**Big Data Basics**

We need big data only when our requirement needs

**V – Volume [Big volume around like 100 to 1000 Tera Bites]**

**V – Velocity [Growth Rate of Data]**

**V – Variety [Structured and unstructured]**

We use big data only when the we have huge volume and has a fast growth rate of data and we are receiving different formats of data.

**10 PB clusters.**

**Big Data Challenges:**

* Storage [Need to store large volume of data]
* Computational Efficiency [Need to fetch very fast]
* Data Loss [Shouldn’t be any data loss. So, we need to replicate the data and save it in multiple nodes]
* Cost [Should be cost effective]

**RDBMS vs Big Data**

Scalability Issues with RDBMS when the data is growing and we might need to do some Optimizations, Aggregations [When data is not required at detail level], adding indexes.

We cannot add more hardware resources/computational nodes to improve the performance.

Databases are designed to process only structured format. Database is not a good choice when you have variety of data. [Data in several formats like texts, images, videos, etc.]

Grid Computing will help in multiprocessing the data but it works only for low volume of data.

**Big Data like Hadoop will be a good solution for:**

* Huge volume of data.
* Storage efficiency.
* Data recovery solution.
* Horizontal scalability.
* Cost effective.
* Easy for programmers and non-programmers.

**RDBMS is good at:**

* Giga bytes of data.
* Static Schema.
* Interactive and batch.
* Nonlinear scale.

**Hadoop is good at:**

* Peta Bytes of data.
* Batch.
* Linear Scale.
* Dynamic Schema.

**Hadoop = HDFS – Reliable Shared Storage + MapReduce – Distributed Computation.**

**Hadoop is a framework for distributed processing of large data sets across clusters of commodity computers.**

Can start with 15 node cluster.

**HDFS**

File system is an integral part of every OS.

With the help of File System, we are able to find/navigate data very fast in the storage.

**Functions of File System:**

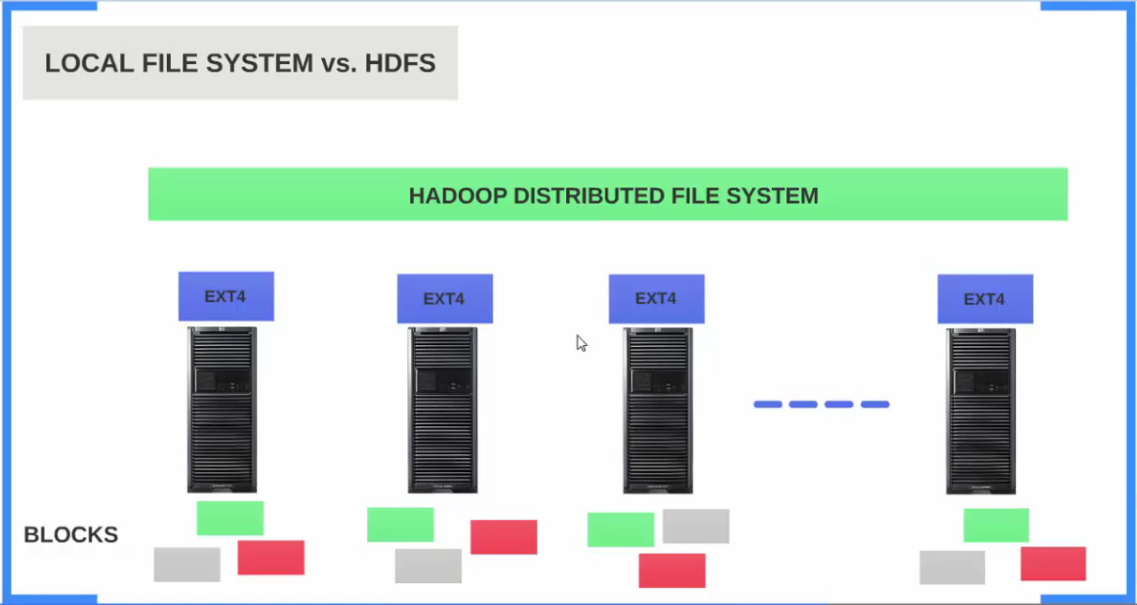
* Control how data is stored and retrieved.
* Metadata about files and folders.
* Permissions and security.
* Manage storage space efficiently.

