**Hive Interview Questions and Answers**

**Q 1. What is the definition of Hive? What is the present version of Hive?**

**Ans:-** Hive is a data warehouse software project built on top of Apache Hadoop for providing data query and analysis. Hive gives an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop.

Present version of hive is 4.0.0-alpha-2

**Q 2. Is Hive suitable to be used for OLTP systems? Why?**

**Ans:- No Hive does not provide insert and update at row level. So it is not suitable for OLTP system.**

**3. How is HIVE different from RDBMS? Does hive support ACID transactions. If not then give the proper reason**

**Ans:-**

| **RDBMS** | **Hive** |
| --- | --- |
| It is used to maintain database. | It is used to maintain data warehouse. |
| It uses SQL (Structured Query Language). | It uses HQL (Hive Query Language). |
| Schema is fixed in RDBMS. | Schema varies in it. |
| Normalized data is stored. | Normalized and de-normalized both type of data is stored. |
| Tables in rdbms are sparse. | Table in hive are dense. |
| It doesn’t support partitioning. | It supports automation partition. |
| No partition method is used. | Sharding method is used for partition. |

Hive supports ACID transactions on tables that store ORC file format. Transaction tables

cannot be accessed from the non-ACID Transaction Manager (DummyTxnManager)

session. External tables cannot be created to support ACID since the changes on external

tables are beyond Hive control.

**4. Explain the hive architecture and the different components of a Hive architecture?**

**Ans:-** Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table, their data types, and HDFS mapping. HiveQL is similar to SQL for querying on schema info on the Metastore

**Q5. Mention what Hive query processor does? And Mention what are the components of a Hive**

**query processor?**

**Ans -**

**Following are the components of a Hive Query Processor:**

* Parse and Semantic Analysis (ql/parse)
* Metadata Layer (ql/metadata)
* Type Interfaces (ql/typeinfo)
* Sessions (ql/session)
* Map/Reduce Execution Engine (ql/exec)
* Plan Components (ql/plan)
* Hive Function Framework (ql/udf)
* Tools (ql/tools)

**Q6. What are the three different modes in which we can operate Hive?**

**Ans -**

There are three modes for Hive Metastore deployment:

* Embedded Metastore.
* Local Metastore.
* Remote Metastore.

**Q7. Features and Limitations of Hive. Hive Features**

**Ans -**

Some Hive new features are discussed below:

**i. Framework**

Apache Hive is built on top of Hadoop distributed framework system (HDFS).

**ii. Large datasets**

However, in distributed storage, it helps to query large datasets residing.

**iii. Warehouse**

Also, we can say Hive is a distributed data warehouse.

**iv. Language**

Queries data using a SQL-like language called HiveQL (HQL).

**v. Declarative language**

HiveQL is a declarative language like SQL.

**vi. Table structure**

Table structure/s is/are similar to tables in a relational database.

**vii. Multi-user**

Multiple users can simultaneously query the data using Hive-QL.

**viii. Data Analysis**

However, to perform more detailed data analysis, Hive allows writing custom MapReduce framework processes.

**ix. ETL support**

Also, it is possible to extract/transform/load (ETL) Data easily.

**x. Data Formats**

Moreover, Hive offers the structure on a variety of data formats.

**xi. Storage**

Hive allows access files stored in HDFS. Also, similar others data storage systems such as Apache HBase.

**x. Format conversion**

Moreover, it allows converting the variety of format from to within Hive. Although, it is very simple and possible.

**Limitations of Hive**

**i. OLTP Processing issues**

However, Hive is not designed for Online transaction processing (OLTP). Although, we can use it for the Online Analytical Processing (OLAP).

**ii. No Updates**

It does not support updates and deletes, however, it does support overwriting or apprehending data.

**iii. Subqueries**

Basically, in Hive, Subqueries are not supported.

**Q 8. How to create a Database in HIVE?**

**Ans:-**

CREATE DATABASE|SCHEMA [IF NOT EXISTS] &lt;database name&gt;

**Q9. How to create a table in HIVE?**

**Ans:-**

Create Table is a statement used to create a table in Hive. The syntax and example are as follows:

CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db\_name.] table\_name

[(col\_name data\_type [COMMENT col\_comment], ...)]

