

# 软件工程：分析设计文档

指导老师:白晓颖

伍一鸣 2012011347 陈立康 2012011304 梁明 2012012439

## 一.项目概述

### 1.1 人人贷数据业务报表平台项目背景

随着企业的发展以及业务规模的增长,数据已经成为企业核心价值的一部分。数据可以反映企业的经营现状、用于分析策划下一步战略、并帮助管理者做出决策。如果将管理者比作大海中航行的舵手,数据就是其手中的六分仪。而数据缺失或统计分析错误,就好像舵手拿了一个损坏的六分仪,非常容易偏离正确航线甚至越走越远。

作为与互联网有关的企业,原始数据来源于用户的访问和行为。经过一定的处理和抽取,有用的数据被导入数据库,并经过进一步详细的统计和分析,形成基于各种维度、展示不同内容的报表。这些报表可能是需要时临时生成的,也可能是每天或每月定时分析生成,也有依赖于其它报表再次计算所得。

在报表数量不大的时候,这些报表可以是业务分析人员与数据管理人员共同维护,并设定一定数量的定时任务,附加手动数据库查询生成。但随着业务量增大,报表数量增多,企业需要一个数据平台来管理和维护生成报表的工作:方便新报表的创建、现有报表的修改和维护、以及生成报表的任务的状态检查和任务失败后的重试等等工作。

### 1.2 功能需求

#### 1.2.1 前端 Web 页面

##### SQL 编辑器

在 Web 页面中实现 SQL 语言的编辑器,允许用户编辑 SQL 代码,并执行,并支持以下功能:

1. SQL 语法的高亮
2. 在原本的 SQL 语言基础上增加变量支持,要求支持时间变量,例如  $\${now}$ 、 $\${today}$  等,时间间隔变量  $\${时间 \pm 数字 sec/min/day/month/year}$
3. 用户编辑完成 SQL 命令之后,可以通过按钮立即执行编辑好的语句
4. 用户可以导入一个已经保存的 SQL 语言文件,或将当前编辑器中的语句导出到文件
5. (扩展) 代码补全: 针对 MySQL 语法以及实时数据库查询的库、表、列名进行补全提示
6. (扩展) 危险警告: 对 Delete、Update 等操作做出提示,对 Drop 等危险操作做出警告
7. (扩展) 语句分析: 对输入的 MySQL 语句进行解释分析,即 EXPLAIN 功能
8. (扩展) 用户可以在 Web 中保存语句列表,并可以在任务编辑时导入

