2020 Election Contribution Analysis

Nick Smailer

1/8/2024

**Table of Contents**

[Introduction 3](#_Toc160707742)

[Purpose 3](#_Toc160707743)

[Project Summary 3](#_Toc160707744)

[Requirements Definition 4](#_Toc160707745)

[Considerations 5](#_Toc160707746)

[Document Change Log 5](#_Toc160707747)

[2. Architecture Design 7](#_Toc160707748)

[2.1 Relational Data Warehouse 7](#_Toc160707749)

[Data Dictionary 7](#_Toc160707750)

[Tables schemas 8](#_Toc160707751)

[2.2 Hadoop Implementation 10](#_Toc160707757)

[2.3 Reflective analysis of using a data warehouse vs Hadoop. 11](#_Toc160707758)

[3. Data Preparation 12](#_Toc160707759)

[3.1 Relational Data Warehouse Implementation 12](#_Toc160707760)

[ETL considerations 12](#_Toc160707761)

[ETL Process Flow with description 12](#_Toc160707762)

[3.2 Hadoop Implementation 14](#_Toc160707763)

[3.3 Reflective analysis of data preparation in relational data warehouse vs Hadoop. 15](#_Toc160707764)

[4. Reporting System 20](#_Toc160707765)

[4.1 Relational Data Warehouse Implementation 20](#_Toc160707766)

[4.2 Hadoop Implementation 25](#_Toc160707771)

[4.3 Reflective analysis of result in relational data warehouse vs Hadoop. 27](#_Toc160707772)

[Conclusions 28](#_Toc160707773)

Comparing Relational Data Warehousing Versus Hadoop

Nick Smailer

# Introduction

This project will be completed week by week, one step at a time, with the end goal being a comparison between two different information systems that we created. The first part of the project will focus on creating a data warehouse using a star schema, while the second part will focus on Hadoop. Both will use the same data, information on election contributions from the 2020 United States presidential election. This document will contain information regarding the creation and implementation of each part of the project, as well as discussing the end results for each.

# Purpose

The purpose of this document is to outline the creation of two different information systems, then compare the end result.

# Project Summary

1. **Objectives**

Gain insight on two different data storage options. Compare and contrast the two and decide which suited our data storage needs the best. Determine when to use each option in regard to future projects.

1. **Scope**

Answer business questions regarding election contributions for various candidates in quarters 2 – 4 from the 2020 election using a data warehouse to store the information, then repeating the process in Hadoop.

1. **References**

*2020 Presidential Contribution Data by State*, 26 Oct. 2020, projects.propublica.org/itemizer/presidential-contributors/2020.

1. **Outstanding Issues**

n/a

# Requirements Definition

1. **Goals**

* Store historical data about campaign contributions from the 2020 election.
* Include multiple states, quarters, and candidates.
* Gain insight on trends across states and quarters for different candidates.
* Determine popular candidates across states and quarters.
* Efficient querying and reporting.

1. **Usability Requirements**

Users must be able to efficiently query aggregated information about election contributions. Users should be able to easily report on the data using Knime. Also, most users should not be able to change the data.

1. **System Security Requirements**

After adding the data to the system, it should not be modified or added to. The point of the data warehouse is to store nonvolatile information, so it is crucial that the data remains consistent. To ensure this, normal users should not have the capability to edit data, this ability will be restricted to higher level users and still should not be used except in special circumstances.

1. **Business Questions**

* What was the distribution of campaign contributions in PA from Q2 to Q4?
* Were there similar donations to Trump, Biden, and Bernie from PA from Q2 to Q4?
* How different were total donations by quarter from each state?
* How much was donated to Trump, Biden, and Bernie across the four states from Q2 to Q4?

1. **Data Requirements**

To answer the questions above, information from the 2020 election contributions will be used. The data will be selected based on state, month, and candidate. After downloading the data, it will be transformed into a format that is useful and easily able to be loaded into our warehouse and Hadoop. By aggregating donations for each state as well as the campaign they supported we can begin answering the questions above. Analyzing donations by candidate and quarter will help us answer our questions.

1. **Design Constraints**

**Project 1:**

Star schema should be followed with a central fact table connected to dimension tables.

**Project 2:**

Data will be stored in one table, as Hadoop does not operate using keys.

# Considerations

Data that was deemed not helpful to answering the research questions was not included in the warehouse.

# Document Change Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Change Date** | **Version** | **CR #** | **Change Description** | **Author and Organization** |
| 1/30/24 | 1.0 |  | Added Introduction, Purpose, Summary, Requirements | Nick Smailer, PSU |
| 2/8/24 | 1.1 |  | Completed section 2.1 and project 1 portion of 2.3 | Nick Smailer, PSU |
| 2/10/24 | 1.2 |  | Added ERD | Nick Smailer, PSU |
| 2/20/24 | 1.3 |  | Updated document to reflect changes after feedback from professor | Nick Smailer, PSU |
| 2/25/24 | 1.4 |  | Completed project 1 portions of section 3 | Nick Smailer, PSU |
| 3/1/24 | 1.5 |  | Completed project 1 portions of section 4, added charts from reports | Nick Smailer, PSU |
| 3/6/24 | 1.6 |  | Reviewed document and made final updates/changes | Nick Smailer, PSU |
| 3/10/24 | 1.7 |  | Gave document a final review prior to submitting | Nick Smailer, PSU |
| 3/22/24 | 1.8 |  | Completed sections 2.2, 2.3, 3.2, 3.3 with current progress for project 2 draft | Nick Smailer, PSU |
| 4/1/24 | 1.9 |  | Continued sections from version 1.8 as I completed more of the project | Nick Smailer, PSU |
| 4/7/24 | 2.0 |  | Added screenshots of Pig scripts and finished documenting ETL | Nick Smailer, PSU |
| 4/12/24 | 2.1 |  | Expanded upon sections 2.3 and 3.3 | Nick Smailer, PSU |
| 4/15/24 | 2.2 |  | Added descriptions to Pig screenshots | Nick Smailer, PSU |
| 4/18/24 | 2.3 |  | Completed Hadoop Reporting section | Nick Smailer, PSU |
| 4/19/24 | 2.4 |  | Added Hadoop Reporting screenshots | Nick Smailer, PSU |
| 4/21/24 | 2.5 |  | Finished conclusions and final touches | Nick Smailer, PSU |

