HIVE INTERVIEW QUESTIONS

<https://data-flair.training/blogs/apache-hive-interview-questions/>

<https://www.edureka.co/blog/interview-questions/hive-interview-questions/>

<https://intellipaat.com/blog/interview-question/hive-interview-questions/>

<https://www.tutorialspoint.com/hive/hive_interview_questions.htm>

<https://acadgild.com/blog/top-15-hive-interview-question-2017>

<https://www.dezyre.com/article/hive-interview-questions-and-answers-for-2018/246>

<https://career.guru99.com/top-30-hive-interview-questions/>

<https://iqsanswers.com/hive-interview-questions-answers/>

<https://www.zeolearn.com/interview-questions/hive>

**What is Apache Hive?**

**Ans.** Basically, a tool which we call a data warehousing tool is Hive. However, Hive gives [**SQL queries**](https://data-flair.training/blogs/sql-subquery/) 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.

**Que 2. What kind of applications is supported by Apache Hive?**

**Ans.** All those client applications which are written in [**Java**](https://data-flair.training/blogs/java-tutorial/), PHP, [**Python**](https://data-flair.training/blogs/python-tutorial-for-beginners/), C++ or Ruby by exposing its thrift server, Hive supports them.

**Que 3. 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.

**Que 4. 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.   
  
**Que 5. What is a metastore in Hive?**

**Ans**. Basically, to store the metadata information in the Hive we use [**Metastore**](https://data-flair.training/blogs/apache-hive-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.

**Que 6. 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.  
  
**Que 7. What is the difference between local and remote metastore?**

**Ans. Local Metastore:**

It is the metastore service runs in the same [**JVM**](https://data-flair.training/blogs/java-virtual-machine-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 this configuration, the metastore service runs on its own separate JVM and not in the Hive service JVM.  
  
**Que 8. 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.

**Que 9. 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.

[**Read more about Hive internal tables vs External tables**](https://data-flair.training/blogs/hive-internal-tables-vs-external-tables-comparison/)

**Que 10. Is it possible to change the default location of a managed table?**

**Ans.** Yes, by using the clause – LOCATION ‘<hdfs\_path>’ we can change the default location of a managed table.

**Hive Interview Questions for Freshers- Q. 1,2,3,4,5,7,8,9,10**

**Hive Interview Questions for Experience- Q. 6**  
  
**Que 11. When should we use SORT BY instead of ORDER BY?**

**Ans**. Despite [**ORDER BY**](https://data-flair.training/blogs/hiveql-order-by-query/) we should use SORT BY. Especially while we have to sort huge datasets. The reason is SORT BY clause sorts the data using multiple reducers. ORDER BY sorts all of the data together using a single reducer. Hence, using ORDER BY will take a lot of time to execute a large number of inputs.

**Que 12. 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**](https://data-flair.training/blogs/apache-hive-partitioning/), in other words, it is a sub-directory in the table directory.

**Que 13. 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.  
  
**Que 14. 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.

**Que 15. Why do we need buckets?**

**Ans.** Basically, for [**performing bucketing**](https://data-flair.training/blogs/bucketing-in-hive/) 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.

**Que 16. 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

**Que 17.  What is indexing and why do we need it?**

**Ans.** [**Hive index**](https://data-flair.training/blogs/hive-view-hive-index/) is a Hive query optimization techniques. Basically, we use it to speed up the access of a column or set of columns in a Hive database. Since, the database system does not need to read all rows in the table to find the data with the use of the index, especially that one has selected.

**Que 18. 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.

**Que 19. Where is table data stored in Apache Hive by default?**

**Ans.** hdfs: //namenode\_server/user/hive/warehouse

**Que 20. Are multi-line comments supported in Hive?**

**Ans.**No

**Hive Interview Questions for Freshers- Q. 12,13,14,15,17,18,19,20**

**Hive Interview Questions for Experience- Q. 11,16**

**Que 21. What is ObjectInspector functionality?**

**Ans.** To analyze the structure of individual columns and the internal structure of the row objects we use ObjectInspector. Basically, it provides access to complex objects which can be stored in multiple formats in Hive.

**Que 22. Explain about the different types of join in Hive**.

**Ans.** There are  4 different types of [**joins in HiveQL**](https://data-flair.training/blogs/hive-join/) –

* **JOIN-** It is very similar to Outer Join in SQL
* **FULL OUTER JOIN –** This join Combines the records of both the left and right outer tables. Basically, that fulfill the join condition.
* **LEFT OUTER JOIN-** Through this Join, All the rows from the left table are returned even if there are no matches in the right table.
* **RIGHT OUTER JOIN –** Here also, all the rows from the right table are returned even if there are no matches in the left table.

