****

**Bilibili数据库系统设计**

[数据库实践项目报告]

指导教师：陈世展

[2021.10.30]

目录

[一. 需求分析 3](#_Toc1669)

[1.1 项目介绍 3](#_Toc17887)

[1.2 项目需求分析 3](#_Toc18180)

[1.3 项目重点概述 3](#_Toc22694)

[1.4 项目过程分析 4](#_Toc11390)

[1.4.1 存储设计 4](#_Toc32498)

[1.4.2 爬取数据 4](#_Toc9492)

[1.4.3 前后端开发 4](#_Toc993)

[二. 功能实现 4](#_Toc14030)

[3.1 百大UP主的视频数据分析 4](#_Toc10041)

[3.2 各个专区的活跃数据分析 4](#_Toc5325)

[3.3 特定专区热门视频的数据分析 4](#_Toc16453)

[3.4 特定百大UP主视频的数据分析 4](#_Toc3893)

[3.5 查看出现在分析列表中的视频的详细信息 4](#_Toc25742)

[3.6 UP主的粉丝数据分析 5](#_Toc15225)

[3.7 对专栏、番剧区的数据分析 5](#_Toc6533)

[3.8 对评论区的数据分析 5](#_Toc24551)

[三. 数据库表设计 5](#_Toc4187)

[3.1 数据库表设计文档 5](#_Toc17573)

[3.2 数据库表设计ER图 5](#_Toc9611)

[3.3 用户表详解 6](#_Toc23930)

[3.4 视频表详解 6](#_Toc6948)

[四. 爬虫接口设计 6](#_Toc26987)

[4.1 返回up的视频集合 6](#_Toc26626)

[4.2 获取UP主的前250位粉丝 7](#_Toc17194)

[4.3 获取一个账号的卡片信息 7](#_Toc13143)

[4.4 获取一个专区的视频 7](#_Toc2709)

[五. 前后端接口设计 7](#_Toc18299)

[5.1 前后端接口说明 7](#_Toc23325)

[5.2 前后端接口文档 7](#_Toc19116)

[六. 心得体会 8](#_Toc8270)

**一．需求分析**

* 1. **项目介绍**

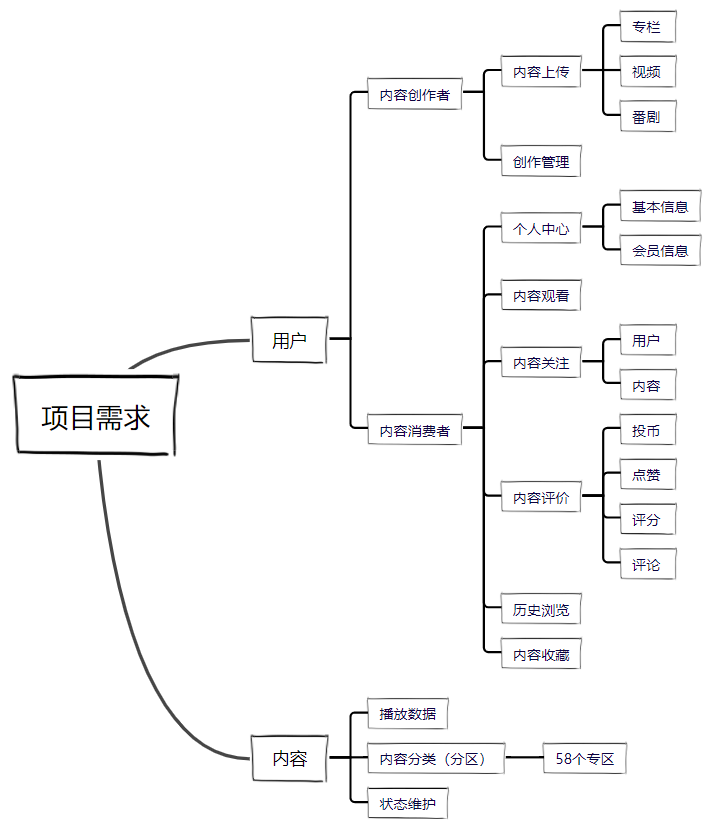
哔哩哔哩（英文名称：Bilibili，简称B站），一个主打内容创作与分享的视频网站，是年轻人拥抱多元文化的娱乐社区。项目围绕消费者、创作者和内容三方面，构建一个源源不断产生优质内容的生态系统，拥有数据分析能力。

