期末项目实验报告

2020 美国大选数据库管理系统

杨 馥 冰
19307130304

2021 年 6 月 3 日

目 录

1 实验题目

2020 美国大选数据库管理系统

1.1 项目背景及需求分析

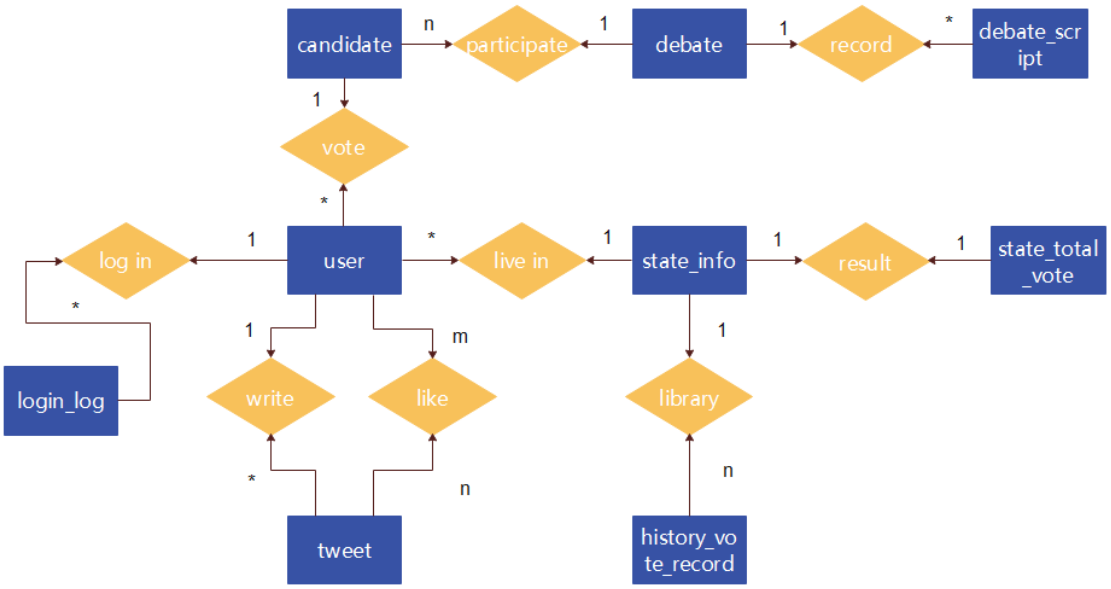
每年的美国大选都会产生大量的数据，种类丰富，包括且不限于选票信息，舆论文本信息，竞选演讲语音信息，等等。通过分析这些数据，可以对人权平等问题，文本情感分析问题，舆论支持分析问题，等等，进行广泛的研究。由于这些数据规模巨大，结构复杂，查询修改频繁，因此这些研究的背后，需要一个数据库进行数据的存储与管理。

2 开发环境

操作系统：Windows10、Linux
数据库管理软件：MySQL
编程语言：Python
Web 开发环境：Django
Django 对各种数据库提供了很好的支持，为它们提供了统一的调用 API。本项目使用 Django 自带的 ORM 实现 Python 代码和 MySQL DataBase 之间的数据转换。
提交工程文件已连接到一个搭建在服务器端的完备数据库，可以直接运行。(本地运行：根目录下执行 python manage.py runserver 0.0.0.0:8000，从浏览器中访问。服务器端运行：直接访问 124.70.16.215:8000)
如想复现建立 DataBase 的过程可以执行/SQL 下的 SQL 脚本。