## 数据分析任务管理系统

数据分析平台允许用户在 Web 页面中管理数据分析的任务。任务的描述中不应少于以下内容：

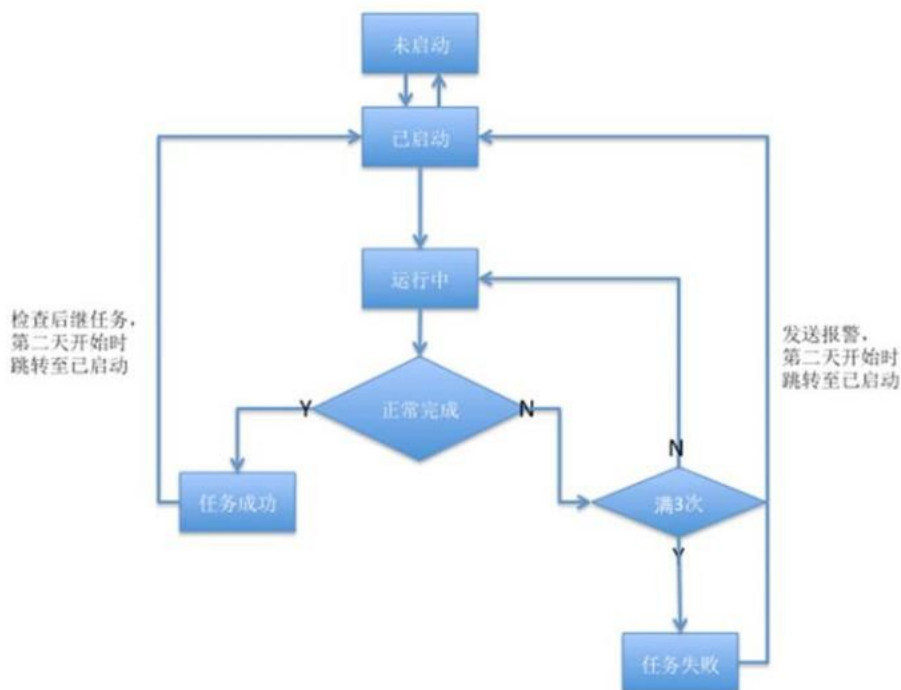
1. 任务名称

2. 任务描述: 任务的详细说明

3. SQL 语句: 任务实际执行的 SQL 语句

4. 任务类型: 包括两种, 普通类型或定时类型, 定时类型任务需要在规定的时间运行, 例如“每周日上午 8 点”, 定时任务周期最小单位是天

5. 前驱任务: 任务运行前需要完成的任务。定时类型不可以有前驱任务。数据分析平台中的任务可以保证所有单个任务都可以在一天完成, 任务的执行中会经历如下的状态图：



任务在被添加入系统后处于未启动状态, 可以由用户启动。启动后, 普通任务会在其前驱任务均执行完毕后被启动进入运行中状态, 定时任务会在指定的时间开始运行。若运行正常完成, 则进入任务成功状态。若未正常完成, 则根据失败次数判断是否重新执行。若失败小于 3 次, 则重新执行; 若连续 3 次失败, 则进入任务失败状态, 并发送警报。所有任务状态在新的一天开始的时候强制进入已启动状态。

在数据分析任务管理系统中, 用户可以完成以下需求:

1. 用户可以添加、删除、修改一个任务, 删除、修改的任务必须是不在运行中的任务, 并且不被正在运行中的任务依赖。添加任务时 SQL 语句可以从之前保存的列表中的导入

2. 用户可以查看系统中所有定义好的任务列表, 能以有向无环图的形式展现。可以显示出各个任务当前的状态 (未启动、已启动、运行中、运行失败)

3. 用户可以启动任务、终止任务。用户启动一个任务后, 若是普通任务, 则当前驱任务

完成时执行;若是定时任务,则在规定时间开始运行。用户终止任务后将任务标记为未启动

4. 用户可以停止正在执行的任务,并标记为任务失败;用户可以指定一个或若干个任务强制立即执行

5. 用户可以点击单个任务信息,查看任务信息,以及运行日志,运行前驱依赖,和后继

### **结果输出及展示**

数据分析平台允许用户选择数据库中的一张表,并输出报表或图标,具体需要支持以下功能:

1. 用户可以选择数据库中的一张表,在网页中预览数据(若表非常大,则预览部分数据)

2. 用户可以将表导出到 csv 文件

3. 用户可以由表的内容,指定列作为横纵坐标或分组标准,生成图表。至少应满足可以生成折线图/饼图/直方图,可以使用 Highcharts 等开源工具

4. (扩展) 用户可以将图表导出成图片

## **1.2.2 后台系统**

### **任务调度系统**

在数据分析平台的后台,需要能管理运行任务,具体需要包括以下功能:

1. 接收并启动手动触发任务的运行实例

2. 定时任务依据时间启动定时任务的运行实例

3. 子任务实例完成后检查任务的依赖条件并启动运行实例

4. 维护任务实例的状态转换关系

## **1.2.3 其它扩展功能**

### **权限认证**

\* 对不同的 web 登录用户设定权限,例如 MySQL 语句以及任务的添加、查看、执行、删除

\* 权限认证粒度扩展为每个语句和任务的添加、查看、执行、删除

## 二.工具使用

### 2.1 环境配置

#### 2.1.2 操作系统:

Ubuntu 15.10

#### 2.1.3 编程语言:

##### 2.1.3.1 语言 Python 2.7.10:

```
sudo apt-get install python
```

```
sudo apt-get install python-mysqldb
```

##### 2.1.3.2 Python 包管理工具 Python pip:

```
sudo apt-get install python-setuptools python-dev build-essential
```

```
sudo easy_install pip
```

```
sudo pip install --upgrade virtualenv
```

