Phase 3:
Design and Implementation for
PCD Database

Ibrahim Al-Saud, Emanuelle Crespi

Table of Contents

CHANGES TO PHASE 1 & 2:

There are changes from phase 1 and 2 that show in our final product. The main change that was made involved the relational schema. We had decided to simplify the table by getting rid of the election entity table and minimizing some links to the president entity by removing CID and leaving the NID as an identifier. This was to avoid confusion between presidents who have not won an election, but still ended up in presidency. We added a table of winners to identify the election winners for a corresponding year. However the main reason for this was that some winners were not winning by electoral vote, but by the house of representative decision. The diagrams on the following pages, represent our final schema and relational model for this database.

PURPOSE:

PCD Database is meant to be used for users to identify patterns based on the information gathered on presidential candidates from 1789 to 2016. The server is responsible for receiving user requests, in optional text boxes and modes, tol request from the database; thus, giving the server a temporary relation that is turned into an html page to be displayed to the user.

DESIGN/IMPLEMENTATION:

The design offers the user access to selecting a mode and three attributes, year, candidate, party. These are used as the query information when accessing the database. A query is processed based on the mode and returns the information to the user as a formatted table. The following limitations have been considered in this process.

Assumptions and Limitation

- User speaks/reads English language.
- Understands US election process.
- Knows how to read SQL table format.
- User does not need authentication to access.
- Basic web browsing skills.
- Assuming Campaign slogans start at 1936 (the rest are null unless populated later)
- Assuming vices are linked to candidates from 1852 (the rest are not in the vice table).
- Spending data only shows from 1840 till 2012.
- Contribution and donation data only shows from 2008 till 2016.

Apache Server Implementation

The use of apache is implemented through a third party software known as XAMPP. This allows use to start, reset, and stop our database and web server at our convenience. It also allows us to link our web page to a localhost connection for demo purposes. Python 2.7 uses the MySQLdb library to establish a connection with the database when necessary. This is only when the user has clicked the SUBMIT option on the web interface. Further explanation of this is found in the user guide.

Implementation of MODES (1-9)

- One (select only a year to query)

```
SELECT C.year, C.CID, P.P_name, N.name, B.electoral, W.ISA
FROM Nominee N, Cand_Nom CN, Candidate C, Ballot B, winners W, party P, Affiliated A
WHERE C.year = 2008
AND N.NID = CN.NID
AND CN.CID = C.CID
AND C.CID = B.CID
AND W.CID = C.CID
AND A.CID = C.CID
AND A.PID = P.PID
```

- Two (Query the DB on a selected candidate)

```
SELECT C.year, C.CID, N.name, B.electoral, B.popular, B.polls, W.ISA
FROM Nominee N, Cand_Nom CN, Candidate C, Ballot B, winners W
WHERE N.name = 'Theodore Roosevelt'
AND N.NID = CN.NID
AND CN.CID = C.CID
AND C.CID = B.CID
AND W.CID = C.CID
```

- Three (Query of non-contiguous presidents)

```
SELECT name
FROM president
WHERE inoffice like '%/%';
```

- Four (Query on swing candidates)

```
SELECT C.year, C.CID, N.name, B.electoral, B.popular, B.polls, W.ISA, P.P_name
FROM Nominee N, Cand_Nom CN, Candidate C, Ballot B, winners W, Affiliated A, Party P,

(SELECT Nominee.name
FROM Nominee
WHERE (SELECT count(DISTINCT(P.P_name))
FROM Nominee N, Cand_Nom CN, Candidate C, Ballot B, winners W, party P, Affiliated A
                WHERE N.name = Nominee.name
                AND N.NID = CN.NID
                AND CN.CID = C.CID
                AND C.CID = B.CID
                AND W.CID = C.CID
                AND A.CID = C.CID
                AND A.PID = P.PID
                GROUP BY N.name) > 1) swing

WHERE N.name = swing.name
AND N.NID = CN.NID
AND CN.CID = C.CID
AND C.CID = B.CID
AND W.CID = C.CID
AND A.CID = C.CID
AND A.PID = P.PID
```