**Examples for file system are:**

* Windows: FAT32, NTFS.
* IOS: HFS, HFS+
* Linux: ext3, ext4, XFS [Command to find file system in Linux/Unix: df -T]

We already have File System. So, why we need HDFS [Hadoop Distributed File System]?

Local File System[Node] will take care of processing the data in a particular/specific node. To aggregate data coming from different nodes/synchronization between nodes, to be more precise, File system should have a distributed view of your files/blocks in your cluster. So, we need a file system on the top of the nodes. This is where HDFS comes into picture.



It divides 128 MB fixed size blocks.

Default, it stores a block in 3 nodes.

**Benefits of HDFS:**

* Supports distributed processing [blocks (not as whole files)]
* Handle Failures [Replicate blocks across different nodes]
* Scalability [Able to support future expansion]
* Cost Effective [Commodity hardware]

**Commands in HDFS:**

**Local File System Commands:**

ls

mkdir

cp

mv

rm

**HDFS commands:**

**Listing root directory**

hadoop fs -ls /

**Listing default or home directory**

hadoop fs -ls

hadoop fs -ls /user/nithinc2790

**Creating directory in HDFS**

hadoop fs -mkdir test

**Copy from local fs to HDFS**

hadoop fs -copyFromLocal /opt/files/xyz.txt test

**Copy from HDFS to local fs**

hadoop fs -copyToLocal test/xyz.txt .

**Copy from HDFS to HDFS**

hadoop fs -cp test/xyz.txt test1

**Move from HDFS to HDFS**

hadoop fs -mv test1/xyz.txt test2

**Change or Set replication factor**

hadoop fs -Ddfs.replication=2 -cp test2/xyz.txt test1/xyz.txt

**Changing file permissions**

hadoop fs -chmod 777 test/xyz.txt

**File System check command [It requires admin privileges]**

sudo -u hdfs hdfs fack /user/nithinc2790/test1 -files -blocks -locations

**Nodes information is stored in hdfs-site.xml below is the path**

cd /etc/Hadoop/conf

**blocks information is stored by NameNode. NameNode also has the metadata of files and folders.**

**MAP REDUCE**

Distributed programming model for processing large data sets.

Can be implemented in any programming language.

MapReduce is not a programming language.

Hadoop implements MapReduce.

MapReduce System – Manage the communications, data transfers, parallel execution across distributed servers.

**Number of mappers are equal to number of input splits.**

**Number of reducers can be set by user.**

**Shuffle Phase is a place where Mappers output is transferred to reducers input.**

**Benefits of Cloudera Manager**

Fast and easy deployment.

Enhanced cluster monitoring

Alerting capabilities.

Effective cluster maintenance.

**1.Compare Pig and Hive**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Pig** | **Hive** |
| Architecture | Procedural data flow language | SQL type declarative language |
| Application | Programming purposes | Report creation |
| Operational field | Client side | Server side |
| Support for avro files | Yes | No |

**2. What is the definition of Hive? What is the present version of Hive and explain about ACID transactions in Hive?**

Hive is an open source data warehouse system. We can use Hive for analyzing and querying in large data sets of Hadoop files. It’s similar to SQL.

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

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

Hive variable is basically created in the Hive environment that is referenced by Hive scripting languages. It provides to pass some values to the hive queries when the query starts executing. It uses the source command.

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

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

* Analyzing the relatively static data.
* Less Responsive time.
* No rapid changes in data.