[COMMENT table\_comment]

[ROW FORMAT row\_format]

[STORED AS file\_format]

**Q10. What do you mean by describe and describe extended and describe formatted with respect to database and table**

**Q11.How to skip header rows from a table in Hive?**

**Ans:-** create external table employee (id int, name string) lines terminated by &#39;\n&#39; location &#39;/user/hirw/employees’ tblproperties (&quot;skip.header.line.count&quot;=&quot;1&quot;);

**Q12.What is a hive operator? What are the different types of hive operators?**

**Ans:-** Hive operators are used for mathematical operations on operands. It returns specific value as per the logic applied.

**There are four types of operators in Hive:**

* Relational Operators
* Arithmetic Operators
* Logical Operators
* Complex Operators

**Q13.Explain about the Hive Built-In Functions**

**Ans:-**

The Hive Built-in functions are categorized as Mathematical function, Collection function, Type conversion function, Date function, Conditional function, and String function.

**14. Write hive DDL and DML commands.**

**Ans:-**

**The several types of Hive DDL commands are:**

* CREATE.
* SHOW.
* DESCRIBE.
* USE.
* DROP.
* ALTER.
* TRUNCATE.
* The various Hive DML commands are:
* LOAD.
* SELECT.
* INSERT.
* DELETE.
* UPDATE.
* EXPORT.
* IMPORT.

**15.Explain about SORT BY, ORDER BY, DISTRIBUTE BY and CLUSTER BY in Hive**

**Ans –**

**SORT BY:** Sort is organizing data in a particular order allowing for information to be found easier. For example, names and contact information may be sorted in alphabetical order to allow the person looking for a name to see if it's available.

**ORDER BY:** The ORDER BY syntax in HiveQL is similar to the syntax of ORDER BY in SQL language.

Order by is the clause we use with “SELECT” statement in Hive queries, which helps sort data. Order by clause use columns on Hive tables for sorting particular column values mentioned with Order by. For whatever the column name we are defining the order by clause the query will selects and display results by ascending or descending order the particular column values.

**DISTRIBUTE BY:** Distribute BY clause used on tables present in Hive. Hive uses the columns in Distribute by to distribute the rows among reducers. All Distribute BY columns will go to the same reducer.

* It ensures each of N reducers gets non-overlapping ranges of column
* It doesn’t sort the output of each reducer

**CLUSTER BY:** Cluster By used as an alternative for both Distribute BY and Sort BY clauses in Hive-QL.

Cluster BY clause used on tables present in Hive. Hive uses the columns in Cluster by to distribute the rows among reducers. Cluster BY columns will go to the multiple reducers.

* It ensures sorting orders of values present in multiple reducers

For example, Cluster By clause mentioned on the Id column name of the table employees\_guru table. The output when executing this query will give results to multiple reducers at the back end. But as front end it is an alternative clause for both Sort By and Distribute By.

**Q 16. Difference between "Internal Table" and "External Table" and Mention when to choose “Internal Table” and “External Table” in Hive?**

**Ans -** In a typical table, the data is stored in the database; however, in an external table, the data is stored in files in an external stage. External tables store file-level metadata about the data files, such as the filename, a version identifier and related properties.

Internal tables are useful if you want Hive to manage the complete lifecycle of your data including the deletion, whereas external tables are useful when the files are being used outside of Hive.

**Q 17. Where does the data of a Hive table get stored?**

**Ans -** Hive stores its database and table metadata in a metastore, which is a database or file backed store that enables easy data abstraction and discovery.

**Q 18. Is it possible to change the default location of a managed table?**

**Ans -** Yes, you can do it by using the clause – LOCATION '<hdfs\_path>' we can change the default location of a managed table.

**Q 19. What is a metastore in Hive? What is the default database provided by Apache Hive for metastore?**

**Ans -** Hive Metastore projects types from an underlying store to supported HSQL types, and stores information about the location of the underlying data. This data is stored in the metastore database, which is usually MySQL, Postgres, or Derby. But the database itself is only an implementation detail.