# 2. Architecture Design

## 2.1 Relational Data Warehouse

***ERD:***

***A diagram of a computer

Description automatically generated***

### Data Dictionary

The structure of the data warehouse is a star schema with one fact table and four dimension tables. The fact table contains the primary keys of all other tables as foreign keys, as well as data for aggregation purposes. The dimension tables contain information specific to each donation- personal, location, date, and candidate. The tables outlined below explain the columns in each table- what information is stored there, the variable type, and an example value.

### Tables schemas

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name of the table*** | personal |  |  |
| **Description** | This table describes personal information about the individual who donated. | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **personalID** | ID of donator | Integer | Between 1 and 999999999 |
| **occupation** | Job of donator | Varchar | Engineer |
| **employer** | Company they work for | Varchar | Tesla |
| **org\_name** | Organization donor is a part of- blank if n/a | Varchar | Committee for Freedom |
| **…** |  |  |  |
| **Primary Key** | personalID |  |  |
| **Foreign Keys** |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name of the table*** | date\_info |  |  |
| **Description** | This table contains a variety of information about the date of the donation. | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **dateID** | Auto incrementing value, unique for each record | Integer | Between 1 and 999999999 |
| **year** | Year of donation | Integer | 2020 |
| **month** | Month of donation | Integer | 9 |
| **weekday** | 1-7 for Monday-Sunday | Integer | 2 |
| **quarter** | Quarter of year | Integer | 4 |
| **…** |  |  |  |
| **Primary Key** | dateID | | |
| **Foreign Keys** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name of the table*** | location |  |  |
| **Description** | This table describes where the donation came from. | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **stateID** | Value associated with individual state | Integer | Between 1 and 4 |
| **state** | Name of the state | Varchar | California |
| **…** |  |  |  |
| **Primary Key** | stateID |  |  |
| **Foreign Keys** |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name of the table*** | candidate |  |  |
| **Description** | This table describes the candidate receiving the donation. | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **candidateID** | Id of the specific candidate | Integer | Between 1 and 29 |
| **candidate\_name** | Name of the candidate | Varchar | Abe Lincoln |
| **…** |  |  |  |
| **Primary Key** | candidateID |  |  |
| **Foreign Keys** |  | | |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name of the table*** | fact\_table |  |  |
| **Description** | This table stores facts about the data as well as joining to all other tables. | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **personalID** | Foreign key from personal table | Varchar | A1000002 |
| **stateID** | Foreign key from location table | Integer | Between 1 and 4 |
| **candidateID** | Foreign key from candidate table | Integer | Between 1 and 29 |
| **dateID** | Foreign key from date\_info table | Integer | Between 1 and 365 |
| **amount** | Amount of specific donation | Integer | Between 1 and 999999999 |
| **count** | Created to allow for other types of aggregation | Integer | Between 1 and 999999999 |
| **zip** | Zip code of donor | Integer | 16801 |
| **…** |  |  |  |
| **Primary Key** |  | | |
| **Foreign Keys** |  | | |

## 2.2 Hadoop Implementation

The architecture of my data in Hadoop is much simpler than the data warehouse. Considering Hive allows for data to be stored in table format and our data is structured, I decided to take an approach similar to what we did in the lesson with the superstore data. Initially, I considered following the same architecture I used for the data warehouse in project 1, but even though Hive is similar to data warehousing, it does not operate using keys, so this would not have made sense. In the data warehouse, we were able to use the combination of foreign keys from the dimension tables as the unique identifier in the fact table; obviously this structure would not work in Hadoop. We also relied on this for querying, joining tables based on stateid, candidateid, etc. Using the examples from the lesson which select all using certain criteria, I learned how one can achieve the same end result without joining different tables based on keys. I find that this is a bit easier because all records are on one table, so you do not need the joins or different query nodes in Knime- one query will suffice. That being said, this can lead to complex queries to filter the data accordingly, so I made data cubes based on simpler queries for reporting which will be discussed later.

As I mentioned, I took an approach similar to the lesson, so after processing my data with Pig (which I discuss in later sections) I created my database and one table called contributions\_data. While creating the table, I only created fields for the fields I included in project 1 because I maintain my decisions regarding the relevant and irrelevant fields that I made using Kimball’s process for project 1. I also decided not to include some of the columns that I did not use in the first project. For example, I am mainly interested in changes by quarter, so I did not include the fields weekday, month, and year that I created during project 1. As a result, the datasets are not exactly the same, but I don’t think they need to be because the fields I eliminated were not used to answer my business questions, and Hive does not operate using keys. I initially intended on creating the quarter field in Pig, but ended up doing that in Hive, so I had to create another table called contributions1. To create this table, I copied all the data from my original table, and added the quarter field in the process using a case statement to define the criteria for each quarter.

