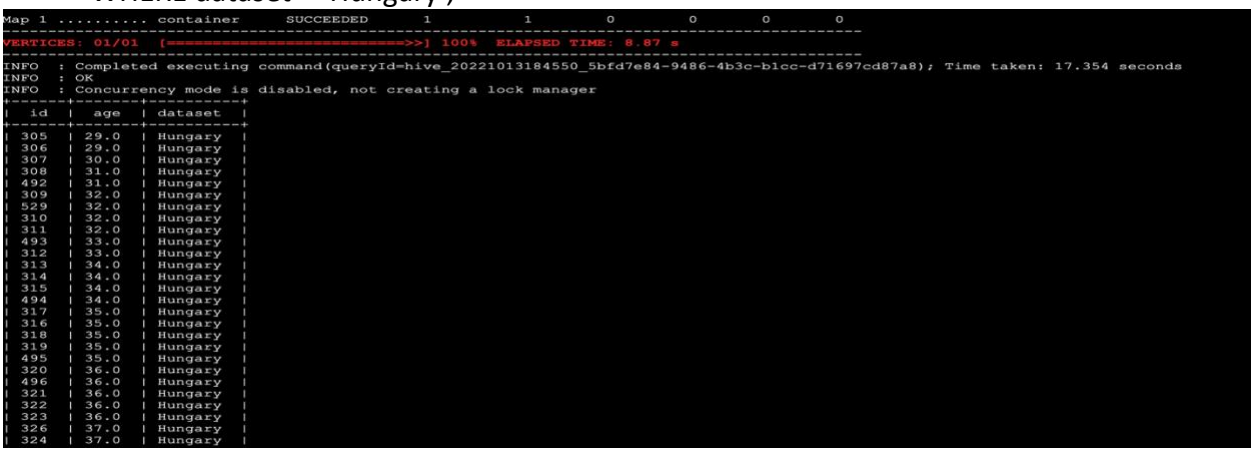
Naveenkumar

Data Scale Computing Lab3: -

1.

Write a Hive query to retrieve id, age, and dataset where the dataset value is “Hungary”.

```
gcloud dataproc jobs submit hive \
--cluster hive-cluster \
--region ${REGION} \
--execute "
SELECT id, age, dataset
FROM personal_detail
WHERE dataset = 'Hungary';"
```