**Que 23. How can you configure remote metastore mode in Hive?**  
**Ans.** Basically, hive-site.xml file has to be configured with the below property, to configure metastore in Hive –  
hive.metastore.uris  
 thrift: //node1 (or IP Address):9083  
 IP address and port of the metastore host

**Que 24. Is it possible to change the default location of Managed Tables in Hive, if so how?**

**Ans.** Yes, by using the LOCATION keyword while creating the managed table, we can change the default location of Managed tables. But the one condition is, the user has to specify the storage path of the managed table as the value of the LOCATION keyword.

**Que 25.  How does data transfer happen from HDFS to Hive?**

**Ans.**Basically, the user need not LOAD DATA that moves the files to the /user/hive/warehouse/. But only if data is already present in HDFS. Hence, using the keyword external that creates the table definition in the hive metastore  the user just has to define the table.  
Create external table table\_name (  
 id int,  
 myfields string  
)  
location ‘/my/location/in/hdfs’;

**Que 26. What are the different components of a Hive architecture?**

**Ans.** There are several components of [**Hive Architecture**](https://data-flair.training/blogs/apache-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.

**Que 27. Wherever (Different Directory) I run the hive query, it creates new metastore\_db, please explain the reason for it?**

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

**Que 28. Is it possible to use the same metastore by multiple users, in case of the embedded hive?**

**Ans.** No, we cannot use metastore in sharing mode. It is possible to use it in standalone “real” database. Such as MySQL or PostGresSQL.

**Que 29. Usage of Hive.**

**Ans.** Here, we will look at following [**Hive**](https://hive.apache.org/) usages.  
– We use Hive for Schema flexibility as well as evolution.  
– Moreover, it is possible to portion and bucket, tables in Apache Hive.  
– Also, we can use JDBC/ODBC drivers, since they are available in Hive.

**Que 30. 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.

**Define the difference between Hive and HBase?**

|  |  |
| --- | --- |
| **Hive vs HBase** | |
| **HBase** | **Hive** |
| 1. HBase is built on the top of HDFS | 1. It is a data warehousing infrastructure |
| 2. HBase operations run in a real-time on its database rather | 2. Hive queries are executed as MapReduce jobs internally |
| 3. Provides low latency to single rows from huge datasets | 3. Provides high latency for huge datasets |
| 4. Provides random access to data | 4. Provides random access to data |

**5. 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.

**6. 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.

**7. 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.

**8. Scenario:**

***Suppose I have installed Apache Hive on top of my Hadoop cluster using default metastore configuration. Then, what will happen if we have multiple clients trying to access Hive at the same time?***

The default metastore configuration allows only one Hive session to be opened at a time for accessing the metastore. Therefore, if multiple clients try to access the metastore at the same time, they will get an error. One has to use a standalone metastore, i.e. Local or remote metastore configuration in Apache Hive for allowing access to multiple clients concurrently.

Following are the steps to configure MySQL database as the local metastore in Apache Hive:

* One should make the following changes in hive-site.xml:
  + *javax.jdo.option.ConnectionURL* property should be set to jdbc:*mysql*:*//host/*dbname?createDataba  
    *seIfNotExist=true.*
  + *javax.jdo.option.ConnectionDriverName*property should be set to *com.mysql.jdbc.Driver.*
  + One should also set the username and password as:
    - javax.jdo.option.ConnectionUserName is set to desired username.
    - javax.jdo.option.ConnectionPassword is set to the desired password.
* The JDBC driver JAR file for MySQL must be on the Hive’s classpath, i.e. The jar file should be copied into the Hive’s lib directory.
* Now, after restarting the Hive shell, it will automatically connect to the MySQL database which is running as a standalone metastore.

**13. 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.

**14. 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.

**15. 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.

**16. How can you add a new partition for the month December in the above partitioned table?**

For adding a new partition in the above table partitioned\_transaction, we will issue the command give below:

*ALTER TABLE partitioned\_transaction ADD PARTITION (month=’Dec’) LOCATION  ‘/partitioned\_transaction’;*

***Note:*** I suggest you to go through the dedicated blog on [***Hive Commands***](https://www.edureka.co/blog/hive-commands-with-examples) where all the commands present in Apache Hive have been explained with an example.

**17. 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>

**18. 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;*

* By default, hive.exec.dynamic.partition configuration property is set to False in case you are using Hive whose version is prior to 0.9.0.
* hive.exec.dynamic.partition.mode is set to strict by default. Only in non – strict mode Hive allows all partitions to be dynamic.

**Why do we need buckets?**

There are two main reasons for performing bucketing to a partition:

* A[***map side join***](https://www.edureka.co/blog/map-side-join-vs-join/)requires the data belonging to a unique join key to be present in the same partition. But what about those cases where your partition key differs from that of join key? Therefore, in these cases you can perform a map side join by bucketing the table using the join key.
* Bucketing makes the sampling process more efficient and therefore, allows us to decrease the query time.

