----
```

```
DROP TABLE IF EXISTS `t_order`;  
CREATE TABLE `t_order` (  
  `order_id` int(4) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '订单 id',  
  `user_id` int(4) UNSIGNED NULL DEFAULT NULL COMMENT '用户 id',  
  `schedule_id` int(4) UNSIGNED NULL DEFAULT NULL COMMENT '影院 id',  
  `order_phone` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci  
  NULL DEFAULT NULL COMMENT '下单手机',  
  `order_date` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci  
  NULL DEFAULT NULL COMMENT '用户下单时间',  
  `ticket_num` int(1) UNSIGNED NULL DEFAULT NULL COMMENT '购买电影票数',
```

```

    `ticket_total_price` decimal(10, 2) UNSIGNED NULL DEFAULT NULL COMMENT '电影票单价',
    `order_seat_info` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '座位信息',
    `pay_type` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '支付方式',
    `phone_code` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '手机取票码',
    PRIMARY KEY (`order_id`) USING BTREE,
    INDEX `t_order_ibfk_1` (`user_id`) USING BTREE,
    INDEX `t_order_ibfk_2` (`schedule_id`) USING BTREE,
    CONSTRAINT `t_order_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `t_user` (`user_id`) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT `t_order_ibfk_2` FOREIGN KEY (`schedule_id`) REFERENCES `t_schedule` (`schedule_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE = InnoDB AUTO_INCREMENT = 101 CHARACTER SET = utf8 COLLATE = utf8_unicode_ci ROW_FORMAT = Dynamic;

```

```

-----
-- Records of t_order
-----

```

```

INSERT INTO `t_order` VALUES (73, 1, 69, '13414850282', '2019-5-6', 3, 168.00, '[25,26,35]', '1', '341593');
INSERT INTO `t_order` VALUES (75, 1, 53, '13414850282', '2019-5-7', 3, 294.00, '[26,25,35]', '1', '612368');
INSERT INTO `t_order` VALUES (79, 50, 50, '13672606065', '2019-5-10', 1, 56.00, '[35]', '0', '823445');
INSERT INTO `t_order` VALUES (80, 50, 60, '13672606065', '2019-5-10', 2, 90.00, '[25,24]', '0', '961831');
INSERT INTO `t_order` VALUES (81, 25, 55, '13877777766', '2019-5-10', 3, 168.36, '[25,35,36]', '0', '393432');
INSERT INTO `t_order` VALUES (82, 29, 53, '13482638273', '2019-5-10', 1, 98.00, '[27]', '0', '149857');
INSERT INTO `t_order` VALUES (84, 35, 53, '13457555444', '2019-5-10', 4, 392.00, '[24,5,15,14]', '0', '285591');
INSERT INTO `t_order` VALUES (85, 35, 60, '13457555444', '2019-5-10', 2, 90.00, '[26,36]', '0', '194678');
INSERT INTO `t_order` VALUES (86, 23, 70, '13478565555', '2019-5-11', 3, 168.00, '[25,26,35]', '1', '674698');
INSERT INTO `t_order` VALUES (87, 1, 53, '13414850282', '2019-5-11', 3, 294.00, '[45,46,36]', '0', '897439');
INSERT INTO `t_order` VALUES (88, 1, 54, '13414850282', '2019-5-11', 4, 176.00, '[25,26,36,35]', '0', '481319');

```