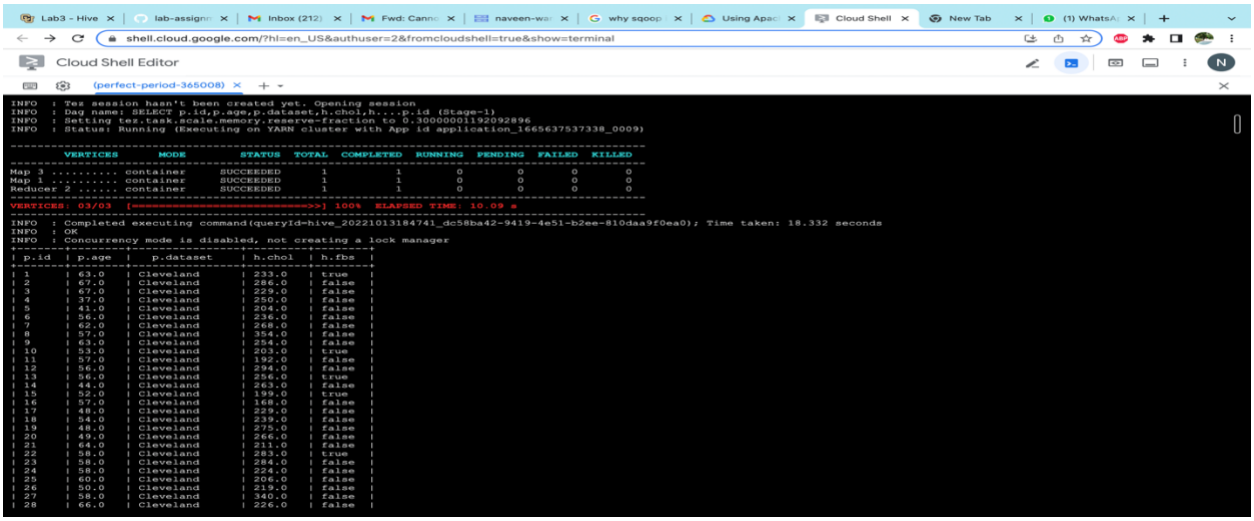


```
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 8.87 s
----------------------------------------------------------------------------------------------
INFO  : Completed executing command(queryId=hive_20221013184550_5bfd7e84-9486-4b3c-b1cc-d71697cd87a8); Time taken: 17.354 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
+------+-------+-----------+
|  id  |  age  |  dataset  |
+------+-------+-----------+
| 305  | 29.0  | Hungary   |
| 306  | 29.0  | Hungary   |
| 307  | 30.0  | Hungary   |
| 308  | 31.0  | Hungary   |
| 492  | 31.0  | Hungary   |
| 309  | 32.0  | Hungary   |
| 529  | 32.0  | Hungary   |
| 310  | 32.0  | Hungary   |
| 311  | 32.0  | Hungary   |
| 493  | 33.0  | Hungary   |
| 312  | 33.0  | Hungary   |
| 313  | 34.0  | Hungary   |
| 314  | 34.0  | Hungary   |
| 315  | 34.0  | Hungary   |
| 494  | 34.0  | Hungary   |
| 317  | 35.0  | Hungary   |
| 316  | 35.0  | Hungary   |
| 318  | 35.0  | Hungary   |
| 319  | 35.0  | Hungary   |
| 495  | 35.0  | Hungary   |
| 320  | 36.0  | Hungary   |
| 496  | 36.0  | Hungary   |
| 321  | 36.0  | Hungary   |
| 322  | 36.0  | Hungary   |
| 323  | 36.0  | Hungary   |
| 326  | 37.0  | Hungary   |
| 324  | 37.0  | Hungary   |
```

2.

Write a Hive query to retrieve id, age, dataset, chol and fbs and sort the values in ascending order of id.

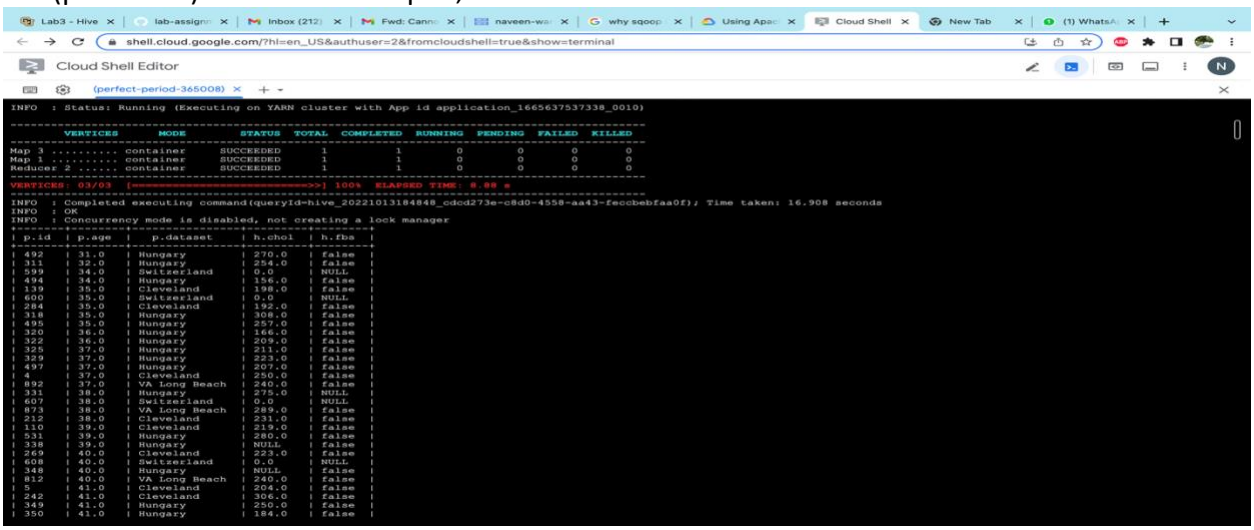
```
gcloud dataproc jobs submit hive --cluster hive-cluster --region ${REGION} --execute "
SELECT p.id, p.age, p.dataset, h.chol, h.fbs
FROM personal_detail p JOIN health_details h
ON (p.id = h.id) ORDER BY p.id;"
```



3.

