**Shaikh, Heenasalim**

**3.6 years**

**LETS PREPARE FOR INTERVIEW**

**Its time to leave sears holding India**

**HADOOP AND MAPREDUCE QUESTIONS**

1. **What are the basic differences between relational database and HDFS?**

|  |  |  |
| --- | --- | --- |
|  | **RDBMS** | **Hadoop** |
| **Data Volume** | It can handle smaller data | It can handle huge amount data |
| **Data Types** | It can process only structured data | It can process structured, unstructured or semi-structured. |
| **Cost** | Licensed and payable software, | Hadoop is an open source framework. |
| **Engine** | OLTP | OLAP |
| **Processing** | Fast ACID transactions | Slow ACID transactions |
|  | Quickly access the row | Cannot quickly access the row |
| **Read/Write Speed** | Fast reading of the data | Fast writing of the data |

1. **Explain “Big Data” and what are five V’s of Big Data?**

* **Volume:**

Big data can store Petabytes/Exabytes of the data.

* **Velocity**:

And this data process with higher speed .

* **Variety:**

And data can be structured, semi-structured and un-structured data.

* **Value:-**

and providing value ie high return of investment to organization.  
**Veracity:-**

### **What is Hadoop and its components.**

Apache Hadoop is framework which is use to load, process and store the huge volume of Structured , nonstructural and semi-structured data with high speed.

Hadoop have 2 Components :-

* Storage unit– HDFS (NameNode, DataNode)
* Processing unit– YARN (ResourceManager, NodeManager)

**4. What is HDFS:-**

Hadoop distributed file system used to store variety of the data.

HDFS have two components:-

**NameNode: -**

* It is masternode which store the metadata eg replication factor and block size, block number etc
* And manages the datanodes

**DataNode:-**

Datanodes stores the actual data.

**5. What is YARN:-**

Yet another resource negotiator

YARN have two components:-

**ResourceManager:-**

* It manages the resources base on need
* allocates task to node managers.

**NodeManager:-**

It is installed program on every datanode to execute task allocated by resource manager.

### **6. List the difference between Hadoop 1 and Hadoop 2.**

### In Hadoop 1.x, “NameNode” is the single point of failure.

### In Hadoop 2.x, we have Active and Passive “NameNodes”. If the active “NameNode” fails, the passive “NameNode” come into picture.

|  |  |  |
| --- | --- | --- |
|  | Hadoop 1.x | Hadoop 2.x |
| Passive  NameNode | NameNode is a Single Point of Failure | Active & Passive NameNode | |
| Processing | MRV1 (Job Tracker & Task Tracker) | MRV2/YARN (ResourceManager & NodeManager) | |
|  | Low scalability | Higher scalability | |

### **7. What are active and passive “NameNodes”?**

Active “NameNode” is the “NameNode” which works/active in the cluster.When the active “NameNode” fails, the passive “NameNode” replaces the active “NameNode”.

### **8. How does NameNode tackle DataNode failures?**

NameNode periodically receives a Heartbeat (signal) from each of the DataNode in the cluster,

If a DataNode does not send a heartbeat message, after a specific period of time it is marked dead.

### **9. What will you do when NameNode is down?**

1. I will use the file system metadata replica (FsImage) to start a new NameNode.
2. Then, will configure the DataNodes and clients so that they will know new NameNode, t is started.

### **10. What is a checkpoint?**

1. “Checkpointing” is a process that combine FsImage, edit log produce new FsImage.
2. Checkpointing is saving the latest system state.
3. Checkpointing is performed by Secondary NameNode.
4. Hence reduces NameNode startup time when it got failed

### **11. How do you define “block” in HDFS? What is the default block size in Hadoop 1 and in Hadoop 2? Can it be changed?**, **difference between an “HDFS Block” and an “Input Split”?**

The “HDFS Block” is the physical division of the data and

“Input Split” is the logical division of the data received from the client.

**15. Name the three modes in which Hadoop can run.**

The three modes in which Hadoop can run are as follows:

1. **Standalone (local) mode:**

the components of Hadoop, such NameNode, DataNode, ResourceManager, and NodeManager, run as a **single Java process in the local filesystem**

1. **Pseudo-distributed mode:**

In this mode, master and the slave nodes services execute **on a single node.**

1. **Fully distributed mode:**

The Hadoop master and slave services **run on separate nodes**, are stated as fully distributed mode.

### **16. What is the purpose of “RecordReader” in Hadoop?**

The “RecordReader” class reads the record from source and converts it into (key, value) pairs which can be used by “Mapper” task.

**17. How do “reducers” communicate with each other?**

“Reducers” do not communicate with each other, run in isolation/runs separately.

**18. What does a “MapReduce Partitioner” do?**

A “MapReduce Partitioner” it passes common keys to the same “reducer” and

### **19. What is a “Combiner”?**

Combines the input from the “mapper” and Sends to the “reducer”. “

### **20. What are the benefits of Apache Pig over MapReduce?**

|  |  |  |
| --- | --- | --- |
| Unit | Pig | Mapreduce |
| Language | High level data processing language | Low level data processing language |
| Coding | Easy to learn | Complex to learn |
| Code length | 10 lines of code | 100 lines of code |
| Engine | Built in functions available for joins, filters, ordering, sorting | We need to write custom programs for  joins, filters, ordering, sorting |
| Read/Write Speed | Joining is faster | Joining is slow. |

### **21. What are the different data types in Pig Latin ?**

### Pig Latin can handle

### Atomic data types :- int, float, long, double,char etc.

### Complex data types:- tuple, bag and map.

### **22. What are the different relational operations in “Pig Latin” you worked with?**

Different relational operators are:

1. for each
2. filter
3. order by
4. group by
5. distinct
6. join
7. limit

### **23. What is a UDF?**

If some functionalities are unavailable in built-in operators, user create functions (UDF) in languages like java, python,ruby.

### **24. What is the default location where “Hive” stores table data?**

### /user/hive/warehouse

Data warehouse location can be set using hive.metastore.warehouse.dir  property in the hive-site.xml.

### **25. What are the core methods of a Reducer?**

### The 3 core methods of a reducer are –

### setup () – This set up parameters like data size, cache, heap size, etc.

### Function Definition- public void setup (context)

### reduce () it is heart of the reducer, which is called once per key.

### Function Definition -public void reduce (Key,Value,context)

### cleanup () - called only once for clearing all the temporary files.

### Function Definition -public void cleanup (context)

### **26. Explain about the partitioning, shuffle and sort phase**

### Shuffle Phase-

### The process of moving the mapper intermediate outputs to the reducer.

### Sort Phase- Sorts of the intermediate keys before passing to the reducer.

### Partitioning Phase-The process of assigning key value pair to respective reducers is known as partitioning.

**27. Explain about co-group in Pig.**

grouping two or more relations.

It can take 127 relations at a time.

**28. What is SerDe in Hive? How can you write your own custom SerDe?**

SerDe is a Serializer DeSerializer.