**21. What will happen in case you have not issued the command:  *‘SET hive.enforce.bucketing=true;’* before bucketing a table in Hive in Apache Hive 0.x or 1.x?**

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. As an alternative, one may also set the number of reducer equal to the number of buckets by using *set mapred.reduce.task = num\_bucket*.

**23. Scenario:**

***Suppose, I have a CSV file – ‘sample.csv’ present in ‘/temp’ directory with the following entries:***

**id first\_name last\_name email gender ip\_address**

1 Hugh Jackman hughjackman@cam.ac.uk Male 136.90.241.52

2 David Lawrence dlawrence1@gmail.com Male 101.177.15.130

3 Andy Hall andyhall2@yahoo.com Female 114.123.153.64

4 Samuel Jackson samjackson231@sun.com Male 89.60.227.31

5 Emily Rose rose.emily4@surveymonkey.com Female 119.92.21.19

***How will you consume this CSV file into the Hive warehouse using built SerDe?***

SerDe stands for serializer/deserializer. A SerDe allows us to convert the unstructured bytes into a record that we can process using Hive. SerDes are implemented using Java. Hive comes with several built-in SerDes and many other third-party SerDes are also available.

Hive provides a specific SerDe for working with CSV files. We can use this SerDe for the sample.csv by issuing following commands:

*CREATE EXTERNAL TABLE sample*

*(id int, first\_name string,*

*last\_name string, email string,*

*gender string, ip\_address string)*

*ROW FORMAT SERDE ‘org.apache.hadoop.hive.serde2.OpenCSVSerde’*

*STORED AS TEXTFILE LOCATION ‘/temp’;*

Now, we can perform any query on the table ‘sample’:

*SELECT first\_name FROM sample WHERE gender = ‘male’;*

**24. Scenario:**

***Suppose, I have a lot of small CSV files present in /input directory in HDFS and I want to create a single Hive table corresponding to these files. The data in these files are in the format: {id, name, e-mail, country}. Now, as we know, Hadoop performance degrades when we use lots of small files.***

***So, how will you solve this problem where we want to create a single Hive table for lots of small files without degrading the performance of the system?***

One can use the SequenceFile format which will group these small files together to form a single sequence file. The steps that will be followed in doing so are as follows:

* Create a temporary table:

*CREATE TABLE temp\_table (id INT, name STRING, e-mail STRING, country STRING)*

*ROW FORMAT FIELDS DELIMITED TERMINATED BY ‘,’ STORED AS TEXTFILE;*

* Load the data into temp\_table:

*LOAD DATA INPATH ‘/input’ INTO TABLE temp\_table;*

* Create a table that will store data in SequenceFile format:

*CREATE TABLE sample\_seqfile*(*id INT, name STRING, e-mail STRING, country STRING)*

*ROW FORMAT FIELDS DELIMITED TERMINATED BY ‘,’ STORED AS SEQUENCEFILE;*

* Transfer the data from the temporary table into the sample\_seqfile table:

*INSERT OVERWRITE TABLE sample SELECT \* FROM temp\_table;*

Hence, a single SequenceFile is generated which contains the data present in all of the input files and therefore, the problem of having lots of small files is finally eliminated.

**Differentiate between Pig and Hive.**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Apache Pig** | **Apache Hive** |
| Nature | Procedural data flow language | Declarative SQL-like language |
| Application | Used for programming | Used for report creation |
| Used by | Researchers and programmers | Mainly Data Analysts |
| Operates on | The client-side of a cluster | The server-side of a cluster |
| Accessing raw data | Not as fast as HiveQL | Faster with in-built features |
| Schema or data type | Always defined in the script itself | Stored in the local database |
| Ease of learning | Takes little extra time and effort to master | Easy to learn from database experts |

**2. What is the definition of Hive? What is the present version of Hive? Explain ACID transactions in Hive.**

Hive is an open-source data warehouse system. We can use Hive for analyzing and querying large datasets. It’s similar to SQL. The present version of Hive is 0.13.1. Hive supports ACID (Atomicity, Consistency, Isolation, and Durability) transactions. ACID transactions are provided at row levels. Following are the options Hive uses to support ACID transactions:

* Insert
* Delete
* Update

**3. What is a Hive variable? What do we use it for?**

Hive variables are basically created in the Hive environment that is referenced by Hive scripting languages. They allow to pass some values to a Hive query when the query starts executing. They use the source command.