* 1. **项目需求分析**

Bilibili作为内容创作与分享网站，需要管理内容、服务用户。用户可分为内容创作者与内容消费者，用户可以同时拥有着两种身份，并不冲突。本此项目对b站数据库的结构分析主要在于内容生态系统。

用户模块包含用户个人信息管理、创作中心，以及大会员增值服务。

视频内容细化分区为拥有不同爱好的用户提供了良好的优质内容获取渠道，区内排名使得小众分区也能良性发展。除了排行榜外，热门功能中还有每周必看、入站必刷等精品内容集合；收藏夹主要为用户想要二次观看进行收藏过操作的内容，方便用户观看已收藏内容。



* 1. **项目重点概述**

1. 分析bilibili视频网站数据库需求、数据、逻辑结构。
2. 通过数据库设计与构建，爬取并存储部分bilibili网站后台的数据。
3. 使用存储在本地的bilibili后台数据，开发出一个前端系统用于数据分析。
   1. **项目过程分析**
      1. **存储设计**

根据bilibili网页版所呈现的数据，对其数据库进行分析整理，并结合实际用户需求进行补充，最终设计出对应数据表、关系表以及触发器。

项目使用MYSQL存储数据。

* + 1. **爬取数据**

需要根据bilibili公司暴露出来的后台接口，编写对应的爬虫程序，将数据爬取为csv格式，之后再将csv格式的数据录入设计好的数据表中。

* + 1. **前后端开发**

**后端开发**：

使用Node.js，框架为express；同时使用自行开发的mysql-builder-plus库进行数据库的连接与异步查询操作。

**前后端文档**：

使用Notion进行接口文档的管理，前后端开发可以由不同同学异步完成。

**前端开发**：

使用Vue.js 3.0进行开发，同时使用了axios库以调用后端提供的接口。

**数据展示**：

数据展示则通过ECharts完成，其最大的优点就是可以动态、交互式地显示数据。

**二．功能设计**

* 1. **百大UP主的视频数据分析**