Derby is the default database for the embedded metastore.

**Q 20. Why does Hive not store metadata information in HDFS?**

**Ans -** 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.

**Q 21. What is a partition in Hive? And Why do we perform partitioning in Hive?**

**Ans -** The partitioning in Hive means dividing the table into some parts based on the values of a particular column like date, course, city or country. The advantage of partitioning is that since the data is stored in slices, the query response time becomes faster.

As we know that Hadoop is used to handle the huge amount of data, it is always required to use the best approach to deal with it. The partitioning in Hive is the best example of it.

**Q 22. What is the difference between dynamic partitioning and static partitioning?**

**Ans -** Dynamic Partition takes more time in loading data compared to static partition. When you have large data stored in a table then the Dynamic partition is suitable. If you want to partition a number of columns but you don't know how many columns then also dynamic partition is suitable.

**Q 23. How do you check if a particular partition exists?**

**Ans -** Basically, with the following query, we can check whether a particular partition exists or not: SHOW PARTITIONS table\_name ...

1. [db\_name.] : Is an optional clause. This is used to list partitions of the table from a given database.
2. [PARTITION(partition\_spec)] : Is an optional clause. This is used to list a specific partition of a table.

**Q 24. How can you stop a partition form being queried?**

**Ans -** By using the ENABLE OFFLINE clause with ALTER TABLE atatement.

**Q 25. Why do we need buckets? How Hive distributes the rows into buckets?**

**Ans -** Bucketing in hive is useful when dealing with large datasets that may need to be segregated into clusters for more efficient management and to be able to perform join queries with other large datasets. The primary use case is in joining two large datasets involving resource constraints like memory limits.

**Q 26. In Hive, how can you enable buckets?**

**Ans -** So, we can enable dynamic bucketing while loading data into hive table By setting this property. ii. Moreover, it will automatically set the number of reduce tasks to be equal to the number of buckets mentioned in the table definition (for example 32 in our case).

**Q 27. How does bucketing help in the faster execution of queries?**

**Ans -** In bucketing, the partitions can be subdivided into buckets based on the hash function of a column. It gives extra structure to the data which can be used for more efficient queries.

**Q 28. How to optimise Hive Performance? Explain in very detail.**

**Ans -** Following are the Hive optimization techniques for Hive Performance Tuning, let’s discuss them one by one:

**1-** Tez-Execution Engine in Hive

Tez Execution Engine – Hive Optimization Techniques, to increase the Hive performance of our hive query by using our execution engine as Tez. On defining Tez, it is a new application framework built on Hadoop Yarn.

That executes complex-directed acyclic graphs of general data processing tasks. However, we can consider it to be a much more flexible and powerful successor to the map-reduce framework.  
  
In addition, to write native YARN applications on Hadoop that bridges the spectrum of interactive and batch workloads Tez offers an API framework to developers. To be more specific,  to work with petabytes of data over thousands of nodes it allows those data access applications.

**2 -** Usage of Suitable File Format in Hive

ORCFILE File Formate – Hive Optimization Techniques, if we use appropriate file format on the basis of data. It will drastically increase our query performance.

Basically, for increasing your query performance ORC file format is best suitable. Here, ORC refers to Optimized Row Columnar. That implies we can store data in an optimized way than the other file formats.

To be more specific, ORC reduces the size of the original data up to 75%. Hence,  data processing speed also increases. On comparing to Text, Sequence and RC file formats, ORC shows better performance.

Basically, it contains rows data in groups. Such as Stripes along with a file footer.  Therefore, we can say when Hive is processing the data ORC format improves the performance.

**3 -** Hive Partitioning

Hive Partition – Hive Optimization Techniques, Hive reads all the data in the directory Without partitioning. Further, it applies the query filters on it.  Since all data has to be read this is a slow as well as expensive.  
  
Also, users need to filter the data on specific column values frequently. Although, users need to understand the domain of the data on which they are doing analysis, to apply the partitioning in the Hive.  
  
