

**Q.**≡

# Hadoop Interview

**A.**≡

## Questions and Answers



## Table of Contents

<u>BASIC HADOOP INTERVIEW QUESTIONS AND ANSWERS</u>	3
<u>HDFS INTERVIEW QUESTIONS AND ANSWERS</u> .....	7
<u>PIG INTERVIEW QUESTIONS AND ANSWERS</u> .....	14
<u>HIVE INTERVIEW QUESTIONS AND ANSWERS</u> .....	20
<u>HBASE INTERVIEW QUESTIONS AND ANSWERS</u> .....	28
<u>SQOOP INTERVIEW QUESTIONS AND ANSWERS</u> .....	33

## Basic Hadoop Interview Questions and Answers

### 1. What is Big Data?

Big data is defined as the voluminous amount of structured, unstructured or semi-structured data that has huge potential for mining but is so large that it cannot be processed using traditional database systems. Big data is characterized by its high velocity, volume and variety that requires cost effective and innovative methods for information processing to draw meaningful business insights. More than the volume of the data – it is the nature of the data that defines whether it is considered as Big Data or not.

### 2. What do the four V's of Big Data denote?

IBM has a nice, simple explanation for the four critical features of big data:

- a) Volume –Scale of data
- b) Velocity –Analysis of streaming data
- c) Variety – Different forms of data
- d) Veracity –Uncertainty of data

### 3. How big data analysis helps businesses increase their revenue? Give example.

Big data analysis is helping businesses differentiate themselves – for example Walmart the world's largest retailer in 2014 in terms of revenue - is using big data analytics to increase its sales through better predictive analytics, providing customized recommendations and launching new products based on customer preferences and needs. Walmart observed a significant 10% to 15% increase in online sales for \$1 billion in incremental revenue. There are many more companies like Facebook, Twitter, LinkedIn, Pandora, JPMorgan Chase, Bank of America, etc. using big data analytics to boost their revenue.

### 4. Name some companies that use Hadoop.

Yahoo (One of the biggest user & more than 80% code contributor to Hadoop)

Facebook

Netflix

Amazon

Adobe

eBay

Hulu

Spotify

Rubikloud

Twitter

## **5. Differentiate between Structured and Unstructured data.**

Data which can be stored in traditional database systems in the form of rows and columns, for example the online purchase transactions can be referred to as Structured Data. Data which can be stored only partially in traditional database systems, for example, data in XML records can be referred to as semi structured data. Unorganized and raw data that cannot be categorized as semi structured or structured data is referred to as unstructured data. Facebook updates, Tweets on Twitter, Reviews, web logs, etc. are all examples of unstructured data.

## **6. On what concept the Hadoop framework works?**

Hadoop Framework works on the following two core components-

1)HDFS – [Hadoop Distributed File System](#) is the java based file system for scalable and reliable storage of large datasets. Data in HDFS is stored in the form of blocks and it operates on the Master Slave Architecture.

2)Hadoop MapReduce-This is a java based programming paradigm of Hadoop framework that provides scalability across various Hadoop clusters. [MapReduce](#) distributes the workload into various tasks that can run in parallel. Hadoop jobs perform 2 separate tasks- job. The map job breaks down the data sets into key-value pairs or tuples. The reduce job then takes the output of the map job and combines the data tuples to into smaller set of tuples. The reduce job is always performed after the map job is executed.

## **7) What are the main components of a Hadoop Application?**

Hadoop applications have wide range of technologies that provide great advantage in solving complex business problems.

Core components of a Hadoop application are-

1) Hadoop Common

2) HDFS

3) Hadoop MapReduce

4) YARN

Data Access Components are - Pig and Hive

Data Storage Component is - HBase

Data Integration Components are - Apache Flume, Sqoop, Chukwa

Data Management and Monitoring Components are - Ambari, Oozie and Zookeeper.

Data Serialization Components are - Thrift and Avro

Data Intelligence Components are - Apache Mahout and Drill.

## **8. What is Hadoop streaming?**

Hadoop distribution has a generic application programming interface for writing Map and Reduce jobs in any desired programming language like Python, Perl, Ruby, etc. This is referred to as Hadoop Streaming. Users can create and run jobs with any kind of shell scripts or executable as the Mapper or Reducers.

## **9. What is the best hardware configuration to run Hadoop?**

The best configuration for executing Hadoop jobs is dual core machines or dual processors with 4GB or 8GB RAM that use ECC memory. Hadoop highly benefits from using ECC memory though it is not low - end. ECC memory is recommended for running Hadoop because most of the Hadoop users have experienced various checksum errors by using non ECC memory. However, the hardware configuration also depends on the workflow requirements and can change accordingly.