3 数据库设计

3.1 概念设计 ER 图



3.2 结构设计表

建表代码：/project/votedb/models.py

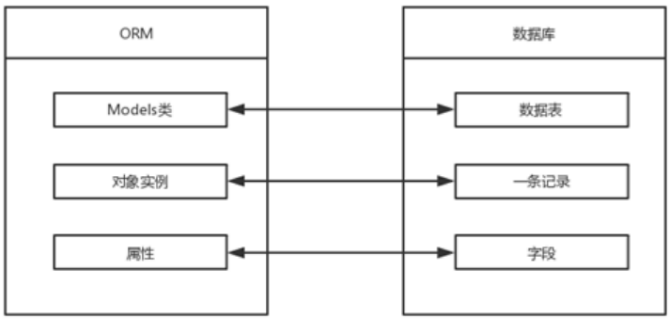
表名	属性	主码	约束	描述
candidate	candidate_name	√	VARCHAR	候选人个人信息表
	candidate_party		VARCHAR	
	candidate_description		TEXT	
	pic		TEXT	
state_info	state	√	VARCHAR	州详细信息表
	state_abbr		VARCHAR	
	state_vote		INT	
	has_votes_right_population		INT	
	in_votes_ages_population		INT	
	prisoner_population		INT	
state_total_vote	state	√	foreign key: state_info.state	州选票总数表
	tot_votes		INT	

history_vote_record	id	√	INT	历史选举信息表
	record_year		INT	
	state		foreign key: state_info.state	
	party		VARCHAR	
	votes		INT	
tweet	id	√	INT	推特信息表
	create_time		INT 使用 Unix Timestamp 表示时间	
	text		TEXT	
	likes		INT	
	retweet		INT	
	username		VARCHAR	
	user_screen_name		VARCHAR	

debate	id	√	INT	辩论信息表
	title		TEXT	
	pros		foreign key: candidate.candidate_name	
	defense		foreign key: candidate.candidate_name	
	duration		INT	
	voice_location		TEXT	
debate_script	id	√	INT	辩论脚本表
	debate_id		foreign key: debate.id	
	step		INT	
	speaker		VARCHAR	
	text		TEXT	
	time_stamp		INT	

user	username	√	VARCHAR	用户表
	pwd		VARCHAR 使用md5加密	
	state		foreign key: state_info.state	
	authority		INT	
	voted		foreign key: candidate.candidate_name 可以为 null	
login_log	id	√	INT	用户登录日志表
	time_stamp		INT 使用 Unix Timestamp 表示时间	
	content		TEXT	

3.3 数据库实现



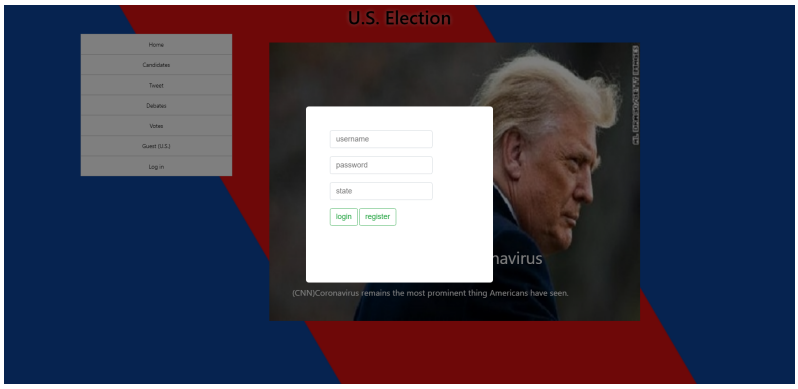
4 系统设计

4.1 关键代码

本项目基于 Python+Django 实现。各类查询以及增删改操作包含在 project/pj。

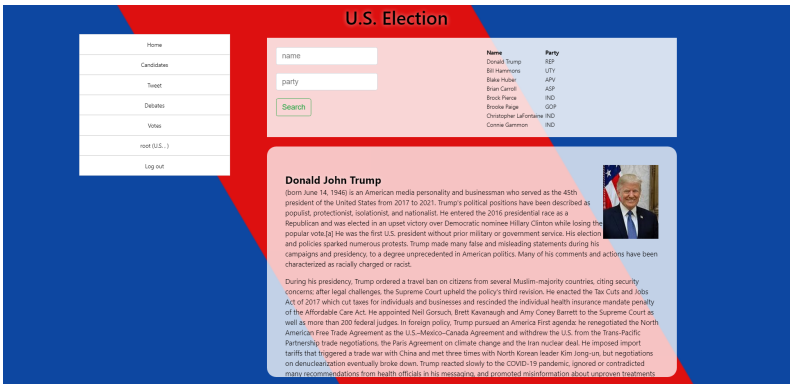
4.2 典型页面

1. 主页 (Home)
后端数据库逻辑：/pj/welcome.py
前端页面设计：/templates/welcome.html
本页面为欢迎页面，包括左侧导航栏以及右侧自动滚动播放的新闻栏。