Basically, by Partitioning all the entries for the various columns of the dataset are segregated and stored in their respective partition.

Hence, While we write the query to fetch the values from the table, only the required partitions of the table are queried. Thus it reduces the time taken by the query to yield the result.

**4 -** Bucketing in Hive

**Bucketing in Hive** – Hive Optimization Techniques, let’s suppose a scenario. At times, there is a huge dataset available. However, after partitioning on a particular field or fields, the partitioned file size doesn’t match with the actual expectation and remains huge.

Still, we want to manage the partition results into different parts. Thus, to solve this issue of partitioning, Hive offers Bucketing concept. Basically,  that allows the user to divide table data sets into more manageable parts.  
  
Hence, to maintain parts that are more manageable we can use Bucketing. Through it, the user can set the size of the manageable parts or Buckets too.

**5 -** Vectorization In Hive

Vectorization In Hive – Hive Optimization Techniques, to improve the performance of operations we use Vectorized query execution. Here operations refer to scans, aggregations, filters, and joins. It happens by performing them in batches of 1024 rows at once instead of single row each time.  
  
However, this feature is introduced in Hive 0.13. It significantly improves query execution time, and is easily enabled with two parameters settings:  
  
**set hive.vectorized.execution = true  
  
set hive.vectorized.execution.enabled = true**

**6 -** Cost-Based Optimization in Hive (CBO)

Cost-Based Optimization in Hive – Hive Optimization Techniques, before submitting for final execution Hive optimizes each Query’s logical and physical execution plan. Although, until now these optimizations are not based on the cost of the query.  
  
However, CBO, performs, further optimizations based on query cost in a recent addition to Hive. That results in potentially different decisions: how to order joins, which type of join to perform, the degree of parallelism and others.  
  
To use CBO, set the following parameters at the beginning of your query:  
  
**set hive.cbo.enable=true;  
  
set hive.compute.query.using.stats=true;  
  
set hive.stats.fetch.column.stats=true;  
  
set hive.stats.fetch.partition.stats=true;**  
Then, prepare the data for CBO by running Hive’s “analyze” command to collect various statistics on the tables for which we want to use CBO.

**7 -** Hive Indexing

Hive Index – Hive Optimization Techniques, one of the best ways is Indexing. To increase your query performance indexing will definitely help. Basically, for the original table use of indexing will create a separate called index table which acts as a reference.  
  
As we know, there are many numbers of rows and columns, in a**Hive table**. Basically, it will take a large amount of time if we want to perform queries only on some columns without indexing. Because queries will be executed on all the columns present in the table.  
  
Moreover,  there is no need for the query to scan all the rows in the table while we perform a query on a table that has an index, it turned out as the major advantage of using indexing. Further, it checks the index first and then goes to the particular column and performs the operation.  
Hence, maintaining indexes will be easier for Hive query to look into the indexes first and then perform the needed operations within less amount of time. Well, time is the only factor that everyone focuses on, eventually.

This was all about Hive Optimization Techniques Tutorial. Hope you like our explanation of Hive Performance Tuning.

So, this was all in Hive Query Optimization Techniques. Hope you like our explanation.

**Q 29. What is the use of Hcatalog?**

**Ans -** HCatalog is a tool that allows you to access Hive metastore tables within Pig, Spark SQL, and/or custom MapReduce applications. HCatalog has a REST interface and command line client that allows you to create tables or do other operations. You then write your applications to access the tables using HCatalog libraries.

**Q 30. Explain about the different types of join in Hive.**

**Ans –** This section describes the types of joins you can use to obtain specific information.

[**Cross join**](https://www.ibm.com/docs/en/SSULQD_7.1.0/com.ibm.nz.dbu.doc/c_dbuser_cross_join.html)**:** A cross join returns all possible combinations of rows of two tables (also called a Cartesian product).

[**Join/inner join**](https://www.ibm.com/docs/en/SSULQD_7.1.0/com.ibm.nz.dbu.doc/c_dbuser_join_inner_join.html)**:** An inner join, also known as a simple join, returns rows from joined tables that have matching rows. It does not include rows from either table that have no matching rows in the other.