- Five (Query on an analysis of the Parties)

```
SELECT P.P_name, count(DISTINCT(C.CID)) as num_candidates,SUM(B.popular) as popular,
SUM(B.electoral) as electoral, SUM(W.ISA) as wins
FROM Nominee N, Cand_Nom CN, Candidate C, Ballot B, winners W, party P, Affiliated A
WHERE N.NID = CN.NID
AND CN.CID = C.CID
AND C.CID = B.CID
AND W.CID = C.CID
AND A.CID = C.CID
AND A.PID = P.PID
GROUP BY P.P_name
```

- Six (Query on each candidate campaign)

```
SELECT N.name, C.year, CP.expenses, CP.CONTRIB, CP.SLOGAN
FROM Nominee N, Cand_Nom CN, Candidate C, Campaign CP
WHERE N.name = 'Barack H. Obama'
AND CN.NID = N.NID
AND CN.CID = C.CID
AND C.year = 2008
AND CP.CID = C.CID
```

- Seven (Query a Candidate for their vice)

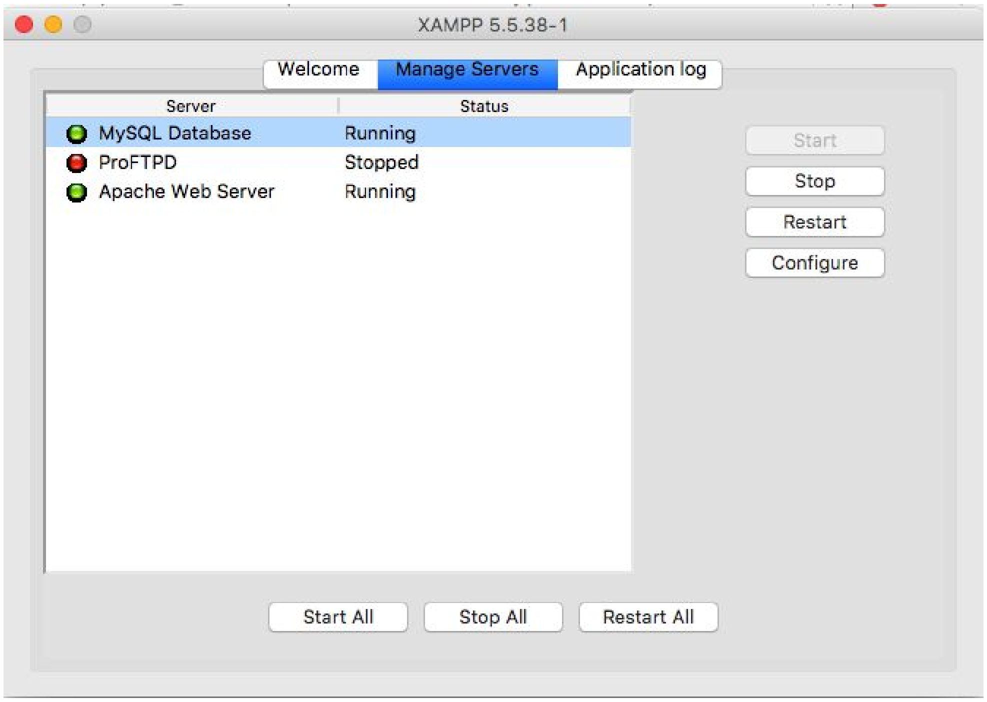
  ```
  SELECT C.year, N.name, V.V_Name
  FROM Nominee N, Cand_Nom CN, Candidate C, Vice V
  WHERE N.name = 'Barack H. Obama'
  AND N.NID = CN.NID
  AND CN.CID = C.CID
  AND V.CID = C.CID
  ```

- Eight (Query for presidents from the same homestate)

  ```
  SELECT p.homestate, count(DISTINCT(p.name)) as num_presidents
  FROM president p
  GROUP BY p.homestate
  ```