```

INSERT INTO `t_order` VALUES (90, 1, 64, '13414850282', '2019-7-5', 2, 156.00, '[15,16]', '0', '146745');
INSERT INTO `t_order` VALUES (91, 1, 68, '13414850282', '2019-7-12', 4, 136.00, '[25,26,36,35]', '1', '572118');
INSERT INTO `t_order` VALUES (92, 1, 64, '13414850282', '2019-7-15', 4, 312.00, '[35,26,36,25]', '1', '859679');
INSERT INTO `t_order` VALUES (93, 1, 64, '13414850282', '2019-7-15', 2, 156.00, '[34,37]', '0', '847345');
INSERT INTO `t_order` VALUES (94, 1, 87, '13414850282', '2019-7-15', 4, 308.00, '[25,26,36,35]', '1', '987546');
INSERT INTO `t_order` VALUES (95, 1, 82, '13414850282', '2019-7-15', 4, 260.00, '[25,26,36,35]', '1', '751292');
INSERT INTO `t_order` VALUES (96, 1, 87, '13414850282', '2019-7-15', 2, 154.00, '[45,46]', '0', '135615');
INSERT INTO `t_order` VALUES (97, 1, 87, '13414850282', '2019-7-15', 1, 77.00, '[37]', '1', '586315');
INSERT INTO `t_order` VALUES (98, 1, 87, '13414850282', '2019-7-15', 1, 77.00, '[34]', '1', '666362');
INSERT INTO `t_order` VALUES (99, 1, 67, '13414850282', '2019-7-24', 4, 264.00, '[25,27,46,34]', '1', '323495');
INSERT INTO `t_order` VALUES (100, 1, 80, '13414850282', '2019-7-25', 4, 152.00, '[16,26,25,15]', '1', '577429');
INSERT INTO `t_order` VALUES (101, 1, 80, '13414850282', '2019-8-6', 2, 76.00, '[35,36]', '1', '821579');

-- -----
-- Table structure for t_schedule
-- -----

DROP TABLE IF EXISTS `t_schedule`;
CREATE TABLE `t_schedule` (
  `schedule_id` int(4) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '排片id',
  `movie_id` int(4) UNSIGNED NULL DEFAULT NULL COMMENT '电影id',
  `cinema_id` int(4) UNSIGNED NULL DEFAULT NULL COMMENT '影院id',
  `hall_name` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '影厅名字',
  `show_date` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '放映日期',
  `show_time` varchar(10) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '放映时间',
  `price` decimal(10, 2) NULL DEFAULT NULL COMMENT '电影售价',
  `seat_info` text CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL COMMENT '座位信息',
  PRIMARY KEY (`schedule_id`) USING BTREE,

```

```

INDEX `hall_id`(`hall_name`) USING BTREE,
INDEX `t_schedule_ibfk_1`(`movie_id`) USING BTREE,
INDEX `t_schedule_ibfk_2`(`cinema_id`) USING BTREE,
CONSTRAINT `t_schedule_ibfk_1` FOREIGN KEY (`movie_id`) REFERENCES `t
_movie` (`movie_id`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `t_schedule_ibfk_2` FOREIGN KEY (`cinema_id`) REFERENCES `
t_cinema` (`cinema_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE = InnoDB AUTO_INCREMENT = 102 CHARACTER SET = utf8 COLLATE = u
tf8_unicode_ci ROW_FORMAT = Dynamic;

```