[**Left outer join/left join**](https://www.ibm.com/docs/en/SSULQD_7.1.0/com.ibm.nz.dbu.doc/c_dbuser_left_outer_join.html)**:** Returns all the rows from the left table that are specified in the left outer join clause, not just the rows in which the columns match.

[**Right outer join/right join**](https://www.ibm.com/docs/en/SSULQD_7.1.0/com.ibm.nz.dbu.doc/c_dbuser_right_outer_join.html)**:** Returns all the rows from the right table that are specified in the right outer join clause, not just the rows in which the columns match.

[**Full outer join**](https://www.ibm.com/docs/en/SSULQD_7.1.0/com.ibm.nz.dbu.doc/c_dbuser_full_outer_join.html)**:** A full outer join returns all joined rows from both tables, plus one row for each unmatched left row (extended with nulls on the right), plus one row for each unmatched right row (extended with nulls on the left).

**Q 31. Is it possible to create a Cartesian join between 2 tables, using Hive?**

**Ans -** In this recipe, you will learn how to use a cross join in Hive. Cross join, also known as Cartesian product, is a way of joining multiple tables in which all the rows or tuples from one table are paired with the rows and tuples from another table.

**Q 32. Explain the SMB Join in Hive?**

**Ans -** SMB is a join performed on bucket tables that have the same sorted, bucket, and join condition columns. It reads data from both bucket tables and performs common joins (map and reduce triggered) on the bucket tables.

**Q 33. What is the difference between order by and sort by which one we should use?**

**Ans - ORDER BY** performs a total ordering of the query result set. This means that all the data is passed through a single reducer, which may take an unacceptably long time to execute for larger data sets.

**SORT BY** orders the data only within each reducer, thereby performing a local ordering, where each reducer’s output will be sorted. You will not achieve a total ordering on the dataset. Better performance is traded for total ordering.

**Q 34. What is the usefulness of the DISTRIBUTED BY clause in Hive?**

**Ans -** 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.

**Q 35. How does data transfer happen from HDFS to Hive?**

**Ans -**

1. Ingest the data. You create a single Sqoop import command that imports data from diverse data sources, such as a relational database, into HDFS.
2. Convert the data to ORC format. ...
3. Incrementally update the imported data.

**Q 36. Wherever (Different Directory) I run the hive query, it creates a new metastore\_db, please explain the reason for it?**

**Ans -** Whenever you run the hive in embedded mode, it creates the local metastore. And before creating the metastore it looks whether metastore already exist or not. This property is defined in configuration file hive-site.xml. Property is “javax.jdo.option.ConnectionURL” with default value “jdbc:derby:;databaseName=metastore\_db;create=true”. So to change the behavior change the location to absolute path, so metastore will be used from that location.

**Q 37. What will happen in case you have not issued the command: ‘SET hive.enforce.bucketing=true;’ before bucketing a table in Hive?**

**Ans -** The command: 'SET hive. enforce. bucketing=true;' allows one to have the correct number of reducer while using 'CLUSTER BY' clause for bucketing a column. In case it's not done, one may find the number of files that will be generated in the table directory to be not equal to the number of buckets.

**Q 38.Can a table be renamed in Hive?**

**Ans -** You can rename the table name in the hive. You need to use the alter command. This command allows you to change the table name as shown below.

ALTER TABLE name RENAME TO new\_name ALTER TABLE name ADD COLUMNS (col\_spec[, col\_spec ...])

**Q 39. Write a query to insert a new column(new\_col INT) into a hive table at a position before an existing column (x\_col)**

**Ans -** The following query will insert a new column:ALTER TABLE table\_name

CHANGE COLUMN new\_col INT

BEFORE x\_col

**Q 40. What is serde operation in HIVE?**

**Ans -** A SerDe allows Hive to read in data from a table, and write it back out to HDFS in any custom format. Anyone can write their own SerDe for their own data formats. See Hive SerDe for an introduction to SerDes.