Hive uses SerDe to read and write data into tables.

**29. What are the stable versions of Hadoop?**

Release 2.7.1 (stable)

Release 2.4.1

Release 1.2.1 (stable)

**30. The number of nodes you have worked with in a cluster.**

600

### **31. What is data voume u dealt in the company recently?**

### 1TB

### **32. Name some companies that use Hadoop**

### Facebook Netflix Amazon eBay Hulu Twitter

### **33. What is commodity hardware?**

### Commodity Hardware is cheaper systems that do not have high scalability or high quality.

### Hadoop can be run on any commodity hardware and does not require super computers.

### **34. What is the port number for NameNode, Task Tracker and Job Tracker?**[**Click here to Tweet**](http://ctt.ec/wf0Uv)

### Job Tracker 50030

### Task Tracker 50060

### NameNode 50070

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

### hadoop fs –setrep –w 5 /user/heena

### 

### **36. What is the default replication factor?**

### 3

### **37. What are different hdfs dfs shell commands to perform copy operation?**

### $ hadoop fs -copyToLocal $ hadoop fs –copyFromLocal $ hadoop fs –put

### $hadoop fs –get

### 38. What different type of schedulers and type of scheduler did you use

### Capacity Scheduler It is designed to run Hadoop applications as a shared, multi-tenant cluster.

### It maximize throughput and utilization of the cluster.

### Fair Scheduler

### assign equal resources to application.

### **39. What is distcp?**

### **T**he program comes with Hadoop for copying bigdata to different Hadoop servers

### no reducers are used here.

### **40. Command to format the NameNode?**

### $ hdfs namenode –format

### **41. What are different daemons in YARN?**

### ResourceManager: Global resource manager.

### NodeManager: One per data node, It manages and monitors resource usage in terms of Memory, CPU.

### ApplicationMaster: One per application, Tasks are started by NodeManager

### **42. What are the site-specific configuration files in Hadoop?**

### conf/core-site.xml

### conf/hdfs-site.xml

### conf/yarn-site.xml

### conf/mapred-site.xml.

### conf/hadoop-env.sh

### conf/yarn-env.sh

### **43. What are the two main components of ResourceManager?**

### Scheduler

### It allocates the resources to various running applications such as memory, CPU, disk etc.

### ApplicationManager It accepts jobs, resource for executing the application specific ApplicationMaster and provides the service for restarting the ApplicationMaster container on failure.

### **44. What is the function of NodeManager?**

### It manages resources,

### Monitores resource usage (cpu, memory, disk, network)

### Reporting the same to the ResourceManager

### **45. What is the function of ApplicationMaster?**

### ApplicationMaster is per application

### Moves appropriate resources from the Scheduler

### Track resource status

### Monitor resource progress.

### **46. What is a partitioner and how the user can control which key will go to which reducer?**

### Partitioner do partitioning of the keys of the intermediate map-outputs.

### It uses hash function for it.

### There are two partitioner program.

### HashPartitioner.

### A custom partitioner is implemented to decide which keys go to which Reducer.

### Method use to create custom partition is :-

### public class SamplePartitioner extends Partitioner {

### @Override

### public int getPartition(Text key, Text value, int numReduceTasks) {

### }}

### **47. What are combiners and its purpose?**

### Combiners are used to

### It is used to aggregate intermediate map output locally.

### Reduce the amount of data that needs to be transferred across to the reducers.

### Hadoop may or may not execute a combiner.

### **48. How a number of partitioners and reducers are related?**

### The total numbers of partitions are the same as the number of reduce tasks for the job.

### **49. What are a Counter and its purpose?**

### The counter is facility for MapReduce applications

### to report its statistics.

### to track job progress in simple manner

### Counter can be of any Enum type.

### **50. Define different types of Counters?**

### Built in Counters:

### Map Reduce Task Counters

### Job Counters

### Custom Java Counters:

### User can perform their own counting operation by specifying their own counters.

### **51. Why Counter values are shared by all map and reduce tasks across the MapReduce framework?**

### Counters are global so shared across the MapReduce framework and aggregated at the end of the job across all the tasks.

### **53. How to set the number of reducers?**

### The number of reduces for the user sets the job:

### Job.setNumReduceTasks(int)

### -D mapreduce.job.reduces

### **52. What is the reducer and its phases?**

### Reducer reduces a set of intermediate values, which has same key .

### The framework calls reduce() for this. Syntax: reduce(WritableComparable, Iterable, Context) Reducer has three primary phases:

### Shuffle

### Sort

### Reduce

### **54. Detail description of the Reducer phases?**

### Shuffle: Providing the key value pair from mapper to reducer.

### Sort: grouping and sorting the reducer inputs on the basis of the same keys.

### The shuffle and sort phases occur simultaneously;

### Reduce: reduce(WritableComparable, Iterable, Context) method is called to

### Adding the values corresponding to same keys from grouped key value pairs. The output of the reduce task is typically written using Context.write(WritableComparable, Writable).

### **55. Can there be no Reducer?**

### Yes, the number of reducer can be zero if no reduction of values is required.

### Eg. select statement

### **-------------------------------------------------------------------------------------------------------------------------**

### **56. What can be optimum value for Reducer?**

### Value of Reducers can be: 0.95

### 1.75 multiplied by ( \* < number of maximum resources per node>)

### Increasing number of reducers

### Increases the framework overhead

### Increases load balancing

### Lowers the cost of failures

### **57. What is IdentityMapper?**

### Identity mapper is default mapper class provided by Hadoop which invoke directly when no mapper specified

### It writes input key,value pair directly to output.

### **58. What is IdentityReducer?**

### In IdentityReducer is default reducer class provided by Hadoop whuch is invoke directly when no reducers specified.It writes all input data into output.

### **59. Explain speculative execution.**

### Speculative execution is a way of dealing with machine’s performance. As there are lots of machines in the cluster, some machines can have low performance, which affects the performance of the whole job.

### Speculative execution in Hadoop can run multiple copies of the same map- reduce task on multiple task tracker nodes.

### **60. What is DistributedCache and its purpose?**

### DistributedCache is a facility provided by the MapReduce framework to cache files (text, archives, jars etc.) needed for applications. It distributes large, read-only files efficiently.

### **61. What is the Job interface in MapReduce framework?**

### Job interface is used to submit mapreduce job for execution by users.

### Some basic parameters are configured for example:

### Job.setNumReduceTasks(int)

### Configuration.set(JobContext.NUM\_MAPS, int)

### Mapper

### Combiner (if any)

### Partitioner

### Reducer

### InputFormat

### OutputFormat implementations

### setMapSpeculativeExecution(boolean))/ setReduceSpeculativeExecution(boolean))

### Maximum number of attempts per task (setMaxMapAttempts(int)/ setMaxReduceAttempts(int)) etc.

### DistributedCache for large amounts of (read-only) data.

### **62. What is the default value of map and reduce max attempts?**

4

### **63. Explain InputFormat?**

### InputFormat describes the input-specification for a MapReduce job.

### The MapReduce framework then splits the input file(s) into logical InputSplits

### which is then to an individual Mapper.

### Default: TextInputFormat

### **64. What is InputSplit and RecordReader?**

### InputSplit specifies logical division of the data. InputSplit is a byte-oriented view .

### Default: FileSplit RecordReader reads pairs from an InputSplit, then processes them and presents record-oriented view.

