总结：实体和实体之间，若是n:n的关系，就说明要创建三个表，实体表之间独立，行为单独创一个表；实体和实体之间，若是n:1的关系，就创建两个表，n方的实体包含1方的主键ID；若1:1的关系，还没碰到这种情况。



主页展示视频、收藏夹…订阅；投稿就是按时间顺序展示视频；合集按照视频上传者的分类设置展示视频（用户和创建合集是一对n的关系，合集中直接包含创建者用户的ID；而合集和视频又是一对n的关系，所以视频包含合集的ID）；收藏按则是按照收藏夹分类展示(用户和收藏夹1对n，所以收藏夹包含创建用户ID；收藏夹和视频是n：n的，所以新建一个表)；订阅是一种行为，我的理解是订阅合集（合集和订阅用户的关系是n对n的关系，一个合集可以被多个用户订阅，一个用户也可以订阅多个合集，所以用户和合集之间还需要创建一个订阅关联表）。

关注数和粉丝数由新表维护；视频点赞数和用户获赞数由新表维护；播放数由历史记录表维护（播放时长>10s有效）。



需要考虑的是，当文件数量很多时，头像的索引效率会下降。因此，需要设计算法来对用户头像的路径进行划分。

1. user

|  |  |  |
| --- | --- | --- |
| id | bigint | 雪花算法生成ID |
| username | varchar(32) | 手机号码（唯一索引） |
| nickname | varchar(32) | 昵称 |
| password | char(32) | 加密后的密码 |
| email | varchar(30) | 邮箱 |
| tags | varchar(120) | 标签，便于推荐 |
| image\_id | bigint | 用户头像Id |
| signature | varchar(50) | 个性签名 |
| create\_time | dateTime | 创建时间 |
| role | shortInt | 0默认值代表普通用户，1代表管理员 |
| status | shortInt | 0默认值代表激活状态，1代表注销状态 |

1. fan：用户和用户之间n：n的关系，所以需要创建新表

|  |  |  |
| --- | --- | --- |
| id | bigint | id |
| user\_id | bigint | 被关注者 |
| fan\_id | bigint | 关注者 |
| create\_time | datetime | 关注时间 |



文件系统路径进一步细分，电影、番剧和娱乐视频为一层索引，年为一层索引，月为一层索引，又为一层索引。文件系统中只存MP4文件，在服务器中进行格式转换后再上传至文件系统。

1. video

|  |  |  |
| --- | --- | --- |
| id | bigint | 雪花ID |
| series\_id | bigint | 合集ID |
| md5 | char(32) | MD5校验，实现秒传 |
| title | varchar(30) | 视频标题 |
| description | varchar(255) | 视频描述（简介） |
| tags | varchar(120) | JSON存储标签 |
| video\_path | varchar(255) | 视频存储路径 |
| video\_suffix | varchar(10) | 视频后缀 |
| image\_id | bigint | 封面名称 |
| create\_time | DateTime | 上传时间 |
| size | bigint | 视频文件的大小 |

1. series (合集)

用户可以创建合集，电视剧、动漫…

|  |  |  |
| --- | --- | --- |
| id | bigint | 雪花算法生成ID |
| user\_id | bigint | 创建合集的用户ID |
| image\_id | bigint | 合集封面 |
| title | varchar(20) | 合集标题 |
| description | varchar(255) | 简介 |
| create\_time | datetime | 创建时间 |

1. user-series（用户-合集）

|  |  |  |
| --- | --- | --- |
| id | bigint | 雪花算法生成ID |
| user\_id | bigint |  |
| series\_id | bigint |  |
| create\_time | datetime | 订阅时间 |

1. store

|  |  |  |
| --- | --- | --- |
| id | bigint | 雪花算法生成ID |
| user\_id | bigint | 创建收藏夹的用户ID |
| image\_id | bigint | 封面ID |
| description | varchar(256) | 简介 |
| create\_time | datetime | 创建时间 |

1. user\_store\_video

|  |  |  |
| --- | --- | --- |
| id | bigint | 雪花算法生成ID |
| store\_id | bigint | 收藏夹的ID |
| video\_id | bigint | 视频的ID |
| create\_time | datetime | 收藏时间 |

1. appreciation（点赞）

|  |  |  |
| --- | --- | --- |
| id | bigint | 雪花算法生成ID |
| user\_id | bigint |  |
| video\_id | bigint |  |
| create\_time | datetime | 点赞时间 |

历史记录播放初步解决方案：

1. history

|  |  |  |
| --- | --- | --- |
| id | bigint | 雪花算法生成ID |
| user\_id | bigint |  |
| video\_id | bigint |  |
| create\_time | datetime | 观看退出时时间 |
| end\_point | bigint | 视频播放终止时间(ms) |

视频和弹幕，一对N，用户和弹幕，一对N

1. barrage

|  |  |  |
| --- | --- | --- |
| id | bigint | 雪花算法生成ID |
| user\_id | bigint |  |
| video\_id | bigint |  |
| content | varchar(30) | 弹幕内容 |
| create\_time | datetime | 弹幕发送时间 |

1. comment

评论这个功能，说白了就是对别人评价，因此slave为空的评论就是主评论，而之后链表式连接的就是子评论。

被评论用户(1) -评论用户（1）-评论（n）-视频（1）

|  |  |  |
| --- | --- | --- |
| id | bigint | 雪花算法生成ID |
| slave\_user\_id | bigint | 被评价的ID |
| master\_user\_id | bigint | 评价的ID |
| video\_id | bigint |  |
| content | text | 文本类型？还是使用mongodb不了解，不确定待实现功能 |
| create\_time | datetime | 评论时间 |

1. image

|  |  |  |
| --- | --- | --- |
| id | bigint | 雪花算法生成ID |
| image\_path | varchar(128) | 图片存储路径 |
| create\_time | datetime |  |