Questions. (10 points each)

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

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

3. Modify the query in Q2 by using “DISTRIBUTE BY” and explain the difference.

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

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

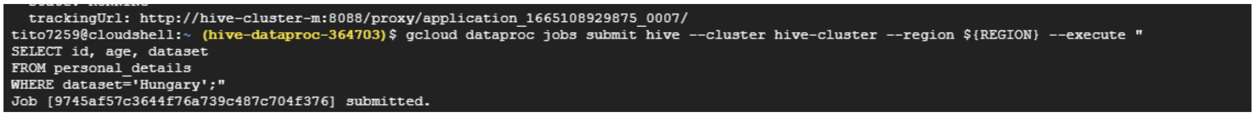
Q1

$ gcloud dataproc jobs submit hive --cluster hive-cluster --region ${REGION} --execute "

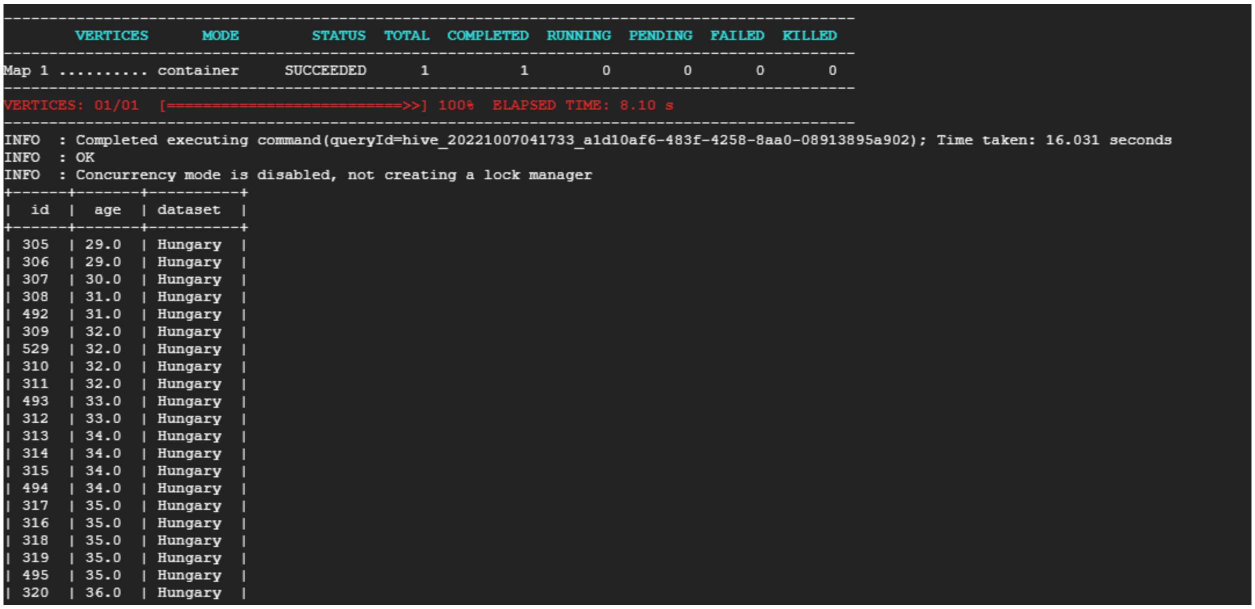
SELECT id, age, dataset

FROM personal_details

WHERE dataset='Hungary';"



```
  trackingUrl: http://hive-cluster-m:8088/proxy/application_1665108929875_0007/
tito7259@cloudshell:~ (hive-dataproc-364703)$ gcloud dataproc jobs submit hive --cluster hive-cluster --region ${REGION} --execute "
SELECT id, age, dataset
FROM personal_details
WHERE dataset='Hungary';"
Job [9745af57c3644f76a739c487c704f376] submitted.
```



```
----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 8.10 s
----------------------------------------------------------------------------------------------
INFO  : Completed executing command(queryId=hive_20221007041733_a1d10af6-483f-4258-8aa0-08913895a902); Time taken: 16.031 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  |
```