### **65. Explain the Job OutputFormat?**

### OutputFormat describes details of the output-specification for MapReduce job.

### Default: TextOutputFormat

### **66. How is the option in Hadoop to skip the bad records?**

### SkipBadRecords class.

**69. What is Data Integrity?**

**70. What is Hadoop File System?**

### **71. What are the problems with Hadoop 1.0?**

### NameNode: No Horizontal Scalability and No High Availability

### Job Tracker: Overburdened.

### MRv1: It can only understand Map and Reduce tasks

**72. Explain Hadoop architechture in detail.**

**73. Explain Mapreduce architechture in detail .**

### **HIVE**

### **1. What is Hive?**

### It is open source framework use for querying and analyzing structured big data like SQL

Hive it is not a database but querying language

The present version of hive is 0.13.1.

**2.In which scenario Hive is good fit ?/ What kind of data warehouse application is suitable for Hive?**

### Data warehousing

### OLAP

### For structured query

### Fast response not needed

Not suitable for OLTP, Online Transaction Processing.

**5.Features and Limitations of Hive.**

**Ans. Features of Hive**

### Data warehousing

### OLAP

### For structured query

### Fast response not needed

Not suitable for OLTP, Online Transaction Processing.

**Limitation of Hive**

Can not perform

OLTP

ACID

### High speed queries

**6. Explain about ACID transactions in Hive?**

Hive supports ACID transactions.

ACID is Atomicity, Consistency, Isolation, and Durability, Insert, Delete, and Update options is possible in Hive.

Insert

Delete

Update

### **7.what are the differences between Hive and RDBMS?**

|  |
| --- |
| HIVE: RDBMS: |
| Schema on Read Schema on write |
| Batch processing Real time processing |
| Data stored in HDFS format Data stored in tabular format |
| MapReduce Processing Non-MapReduce Processing |
| No ACID ACID |

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

There are 4 components of Hive architechture

* **User Interface** – interface to the driver. User can send command to hive engine.
* **Compiler**- Compiler converts the query into the object form.
* **Parser** :- It parses the query and creates DAG tree
* **Metastore** – It is used to send the metadata to the compiler.
* **Execute Engine**- Mapreduce program execute for code.
* **Logical Plan of Generation.**
* **Physical Plan of Generation.**
* **Optimizer.**

### **10. What is the purpose of Hive Driver?**

### Hive Driver is used for compiling, optimizing and then executing the Hive query.

### **11. What is the purpose of storing the metadata?**

### People want to read data with a particular schema

### Eg. Manager may be interested to look attendance of the particular emplyoee

### Principle may be interested to look marks of the particular student. So to access particular field again and again and to avoid efforts, we use schema.

### **12.List the various options available with the Hive command/ the different services that can be invoked using the Hive command.**

### Syntax:

### cli

### hiveserver

### hwi

### jar

### metastore

### rcfile

### **14. Can you execute Hadoop dfs Commands from Hive CLI? How?** hive > dfs -ls / ;

### **15. How to give multiline comments in Hive Scripts?**

### Hive does not support multiline comments. We can give single line comments by using – .

### **16. What is the reason for creating a new metastore\_db whenever Hive query is run from a different directory?** In embedded mode, hive checks the metastore exists or not. If metastore not exist then it creates the local metastore.

### **17. When Hive is run in embedded mode, how to share the metastore within multiple users?** For sharing use the standalone database (like MySQL, PostGresQL) .

### **18. How can an application connect to Hive run as a server?**

### Hive can connect with three drivers:-

### Thrift Client: Hive commands can call hive command through programming languages like Java, PHP, Python, Ruby, C++ JDBC Driver: Type 4 (pure Java) JDBC Driver ODBC driver:  ODBC protocol

**19. What are the different types of tables available in HIve?**

There are two types of tabkle:- Managed table and external table.

We can store both data and metadata in managed table and if you drop the managed table, data and metadata get deleted.

In the external table if we drop table, only metadata get deleted.

**21. Explain the External Table features in Hive?**

**22. Explain the Internal Table features in Hive?**

## **23. Difference between "Internal Table" and "External Table"**

Ans. "Internal Table" also known as Managed Table, is the one that is managed by Hive. When you point data in HDFS to such table, the data is moved to Hive default location /user/hive/warehouse/. And, then if such internal table is dropped, the data is deleted along with.

"External table" on the other hand is user managed, and data is not moved to hive default directory after loading i.e, any custom location can be specified. Consecutively, when you drop such table, no data is deleted, only table schema is dropped.

**24. Mention when to choose “Internal Table” and “External Table” in Hive?**

In Hive you can choose internal table,

If data is needed for temporary basis we can use managed table ,

If data is critical then we need to use managed table

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

It does not support ACID transactions ,So it is not suitable for OLTP system.

**26. Can a table be renamed in Hive?**

Alter Table table\_name RENAME TO new\_name

**27. Can we change the data type of a column in a hive table?**

Using REPLACE column option

ALTER TABLE table\_name REPLACE COLUMNS ……

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

Users need to write their own java code for complex data formatting.

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

* Local mode
* Distributed mode
* Pseudo distributed mode

**31. Is there a date data type in Hive?**

Yes. TIMESTAMP data stores date in java.sql.timestamp format

**32. What are collection data types in Hive?**

There are three collection data types in Hive.

* ARRAY
* MAP
* STRUCT

**33. Can we run unix shell commands from hive? Give example.**

Yes, using the ! mark before the command.

example !pwd will list current directory.

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

org.apache.hadoop.mapred.TextInputFormat

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

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

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

If table we are dropping does not exist hive throws error table does not exist, so to avoid warning message we use if exists clause.

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

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

**38. Will the reducer work or not if you use “Limit 1” in any Hive query or If we use the "Limit 1" in any SQL query in Hive, will Reducer work or not.**

Ans. It will work , as hive fetch random record.

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

To convert complex data types into desired table formats Hive is essentially using explode.

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

ALTER TABLE table\_name

CHANGE COLUMN new\_col\_name new\_column\_datatype

BEFORE x\_col

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

No. It only reduces the number of files which become easy for namenode to manage.

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

By using the ENABLE OFFLINE clause with ALTER TABLE satatement.

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

By Omitting the LOCAL CLAUSE in the LOAD DATA statement.

**44. If you omit the OVERWRITE clause while creating a hive table,what happens to file ?**

It will append the data to existing data.

**45. What does the following query do?**

INSERT OVERWRITE TABLE employees

PARTITION (country, state)

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

FROM staged\_employees se;

This will overwrite the data of the two partitions and data is seleted from temporary table.

