Hive

**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 us-central1 \

--execute "

SELECT id,age,dataset

FROM personal\_details

WHERE dataset = 'Hungary';"

Text

Description automatically generated

**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 us-central1 --execute "

SELECT pd.id,pd.age,pd.dataset,hd.chol,hd.fbs

FROM personal\_details pd JOIN health\_deatils hd

ON (pd.id = hd.id) ORDER BY pd.id;"

A computer screen capture

Description automatically generated with medium confidence

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

gcloud dataproc jobs submit hive --cluster hive-cluster --region us-central1 --execute "

SELECT pd.id,pd.age,pd.dataset,hd.chol,hd.fbs

FROM personal\_details pd JOIN health\_deatils hd

ON (pd.id = hd.id) DISTRIBUTE BY pd.id;"

ORDER BY clause orders the data globally. It ensures the global ordering of the data, all the data need to be passed from a single reducer only. As a result, the order by clause outputs one single file only.

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. It partitions the data on some key column, we can use the DISTRIBUTE BY clause in the hive queries. It is used to distribute the data to the reducers.

Text

Description automatically generated

**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 us-central1 --execute "

SELECT pd.id,pd.age,pd.dataset,hd.chol,hd.fbs

FROM personal\_details pd JOIN health\_deatils hd

ON (pd.id = hd.id) CLUSTER BY pd.id;"

Text

Description automatically generated

ORDER BY orders the data globally. It ensures the global ordering of the data, all the data need to be passed from a single reducer only. As a result, the order by clause outputs one single file only.

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. DISTRIBUTE BY does not sort the data either at the reducer level or globally. DISTRIBUTE BY clause may output N number of unsorted files.

CLUSTER BY clause is a combination of DISTRIBUTE BY and SORT BY clauses together. The CLUSTER BY clause distributes the data based on the key column and then sorts the output data by putting the same key column values adjacent to each other. CLUSTER by clause ensures that we are getting non-overlapping data ranges into the final outputs.

**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 us-central1 --execute "

SELECT pd.id,pd.age,pd.dataset,hd.chol,hd.fbs,hd.weight,d.diet

FROM personal\_details pd JOIN health\_deatils hd JOIN diet d

ON ((pd.id = hd.id) AND (hd.weight = d.weight)) ORDER BY pd.id;"

Table

Description automatically generated

There is data redundancy when the 3 tables are joined. Since there the health details table and diet table have only one column in common which is the weight column. So when the two tables are joined based on weight column the redundant data is formed. The ID column is the primary key in health details and personal details tables hence joining them does not give duplicate data.

**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)**

Hive enables users to read,write and manage the petabytes of data using SQL. Analytics can be performed at a very large scale due to its distributed, fault-tolerant data warehousing technology. It executes queries, gets metadata, sends metadata, sends plan, execute plan and fetches results. It uses bucketing,compression. Bucketing is similar to partitioning; it is a Hive query tuning tactic that allows to target a subset of data. Compression is one of the best Hive query optimization techniques. Big data compression helps in cutting down the amount of storage and bandwidth required to handle large data sets. Eliminating small file operations from the query is an effective Hive performance tuning tactic. Doing so enables a healthy Hive ecosystem.

**2. List out the advantages and disadvantages of HIVE**

**Advantages of HIVE**

Data to be analyzed is stored in HDFS which provides all features like scalability,redundancy etc and SQL like query over data in Hadoop.

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.

**Disadvantages of HIVE**

Hive doesn’t support OLTP. Hive supports Online Analytical Processing (OLAP), but not Online Transaction Processing (OLTP).

It is useful when the data is structured.

It doesn’t support subqueries.

It has a high latency.

Hive tables don’t support delete or update operations.