CS377, Fall 2024

Assignment 2 - Part 1 (40 points)

Due: Friday October 11 at 11:59PM. You can optionally work in pairs on this assignment. (Late submissions policy: see syllabus for details.)

General Instructions

Ready for some SQL fun with a dataset from the real world? Let's do this! This is the first part of your SQL assignment (worth 40% of A2's grade). The second part will be published after the midterm test and will involve more advanced SQL operations. Note: You are encouraged to work on your *Google Cloud Platform* PostgreSQL instance for this assignment (using your free academic credit!).

Download the starter zip folder from the A2 directory on Canvas. You'll find the following:

- The database schema script (`a2_dimeDB.sql`). Execute the lines in this SQL script in your postgres instance (or in pgAdmin if you're working locally). This script will create the schema and populate its tables from a public Google storage bucket. Note that your submission will be tested later on a *different* dataset from the one that will be loaded into your database here, so make sure your queries can work on any valid database instance and don't contain *hard-coded* identifiers, for example.
- Expected answers for some of the queries (in `expected_answers/`), based on the provided test dataset. Note that these answers were produced on a GCP PostgreSQL instance. There may be minor differences in the ordering of *strings*, if you run your queries on a different platform or operating system with different text encoding.

Once you have submitted your files, be sure to check that you have submitted the correct version; new or missing files will not be accepted after the due date (unless you and your partner -if applicable- have a late-token remaining and you resubmit within 24 hours of the deadline).

DIME Schema Description

Congratulations, you've been hired by a research think tank in D.C. to study how campaign financing and political donations impact politicians in America. You will be using a real-world data set collected and published by scholars at Stanford University, known as: The Database on Ideology, Money in Politics, and Elections (DIME) Adam [2016], Bonica [2016]. This database was developed as part of a large-scale effort to "construct a comprehensive ideological mapping of political elites, interest groups, and donors."

This database contains 5 tables, described briefly below. For detailed descriptions of the attributes in each table, you will read specific sections from the official DIME documentation (referenced below). You can find the official documentation files in our A2 Canvas folder.

- **Recipient**: A tuple in this relation represents a political candidate and contains information on the election cycle, fundraising statistics, election outcomes, and characteristics like name and gender, if applicable – among other features. Note that we refer to candidates as *recipients* because they *receive* donations from contributors. Each candidate is identified by a unique recipient ID referred to as "bonica.rid". [1]

  Read the detailed description of this table's attributes in **section 5.1** of the document **dime_codebook_v3_1.pdf**, pages 6-9.

- **Contributor**: A tuple in this relation represents a donor, also known as a contributor. Donors can be individuals or organizations. Each donor is identified by a unique contributor ID referred to as "bonica.cid". This table also contains a column for each election year covered by the data, storing the amount of donations given by a contributor during that year. For example, the column "amount.2014" stores the amount of campaign donations made by a contributor during the 2014 election cycle, and so on.

  Read the description of this table's attributes in **section 5.2** of **dime_codebook_v3_1.pdf**, pages 10-11.

[1] Bonica is the last name of the scholar who supervised the DIME project at Stanford.

- **Contribution**: A tuple in this relation represents a donation made by a contributor (identified by "bonica.cid") to a recipient/candidate (identified by "bonica.rid") during a given election cycle.

  Read the description of this table's attributes in **section 5.3** of **dime_codebook_v3_1.pdf**, pages 11-13.

- **Bill**: A tuple in this relation represents a congressional bill. Each bill is uniquely identified by a "bill.id". A bill has a name, a description, a list of sponsors and co-sponsors, among other things.

  Read the description of this table's attributes in **section 2.3** of the second document **dime_plus_codebook.pdf**, page 5. (Some additional attributes in the database are incorrectly marked as being part of another table called Congressional Text, described in **section 2.1**, pages 4-5, of the same document.)

- **Vote**: And finally, this table provides information on how congress members voted for a given bill. Each tuple represents a single vote made by a legislator (identified by their "bonica.rid" value) on a bill (identified by "bill.id"), showing their "vote.choice", whether they were the sponsor/cosponsor for the bill or not, and more.

  Read the description of this table's attributes in **section 2.2** of the document **dime_plus_codebook.pdf**.

SQL Statements

After reading the documentation sections referenced above and developing a good understanding of the schema, you will now write SQL queries that extract interesting information from this database.

Important things to keep in mind when writing your queries:

- Write your SQL statement(s) for each question in separate files named `q1.sql`, `q2.sql`, ... etc., and submit each file on **Gradescope**. You can use views if it helps you break down your solution into multiple steps that are easier to tackle, and it can make your queries more readable. Note that the views you create in one file cannot be accessed in another file.
- Each of your files must begin with the line: `SET search_path TO dimeDB;`

  Failure to do so may cause your query to raise an error, leading you to get a 0 for that question.
- The output of each query must match **exactly** the specifications given in the question: attribute names and order, ordering of the tuples, and how to treat duplicates. It is your responsibility to make sure your code runs with no errors and the output matches our expectations exactly, to avoid losing points due to autograder crashes.
- Some attributes in the schema have a dot "." symbol in their name. These attributes must be enclosed in double quotes within your SQL queries. For example, every time you need to reference the attribute `bonica.cid`, you must type it as `"bonica.cid"`.