## **10. What are the most commonly defined input formats in Hadoop?**

The most common Input Formats defined in Hadoop are

- Text Input Format- This is the default input format defined in Hadoop.

- Key Value Input Format- This input format is used for plain text files wherein the files are broken down into lines.
- Sequence File Input Format- This input format is used for reading files in sequence.

## HDFS Interview Questions and Answers

### 1. What is a block and block scanner in HDFS?

**Block** - The minimum amount of data that can be read or written is generally referred to as a “block” in HDFS. The default size of a block in HDFS is 64MB.

**Block Scanner** - Block Scanner tracks the list of blocks present on a DataNode and verifies them to find any kind of checksum errors. Block Scanners use a throttling mechanism to reserve disk bandwidth on the datanode.

### 2. Explain the difference between NameNode, Backup Node and Checkpoint

**NameNode.** **NameNode:** NameNode is at the heart of the HDFS file system which manages the metadata i.e. the data of the files is not stored on the NameNode but rather it has the directory tree of all the files present in the HDFS file system on a hadoop cluster. NameNode uses two files for the namespace-

**fsimage file-** It keeps track of the latest checkpoint of the namespace.

**edits file-**It is a log of changes that have been made to the namespace since checkpoint.

#### **Checkpoint Node-**

Checkpoint Node keeps track of the latest checkpoint in a directory that has same structure as that of NameNode’s directory. Checkpoint node creates checkpoints for the namespace at regular intervals by downloading the edits and fsimage file from the NameNode and merging it locally. The new image is then again updated back to the active NameNode.

#### **BackupNode:**

Backup Node also provides check pointing functionality like that of the checkpoint node but it also maintains its up-to-date in-memory copy of the file system namespace that is in sync with the active NameNode.

### 3. What is commodity hardware?

Commodity Hardware refers to inexpensive systems that do not have high availability or high quality. Commodity Hardware consists of RAM because there are specific services that need

to be executed on RAM. Hadoop can be run on any commodity hardware and does not require any super computers or high end hardware configuration to execute jobs.

#### **4. What is the port number for NameNode, Task Tracker and Job Tracker?**

NameNode 50070

Job Tracker 50030

Task Tracker 50060

#### **5. Explain about the process of inter cluster data copying.**

HDFS provides a distributed data copying facility through the DistCP from source to destination. If this data copying is within the hadoop cluster then it is referred to as inter cluster data copying. DistCP requires both source and destination to have a compatible or same version of hadoop.

#### **6. How can you overwrite the replication factors in HDFS?**

The replication factor in HDFS can be modified or overwritten in 2 ways-

1) Using the Hadoop FS Shell, replication factor can be changed per file basis using the below command-

`$hadoop fs -setrep -w 2 /my/test_file` (test\_file is the filename whose replication factor will be set to 2)

2) Using the Hadoop FS Shell, replication factor of all files under a given directory can be modified using the below command-

3) `$hadoop fs -setrep -w 5 /my/test_dir` (test\_dir is the name of the directory and all the files in this directory will have a replication factor set to 5)

**7. Explain the difference between NAS and HDFS.** NAS runs on a single machine and thus there is no probability of data redundancy whereas HDFS runs on a cluster of different machines thus there is data redundancy because of the replication protocol.



- NAS stores data on a dedicated hardware whereas in HDFS all the data blocks are distributed across local drives of the machines.
- In NAS data is stored independent of the computation and hence Hadoop MapReduce cannot be used for processing whereas HDFS works with Hadoop MapReduce as the computations in HDFS are moved to data.

**8. Explain what happens if during the PUT operation, HDFS block is assigned a replication factor 1 instead of the default value 3.**

Replication factor is a property of HDFS that can be set accordingly for the entire cluster to adjust the number of times the blocks are to be replicated to ensure high data availability. For every block that is stored in HDFS, the cluster will have  $n-1$  duplicated blocks. So, if the replication factor during the PUT operation is set to 1 instead of the default value 3, then it will have a single copy of data. Under these circumstances when the replication factor is set to 1, if the DataNode crashes under any circumstances, then only single copy of the data would be lost.

**9. What is the process to change the files at arbitrary locations in HDFS?**

HDFS does not support modifications at arbitrary offsets in the file or multiple writers but files are written by a single writer in append only format i.e. writes to a file in HDFS are always made at the end of the file.

**10. Explain about the indexing process in HDFS.**

Indexing process in HDFS depends on the block size. HDFS stores the last part of the data that further points to the address where the next part of data chunk is stored.

**11. What is a rack awareness and on what basis is data stored in a rack?**

All the data nodes put together form a storage area i.e. the physical location of the data nodes is referred to as Rack in HDFS. The rack information i.e. the rack id of each data node is acquired by the NameNode. The process of selecting closer data nodes depending on the rack information is known as Rack Awareness.