Q2

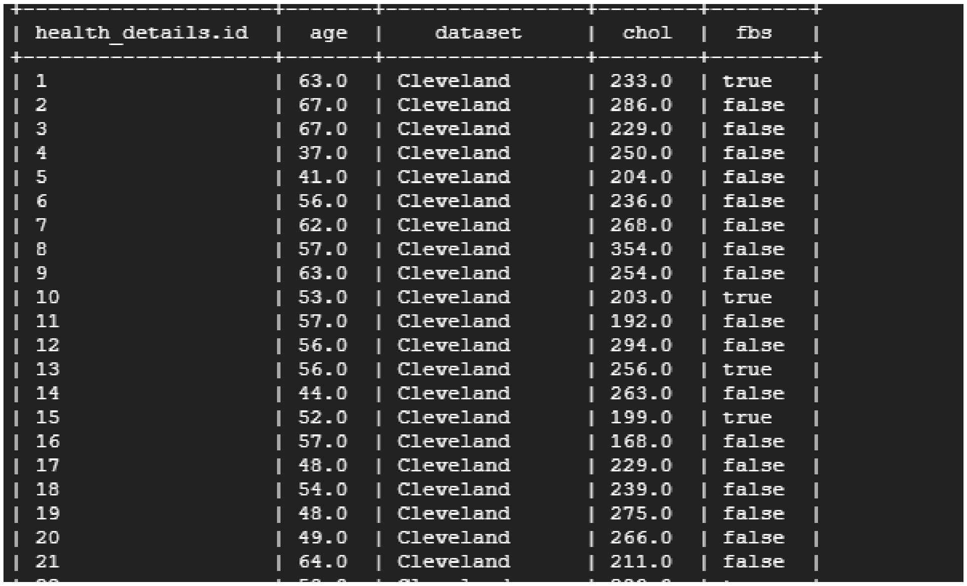
$ gcloud dataproc jobs submit hive --cluster hive-cluster --region ${REGION} --execute "

SELECT health_details.id,age,dataset,chol,fbs

FROM health_details

INNER JOIN personal_details ON health_details.id=personal_details.id

ORDER BY health_details.id ;"



```
+--------------------+-------+-----------------+--------+--------+
| health_details.id  |  age  |     dataset     |  chol  |  fbs   |
+--------------------+-------+-----------------+--------+--------+
| 1                  | 63.0  | Cleveland       | 233.0  | true   |
| 2                  | 67.0  | Cleveland       | 286.0  | false  |
| 3                  | 67.0  | Cleveland       | 229.0  | false  |
| 4                  | 37.0  | Cleveland       | 250.0  | false  |
| 5                  | 41.0  | Cleveland       | 204.0  | false  |
| 6                  | 56.0  | Cleveland       | 236.0  | false  |
| 7                  | 62.0  | Cleveland       | 268.0  | false  |
| 8                  | 57.0  | Cleveland       | 354.0  | false  |
| 9                  | 63.0  | Cleveland       | 254.0  | false  |
| 10                 | 53.0  | Cleveland       | 203.0  | true   |
| 11                 | 57.0  | Cleveland       | 192.0  | false  |
| 12                 | 56.0  | Cleveland       | 294.0  | false  |
| 13                 | 56.0  | Cleveland       | 256.0  | true   |
| 14                 | 44.0  | Cleveland       | 263.0  | false  |
| 15                 | 52.0  | Cleveland       | 199.0  | true   |
| 16                 | 57.0  | Cleveland       | 168.0  | false  |
| 17                 | 48.0  | Cleveland       | 229.0  | false  |
| 18                 | 54.0  | Cleveland       | 239.0  | false  |
| 19                 | 48.0  | Cleveland       | 275.0  | false  |
| 20                 | 49.0  | Cleveland       | 266.0  | false  |
| 21                 | 64.0  | Cleveland       | 211.0  | false  |
```