**Q 41. Explain how Hive Deserializes and serialises the data?**

**Ans -** SerDe is short for Serializer/Deserializer. Hive uses the SerDe interface for IO. The interface handles both serialization and deserialization and also interpreting the results of serialization as individual fields for processing.

**Q 42. Write the name of the built-in serde in hive.**

**Ans –**

**Q 43. What is the need of custom Serde?**

**Ans -** Main use of SerDe interface is for IO operations. A SerDe allows hive to read the data from the table and write it back to the HDFS in any custom format. If we have unstructured data, then we use RegEx SerDe which will instruct hive how to handle that record. We can also write our own Custom SerDe in any format.

**Q 44. Can you write the name of a complex data type(collection data types) in Hive?**

**Ans -** Similar to Spark, Hive also support complex data types which includes Array, Map, Struct and union. Array is used to store the list of elements. Map is used to store key/value pair. Struct is for parent and child assosiations.

**Q 45.Can hive queries be executed from script files? How?**

**Ans -** We can execute Hive queries from the script files using the source command.

**Q 46. What are the default record and field delimiter used for hive text files?**

**Ans -** The default record delimiter is − \n And the filed delimiters are − \001,\002,\003

**Q 47. How do you list all databases in Hive whose name starts with s?**

**Ans -** To list out the databases in Hive warehouse, enter the command 'show databases'. The database creates in a default location of the Hive warehouse. In Cloudera, Hive database store in a /user/hive/warehouse.

**Q 48. What is the difference between LIKE and RLIKE operators in Hive?**

**Ans -** LIKE is an operator similar to LIKE in SQL. We use LIKE to search for string with similar text.

**E.g.** user\_name LIKE ‘%Smith’

RLIKE (Right-Like) is a special function in Hive where if any substring of A matches with B then it evaluates to true. It also obeys Java regular expression pattern. Users don't need to put % symbol for a simple match in RLIKE.

Hive provides RLIKE operator that can be used for searching advanced Regular Expressions in Java.

**E.g.** user\_name RLIKE ‘.(Smith|Sam).’

**Q 49. How to change the column data type in Hive?**

**Ans -** By using this command below one can change the column data type: ALTER TABLE table\_name CHANGE column\_name column\_name new\_datatype; I hope this works.

**Q 50. How will you convert the string ’51.2’ to a float value in the particular column?**

**Ans –**

1. public class StringToFloatExample{
2. public static void main(String args[]){
3. String s="23.6";
4. float f=Float.parseFloat("23.6");
5. System.out.println(f);
6. }}

**Q 51.What will be the result when you cast ‘abc’ (string) as INT?**

**Ans -** Hive will return NULL

**Q 52.What does the following query do?**

**a. INSERT OVERWRITE TABLE employees**

**b. PARTITION (country, state)**

**c. SELECT ..., se.cnty, se.st**

**d. FROM staged\_employees se;**

**Ans -** It creates partition on table employees with partition values coming from the columns in the select clause. It is called Dynamic partition insert.

**Q 53.Write a query where you can overwrite data in a new table from the existing table.**

**Ans –**

1. Use select \* INTO D1.dbo.T1 FROM D2.dbo.T1.
2. Then refreshed D1 from prod.
3. Then truncate T1 with the following step: SELECT COUNT(\*) AS BeforeTruncateCount FROM T1; GO TRUNCATE TABLE T1; GO SELECT COUNT(\*) AS AfterTruncateCount FROM T1; GO.

**Q 54. What is the maximum size of a string data type supported by Hive? Explain how Hive supports binary formats.**

**Ans -** The maximum size of a string data type supported by Hive is 2 GB. Hive supports the text file format by default, and it also supports the binary format sequence files, ORC files, Avro data files, and Parquet files. Sequence file: It is a splittable, compressible, and row-oriented file with a general binary format.

**Q 55. What File Formats and Applications Does Hive Support?**

**Ans -** Hive supports all Avro types. However, Impala does not support complex or nested types with Avro, such as enum, array, fixed, map, union, and record (nested). In the next section, we will discuss how to import data in Parquet file format using Sqoop.