The contents present in the file are divided into data block as soon as the client is ready to load the file into the hadoop cluster. After consulting with the NameNode, client allocates 3

data nodes for each data block. For each data block, there exists 2 copies in one rack and the third copy is present in another rack. This is generally referred to as the Replica Placement Policy.

### **12. What happens to a NameNode that has no data?**

There does not exist any NameNode without data. If it is a NameNode then it should have some sort of data in it.

### **13. What happens when a user submits a Hadoop job when the NameNode is down- does the job get in to hold or does it fail.**

The Hadoop job fails when the NameNode is down.

### **14. What happens when a user submits a Hadoop job when the Job Tracker is down- does the job get in to hold or does it fail.**

The Hadoop job fails when the Job Tracker is down.

### **15. Whenever a client submits a hadoop job, who receives it?**

NameNode receives the Hadoop job which then looks for the data requested by the client and provides the block information. JobTracker takes care of resource allocation of the hadoop job to ensure timely completion.

### **16) What does the HDFS error “File could only be replicated to 0 nodes, instead of 1” mean?**

This exception occurs when the DataNode is not available to the NameNode (i.e. the client is not able to communicate with the DataNode) due to one of the following reasons -

- In hdfs-site.xml file, if the block size is a negative value.
- If there are any network fluctuations between the DataNode and NameNode, as a result of which the primary DataNode goes down whilst write is in progress.
- Disk of DataNode is full.
- DataNode is eventful and occupied with block reporting and scanning.

### **17) How will you measure HDFS space consumed?**

The two popular utilities or commands to measure HDFS space consumed are `hdfs dfs -du` and `hdfs dfsadmin -report`. HDFS provides reliable storage by copying data to multiple nodes. The number of copies it creates is usually referred to as the replication factor which is greater than one.

- `hdfs dfs -du` –This command shows the space consumed by data without replications.
- `hdfs dfsadmin -report` - This command shows the real disk usage by considering data replication. Therefore, the output of `hdfs dfsadmin -report` will always be greater than the output of `hdfs dfs -du` command.

### **18) Is it a good practice to use HDFS for multiple small files?**

It is not a good practice to use HDFS for multiple small files because NameNode is an expensive high performance system. Occupying the NameNode space with the unnecessary amount of metadata generated for each of the small multiple files is not sensible. If there is a large file with loads of data, then it is always a wise move to use HDFS because it will occupy less space for metadata and provide optimized performance.

### **19) I have a file “Sample” on HDFS. How can I copy this file to the local file system?**

This can be accomplished using the following command -

```
bin/hadoop fs -copyToLocal /hdfs/source/path /localfs/destination/path
```

### **20) What do you understand by Inodes?**

HDFS namespace consists of files and directories. Inodes are used to represent these files and directories on the NameNode. Inodes record various attributes like the namespace quota, disk space quota, permissions, modified time and access time.

### **21) Replication causes data redundancy then why is it still preferred in HDFS?**

As we know that Hadoop works on commodity hardware, so there is an increased probability of getting crashed. Thus to make the entire Hadoop system highly tolerant, replication factor is preferred even though it creates multiple copies of the same data at different locations. Data on HDFS is stored in at least 3 different locations. Whenever one copy of the data is corrupted and the other copy of the data is not available due to some technical glitches then the data can be accessed from the third location without any data loss.

**22) Data is replicated at least thrice on HDFS. Does it imply that any alterations or calculations done on one copy of the data will be reflected in the other two copies also?**

Calculations or any transformations are performed on the original data and do not get reflected to all the copies of data. Master node identifies where the original data is located and performs the calculations. Only if the node is not responding or data is corrupted then it will perform the desired calculations on the second replica.

**23) How will you compare two HDFS files?**

UNIX has a diff command to compare two HDFS files but there is no diff command with Hadoop. However, redirections can be used in the shell with the diff command as follows-

```
diff < (hadoop fs -cat /path/to/file) < (hadoop fs -cat /path/to/file2)
```

If the goal is just to find whether the two files are similar or not without having to know the exact differences, then a checksum-based approach can also be followed to compare two files. Get the checksums for both files and compare them.

**24) How will you copy a huge file of size 80GB into HDFS parallelly?**

Using the distCP tools huge files can be copied within or in between various Hadoop clusters.

**25) Are Job Tracker and Task Tracker present on the same machine?**

No, they are present on separate machines as Job Tracker is a single point of failure in Hadoop MapReduce and if the Job Tracker goes down all the running Hadoop jobs will halt.