**47. How can Hive avoid mapreduce?**

If we set the property hive.exec.mode.local.auto = true then hive will avoid mapreduce.

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

The LIKE operator will select data like particular word .

Example −

street\_name like ‘%Pune’

RLIKE will select data based on OR ,AND,true/false conditions

Example − street\_name RLIKE ‘.\*(Pune|Mumbai).\*’

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

No.

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

In a join query the smallest table should be at first position and largest table should at last position.

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

select cast(price as FLOAT)

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

Hive will return NULL

**53. Mention what are views in Hive?**

* We can save any result set data as a view in Hive
* All type of DML operations can be performed on a view

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

No. The name of a view must be unique in one database.

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

No. We cannot perform INSERT or LOAD on view.

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

Indexes need space and processing cost

**57. Give the command to see the indexes on a table.**

SHOW INDEX ON table\_name

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

It decide the map output is distributed among the reducers. It is useful for streaming data.

1. **What does /\*streamtable(table\_name)\*/ do?**
2. **Can a partition be archived? What are the advantages and Disadvantages?**

Yes. A partition can be archived. Advantage is it decreases the number of files stored in namenode and it become easy for namenode to manage.

The disadvantage is less efficient query and no space saving.

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

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

OVERWRITE INTO TABLE address;

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

We can mention:-

TBLPROPERTIES(‘creator’= ‘Joan’)

### **64. Define the difference between Hive and HBase?**

|  |  |  |
| --- | --- | --- |
| **HBase** | | **Hive** |
| low latency | | high latency |
| Allow SQL queries | doesn’t allow SQL queries | |
| Allow ACID transactions | Allow ACID transactions | |
| It is NoSQL database | It is a data warehouse framework | |
| Not Mapreduce | MapReduce jobs | |

### **Is it possible to change the default location of a managed table?**

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

### **What is a partition in Hive? What is partitioning in Hive?**

* Partitioning is grouping similar data based on a column
* A partition is sub-directory in the table directory.

### **Why do we perform partitioning in Hive?**

It provides faster execution, by fetching only **relevant** partitioned data .

For example, suppose we need to accces january month data then we can create partition on monthly basis .

### **68. How to enable dynamic partitioning in Hive?** set hive.exec.dynamic.parition.mode = nonstrict;

Example,  
insert overwrite table emp\_details\_partitioned  
partition(location)  
select \* from emp\_details;

### **What is the difference between static and dynamic partitioning in Hive?**

**Static Partition in Hive**

* In static partitioning, you must specify the partition column value with LOAD statement.
* Static Partition is timesaving than dynamic partition.
* We can alter the partition in the static partition

**Dynamic Partition in Hive**

* To load the data from a non-partitioned table, you can use a dynamic partition to insert partitions individually.
* Dynamic Partition takes more time to load data than static partition.
* Dynamic partition is suitable for large data or to partition many columns
* You can use a dynamic partition on hive external table and managed tables.

### **What is dynamic partitioning and when is it used?**

In dynamic partitioning values for partition columns are known at the runtime, i.e. It is known during loading of the data into a Hive table.

We can dynamic partition in following two cases:

1. Loading data from an existing non-partitioned table to improve the sampling and therefore, decrease the query latency.
2. When we dont know values of the partitions before
3. Finding these partition values manually for huge data sets

### **Scenario:**

**Suppose, I create a table that contains details of all the transactions done by the customers of year 2016:**

**CREATE TABLE transaction\_details (cust\_id INT, amount FLOAT, month STRING, country STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘,’ ;**

**Now, after inserting 50,000 tuples in this table, I want to know the total revenue generated for each month. But, Hive is taking too much time in processing this query.** **How will you solve this problem and list the steps that I will be taking in order to do so?**

We can solve this problem of query latency by partitioning table according to each month.

As we know, we can’t partition an existing non-partitioned table directly. So, we need to perform dynamic partitioning:-

* 1. Create a partitioned table, say partitioned\_transaction:

CREATE TABLE partitioned\_transaction (cust\_id INT, amount FLOAT, country STRING) PARTITIONED BY (month STRING)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘,’ ;

* 1. Enable dynamic partitioning in Hive:

SET hive.exec.dynamic.partition = true;

SET hive.exec.dynamic.partition.mode = nonstrict;

* 1. Transfer the data from the non – partitioned table into the newly created partitioned table:

INSERT OVERWRITE TABLE partitioned\_transaction PARTITION (month) SELECT cust\_id, amount, country, month FROM transaction\_details;

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

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

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

* 1. **What is the default maximum dynamic partition that can be created by a mapper/reducer? How can you change it?**

By default the number of maximum partition is s 100. One can change it by issuing the following command:

SET hive.exec.max.dynamic.partitions.pernode = <value>

### **72.I am inserting data into a table based on partitions dynamically. But, I received an error – FAILED ERROR IN SEMANTIC ANALYSIS: Dynamic partition strict mode requires at least one static partition column.** **How will you remove this error?**

To remove this error one has to execute following commands:

SET hive.exec.dynamic.partition = true;

SET hive.exec.dynamic.partition.mode = nonstrict; by default it is strict

**73. What is bucketing ?**

 In Hive Tables or partition are subdivided into buckets based on the hash function of a column /key of the table to acces the queries more efficiently is know as bucketing.

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

The partitioned data is divided into different Buckets. This data is divided on the basis of Hash key of the particular columns.

### **75. Why do we need buckets?**

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

* A[**map side join**](https://www.edureka.co/blog/map-side-join-vs-join/)can be perform only when join key and partition key is equal.
* But when partition key differs join key we can perform a map side join by bucketing the table using the join key.
* Bucketing makes the sampling process more efficient and therefore, allows us to decrease the query time.

### **76. How Hive distributes the rows into buckets?**

Hive determines the bucket number for a data by using the formula:

hash\_function (bucketing\_column) modulo (num\_of\_buckets). \

**77. In Hive, how can you enable buckets?**

In Hive, you can enable buckets by using the following command,

set.hive.enforce.bucketing=true;

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

The command:  ‘SET hive.enforce.bucketing=true;’ allows to have the correct number of reducer for bucketing a column.

The reducers generated will not equal to the number of buckets.

We can also set the number of reducer equal to the number of buckets by using set mapred.reduce.task = num\_bucket.

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

Yes we can ,

hive> SET hive.enforce.bucketing=true;

### **What kind of applications is supported by Apache Hive?**

Hive supports all those client applications that are written in:

* Java
* PHP
* Python
* C++
* Ruby
* by exposing its Thrift server.

### **Why Hive does not store metadata information in HDFS?**

Hive stores metadata information in the metastore using RDBMS instead of HDFS because

HDFS read/write operations are time consuming processes.

### **What is the difference between local and remote metastore?**

**Local Metastore:**

In local metastore , the metastore service and Hive service runs in same JVM and database runs in a separate JVM which can located in same machine or in remote machine.