Hive doesn’t provide fundamental features required for OLTP, Online Transaction Processing.Hive is suitable for data warehouse applications in large data sets.**Two types of tables in Hive**

1. Managed table.
2. External table.

**5. Can We Change settings within Hive Session? If Yes, How?**

Yes we can change the settings within Hive session, using the SET command. It helps to change Hive job settings for an exact query.  
Example: The following commands shows 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  
And this list will not include defaults of Hadoop. So we should use the below like  
SET -v

**6. Is it possible to add 100 nodes when we have 100 nodes already in Hive? How.**

Yes, we can add the nodes by following the below steps.  
1)Take a new system create a new username and password  
2) Install the SSH and with master node setup ssh connections  
3) Add ssh public\_rsa id key to the authorized keys file  
4) Add the new data node host name, IP address and other details in /etc/hosts slaves file  
192.168.1.102 slave3.in slave3  
5) Start the Data Node on New Node  
6) Login to the new node like suhadoop or ssh -X hadoop@192.168.1.103  
7) Start HDFS of a newly added slave node by using the following command  
./bin/hadoop-daemon.sh start data node  
8) Check the output of jps command on a new node

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

Concatenate function will join the input strings.We can specify the  
‘N’ number of strings separated by a comma.  
Example:

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

Output:

Intellipaat-is-a-eLearning-provider

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

CONCAT\_WS. In this case,we have to specify the set limits of operator first.

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

Output: Intellipaat-is-a-eLearning-provider.

**8. Trim and Reverse function in Hive with examples?**

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 Reverse function, characters are reversed in the string.

Example:

REVERSE(‘INTELLIPAAT’);

Output:

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.  
The syntax is :

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

Example: If we want to change the data type of the salary column from integer to bigint in the employee table.  
ALTER TABLE employee CHANGE salary salary BIGINT;RLIKE: Its full form is Right-Like and it is a special function in the Hive. It helps to examine the two substrings. i.e, if the substring of A matches with B then it evaluates to true.  
Example:

True◊‘Intellipaat’ RLIKE ‘tell’

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

**Bottom of Form**

**Top of Form**

**Bottom of Form**

**10.What are the components used in Hive query processor?**

The components of a Hive query processor include  
• Logical Plan of Generation  
• Physical Plan of Generation  
• Execution Engine  
• Operators  
• UDF’s and UDAF’s  
• Optimizer  
• Parser  
• Semantic Analyzer  
• Type Checking

**11. What is Buckets in Hive?**

The data which is currently present are partitioned and divided further into Buckets.  
The division is performed based on Hash of particular columns that is selected in the table.

**12. Explain process to access sub directories recursively in Hive queries?**

By using below commands we can access sub directories 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 which is like /data/country/state/city/

**13.What are the components used in Hive query processor?**

The components of a Hive query processor include  
• Logical Plan of Generation  
• Physical Plan of Generation  
• Execution Engine  
• Operators  
• UDF’s and UDAF’s  
• Optimizer  
• Parser  
• Semantic Analyzer  
• Type Checking

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

Header records in log files  
System=….  
Version=…  
Sub-version=….  
In the above three lines of headers that we do not want to include in our Hive query. To skip header lines from our tables in the Hive,set a table property that will allow us to skip the header lines.

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”);

**15. The maximum size of string data type supported by hive? Mention the Hive support binary formats?**

The maximum size of string data type supported by hive is 2 GB.  
Hive supports the text file format by default and it supports the binary format Sequence files, ORC files, Avro Data files, Parquet files.  
Sequence files: Splittable, compressible and row oriented are the general binary format.  
ORC files: Full form of ORC is optimized row columnar format files. It is a Record columnar file and column oriented storage file. It divides the table in row split. In each split stores that value of the first row in the first column and followed sub subsequently.  
AVRO data files: It is same as a sequence file splittable, compressible and row oriented, but except the support of schema evolution and multilingual binding support.