Modify the query in Q2 by using "DISTRIBUTE BY" and explain the difference.

gcloud dataproc jobs submit hive --cluster hive-cluster --region ${REGION} --execute "
SELECT p.id, p.age, p.dataset, h.chol, h.fbs
FROM personal_detail p JOIN health_details h
ON (p.id = h.id) DISTRIBUTE BY p.id;"



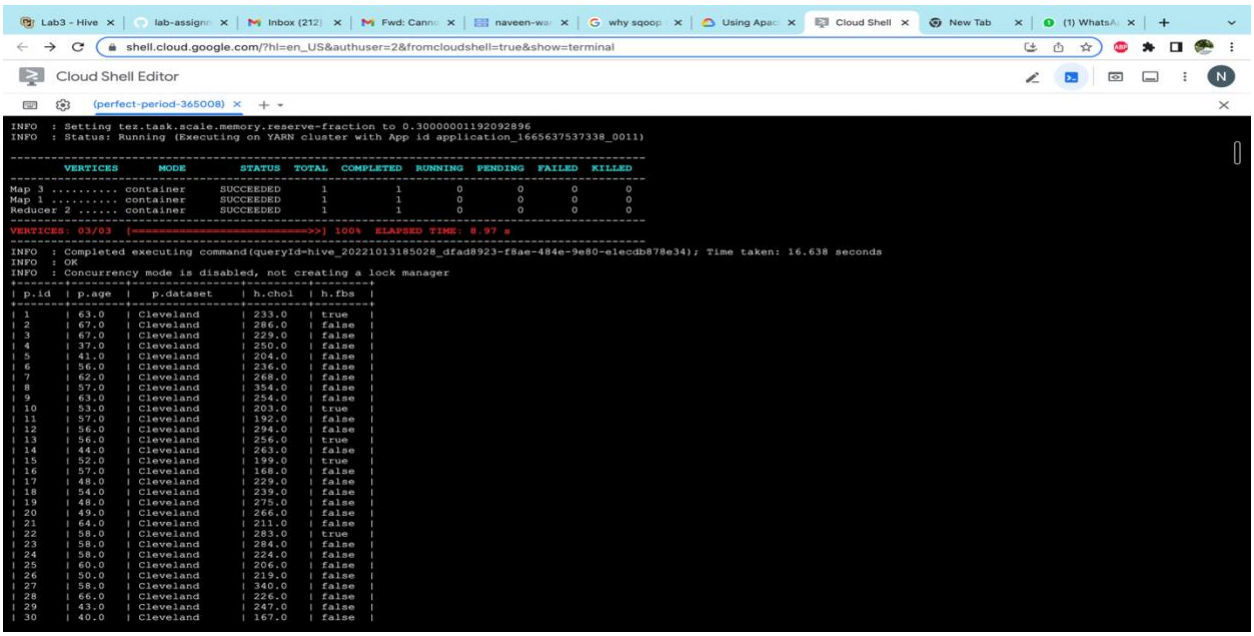
4.

Modify the query in Q2 by using "CLUSTER BY" and explain the difference between Q2, Q3 and Q4.

gcloud dataproc jobs submit hive --cluster hive-cluster --region ${REGION} --execute "
SELECT p.id, p.age, p.dataset, h.chol,h.fbs
FROM personal_detail p JOIN health_details h
ON (p.id = h.id) CLUSTER BY p.id;"