**What kind of data warehouse application is suitable for Hive? What are the types of tables in Hive?**

Hive is not considered a full database. The design rules and regulations of Hadoop and HDFS have put restrictions on what Hive can do. However, Hive is most suitable for data warehouse applications because it:

* Analyzes relatively static data
* Has less responsive time
* Does not make rapid changes in data

Although Hive doesn’t provide fundamental features required for Online Transaction Processing (OLTP), it is suitable for data warehouse applications in large datasets. There are two types of tables in Hive:

* Managed tables
* External tables

***Get a better understanding of Hive by going through this***[***Hive Tutorial***](https://intellipaat.com/tutorial/hadoop-tutorial/apache-hive/)***now!***

**5. Can we change the settings within a Hive session? If yes, how?**

Yes, we can change the settings within a Hive session using the SET command. It helps change the Hive job settings for an exact query. For example, the following command shows that buckets are occupied according to the table definition:

hive> SET hive.enforce.bucketing=true;

We can see the current value of any property by using SET with the property name. SET will list all the properties with their values set by Hive.

hive> SET hive.enforce.bucketing;

hive.enforce.bucketing=true

This list will not include the defaults of Hadoop. So, we should use the below code:

SET -v

It will list all the properties including the Hadoop defaults in the system.

***Interested in learning Hive? Well, we have a comprehensive master’s***[***Big Data Hadoop Course***](https://intellipaat.com/big-data-hadoop-training/)***to give you an idea of all concepts in Hive!***

**6. Is it possible to add 100 nodes when we already have 100 nodes in Hive? If yes, how?**

**Yes, we can add the nodes by following the below steps:**

**Step 1**: Take a new system; create a new username and password  
**Step 2**: Install SSH and with the master node setup SSH connections  
**Step 3**: Add ssh public\_rsa id key to the authorized keys file  
**Step 4**: Add the new DataNode hostname, IP address, and other details in /etc/hosts slaves file:

192.168.1.102 slave3.in slave3

**Step 5**: Start the DataNode on a new node  
**Step 6**: Login to the new node like suhadoop or:

ssh -X hadoop@192.168.1.103

**Step 7**: Start HDFS of the newly added slave node by using the following command:

./bin/hadoop-daemon.sh start data node

**Step 8**: Check the output of the jps command on the new node

**7. Explain the concatenation function in Hive with an example.**

The concatenate function will join the input strings. We can specify  
‘n’ number of strings separated by a comma.

**Example**:

CONCAT ('Intellipaat','-','is','-','a','-','eLearning',’-’,’provider’);

**Output**:

Intellipaat-is-a-eLearning-provider

Every time, we set the limits of the strings by ‘-‘. If it is common for every string, then Hive provides another command:

CONCAT\_WS

In this case, we have to specify the set limits of the operator first as follows:

CONCAT\_WS ('-',’Intellipaat’,’is’,’a’,’eLearning’,‘provider’);

**Output**:

Intellipaat-is-a-eLearning-provider

**Explain the Trim and Reverse functions in Hive with examples.**

The trim function will delete the spaces associated with a string.

**Example**:

TRIM(‘ INTELLIPAAT ‘);

**Output**:

INTELLIPAAT

To remove the leading space:

LTRIM(‘ INTELLIPAAT’);

To remove the trailing space:

RTRIM(‘INTELLIPAAT ‘);

* In the reverse function, characters are reversed in the string.

**Example**:

REVERSE(‘INTELLIPAAT’); O/P -> TAAPILLETNI

**9. How to change the column data type in Hive? Explain RLIKE in Hive.**

We can change the column data type by using ALTER and CHANGE as follows:

ALTER TABLE table\_name CHANGE column\_namecolumn\_namenew\_datatype;

For example, if we want to change the data type of the salary column from integer to bigint in the employee table, we can use the following:

ALTER TABLE employee CHANGE salary salary BIGINT;

**RLIKE**: Its full form is Right-Like and it is a special function in Hive. It helps examine two substrings, i.e., if the substring of A matches with B, then it evaluates to true.

**Example**:

‘Intellipaat’ RLIKE ‘tell’ ◊ True

‘Intellipaat’ RLIKE ‘^I.\*’ ◊ True (this is a regular expression)

***Learn more about***[***Apache Hive***](https://intellipaat.com/blog/what-is-apache-hive/)***from this detailed blog post now!***

**10. What are the components used in Hive Query Processor?**

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)
* Optimizer (ql/optimizer)

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

By using the below commands, we can access subdirectories recursively in Hive:

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

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

Hive tables can be pointed to the higher level directory, and this is suitable for the directory structure like:

/data/country/state/city/

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

Imagine that header records in a table are as follows:

System=…

Version=…

Sub-version=…

Suppose, we do not want to include the above three lines of headers in our Hive query. To skip the header lines from our table in Hive, we will set a table property.

CREATE EXTERNAL TABLE employee (

name STRING,

job STRING,

dob STRING,

id INT,

salary INT)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘ ‘ STORED AS TEXTFILE

LOCATION ‘/user/data’

TBLPROPERTIES("skip.header.line.count"="2”);

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

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.
* **ORC file**: Optimized row columnar (ORC) format file is a record-columnar and column-oriented storage file. It divides the table in row split. Each split stores the value of the first row in the first column and follows subsequently.
* **Avro data file**: It is the same as a sequence file that is splittable, compressible, and row-oriented but without the support of schema evolution and multilingual binding.
* **Parquet file:** In Parquet format, along with storing rows of data adjacent to one another, we can also store column values adjacent to each other such that both horizontally and vertically datasets are partitioned.