**16. What is the precedence order of HIVE configuration?**

We are using a precedence hierarchy for setting the properties  
1. 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

17. If you run a select \* query in Hive, Why does it not run MapReduce?

It’s an optimization technique. Hive, fetch, task conversion property can minimize the latency of map-reduce overhead. When queried like SELECT, FILTER LIMIT queries, this property skips map reduce and using FETCH task. As a result, Hive can execute queries without running MapReduce task.  
By default its value is minimal. Which optimize- SELECT \*, FILTER on partition columns, LIMIT queries only, Where another value is more which optimize- SELECT, FILTER, LIMIT.

18. How Hive can improve performance with ORC format tables?

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

**19. Explain the functionality of Object-Inspector?**

It helps to analyze the internal structure of row object and 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.  
Instance of Java class  
A standard Java object  
A lazily initialized object  
The Object-Inspector tells structure of the object and also ways to access the internal fields inside the object.

**20. Explain the reason for whenever we run hive query, new metastore\_db is created. Why?**

Local metastore is created when we run Hive in embedded mode. And before creating it checks whether the metastore exists or not and this metastore property is defined in the 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 of the location to an absolute path, so that from that location meta-store will be used.

**21. Differentiate between Hive and Hbase**

|  |  |
| --- | --- |
| **Hive** | **HBase** |
| Enables most of the SQL queries | This doesn’t allow SQL queries |
| Doesn’t support record level insert, update, and delete operations on table | It supports |
| It is a data warehouse framework | It is NoSQL database |
| Hive run on the top of MapReduce | HBase runs on the top of HDFS |

**22. Differentiate between Hive and Hbase**

By using below commands we can access sub directories 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 which is like /data/country/state/city/

**23. What are the uses of explode Hive?**

Hadoop developers consider the array as their inputs and convert them into a separate table row. To convert complicate data types into desired table formats Hive is essentially using explode.

**24. What is available mechanism for connecting from applications, when we run hive as a server?**

1. Thrift Client: Using thrift you can call hive commands from various programming languages. Example: C++, PHP,Java, Python and Ruby.  
2. JDBC Driver: JDBC Driver supports the Type 4 (pure Java) JDBC Driver  
3. ODBC Driver: ODBC Driver supports the ODBC protocol.

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

End users want to read their own data format instead of writing, so the user wants to write a Deserializer than SerDe.  
Example: The RegexDeserializer will deserialize the data using the configuration parameter ‘regex’, and a list of column names.  
If our SerDe supports DDL, we probably want to implement a protocol based on DynamicSerDe. It’s non-trivial to write a “thrift DDL” parser.

**26. Mention the date data type in Hive? Name the Hive data type collection.**

The TIMESTAMP data type stores date in java.sql.timestamp format.  
Three collection data types in Hive  
• ARRAY  
• MAP  
• STRUCTGo through this IBM article and learn how [Hive works perfectly with Big SQL](https://www-01.ibm.com/software/data/infosphere/hadoop/hive/)

**27. Can we run UNIX shell commands from Hive? Can Hive queries be executed from script files? How? Give an example.**

Yes, we can run UNIX shell commands from Hive using the! Mark before the command .For example: !pwd at hive prompt will list the current directory.  
We can execute Hive queries from the script files by using the source command.  
Example −

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

**1) What is the difference between Pig and Hive ?**

|  |  |  |
| --- | --- | --- |
| **Pig vs Hive** | | |
| **Criteria** | **Pig** | **Hive** |
| Type of Data | Apache Pig is usually used for semi structured data. | Used for Structured Data |
| Schema | Schema is optional. | Hive requires a well-defined Schema. |
| Language | It is a procedural data flow language. | Follows SQL Dialect and is a declarative language. |
| Purpose | Mainly used for programming. | It is mainly used for reporting. |
| General Usage | Usually used on the client side of the hadoop cluster. | Usually used on the server side of the hadoop cluster. |
| Coding Style | Verbose | More like SQL |