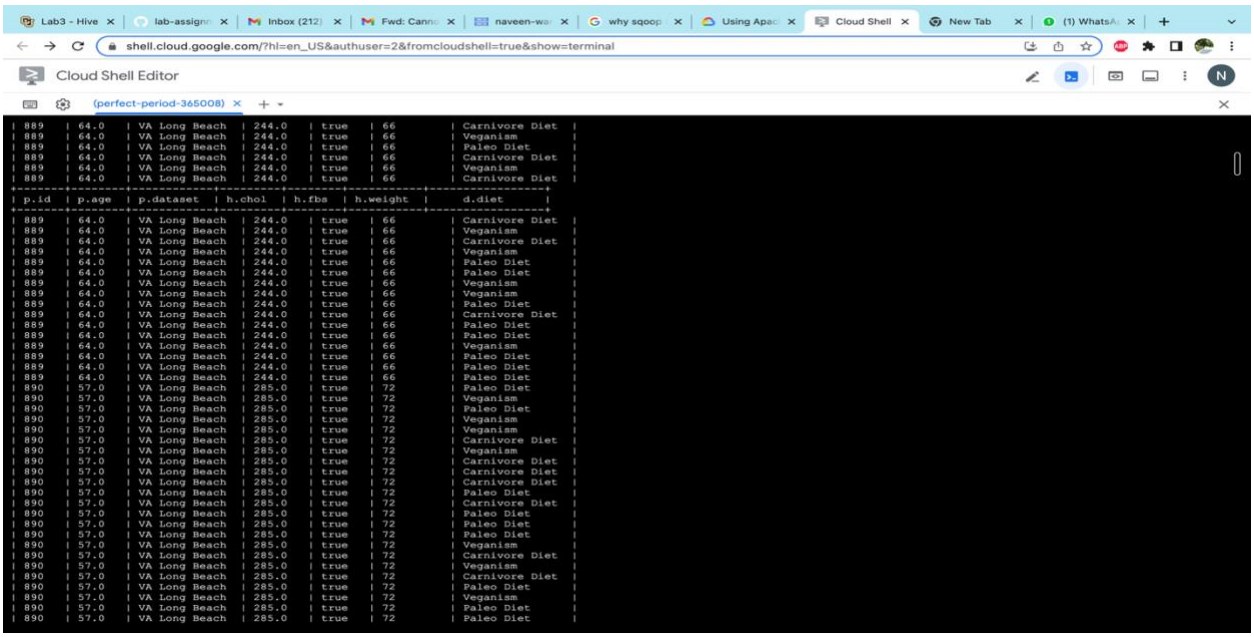
To arrange the output data in ascending or descending order, use the sort by and order by functions. The data is distributed to the reducers using the clustering and distribution methods, respectively.
The data is only sorted in the reducers when using sort by; they are not sorted globally. Therefore, each file in the output files from the reducers will be sorted, but the two files combined won't be sorted. Additionally, there can be overlaps in the data sets. The same names may appear in both reducers when we sort by name, for instance.
Data is sorted globally by order by, which sends all the data to a single reducer. The high file size could have a negative effect on performance.

When we distribute by name, all the same names are sent to the same reducer. Distribute by distributes the data to all reducers in a way that prevents data range overlap. For instance, reducer 2 will not have the name Regina if it receives it from reducer 1, reducer 2, etc. Distribute without sorting the data globally or within each reducer.
Cluster by does not sort the output data globally; instead, it sorts the data in each reducer and ensures that no data ranges overlap.



5.
Write a query to join tables personal_details, health_details and diet. Observe the results and point out the error/issue if any.

gcloud dataproc jobs submit hive --cluster hive-cluster --region ${REGION} --execute "
SELECT p.id, p.age, p.dataset, h.chol, h.fbs,h.weight,d.diet
FROM personal_detail p JOIN health_details h JOIN diet d
ON ((p.id = h.id) AND (h.weight = d.weight)) ORDER BY p.id;"