```

-- -----
-- Records of t_schedule
-- -----

```

```

INSERT INTO `t_schedule` VALUES (50, 20, 5, '1 号激光厅', '2019-5-
22', '20:05', 56.00, '[35]');
INSERT INTO `t_schedule` VALUES (53, 13, 6, '1 号激光厅', '2019-5-
31', '12:05', 98.00, '[35,27,24,5,15,14,45,46,36]');
INSERT INTO `t_schedule` VALUES (54, 16, 4, '1 号激光厅', '2019-7-
10', '20:05', 44.00, '[25,26,36,35]');
INSERT INTO `t_schedule` VALUES (55, 19, 6, '1 号激光厅', '2019-5-
14', '16:05', 56.12, '[25,35,36]');
INSERT INTO `t_schedule` VALUES (56, 17, 8, '1 号激光厅', '2019-5-
16', '16:05', 38.10, '[]');
INSERT INTO `t_schedule` VALUES (59, 5, 6, '1 号激光厅', '2019-5-
24', '16:05', 76.00, NULL);
INSERT INTO `t_schedule` VALUES (60, 6, 2, '1 号激光厅', '2019-5-
22', '08:05', 45.00, '[25,24,26,36]');
INSERT INTO `t_schedule` VALUES (63, 4, 2, '1 号激光厅', '2019-5-
23', '08:05', 46.10, NULL);
INSERT INTO `t_schedule` VALUES (64, 8, 7, '1 号激光厅', '2019-9-
24', '12:05', 78.00, '[15,16,35,26,36,25,34,37]');
INSERT INTO `t_schedule` VALUES (67, 12, 6, '1 号激光厅', '2019-11-
18', '16:05', 66.00, '[25,27,46,34]');
INSERT INTO `t_schedule` VALUES (68, 15, 1, '1 号激光厅', '2019-7-
21', '16:05', 34.00, '[25,26,36,35]');
INSERT INTO `t_schedule` VALUES (69, 29, 5, '2 号激光厅', '2019-5-
20', '20:05', 56.00, '[25,26,35]');
INSERT INTO `t_schedule` VALUES (70, 20, 5, '1 号激光厅', '2019-5-
24', '16:05', 56.00, '[25,26,35]');
INSERT INTO `t_schedule` VALUES (74, 34, 4, '1 号激光厅', '2019-6-
26', '12:05', 78.90, NULL);
INSERT INTO `t_schedule` VALUES (75, 3, 6, '1 号激光厅', '2019-5-
21', '16:05', 36.70, NULL);

```

```

INSERT INTO `t_schedule` VALUES (76, 35, 4, '1号激光厅', '2019-5-24', '12:05', 76.00, NULL);
INSERT INTO `t_schedule` VALUES (77, 3, 2, '1号激光厅', '2019-5-31', '12:05', 78.00, NULL);
INSERT INTO `t_schedule` VALUES (78, 13, 2, '1号激光厅', '2019-6-30', '20:05', 78.00, NULL);
INSERT INTO `t_schedule` VALUES (79, 1, 7, '1号激光厅', '2019-7-11', '08:05', 45.00, '');
INSERT INTO `t_schedule` VALUES (80, 4, 8, '1号激光厅', '2019-8-14', '20:05', 38.00, '[16,26,25,15,35,36]');
INSERT INTO `t_schedule` VALUES (81, 37, 4, '1号激光厅', '2019-7-30', '12:05', 76.00, NULL);
INSERT INTO `t_schedule` VALUES (82, 16, 1, '1号激光厅', '2019-7-21', '20:05', 65.00, '[25,26,36,35]');
INSERT INTO `t_schedule` VALUES (83, 18, 2, '1号激光厅', '2019-7-30', '12:05', 34.00, NULL);
INSERT INTO `t_schedule` VALUES (84, 36, 7, '1号激光厅', '2019-11-22', '12:05', 78.00, NULL);
INSERT INTO `t_schedule` VALUES (85, 37, 5, '2号激光厅', '2019-7-24', '20:05', 56.00, NULL);
INSERT INTO `t_schedule` VALUES (86, 33, 7, '1号激光厅', '2019-7-30', '16:05', 83.00, NULL);
INSERT INTO `t_schedule` VALUES (87, 13, 8, '1号激光厅', '2019-7-24', '16:05', 77.00, '[25,26,36,35,45,46,37,34]');
INSERT INTO `t_schedule` VALUES (88, 3, 6, '1号激光厅', '2019-7-23', '20:05', 53.00, NULL);
INSERT INTO `t_schedule` VALUES (89, 34, 5, '2号激光厅', '2019-7-31', '20:05', 45.00, NULL);
INSERT INTO `t_schedule` VALUES (90, 10, 2, '1号激光厅', '2019-11-30', '12:05', 56.00, NULL);
INSERT INTO `t_schedule` VALUES (91, 18, 5, '2号激光厅', '2019-7-25', '20:05', 78.00, NULL);
INSERT INTO `t_schedule` VALUES (92, 7, 1, '1号激光厅', '2019-8-31', '16:05', 89.00, NULL);
INSERT INTO `t_schedule` VALUES (93, 8, 6, '1号激光厅', '2019-9-30', '20:05', 49.00, '');
INSERT INTO `t_schedule` VALUES (94, 9, 2, '1号激光厅', '2019-7-30', '16:05', 45.00, NULL);
INSERT INTO `t_schedule` VALUES (95, 5, 4, '1号激光厅', '2019-7-25', '12:05', 34.00, NULL);
INSERT INTO `t_schedule` VALUES (96, 35, 2, '1号激光厅', '2019-7-24', '16:05', 34.00, NULL);
INSERT INTO `t_schedule` VALUES (98, 38, 2, '1号激光厅', '2019-7-17', '16:05', 45.00, NULL);

```