**2) What is the difference between HBase and Hive ?**

|  |  |
| --- | --- |
| Hive vs HBase | |
| **HBase** | **Hive** |
| HBase does not allow execution of SQL queries. | Hive allows execution of most SQL queries. |
| HBase runs on top of HDFS. | Hive runs on top of Hadoop MapReduce. |
| HBase is a NoSQL database. | Hive is a datawarehouse framework. |
| Supports record level insert, updated and delete operations. | Does not support record level insert, update and delete. |

**2) I do not need the index created in the first question anymore. How can I delete the above index named index\_bonuspay?**

DROP INDEX index\_bonuspay ON employee;

**3) 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

**4) Suppose that I want to monitor all the open and aborted transactions in the system along with the transaction id and the transaction state. Can this be achieved using Apache Hive?**

Hive 0.13.0 and above version support SHOW TRANSACTIONS command that helps administrators monitor various hive transactions.

**5) What is the use of Hcatalog?**

Hcatalog can be used to share data structures with external systems. Hcatalog provides access to hive metastore to users of other tools on Hadoop so that they can read and write data to hive’s data warehouse.

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

Alter Table Student RENAME to Student\_New

**7) Where is table data stored in Apache Hive by default?**

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

**8) Explain the difference between partitioning and bucketing.**

* Partitioning and Bucketing of tables is done to improve the query performance. Partitioning helps execute queries faster, only if the partitioning scheme has some common range filtering i.e. either by timestamp ranges, by location, etc. Bucketing does not work by default.
* Partitioning helps eliminate data when used in WHERE clause. Bucketing helps organize data inside the partition into multiple files so that same set of data will always be written in the same bucket. Bucketing helps in joining various columns.
* In partitioning technique, a partition is created for every unique value of the column and there could be a situation where several tiny partitions may have to be created. However, with bucketing, one can limit it to a specific number and the data can then be decomposed in those buckets.
* Basically, a bucket is a file in Hive whereas partition is a directory.

**9) Explain about the different types of partitioning in Hive?**

Partitioning in Hive helps prune the data when executing the queries to speed up processing. Partitions are created when data is inserted into the table. In static partitions, the name of the partition is hardcoded into the insert statement whereas in a dynamic partition, Hive automatically identifies the partition based on the value of the partition field.

Based on how data is loaded into the table, requirements for data and the format in which data is produced at source- static or dynamic partition can be chosen. In dynamic partitions the complete data in the file is read and is partitioned through a MapReduce job based into the tables based on a particular field in the file. Dynamic partitions are usually helpful during ETL flows in the data pipeline.

When loading data from huge files, static partitions are preferred over dynamic partitions as they save time in loading data. The partition is added to the table and then the file is moved into the static partition. The partition column value can be obtained from the file name without having to read the complete file.

**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.

**11) 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.

**12) What are the components of a Hive query processor?**

Query processor in Apache Hive converts the SQL to a graph of MapReduce jobs with the execution time framework so that the jobs can be executed in the order of dependencies. The various components of a query processor are-

* Parser
* Semantic Analyser
* Type Checking
* Logical Plan Generation
* Optimizer
* Physical Plan Generation
* Execution Engine
* Operators
* UDF’s and UDAF’s.

**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.

**21) What is a Hive Metastore?**

Hive Metastore is a central repository that stores metadata in external database.

**22) Are multiline comments supported in Hive?**

No

**23) What is ObjectInspector functionality?**

ObjectInspector is used to analyse the structure of individual columns and the internal structure of the row objects. ObjectInspector in Hive provides access to complex objects which can be stored in multiple formats.

**24) Explain about the different types of join in Hive.**

HiveQL has 4 different types of joins –

JOIN- Similar to Outer Join in SQL

FULL OUTER JOIN – Combines the records of both the left and right outer tables that fulfil the join condition.

LEFT OUTER JOIN- All the rows from the left table are returned even if there are no matches in the right table.

RIGHT OUTER JOIN-All the rows from the right table are returned even if there are no matches in the left table.