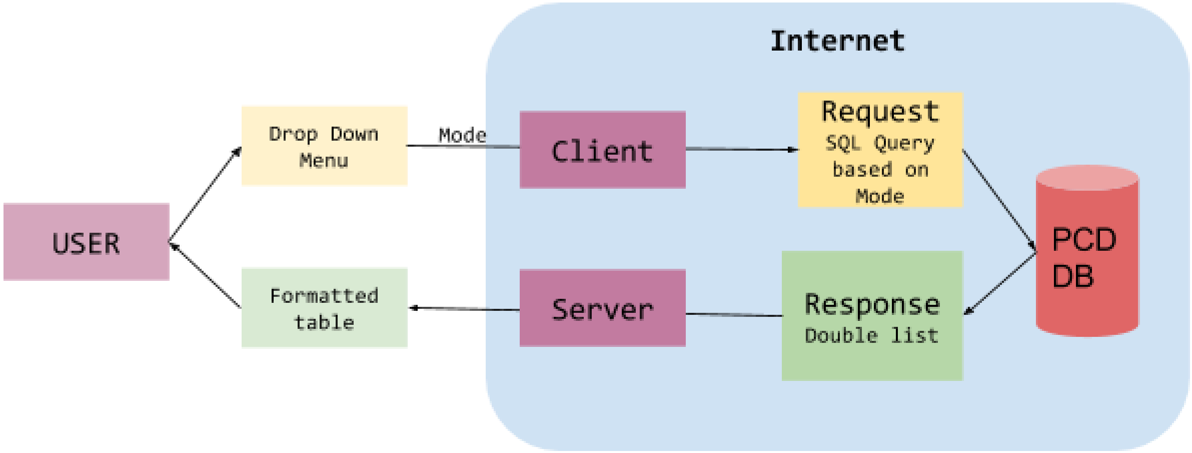
- Nine (Query for presidents information)

  ```
  SELECT p.name, p.dob, p.birthplace, p.homestate FROM president p
  ```



Server	Status
🟢 MySQL Database	Running
🔴 ProFTPD	Stopped
🟢 Apache Web Server	Running