Table schema:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name of the table*** | contributions1 |  |  |
| **Description** | This table stores all the campaign donation information for the selected states across the selected quarters | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **transaction\_date** | Integer assigned to each day within the dataset | Integer | Between 1 and 365 |
| **amount** | Amount of specific donation | Integer | Between 1 and 999999999 |
| **count** | Created to allow for other types of aggregation | Integer | Between 1 and 999999999 |
| **zip** | Zip code of donor | Integer | 16801 |
| **occupation** | Job of donator | Varchar | Engineer |
| **employer** | Company they work for | Varchar | Tesla |
| **org\_name** | Organization donor is a part of- blank if n/a | Varchar | Committee for Freedom |
| **state** | Name of state | Varchar | Pennsylvania |
| **candidate\_name** | Name of candidate | Varchar | Abe Lincoln |
| **qtr** | Quarter donation was received in | Int | 3 |
| **…** |  |  |  |
| **Primary Key** | n/a |  |  |
| **Foreign Keys** | n/a |  |  |
|  |  |  |  |

## 2.3 Reflective analysis of using a data warehouse vs Hadoop.

I thought using the data warehouse was very logical. When discussing the architecture, I enjoyed the process of grouping related fields together and determining what should go where. Those decisions are crucial, so it was interesting to see what fit together where to create the warehouse. I already had experience with RDBMS, so I was already familiar with a lot of the ideas and thought processes that go into it. Kimball’s process was new to me, and the lessons discussing it really helped me to identify what to include in my warehouse, which also applied to Hadoop.

In regard to working with Hadoop, things were a little more complicated. This was mainly because I am familiar with the traditional RDBMS structure which coincided well with the data warehouse, but not so much Hadoop. Also, Hadoop, Pig, and Hive all have different keywords and operators that I had to get accustomed to. When specifically comparing the architecture of the two, I believe the architecture I created in Hadoop is much more simplistic and easier for querying but provides less structure than my data warehouse. The structure is important to me, because in our example there are less than 10 fields in Hadoop, but what if there were 100+? In that case does it make sense to store everything on the same table? Obviously, there are lots of tradeoffs between the two data storage options, so it was interesting to how the differences present themselves in terms of architecture and the effects those have downstream.

3. Data Preparation

## 3.1 Relational Data Warehouse Implementation

## ETL considerations

My main concern prior to beginning ETL was defining what data was important and what data was not important for answering my business questions. I did not want to include irrelevant data in my warehouse, so before starting anything I defined what attributes I wanted each dimension to have and what I wanted to store on the fact table. My decisions can be seen in the image below- screenshots from Excel documenting my choices following the reduced version of Kimball’s process.

## ETL Process Flow with description

To begin, I downloaded the second, third, and fourth quarter campaign donations datasets for Alaska, Hawaii, Oregon, and Pennsylvania. I then added a quarter field to identify the quarter the donations came from. Next, I used Power Query in Excel to combine the datasets by quarter to be loaded separately and eliminated the attributes I deemed unnecessary. Next, I created the primary keys for the dimension tables. For personalID I was able to repurpose transaction ID, but I noticed that out of the 200,000+ observations I had, there were a little over 1,000 duplicated transaction IDs. Considering this was less than 0.5% of the data I removed the duplicates and continued. I was also able to use the existing date field as dateID. I decided on a daily grain focusing on quarterly totals. I assigned each candidate an ID (1-29 for the 29 campaigns represented) and assigned each state an ID (1-4 for AK, HI, OR, and PA respectively). I also created month and year fields for the date\_info table. All of this was completed in Excel prior to importation into Knime. I also created the data warehouse structure in DBeaver before loading the data into Knime. I first created the tables one at a time which was easy, but I had a hard time creating the columns. Eventually, I found the button to do so, but for a bunch of the columns I chose \_int4 instead of int4, and \_int4 is for an array of integers not just one integer. As a result, I had to eliminate and recreate those columns. Eventually, I matched the tables up to the table structure seen above, and considered the preliminary steps to be completed.

To load the data, I used Knime, and I used a similar workflow to the one we used with our example warehouse. I first read the dataset in using an Excel reader node, then I used column filter nodes to divide the data into groups based on the tables they belonged to (personal, date\_info, candidate, location, fact\_table), then I grouped them by their unique IDs using a group by node. Next, I used column renamer nodes to ensure the column names matched the names in the data warehouse. Finally, I established the PostgreSQL connection and used a DB Writer node to load the data into DBeaver. Below are images of my decisions regarding the data and my workflow in Knime. In the image, only the final column renamer is connected to the DB Writer node, but all other column renamers previously held that connection, as well.