**25) How can you configure remote metastore mode in Hive?**

To configure metastore in Hive, hive-site.xml file has to be configured with the below property –

 hive.metastore.uris

   thrift: //node1 (or IP Address):9083

   IP address and port of the metastore host

**26) Is it possible to change the default location of Managed Tables in Hive, if so how?**

Yes, we can change the default location of Managed tables using the LOCATION keyword while creating the managed table. The user has to specify the storage path of the managed table as the value to the LOCATION keyword.

**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';

**28) In case of embedded Hive, can the same metastore be used by multiple users?**

We cannot use metastore in sharing mode. It is suggested to use standalone real database like PostGreSQL and MySQL.

**29)  The partition of hive table has been modified to point to a new directory location. Do I have to move the data to the new location or the data will be moved automatically to the new location?**

Changing the point of partition will not move the data to the new location. It has to be moved manually to the new location from the old one.

**30)  What will be the output of cast (‘XYZ’ as INT)?**

It will return a NULL value.

**31) 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.

**1. Compare HDFS & HBase**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **HDFS** | **HBase** |
| Data write process | Append method | Bulk incremental, random write |
| Data read process | Table scan | Table scan/random read/small range scan |
| Hive SQL querying | Excellent | Average |

**4. What is the use of Hadoop?**

With Hadoop the user can run applications on the systems that have thousands of nodes spreading through innumerable terabytes. Rapid data processing and transfer among nodes helps uninterrupted operation even when a node fails preventing system failure.

**5. What are the operating systems on which Hadoop works?**

Windows and Linux are the preferred operating system though Hadoop can work on OS x and BSD.

**6. What is meant by Big Data?**

Big Data refers to assortment of huge amount of data which is difficult capturing, storing, processing or reprieving. Traditional database management tools cannot handle them but Hadoop can.

**7. Can you indicate Big Data examples?**

Facebook alone generates more than 500 terabytes of data daily whereas many other organizations like Jet Air and Stock Exchange Market generates 1+ terabytes of data every hour.

**8. What are major characteristics of Big Data?**

The three characteristics of Big Data are volume, velocity, and veracity. Earlier it was assessed in megabytes and gigabytes but now the assessment is made in terabytes.

**9. What is the use of Big Data Analysis for an enterprise?**

Analysis of Big Data identifies the problem and focus points in an enterprise. It can prevent big losses and make profits helping the entrepreneurs take informed decision.

**10. What are the characteristics of data scientists?**

Data scientists analyze data and provide solutions for business problems. They are gradually replacing business and data analysts.

**11. What are the basic characteristics of Hadoop?**

Written in Java, Hadoop framework has the capability of solving issues involving Big Data analysis. Its programming model is based on Google MapReduce and infrastructure is based on Google’s Big Data and distributed file systems. Hadoop is scalable and more nodes can be added to it.

**12. Which are the major players on the web that uses Hadoop?**

Introduce in 2002 by Doug Cutting, Hadoop was used in Google MapReduce and HDFS project in 2004 and 2006. Yahoo and Facebook adopted it in 2008 and 2009 respectively. Major commercial enterprises using Hadoop include EMC, Hortonworks, Cloudera, MaOR, Twitter, EBay, and Amazon among others.

**13. How is Hadoop different from traditional RDBMS?**

[RDBMS](https://intellipaat.com/tutorial/sql-tutorial/rdbms/) can be useful for single files and short data whereas Hadoop is useful for handling Big Data in one shot.

**14. What are the main components of Hadoop?**

Main components of Hadoop are HDFS used to store large databases and MapReduce used to analyze them.

**15. What is HDFS?**

HDFS is filing system use to store large data files. It handles streaming data and running clusters on the commodity hardware.

**16. What are the main features of HDFS?**

Great fault tolerance, high throughput, suitability for handling large data sets, and streaming access to file system data are the main features of HDFS. It can be built with commodity hardware.