NOTE: Implementation of the web server has not been configured for remote access



Network Flow chart

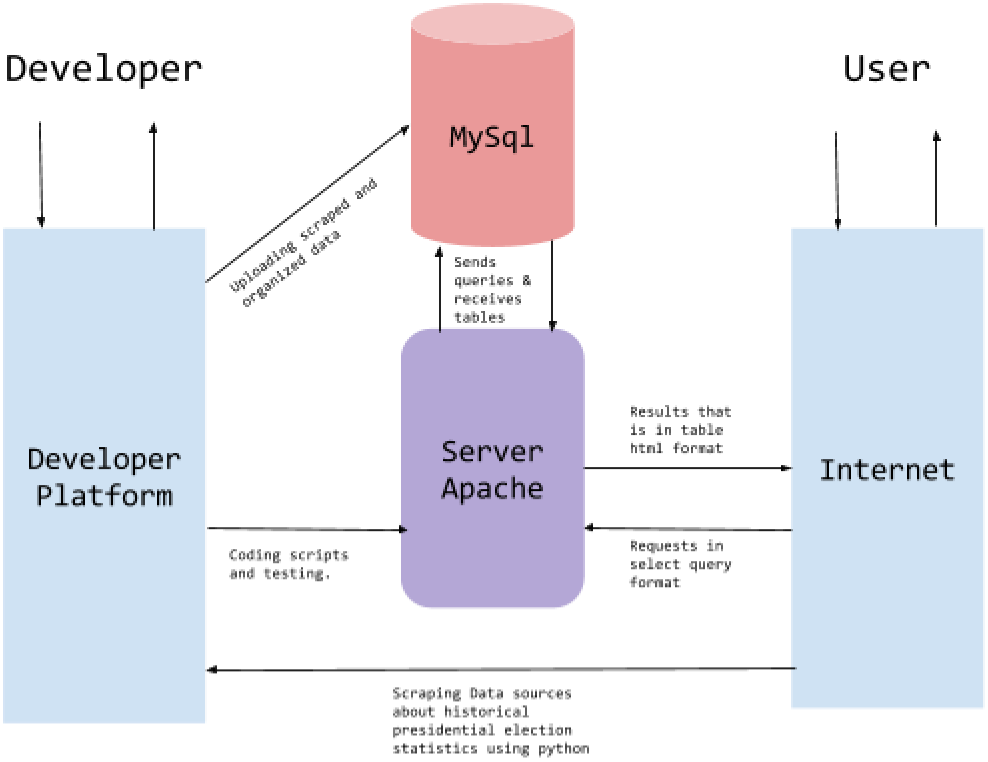
Extract Load Transform

Our ETL process was straightforward as after we found our sources, provided in the bottom, each website had specific code for it to be crawled. In the extraction step, we downloaded the sources and crawled each using XPATH address to the elements we needed. The XPATH is inserted into the Selenium driver, which helps loading javascript pages. Selenium also enables us to control the browser we're crawling from. In our case, we used Firefox and a browser profile that blocked ads to assure no interruption. After obtaining the html element we needed, we inserted them into BeautifulSoup, which enabled us to navigate through tables' rows and columns easily. Crawling sites with tables took seconds, but crawling sites with multiple links took us hours.

After this data was organized into hash tables and lists, we load it as a JSON file. JSON was our choice of data-interchange language because it is fast, lightweight, and easily editable. The first site to be crawled was 270towin.com which contains all the candidate names and their party affiliation, popular and electorial votes. In addition, we got the years in which all the elections happened. This was a setting stone for our database. Then, we moved on to crawling slogans, campaign expenses and contributions, polls prior to election, and details about presidents.

After our extraction and loading steps were done, we moved on to cleansing, removing unwanted repetition, the data and transforming it into usable insertion commands. Starting from our Nominee table as it only required the candidate names we got from 270towin.com. Then, we moved on to candidate, Affiliation, party, and campaign. Now we're left with winners, President, and ballot. We took on ballot which was inconsistent as data collected doesn't cover the table. As we know popular votes only started in the 1840s which means we have 1789-1836 with null values in popular vote. The same can be said about polls prior to the election. These inconsistencies were resolved by adding null values or -1. Nonetheless, we were able to make all the needed inserts to fill the database. We chose this approach as it makes it easier to debug the database if anything wrong was found in the insert commands.

We faced difficulties in our ETL as we kept finding irregularities that would challenge the organization of our database, but they were often resolved quickly by researching the irregularity and changing the schema accordingly. We opted out of using some of the tools we specified in phase one and two, such as Scrapy. Scrapy was slow compared to Selenium and used more computing power.