##### 2.1.3.3 定时任务模块 APScheduler:

```
sudo pip install apscheduler
```

#### 2.1.4 网站框架

##### Django1.8:

<https://www.djangoproject.com/download/>

下载完成,解压后在解压的目录下运行 `sudo python setup.py install`

#### 2.1.5 数据库:

##### MySQL 5.6.27:

```
sudo apt-get install mysql
```

## 2.2 工具介绍

### 2.2.1 前端平台

#### 2.2.1.2 网页工具

##### 1.d3.v3.min.js

D3.js 是基于数据驱动文档工作方式的一款 JavaScript 函数库,主要用于网页作图、生成互动图形,是最流行的可视化库之一,它被很多其他的表格插件所使用。D3 使你有能力借助 HTML,SVG 和 CSS 来生动地可视化各种数据。D3 不需要你使用某个特定的框架,它的重点在于对现代主流浏览器的兼容,同时结合了强大的可视化组件,以数据驱动的方式去操作 DOM。

## 2.dagre-d3.js

这是在 github 上找到的一个开源项目,它是基于 D3 库的.众所周知 D3 是一个专门用于前端图形绘制的库,dagre-d3 就是实现自动布局并且绘制流程图的这么一个东西.

## 3. prism.js

prism 是一个轻量级,可扩展的语法着色工具,符合 Web 标准。它压缩后只有 2KB,每种语言定义约增加 300-500 自己大小。非常易于使用,只需要插入一个 CSS 和 JS 文件即可。支持多种着色主题。

## 4.Highcharts

Highcharts 是一个用纯 JavaScript 编写的一个图表库,能够很简单便捷的在 web 网站或是 web 应用程序添加有交互性的图表,并且免费提供给个人学习、个人网站和非商业用途使用。HighCharts 支持的图表类型有曲线图、区域图、柱状图、饼状图、散状点图和综合图表。

## 2.3 数据库

根据项目的数据存储情况,以及助教的推荐,我们选择了 mysql 数据库作为我们后台存储用户数据、报表数据等数据的管理系统。

MySQL 是一个关系型数据库管理系统,由瑞典 MySQL AB 公司开发,目前属于 Oracle 旗下公司。MySQL 最流行的关系型数据库管理系统,在 WEB 应用方面 MySQL 是最好的 RDBMS (RelationalDatabase Management System,关系数据库管理系统) 应用软件之一。MySQL 是一种关联数据库管理系统,关联数据库将数据保存在不同的表中,而不是将所有数据放在一个大仓库内,这样就增加了速度并提高了灵活性。MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。

## 2.4 服务器

### 1.Python

Python, 是一种面向对象、解释型计算机程序设计语言。Python 强制用空白符(white space)作为语句缩进。Python 具有丰富和强大的库。它常被昵称为胶水语言,能够把用其他语言制作的各种模块(尤其是 C/C++) 很轻松地联结在一起。常见的一种应用情形是,使用 Python 快速生成程序的原型(有时甚至是程序的最终界面),然后对其中有特别要求的部分,用更合适的语言改写,比如 3D 游戏中的图形渲染模块,性能要求特别高,就可以用 C/C++ 重写,而后封装为 Python 可以调用的扩展类库。需要注意的是在您使用扩展类库时可能需要考虑平台问题,某些可能不提供跨平台的实现。

### 2.Django

Django 是一个开放源代码的 Web 应用框架,由 Python 写成。采用了 MVC 的软件设计模式,即模型 M,视图 V 和控制器 C。它最初是被开发来用于管理劳伦斯出版集团旗下的一些以新闻内容为主的网站的,即是 CMS(内容管理系统)软件。并于 2005 年 7 月在 BSD 许可证下发布。这套框架是以比利时的吉普赛爵士吉他手 Django Reinhardt 来命名的。