**15. What is the precedence order of Hive configuration?**

We are using a precedence hierarchy for setting properties:

1. The SET command in Hive
2. The command-line –hiveconf option
3. Hive-site.XML
4. Hive-default.xml
5. Hadoop-site.xml
6. Hadoop-default.xml

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

The hive.fetch.task.conversion property of Hive lowers the latency of MapReduce overhead, and in effect when executing queries such as SELECT, FILTER, LIMIT, etc. it skips the MapReduce function.

***If you have any doubts or queries related to Hive, get them clarified from Hadoop experts on our***[***Hive Community***](https://intellipaat.com/community/search?q=hive)***!***

**17. How can we improve the performance with ORC format tables in Hive?**

We can store Hive data in a highly efficient manner in an Optimized Row Columnar (ORC) file format. It can simplify many Hive file format limitations. We can improve the performance by using ORC files while reading, writing, and processing data.

Set hive.compute.query.using.stats-true;

Set hive.stats.dbclass-fs;

CREATE TABLE orc\_table (

idint,

name string)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘\:’

LINES TERMINATED BY ‘\n’

STORES AS ORC;

***Need a reason to learn Apache Hadoop and Hive? Well, go through this blog post to find out***[***Why Hadoop is the new black?***](https://intellipaat.com/blog/hadoop-is-the-new-black/)

**18. Explain the functionality of ObjectInspector.**

ObjectInspector helps analyze the internal structure of a row object and the individual structure of columns in Hive. It also provides a uniform way to access complex objects that can be stored in multiple formats in the memory.

* An instance of Java class
* A standard Java object
* A lazily initialized object

ObjectInspector tells the structure of the object and also the ways to access the internal fields inside the object.

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

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.

**22. What are the uses of Hive Explode?**

Hadoop Developers consider an array as their input and convert it into a separate table row. To convert complicated data types into desired table formats, Hive uses Explode.

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

* **Thrift Client**: Using Thrift, we can call Hive commands from various programming languages, such as C++, PHP, Java, Python, and Ruby.
* **JDBC Driver**: 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.

**24. How do we write our own custom SerDe?**

Mostly, end-users prefer writing a Deserializer instead of using SerDe as they want to read their own data format instead of writing to it, e.g., RegexDeserializer deserializes data with the help of the configuration parameter ‘regex’ and with a list of column names.

If our SerDe supports DDL (i.e., SerDe with parameterized columns and column types), we will probably implement a protocol based on DynamicSerDe, instead of writing a SerDe. This is because the framework passes DDL to SerDe through the ‘Thrift DDL’ format and it’s totally unnecessary to write a “Thrift DDL” parser.

***Are you interested in learning Hadoop from experts? Enroll in our***[***Hadoop Course in Bangalore***](https://intellipaat.com/big-data-hadoop-training-bangalore/)***now!***

**25. Mention various date types supported by Hive.**

The timestamp data type stores date in the java.sql.timestamp format.

**Three collection data types in Hive are:**

* Arrays
* Maps
* Structs

**26. Can we run UNIX shell commands from Hive? Can Hive queries be executed from script files? If yes, how? Give an example.**

Yes, we can run UNIX shell commands from Hive using an ‘**!**‘ mark before the command. For example, **!pwd** at Hive prompt will display the current directory.  
We can execute Hive queries from the script files using the source command.

**Example**:

Hive> source /path/to/file/file\_with\_query.hql

**What is the need for custom Serde?**

Depending on the nature of data the user has, the inbuilt SerDe may not satisfy the format of the data. SO users need to write their own java code to satisfy their data format requirements.

**Why do we need Hive?**

Hive is a tool in Hadoop ecosystem which provides an interface to organize and query data in a databse like fashion and write SQL like queries. It is suitable for accessing and analyzing data in Hadoop using SQL syntax.

What is the default location where hive stores table data?

hdfs://namenode\_server/user/hive/warehouse

What are the three different modes in which hive can be run?

* Local mode
* Distributed mode
* Pseudodistributed mode

**What is the importance of .hiverc file?**

It is a file containing list of commands needs to run when the hive CLI starts. For example setting the strict mode to be true etc.

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

The default record delimiter is − \n

And the filed delimiters are − \001,\002,\003

**What do you mean by schema on read?**

The schema is validated with the data when reading the data and not enforced when writing data.

**How do you list all databases whose name starts with p?**

SHOW DATABASES LIKE ‘p.\*’

**What does the “USE” command in hive do?**

With the use command you fix the database on which all the subsequent hive queries will run.