**26) Can you create multiple files in HDFS with varying block sizes?**

Yes, it is possible to create multiple files in HDFS with different block sizes using an API. The block size can be specified during the time of file creation. Below is the signature of the method that helps achieve this –

```
public FSDataOutputStream (Path f, boolean overwrite, int buffersize, short replication, long blocksize) throws IO Exception
```

**27) What happens if two clients try writing into the same HDFS file?**

HDFS provides support only for exclusive writes so when one client is already writing the file, the other client cannot open the file in write mode. When the client requests the NameNode to open the file for writing, NameNode provides lease to the client for writing to the file. So, if another client requests for lease on the same it will be rejected.

**28) What do you understand by Active and Passive NameNodes?**

The NameNode that works and runs in the Hadoop cluster is often referred to as the Active NameNode. Passive NameNode also known as Standby NameNode is the similar to an active NameNode but it comes into action only when the active NameNode fails. Whenever the active NameNode fails, the passive NameNode or the standby NameNode replaces the active NameNode, to ensure that the Hadoop cluster is never without a NameNode.

**29) How will you balance the disk space usage on a HDFS cluster?**

Balancer tool helps achieve this by taking a threshold value as input parameter which is always a fraction between 0 and 1. The HDFS cluster is said to be balanced, if, for every DataNode, the ratio of used space at the node to total capacity of the node differs from the ratio of used space in the cluster to total capacity of the cluster - is not greater than the threshold value.

**30) If a DataNode is marked as decommissioned, can it be chosen for replica placement?**

Whenever a DataNode is marked as decommissioned it cannot be considered for replication but it continues to serve read request until the node enters the decommissioned state completely i.e. till all the blocks on the decommissioning DataNode are replicated.

**31) How will you reduce the size of large cluster by removing a few nodes?**

A set of existing nodes can be removed using the decommissioning feature to reduce the size of a large cluster. The nodes that have to be removed should be added to the exclude file. The name of the exclude file should be stated as a config parameter dfs.hosts.exclude. By editing the exclude files or the configuration file, the decommissioning process can be ended.

**32) What do you understand by Safe Mode in Hadoop?**

The state in which NameNode does not perform replication or deletion of blocks is referred to as Safe Mode in Hadoop. In safe mode, NameNode only collects block reports information from the DataNodes.

**33) How will you manually enter and leave Safe Mode in Hadoop?**

Below command is used to enter Safe Mode manually –

```
$ Hdfs dfsadmin -safe mode enter
```

Once the safe mode is entered manually, it should be removed manually.

Below command is used to leave Safe Mode manually –

```
$ hdfs dfsadmin -safe mode leave
```

### **34) What are the advantages of a block transfer?**

The size of a file can be larger than the size of a single disk within the network. Blocks from a single file need not be stored on the same disk and can make use of different disks present in the Hadoop cluster. This simplifies the entire storage subsystem providing fault tolerance and high availability.

### **35) How will you empty the trash in HDFS?**

Just like many desktop operating systems handle deleted files without a key, HDFS also moves all the deleted files into trash folder stored at /user/hdfs/.Trash. The trash can be emptied by running the following command-

```
hdfs -dfs expunge
```

## **Pig Interview Questions and Answers**

### **1) What do you mean by a bag in Pig?**

Collection of tuples is referred as a bag in Apache Pig

### **2) Does Pig support multi-line commands?**

Yes

### **3) Explain the need for MapReduce while programming in Apache Pig.**

Apache Pig programs are written in a query language known as Pig Latin that is similar to the SQL query language. To execute the query, there is a need for an execution engine. The Pig

engine converts the queries into MapReduce jobs and thus MapReduce acts as the execution engine and is needed to run the programs.

#### **4) Explain about the BloomMapFile.**

BloomMapFile is a class, that extends the MapFile class. It is used in HBase table format to provide quick membership test for the keys using dynamic bloom filters.

#### **5) Differentiate between Hadoop MapReduce and Pig**

- Pig provides higher level of abstraction whereas MapReduce provides low level of abstraction.
- MapReduce requires the developers to write more lines of code when compared to Apache Pig.
- Pig coding approach is comparatively slower than the fully tuned MapReduce coding approach.

Read More in Detail- <http://www.dezyre.com/article/-mapreduce-vs-pig-vs-hive/163>

#### **6) What is the usage of foreach operation in Pig scripts?**

FOREACH operation in Apache Pig is used to apply transformation to each element in the data bag, so that respective action is performed to generate new data items.

Syntax- FOREACH data\_bagname GENERATE exp1, exp2

#### **7) Explain about the different complex data types in Pig.**

Apache Pig supports 3 complex data types-

- Maps- These are key, value stores joined together using #.
- Tuples- Just similar to the row in a table, where different items are separated by a comma. Tuples can have multiple attributes.
- Bags- Unordered collection of tuples. Bag allows multiple duplicate tuples.