```

INSERT INTO `t_schedule` VALUES (99, 8, 7, '1号激光厅', '2019-9-30', '16:05', 45.00, NULL);
INSERT INTO `t_schedule` VALUES (100, 8, 7, '1号激光厅', '2019-10-1', '20:05', 40.00, NULL);
INSERT INTO `t_schedule` VALUES (101, 4, 7, '1号激光厅', '2019-8-9', '20:05', 45.00, NULL);

-- -----
-- Table structure for t_user
-- -----

DROP TABLE IF EXISTS `t_user`;
CREATE TABLE `t_user` (
  `user_id` int(4) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '用户id',
  `user_name` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '用户名',
  `avatar` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '用户头像',
  `password` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '用户密码',
  `phone` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '用户手机号码',
  `sex` varchar(10) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '用户性别',
  `birthday` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '用户出生年月日',
  `sign` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '用户签名',
  PRIMARY KEY (`user_id`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 55 CHARACTER SET = utf8 COLLATE = utf8_unicode_ci ROW_FORMAT = Dynamic;

-- -----
-- Records of t_user
-- -----

INSERT INTO `t_user` VALUES (1, 'LyuJlong', '/images/avatar/man.jpg', '123456', '13414850282', '男', '2019-8-12', '我命由我不由天');
INSERT INTO `t_user` VALUES (19, '张三', '/images/avatar/1557241550286.jpg', '123', '13423232323', '男', '2019-4-8', '66666');
INSERT INTO `t_user` VALUES (23, '13478565555', '/images/avatar/monkey.png', '123456', '13478565555', '男', NULL, '加油');
INSERT INTO `t_user` VALUES (25, '13877777766', '/images/avatar/monkey.png', '123456', '13877777766', '男', NULL, '生而无畏');

```

```

INSERT INTO `t_user` VALUES (29, '13482638273', '/images/avatar/monkey.png', '123456', '13482638273', '女', NULL, '向前跑，用尽全部力气');
INSERT INTO `t_user` VALUES (30, '13428372632', '/images/avatar/monkey.png', '123456', '13428372632', '女', NULL, '生活不止眼前的枸杞，还有诗和远方');
INSERT INTO `t_user` VALUES (35, '13457555444', '/images/avatar/monkey.png', '123456', '13457555444', '男', NULL, 'to be');
INSERT INTO `t_user` VALUES (36, '13415656565', '/images/avatar/monkey.png', '123456', '13415656565', '女', NULL, 'Oh');
INSERT INTO `t_user` VALUES (37, '13428273363', '/images/avatar/monkey.png', '123456', '13428273363', '女', NULL, 'JavaScript');
INSERT INTO `t_user` VALUES (44, '小李', '/images/avatar/monkey.png', '123456', '13414232321', '男', NULL, '一起加油');
INSERT INTO `t_user` VALUES (46, '王大锤', '/images/avatar/1557241132762.jpg', '123456', '13412121211', '男', '1990-06-16', '开心就好');
INSERT INTO `t_user` VALUES (48, '大奔', '/images/avatar/monkey.png', '123456', '13414850215', '男', '2019-5-2', '777777');
INSERT INTO `t_user` VALUES (49, '13414851033', '/images/avatar/monkey.png', '123456', '13414851033', '女', NULL, '99999');
INSERT INTO `t_user` VALUES (50, '13672606065', '/images/avatar/monkey.png', '999999', '13672606065', '男', '', '7777777');
INSERT INTO `t_user` VALUES (51, '王五', '/images/avatar/1557474974747.jpg', '123456', '13478343221', '男', '2019-5-1', '66666666666666666666');
INSERT INTO `t_user` VALUES (52, '老王', '/images/avatar/monkey.png', '123', '13414850242', '女', '2019-7-8', '帅');
INSERT INTO `t_user` VALUES (53, '13411111111', '/images/avatar/monkey.png', '123456', '13411111111', NULL, NULL, NULL);
INSERT INTO `t_user` VALUES (54, '13499090909', '/images/avatar/monkey.png', '123456', '13499090909', NULL, NULL, NULL);

```