**Remote Metastore:**

In the remote metastore , the metastore service and Hive service runs in separate JVM. Processes communicate with the metastore server using Thrift Network APIs.

### **What is the default database provided by Apache Hive for metastore?**

By default, Hive provides an embedded Derby database metastore.

### **Scenario:**

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

The default metastore configuration allows only one Hive session to be opened at a time for accessing the metastore.

Therefore, if multiple clients try to access the metastore at the same time, they will get an error.

One has to use a standalone metastore, i.e. Local or remote metastore configuration in Apache Hive for allowing access to multiple clients concurrently.

###### **Is it possible to create multiple table in hive for same data?**

As hive creates schema and append on top of an existing data file. One can have multiple schema for one data file, schema will be saved in hive’s metastore and data will not be parsed or serialized to disk in given schema. When we will try to retrieve data, schema will be used. For example if we have 5 column (name, job, dob, id, salary) in the data file present in hive metastore then, we can have multiple schema by choosing any number of columns from the above list. (Table with 3 columns or 5 columns or 6 columns).

But while rying, if we specify any column other than above list, will result in NULL values.

###### 86. What is the maximum size of string data type supported by Hive?

Maximum size is 2 GB.

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

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

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

2 David Lawrence dlawrence1@gmail.com Male 101.177.15.130

3 Andy Hall andyhall2@yahoo.com Female 114.123.153.64

4 Samuel Jackson samjackson231@sun.com Male 89.60.227.31

* + 1. Emily Rose rose.emily4@surveymonkey.com Female 119.92.21.19

### **What is indexing and why do we need it?**

We can achieve Hive query optimization using indexing. It speed up the access of a columns in a Hive database as database system does not need to read all rows everytime to find the particular row and it increases speed.

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

SerDe stands for serializer/deserializer.

A SerDe allows us to convert the unstructured bytes into a record that we can process using Hive.

SerDes are implemented using Java. Hive comes with several built-in SerDes.

Hive provides a SerDe for working with CSV files. We can use issuing following commands:

CREATE EXTERNAL TABLE sample

(id int, first\_name string,

last\_name string, email string,

gender string, ip\_address string)

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

STORED AS TEXTFILE LOCATION ‘/temp’;

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

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

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

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

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

Create a temporary table:

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

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

Load the data into temp\_table:

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

Create a table that will store data in SequenceFile format:

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

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

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

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

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

**91.Explain about the different types of join in Hive**.

**Ans.** There are  4 different types of **joins in Hive** –

**INNER JOIN- It returns matching rows from two tables.**

**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.

**FULL OUTER JOIN –** This join combines the records of both the left and right outer tables.

**92.How can you configure remote metastore mode in Hive?**

**Ans.**

Add the below property in hive-site.xml file   
hive.metastore.uris thrift: //IP Address:9083

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

**Ans.**We can create the managed or external table and transfer the data between HDFS and Hive.

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

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

.

**Bottom of Form**

**95.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 |
| 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. |

**96.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.**

Take a new system

create a new username and password.

Install the SSH and with master node setup ssh connections.

Add ssh public\_rsa id key to the authorized keys file.

Add the new data node host name, IP address in /etc/hosts configurations to slave

Login to the new node like using sudo command

Start HDFS and yarn service use command in HADOOP\_HOME.

/bin/dfs.sh

/bin/yarn.sh

Enter jps command to see if all the Hadoop daemons are running properly or not

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

It is variable used to pass values from shell to the hive queries.

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

Using the source command.

**Example −**

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

Hive –f /path/to/file/file\_with\_query.hql

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

It is a file containing list of commands needs to run when the hive CLI/shell starts.

For example setting the strict mode to be true etc.

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

The default record delimiter is − \n

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

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

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

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

SHOW DATABASES LIKE ‘p.\*’

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

For choosing particular database

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

There is no way to delete the DBPROPERTY.

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

This can be done with following query

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

**106. 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 ('Heena','-','is','-','a','-','hadoop',’-’,’developer’);

If string have common delimiter the we can use

CONCAT\_WS ('delimiter',’Heena’,’is’,’a’,’hadoop’,‘developer’); too

**107. Trim and Reverse function in Hive with examples.**

Trim function will delete the white spaces in a string.  
Example:

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’);

**108. 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;

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

AFTER location in create table statement we can specify

LOCATION ‘/user/data’

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

**110. What is the maximum size of string data type supported by hive?**

The maximum size of string data type supported by hive is 2 GB.

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

**1.** Hive-site.XML

1. Hive-default.xml
2. Hadoop-site.xml
3. Hadoop-default.xml

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

The hive.fetch.task.conversion property of Hive lowers the latency of mapreduce and hence while executing queries like SELECT, FILTER, LIMIT, etc., it skips mapreduce function

**113. 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;

**114. Explain the functionality of Object-Inspector.**

* The Object-Inspector tells object structure and also ways to access the internal fields inside the object.
* It helps to analyze the internal structure of row and columns in HIVE.
* It also provides a uniform way to access complex objects like  
  Instance of Java class  
  A standard Java object  
  A lazily initialized object

### **115.How does Object-Inspector help in the faster execution of queries?**

If you have to join two large tables, you can go for reduce side join. But if both the tables have the same number of buckets or same multiples of buckets and also sorted on the same column there is a possibility of SMBMJ in which all the joins take place in the map phase itself by matching the corresponding buckets.

Buckets are basically files that are created inside the HDFS directory.

There are different properties which you need to set for bucket map joins and they are as follows:

* set hive.enforce.sortmergebucketmapjoin = false;
* set hive.auto.convert.sortmerge.join =  false;
* set hive.optimize.bucketmapjoin =  true;
* set hive.optimize.bucketmapjoin.sortedmerge = true;

**116. 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.

**117. Mention when to use Map reduce mode?**

Map reduce mode is used when,

* For large amount of data sets and for parallel query
* Hadoop has multiple data nodes, and data is distributed across different nodes

**118. 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 techni, a partition is created for every uni 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.

**119. 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.

**120. Is it possible to compress json in Hive external table ?**

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

**121.How will you optimize Hive performance**

There are various ways to run Hive ries faster -

* Using vectorization
* Tez Execution Engine – A Hive optimization techni increases the performance of hive ries by using Tez execution engine. It works on the framework of Hadoop Yarn and executes complex-directed acyclic graphs of general data processing tasks.
* ORC File Format – Optimized Row Columnar (ORC) File Format drastically improves ry performance as it stores data in a more optimized way than other formats.
* Hive Partition – By using partitions, the entries in the different columns of the dataset are separated and stored in partitions. Hence, fetching the values requires rying only the required partitions. Thus, it reduces the time taken by a ry to return results.
* Bucketing in Hive – The data in the table are divided into more manageable portions or buckets that reduces the time taken to give results.
* Vectorization– Use of vectorized ry execution to perform scans, aggregations, filters, and joins in batches of 1024 rows at once reduces the time taken to perform operations.
* Cost-Based Optimization– It this process, Hive optimizes the logical as well as physical execution plan of the ry. The results depend on how the order joins, the type of join to performed, the degree of parallelism etc.
* Hive Index – Among the best ways of optimising, it enhances the performance of the ry. The original table is indexed to create a separate index table that acts as a reference.

