

Deliverables

There are nine deliverables: your writeup containing the relational algebra expressions for the problems below (in PDF format), and six SQL scripts: `script1.sql ...script6.sql`, a simple Python script that generates two HTML tables, and a small webpage (inside a directory called “www”).

Pair Programming

You are required to work with your assigned partner on this assignment. You must observe the pair programming guidelines outlined in the course syllabus — failure to do so will be considered a violation of the Davidson Honor Code. *Collaboration across teams is prohibited and at no point should you be in possession of any work that was completed by a person other than you or your partner.*

1 Introduction

In this assignment, we will extract some useful information from the voting data on to the 117th Senate, 1st session, by planning queries using relational algebra, and transforming them into SQL queries performed on MySQL.

1.1 (5 pts) Required Reading

- ☐ Textbook Chapter 4
- ☐ Textbook Chapter 6 up to (including) 6.1.

Please indicate individually that you did the above readings in your writeup.

1.2 Material

The material is a database populated with the voting data for the 117th Senate, 1st session. The SQL script that creates the database schema and all relation schemas, additionally performing data insertions, is provided together with this assignment.

1.3 Policies and Guidelines

Policy 0. Every query must be planned **first** by creating the relational algebra expression.

Policy 1. You are not allowed to use any stored variables (including user-level variables), procedures, and functions in this assignment.

Policy 2. For queries that involve multiple levels of nesting, you must use the appropriate nesting constructs learned in class.

2 (8 pts) Who Missed the Votes?

Examine the database created by the script provided with this homework assignment. Look at the attributes of all relations, and quickly inspect the dataset before proceeding.

Deliverable 1. Return a relation containing last name, first name, and number of absences for all senators that have been absent at least once. Provide the relational algebra operations in your writeup. Save the SQL script in `question1.sql`.

Question (answer as a comment in your first deliverable). Can you quickly google search for a reason for the absences of the top two absent Senators?

3 (12 pts) “Who Missed the Votes” Strikes Back

The query above does not include senators that were present in all voting sessions. One way to include those senators is to perform a suitable join operation with a relation containing all the senator names.

However, when using this technique, the resulting relation may still have **NULL** values for tuples associated with senators with no absences. To transform those **NULL** values into zeroes, look into the MySQL documentation for **COALESCE**.

https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html#function_coalesce

Deliverable 2. Return a relation containing last name, first name, and number of absences for all senators, using the technique above. Provide the relational algebra operations in your writeup. Save the SQL script in `question2.sql`.

4 (25 pts) Agreements and Disagreements

Pick arbitrary senators x and y from different political parties, with first names f_x, f_y ; last names l_x, l_y ; and IDs I_x, I_y . For more interesting results, pick x and y with highly different

political views¹, that do not have too many absences², and that you consider as close as possible to different extremes of the spectrum. **Identify those senators and their IDs in your writeup.**

Deliverables 3-5. Provide the following:

1. A query that returns how many times x and y voted in agreement, when both senators voted. Define this number as the *number of agreements* between x and y , or $A(x, y)$. You can use the senator IDs I_x and I_y in your query directly. If x and y are the same, your query should return the total number of votes of the senator.

Provide the relational algebra operations in your writeup. Save the SQL script in `question3.sql`.

2. A query that returns how many times x and y voted in discordance, when both senators voted. Define this number as the *number of disagreements* between x and y , or $D(x, y)$. You can use the senator IDs I_x and I_y in your query directly. If x and y are the same, your query should return zero. Provide the relational algebra operations in your writeup. Save the SQL script in `question4.sql`.

3. A relation containing last name, first name, and the number of disagreements between every senator and senator x . Your relation should also include a tuple for senator x him/herself, so the relation must contain 102 tuples. You can use senator x 's ID into your query (but not last names or first names), and you are allowed to hardcode appropriate values. Provide the relational algebra operations in your writeup. Save the SQL script in `question5.sql`.

5 (25 pts) The Agreement Index

Choose p between x and y . Return a relation containing the last name, first name, political party, number of agreements between each senator and p , number of disagreements between each senator and p , and agreement index between each senator and p , as defined below.

The agreement index $I(x, y)$ between senators x and y is:

$$\frac{A(x, y) - D(x, y)}{A(x, y) + D(x, y)}.$$

¹For simplicity, assume the 1D left \leftrightarrow right political spectrum. Real life may or may not be like this.

²For whatever definition you come up with.

Order the table by decreasing agreement index.

Deliverable 6. Provide the relational algebra operations for the problem above in your writeup. Save the SQL script in `question6.sql`.

Question (included in a comment in the previous deliverable). Do you see a clear bipartisan division?

For testing: If you picked Senator x as Ted Cruz, Edward Markey's index (the one in most disagreement) is -0.7734. Ted Cruz should be listed in the relation with index 1.0000. If you instead picked Senator x as Elizabeth Warren, Ted Cruz's index (the one in most disagreement) is -0.7684. Elizabeth Warren should be listed in the relation with index 1.0000. It doesn't matter if you pick a Republican or a Democrat: the division will be clear-cut.

6 (20 pts) (due Oct 6) Static Web Page

Deliverables 7 and 8. Now, you are going to create a web-page showing the results from Deliverables 2 and 6. I expect a well-formatted webpage, using HTML and CSS, done by hand. For now, the contents of your webpage is **static**, that is, you are going to hardcode the results from Deliverables 2 and 6 into the webpage.

Deliverable 7. Even though your webpage contains the results of Deliverables 2 and 6 hardcoded on it, the generation of the static content will be done through a Python script that connects to your database, fetches the results, and prints them in HTML format. The output of this script should be HTML code purely copied/pasted into your webpage. **No manual edits should be necessary to get the HTML table into your webpage.** Please refer to the `remote_connection.py` example given in class, and naturally to your solution for HW1 in order to craft your script. Note that this script should be small – my solution is about 50 lines of Python code.

Deliverable 8. A directory **called “www”** containing your web-page with the results from Deliverables 2 and 6. Please use VSCode to edit your HTML files.

7 (5 pts) Style

The remaining 5 points are granted for meeting the following style guidelines. The writeup is organized and properly written. The relational algebra notation is clear. The SQL scripts

are well-indented and organized. The SQL scripts are concise and elegant, and do not use too many unnecessary Cartesian operations.

Good luck,
- Hammurabi