A white and black text on a white background

Description automatically generatedA table of information

Description automatically generated with medium confidenceA diagram of a computer

Description automatically generated

## 3.2 Hadoop Implementation

To begin ETL in Hadoop, I first redownloaded the individual datasets from the ProPublica website. I did this because the instructions for project 2 said it was preferable to do it this way, and I want to gain experience transforming the data using Pig. After downloading the datasets, I put them in a zip file and put that in my temp folder that we used in the lessons. From there, I started my Hadoop container, and navigated to my temp folder within command prompt. Next, I first created a contributions directory in Hadoop, and tried to copy the folders into it, but I ran into some challenges (Figures 1 & 2). I was able to confirm the directory existed, but I was really struggling to find the files even though it said they had been moved. As a result, I went back to the lesson where we went over loading data in Hadoop and I realized I had skipped a few steps.

After realizing my mistakes, I went back and restarted the process. I first created a directory called donations and confirmed its existence. I then copied the zipped file to the tmp folder I made previously within the container. Within the tmp directory, I unzipped the file and tried to view all within /tmp/Data, but I could not find them. I searched for them, and found them within a subdirectory of tmp, so I moved them to the Data subdirectory. Next, I copied all of them to the contributions input subdirectory (Figure 3). I ended up doing this again in my final version, so some of the initial screenshots may reference a directory called contributions while other reference one called donations (donations being the second one I created). From there, I loaded the datasets one at a time into Pig. At first, I tried doing this in batch mode, but it kept hitting errors, so I decided interactive was the best way to go that way I could easily identify any issues. While I am not sure of this, it seemed to me that Pig was struggling to combine the datasets (using UNION) because they all had a header line. As a result, after I imported the datasets, I created line numbers like we did in the lesson using rank, and then removed the first line using filter (Figure 4). After removing the header line, I was able to successfully combine the datasets, but that is where I began to have a hard time.

While I now understand what was going on, I was initially very confused why multiple output files were being created. I saw my union was successful, but when I checked the output, I saw eleven files when I had started with twelve. Immediately I thought somehow the union did not work, or the files were too large, so I tried again combining the data by state to create smaller files. This did not solve the issue, so I took to the Internet. I found someone who had the same issue on Stack Overflow (I will provide the link below) and the responses helped get me back on track.

First, I set the parallel process equal to one, because that seemed to be the main factor in my troubles. After creating the union, I used the distinct function in order to ensure there were no duplicate records. I then grouped the records by state and flattened the output (Figure 5). Next, I stored the file in my output directory and looked over it to ensure it had worked as expected. From there, I went back into Pig and loaded the file I had just created. Next, I removed duplicates using the distinct function in Pig, because I saw in the first project there were repeated transaction IDs. Using the new dataset, I removed the fields I deemed unnecessary using for each and generate to create another dataset made up of only the columns I wanted. My motivations for this were to create a dataset similar to that of project 1, while storing only the data that will help answer my business questions. I also defined the schema to allow for easy importation to Hive (Figure 6). After that, I stored it in a different output directory. Finally, I created the database and the table in Hive, and imported my dataset (Figures 7 & 8). Below are pictures outlining many of the steps I described. Unfortunately, I forgot to take a few screenshots during the process, for example loading the data into Hive.

## 3.3 Reflective analysis of data preparation in relational data warehouse vs Hadoop.

I thought the ETL process for the data warehouse was fairly straightforward and efficient. I knew beforehand that ETL takes a long time, and it is crucial to define what does and doesn’t need to be included prior to starting, so the length and workload associated with this step did not bother me.

I have a lot of prior experience using Excel, and since the datasets were extracted as Excel worksheets, I chose to stay in Excel for the transformation stage, as well. While Excel was easy for me to work with because I already knew how to do what I wanted to do, it definitely was having issues handling such a large dataset on my PC. I did not encounter this issue with Knime, and after working with Knime more, I realized I probably could have done everything I did in Excel in Knime. The loading stage in Knime was intuitive, and following the workflow from the lesson provided me with enough understanding to successfully load the data into my warehouse one table at a time. I thought the process was logical and I liked how after establishing the connection you can write queries in Knime. Overall, I am happy to have learned this efficient method of loading data for relational databases and data warehouses.

While I had a fairly easy time loading the data into the data warehouse, loading it into Hadoop was not the same. Downloading the data was simple; I had deleted the files after making my changes during project 1, so I had to go back to the ProPublica site and repeat the downloads. I had no trouble navigating to the folder where I stored the zip file within command prompt, but from there I began to run into challenges. I made some mistakes starting out because I thought I remembered the process from the poems example, but I was incorrect. After referring to the lesson I was able to get back on track, but I still had a bit of a hard time. When I unzipped the poem file, the poems dropped right into tmp/Data, but that was not the case with the contribution data. I am not sure exactly where they went, but I had to find them in a subdirectory within tmp and move them into Data. This was a little difficult for me because I am still learning some of those more advanced docker commands as well as the syntax.

After successfully creating the donations directory and unzipping the files in the containers tmp folder, things began going a little smoother. Initially, I began copying the files into the donations input subdirectory one at a time, but then I realized I could select all .csv from Data, so I ended up doing that. After that, I checked the contents of the subdirectory, and I could see all my files within. Next, I began the transformation. I outline the steps I took above, so here I will discuss my thoughts and what I learned.