**122. There is a  Python application that connects to Hive database for extracting data, creating sub tables for data processing, drops temporary tables, etc. 90% of the processing is done through hive queries which are generated from python code and are sent to hive server for execution.Assume that there are 100K rows , would it be faster to fetch 100K rows to python itself into a list of tuples and mimic the join or filter operations hive performs and avoid the executuon of 20-50 ries run against hive or you should look into hive ry optimization technis ? Which one is performance efficient ?**

**123. How Facebook Uses Hadoop,Hive and Hbase ?**

Answer: Facebook data stored on HDFS, everyday millions of photos uploaded into facebook with the help of Hadoop Facebook Messages,Likes and statues updates running on top of Hbase Hive to generate reports for third-party developers and advertisers who need to track the success of their applications or campaigns.

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

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

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

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

**126.,Why you should choose Hive instead of Hadoop MapReduce?**

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

ORC stores collections of rows in one file and within the collection the row data will be stored in a columnar format. With columnar format, it is very easy to compress, thus reducing a lot of storage cost.

While rying also, it ries the particular column instead of rying the whole row as the records are stored in columnar format.

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

### **128. How to access HBase tables from Hive?**

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

Along with the above Hive interview stions, also check our [Hbase tutorial](https://acadgild.com/blog/hbase-tutorial-beginners-guide) . These 2 articles would be helpful to attend a Hadoop interview. Enroll for [Big Data and Hadoop Training](https://acadgild.com/big-data/big-data-development-training-certification) with ACADGILD and become a successful Hadoop developer.

###### **129. What is the Hive configuration precedence order?**

There is a precedence hierarchy to setting properties. In the following list, **lower numbers take precedence over higher numbers**:

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

###### **130. Describe REPEAT function in Hive with example?**

REPEAT function will repeat the input string n times specified in the command.

**Example:** REPEAT(‘Hive’,3);  
**Output:** HiveHiveHive.

###### **131. What are the Binary Storage formats supported in Hive?**

By default Hive supports text file format, however hive also supports below binary formats.

Sence Files, Avro Data files, RCFiles, ORC files, Part files

**Sence files:** General binary format. Splittable, compressible and row oriented. a typical example can be. if we have lots of small file, we may use sence file as a container, where file name can be a key and content could stored as value. it support compression which enables huge gain in performance.

**Avro datafiles:** Same as Sence file splittable, compressible and row oriented except support of schema evolution and multilingual binding support.

**RCFiles:** Record columnar file, it’s a column oriented storage file. it breaks table in row split. in each split stores that value of first row in first column and followed sub subsently.

**ORC Files:** Optimized Record Columnar files

###### 132. is HQL case sensitive?

HQL is not case sensitive.

### **[133. How can a developer utilize Hive?](https://www.onlineinterviewquestions.com/hive-interview-questions/" \l "collapseUnfiled8)**

* static information rather than dynamic information.
* When the application is on high latency (high reaction time)
* Huge data
* When we are utilizing queries instead of scripting

### **134. What is lateral view in Hive?**

A lateral view is used to apply the User Defined Tabular Function (UDTF) to every row of a table. It then joins the output and input rows to form a virtual table. Lateral view works in conjunction with user-defined table generating functions like explode().

### **135. How to create a database in Hive?**

CREATE DATABASE|SCHEMA [IF NOT EXISTS] <database name>

### **136. How to load JSON file in Hive?**

To load a JSON file in Hive, you must first create a JSON table in hive. Then load and display the actual contents of the schema. After that, fetch the data values in the existing JSON hierarchy by using get\_json\_object().

### **137. What is the primary purpose of Hive in the Hadoop architecture?**

Hadoop is used to crunch large amounts of data with the help of an array of tools. Hive is one such tool that is used to query and analyse huge datasets. It is an open source data warehousing system, which exclusively stores data in Hadoop storage.

### **138. What is vertex and vectorization in Hive?**

Vectorization allows Hive to process a group of rows together instead of processing a single row at a time. Every batch is an array of a primitive type. In vectorization, operations of the entire column vector at one go, which improves the instruction pipelines as well as cache usage.

### **139. How to update records in Hive?**

To update records in Hive, use the following syntax-

**UPDATE** <target **table**>

**SET** <**set** clause list>

[ **WHERE** <search condition> ]

**140. What type of Read and Write operations perfor**m in Hive?

Hive provides READ Many WRITE Once.

**141. What are windowing functions in Hive?**

OVER, RANK

### **142. Explain map side join in Hive?**

Map join is a feature in Hive that speeds up queries as it works without reducers.

It enables a table to be loaded into memory to perform a join within a mapper, without using the Map/Reduce step. It loads a smaller table in memory and the joins in the map phase of the Map/Reduce operation.

**TECHNICAL INTERVIEW STIONS:-**

**156. What is Managed table?**

**157. What is external table?**

**158. Tell me difference between managed table or external table**

**159. How to load the data from the hdfs ?**

**160. How to load the data from the local?**

**161. Tell the commnd to rename the tabl column?**

**162. How to add new columns into hive table ?**

**163. How to change the data types of the hive table?**

**164. What is clustering?**

**165. What is portioning and types of the partitioning?**

**166. What is custom partitioning?**

**167. What is dynamic partitioning?**

**168. How to do dynmic patitioning?**

**169. How to add the partition to hive table?**

**alter table gold\_\_sales\_sears\_pos\_tax\_general add partition (transaction\_year\_nbr='1996') location '/gold/transaction/pos/sears/tax\_aud\_s5\_general/transaction\_year\_nbr=1996';**

**hive -e "alter table gold\_\_sales\_sears\_pos\_payment add if not exists partition (transaction\_year\_nbr=$PROCESS\_YEAR);"**

**How to drop the partition to hive table?**

**72. What is Serialization?**

Serialization is the process in which Hive converts objects in different programming languages to suitable formats as a stream of bytes, which can be stored in HDFS and used by Hive. This may be done with structured or unstructured data bytes. using SerDe. It optimises and saves the state of an object to recreate it when the need arises.

**74. What is Hlog?**

Hlog used for recovering

* 1. Send heartbeat(loadinfo) to master
  2. Write requests handle
  3. Read request handle
  4. Flush
  5. Compaction
  6. Region Splits(Manage)

### Hadoop Architechture/processing

### Explain whole mapreducer processing

**75. Explain how Hive Deserialize and serialize the data?**

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

**76. Give examples of the SerDe classes which hive uses to Serialize and Deserilize data ?**

Hive currently use these SerDe classes to serialize and deserialize data:

• MetadataTypedColumnsetSerDe: This SerDe is used to read/write delimited records like CSV, tab-separated control-A separated records (quote is not supported yet.)