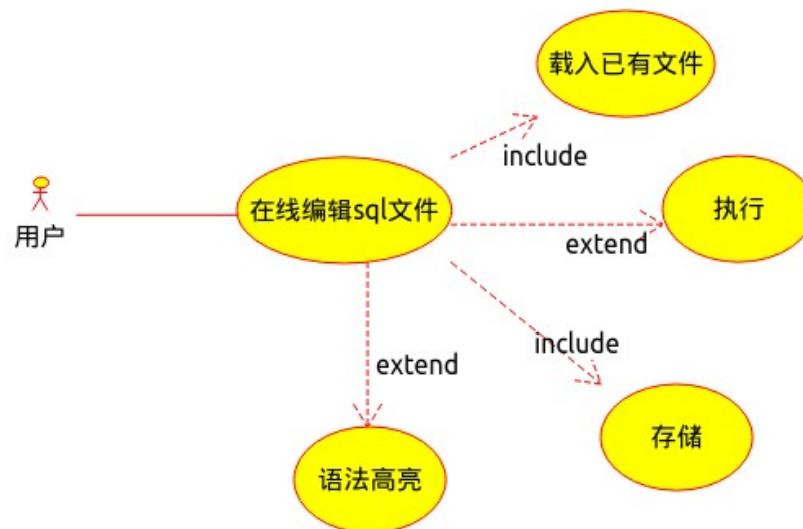
### **3.mysqlldb**

数据库的连接使用了 mysqlldb 库。MySQLdb 是用于 Python 链接 Mysql 数据库的接口，它实现了 Python 数据库 API 规范 V2.0，基于 MySQL C API 上建立的。



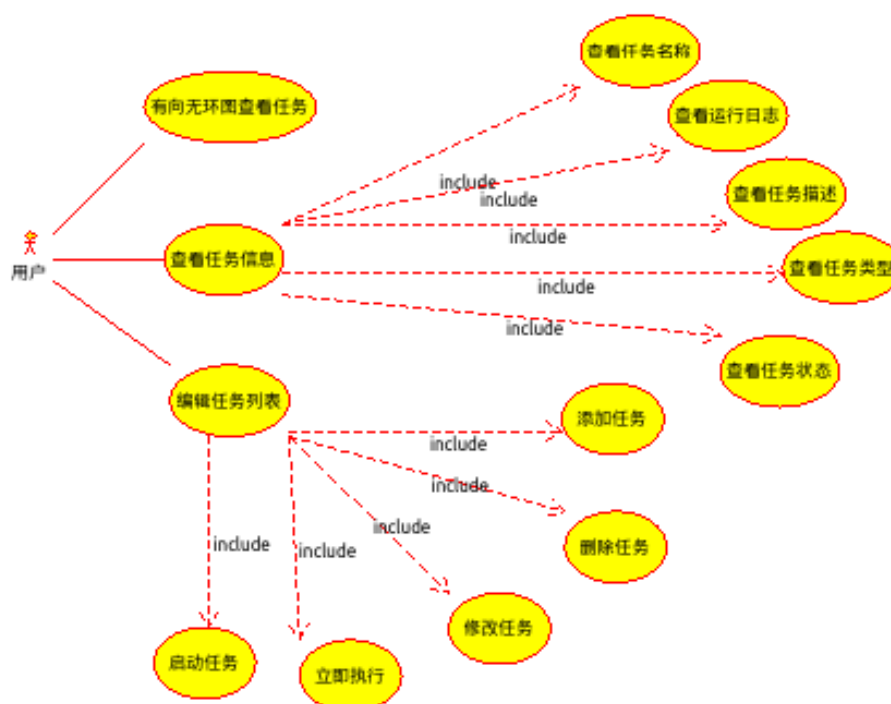
## 三.需求分析

### 3.1sql 编辑器



用户需要一个能够在线编辑的 sql 编辑器,并且支持语法高亮,载入已有文件,执行当前窗口的 sql 语句,存储当前页面的 sql 语句。

### 3.2 数据分析任务管理系统





### 3.2.1 有向无环图查看所有任务

根据用户提出的需求,页面中希望能有一种直观显示出所有任务前驱后继关系的图,以此来方便整体任务的管理,所以应当在任务管理系统中加入一副动态的有向无环图来显示所有任务的前驱后继关系。这个图之所以是有向无环的,是因为任务必须在其所有前驱任务都执行完成后才能开始执行,如果原任务关系中存在环,则这个环上所有任务都无法启动,因此用户需求的这个关系图必定是无环的,这在后续设计和实现上必须留意。