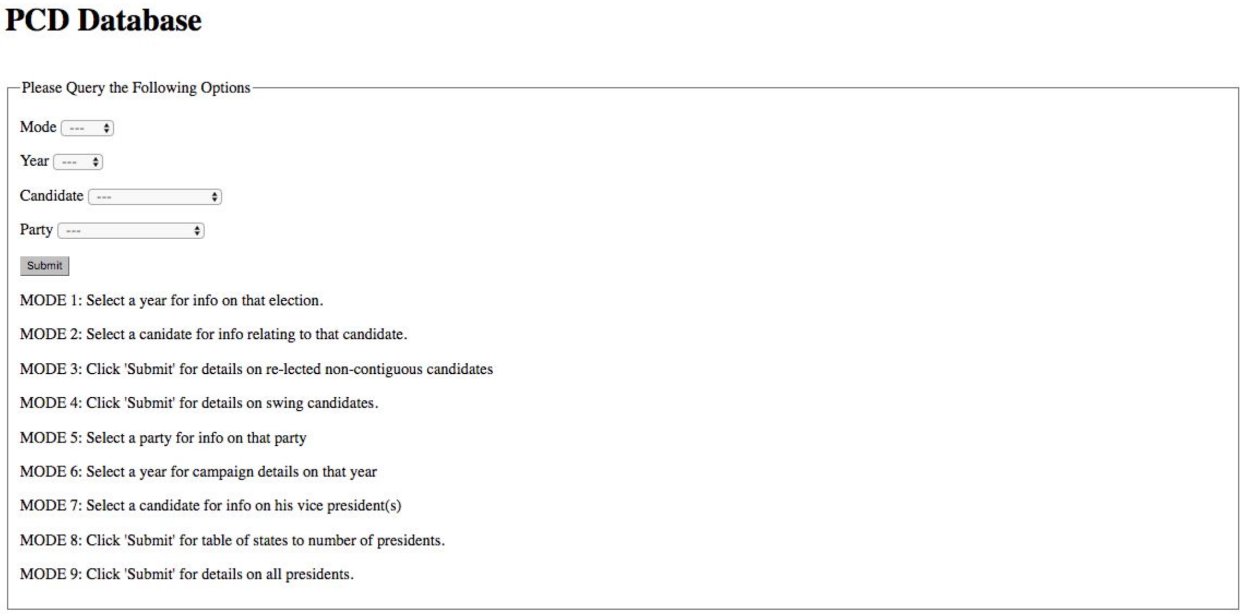

Developer
MySql
User
Uploading scraped and organized data
Sends queries & receives tables
Developer Platform
Server Apache
Internet
Results that is in table html format
Requests in select query format
Coding scripts and testing.
Scraping Data sources about historical presidential election statistics using python

Project flow chart

USER GUIDE:

Upon connection to the web server interface. There is a button with the phrase “Let’s Go!” indicating a means of connection to the query interface.

Clicking this button advances to the interface. Showing 4 selectable menu options necessary for a unique query to the PCD Database. Further explanation of the interface is described below.



PCD Database

Please Query the Following Options

Mode ---

Year ---

Candidate ---

Party ---

Submit

MODE 1: Select a year for info on that election.

MODE 2: Select a canidate for info relating to that candidate.

MODE 3: Click 'Submit' for details on re-lected non-contiguous candidates

MODE 4: Click 'Submit' for details on swing candidates.

MODE 5: Select a party for info on that party

MODE 6: Select a year for campaign details on that year

MODE 7: Select a candidate for info on his vice president(s)

MODE 8: Click 'Submit' for table of states to number of presidents.

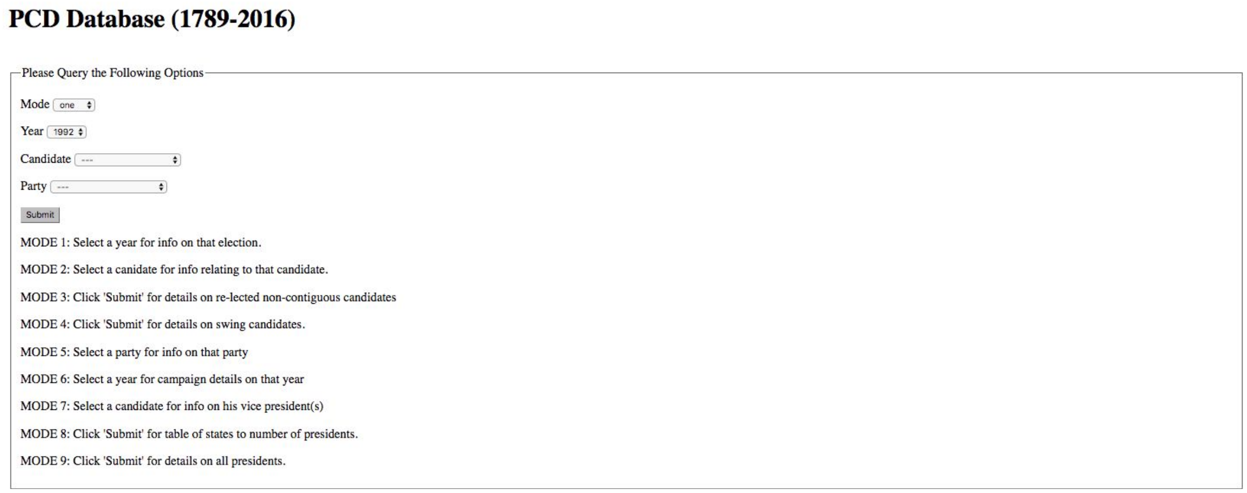
MODE 9: Click 'Submit' for details on all presidents.