**How can you delete the DBPROPERTY in Hive?**

There is no way you can delete the DBPROPERTY.

**What is the significance of the line**

set hive.mapred.mode = strict;

It sets the mapreduce jobs to strict mode.By which the queries on partitioned tables can not run without a WHERE clause. This prevents very large job running for long time.

**How do you check if a particular partition exists?**

This can be done with following query

SHOW PARTITIONS table\_name PARTITION(partitioned\_column=’partition\_value’)

**Which java class handles the Input record encoding into files which store the tables in Hive?**

org.apache.hadoop.mapred.TextInputFormat

**Which java class handles the output record encoding into files which result from Hive queries?**

org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat

**What is the significance of ‘IF EXISTS” clause while dropping a table?**

When we issue the command DROP TABLE IF EXISTS table\_name

Hive throws an error if the table being dropped does not exist in the first place.

**When you point a partition of a hive table to a new directory, what happens to the data?**

The data stays in the old location. It has to be moved manually.

**Write a query to insert a new column(new\_col INT) into a hiev table (htab) at a position before an existing column (x\_col)**

ALTER TABLE table\_name

CHANGE COLUMN new\_col INT

BEFORE x\_col

**Does the archiving of Hive tables give any space saving in HDFS?**

No. It only reduces the number of files which becomes easier for namenode to manage.

**How can you stop a partition form being queried?**

By using the ENABLE OFFLINE clause with ALTER TABLE atatement.

**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 ?**

By Omitting the LOCAL CLAUSE in the LOAD DATA statement.

**If you omit the OVERWRITE clause while creating a hive table, what happens to file which are new and files which already exist?**

The new incoming files are just added to the target directory and the existing files are simply overwritten. Other files whose name does not match any of the incoming files will continue to exist.

If you add the OVERWRITE clause then all the existing data in the directory will be deleted before new data is written.

**What does the following query do?**

INSERT OVERWRITE TABLE employees

PARTITION (country, state)

SELECT ..., se.cnty, se.st

FROM staged\_employees se;

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

**What is a Table generating Function on hive?**

A table generating function is a function which takes a single column as argument and expands it to multiple column or rows. Example explode()

**How can Hive avoid mapreduce?**

If we set the property hive.exec.mode.local.auto to true then hive will avoid mapreduce to fetch query results.

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

The LIKE operator behaves the same way as the regular SQL operators used in select queries. Example −

street\_name like ‘%Chi’

But the RLIKE operator uses more advance regular expressions which are available in java

Example − street\_name RLIKE ‘.\*(Chi|Oho).\*’ which will select any word which has either chi or oho in it.

**Is it possible to create Cartesian join between 2 tables, using Hive?**

No. As this kind of Join can not be implemented in mapreduce

**As part of Optimizing the queries in HIve, what should be the order of table size in a join query?**

In a join query the smallest table to be taken in the first position and largest table should be taken in the last position.

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

It controls how the map output is reduced among the reducers. It is useful in case of streaming data

**How will you convert the string ’51.2’ to a float value in the price column?**

Select cast(price as FLOAT)

**What will be the result when you do cast(‘abc’ as INT)?**

Hive will return NULL

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

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

**Can we LOAD data into a view?**

No. A view can not be the target of a INSERT or LOAD statement.

**What types of costs are associated in creating index on hive tables?**

Indexes occupies space and there is a processing cost in arranging the values of the column on which index is created.

Give the command to see the indexes on a table.

SHOW INDEX ON table\_name

This will list all the indexes created on any of the columns in the table table\_name.

**What does /\*streamtable(table\_name)\*/ do?**

It is query hint to stream a table into memory before running the query. It is a query optimization Technique.

**Can a partition be archived? What are the advantages and Disadvantages?**

Yes. A partition can be archived. Advantage is it decreases the number of files stored in namenode and the archived file can be queried using hive. The disadvantage is it will cause less efficient query and does not offer any space savings.

**What is a generic UDF in hive?**

It is a UDF which is created using a java program to server some specific need not covered under the existing functions in Hive. It can detect the type of input argument programmatically and provide appropriate response.

**The following statement failed to execute. What can be the cause?**

LOAD DATA LOCAL INPATH ‘${env:HOME}/country/state/’

OVERWRITE INTO TABLE address;

The local inpath should contain a file and not a directory. The $env:HOME is a valid variable available in the hive environment.

**How do you specify the table creator name when creating a table in Hive?**

The TBLPROPERTIES clause is used to add the creator name while creating a table.

The TBLPROPERTIES is added like −

TBLPROPERTIES(‘creator’= ‘Joan’)

**3. What is the limitation of Derby database for Hive metastore?**

With derby database, you cannot have multiple connections or multiple sessions instantiated at the same time. Derby database runs in the local mode and it creates a log file so that multiple users cannot access Hive simultaneously.

