211720104050

BIG DATA HOME WORK 5

Question No:1

1) What is a MetaStore in Hive?

A:

- In a relational database, it maintains the partitions' metadata as well as information about Hive tables' schema and location.

- This data is accessible to clients via the metastore service API.

- Hive metastore is composed of two basic components: a service that grants other Apache **Hive** services access to **metastores**

Question No:2

2) Where does the data of a Hive table gets stored?

A:

- Hive is a set of components akin to a data warehouse that is developed on top of the Hadoop Distributed File System.

- Because of this, the actual data for a hive table is kept on HDFS, but the location(path) depends on the type of table defined with the CREATE TABLE command.

- Location for managed tables will be determined by the hive.warehouse.dir property in the hive-site.xml configuration file.

- The /usr/local/hive/warehouse/ default value is used.

- For external tables, the LOCATION 'path>' command in the CREATE TABLE command must be used to specifically define the location.

- Data would only be on HDFS in all instances, but with different path locations.

Question No:3

3) Why Hive does not store metadata information in HDFS?

A:

- Instead of using HDFS, Hive uses RDBMS to store metadata data in the metastore.
- Given how time-consuming HDFS read/write operations are, choosing an RDBMS will help you achieve low latency.

Question No:4

4) What is the difference between local and remote metastore?

A:

<u>Local Metastore:</u>

In a local metastore configuration, a database running in a different JVM, either on the same machine or a distant machine, connects to the metastore service, which is executing in the same JVM as the Hive service.

<u>Remote Metastore:</u>

- The metastore service in the remote metastore configuration runs in a different JVM from the Hive service JVM.
- Thrift Network APIs are used by other processes to interact with the metastore server.
- In this scenario, you can have one or more metastore servers to increase availability.

Question No:5

5) **What is the default database provided by Apache Hive for metastore?**

A:

For the metastore, Hive by default offers an embedded Derby database instance backed by the local disk.

The embedded metastore configuration is what it is called.

Question No:6

6) **What is the difference between external table and managed table?**

A:

- The main distinction between a managed table and an external table is as follows:
- When dropping a managed table, the metadata data is also removed from the Hive warehouse directory along with the table data.
- On the other hand, when dealing with an external table, Hive simply removes the table's metadata while leaving the table's contents in HDFS intact.

Question No:7

7) **Is it possible to change the default location of a managed table?**

A:

Yes, it is possible to alter a managed table's default location. The clause LOCATION '<hdfs_path>' can be used to do this.

Question No:8

8) What is a partition in Hive?

A:

- Based on a column or partition key, Hive divides tables into partitions to group together data of a similar type.
- To identify a certain partition, each Table may have one or more partition keys.
- A partition is merely a subdirectory of the table directory physically.

Question No:9

9) Why do we perform partitioning in Hive?

A:

- Partitioning gives a Hive table more granularity, which decreases query latency by only scanning the relevant partitioned data rather to the entire data set.
- For instance, we can divide up a website's transaction log based on the month, such as January, February, etc.
- Therefore, any analytics pertaining to a specific month, let's say January, will only need to scan the Jan partition (sub-directory), not the entire table data.

Question No:10

10) What is dynamic partitioning and when is it used?

A:

- When using dynamic partitioning, the values for the partition columns are known at runtime, or when the data is being loaded into a Hive table.

- In the two situations listed below, dynamic partition may be used:

- Data is loaded into an existing, non-partitioned table to enhance sampling and, as a result, reduce query latency.

- Finding these partition values manually from large data sets is a laborious operation when one does not know all of the partition values in advance.

Question No:11

11)Suppose, you create a table that contains details of all the transactions done by the customers of year 2022: CREATE TABLE transaction_details (cust_id INT, amount FLOAT, month STRING, country STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ; Now, after inserting 50,000 tuples in this table, you want to know the total revenue generated for each month. But, Hive is taking too much time to process this query. How will you solve this problem and list the steps that you will be taking in order to do so?

A:

- By dividing the table into sections for each month, we can address the issue of query latency. Therefore, rather than scanning entire data sets for each month, we will just do it for the partitioned data.

- We are aware that we cannot directly partition an already-existing non-partitioned table. To remedy this specific issue, we will do as follows:

  1) Make a table that is partitioned with the name "partitioned transaction":

```
CREATE TABLE partitioned_transaction (cust_id INT,
amount FLOAT, country STRING) PARTITIONED BY (month
STRING) ROW FORMAT DELIMITED FIELDS TERMINATED
BY ‘,’ ;
```

2) Enable dynamic partitioning in Hive:

```
SET hive.exec.dynamic.partition = true;

SET hive.exec.dynamic.partition.mode = nonstrict;
```

3) Transfer the data from the non – partitioned table into the newly created partitioned table:

```
INSERT OVERWRITE TABLE partitioned_transaction
PARTITION (month) SELECT cust_id, amount, country,
month FROM transaction_details;
```

- Now, we can perform the query using each partition and therefore, decrease the query time.