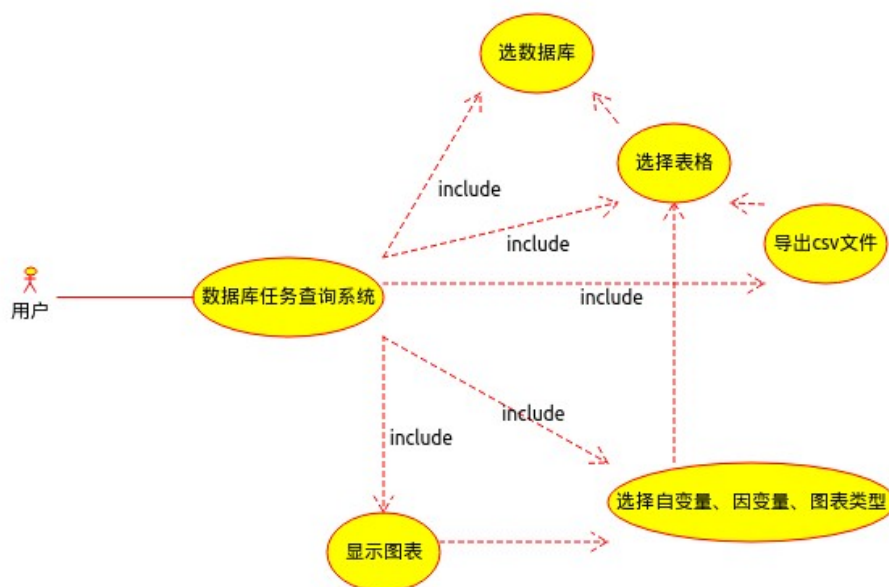
### 3.2.2 查看任务信息

根据用户提出的需求,页面中还希望有一种整齐划一的形式来显示出所有任务,因此可以选择利用列表形式来显示所有的任务的名称和运行状态。选中之后,会显示任务的详细信息,任务的详细信息还包括任务的描述、类型、SQL 文件和运行日志。

### 3.2.3 编辑任务列表

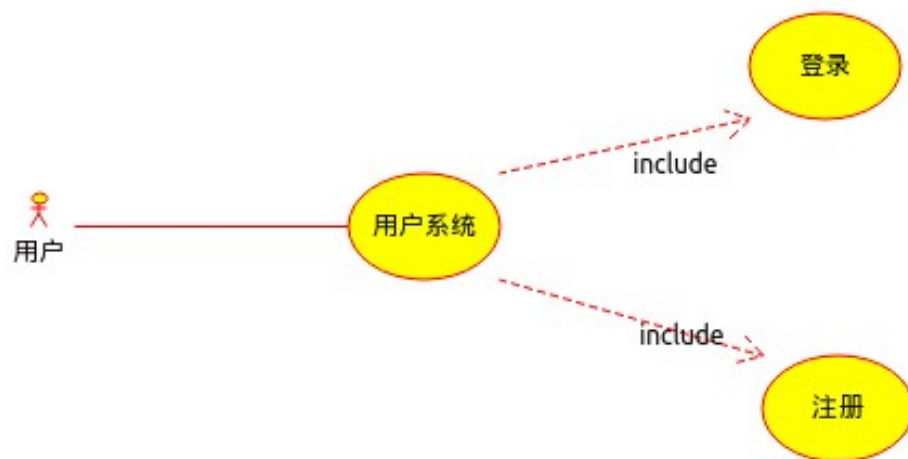
作为一个任务管理系统,所提供的最基本功能需求是对任务列表本身进行编辑。具体包括五个方面:添加任务、删除任务、修改任务、启动任务、立即执行以及终止任务。其中添加任务和删除任务的含义显而易见,而启动任务是指让任务状态从未启动转变到已启动,这是因为根据用户的需求,任务被划分为 5 个不同的阶段,分别是未启动、已启动、正在运行、运行成功和运行失败,任务刚被创建时默认为未启动。立即执行是让已启动任务不顾前驱后继立即变成正在运行状态。运行成功表示任务需要处理的数据已经完成。任务失败指任务因为各种原理不能正常执行完成。

## 3.3 数据库查询系统



用户可以在线查看后台数据库中某个表格的具体信息,并且根据信息的某一个特征通过在线图表显示出折线图、直方图、扇形图等。并且支持到处 csv 文件功能。

### 3.4 用户系统



用户系统支持注册新用户，和登录验证。

## 四.项目设计

### 4.1 所有前端页面

样式表和 html 全部纯手写，在需要判断输入是否为空的地方都加上 js 代码，进行错误提示。使用 `dagre-d3.js` 绘制任务关系图，使用 `prism.js` 实现语法高亮，使用 `Highcharts` 显示在数据库查询系统中显示图表。