**5. What are the complex data types in Hive?**

* Map – The**Map**contains a key-value pair where you can search for a value using the key.
* Struct – A**Struct**is a collection of elements of different data types. For example, if you take the address, it can have different data types. For example, pin code will be in Integer format.
* Array – An**Array**will have a collection of homogeneous elements. For example, if you take your skillset, you can have N number of skills
* Uniontype – It represents a column which can have a value that can belong to any of the data types of your choice.

**6. How does partitioning help in the faster execution of queries?**

With the help of partitioning, a subdirectory will be created with the name of the partitioned column and when you perform a query using the WHERE clause, only the particular sub-directory will be scanned instead of scanning the whole table. This gives you faster execution of queries.

**10. Which method has to be overridden when we use custom UDF in Hive?**

Whenever you write a custom UDF in Hive, you have to extend the UDF class and you have to override the **evaluate()**function.

**13. What is RegexSerDe?**

Regex stands for a regular expression. Whenever you want to have a kind of pattern matching, based on the pattern matching, you have to store the fields. RegexSerDe is present in **org.apache.hadoop.hive.contrib.serde2.RegexSerDe.**

In the SerDeproperties, you have to define your input pattern and output fields. For example, you have to get the column values from line **xyz/pq@def** if you want to take xyz, pq and def separately.

To extract the pattern, you can use:  
‘input.regex’ = ‘(.\*)/(.\*)@(.\*)’  
To specify how to store them, you can use  
‘output.format.string’ = ‘%1$s%2$s%3$s’;

**14. How is ORC file format optimised for data storage and analysis?**

ORC stores collections of rows in one file and within the collection the row data will be stored in a columnar format. With columnar format, it is very easy to compress, thus reducing a lot of storage cost.While querying also, it queries the particular column instead of querying the whole row as the records are stored in columnar format.

ORC has got indexing on every block based on the statistics min, max, sum, count on columns so when you query, it will skip the blocks based on the indexing.

**15. How to access HBase tables from Hive?**

Using Hive-HBase storage handler, you can access the HBase tables from Hive and once you are connected, you can query HBase using the SQL queries from Hive. You can also join multiple tables in HBase from Hive and retrieve the result.

**Can you list few commonly used Hive services?**

* Command Line Interface (cli)
* Hive Web Interface (hwi)
* HiveServer (hiveserver)
* Printing the contents of an RC file using the tool rcfilecat.
* Jar
* Metastore

**6) Write a query to rename a table Student to Student\_New.**

Alter Table Student RENAME to Student\_New

**10) When executing Hive queries in different directories, why is metastore\_db created in all places from where Hive is launched?**

When running Hive in embedded mode, it creates a local metastore. When you run the query, it first checks whether a metastore already exists or not. The property javax.jdo.option.ConnectionURL defined in the hive-site.xml has a default value jdbc: derby: databaseName=metastore\_db; create=true.

The value implies that embedded derby will be used as the Hive metastore and the location of the metastore is metastore\_db which will be created only if it does not exist already. The location metastore\_db is a relative location so when you run queries from different directories it gets created at all places from wherever you launch hive. This property can be altered in the hive-site.xml file to an absolute path so that it can be used from that particular location instead of creating multiple metastore\_db subdirectory multiple times.

**How will you read and write HDFS files in Hive?**

i) TextInputFormat- This class is used to read data in plain text file format.

ii) HiveIgnoreKeyTextOutputFormat- This class is used to write data in plain text file format.

iii) SequenceFileInputFormat- This class is used to read data in hadoop SequenceFile format.

iv) SequenceFileOutputFormat- This class is used to write data in hadoop SequenceFile format.

**13) Differentiate between describe and describe extended.**

Describe database/schema- This query displays the name of the database, the root location on the file system and comments if any.

Describe extended database/schema- Gives the details of the database or schema in a detailed manner.

**14) Is it possible to overwrite Hadoop MapReduce configuration in Hive?**

Yes, hadoop MapReduce configuration can be overwritten by changing the hive conf settings file.

**15) I want to see the present working directory in UNIX from hive. Is it possible to run this command from hive?**

Hive allows execution of UNIX commands with the use of exclamatory (!) symbol. Just use the ! Symbol before the command to be executed at the hive prompt. To see the present working directory in UNIX from hive run !pwd at the hive prompt.

**16)  What is the use of explode in Hive?**

Explode in Hive is used to convert complex data types into desired table formats. explode UDTF basically emits all the elements in an array into multiple rows.

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

SORT BY – Data is ordered at each of ‘N’ reducers where the reducers can have overlapping range of data.

ORDER BY- This is similar to the ORDER BY in SQL where total ordering of data takes place by passing it to a single reducer.

DISTRUBUTE BY – It is used to distribute the rows among the reducers. Rows that have the same distribute by columns will go to the same reducer.

