Assignment #2

Paulvir Dhanda
CMPT459
Simon Fraser University

June 22, 2020

Question 1

1.1

If we let:

$$\begin{aligned} Q_2 &= \textit{Count of every unique token} \\ Q_1 &= \textit{Count of different type of token (noun, verb, adjective..)} \\ Q_3 &= \textit{Count of the unique verb token} \end{aligned}$$

Using the above example, Q_1 is a roll-up of Q_2 and Q_3 is a drill down of Q_1. If there are 1000 different keywords in the data set then using the following formula from the text book we can calculate the number of cuboids in the data cube:

$$\textit{Total number of cubiods} = \prod_{i=1}^{n}(L_i + 1)$$

Since each token is it's own dimension, we can assume that $L_i = 1$ for each dimension. Thus we end with 2^{1000} cuboids in our data cube.

1.2

Consider a set of photos enhanced by some attributes, such as location and time. If we had an AI tool that can identify people in photos, An interesting OLAP query would be to calculate average number of people per photo per city.

1.3

Three interesting Queries in that specific situation would be:

$$\begin{aligned} Q_2 &= \textit{Minimum number of verbs used in Magazines vs. Newspapers} \\ Q_1 &= \textit{Average number of different tokens in the caption per photo} \\ Q_3 &= \textit{Count of pictures used in Newspapers vs. Magazines} \end{aligned}$$

The Demension would be: **Newspapers, Magazine, Images and Text**

Question 2

For this question we start with the base table that looks like this:

Sales

Key	Province_Territory	Sales
S01	BC	$ 1.00
S02	BC	$ 1.00
S03	ON	$ 1.00
S04	QC	$ 1.00
S05	YT	$ 1.00
S06	NT	$ 1.00
...	...	...

Now we separate our bit index tables in two tables. One for provinces one for territories that look something like this:

Territories:

SaleID	YT	NT	NU
S05	1	0	0
S06	0	1	0
...	...	...	...

Provinces:

SaleID	BC	AB	SK	MB	ON	NB	QC	NS	PE	NL
S01	1	0	0	0	0	0	0	0	0	0
S02	1	0	0	0	0	0	0	0	0	0
S03	0	0	0	0	1	0	0	0	0	0
S04	0	0	0	0	0	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...

This way for every sale in a province we only store 10 bits for location and for every sale in a territory we only store 3 bits for the location. Hence, reducing the amount of bits stored overall.

To calculate the total sales in BC all we have to do is look into the provinces table and look up SaleID's that correspond to BC. When we get those SaleID's we aggregate the sum of the total sales in the Sales table.

To calculate the total sales in BC, ON, and NT altogether we must join on the provinces and territories tables to get the SaleID's that come from BC or ON or NT. Once we have those SaleID's we can again then aggregate the total sales on the Sales table.

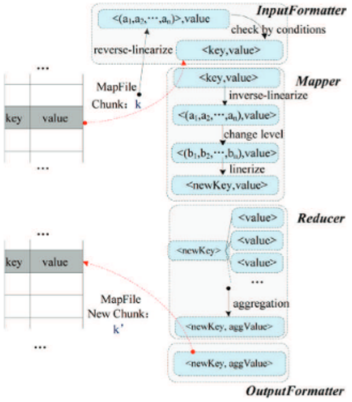

Figure 1: Caption

Question 3

3.1

For this question I had to reference this paper and from that paper I found Figure 1. From this figure I constructed a version of map reduce procedure as follows:

```
func map(row, keyFormat):
    key = row.key()
    KeyTuple = unlinearize(key) // from (key,value) to (a_1, .. a_n)
    newKey = change_level(key, keyFormat)
    newKey = linearize(newKey) // from (a_1, .. a_n) to (key,value)
    newValue = row.query(newKey)
    return (newKey, newValue)

func mapReduce(table):
    rows = table.rows()
    for row in rows:
        for key in row.demension.levels():
            obj = map(row, key) // get map key value pair
            reducedObj = reduce(obj, table) // simply aggregate
            insertIntoCube(reducedObj) // insert value into datacube

Note: func reduce was left out here because it is simply an aggregate on the measure.
```

3.2

The total number of key, value pairs the mapper emits is $O(n * L)$ where n is equal to the number of record in the Table and L is number of levels of our cuboid. Note, This can be sped using parallelization techniques for each individual level.

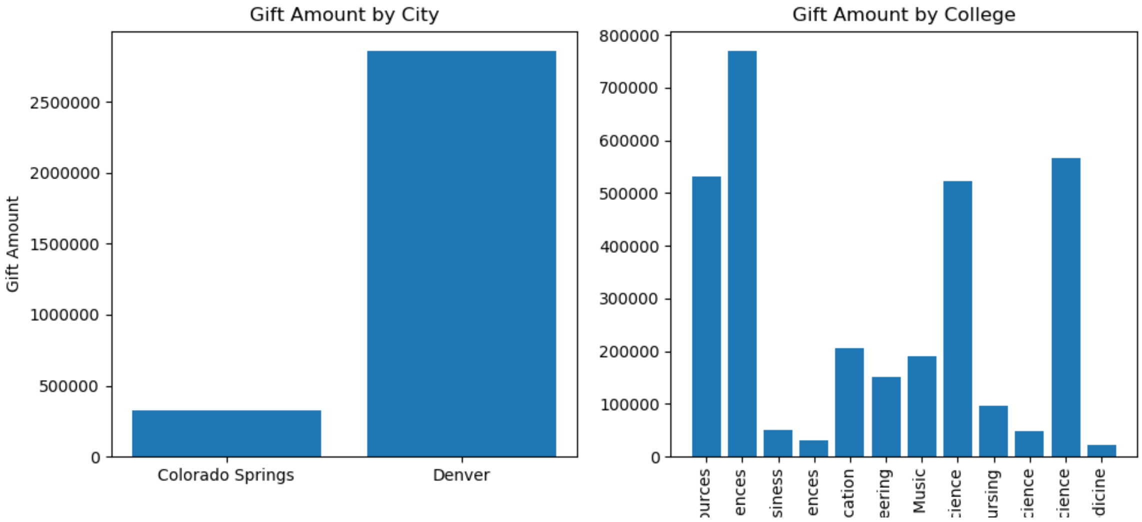

Figure 2: Gift Amount by College and City in CO

Question 4

For these question there is code supplied with this document. I will make references to files from the code.

4.1

For this section, I constructed the graph in Fig 2 using the code in `./4.1` directory of the code. I first found out that the state **CO** was the state with the largest total gift amount. I then constructed the graphs above to see how cities and colleges effect the gift amount on their own. The marker can see how this was done by running `main.py` in `./4.1` directory.

The conclusion I arrived to by visual inspection is that Denver by far contributed out contributed Colorado Springs to the gift amount for the state of **CO**. I has more than double the amount of gits than the other city. However, no one college stands out as the main contributor. This again was done by visual inspection of the graph.

4.2

For this section I constructed a file called `pairs.csv` that has all the pairs as instructed by the question. I also used the origonal 6 dimensional data as an aside. Details on how this was done and how the marker can run themselves is in `./4.2/README.md` file.