**Q 56. How do ORC format tables help Hive to enhance its performance?**

**Ans -** 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.

**Q 57. How can Hive avoid mapreduce while processing the query?**

**Ans -** You can make Hive avoid MapReduce to return query results by setting the hive. exec. mode.

**Q 58. What is view and indexing in hive?**

**Ans – View:** Views are generated based on user requirements. You can save any result set data as a view. The usage of view in Hive is same as that of the view in SQL. It is a standard RDBMS concept. We can execute all DML operations on a view.

**Indexing:** Introduction to Indexes in Hive. Indexes are a pointer or reference to a record in a table as in relational databases. Indexing is a relatively new feature in Hive. In Hive, the index table is different than the main table. Indexes facilitate in making query execution or search operation faster.

**Q 59. Can the name of a view be the same as the name of a hive table?**

**Ans -** No. The name of a view must be unique whne compared to all other tables and views present in the same database.

**Q 60.What types of costs are associated in creating indexes on hive tables?**

**Ans -** Basically, there is a processing cost in arranging the values of the column on which index is created since Indexes occupies.

**Q 61.Give the command to see the indexes on a table.**

**Ans -** To see the index for a specific table use SHOW INDEX: SHOW INDEX FROM yourtable; To see indexes for all tables within a specific schema you can use the STATISTICS table from INFORMATION\_SCHEMA: SELECT DISTINCT TABLE\_NAME, INDEX\_NAME FROM INFORMATION\_SCHEMA.

**Q 62. Explain the process to access subdirectories recursively in Hive queries.**

**Ans -** We can use following commands in Hive to recursively access sub-directories:

hive> Set mapred.input.dir.recursive=true;

hive> Set hive.mapred.supports.subdirectories=true;

Once above options are set to true, Hive will recursively access sub-directories of a directory in MapReduce.

**Q 63.If you run a select \* query in Hive, why doesn't it run MapReduce?**

**Ans -** However, while using "select <column> from <tablename>", Hive requires a map-reduce job since it needs to extract the 'column' from each row by parsing it from the file it loads.

**Q 64. What are the uses of Hive Explode?**

**Ans -** Lateral View Explode is another function in Hive that is used to split a column, but instead of creating multiple rows, it creates multiple columns. This function is beneficial when working with maps. It allows us to split a map column into multiple columns, each containing one key-value pair from the map.

**Q 65. What is the available mechanism for connecting applications when we run Hive as a server?**

**Ans -**

* **Thrift Client:** Using Thrift, we can call Hive commands from various programming languages, such as C++, PHP, Java, Python, and Ruby.
* [**JDBC Driver**](https://intellipaat.com/blog/java-jdbc/)**:** JDBC Driver enables accessing data with JDBC support, by translating calls from an application into SQL and passing the SQL queries to the Hive engine.
* **ODBC Driver:** It implements the ODBC API standard for the Hive DBMS, enabling ODBC-compliant applications to interact seamlessly with Hive.

**Q 66. Can the default location of a managed table be changed in Hive?**

**Ans -** Can do it by using the clause – LOCATION '<hdfs\_path>' we can change the default location of a managed table.

**Q 67. What is the Hive ObjectInspector function?  
Ans -** Hive ObjectInspector is a group of flexible APIs to inspect value in different data representation, and developers can extend those API as needed, so technically, object inspector supports arbitrary data type in java.

**Q 68. What is UDF in Hive?**

**Ans -** The Db2 Big SQL environment in Hadoop includes the Hive user-defined functions package. This set of functions is an optional package that you can install to use some of the Hive open source user-defined functions in your Db2 Big SQL queries.

**Q 69. Write a query to extract data from hdfs to hive.**

**Ans –**

1. Move .CSV data into HDFS: ...
2. Create an external table. ...
3. Create the ORC table. ...
4. Insert the data from the external table to the Hive ORC table.
5. Verify that you imported the data into the ORC-formatted table correctly:

**Q 70. What is TextInputFormat and SequenceFileInputFormat in hive.**

**Ans - SequenceFile** is a flat file consisting of binary key/value pairs. It is extensively used in MapReduce as input/output formats. It is also worth noting that, internally, the temporary outputs of maps are stored using SequenceFile.