p.id	p.age	p.dataset	h.chol	h.fbs	h.weight	d.diet
889	64.0	VA Long Beach	244.0	true	66	Carnivore Diet
889	64.0	VA Long Beach	244.0	true	66	Veganism
889	64.0	VA Long Beach	244.0	true	66	Paleo Diet
889	64.0	VA Long Beach	244.0	true	66	Carnivore Diet
889	64.0	VA Long Beach	244.0	true	66	Veganism
889	64.0	VA Long Beach	244.0	true	66	Carnivore Diet
889	64.0	VA Long Beach	244.0	true	66	Veganism
889	64.0	VA Long Beach	244.0	true	66	Paleo Diet
889	64.0	VA Long Beach	244.0	true	66	Paleo Diet
889	64.0	VA Long Beach	244.0	true	66	Veganism
889	64.0	VA Long Beach	244.0	true	66	Veganism
889	64.0	VA Long Beach	244.0	true	66	Paleo Diet
889	64.0	VA Long Beach	244.0	true	66	Carnivore Diet
889	64.0	VA Long Beach	244.0	true	66	Paleo Diet
889	64.0	VA Long Beach	244.0	true	66	Paleo Diet
889	64.0	VA Long Beach	244.0	true	66	Veganism
889	64.0	VA Long Beach	244.0	true	66	Paleo Diet
889	64.0	VA Long Beach	244.0	true	66	Paleo Diet
889	64.0	VA Long Beach	244.0	true	66	Paleo Diet
890	57.0	VA Long Beach	285.0	true	72	Paleo Diet
890	57.0	VA Long Beach	285.0	true	72	Veganism
890	57.0	VA Long Beach	285.0	true	72	Paleo Diet
890	57.0	VA Long Beach	285.0	true	72	Veganism
890	57.0	VA Long Beach	285.0	true	72	Veganism
890	57.0	VA Long Beach	285.0	true	72	Carnivore Diet
890	57.0	VA Long Beach	285.0	true	72	Veganism
890	57.0	VA Long Beach	285.0	true	72	Carnivore Diet
890	57.0	VA Long Beach	285.0	true	72	Carnivore Diet
890	57.0	VA Long Beach	285.0	true	72	Carnivore Diet
890	57.0	VA Long Beach	285.0	true	72	Paleo Diet
890	57.0	VA Long Beach	285.0	true	72	Carnivore Diet
890	57.0	VA Long Beach	285.0	true	72	Paleo Diet
890	57.0	VA Long Beach	285.0	true	72	Paleo Diet
890	57.0	VA Long Beach	285.0	true	72	Veganism
890	57.0	VA Long Beach	285.0	true	72	Carnivore Diet
890	57.0	VA Long Beach	285.0	true	72	Veganism
890	57.0	VA Long Beach	285.0	true	72	Carnivore Diet
890	57.0	VA Long Beach	285.0	true	72	Paleo Diet
890	57.0	VA Long Beach	285.0	true	72	Veganism
890	57.0	VA Long Beach	285.0	true	72	Paleo Diet
890	57.0	VA Long Beach	285.0	true	72	Paleo Diet

There is a significant amount of data redundancy when we link all 3 tables. This is so because just one column between the diet table and the health details table—the weight column—is not a primary key in either table. Therefore, we obtain redundant data when we merge these two tables based on the weight field. We won't get any duplicate data from connecting the personal details and health details tables because they both contain a primary key column called id.

Theory Questions

1.

In your own words, describe the working of Hive. (Hint - how hive is on top of Hadoop and internally what techniques are used for querying).

Using SQL, Hive enables users to read, write, and manage petabytes of data. Apache Hadoop is an open-source platform used to effectively store and analyze big datasets, and Hive is built on top of it. As a result of its strong integration with Hadoop, Hive is built to process petabytes of data quickly.

Hive was developed to give non-programmers who are familiar with SQL the ability to interact with petabytes of data using a SQL-like interface called HiveQL. Large datasets are difficult for traditional relational databases to process because they were created for interactive queries on small to medium-sized datasets. Hive, on the other hand, uses batch processing to operate quickly across a sizable, distributed database. For use with Apache Hadoop's distributed task scheduling system, Yet Another Resource Negotiator, Hive converts HiveQL queries into MapReduce (YARN)

2. List out the advantages and disadvantages of HIVE.

Pros: -

- Fast. Hive is designed to quickly handle petabytes of data using batch processing.
- Familiar. Hive provides a familiar, SQL -like interface that is accessible to non-programmers.

- Scalable. Hive is easy to distribute, and scale based on your needs.

Cons: -

- Deploy
- Maintenance
- Support