```
-- -----
```

```
-- Table structure for t_wishmovie
```

```
-- -----
```

```
DROP TABLE IF EXISTS `t_wishmovie`;
```

```
CREATE TABLE `t_wishmovie` (
```

```
  `wishMovie_id` int(4) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '想看电影 id',
```

```
  `user_id` int(4) UNSIGNED NULL DEFAULT NULL COMMENT '用户 id',
```

```
  `movie_id` int(4) UNSIGNED NULL DEFAULT NULL COMMENT '电影 id',
```



```

PRIMARY KEY (`wishMovie_id`) USING BTREE,
INDEX `t_wishmovie_ibfk_1` (`user_id`) USING BTREE,
INDEX `t_wishmovie_ibfk_2` (`movie_id`) USING BTREE,
CONSTRAINT `t_wishmovie_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `t
_user` (`user_id`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `t_wishmovie_ibfk_2` FOREIGN KEY (`movie_id`) REFERENCES `
t_movie` (`movie_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE = InnoDB AUTO_INCREMENT = 118 CHARACTER SET = utf8 COLLATE = u
tf8_unicode_ci ROW_FORMAT = Dynamic;

```

```

-- -----
-- Records of t_wishmovie
-- -----
INSERT INTO `t_wishmovie` VALUES (74, 1, 20);
INSERT INTO `t_wishmovie` VALUES (76, 1, 9);
INSERT INTO `t_wishmovie` VALUES (77, 46, 13);
INSERT INTO `t_wishmovie` VALUES (78, 1, 17);
INSERT INTO `t_wishmovie` VALUES (79, 1, 33);
INSERT INTO `t_wishmovie` VALUES (82, 50, 13);
INSERT INTO `t_wishmovie` VALUES (83, 50, 5);
INSERT INTO `t_wishmovie` VALUES (84, 50, 8);
INSERT INTO `t_wishmovie` VALUES (85, 50, 29);
INSERT INTO `t_wishmovie` VALUES (86, 50, 12);
INSERT INTO `t_wishmovie` VALUES (87, 50, 6);
INSERT INTO `t_wishmovie` VALUES (88, 25, 20);
INSERT INTO `t_wishmovie` VALUES (89, 25, 29);
INSERT INTO `t_wishmovie` VALUES (90, 25, 34);
INSERT INTO `t_wishmovie` VALUES (91, 29, 29);
INSERT INTO `t_wishmovie` VALUES (92, 29, 9);
INSERT INTO `t_wishmovie` VALUES (93, 29, 34);
INSERT INTO `t_wishmovie` VALUES (95, 23, 3);
INSERT INTO `t_wishmovie` VALUES (96, 1, 35);
INSERT INTO `t_wishmovie` VALUES (110, 1, 15);
INSERT INTO `t_wishmovie` VALUES (112, 1, 13);
INSERT INTO `t_wishmovie` VALUES (115, 1, 12);
INSERT INTO `t_wishmovie` VALUES (116, 1, 8);
INSERT INTO `t_wishmovie` VALUES (117, 1, 7);
INSERT INTO `t_wishmovie` VALUES (118, 1, 10);

```

```

SET FOREIGN_KEY_CHECKS = 1;

```