#### **8) What does Flatten do in Pig?**

Sometimes there is data in a tuple or a bag and if we want to remove the level of nesting from that data, then Flatten modifier in Pig can be used. Flatten un-nests bags and tuples. For tuples, the Flatten operator will substitute the fields of a tuple in place of a tuple, whereas un-nesting bags is a little complex because it requires creating new tuples.

### **9) How do users interact with the shell in Apache Pig?**

Using Grunt i.e. Apache Pig's interactive shell, users can interact with HDFS or the local file system. To start Grunt, users should invoke Apache Pig with no command –

Executing the command “pig -x local” will result in the prompt -

```
grunt >
```

This is where PigLatin scripts can be run either in local mode or in cluster mode by setting the configuration in PIG\_CLASSPATH.

To exit from grunt shell, press CTRL+D or just type exit.

### **10) What are the debugging tools used for Apache Pig scripts?**

describe and explain are the important debugging utilities in Apache Pig.

- explain utility is helpful for Hadoop developers, when trying to debug error or optimize PigLatin scripts. explain can be applied on a particular alias in the script or it can be applied to the entire script in the grunt interactive shell. explain utility produces several graphs in text format which can be printed to a file.
- describe debugging utility is helpful to developers when writing Pig scripts as it shows the schema of a relation in the script. For beginners who are trying to learn Apache Pig can use the describe utility to understand how each operator makes alterations to data. A pig script can have multiple describes.

### **11) What is illustrate used for in Apache Pig?**

Executing pig scripts on large data sets, usually takes a long time. To tackle this, developers run pig scripts on sample data but there is possibility that the sample data selected, might not execute your pig script properly. For instance, if the script has a join operator there should be



at least a few records in the sample data that have the same key, otherwise the join operation will not return any results. To tackle these kind of issues, illustrate is used. illustrate takes a sample from the data and whenever it comes across operators like join or filter that remove data, it ensures that only some records pass through and some do not, by making modifications to the records such that they meet the condition. illustrate just shows the output of each stage but does not run any MapReduce task.

## **12) Explain about the execution plans of a Pig Script**

**Or**

### **Differentiate between the logical and physical plan of an Apache Pig script**

Logical and Physical plans are created during the execution of a pig script. Pig scripts are based on interpreter checking. Logical plan is produced after semantic checking and basic parsing and no data processing takes place during the creation of a logical plan. For each line in the Pig script, syntax check is performed for operators and a logical plan is created. Whenever an error is encountered within the script, an exception is thrown and the program execution ends, else for each statement in the script has its own logical plan.

A logical plan contains collection of operators in the script but does not contain the edges between the operators.

After the logical plan is generated, the script execution moves to the physical plan where there is a description about the physical operators, Apache Pig will use, to execute the Pig script. A physical plan is more or less like a series of MapReduce jobs but then the plan does not have any reference on how it will be executed in MapReduce. During the creation of physical plan, cogroup logical operator is converted into 3 physical operators namely –Local Rearrange, Global Rearrange and Package. Load and store functions usually get resolved in the physical plan.

## **13) What do you know about the case sensitivity of Apache Pig?**

It is difficult to say whether Apache Pig is case sensitive or case insensitive. For instance, user defined functions, relations and field names in pig are case sensitive i.e. the function COUNT is not the same as function count or X=load 'foo' is not same as x=load

‘foo’. On the other hand, keywords in Apache Pig are case insensitive i.e. LOAD is same as load.

#### **14) What are some of the Apache Pig use cases you can think of?**

Apache Pig big data tools, is used in particular for iterative processing, research on raw data and for traditional ETL data pipelines. As Pig can operate in circumstances where the schema is not known, inconsistent or incomplete- it is widely used by researchers who want to make use of the data before it is cleaned and loaded into the data warehouse.

To build behaviour prediction models, for instance, it can be used by a website to track the response of the visitors to various types of ads, images, articles, etc.

#### **15) Differentiate between PigLatin and HiveQL**

- It is necessary to specify the schema in HiveQL, whereas it is optional in PigLatin.
- HiveQL is a declarative language, whereas PigLatin is procedural.
- HiveQL follows a flat relational data model, whereas PigLatin has nested relational data model.

Read More about [Pig vs. Hive](#)

#### **16) Is PigLatin a strongly typed language? If yes, then how did you come to the conclusion?**

In a strongly typed language, the user has to declare the type of all variables upfront. In Apache Pig, when you describe the schema of the data, it expects the data to come in the same format you mentioned. However, when the schema is not known, the script will adapt to actually data types at runtime. So, it can be said that PigLatin is strongly typed in most cases but in rare cases it is gently typed, i.e. it continues to work with data that does not live up to its expectations.