  E.g.: `SELECT "bonica.cid" FROM Contributor ....`

---

1. **(6 points) Georgia on my mind.** Who ran for the governor seat in the state of Georgia during the 2022 election cycle? Write a query that finds these candidates and reports for each candidate their name and party affiliation. Hint: The attribute `seat` in table Recipient contains information on the type of office seat the candidate was running for during a given election.

   The output of your query must match the exact format described below:

Attribute	
name	Name of the candidate
party	The code for the candidate's party affiliation
Order by	name
Duplicates?	No duplicates

2. **(6 points) Many, many donors.** Who is the candidate that had the highest number of distinct donors for an election, among all candidates (and across all elections present in the data)? Hint: The attribute `num.givers` in table Recipient can help you find this information. If there are ties, report them all. Your query should return that candidate's recipient ID (i.e., `bonica.rid`), their name, the election cycle, and the number of distinct donors (i.e., `num.givers`), renamed as 'numDonors'. When you rename an attribute, make sure you use the exact alias mentioned here, to avoid losing points.

Attribute	
bonica.rid	Unique ID of the recipient with highest distinct donors
name	Name of the candidate
cycle	Election cycle
numDonors	Number of distinct donors (stored in `num.givers`)
Order by	name
Duplicates?	No duplicates

3. **(8 points) Just take my money, 2022!** Who were the top 10 contributors of political donations in the election cycle 2022? Report their contributor IDs (stored in `bonica.cid`), their names (stored in `most.recent.contributor.name`), their type (stored in `contributor.type`), and the total amount of money they donated in the 2022 election cycle.

   You can assume that there will be no ties in the data; i.e., no two contributors will have the exact same total amount of donations in the 2022 election cycle.

   Hint: You can utilize the `LIMIT` clause in your SQL query to return a certain number of rows, if needed.

Attribute	
bonica.cid	Unique ID of the contributor
name	Contributor name
type	Contributor type ('I' = individual, 'C' = committee/organization)
totalAmount	The amount of money they donated in 2022
Order by	totalAmount, in descending order
Duplicates?	No duplicates

4. **(10 points) Take lots of money!** What if we want to consider **all** the years available in the data, not just the year 2022? Find the top 10 contributors in terms of the total amount of money they donated over the years 1980 to 2022 (inclusive). Again, report their contributor ID, contributor name, type, and the sum of money they donated over all years.

   You can assume that there will be no ties in the data; i.e., no two contributors will have the exact same total amount of donations in the data.

   Hint: You can utilize the `LIMIT` clause in your SQL query to return a certain number of rows, if needed.

Attribute	
bonica.cid	Unique ID of the contributor
name	Contributor name
type	Contributor type ('I' = individual, 'C' = committee/organization)
totalAmount	The amount of money they donated over all years
Order by	totalAmount, in descending order
Duplicates?	No duplicates

5. **(10 points) Would you cosponsor me?** Each bill in the Congressional Voting table (known in our schema as table 'Vote'), has an indicator for whether the legislator in a given tuple was a co-sponsor of that bill (1 = yes, 0 = no). Use this information to identify the bill(s) with the highest number of co-sponsors in the database. If there are ties, report them all. (See next page for more details.)

   Report the ID of the bill, the number of cosponsors, and the string description of the bill.

   Hint: You can optionally create a view first that computes the number of co-sponsors for each bill. You can then use that view to help you identify the bill with the highest number of co-sponsors.

Attribute	
bill.id	Unique ID of the bill
numCosponsors	Total number of cosponsors for this bill
bill.str	The string description of this bill
Order by	bill.id
Duplicates?	No duplicates

Honor Code: Solve this assignment without collaborating with classmates (besides your assignment partner), and without consulting external/online resources. The assignment is governed by the College Honor Code and Departmental Policy. Remember, any work you submit must be your own; otherwise you risk being investigated by the Honor Council and facing the consequences of that.

Please remember to include the following comment at the beginning of each SQL file you submit:

```
-- THIS WORK WAS MY (OUR) OWN WORK. IT WAS WRITTEN WITHOUT CONSULTING
-- WORK WRITTEN BY OTHER STUDENTS OR COPIED FROM ONLINE RESOURCES.
-- _Student1_Name_, _Student2_Name_
```

References

Bonica Adam. Database on ideology, money in politics, and elections: Public version 2.0. *Computer file, Stanford University Libraries, accessed at https://data.stanford.edu/dime*, 2016.

Adam Bonica. Dime plus.[computer file]. *Stanford, CA: Stanford University Libraries, accessed at https://data.stanford.edu/dime-plus*, 27:2021, 2016.