Transforming the data in Pig was hard for me. Part of the reason why is because I had a specific way I wanted to do it. I wanted to import the datasets individually, then combine them, then remove the unnecessary columns in a similar fashion to what I did in Excel. While there are likely easier ways to do this that would have saved me time, I really wanted to figure it out my way. It took lots of trial and error, which was frustrating at times, but it allowed me to understand a lot more about Pig and Hadoop that I previously did not know. I was able to do ETL how I wanted, except for one part. I wanted to add a column storing the quarter which would be derived from the transaction date. Ultimately, after a few attempts I decided for time’s sake to add the column in Hive. All things considered; I am happy that I was able to do almost everything the way I wanted to. One of the main challenges was identifying issues. When I tried to execute my code in batch mode, it would provide a line number with the error, but that does not necessarily mean the error was on that line or only on that line. This led me to run my lines interactive instead, so I could see exactly where it failed, which was tedious because I had to re-run each line if I exited Pig, but it worked. My main takeaway was the ease of using Pig to process big data versus Excel which I used in the first project. When I tried to manipulate the combined dataset in Excel, even the slightest tweaks made it freeze and not respond which was extremely frustrating. Pig, on the other hand, easily handled these transactions and allowed me to work through my issues and rework things that would have taken forever in Excel. All things considered, my main difficulties were because Pig and Hadoop were both new to me, so there was a bit of a learning curve. After gaining some comfort using these tools, I think they are much better equipped to process big data (which I think is the general consensus). Loading the data into Hive was the easiest step in this process for me.

Unlike the other steps, loading the data into Hive only took me one try, and I did not struggle at all. After importing the data into Hive, I tried adding the quarter column. I was able to create it, but when to populate it based on the transaction\_date field I learned those types of operations are not supported. As a result, I just created a new table with the same data, except this time I created the quarter field while creating the table, so it was all one step. One small issue I ran into was the connection to Hive being refused repeatedly. All I had to do was keep running the bl line and eventually it was able to make the connection, but I honestly have no idea why it was having that issue. Overall, with both the data warehouse and Hadoop, loading the data was quite straightforward and easy.

One the whole, I had an easier time completing ETL using the data warehouse, but ease of use is not the most important factor. Comparing the two processes showed me that there is a give and take on both sides and it ultimately comes down to user preference and particular use case. Hadoop is certainly a better option for the data transformation stage, at least in comparison to Excel. That being said, I think a data warehouse is a slightly better storage option than Hadoop because of the structure it provides, but that is just personal preference. One drawback of the structure provided by data warehousing compared to Hadoop is querying which will be discussed later.

Figure 1: First, I start the container, then I enter bash mode, then I create the directory the data will be imported to. I ended up redoing this and using a directory called donations, but did not take a screenshot.

A black screen with white text

Description automatically generated

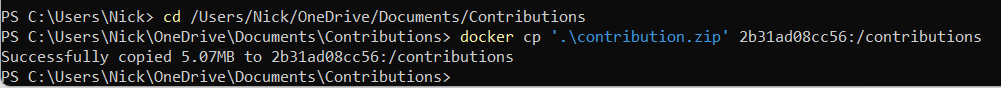
Figure 2: Opening the folder containing the dataset on my local machine, then moving it into the directory created in the previous step. This did not work the way I intended, I ended up importing them into my tmp directory, I just forgot to take a screenshot so I included this one because it was the same procedure.

Figure 3: Initial attempt was unsuccessful, so I repeated the previous steps except first stored the data into tmp. The image below shows moving the files from tmp into the donations input directory, then checking to make sure they are there.

A computer screen shot of a program

Description automatically generated

A black screen with white text

Description automatically generatedFigure 4: Importing the datasets one at a time into Pig. This was achieved in a three-step process. First, load the dataset, then add line numbers (the rank line), then filter out the header line (filter line). This is just a picture of the steps applied to the last three datasets, but this was done for all twelve.

Figure 5: In this image I join the datasets using the union operator, then group them by state, then flatten the result. Finally, I store the output into my output directory. Here is the link to the Stack Overflow page I learned this from:

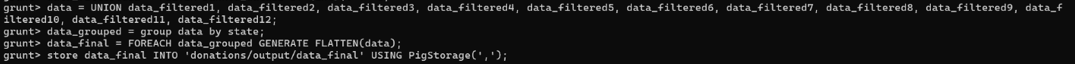
<https://stackoverflow.com/questions/10954883/storing-results-of-union-in-pig-in-a-single-file>

Figure 6: This image shows the schema definition for the cleaned data. Before this, I imported the combined dataset, used distinct to eliminate duplicate records,

A screen shot of a computer

Description automatically generatedand then generated a new dataset containing only the columns deemed useful. Unfortunately, I ran a dump command to inspect the results and could no longer view the lines I ran which is why they are not included in the image.

Figure 7: Here is the cleaned data stored in my output directory.

A computer screen with white text

Description automatically generated

A computer screen shot of a black screen

Description automatically generatedFigure 8: Here I create the database and the table.

Figure 9: To create the quarter field, I was forced to create a whole new table. I used the case expression because otherwise I would have had multiple nested if clauses which I felt did not lend itself well to this situation. I used the substring function to isolate the month, and then assigned the quarter based on the month.

A black screen with white text

Description automatically generated

4. Reporting System

## 4.1 Relational Data Warehouse Implementation

When approaching the reporting part of this project, I sought to answer my research question using charts and graphs that are easy to look at and interpret. I used a workflow similar to the one we saw in our lesson to join the tables and generate reports (Figure A). I started by looking into PA’s contributions from the second quarter through the fourth quarter. Initially, I wanted to look at the contributions to all candidates, but considering there were 29 and some received very little, I decided to only include those who received at least 0.1% of the total contribution. That is still a very small portion, but it filtered out a decent amount of the initial 29. For the line graph displaying the state total contribution, I included all candidates, I just did not think it made sense to do the same for the pie chart.