使用**条形-折线统计图**进行展示，可以方便观察与对比各个百大UP的视频热度、视频质量与用户粘性。

* 1. **各个专区的活跃数据分析**

使用**散点图**进行展示，可以观察不同专区的活跃程度、用户喜好程度与视频质量。

* 1. **特定专区热门视频的数据分析**

使用**散点图**进行展示，可以观察不同专区的具体采样视频情况与一些热门视频的不同之处。

* 1. **特定百大UP主视频的数据分析**

通过对百大UP主的历史视频挖掘，可以观察到UP主大约火起来的时间段与特定的视频。

* 1. **查看出现在分析列表中的视频的详细信息**

鉴于ECharts的可交互性，我们可以通过点击任何一个出现在分析图表中的视频来查看其详细数据。

* 1. **UP主的粉丝数据分析（待完成）**

通过采样对UP主的粉丝平均等级与平均大会员率等参数信息，可以对up主粉丝质量进行评估，能以判断UP主是否购买了流量服务或进行了暗中造假。

* 1. **对专栏、番剧区的数据分析（待完成）**

使用与视频相同的分析模式，我们也可以对专栏、番剧区甚至动态进行相应的分析与结果展示。

* 1. **对评论区的数据分析（待完成）**

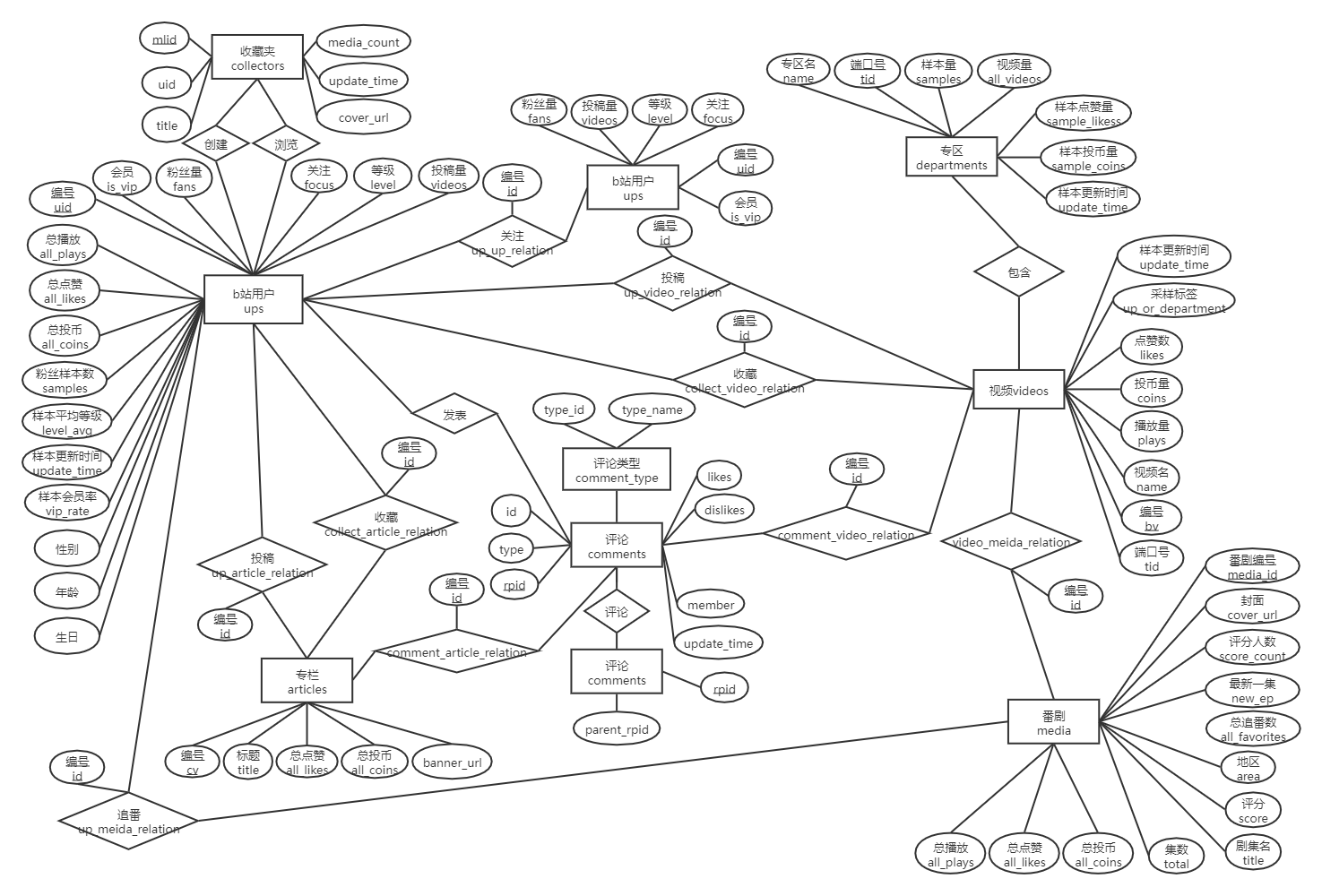
通过对评论区进行分析，可以增强视频表、UP主表的材料强度，也可以通过一个用户的评论分析来判断其在网络上的习惯与性格。

**三．数据库表设计**

**3.1 数据库表设计文档**

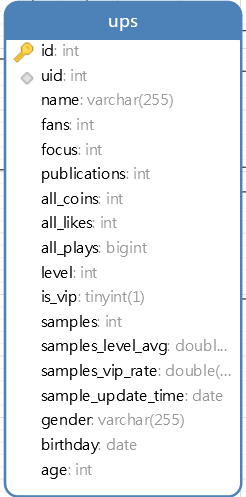
（详见附件《数据表设计文档》）

**3.2 数据库表设计ER图**



（高清版本详见附件）

**3.3 用户表详解**

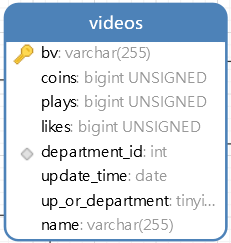
用户表主要由个人基础信息（生日 性别 年龄等）、投稿信息（投稿量 播放量等）、会员信息、粉丝质量评估参数（粉丝平均等级 粉丝会员率等）四大部分构成。