单击导航栏 login 按钮进行登录：
超级用户：用户名 root，密码 root。拥有最高权限。(但 root 用户没有投票权限，这样比较贴合现实的投票规则)
访客用户：单击 register 按钮可注册。如想进行投票，需要在注册时输入州信息。(可视作投票规则)
登录成功后，欢迎页面出现候选人头像，点击头像为他投票！(投票操作会修改 vote 表信息)

2. 候选人页面 (Candidate)
后端数据库逻辑：/pj/candidate.py
前端页面设计：/templates/candidate.html
本页面为候选人信息查询页面。上方查询栏支持输入候选人姓名、党派进行查询。单击 search 右侧弹出可选项，点击查看详细信息。



3. 推特页面 (Tweet)

后端数据库逻辑：/pj/tweet.py

前端页面设计：/templates/tweet.html

本页面为推特交流页面。左下方查询栏支持按获赞数、转发数筛选推文；输入发布者昵称、推文内容进行查询。按回车将按条件查询推文，在右侧显示。访客登录后可以为推文点赞，发表推文。

超级用户除此之外还可以删除推文。

4. 辩论页面 (Debates)

后端数据库逻辑：/pj/debates.py

前端页面设计：/templates/debates.html

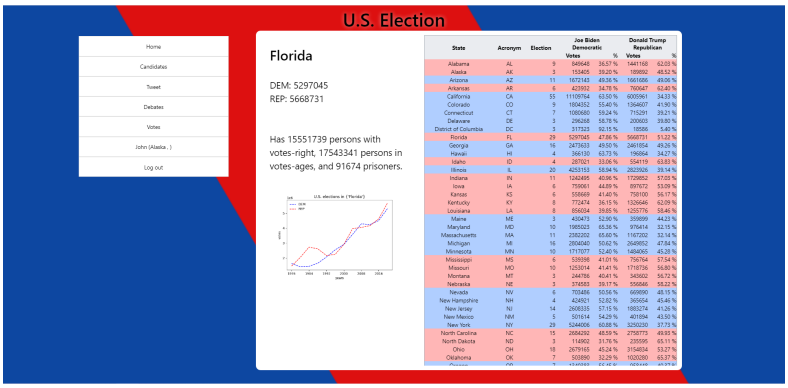
本页面为大选辩论记录查询页面。上方查询栏提供可选项，单击选择辩论，下方显示该场辩论的书记官记录，实现跟随进度条自动翻页。(提交工程只在第一份辩论中实现了该功能)

5. 选举信息页面 (votes)

后端数据库逻辑：/pj/votes.py

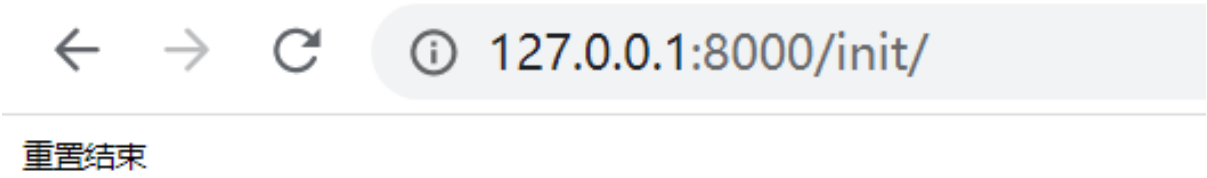
前端页面设计：/templates/votes.html

本页面为选举信息查询页面。右侧表格列出 2020 美国大选各个州的选举信息，底色代表胜方。单击州名左侧显示该州详细信息以及历年选举信息，实现导入 matplotlib 生成折线图。超级管理员可以双击修改得票信息。

