Interview Questions - Hive and Querying

1. **Explain ORDER BY, CLUSTER BY, SORT BY and DISTRIBUTE BY in brief.**

Answer. Both **ORDER BY** and **SORT BY** are used for sorting query results in ascending or descending order. However, one of the differences between them is the way they sort results. **ORDER BY** sorts the entire data using a reducer, whereas **SORT BY** does not guarantee overall sorting of data. There may be overlapping data and it might need more than one reducer.

Both **DISTRIBUTE BY** and **CLUSTER BY** are used for categorising query results on the basis of one or more columns. **CLUSTER BY** is a shortcut for both **DISTRIBUTE BY**and **SORT BY**. Hive uses the columns in DISTRIBUTE BY to distribute the rows among the reducers. All rows with the same **DISTRIBUTE BY** columns will go to the same reducer. However, **DISTRIBUTE BY** does not guarantee clustering or sorting of properties.

2. **Explain the difference between External and Internal tables.**

Answer. **External Table:** Unlike RDBMSes where data and tables are tightly coupled, data in External Tables is loosely coupled. External Tables reside in HDFS. Even if you drop an external table in Hive, the data mapped to it remains intact inside HDFS.

Internal Table: An Internal Table in Hive is similar to the tables in RDBMSes. The data and the table schema are tightly coupled in internal tables. If you drop an internal table in Hive, the data stored in it will get deleted.

3. **In which scenarios do you use external tables?**

Answer. We use external tables in the following scenarios:

1. When we need to store data in a custom location
2. Unlike Internal tables, if we delete external tables, they still continue to reside in HDFS
3. Data from external tables should not be owned by Hive

4. **Is Hive a database or a data warehouse? What are the key differences between Hive and RDBMSes?**

Answer. Hive is a data warehouse. The key difference between Hive and RDBMSes is that an RDBMS is a traditional database where you can store only a limited amount of data, whereas Hive is a data warehouse where you can store data in bulk and also perform data analysis.

5. **What type of Read and Write operations take place in Hive?**

Answer. **READ** Many, **WRITE** Once

6. **What are the instances where you can use Indexing?**

Answer. The key instances where you can use indexing are as follows:

1. When the data set is large
2. When faster query execution is needed
3. For columns that are used more frequently than others
4. For read-heavy applications, where you need to read the data more frequently

7. **Differentiate between Hive and HBase.**

Answer.

Hive	**HBase**
Hive is a 'data warehouse software' that enables you to query and manipulate data using an SQL-like language known as HiveQL.	HBase is a distributed data store built on top of HDFS, and it can leverage all the benefits provided by Hadoop or HDFS.
Hive abstracts the programming complexity of MapReduce and provides a simple SQL-like language known as HiveQL for querying data sets.	HBase does not have a native data-processing engine and relies on Map-Reduce and Spark APIs for data processing.
Hive has a relational DBMS data model.	HBase has a columnar data model.
Apache Hive has high latency as compared with HBase. Hence, it is not preferred for looking up individual records.	HBase provides a random and fast lookup on top of HDFS, which allows a user to query for individual records.

8. **Can Hive be used as an OLTP system like MySQL?**

Answer. Hive does not support insert and update functions at a row-level, which makes it unsuitable for OLTP systems. **Note:** OLTP is an online transaction-processing system that involves **INSERT**, **UPDATE** and **DELETE** operations.

9. **What are the limitations of Hive?**

Answer. Some limitations of Hive are as follows:

- Hive does not support insert and update functions at a row-level, which makes it unsuitable for OLTP systems.
- Hive does not support real-time processing.
- Hive queries have high latency due to the start-up overhead of the MapReduce job.

10. How does Hive improve performance with tables in ORC format?

Answer. Using the ORC format leads to a reduction in the size of the data stored, as this file format has high compression ratios. As the data size is reduced, the time to read and write the data is also reduced. The ORC format improves query performance also by the way it stores data in a file. Data is stored in a columnar format and columns that are not needed in a query can be skipped, thus leading to better performance.