• ThriftSerDe: This SerDe is used to read/write thrift serialized objects. The class file for the Thrift object must be loaded first.

• DynamicSerDe: This SerDe also read/write thrift serialized objects, but it understands thrift DDL so the schema of the object can be provided at runtime. Also it supports a lot of different protocols,including TBinaryProtocol, TJSONProtocol, TCTLSeparatedProtocol (which writes data in delimited records).

**77. How do you write your own custom SerDe ? In most cases, users want to write a Deserializer** instead of a SerDe, because users just want to read their own data format instead of writing to it. •For example, the RegexDeserializer will deserialize the data using the configuration parameter ‘regex’, and possibly a list of column names. •If your SerDe supports DDL (basically, SerDe with parameterized columns and column types), you probably want to implement a Protocol based on DynamicSerDe, instead of writing a SerDe from scratch. The reason is that the framework passes DDL to SerDe through “thrift DDL” format, and it’s non-trivial to write a “thrift DDL” parser.

### **78. How is SerDe different from File format in Hive?**

SerDe stands for Serializer and Deserializer. It determines how to encode and decode the field values or the column values from a record that is: how you serialize and deserialize the values of a column

But file format determines how records are stored in key value format or how do you retrieve the records from the table.

### **79. What is RegexSerDe?**

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

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

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

**=============================================================================**

**PIG**

==========================================================================.

**1. What are different modes of execution in Apache Pig?**

Answer: Apache Pig runs in 2 modes-

* + 1. (Local Mode) - In Local mode , we need local host and all the files are stage on local host.
    2. Hadoop MapReduce (Java) - In MapReduce mode, we need the Hadoop cluster and all the files are stage on hadoop cluster.

2. **Does Pig support multi-line commands?**

Answer: Yes same as java /\*\*/

**3. How would you diagnose or do exception handling in the pig?**

For exception handling of pig script, we can use following operators.

**DUMP** displays the results on screen.

**DESCRIBE** displays the schema of a particular relation.

**ILLUSTRATE** displays step by step execution of pig statements.

**EXPLAIN** displays the execution plan for pig latin statements.

**4. What Are The Debugging Tools Used For Apache Pig Scripts?**

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

**5. What is the difference between store and dumps commands?**

Answer: Dump Command output data displayed on the terminal, but it’s not stored anywhere. For store command data is store in local file system or HDFS.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*All The Best\*\*\*\*\*\*\*\*\*\*\*

**6. Explain The Need For Mapreduce While Programming In Apache Pig.?**

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

7. **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.

8**. What Do You Mean By A Bag In Pig?**

Collection of tuples

**9. What Is The Usage Of Foreach Operation In Pig Scripts?**

**Answer :**

FOREACH operation in Apache Pig is used to apply transformation to each element in the data bag, it is used to select column names of relation that user wants.

**Syntax-** FOREACH data\_bagname GENERATE exp1, exp2.

**10. Explain About The Different Complex Data Types In Pig.?**

**Answer :**

**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.

**Bags-** Unordered collection of tuples. Bag allows multiple duplicate tuples.

**11. What Does Flatten Do In Pig?**

Sometimes there is data in a tuple or a bag and if we want to remove nesting from that data, then Flatten modifier in Pig can be used. Flatten un-nests bags and tuples. Eg. For tuples, the Flatten operator will replace tuples with the fields

**12. How Do Users Interact With The Shell In Apache Pig?**

Using Grunt

**13. To start Grunt in local mode use**

“pig –x local”

**14.To start grunt in mapreduce mode use**

pig

`

**15. What Is Illustrate Used For In Apache Pig?**

Executing pig scripts on large data sets, usually takes a long time. To track 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.

**16. Explain About The Execution Plans Of A Pig Script?<br> Or<br> 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.

**17. What Do You Know About The Case Sensitivity Of Apache Pig?**

**Answer :**

It is difficult to say whether Apache Pig is case sensitive or case insensitive. For eg. 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

A=load ‘foo’ is not same as a=load ‘foo’.

keywords in Apache Pig are case insensitive i.e. LOAD is same as load.

**18. What Are Some Of The Apache Pig Use Cases You Can Think Of?**

**Answer :**

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

19**. 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.

* 1. **Is Piglatin A Strongly Typed Language? If Yes, Then How Did You Come To The Conclusion?**

**Answer :**

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.

**21. 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.

**22. Differentiate Between Group And Cogroup Operators.?**

**Answer :**

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 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.

**23. 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.

**24. 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.

**25. I Have A Relation R. How Can I Get The Top 10 Tuples From The Relation R.?**

TOP (10)

**26. 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 ries.

**27. What Are The Different Types Of Udf’s In Java Supported By Apache Pig?**

Algebraic, Eval and Filter functions

**28. 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?**

**In Pig**

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.

In HDFS :-

Hadoop fs -/use/hshaik0/employee.txt | head -5

* 1. **Explain About The Scalar Datatypes In Apache Pig.?**

integer, float, double, long, bytearray and char array

**30. How Do Users Interact With Hdfs In Apache Pig?**

Using the grunt shell.

* 1. **What Is The Use Of Having Filters In Apache Pig?**

Apache Pig has filters records based on a given condition or predicate.

**Example:-**

X= load ‘inputs’ as(name,address)

Y = filter X by symbol matches ‘Mr.\*’;

* 1. **What Is A Udf In Pig?**

If the in-built operators do not provide some functions then programmers can implement those functionalities by writing user defined functions using other programming languages like Java, Python, Ruby, etc. These User Defined Functions (UDF’s) can then be embedded into a Pig Latin Script.

**33. Can You Join Multiple Fields In Apache Pig Scripts?**

Yes, it is possible to join multiple fields in PIG scripts because the join operations takes records from one input and joins them with another input. This can be achieved by specifying the keys for each input and the two rows will be joined when the keys are equal.

**35. How to cobmbine / concatenate Strings?**

CONCAT(“fruits”,”apple”)

**36. How to FILTER the data?**

FILTER grouped\_data\_gen

BY (int)eligible\_store\_record\_count > 0;

**37. How to sorting in Pig?**

ORDER data by c1,c2,c3;

**38. How can we do optimization in Pig?**

1: Adding compression techniques

and adjusting the mappers\*/

set io.compression.codec.lzo.class com.hadoop.compression.lzo.LzoCodec

set pig.tmpfilecompression true

set pig.tmpfilecompression.codec lzo

2:Setting Mappers:-

set mapred.compress.map.output true

set mapred.min.split.size 524288;

SET mapred.max.split.size 134217728

SET pig.maxCombinedSplitSize 4000000

3:Setting reducers

SET mapreduce.map.java.opts: -Xmx3072m

SET mapreduce.reduce.java.opts: -Xmx6144m

/\*

**39. How to generate map in pig?**

HASH\_MAP = FOREACH FILE1 GENERATE TOMAP(key,value);

**40. How to split Strig?**

TIMESTAMP\_NOW = FOREACH FILTER\_HEADER\_FILE GENERATE STRSPLIT($20,' ',2) as timestamp;