I then narrowed it down to just Trump, Biden, and Bernie. I compared their total donations received across those quarters, then I tracked the changes in the donations they received over the three quarters (Figures B-E). After looking at the results, I was able to answer my first two research questions. When looking at the total distribution of donations from the second quarter through the fourth, it becomes evident that Biden, Trump, and Bernie received the most money from donations, in that order. While this may be fairly obvious considering they were the most popular candidates, I was not aware of this when choosing these three for my business questions, so I found that a little humorous. Having mentioned that, I will now discuss my findings in regard to their campaign contributions from PA across Q2-Q4.

Initially, I focused on the total amount of money received across those three quarters, then I explored the trends of those three candidates across the quarters. When comparing their total donations, Biden received about as much as Trump and Bernie combined which makes sense considering he received the most out of any candidate. Next, looking at the contribution trends across the three quarters, all three displayed different trends. First, Biden started in Q2 with almost $1,000,000 received, then received only about half of that in Q3, then bounced back in Q4. Trump, on the other hand, started out at about $400,000 in Q2, then doubled that in Q3, then plummeted in Q4. Finally, Bernie started out with around $150,000 in Q2 and steadily increased to almost $500,000 in Q4. After focusing heavily on PA, I shifted to my next two business questions which had a broader scope.

One of my final business questions was about the overall composition of donations from Alaska, Hawaii, Oregon, and Pennsylvania in the second through fourth quarters. To answer this, I took a similar approach to my comparison of PA donations across the quarters. I first created pie charts for each state in each quarter Q2-Q4 and compared them to see the overall distribution per state. Then I compared the distributions to that of other states. My findings for Q2 were that Trump and Bernie clearly had the most donations in Alaska, Hawaii, and Oregon, but as I mentioned above Biden had the overwhelming majority in Pennsylvania. Alaska and Hawaii had similar results in Q3, but Oregon saw less donations to Trump and surges from other lesser known (to me at least) candidates. In Pennsylvania Trump greatly increased while Biden greatly decreased. In Q4, Bernie dominated the donations in Alaska, Hawaii, and Oregon. Oregon also saw lots of donations to Friends of John Delaney. Finally, about half of the total Q4 contributions from Pennsylvania went to Biden and Kamala separately, with Bernie having the next most behind those two. Based on the pie charts, it seems like across all four states Bernie had a very strong presence while Trump and Biden fluctuated. Oregon was the only state that saw anyone outside of those three receive large contributions (not including Kamala for PA in Q4 because I consider her to be under Biden’s umbrella since they ran together). Having gained lots of insight into the distribution of donations across the four states in three quarters, I felt that I answered that research question and moved to my last one.

My final research question was how the total amount of donations fluctuated over the three quarters across all four states both on the whole, and with Trump, Biden, and Bernie. To answer this, I used line graphs to compare the different states, then the candidates by state (Figures F-I). When comparing the total donations by state, Pennsylvania had the most by a lot, followed by Oregon, then Hawaii and Alaska which had very similar figures. All four states had an upward trend. These values make sense when the population of each state is taken into consideration. Next, I compared the donation trends of the selected candidates by quarter. Alaska, Oregon, and Hawaii had very similar trends across the quarters. Biden started low, then remained consistent in Q3, then increased in Q4. One minor difference here is that in Hawaii Q3, Biden saw a noticeable decrease, but Alaska and Oregon his donations were very similar. Bernie increased steadily across all three quarters, with a large jump coming in Q4. The only major difference between the two was the starting point of Trump. In Q2 for Oregon, Trump and Bernie were about equal, but in Alaska, Trump was well ahead of the others. Following Q2, both saw Trumps donations increase in Q3, then plummet in Q4. For analysis on Pennsylvania’s trends, see above.

Notes regarding charts below:

I chose to only include snippets from the reports I generated for the sake of space. Additionally, I did not include the pie charts for the states in each quarter for the same reason. Also, the line graphs provide a good state comparison across the quarters.

A diagram of a computer

Description automatically generatedFigure A

A pie chart with many colors

Description automatically generatedFigure B

Figure C

A graph with a line and a blue dot

Description automatically generated

Figure D

A pie chart with text and numbers

Description automatically generated

Figure E

A graph of different colored lines and numbers

Description automatically generated

Figure F

## 

## A graph of a number of donations Description automatically generated

## Figure G

A graph of a graph with colored lines and dots

Description automatically generated with medium confidence

## Figure H

A graph of a donation

Description automatically generated with medium confidence

Figure I

## A graph of a number of colored squares Description automatically generated with medium confidence

## 4.2 Hadoop Implementation

To begin the reporting on the data from Hadoop, I first set up a workflow in Knime to match what we learned in Lesson 13. I started with a Hive connection node which connects to a DB Query Reader node, which connects a Data to Report node. After that, I created the Hive connection in the first node, then ran a select \* query in the second node to ensure it was working as expected. After that, I adjusted my query to only include records containing donations to Bernie, Trump, and Biden. Next, I entered the reporting window and created a data cube with groups of candidate and state and the summary field being amount (Figure N). I then created two bar charts comparing donations to those three candidates by state (Figures J & K). After that, I created another data cube using amount for the summary field again, and using state, candidate, quarter, employer, and occupation for the groups (Figure O). I really wanted to just investigate donations by quarter as per my business questions, but adding the other fields provided flexibility. I then made two more bar charts to show donations to the total donations to the selected campaigns by quarter, and the total donations by each state per quarter to the selected campaigns (Figures L & M). While I already knew generally what to expect from project 1, this is a different type of graph than I used in project 1 so it provides some visual variability. Additionally, I did not graph the total distribution again because I felt that would be redundant. Going along with that, I am not going to detail my findings in this section because I already went into depth on the answers to my business questions in section 4.1 While reporting in Hadoop, my main goal was providing a different perspective to answer the business questions than the one provided using the relational data warehouse. I achieved this by varying the chart types to include bar charts, while with my relational data warehouse I only used pie and line charts.