由于用户之间可以互相关注，因此额外制作了关注关系表，起名为up\_up\_sample\_relation。在获取数据过程中，由于B站用户数据众多，无法全部通过爬虫获取，故只对百大的粉丝进行了数据采样。

设定用户表的前100个用户为百大，方便在数据分析功能中操作百大的视频与粉丝数据。(百大为B站官方每年评选出的100位杰出UP主)。

用户表中拥有几个采样所得的表项，这些数据都可以先通过关系表再通过遍历粉丝获得，经过权衡查找与存储的代价后，选择了存储这些数据。

**3.4 视频表详解**

视频表由视频编号、视频所属专区、投币点赞播放量等字段构成。每个视频都有所属专区（共58个专区），故设置外键链向专区表。

考虑到b站用户可以对任意版块的信息发表评论，包括视频、专栏以及番剧等等，因此我们创建了comment表，以comment\_type表存储评论对象的类型，并且使用外键管理评论的对象；同时对每一种评论与对象的关系创建了关系表，如评论-专栏表、评论-视频表等等。

由于评论自身也可以是嵌套的，我们设计了一个表自身的外键parent\_rpid用于处理无限嵌套问题；对此我们同样设计了一个触发器。

**四．爬虫接口设计**

**4.1 返回up的视频集合**

需要使用页码、页数来请求

每个up的投稿总数需要在这里取得

python3 xxx.py --action=getVideoByUp --up=upIdExample --pn=页码 --ps=每页项数

**4.2 获取UP主的前250位粉丝**

返回一个数组，result[1]为pn=1的查询结果(前50个)，result=2为pn=2的查询结果(50-100个)，以此类推

python3 xxx.py --action=getFans --up=upIdExample

**4.3 获取一个账号的卡片信息**

并没有**archive\_count**字段，也就是这里并不会返回用户的总投稿，我们需要通过调用getVideoByUp获取用户的总投稿

python3 xxx.py --action=getUpInfo --up=upIdExample

**4.4 获取一个专区的视频**

需要使用页码、页数来请求

python3 xxx.py --action=getVideoByDepartment --tid=tidExample --pn=页码 --ps=每页项数

**五．前后端接口设计**

**5.1 前后端接口说明**

1. 数据api统一使用JSON格式
2. 前端渲染
3. node、vue分离

**5.2 前后端接口文档**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 说明 | 方法 | 种类 | 状态 | url |
| 获取所有百大的统计数据 | get | 用户接口 | 可用 | /ups100 |
| 获取所有专区的统计数据 | get | 用户接口 | 可用 | /departments |
| 获取给定UP的所有视频的统计数据 | get | 用户接口 | 可用 | /ups/videos?uid= |
| 获取一个专区的热门视频统计信息 | get | 用户接口 | 可用 | getVideoByDepartment  ?tid=x |
| 获取一个专区的具体信息 | get | 用户接口 | 可用 | /getDepartmentBy  DepartmentID?tid=x |
| 获取一个up主的具体信息 | get | 用户接口 | 可用, 废弃 | /getUserByID?id=x |
| 获取特定视频的up主的具体信息 | get | 用户接口 | 可用, 废弃 | /getUserByBV?bv=x |

**六．心得体会**

世上无难事 ，只怕有心人。

通过参加这次数据库实践项目，我们都有了很多收获。首先是对实现目的的进一步认识，我们充分发挥了自主性、探索性、实践性与协作性，从开始项目的命题、需求分析到程序编写，我们的思维能力、团队协作能力、实践能力都得到了提高，最后做出了完整的数据库系统设计项目。

在项目答辩中和老师交流后，认识到结构设计有一定不足，没有考虑到用户给单集视频评论与给剧集评论的区别。未来需要将重心更多放在逻辑关系与实际用户需求上，现成的bilibili网站数据库设计并不完全是合理的，需要我们从用户出发更多考虑实际情况。