Quorum Coding Challenge

Working with Legislative Data

Before You Start

Please make sure you read the entire document, including the last section with some suggestions, before getting started.

Overview

At Quorum, we collect and organize a large amount of publicly available government data. For example, we provide our clients the ability to visualize all of the bills that legislators voted for or against. In order to represent the data, we designed the database with these models:

Person - An individual legislator elected to government. This includes everyone from President Joe Biden to Representative. David McKinley from West Virginia.

Bill - A piece of legislation introduced in the United States Congress.

Vote - A vote on a particular Bill. Bills can be voted on multiple times, as the Bill itself can undergo changes over the course of its life. For the purposes of this challenge, there will only be up to 1 Vote provided for each Bill.

VoteResult - A vote cast by an individual legislator for or against a piece of legislation. Each vote cast is associated with a specific Vote.

See the provided data for schema information for each of the data models.

Provided Data

You will be provided with a dataset comprised of the following four files:

bills.csv

Field	Type	Description
id	integer	The id of the bill
title	string	The name of the bill
Primary Sponsor	integer	The id of the primary sponsor (of type Person)

legislators.csv

Field	Type	Description
id	integer	The id of the legislator
name	string	The name of the legislator

votes.csv

Field	Type	Description
id	integer	The id of the Vote
bill_id	integer	The id of the bill that this vote is associated with

vote_results.csv

Field	Type	Description
id	integer	The id of the VoteResult
legislator_id	integer	The id of the legislator casting a vote
vote_id	integer	The id of the Vote associated with this cast
vote_type	integer	The type of vote cast - 1 for yea and 2 for nay

Task

Context

Part of our job at Quorum is to transform data into information, and this is what this project is about. Imagine that one of our clients have some questions, and we would like to help them providing the answers through our platform.

Story

As an user, I want to use Quorum to get access to the following information:

1. For every legislator available, how many bills did the legislator support (voted for the bill)? How many bills did the legislator oppose?
2. For every bill available, how many legislators supported the bill? How many legislators opposed the bill? Who was the primary sponsor of the bill?

Requirements

1. Build a web application (ex. a Django web application)
2. Provide the information that the client needs
3. Choose the best way to display this information.
    a. For example, both questions can be answered with simple tables.

b. You to decide how much creativity and effort you would like to put into the solution. You are free to use any UI elements that you want.

ID	Legislator	Supported bills	Opposed bills
321	Senator Lorem Ipsum	1	2

Example of table used to answer the first question

ID	Bill	Supporters	Opposers	Primary Sponsor
123	BBB 23: Build Brazil Better	10	0	Rep. Lorem Ipsum

Example of table used to answer the second question

Resources

You will be provided (CSV) with a list of legislators, bills, votes, and vote results as specified above.

Questions

After completing your implementation, you should include a write up that answers the following questions:

1. Discuss your strategy and decisions implementing the application. Please, consider time complexity, effort cost, technologies used and any other variable that you understand important on your development process.
2. How would you change your solution to account for future columns that might be requested, such as “Bill Voted On Date” or “Co-Sponsors”?
3. How would you change your solution if instead of receiving CSVs of data, you were given a list of legislators or bills that you should generate a CSV for?
4. How long did you spend working on the assignment?

You should send us a zip file (your_full_name.zip) that contains the following:

- Source code for your implementation
- A readme with steps to run your code (readme.md)
- A writeup that responds to the questions asked above (questions.md)

Evaluation Criteria

Your solution will be evaluated based on the following criteria:

1. **Correctness** - Is your output correct based on the provided data? How did you prove correctness?
2. **Structure/Readability** - Is your code well organized and easy to read? Can it be extended reasonably if new requirements are given?
3. **Proficiency with Language** - Does your code make use of typical conventions in your language of choice?

Suggestions

- **Use this test as an opportunity to demonstrate your skills**, we are going to evaluate the code quality as well as the decision making process.
- You may use any programming language you like. Feel free to make use of any external libraries (such as parsing CSVs), but you should not need anything complicated such as a database.
- Consider committing your work with git so that we can see your progress