#### **17) What do you understand by an inner bag and outer bag in Pig?**

A relation inside a bag is referred to as inner bag and outer bag is just a relation in Pig

#### **18) Differentiate between GROUP and COGROUP operators.**

Both GROUP and COGROUP operators are identical and can work with one or more relations. GROUP operator is generally used to group the data in a single relation for better readability, whereas COGROUP can be used to group the data in 2 or more relations. COGROUP is more like a combination of GROUP and JOIN, i.e., it groups the tables based on a column and then joins them on the grouped columns. It is possible to cogroup up to 127 relations at a time.

**19) Explain the difference between COUNT\_STAR and COUNT functions in Apache Pig?**

COUNT function does not include the NULL value when counting the number of elements in a bag, whereas COUNT\_STAR (0 function includes NULL values while counting.

**20) What are the various diagnostic operators available in Apache Pig?**

1. **Dump Operator-** It is used to display the output of pig Latin statements on the screen, so that developers can debug the code.
2. **Describe Operator-** Explained in apache pig interview question no- 10
3. **Explain Operator-** Explained in apache pig interview question no -10
4. **Illustrate Operator-** Explained in apache pig interview question no -11

**21) How will you merge the contents of two or more relations and divide a single relation into two or more relations?**

This can be accomplished using the UNION and SPLIT operators.

**22) I have a relation R. How can I get the top 10 tuples from the relation R.?**

TOP () function returns the top N tuples from a bag of tuples or a relation. N is passed as a parameter to the function top () along with the column whose values are to be compared and the relation R.

**23) What are the commonalities between Pig and Hive?**

HiveQL and PigLatin both convert the commands into MapReduce jobs.

They cannot be used for OLAP transactions as it is difficult to execute low latency queries.

**24) What are the different types of UDF's in Java supported by Apache Pig?**

Algebraic, Eval and Filter functions are the various types of UDF's supported in Pig.

**25) You have a file employee.txt in the HDFS directory with 100 records. You want to see only the first 10 records from the employee.txt file. How will you do this?**

The first step would be to load the file employee.txt into with the relation name as Employee.

The first 10 records of the employee data can be obtained using the limit operator -

Result= limit employee 10.

## Hive Interview Questions and Answers

**1) 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.

**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 TRABSCTIONS 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.

**DISTRIBUTE 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.

## HBase Interview Questions and Answers

### **1) What do you understand by CAP theorem and which features of CAP theorem does HBase follow?**

CAP stands for Consistency, Availability and Partition Tolerance.

Consistency –At a given point of time, all nodes in a cluster will be able to see the same set of data.

Availability- Every request generates a response, regardless of whether it is a success or a failure.

Partition Tolerance – System continues to work even if there is failure of part of the system or intermittent message loss.

HBase is a column oriented databases providing features like partition tolerance and consistency.

## **2) What is the difference between HBase and HDFS?**

HDFS is a local file system in Hadoop for storing large files but it does not provide tabular form of storage. HDFS is more like a local file system (NTFS or FAT). Data in HDFS is accessed through MapReduce jobs and is well suited for high latency batch processing operations.

HBase is a column oriented database on Hadoop that runs on top of HDFS and stores data in tabular format. HBase is like a database management system that communicates with HDFS to write logical tabular data to physical file system. One can access single rows using HBase from billions of records it has and is well-suited for low latency operations. HBase puts data in indexed StoreFiles present on HDFS for high speed lookups.

## **3) Name few other popular column oriented databases like HBase.**

CouchDB, MongoDB, Cassandra

## **4) What do you understand by Filters in HBase?**

HBase filters enhance the effectiveness of working with large data stored in tables by allowing users to add limiting selectors to a query and eliminate the data that is not required. Filters have access to the complete row to which they are applied. HBase has 18 filters –

TimestampsFilter

PageFilter

MultipleColumnPrefixFilter

FamilyFilter

ColumnPaginationFilter

SingleColumnValueFilter

RowFilter

QualifierFilter

ColumnRangeFilter

ValueFilter

PrefixFilter

SingleColumnValueExcludeFilter

ColumnCountGetFilter

InclusiveStopFilter

DependentColumnFilter

FirstKeyOnlyFilter

KeyOnlyFilter

## **5) Explain about the data model operations in HBase.**

Put Method – To store data in HBase

Get Method – To retrieve data stored in HBase.

Delete Method- To delete the data from HBase tables.

Scan Method –To iterate over the data with larger key ranges or the entire table.

## **6) How will you back up an HBase cluster?**

HBase cluster backups are performed in 2 ways-

Live Cluster Backup

Full Shutdown Backup

In live cluster backup strategy, copy table utility is used to copy the data from one table to another on the same cluster or another cluster. Export utility can also be used to dump the contents of the table onto HDFS on the same cluster.

In full shutdown backup approach, a periodic complete shutdown of the HBase cluster is performed so that the Master and Region Servers go down and if there are hardly any chances of losing out the in-flight changes happening to metadata or StoreFiles. However, this kind of approach can be used only for back-end analytic capacity and not for applications that serve front end webpages.