Modes

- One (select only a year to query)
- Two (Query the DB on a selected candidate)
- Three (Query of non-contiguous presidents)
- Four (Query on swing candidates)
- Five (Query on an analysis of the Parties)
- Six (Query on candidates campaign)
- Seven
- Eight (Query on states counting the total on each president)
- Nine (Query on presidential information)

Demo

NOTE: Below is an example of interacting with MODE 1



PCD Database (1789-2016)

Please Query the Following Options

Mode one

Year 1992

Candidate ---

Party ---

MODE 1: Select a year for info on that election.

MODE 2: Select a canidate for info relating to that candidate.

MODE 3: Click 'Submit' for details on re-lected non-contiguous candidates

MODE 4: Click 'Submit' for details on swing candidates.

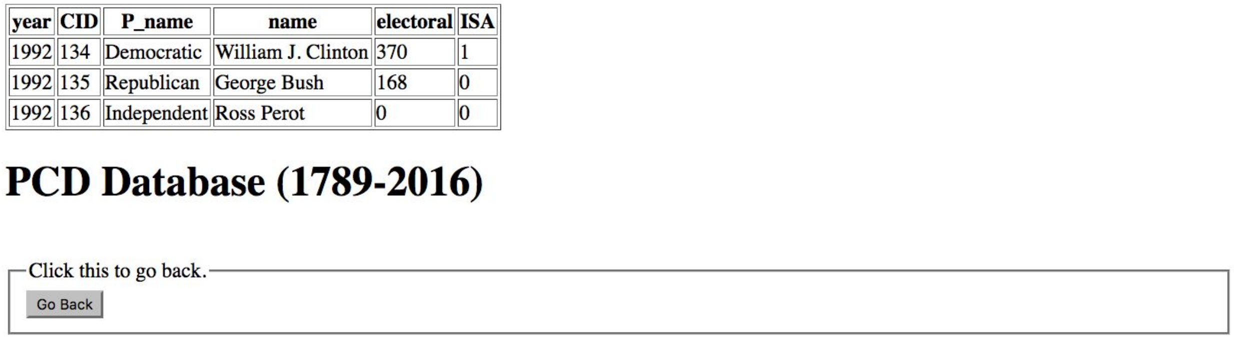
MODE 5: Select a party for info on that party

MODE 6: Select a year for campaign details on that year

MODE 7: Select a candidate for info on his vice president(s)

MODE 8: Click 'Submit' for table of states to number of presidents.

MODE 9: Click 'Submit' for details on all presidents.



year	CID	P_name	name	electoral	ISA
1992	134	Democratic	William J. Clinton	370	1
1992	135	Republican	George Bush	168	0
1992	136	Independent	Ross Perot	0	0

PCD Database (1789-2016)

Click this to go back.

Works Cited (Scrape Sources)

Elections data by year
http://www.270towin.com/historical-presidential-elections/

Percentages/Ballot Statistics
https://www.loc.gov/rr/program/bib/elections/statistics.html
http://www.infoplease.com/ipa/A0781450.html

U.S. population
https://fusiontables.google.com/DataSource?dsrcid=225439#rows:id=1

President Homestates
https://en.wikipedia.org/wiki/List_of_Presidents_of_the_United_States_by_home_state

Political Parties
https://en.wikipedia.org/wiki/List_of_political_parties_in_the_United_States
http://www.globalelectionsdatabase.com/index.php/index

List of Candidates
https://en.wikipedia.org/wiki/List_of_United_States_presidential_candidates

Campaign slogan (1840 - 2016)
https://en.wikipedia.org/wiki/List_of_U.S._presidential_campaign_slogans
http://www.presidentsusa.net/campaignslogans.html