TextInputFormatis one of the file formats of Hadoop. It is a default type format of hadoop MapReduce that is if we do not specify any file formats then RecordReader will consider the input file format as textinputformat.

**Q 71. How can you prevent a large job from running for a long time in a hive?**

**Ans -** This can be achieved by setting the MapReduce jobs to execute in strict mode set hive.mapred.mode=strict;

The strict mode ensures that the queries on partitioned tables cannot execute without defining a WHERE clause.

**Q 72. When do we use explode in Hive?**

**Ans -** Lateral View Explode is another function in Hive that is used to split a column, but instead of creating multiple rows, it creates multiple columns. This function is beneficial when working with maps. It allows us to split a map column into multiple columns, each containing one key-value pair from the map.

**Q 73. Can Hive process any type of data formats? Why? Explain in very detail**

**Ans -** Hive supports several sizes of integer and floating-point types, a Boolean type, and character strings of arbitrary length. Hive v0. 8.0 added types for timestamps and binary fields. Table 3-1 lists the primitive types supported by Hive.

**Q 74. Whenever we run a Hive query, a new metastore\_db is created. Why?**

**Ans -** A local metastore is created when we run Hive in an embedded mode. Before creating, it checks whether the metastore exists or not, and this metastore property is defined in the configuration file, hive-site.xml. The property is:

javax.jdo.option.ConnectionURL

with the default value:

jdbc:derby:;databaseName=metastore\_db;create=true

Therefore, we have to change the behavior of the location to an absolute path so that from that location the metastore can be used.

**Q75. Can we change the data type of a column in a hive table? Write a complete query.**

**Ans -** By using this command below one can change the column data type:

ALTER TABLE table\_name CHANGE column\_name column\_name new\_datatype;

**Q 76. While loading data into a hive table using the LOAD DATA clause, how do you specify it is a hdfs file and not a local file ?**

**Ans –**

**Q 77. What is the precedence order in Hive configuration?**

**Ans -  There is a precedence hierarchy for setting properties:**

* **The Hive SET command**
* **The command line -hiveconf option**
* **hive-site.xml**
* **hive-default.xml**
* **hadoop-site.xml (or, equivalently, hdfs-site.xml, core-site.xml, and mapred-site.xml)**
* **hadoop-default.xml (or, equivalently, hdfs-default.xml, core-default.xml, and mapred-default.xml)**

**Q 78. Which interface is used for accessing the Hive metastore?**

**Ans -** WebHCat API web interface can be used for Hive commands. It is a REST API that allows applications to make HTTP requests to access the Hive metastore (HCatalog DDL). It also enables users to create and queue Hive queries and commands.

**Q 79. Is it possible to compress json in the Hive external table ?**

**Q 80. What is the difference between local and remote metastores?**

**Ans - Local Metastore:-** Here metastore service still runs in the same JVM as Hive but it connects to a database running in a separate process either on same machine or on a remote machine.

**Remote Metastore:-** Metastore runs in its own separate JVM not on hive service JVM.

**Q 81. What is the purpose of archiving tables in Hive?**

**Ans -** You can use Hadoop archiving to reduce the number of hdfs files in the Hive table partition. Hive has built in functions to convert Hive table partition into Hadoop Archive (HAR). HAR does not compress the files, it is analogous to the Linux tar command.

**Q 82. What is DBPROPERTY in Hive?**

**Ans -** The DB properties are nothing but mentioning the details about the database created by the user. Suppose the name of the user, the type of the database and the tables it has, the date on which the database is created etc. This makes the other user easy the recognize the database and use it according to the requirement.

**Q 83. Differentiate between local mode and MapReduce mode in Hive.**

**Ans -** Local mode is actually a local simulation of MapReduce in Hadoop's LocalJobRunner class. MapReduce mode (also known as Hadoop mode): Pig is executed on the Hadoop cluster. In this case, the Pig Script gets converted into a series of MapReduce jobs that are then run on the Hadoop cluster.