Figure J

A graph of candidates for the presidential election

Description automatically generatedA graph of different colored bars

Description automatically generated

Figure K

A graph of different colored bars

Description automatically generated

Figure L

A graph of different colored bars

Description automatically generated

Figure M

A graph of different colored bars

Description automatically generated

## Figure N

## A screenshot of a computer Description automatically generated

Figure O

A screenshot of a computer

Description automatically generated

## 4.3 Reflective analysis of result in relational data warehouse vs Hadoop.

Comparing the end results of the two projects is a little bit difficult. This is because the first project answered my business questions, so the results in the second project were not new information. That being said, I found new ways to present this information which I found valuable. I found reporting to be very straightforward using the Hadoop version, just because only one query was required. Also, I only needed three nodes in Knime. While my workflow for project 1 was not extremely complicated, there were many more nodes, and I had to query each table. From a time standpoint, the Hadoop version was much quicker because I did not have to worry about adjusting multiple nodes and joining tables on certain criteria. Another thing I liked better about the second project was the data cubes. I’m sure these could have been used during project 1, but I was not aware of that tool yet. I enjoyed how this allowed me to make a subset of the data containing exactly what I wanted to chart, even if I ran a select all query.

Even though I liked the simplicity of reporting for project 2 better, I prefer my results from project 1. At the end of project 1, I learned so much about the dataset and campaign contributions from my selected states to my selected candidates. I was able to directly answer my business questions with a variety of informative graphs and charts that someone unfamiliar with the project would be able to understand. I enjoy this process of eliciting the desired knowledge from a huge dataset, I find it rewarding and valuable. Considering my business questions had already been answered prior to project 2, this aspect was sort of removed from the project. I did enjoy trying to find new ways to present the information, but at that point the findings were not new to me, which was part of the excitement of project 1. When I designed those business questions, I was genuinely curious about the answers, so it was satisfying to see my work pay off with the answers at the end of the project. I had a similar experience regarding the data transformation in Pig where I had a very specific plan in mind and it was rewarding to see it realized, but the reporting stage lost some excitement because I was no longer curious. Obviously, this is a reflection on myself, not the data storage option, just something I thought may be valuable to include in my discussion. All things considered, I could have created exactly the same reports using either tool, so to me, in terms of reporting the options are interchangeable.

Conclusions

**Project 1:**

Upon completing project 1, I have a few major takeaways. First of all, this was my first major project using Knime and I learned quite a lot about its capabilities and usage. I have used similar drag and drop applications before, but more for data mining than ETL and reporting. I thought it was very user friendly, and simplified things that would have been difficult to achieve by other means. As a result of that, I now feel confident in using Knime and would recommend it to others undertaking a project like this. I also was new to DBeaver, so while I feel most RDBMS interfaces are similar, I am happy to have the opportunity to learn as many as I can just in case I run into situations where one would perform better.

Another takeaway I had was how valuable hands-on experience and trial and error are. Throughout the course of this project, as I mentioned above, I was using applications that were new to me. If I was just reading instructions or looking at a tutorial, not actually doing the tasks required for the project, there is no way I would feel as comfortable as I do using these tools. Also, there were a few times I was struggling with certain things like how to add a column in DBeaver, or why my charts weren’t rendering in Knime, and I feel like the trial and error involved in that helped me learn, as well. While those were only small issues that were simple to fix, it does prove a point that there will always be challenges to overcome even in the steps you thought would be “easy”.

I also learned a great deal during the ETL stage. I created a relational database in Oracle in a class I took last semester, but the data was all fictious, we made it up and entered it one record at a time. While that satisfies the L portion (loading), I did not have experience with extracting or transforming. Determining what data to keep based on the business questions and goals of the project required lots of careful review and consideration which is integral to this process. I believe making those decisions and loading using a real dataset of a very large size was a great way to take the concepts and skills I learned last semester one step further. I also enjoyed the information gain seen in the reporting stage.

One of my goals for the system was to generate reports that are easy to interpret for the end user. In certain situations, the individual viewing the chart may have very little background information, so it is crucial to maximize information gain while keeping it as simple as possible. For example, if I would have chosen a line chart instead of pie chart to represent the 29 candidates across quarters 2-4, it would have looked very chaotic and confusing. Even with the pie chart, some of the colors are very similar and can be difficult to differentiate. I felt that for the most part I was able to choose options that provide valuable insight while also needing little explanation to interpret.