**17. Why replication is pursued in HDFS though it may cause data redundancy?**

Systems with average configuration are vulnerable to crash at any time. HDFS replicates and stores data at three different locations that makes the system highly fault tolerant. If data at one location becomes corrupt and is inaccessible it can be retrieved from another location.

**18. Would the calculations made on one node be replicated to others in HDFS?**

No! The calculation would be made on the original node only. In case the node fails then only the master node would replicate the calculation on to a second node.

**19. What is meant by streaming access?**

HDFS works on the principle of “*write once, read many”* and the focus is on fast and accurate data retrieval. Steaming access refers to reading the complete data instead of retrieving single record from the database.

**20. What is meant by ‘commodity hardware’? Can  Hadoop work on them?**

Average and non-expensive systems are known as commodity hardware and Hadoop can be installed on any of them. Hadoop does not require high end hardware to function.

**21. Which one is the master node in HDFS? Can it be commodity?**

Name node is the master node in HDFS and job tracker runs on it. The node contains metadata and works as high availability machine and single pint of failure in HDFS. It cannot be commodity as the entire HDFS works on it.

**22. What is meant by Data node?**

Data node is the slave deployed in each of the systems and provides the actual storage locations and serves read and writer requests for clients.

**23. What is daemon?**

Daemon is the process that runs in background in the UNIX environment. In Windows it is ‘*services’* and in DOS it is ‘TSR’.

**24. What is the function of ‘job tracker’?**

Job tracker is one of the daemons that runs on name node and submits and tracks the MapReduce tasks in Hadoop. There is only one job tracker who distributes the task to various task trackers. When it goes down all running jobs comes to a halt.

**25. What is the role played by task trackers?**

Daemons that run on What data nodes, the task tracers take care of individual tasks on slave node as entrusted to them by job tracker.

**26. What is meant by heartbeat in HDFS?**

Data nodes and task trackers send heartbeat signals to Name node and Job tracker respectively to inform that they are alive. If the signal is not received it would indicate problems with the node or task tracker.

**27. Is it necessary that Name node and job tracker should be on the same host?**

No! They can be on different hosts.

**28. What is meant by ‘block’ in HDFS?**

Block in HDFS refers to minimum quantum of data for reading or writing. Default block size is 64 MB in HDFS. If a file is 52 MB then HDFS would store it and leave 12 MB empty and ready to use.

**29. Can blocks be broken down by HDFS if a machine does not have the capacity to copy as many blocks as the user wants?**

Blocks in HDFS cannot be broken. Master node calculates the required space and how data would be transferred to a machine having lower space.

**30. What is the process of indexing in HDFS?**

Once data is stored HDFS will depend on the last part to find out where the next part of data would be stored.

**31. How a data node is identified as saturated?**

When a data node is full and has no space left the name node will identify it.

**32. What type of data is processed by Hadoop?**

Hadoop processes the digital data only.

**33. How Name node determines which data node to write on?**

Name node contains metadata or information in respect of all the data nodes and it will decide which data node to be used for storing data.

**34. Who is the ‘user’ in HDFS?**

Anyone who tries to retrieve data from database using HDFS is the user. Client is not end user but an application that uses job tracker and task tracker to retrieve data.

**35. How the client communicates with Name node and Data node in HDFS?**

The communication mode for clients with name node and data node in HDFS is SSH.

**36. What is a rack in HDFS?**

Rack is the storage location where all the data nodes are put together. Thus it is a physical collection of data nodes stored in a single location.

1. **How will you optimize Hive performance?**

We can improve Hive performance by following few techniques:

* Using Apache Tez engine 🡪 set hive.execution.engine=tez;
* Using ORC file, it has two compress methods one is snappy, Zlib
* Vectorization 🡪 which process rows in batches of 1024 instead of single row each time.
* Cost based query optimization.
* Efficient SQL query.0

1. **Will the reducer work or not if you use “Limit 1” in any HiveQL query?**
2. **Why you should choose Hive instead of Hadoop MapReduce?**