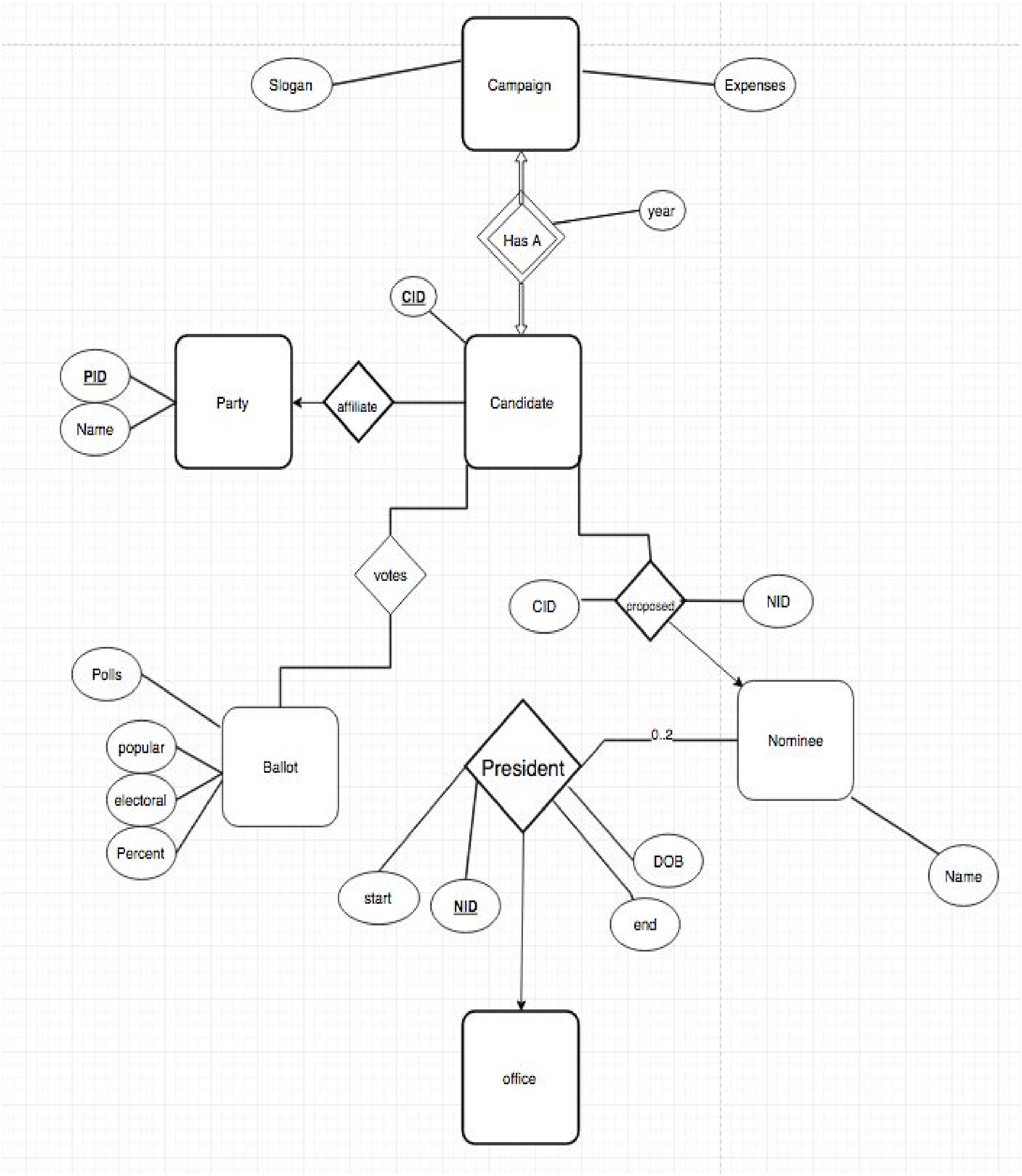
Campaign Expenses
http://www.fec.gov/disclosurep/pnational.do

Net Worth of candidates
http://www.huffingtonpost.com/2011/02/21/the-net-worth-of-the-amer_n_825939.html
http://www.usatoday.com/story/news/politics/elections/2015/08/26/24-7-wall-st-net-worth-presidential-candidates/32409491/