Finally, I apologize because I misread one part of the instructions. I thought the instructions asked for a candidate from at least two parties, not at least two candidates from each party, which is why I selected Trump, Biden, and Bernie. I hope my investigation and discussion of the overall distribution of donations by state satisfies this requirement, but I understand if I lose points as a result of this. I am looking forward to further developing my knowledge as we move into project 2 and the second part of the semester and can correct this mistake going forward at your request.

**Project 2:**

After completing project 2, I feel as though I have gained a lot of knowledge and experience using a variety of data processing tools. Hadoop, Pig, and Hive were all new to me prior to this project, so going in I was completely unfamiliar with those tools. First, I will discuss some of my takeaways about Hadoop. One major benefit that I found useful and advantageous is the ability to enter the other tools using bash commands. For example, you can enter Pig or Hive by running the pig or bl commands respectively. This is very convenient when compared to having to open a whole new application. It also allows the ability to toggle back and forth between the tools efficiently. I found the importation and storage capabilities to be quite user friendly, as well. It took me a little bit to familiarize myself with the commands, for example mkdir, fs -ls, fs -cat, etc, but once I learned how to operate within the container, things became much easier. In terms of data storage, one thing I liked and leveraged was the directory and subdirectory capabilities. I created a directory for the dataset, then an input directory where I stored the data, and corresponding output directories for whatever I had done with the data (for example, joining the data, cleaning the data, etc.). I felt like this allowed me to save my work one step at a time, keep my work organized, and remember where everything was stored. Now to discuss Pig.

I found Pig challenging, but extremely useful. The big data processing capabilities are too good to be ignored. When working with the dataset in Excel, any interaction with the data caused the program to freeze. Pig on the other hand could handle transactions affecting the entire dataset without any difficulties. This dynamic processing power is very desirable and outweighs a lot of the struggles I discussed previously. I had a very specific course of action in mind in regards to data processing with Pig, so while I hit bumps in the road, it was very rewarding to eventually achieve what I had envisioned. Finally, I will discuss my experience with Hive. I really did not do too much with Hive, so I found it easy and straightforward. Since Hive is similar to RDBMS except without the key structure, lots of my previous knowledge translated well into using Hive. I feel like on the whole, this project provided lots of exposure and the hands-on experience necessary to understand how to leverage these tools. That being said, I’m sure there is tons more I still stand to learn with Hadoop, Pig, and Hive, but I am happy to have completed a project such as this using them. One tool I felt like I used with ease this time around was Knime.

I was still getting the hang of Knime during project 1, so while I think I did a good job with reporting, I did not necessarily do an efficient job. This time around, my workflow was much more simplistic, as all the processing was done prior to importation into Knime. Having built my knowledge of Knime throughout the course of the semester, I leveraged the data cube option to create subsets that were easy to report on. This allowed me to quickly generate reports that are easy to interpret and answer my business questions in a different way than project 1.

**Overall:**

When it comes to comparing both projects, there is a lot to think about. While I enjoyed the first project more than the second, that is not the most important thing to consider. First and foremost, I think about what I learned. Data warehousing was new to me, so learning how to construct a data warehouse and how to query the data and report on it was interesting and absolutely useful. That being said, I already was familiar with SQL and RDBMS. I think this is likely why I liked this project more- it was a natural extension of my existing knowledge. I was able to build on my skills in a way I was comfortable with. The second project, on the other hand, got me out of my comfort zone a bit. I say that mainly because I was brand new to the Apache tools, so every step of the way I was learning something new and doing things I had never done before. While this was challenging at times, as I mentioned with project 1, I firmly believe trial and error is a great way to learn and retain what you learned. Now that I have compared my thoughts about what I learned, I will discuss my findings on how the data storage options compare.

To me, the difference in storage options presents itself mainly in the schema of the system and querying. I enjoyed the structure of the relational data warehouse in comparison to the Hadoop counterpart. I found it very logical to separate the data into dimensions, then return the data using the keys from said tables. While I view that as an advantage, it creates a disadvantage when it comes to querying. To return the desired data from the relational warehouse, the tables must be joined. This is not the case in Hadoop. Hive does not operate on using a key structure, and in my database, I used only one table. This allows users to write queries that will return the same information without joining tables. This leads to more efficient queries that are easy to follow and interpret. Overall, I think the advantages and disadvantages of either option juxtapose each other well. The structured environment of a relational data warehouse forces more difficult queries, and while the queries are easier in Hadoop, there is much less structure. It all depends on what you want to prioritize and sacrifice. Finally, I will compare the output of the two options.

I found reporting on the data to be fairly intuitive for both projects, but I definitely had an easier time with project 2. All things considered, I think this is more of a reflection on the development of my skills in Knime, not the data storage method. The main difference is what I mentioned above regarding querying. For project 1 I had to join the tables prior to reporting, but in project 2 I just had to select what I wanted. To me, that gives a slight upper hand to the Hadoop storage option because ultimately it allowed for easier reporting for the end user. As far as the information returned and the actual reports are concerned, the results were the same. I relied heavily on the keys for reporting in my first project, but in the second I just filtered my queries accordingly and achieved the same result.

Throughout this semester, I enjoyed the learning experience of these projects. I felt that they provided a thorough and comprehensive way to apply the concepts learned in the weekly lessons and compare two widely used data storage options. In the end, I personally am more partial to the structure provided by the relational data warehouse, but the data processing power and querying ease of Hadoop makes a strong case for Hadoop. Ultimately, my decision would depend on the use case, and I am happy to have gained the knowledge and skills required to create both options.