**Hive interview Questions**

**Scenario:**

Suppose, I create a table that contains details of all the transactions done by the customers of year 2016: 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, I want to know the total revenue generated for each month. But, Hive is taking too much time in processing this query. How will you solve this problem and list the steps that I will be taking in order to do so?

We can solve this problem of query latency by partitioning the table according to each month. So, for each month we will be scanning only the partitioned data instead of whole data sets.

As we know, we can’t partition an existing non-partitioned table directly. So, we will be taking following steps to solve the very problem:

1. Create a partitioned table, say 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.

Scenario:

I am inserting data into a table based on partitions dynamically. But, I received an error – FAILED ERROR IN SEMANTIC ANALYSIS: Dynamic partition strict mode requires at least one static partition column. How will you remove this error?

To remove this error one has to execute following commands:

SET hive.exec.dynamic.partition = true;

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

What is the default maximum dynamic partition that can be created by a mapper/reducer? How can you change it?

By default the number of maximum partition that can be created by a mapper or reducer is set to 100. One can change it by issuing the following command:

SET hive.exec.max.dynamic.partitions.pernode = <value>

Note: You can set the total number of dynamic partitions that can be created by one statement by using: SET hive.exec.max.dynamic.partitions = <value>

What is a partition in Hive?

Hive organizes tables into partitions for grouping similar type of data together based on a column or partition key.

Each Table can have one or more partition keys to identify a particular partition.

Physically, a partition is nothing but a sub-directory in the table directory.

Why do we perform partitioning in Hive?

Partitioning provides granularity in a Hive table and therefore, reduces the query latency by scanning only relevant partitioned data instead of the whole data set.

For example, we can partition a transaction log of an e – commerce website based on month like Jan, February, etc.

So, any analytics regarding a particular month, say Jan, will have to scan the Jan partition (sub – directory) only instead of the whole table data.

What is dynamic partitioning and when is it used?

In dynamic partitioning values for partition columns are known in the runtime, i.e. It is known during loading of the data into a Hive table.

One may use dynamic partition in following two cases:

Loading data from an existing non-partitioned table to improve the sampling and therefore, decrease the query latency.

When one does not know all the values of the partitions before hand and therefore, finding these partition values manually from a huge data sets is a tedious task.

Where does the data of a Hive table gets stored?

By default, the Hive table is stored in an HDFS directory – /user/hive/warehouse.

One can change it by specifying the desired directory in hive.metastore.warehouse.dir configuration parameter present in the hive-site.xml.

What kind of applications is supported by Apache Hive?

Hive supports all those client applications that are written in:

Java ,PHP , Python, C++Ruby

What is a metastore in Hive?

Metastore in Hive stores the meta data information using RDBMS and an open source ORM (Object Relational Model) layer called Data Nucleus which converts the object representation into relational schema and vice versa.

Why Hive does not store metadata information in HDFS?

Hive stores metadata information in the metastore using RDBMS instead of HDFS.

The reason for choosing RDBMS is to achieve low latency as HDFS read/write operations are time consuming processes.

What is the difference between local and remote metastore?

Local Metastore:

In local metastore configuration, the metastore service runs in the same JVM in which the Hive service is running and connects to a database running in a separate JVM, either on the same machine or on a remote machine.

Remote Metastore:

In the remote metastore configuration, the metastore service runs on its own separate JVM and not in the Hive service JVM. Other processes communicate with the metastore server using Thrift Network APIs. You can have one or more metastore servers in this case to provide more availability.

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

By default, Hive provides an embedded Derby database instance backed by the local disk for the metastore. This is called the embedded metastore configuration.

What is the difference between external table and managed table?

Here is the key difference between an external table and managed table:

In case of managed table, If one drops a managed table, the metadata information along with the table data is deleted from the Hive warehouse directory.

On the contrary, in case of an external table, Hive just deletes the metadata information regarding the table and leaves the table data present in HDFS untouched.

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

Yes, it is possible to change the default location of a managed table. It can be achieved by using the clause – LOCATION ‘<hdfs\_path>’.

When should we use SORT BY instead of ORDER BY?

We should use SORT BY instead of ORDER BY when we have to sort huge datasets because SORT BY clause sorts the data using multiple reducers whereas ORDER BY sorts all of the data together using a single reducer. Therefore, using ORDER BY against a large number of inputs will take a lot of time to execute.

=========================================================

What is Apache Hive?

