1. 以健康为主题的社交分享及大数据可视化分析平台的设计与实现

随着社会的快速发展，一切都在进步的同时，越来越多的健康问题也随之发生，健康的生活成为了人们日常生活中一个越来越重要的话题，保持身体和精神的健康，不受疾病的困扰已被越来越多的人所关注，所以我们想打造一个平台，给人们提供与健康相关的信息，为人们提供一个健康相关的分享和交流的平台，以及对与健康相关的数据进行收集和分析，并以可视化的界面呈现给用户，为用户提供一定的参考，引导用户走向健康之路。

二.数据库设计

1. 用户账号设计

(1)个人资料：id递增，账号(电话),密码，头像文件地址，昵称，地址，联系方式等等

(2)账户资料：作品数量，粉丝，好友，账号访问次数，积分，等级，排名，VIP等级，举报，账户状态是否可用，等等。

(3)个人动态：发布的文章视频，关注的人，收藏的文章等等。

实现更新个人信息，发布文章视频同时更新文章视频表。设计触发器，发布文章积分增加，积分到了自动升级等等。

1. 文章管理

文章管理包括用户发布的文章，我们官方发布的文章等

包括文章id(递增)，根据URL+id可搜索到该行记录，用户id，文章标题，内容，标签，浏览人数，评论人数，喜欢人数，最后更新时间，评论id（根据id查找更新评论内容并显示出来），等等。设计相应的触发器，插入文章时自动计算文章数量。实现CRUD操作，文章模糊查询，根据标题，内容，标签等模糊查询可查出所有相关文章，根据评论的id查找到评论表里边所对应的内容。

1. 视频发布管理

包括视频id，发布的用户id，视频所在的服务器的文件夹的地址，标题，内容，标签，浏览人数，点赞人数，发布或者是最后更新的时间，评论id等等。实现CRUD操作，触发器，模糊查询，实现和文章管理类似的功能等等。

1. 用户评论管理

包括评论id，评论的用户的id（根据id查找作者，显示作者资料），评论的内容，时间，有图片的话还需要图片的文件地址，赞踩数量。实现CRUD操作等。

1. 交互信息管理

该部分记录用户，群之间的交互信息，所涉及的字段有，消息id，发出消息用户id，接收消息用户id，消息的内容，发出消息的最后时间等。

1. 问题管理

用户可进行提问，该部分主包括问答的用户id，问答题目，问答内容，问答时间，问题的状态（是否解决），解决者的id，解决者可得到积分。需进行CRUD操作等。

1. 回答管理

该表保存问题的id，问答者的id，问答的问题的id，问答的标题，问答的内容，回答的时间等，以及是否被采纳的状态，点赞数量等。需进行CRUD操作等。

1. 兴趣小组管理

用户可以申请创建一个小组，方便志同道合的人一起交流

该部分有以下几个部分，创建人id，创建时间，标语，简介，公告，status是否被审批，申请该组的用户，允许加入的用户，该兴趣小组发布的文章id，视频id等等。功能可模糊查询该组的文章视频等，以及该组的特定成员等。

9.系统管理

包括，对个其他内容的CRUD操作，管理用户，文章评论，小组等的审核，记录发布文章，评论的id，根据id查找到后进行模糊查询，根据排除敏感关键字进行审核。

10.大数据分析可视化(1)

爬取相关数据进行查找整理存储和分析，以可视化的界面呈献给用户，给予用户一定的参考。

例：1.药品分析

根据正则表达式筛取感冒关键词相关的数据，并对数据进行爬取，记录的字段有

药品名称，平均价格，有效期，药品类型，关注度排名，用户的综合评价（点赞数量），药品的相关连接等，如果用户有兴趣可以查看。基于客户的不同请求，根据数据库中的数据进行查询，以折线图，扇形图等呈现给用户，用户可查看不同的排序结果。

2.心里健康分析

3.运动情况分析

…………

一、数据库命名规范，养成良好习惯

采用26个英文字母(区分大小写)和0-9的自然数(经常不需要)加上下划线'\_'组成，命名简洁明确，多个单词用下划线'\_'分隔，一个项目一个数据库，多个项目慎用同一个数据库

二、数据库表命名规范

2.1数据表命名规范

（1）采用26个英文字母(区分大小写)和0-9的自然数(经常不需要)加上下划线'\_'组成，命名简洁明确，多个单词用下划线'\_'分隔

（2）全部小写命名，禁止出现大写

（3）禁止使用数据库关键字，如：name，time ，datetime，password等

（4）表名称不应该取得太长（一般不超过三个英文单词）

（5）表的名称一般使用名词或者动宾短语

（6）用单数形式表示名称，例如，使用 employee，而不是 employees

明细表的名称为：主表的名称+字符dtl（detail缩写）

例如：采购定单的名称为：po\_order，则采购定单的明细表为：po\_orderdtl

（7）表必须填写描述信息（使用SQL语句建表时）

2.2命名规范

①模块\_+功能点 示例：alllive\_log alllive\_category

②功能点 示例：live message

③通用表 示例：all\_user

2.3待优化命名示例

①冗余：

错误示例：yy\_alllive\_video\_recomment yy\_alllive\_open\_close\_log

说明：去除项目名，简化表名长度，去”yy\_”