### 4.2sql 编辑器

目标是实现一个能够在线编辑的 sql 编辑器,并且支持语法高亮，载入已有文件，执行当前窗口的 sql 语句，存储当前页面的 sql 语句。

初始时，会载入配置文件定下所选的数据库以及用户名和密码。

有存储操作时，会根据用户输入的名字，保存好。

有载入操作时，会查看当前目录下的所有 .sql 文件，并返回列表供用户选择。

有执行操作时，会先将所有的时间变量替换，然后再扔给数据库执行。

### 4.3 数据分析任务管理系统

#### 4.3.1 有向无环图查看所有任务

根据实际情况，任务之间的依赖关系一定是有向无环图。在实现是要注意判断。

从数据库中获取所有的任务的列表，和前驱后继的边，让 `dagre-d3.js` 来完成绘图工作。

#### 4.3.2 查看任务信息

列表形式来显示所有的任务的名称和运行状态。选中任务之后之后，会跳转显示出任务的详细信息，任务的详细信息还包括任务的描述、类型、SQL 文件和运行日志。

初始时会返回数据库中当前所有的任务的名称和状态。随后，根据选中的任务，跳转对应的页面，然后从数据库中读取详细的任务信息。

#### 4.3.3 编辑任务列表

作为一个任务管理系统,所提供的最基本功能需求是对任务列表本身进行编辑。具体包括五个方面：添加任务、删除任务、修改任务、启动任务、立即执行以及终止任务。其中添加任务和删除任务的含义显而易见,而启动任务是指让任务状态从未启动转变到已启动,这是因为根据用户的需求,任务被划分为 5 个不同的阶段,分别是未启动、已启动、正在运行、运行成功和运行失败,任务刚被创建时默认为未启动。立即执行是让已启动任务不顾前驱后继立即变成正在运行状态。运行成功表示任务需要处理的数据已经完成。任务失败指任务因为各种原理不能正常执行完成。

添加、删除、修改（包括前驱后继）任务的操作出现时，会调整相应的页面进行处理，然后在数据库中更新任务信息。

启动任务，立即执行，终止任务的操作出现时，按照相应的规则对任务的进程进行操作。操作之后注意更新数据库中的任务状态。

## 4.4 数据库查询系统

用户可以在线查看后台数据库中某个表格的具体信息,并且根据信息的某一个特征通过在线图表显示出折线图、直方图、扇形图等。并且支持到处 csv 文件功能。

初始时查询数据库中有多少张表单并返回给查询页面。选择 csv 功能时通过数据库本身的功能生成 csv 文件并提供下载。画图时，通过传到后台的参数，在数据库中查询出相应的数据。然后在通过 Highcharts 库来绘制对应的统计图表。

## 4.5 用户系统

用户系统支持注册新用户，和登录验证。

登录时，通过服务器和数据库交互，验证是否用户名密码都正确，并返回给前端网页。

注册时，注意通过服务器和数据库交互，判断是否注册成功，并返回给前端网页。

## 4.6 任务调度系统

### 4.6.2 系统初始化

每天系统第一次运行时执行初始化操作，将所有的任务状态改为已经启动。

### 4.6.2 定时任务获取及执行

后台调度系统需要及时获取当前需要执行的定时任务并执行,每次有任务的状态从未启动改为已经启动时开始将这个任务进行测试，看是否是可以开始执行的任务。系统对每个定时任务都单独开一个进程来单独运行,这样,定时查询可执行任务也可以防止同时开启过多线程。