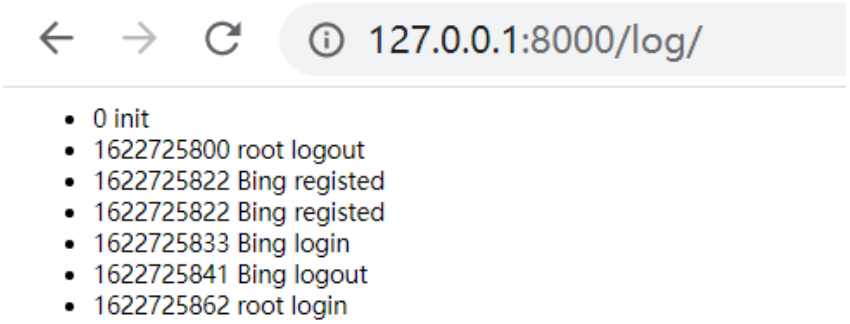


6. 数据重置 (init)

本项目提供重置数据的功能，在超级管理员权限下，可以对数据可进行数据重置，此操作将还原数据到初始状态。



7. 查询登录登出 log 信息 (log)
超级管理员可以查询所有用户的登录登出信息。以表格形式在页面内显示。



5 特色和创新点

5.1 对选票信息管理

系统维护 2020 年美国大选中以州为粒度的选票信息，为满足范式，我对原始数据进行了拆分，将该州得票总数的相关信息提取到一个新表中，也可以方便聚合。其中，每个州的选票信息包含了州名，选举人，获得的选票数，是否在该城市选举获胜等信息。

虽然目前选票早已统计完毕，但为了便于后续修改补充，我依然在系统中支持了对每个城市选票信息的增删改操作 (仅限管理员用户 @root)。此外，系统也应能根据数据自动填充每个城市的选举获胜者以及得票率 (触发器以及函数实现)。

5.2 对州合法选举信息统计与校验

为了调查美国大选是否能充分反映美国公民的的意愿，另有一个数据集对合法选票，拥有投票权的总人口数，在投票年龄段内的总人口数，非公民人口数等等进行了统计，可以用于推断每个州的选票是否能充分反映当地民众的意愿。本项目将这些信息存放在表 state_info 中。

5.3 舆论语料的管理与查询

在选举期间，推特上充斥着大量关于某两位总统候选人——特朗普与拜登——的讨论，这些文本是进行情感分析的主要语料。为了加速训练情感分析模型时对文本的查询，我们也需要对这些语料建立索引，从而可以快速查询所需要的文本，例如包含某些词的语料，某个用户的所有发言，某个时间段被点赞/转推最多的发言，等等。本项目对这些限制条件进行自由组合并实现。

由于赋予管理员用户更高的权限，他对 tweet 表信息可以进行增删改。此外，成功登录系统的访客用户也可以发表推特（增），点赞（改）。

5.4 竞选辩论音频查询

在选举期间，两名候选人进行了多次正面交锋，这些辩论的语料被用音频和文本的方式记录了下来。除了简单的将音频和文本进行存储与播放外，我还实现了秒级的音频定位，并同步显示对应的文本。

5.5 用户管理

虽然该系统的数据来源公开可查询，但依然需要对用户进行分级，保证数据不被篡改。因此需要将系统用户分为访客、用户和超级管理员。访客只能进行查询类操作；用户除此之外可以发表推文，投票，点赞；超级管理员用户还可以对竞选数据和推特数据进行增删查改。