## **7) Does HBase support SQL like syntax?**

SQL like support for HBase is not yet available. With the use of Apache Phoenix, user can retrieve data from HBase through SQL queries.

**8) Is it possible to iterate through the rows of HBase table in reverse order?**

No.

Column values are put on disk and the length of the value is written first and then the actual value is written. To iterate through these values in reverse order-the bytes of the actual value should be written twice.

**9) Should the region server be located on all DataNodes?**

Yes. Region Servers run on the same servers as DataNodes.

**10) Suppose that your data is stored in collections, for instance some binary data, message data or metadata is all keyed on the same value. Will you use HBase for this?**

Yes, it is ideal to use HBase whenever key based access to data is required for storing and retrieving.

**11) Assume that an HBase table Student is disabled. Can you tell me how will I access the student table using Scan command once it is disabled?**

Any HBase table that is disabled cannot be accessed using Scan command.

**12) What do you understand by compaction?**

During periods of heavy incoming writes, it is not possible to achieve optimal performance by having one file per store. Thus, HBase combines all these HFiles to reduce the number of disk seeks for every read. This process is referred to as Compaction in HBase.

**13) Explain about the various table design approaches in HBase.**

Tall-Narrow and Flat-Wide are the two HBase table design approaches that can be used. However, which approach should be used when merely depends on what you want to achieve and how you want to use the data. The performance of HBase completely depends on the RowKey and hence on directly on how data is accessed.

On a high level the major difference between flat-wide and tall-narrow approach is similar to the difference between get and scan. Full scans are costly in HBase because of ordered RowKey storage policy. Tall-narrow approach can be used when there is a complex RowKey so that focused scans can be performed on logical group of entries.

Ideally, tall-narrow approach is used when there are less number of rows and large number of columns whereas flat-wide approach is used when there are less number of columns and large number of rows.

**14) Which one would you recommend for HBase table design approach – tall-narrow or flat wide?**

There are several factors to be considered when deciding between flat-wide (millions of columns and limited keys) and tall-narrow (millions of keys with limited columns), however, a tall-narrow approach is often recommended because of the following reasons –

- Under extreme scenarios, a flat-wide approach might end up with a single row per region, resulting in poor performance and scalability.
- Table scans are often efficient over multiple reads. Considering that only a subset of the row data will be required, tall-narrow table design approach will provide better performance over flat-wide approach.

**15) What is the best practice on deciding the number of column families for HBase table?**

It is ideal not to exceed the number of columns families per HBase table by 15 because every column family in HBase is stored as a single file, so large number of columns families will be required to read and merge multiple files.

**16) How will you implement joins in HBase?**

HBase does not support joins directly but by using MapReduce jobs join queries can be implemented to retrieve data from various HBase tables.

## **HBase Interview Questions for Experienced**

1) How will you design the HBase Schema for Twitter data?

2) You want to fetch data from HBase to create a REST API. Which is the best way to read HBase data using a Spark Job or a Java program?



- 3) Design a HBase table for many to many relationship between two entities, for example employee and department.
- 4) Explain an example that demonstrates good de-normalization in HBase with consistency.
- 5) Should your HBase and MapReduce cluster be the same or they should be run on separate clusters?

## Sqoop Interview Questions and Answers

### 1) What is the default file format to import data using Apache Sqoop?

Sqoop allows data to be imported using two file formats

#### *i. Delimited Text File Format*

This is the default file format to import data using Sqoop. This file format can be explicitly specified using the `--as-textfile` argument to the import command in Sqoop. Passing this as an argument to the command will produce the string based representation of all the records to the output files with the delimited characters between rows and columns.

#### *ii. Sequence File Format*

It is a binary file format where records are stored in custom record-specific data types which are shown as Java classes. Sqoop automatically creates these data types and manifests them as java classes.

**2) Below command is used to specify the connect string that contains hostname to connect MySQL with local host and database name as test\_db –**

**–connect jdbc: mysql: //localhost/test\_db**

**Is the above command the best way to specify the connect string in case I want to use Apache Sqoop with a distributed hadoop cluster?**

When using Sqoop with a distributed Hadoop cluster the URL should not be specified with localhost in the connect string because the connect string will be applied on all the DataNodes with the Hadoop cluster. So, if the literal name localhost is mentioned instead of the IP address or the complete hostname then each node will connect to a different database on their localhosts. It is always suggested to specify the hostname that can be seen by all remote nodes.

**3) I have around 300 tables in a database. I want to import all the tables from the database except the tables named Table298, Table 123, and Table299. How can I do this without having to import the tables one by one?**

This can be accomplished using the import-all-tables import command in Sqoop and by specifying the exclude-tables option with it as follows-

sqoop import-all-tables

--connect --username --password --exclude-tables Table298, Table 123, Table 299

**4) Does Apache Sqoop have a default database?**

Yes, MySQL is the default database.