Q3

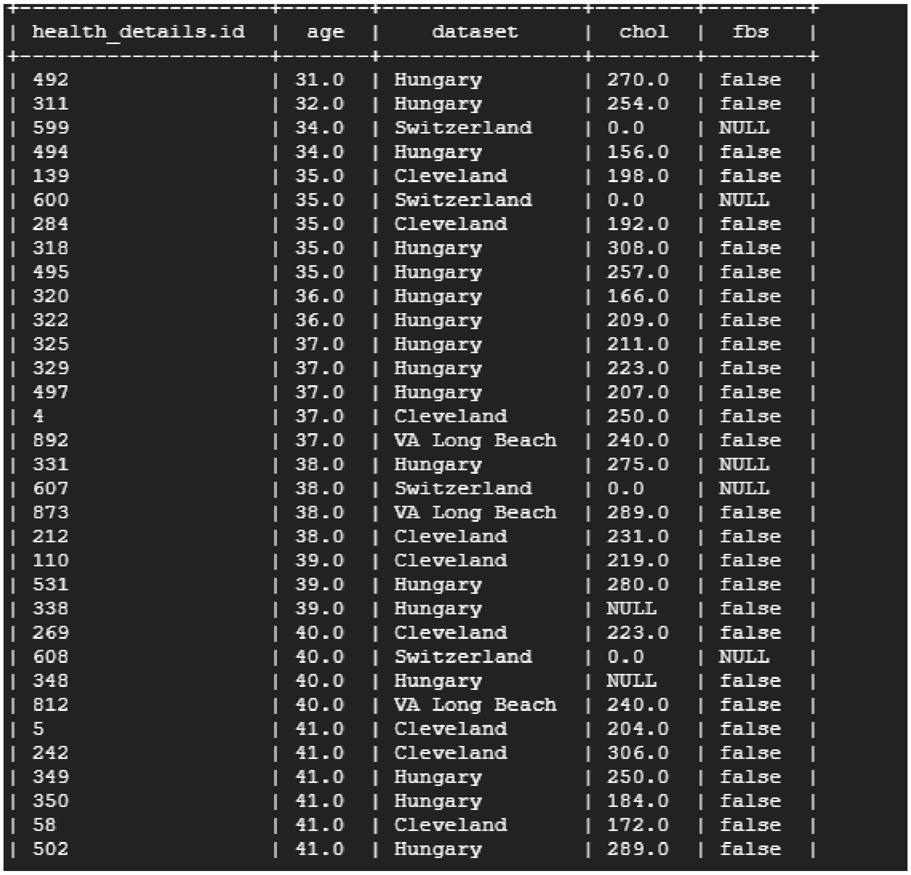
DISTRIBUTE BY clause is used to **distribute the input rows among reducers**. It ensures that all rows for the same key columns are going to the same reducer, whereas ORDER BY clause **orders the data globally**. Because it ensures the global ordering of the data, all the **data need to be passed from a single reducer only**.

SELECT health_details.id,age,dataset,chol,fbs

FROM health_details

INNER JOIN personal_details ON health_details.id=personal_details.id

DISTRIBUTE BY health_details.id ;"



```
+--------------------+-------+----------------+--------+--------+
| health_details.id  |  age  |    dataset     |  chol  |  fbs   |
+--------------------+-------+----------------+--------+--------+
| 492                | 31.0  | Hungary        | 270.0  | false  |
| 311                | 32.0  | Hungary        | 254.0  | false  |
| 599                | 34.0  | Switzerland    | 0.0    | NULL   |
| 494                | 34.0  | Hungary        | 156.0  | false  |
| 139                | 35.0  | Cleveland      | 198.0  | false  |
| 600                | 35.0  | Switzerland    | 0.0    | NULL   |
| 284                | 35.0  | Cleveland      | 192.0  | false  |
| 318                | 35.0  | Hungary        | 308.0  | false  |
| 495                | 35.0  | Hungary        | 257.0  | false  |
| 320                | 36.0  | Hungary        | 166.0  | false  |
| 322                | 36.0  | Hungary        | 209.0  | false  |
| 325                | 37.0  | Hungary        | 211.0  | false  |
| 329                | 37.0  | Hungary        | 223.0  | false  |
| 497                | 37.0  | Hungary        | 207.0  | false  |
| 4                  | 37.0  | Cleveland      | 250.0  | false  |
| 892                | 37.0  | VA Long Beach  | 240.0  | false  |
| 331                | 38.0  | Hungary        | 275.0  | NULL   |
| 607                | 38.0  | Switzerland    | 0.0    | NULL   |
| 873                | 38.0  | VA Long Beach  | 289.0  | false  |
| 212                | 38.0  | Cleveland      | 231.0  | false  |
| 110                | 39.0  | Cleveland      | 219.0  | false  |
| 531                | 39.0  | Hungary        | 280.0  | false  |
| 338                | 39.0  | Hungary        | NULL   | false  |
| 269                | 40.0  | Cleveland      | 223.0  | false  |
| 608                | 40.0  | Switzerland    | 0.0    | NULL   |
| 348                | 40.0  | Hungary        | NULL   | false  |
| 812                | 40.0  | VA Long Beach  | 240.0  | false  |
| 5                  | 41.0  | Cleveland      | 204.0  | false  |
| 242                | 41.0  | Cleveland      | 306.0  | false  |
| 349                | 41.0  | Hungary        | 250.0  | false  |
| 350                | 41.0  | Hungary        | 184.0  | false  |
| 58                 | 41.0  | Cleveland      | 172.0  | false  |
| 502                | 41.0  | Hungary        | 289.0  | false  |
```

Q4

CLUSTER BY clause **is a combination of DISTRIBUTE BY and SORT BY clauses together**. That means the output of the CLUSTER BY clause is equivalent to the output of **DISTRIBUTE BY + SORT BY** clauses.