系统初始自动创建一个超级管理员用户，且后续无法再次添加新的超级管理员。用户的密码不能以明文形式保存于数据库中，而是经过 md5 加密后保存在数据库中。

超级管理员可以查询所有用户的登录登出记录。为了方便查询，该记录也应当包含时间，用户信息。

6 提交文件说明

6.1 /report

实验报告

6.2 /SQL

SQL 脚本

1. 建表

a.(/database/1.sql) 建库，创建用户——访客、授权，开启表空间

```
create database dbpj;

create user 'guest'@'%' identified by '123123';
grant select on dbpj.* to 'guest'@'%';
grant all privileges on dbpj.django_session to 'guest'@'%';
flush privileges;

SET GLOBAL innodb_file_per_table=ON;
```

b.(/database/2.sh)Django 的数据迁移，将数据导入数据库

```
python manage.py makemigrations
python manage.py migrate --database=superadmin
```

2. 视图

创建视图，用来进行 candidate 的查询。如果 candidate 发生了增删改，就更新这个视图。

```
create or replace view candidate_list as
select name, party from votedb\_candidate;
```

3. 触发器

创建触发器,当 root 超级用户修改某个票数时,自动更新表 votedb_state_vote 中的总得票数。

```
CREATE TRIGGER trig_vote AFTER UPDATE
ON votedb\_state_vote FOR EACH ROW
BEGIN
    CALL renew_state_total_vote(NEW.state);
END
```

4. 存储过程

更新表 votedb_state_vote 中的总得票数。

```
CREATE PROCEDURE renew_state_total_vote(IN name varchar(50))
```

```
BEGIN
    DECLARE s INT DEFAULT 0;

    SET s = aggregation_state_vote(name);

  update votedb_state_total_vote
  set votes = s
  where state = name;
END
```

5. 函数

计算表 votedb_state_vote 中的总得票数。

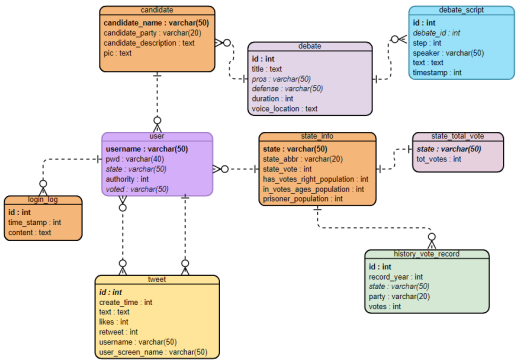
```
create function aggregation_state_vote(name varchar(50)) Returns INT
Begin
  DECLARE s INT DEFAULT 0;

  select sum(votes)
  INTO s
  from votedb_state_vote
  where state = name
  group by state;

  return s;
end;
```

6.3 /models

数据库模型图



6.4 /project

项目工程文件

1./data

本项目数据库的数据集
2./pj
后端查询部分的逻辑，python 开发。
3./statics
前端页面用到的字体、图片、音频等文件。
4.templates
各页面源码，html 开发。
5.votedb
后端建立数据库部分的逻辑，python 开发。
6.manage.py
启动工程文件。

7 实验总结

通过本次实验，我对基于 MySQL 和 Django 的数据库及 Web 开发有了一定的了解。我认为开发一个数据库应用，重点在于开发和管理。

首先是开发，基于 Python 建立一个数据库，从建表、添加约束、检查冗余，到编写触发器、函数、过程，以实现更多的功能，这些提高了我的编程能力，也加深了我对数据库模型的理解。付诸实践后，我体会到了数据库系统的完备和复杂，将冗杂的数据整理成表对逻辑思维和耐心程度要求很高，在本次实验中，我采用的 tweet 数据集含有较多难以编码的表情和符号，在洗数据集的过程中，我不仅学习了新的编码规则，也更深刻地理解了范式、主码、外键和约束。

其次是管理。通过本次实验我认识到维护一个数据库的不易，本项目仅包含 9 个实体集，在一些边界查询和索引中，我遇到了很多问题，在解决问题的过程中，我体会到管理一个数据库不仅需要良好的逻辑思维，更需要对该数据库的设计和功能十分了解，也需要管理数据库的经验。

通过本次实验，我也学到了许多 UI 设计的知识，包括 HTML,CSS,JS 等。前端开发是经验性的工作，本项目在 UI 设计上有许多不足，但也是一次很好的实践，让我对 web 开发有了初步的认知。希望通过以后的学习来逐步完善。