**5) How can I import large objects (BLOB and CLOB objects) in Apache Sqoop?**

Apache Sqoop import command does not support direct import of BLOB and CLOB large objects. To import large objects, I Sqoop, JDBC based imports have to be used without the direct argument to the import utility.

**6) How can you execute a free form SQL query in Sqoop to import the rows in a sequential manner?**

This can be accomplished using the –m 1 option in the Sqoop import command. It will create only one MapReduce task which will then import rows serially.

**7) How will you list all the columns of a table using Apache Sqoop?**

Unlike sqoop-list-tables and sqoop-list-databases, there is no direct command like sqoop-list-columns to list all the columns. The indirect way of achieving this is to retrieve the columns

of the desired tables and redirect them to a file which can be viewed manually containing the column names of a particular table.

```
Sqoop import --m 1 --connect 'jdbc: sqlserver: //nameofmyserver;  
database=nameofmydatabase; username=DeZyre; password=mypassword' --query "SELECT  
column_name, DATA_TYPE FROM INFORMATION_SCHEMA.Columns WHERE  
table_name='mytableofinterest' AND \${CONDITIONS}" --target-dir  
'mytableofinterest_column_name'
```

### **8) What is the difference between Sqoop and DistCP command in Hadoop?**

Both distCP (Distributed Copy in Hadoop) and Sqoop transfer data in parallel but the only difference is that distCP command can transfer any kind of data from one Hadoop cluster to another whereas Sqoop transfers data between RDBMS and other components in the Hadoop ecosystem like HBase, Hive, HDFS, etc.

### **9) What is Sqoop metastore?**

Sqoop metastore is a shared metadata repository for remote users to define and execute saved jobs created using sqoop job defined in the metastore. The sqoop –site.xml should be configured to connect to the metastore.

### **10) What is the significance of using –split-by clause for running parallel import tasks in Apache Sqoop?**

--Split-by clause is used to specify the columns of the table that are used to generate splits for data imports. This clause specifies the columns that will be used for splitting when importing the data into the Hadoop cluster. —split-by clause helps achieve improved performance through greater parallelism. Apache Sqoop will create splits based on the values present in the columns specified in the –split-by clause of the import command. If the –split-by clause is not specified, then the primary key of the table is used to create the splits while data import. At times the primary key of the table might not have evenly distributed values between the minimum and maximum range. Under such circumstances –split-by clause can be used to specify some other column that has even distribution of data to create splits so that data import is efficient.

### **11) You use –split-by clause but it still does not give optimal performance then how will you improve the performance further.**

Using the `–boundary-query` clause. Generally, sqoop uses the SQL query `select min (), max ()` from to find out the boundary values for creating splits. However, if this query is not optimal then using the `–boundary-query` argument any random query can be written to generate two numeric columns.

**12) During sqoop import, you use the clause `–m` or `–numb-mappers` to specify the number of mappers as 8 so that it can run eight parallel MapReduce tasks, however, sqoop runs only four parallel MapReduce tasks. Why?**

Hadoop MapReduce cluster is configured to run a maximum of 4 parallel MapReduce tasks and the sqoop import can be configured with number of parallel tasks less than or equal to 4 but not more than 4.

**13) You successfully imported a table using Apache Sqoop to HBase but when you query the table it is found that the number of rows is less than expected. What could be the likely reason?**

If the imported records have rows that contain null values for all the columns, then probably those records might have been dropped off during import because HBase does not allow null values in all the columns of a record.

**14) The incoming value from HDFS for a particular column is NULL. How will you load that row into RDBMS in which the columns are defined as NOT NULL?**

Using the `–input-null-string` parameter, a default value can be specified so that the row gets inserted with the default value for the column that it has a NULL value in HDFS.

**15) If the source data gets updated every now and then, how will you synchronise the data in HDFS that is imported by Sqoop?**

Data can be synchronised using incremental parameter with data import .Incremental parameter can be used with one of the two options-

i) **append**-If the table is getting updated continuously with new rows and increasing row id values then incremental import with append option should be used where values of some of the columns are checked (columns to be checked are specified using `–check-column`) and if it discovers any modified value for those columns then only a new row will be inserted.

ii) **lastmodified** – In this kind of incremental import, the source has a date column which is checked for. Any records that have been updated after the last import based on the lastmodified column in the source, the values would be updated.

# Sqoop Interview Questions for Experienced

1) I have 20000 records in a table. I want copy them to two separate files( records equally distributed) into HDFS (using Sqoop).

How do we achieve this, if table does not have primary key or unique key?