CLUSTER BY- It is a combination of DISTRIBUTE BY and SORT BY where each of the N reducers gets non overlapping range of data which is then sorted by those ranges at the respective reducers.

**18) Difference between HBase and Hive.**

* HBase is a NoSQL database whereas Hive is a data warehouse framework to process Hadoop jobs.
* HBase runs on top of HDFS whereas Hive runs on top of Hadoop MapReduce.

**19) Write a hive query to view all the databases whose name begins with “db”**

SHOW DATABASES LIKE ‘db.\*’

**20) How can you prevent a large job from running for a long time?**

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.

**27) How data transfer happens from HDFS to Hive?**

If data is already present in HDFS then the user need not LOAD DATA that moves the files to the /user/hive/warehouse/. So the user just has to define the table using the keyword external that creates the table definition in the hive metastore.

Create external table table\_name (

  id int,

  myfields string

)

location '/my/location/in/hdfs';

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

Hive Architecture consists of a –

* User Interface – UI component of the Hive architecture calls the execute interface to the driver.
* Driver create a session handle to the query and sends the query to the compiler to generate an execution plan for it.
* Metastore - Sends the metadata to the compiler for the execution of the query on receiving the sendMetaData request.
* Compiler- Compiler generates the execution plan which is a DAG of stages where each stage is either a metadata operation, a map or reduce job or an operation on HDFS.
* Execute Engine- Execution engine is responsible for submitting each of these stages to the relevant components by managing the dependencies between the various stages in the execution plan generated by the compiler.

**32) What happens on executing the below query? After executing the below query, if you modify   the column –how will the changes be tracked?**

Hive> CREATE INDEX index\_bonuspay ON TABLE employee (bonus)

AS 'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler';

The query creates an index named index\_bonuspay which points to the bonus column in the employee table. Whenever the value of bonus is modified it will be stored using an index value.

**34) Is it possible to compress json in Hive external table ?**

Yes, you need to gzip your files and put them as is (\*.gz) into the table location.

**When to use Hive?**

* Hive is useful when making data warehouse applications
* When you are dealing with static data instead of dynamic data
* When application is on high latency (high response time)
* When a large data set is maintained
* When we are using queries instead of scripting

**3) Mention what are the different modes of Hive?**

Depending on the size of data nodes in Hadoop, Hive can operate in two modes.

These modes are,

* Local mode
* Map reduce mode

**4) Mention when to use Map reduce mode?**

Map reduce mode is used when,

* It will perform on large amount of data sets and query going to execute in a parallel way
* Hadoop has multiple data nodes, and data is distributed across different node we use Hive in this mode
* Processing large data sets with better performance needs to be achieved

**8) Mention what Hive is composed of ?**

Hive consists of 3 main parts,

1. Hive Clients
2. Hive Services
3. Hive Storage and Computing

**9) Mention what are the type of database does Hive support ?**

For single user metadata storage, Hive uses derby database and for multiple user Metadata or shared Metadata case Hive uses MYSQL.

**10) Mention Hive default read and write classes?**

Hive default read and write classes are

1. TextInputFormat/HiveIgnoreKeyTextOutputFormat
2. SequenceFileInputFormat/SequenceFileOutputFormat

**16) Mention what is (HS2) HiveServer2?**

It is a server interface that performs following functions.

* It allows remote clients to execute queries against Hive
* Retrieve the results of mentioned queries

Some advanced features Based on Thrift RPC in its latest version include

* Multi-client concurrency
* Authentication

**17) Mention what Hive query processor does?**

Hive query processor convert graph of MapReduce jobs with the execution time framework.  So that the jobs can be executed in the order of dependencies.

**20) Mention when to choose “Internal Table” and “External Table” in Hive?**

In Hive you can choose internal table,

* If the processing data available in local file system
* If we want Hive to manage the complete lifecycle of data including the deletion

You can choose External table,

* If processing data available in HDFS
* Useful when the files are being used outside of Hive

**22) Mention what are views in Hive?**

In Hive, Views are Similar to tables. They are generated based on the requirements.

* We can save any result set data as a view in Hive
* Usage is similar to as views used in SQL
* All type of DML operations can be performed on a view

**23) Explain how Hive Deserialize and serialize the data?**

Usually, while read/write the data, the user first communicate with inputformat. Then it connects with Record reader to read/write record.  To serialize the data, the data goes to row. Here deserialized custom serde use object inspector to deserialize the data in fields.

**28) Mention what is the difference between order by and sort by in Hive?**

* SORT BY will sort the data within each reducer. You can use any number of reducers for SORT BY operation.
* ORDER BY will sort all of the data together, which has to pass through one reducer. Thus, ORDER BY in hive uses a single reducer

**30) Mention how can you stop a partition form being queried?**

You can stop a partition form being queried by using the ENABLE OFFLINE clause with ALTER TABLE statement.