The difference between Q2, Q3, and Q4.

Q2: Globally sort + Non-overlapping data range

Q3: Not sorted + Non-overlapping data range

Q4: Reducer sorted + Non-overlapping data range

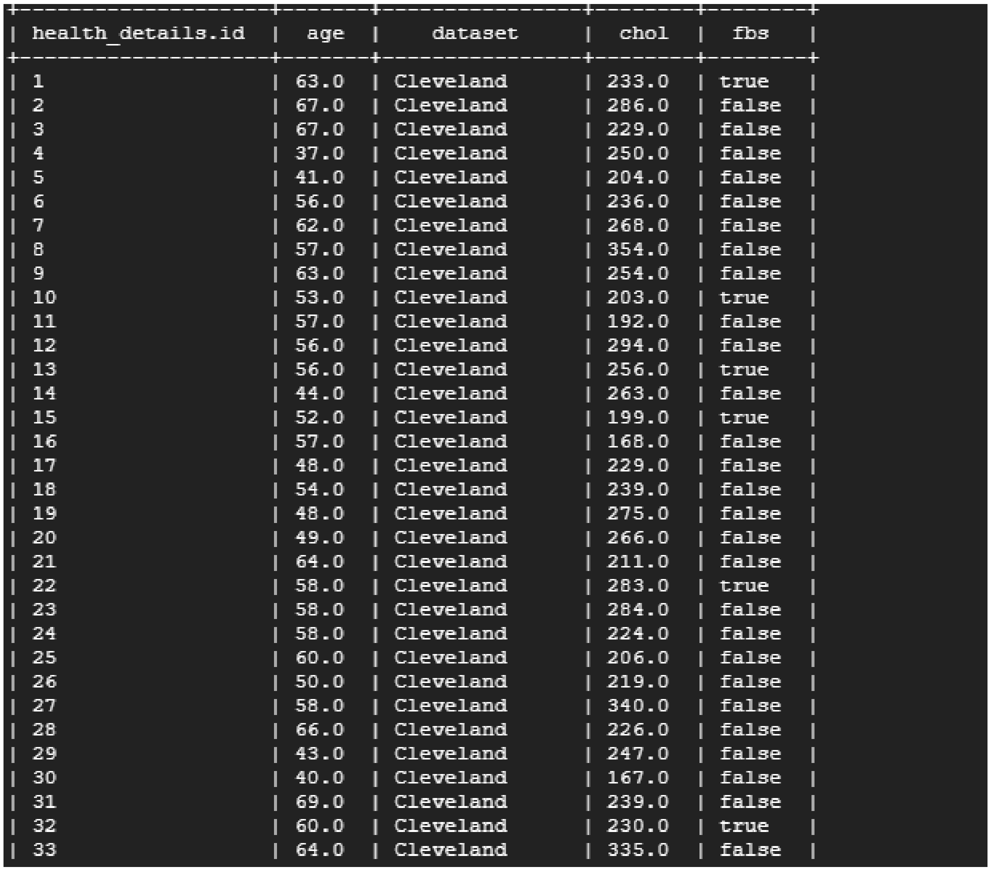
```
$ gcloud dataproc jobs submit hive --cluster hive-cluster --region ${REGION} --execute "

SELECT health_details.id,age,dataset,chol,fbs

FROM health_details

INNER JOIN personal_details ON health_details.id=personal_details.id

CLUSTER BY health_details.id ;"
```



```
+--------------------+-------+----------------+--------+--------+
| health_details.id  |  age  |    dataset     |  chol  |  fbs   |
+--------------------+-------+----------------+--------+--------+
| 1                  | 63.0  | Cleveland      | 233.0  | true   |
| 2                  | 67.0  | Cleveland      | 286.0  | false  |
| 3                  | 67.0  | Cleveland      | 229.0  | false  |
| 4                  | 37.0  | Cleveland      | 250.0  | false  |
| 5                  | 41.0  | Cleveland      | 204.0  | false  |
| 6                  | 56.0  | Cleveland      | 236.0  | false  |
| 7                  | 62.0  | Cleveland      | 268.0  | false  |
| 8                  | 57.0  | Cleveland      | 354.0  | false  |
| 9                  | 63.0  | Cleveland      | 254.0  | false  |
| 10                 | 53.0  | Cleveland      | 203.0  | true   |
| 11                 | 57.0  | Cleveland      | 192.0  | false  |
| 12                 | 56.0  | Cleveland      | 294.0  | false  |
| 13                 | 56.0  | Cleveland      | 256.0  | true   |
| 14                 | 44.0  | Cleveland      | 263.0  | false  |
| 15                 | 52.0  | Cleveland      | 199.0  | true   |
| 16                 | 57.0  | Cleveland      | 168.0  | false  |
| 17                 | 48.0  | Cleveland      | 229.0  | false  |
| 18                 | 54.0  | Cleveland      | 239.0  | false  |
| 19                 | 48.0  | Cleveland      | 275.0  | false  |
| 20                 | 49.0  | Cleveland      | 266.0  | false  |
| 21                 | 64.0  | Cleveland      | 211.0  | false  |
| 22                 | 58.0  | Cleveland      | 283.0  | true   |
| 23                 | 58.0  | Cleveland      | 284.0  | false  |
| 24                 | 58.0  | Cleveland      | 224.0  | false  |
| 25                 | 60.0  | Cleveland      | 206.0  | false  |
| 26                 | 50.0  | Cleveland      | 219.0  | false  |
| 27                 | 58.0  | Cleveland      | 340.0  | false  |
| 28                 | 66.0  | Cleveland      | 226.0  | false  |
| 29                 | 43.0  | Cleveland      | 247.0  | false  |
| 30                 | 40.0  | Cleveland      | 167.0  | false  |
| 31                 | 69.0  | Cleveland      | 239.0  | false  |
| 32                 | 60.0  | Cleveland      | 230.0  | true   |
| 33                 | 64.0  | Cleveland      | 335.0  | false  |
```