②相同类别表命名存在差异，管理性差

错误示例：yy\_all\_live\_category yy\_alllive\_comment\_user

说明：去除项目名，统一命名规则，均为”yy\_alllive\_”开头即可

③命名格式存在差异

错误示例：yy\_showfriend yy\_user\_getpoints yy\_live\_program\_get

说明：去除项目名，统一命名规则，动宾短语分离且动宾逻辑顺序统一

三、数据库字段命名规范

3.1字段命名规范

（1）采用26个英文字母(区分大小写)和0-9的自然数(经常不需要)加上下划线'\_'组成，命名简洁明确，多个单词用下划线'\_'分隔

（2）全部小写命名，禁止出现大写

（3）字段必须填写描述信息

（4）禁止使用数据库关键字，如：name，time ，datetime password 等

（5）字段名称一般采用名词或动宾短语

（6）采用字段的名称必须是易于理解，一般不超过三个英文单词

（7）在命名表的列时，不要重复表的名称

例如，在名employe的表中避免使用名为employee\_lastname的字段

（8）不要在列的名称中包含数据类型

（9）字段命名使用完整名称，禁止缩写

3.2命名规范

①名词 示例：user\_id user\_name sex

②动宾短语 示例：is\_friend is\_good

3.3待优化命名示例

①大小写规则不统一

错误示例：user\_id houseID

说明：使用统一规则，修改为”user\_id”，”house\_id”

②加下划线规则不统一

错误示例：username userid isfriend isgood

说明：使用下划线进行分类，提升可性，方便管理，修改为”user\_name”，”user\_id”，”is\_friend”，”is\_good”

③字段表示不明确

错误示例：uid pid

说明：使用完整名称，提高可读性，修改为”user\_id”，”person\_id”

3.4字段类型规范

（1）所有字段在设计时，除以下数据类型timestamp、image、datetime、smalldatetime、uniqueidentifier、binary、sql\_variant、binary 、varbinary外，必须有默认值，字符型的默认值为一个空字符值串’’，数值型的默认值为数值0，逻辑型的默认值为数值0

（2）系统中所有逻辑型中数值0表示为“假”，数值1表示为“真”，datetime、smalldatetime类型的字段没有默认值，必须为NULL

（3）用尽量少的存储空间来存储一个字段的数据

使用int就不要使用varchar、char，

用varchar(16)就不要使varchar(256)

IP地址使用int类型

固定长度的类型最好使用char，例如：邮编(postcode)

能使用tinyint就不要使用smallint，int

最好给每个字段一个默认值，最好不能为null

（4）用合适的字段类型节约空间

字符转化为数字(能转化的最好转化，同样节约空间、提高查询性能)

避免使用NULL字段(NULL字段很难查询优化、NULL字段的索引需要额外空间、NULL字段的复合索引无效)

少用text类型(尽量使用varchar代替text字段)

3.5数据库中每个字段的规范描述

（1）尽量遵守第三范式的标准（3NF）

表内的每一个值只能被表达一次

表内的每一行都应当被唯一的标示

表内不应该存储依赖于其他键的非键信息

(2)如果字段事实上是与其它表的关键字相关联而未设计为外键引用，需建索引

(3)如果字段与其它表的字段相关联，需建索引

(4)如果字段需做模糊查询之外的条件查询，需建索引

(5)除了主关键字允许建立簇索引外，其它字段所建索引必须为非簇索引

四、SQL语言编码规范

4.1大小写规范

（1）所有关键字必须大写，如：INSERT、UPDATE、DELETE、SELECT及其子句，IF……ELSE、CASE、DECLARE等

（2）所有函数及其参数中除用户变量以外的部分必须大写

（3）在定义变量时用到的数据类型必须小写

4.2注释

注释可以包含在批处理中，在触发器、存储过程中包含描述性注释将大大增加文本的可读性和可维护性，本规范建议：

（1）注释以英文为主，实际应用中，发现以中文注释的SQL语句版本在英文环境中不可用，为避免后续版本执行过程中发生某些异常错误，建议使用英文注释

（2）注释尽可能详细、全面创建每一数据对象前，应具体描述该对象的功能和用途，传入参数的含义应该有所说明，如果取值范围确定，也应该一并说明，取值有特定含义的变量（如boolean类型变量），应给出每个值的含义

（3）注释语法：单行注释、多行注释

单行注释：注释前有两个连字符（--）对变量、条件子句可以采用该类注释

多行注释：符号之间的内容为注释内容，对某项完整的操作建议使用该类注释

（4）注释简洁，同时应描述清晰

（5）函数注释：

编写函数文本--如触发器、存储过程以及其他数据对象--时，必须为每个函数增加适当注释，该注释以多行注释为主，主要结构如下：

CREATE PROCEDURE sp\_xxx