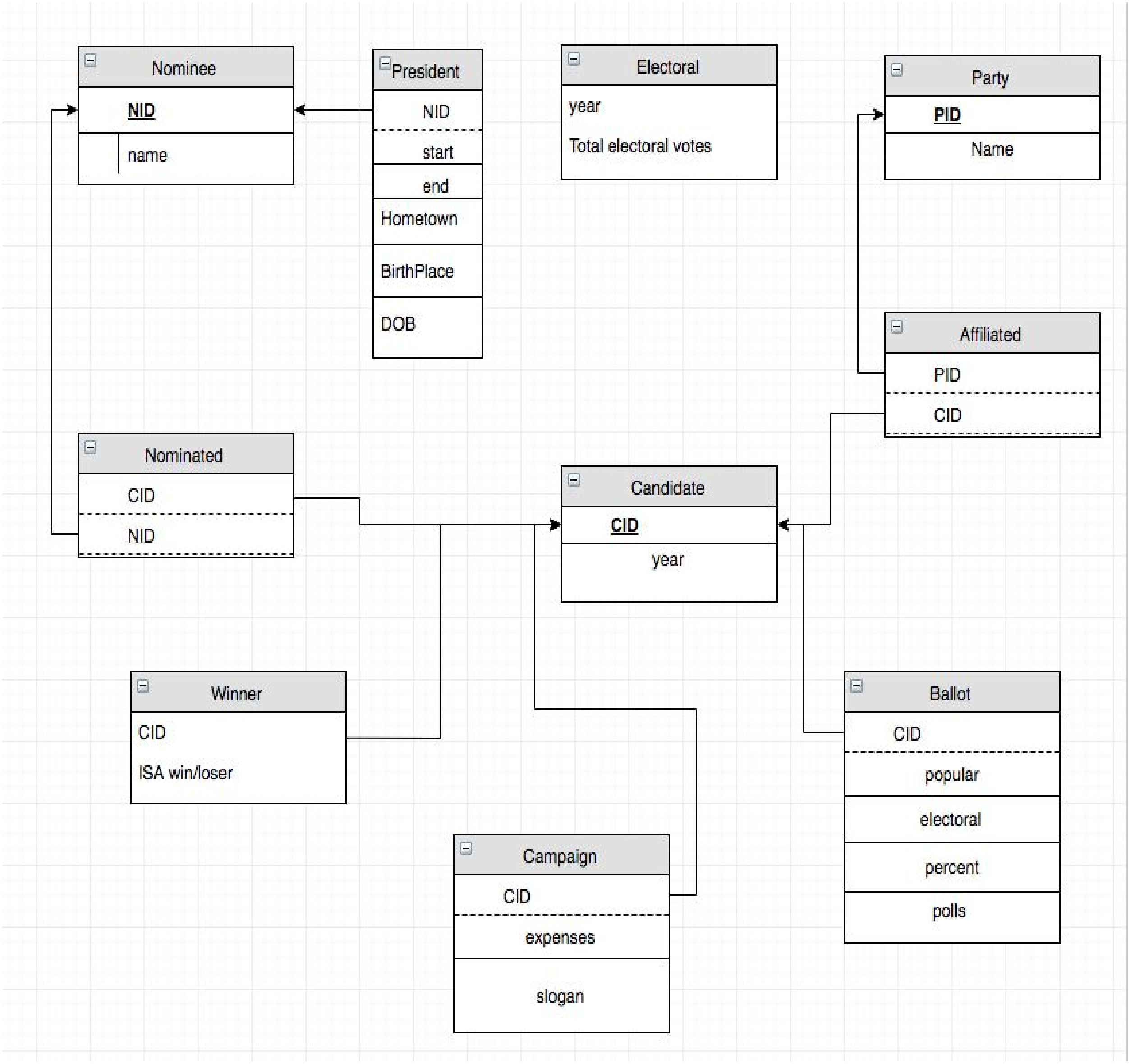
Vice Presidents
https://en.wikipedia.org/wiki/List_of_United_States_presidential_candidates

Electoral College Historical data
https://en.wikipedia.org/wiki/Electoral_College_(United_States)

LOGICAL MODEL:



RELATIONAL SCHEMA:


Nominee
NID
name
President
NID
start
end
Hometown
BirthPlace
DOB
Electoral
year
Total electoral votes
Party
PID
Name
Affiliated
PID
CID
Nominated
CID
NID
Candidate
CID
year
Winner
CID
ISA win/loser
Ballot
CID
popular
electoral
percent
polls
Campaign
CID
expenses
slogan