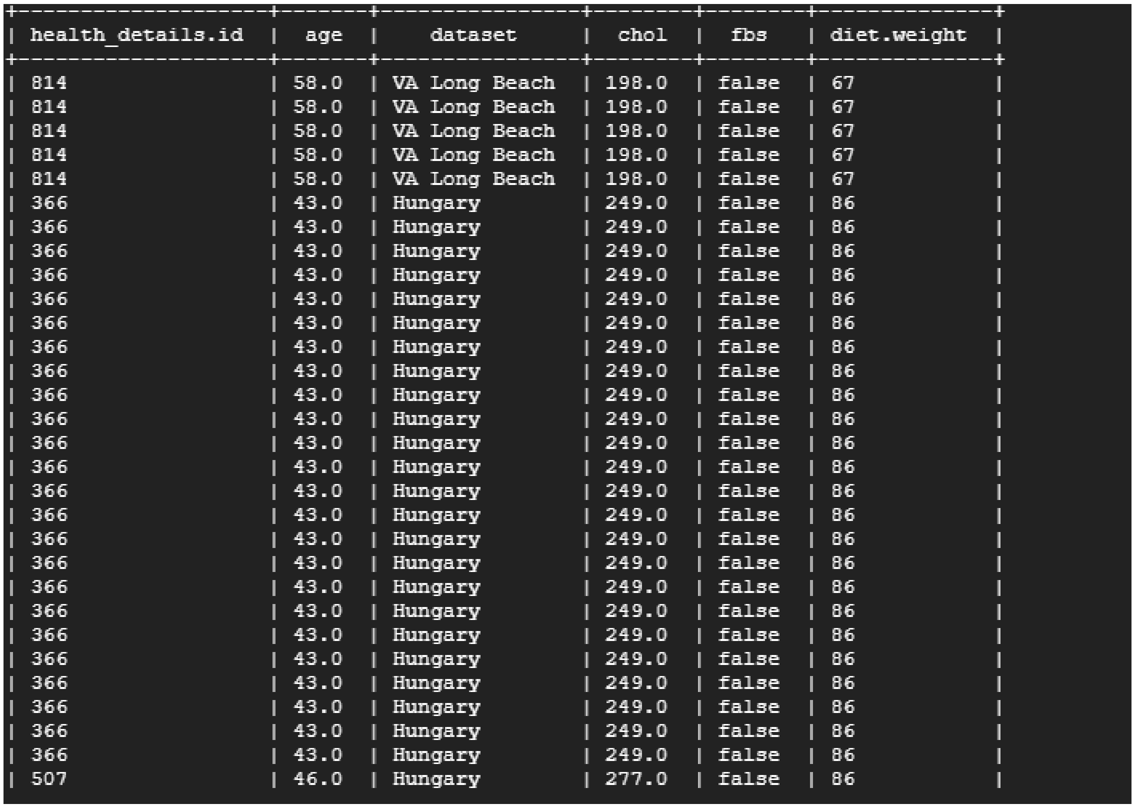
Q5

$ gcloud dataproc jobs submit hive --cluster hive-cluster --region ${REGION} --execute "

SELECT health_details.id,age,dataset,chol,fbs,diet.weight

FROM health_details INNER JOIN personal_details ON health_details.id=personal_details.id INNER JOIN diet ON health_details.weight=diet.weight LIMIT 10 ;"

The error is the data are duplicated in the table



```
+--------------------+-------+----------------+--------+--------+--------------+
| health_details.id  |  age  |    dataset     |  chol  |  fbs   | diet.weight  |
+--------------------+-------+----------------+--------+--------+--------------+
| 814                | 58.0  | VA Long Beach  | 198.0  | false  | 67           |
| 814                | 58.0  | VA Long Beach  | 198.0  | false  | 67           |
| 814                | 58.0  | VA Long Beach  | 198.0  | false  | 67           |
| 814                | 58.0  | VA Long Beach  | 198.0  | false  | 67           |
| 814                | 58.0  | VA Long Beach  | 198.0  | false  | 67           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 366                | 43.0  | Hungary        | 249.0  | false  | 86           |
| 507                | 46.0  | Hungary        | 277.0  | false  | 86           |
```

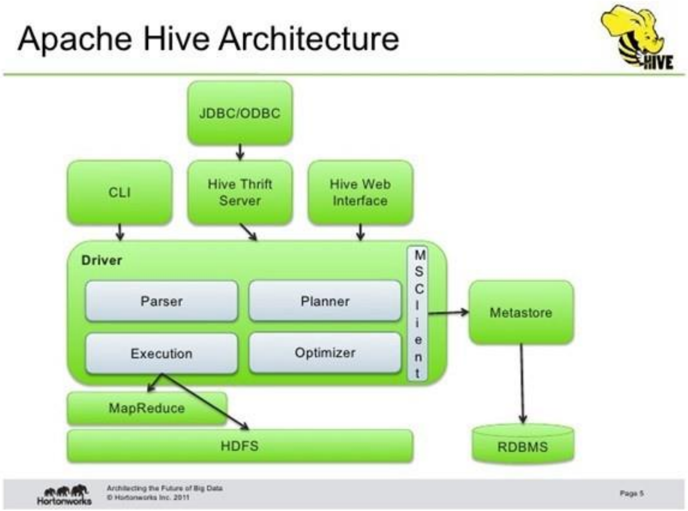
Theory Questions - (10 points each)

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)

2. List out the advantages and disadvantages of HIVE

Answer

1. Hive was created to allow non-programmers familiar with SQL to work with big data, using a SQL-like interface called HiveQL. Traditional relational databases are designed for interactive queries on small to medium datasets and do not process huge datasets well. Hive instead uses batch processing so that it works quickly across a very large distributed database. Hive transforms HiveQL queries into MapReduce or Tez jobs that run on Apache Hadoop’s distributed job scheduling framework. It queries data stored in a distributed storage solution, like the Hadoop Distributed File System (HDFS). Hive stores its database and table metadata in a metastore, which is a database or file backed store that provides easy data abstraction and discovery.



Apache Hive Architecture

Architecting the Future of Big Data
© Hortonworks Inc. 2011

Page 5

2.

Advantages of Hive

- Keeps queries running fast
- Takes very little time to write Hive query in comparison to MapReduce code
- HiveQL is a declarative language like SQL
- Provides the structure on an array of data formats
- Multiple users can query the data with the help of HiveQL
- Very easy to write query including joins in Hive
- Simple to learn and use

Disadvantages of Hive

- Useful when the data is structured
- You can do any analytical operation using MR programming

- Debugging code is very difficult
- You can’t do complicated operations