### 4.6.3 后继任务及执行

在每个单独开启的执行进程中,在成功执行一个任务后,查询该任务的所有后继任务，若可以开始执行就开启新线程执行,保证每个任务的运行。

这样设计可以保证各个线程间任务的互斥,即两个线程不会对同一个任务进行操作。

### 4.6.4 任务状态维护

在进程调用任务执行结束后,又会对任务状态做自动更改,之后可以根据任务状态进行判断。

## 五.项目实现

### 1.前端页面

addpage.html:添加新任务页面

configpage.html:配置 sql 编辑器的相关选项如数据库的地址，用户名，密码等

infopage.html:查看任务信息页面

loadpage.html:sql 编辑器载入服务器上的代码

loginpage.html:登录页面

modifypage.html:修改任务页面

prepage.html:添加前驱后继页面

registerpage.html:注册页面

showpage.html:数据库查询页面

storepage.html:sql 编辑器存储功能页面

taskpage.html:数据分析任务管理系统页面

workpage.html:sql 编辑器页面

outline.css:所有其他页面的样式表

prism.css:语法高亮的样式表

d3.v3.min.js:D3 库

dagre-d3.js:绘制有向无环图的工具

prism.js:语法高亮的工具

### 2.sql 编辑器

work.py:sql 编辑器与前端交互，执行 sql 语句并返回结果

config.py:配置数据库相关的参数

load.py:sql 编辑器载入功能

store.py:sql 编辑器存储功能

timevar.py:sql 编辑器对时间变量的支持，目前不支持时间加减。

### 3.数据分析任务管理系统

list.py:数据分析任务管理系统与前端交互，以及处理启动，中断和强制执行

addpage.py:添加任务，并参与交互

infopage.py:查看任务详细信息，并参与交互

modifypage.py:修改任务，并参与交互  
addedge.py:添加前驱后继，并参与交互  
dag.py:有向无环图显示，并参与交互

#### **4.数据库查询系统**

show.py:目前只支持查询数据库中的结果，并显示。与前端交互。

#### **5.用户系统**

check.py:验证密码，并参与交互  
register.py:注册功能，并参与交互

#### **6.任务调度系统**

newtask.py:runsql(taskname,sqlfile)。开始执行 sqlfile 的文件。judge(task),判断 task 是否可以立刻执行任务。

scheduler.py:开启一个新任务,用于多进程处。

init.py：初始化任务状态，并且开始最初的调度。

#### **7.数据库接口**

database.py:init()数据库初始化操作，execute(cmd)执行 sql 语句，getlist(cmd)查询并返回结果列表，delete(task)删除任务 task，并且同时删除掉和它有关的前驱后继关系。

# 六.附录

## 6.1 数据库表单信息

Database	Table	Information
Python	tasklist	任务信息
	edges	任务关系
	user	用户名和密码

### 6.1.1 tasklist 表单

项目	类型	含义
name	varchar(125)	任务的名字
type	varchar(100)	NULL 为普通任务，否则为定时任务的参数，中间用-分隔
file	varchar(100)	对应的 sql 脚本的文件名
status	varchar(25)	任务的状态：did not start（未启动），started（已启动，尚未执行），running（正在执行），finished（成功完成），failed（任务失败）
description	varchar(2550)	任务描述

### 6.1.2 edges 表单

项目	类型	含义
u	varchar(125)	前驱节点
v	varchar(125)	后继节点

### 6.1.3 user 表单

项目	类型	含义
username	varchar(20)	用户名
password	varchar(20)	密码

## **6.2 人员分工**

### **6.2.1 项目设计**

伍一鸣：总体框架设计，人员分工，数据库设计，任务调度设计

陈立康：注册和 sql 编辑器设计

梁明：前端网页设计

### **6.2.2 项目实施**

伍一鸣：后台数据库与服务器借口的实现，任务调度系统后台程序的实现，前端和数据库的连接，结果展示页面的子系统，前端、后台 bug 的调试

陈立康：注册页面后台，sql 编辑器的后台

梁明：除结果展示之外的所有前端网页，语法高亮，任务关系图绘制

### **6.2.3 项目测试**

伍一鸣：对整个项目进行检查和测试,进行 bug 修复。

### **6.2.4 项目报告**

伍一鸣：所有报告。



## 七.总结

这次项目总的来说和预想的结果差得比较远。主要的失误有以下几点：

1.在开始之前没有熟悉好 git 和 django 框架，也没有去调研关于前端页面框架的事情，平白增大了许多工作量。

2.三人协作时，存在互相踢皮球的情况，在存在相互依赖的任务面前总是要等到快到ddl才开始动。

3.前期落下了进度，在开始补的时候，没有预料到梁明退课。因为没有做好预案对于梁明之前的工作不熟悉，所以在补的时候，只能去将就梁明留下来的成果。

4.组员之间第一次合作，对各人的清空了解不够深入，没有考虑到编程水平的差距，又没有用好 UML 图这个工具，影响了团队效率。

这是我们人生中的第一个完整的项目，经过这次实践对于 git，对于 python，对于数据库，对于架构我们有了初步的了解。有了这次的经验，以后一定能够做得更好。