Ans. Basically, a tool which we call a data warehousing tool is Hive.

However, Hive gives SQL queries to perform an analysis and also an abstraction.

Although, Hive it is not a database it gives you logical abstraction over the databases and the tables.

Is Hive suitable to be used for OLTP systems? Why?

Ans. No, it is not suitable for OLTP system since it does not offer insert and update at the row level.

Where does the data of a Hive table gets stored?

Ans. In an HDFS directory – /user/hive/warehouse, the Hive table is stored, by default only.

Moreover, by specifying the desired directory in hive.metastore.warehouse.dir configuration parameter present in the hive-site.xml, one can change it.

What is a metastore in Hive?

Ans. Basically, to store the metadata information in the Hive we use Metastore.

Though, it is possible by using RDBMS and an open source ORM (Object Relational Model) layer called Data Nucleus. That converts the object representation into the relational schema and vice versa.

**Why does Hive not store metadata information in HDFS?**

**Ans.** Using RDBMS instead of HDFS, Hive stores metadata information in the metastore. Basically, to achieve low latency we use RDBMS. Because **HDFS read/write operations** are time-consuming processes.

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

**Ans.** It offers an embedded Derby database instance backed by the local disk for the metastore, by default. It is what we call embedded metastore configuration.

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

**Ans. Managed table**

The metadata information along with the table data is deleted from the Hive warehouse directory if one drops a managed table.\

**External table**  
Hive just deletes the metadata information regarding the table. Further, it leaves the table data present in HDFS untouched.

**What is a partition in Hive?**

**Ans.** Basically, for the purpose of grouping similar type of data together on the basis of column or partition key, Hive organizes tables into partitions.

Moreover, to identify a particular partition each table can have one or more partition keys. On defining **Hive Partition**, in other words, it is a sub-directory in the table directory.

**Why do we perform partitioning in Hive?**

**Ans.** In a Hive table, Partitioning provides granularity. Hence, by scanning only relevant partitioned data instead of the whole dataset it reduces the query latency.

**What is dynamic partitioning and when is it used?**

**Ans.** Dynamic partitioning values for partition columns are known in the runtime. In other words, it is known during loading of the data into a Hive table.

* **Usage:**

1. While we Load data from an existing non-partitioned table, in order to improve the sampling. Thus it decreases the query latency.
2. Also, while we do not know all the values of the partitions beforehand. Thus, finding these partition values manually from a huge dataset is a tedious task.

**Why do we need buckets?**

**Ans.** Basically, for **performing bucketing** to a partition there are two main reasons:

* A map side join requires the data belonging to a unique join key to be present in the same partition.
* It allows us to decrease the query time. Also, makes the sampling process more efficient.

**How Hive distributes the rows into buckets?**

**Ans.** By using the formula: hash\_function (bucketing\_column) modulo (num\_of\_buckets) Hive determines the bucket number for a row. Basically, hash\_function depends on the column data type. Although, hash\_function for integer data type will be:  
hash\_function (int\_type\_column)= value of int\_type\_column

**What is the use of Hcatalog?**

**Ans.** Basically, to share data structures with external systems we use Hcatalog. It offers access to hive metastore to users of other tools on Hadoop. Hence, they can read and write data to hive’s data warehouse.

**What are the different components of a Hive architecture?**

**Ans.** There are several components of **Hive Architecture**. Such as –

1. User Interface – Basically, it calls the execute interface to the driver. Further, driver creates a session handle to the query. Then sends the query to the compiler to generate an execution plan for it.
2. Metastore – It is used to Send the metadata to the compiler. Basically, for the execution of the query on receiving the send MetaData request.
3. Compiler- It generates the execution plan. Especially, that is a DAG of stages where each stage is either a metadata operation, a map or reduce job or an operation on HDFS.
4. Execute Engine- Basically,  by managing the dependencies for submitting each of these stages to the relevant components we use Execute engine.

**Features and Limitations of Hive.**

**Ans. Features of Hive**

1. The best feature is it offers data summarization, query, and analysis in much easier manner.
2. To process data without actually storing in HDFS, Hive supports external tables.
3. Moreover, it fits the low-level interface requirement of Hadoop perfectly.

* **Limitation of Hive**

1. We can not perform real-time queries with Hive. Also, it does not offer row-level updates.
2. Moreover,  for interactive data browsing Hive offers acceptable latency.
3. Also, we can say Hive is not the right choice for online transaction processing

================Then End========================