YEAR\_MONTH\_DATE = FOREACH SPLITANTS\_OF\_DATE GENERATE STRSPLIT($0,'-',3);

SPLITANTS = FOREACH TIMESTAMP\_NOW GENERATE FLATTEN(TOBAG(timestamp));

**41. How to make string smallcase?**

VENDOR\_CHK\_STR\_GENERATE = FOREACH REPLACE\_BLANKS GENERATE LOWER($1) AS VENDOR\_CHK\_STR;

**42. Give me cogroup Example?**

FINAL\_OUTPUT = COGROUP $LOAD\_TS BY $0,YEAR\_MONTH\_DATE\_DAY.YEAR BY $0

**43. Compare Apache Pig And Sql?**

**Answer :**

Apache Pig differs from SQL in its usage for ETL, lazy evaluation, store data at any given point of time in the pipeline, support for pipeline splits and explicit declaration of execution plans. SQL is oriented around ries which produce a single result. SQL has no in-built mechanism for splitting a data processing stream and applying different operators to each sub-stream.

Apache Pig allows user code to be included at any point in the pipeline whereas if SQL where to be used data needs to be imported to the database first and then the process of cleaning and transformation begins.

TECHNICAL INTERVIEW QUSTIONS

**44. Write a syntax for loading the file ?**

a = load '/home/auto/hshaik0/simple1.txt' USING PigStorage(',') as (id:chararray,name:chararray,adress:chararray,local\_id:chararray);

**45. Explain the joins in pig?**

**46. Which are different types of the storages in pig**

**47. How we access csv files in pig?**

**48. How to define variable in Pig**

%declare load\_ts `date +"%Y-%m-%d %H:%M:%S"`

url\_chk\_str matches '.\* KMART.COM.\*' ? 'Kmart mWeb' :'1'

VENDOR\_CHK\_STR matches '.\*(huawei).\*' ? 'huawei' : VENDOR\_CHK\_STR

**49. How to check data is empty or not?**

IsEmpty() OR IsNull?

**50. Why we use flatten and group ?**

51. What is bag and how it forms?

By using TOBAG(timestamp)

**52. Why we use trim?**

**53. How to do casting in Pig?**

(chararray)(int)(name);

**SQOOP**

**1.What is Sqoop?**

“**SQ**L to Had**oop** and Hadoop to SQL”

Open source to transfer data between Hadoop and relational databases , Sqoop is the best tool

Sqoop uses two main tools. Like:

* [**Sqoop import**](https://data-flair.training/blogs/sqoop-import/) (Copy data from RDBMS to HDFS)
* [**Sqoop export**](https://data-flair.training/blogs/sqoop-export/) (Copy data from HDFS to RDBMS)

**2. Give a basic introduction to Sqoop**  
When it comes to transfer data between relational database servers and Hadoop, you should know that Sqoop is one of the best tools. In order to be more specific, you should use it in importing data from various types of relational databases. It is important for you to note that you can import data from varied types of databases such as MySQL, HDFS, and Hadoop. It is also interesting to note that you have the option to export data from the Hadoop file with the help of Sqoop. This functionality is being provided by the Apache Software Foundation.

It is also important to mention that Sqoop utilizes two main tools. They are in the form of Sqoop export and Sqoop import. With the help of these two tools, you can now extract data information form varied types of databases.

**Related Article:**[Difference between HBase and RDBMS – Hadoop](https://mindmajix.com/hadoop/difference-between-hbase-rdbms)

**3. Mention the best features of Apache Sqoop.**  
**Ans.** Apache Sqoop is a tool in Hadoop ecosystem have several advantages. Like

1. Parallel import/export
2. Connectors for all major RDBMS Databases
3. Import results of SQL ry
4. Incremental Load
5. Full Load
6. Kerberos Security Integration
7. Load data directly into [**Hive**](https://data-flair.training/blogs/apache-hive-tutorial/) / [**HBase**](https://data-flair.training/blogs/hadoop-hbase-tutorial/)
8. Compression
9. Support for Accumulo

**4. How can you control the number of mappers used by the sqoop command?**  
**Ans.**To control the number of mappers executed by a sqoop command we use the parameter –num-mappers.

**5. What is the default extension of the files produced from a sqoop import using the –compress parameter?**  
**Ans.** .gz,tar

**6. What are the basic commands in Apache Sqoop and its uses?**

uses of Apache Sqoop basic commands are:

1. Codegen- It helps to generate code to interact with database records.
2. Create- hive-table- It helps to Import a table definition into a hive
3. Eval- It helps to evaluate SQL statement and display the results
4. Export- It helps to export an HDFS directory into a database table
5. Help- It helps to list the available commands
6. Import- It helps to import a table from a database to HDFS
7. Import-all-tables- It helps to import tables from a database to HDFS
8. List-databases- It helps to list available databases on a server
9. List-tables- It helps to list tables in a database
10. Version- It helps to display the version information

**7. How Sqoop word came? Sqoop is which type of tool and the main use of sqoop?**  
**Ans.** Sqoop word came from **SQ**L+HAD**OOP**=SQOOP.   
Basically, it is a data transfer tool. We use Sqoop to import and export a large amount of data from RDBMS to HDFS and vice versa.

**8. Difference Between Apache Sqoop vs Flume.**

**Ans.** So, let’s discuss all the differences on the basis of features.

**a. Data Flow**  
Apache Sqoop – Basically, [**Sqoop**](http://sqoop.apache.org/) works with any type of relational database system (RDBMS) that has the basic JDBC connectivity. Also, Sqoop can import data from NoSQL databases like MongoDB, Cassandra and along with it. Moreover, it allows data transfer to Apache Hive or HDFS.  
[Apache Flume](https://data-flair.training/blogs/apache-flume-tutorial/)**–**Likewise, Flume works with streaming data sources those are generated continuously in Hadoop environments. Like log files.

**b. Type of Loading**  
Apache Sqoop – Basically,  Sqoop load is not driven by events.  
Apache Flume – Here, data loading is completely event-driven.

**c. When to use**  
Apache Sqoop – However, if the data is being available in Teradata, Oracle, MySQL, PostreSQL or any other JDBC compatible database it is considered an ideal fit.  
Apache Flume – While we move bulk of streaming data from sources likes JMS or spooling directories, it is the best choice.

**d. Link to HDFS**  
Apache Sqoop – Basically, for importing data in Apache Sqoop, HDFS is the destination  
Apache Flume – In Apache Flume, data generally flow to HDFS through channels

**e. Architecture**  
Apache Sqoop – Basically, it has connector based architecture. However, that means the connectors know a great deal in connecting with the various data sources. Also to fetch data correspondingly.  
Apache Flume – However, it has agent-based architecture. Basically, it means code written in Flume is we call agent that may responsible for fetching the data.  
Also, learn complete comparison, follow link [**Apache Sqoop vs Flume- Comparison**](https://data-flair.training/blogs/apache-sqoop-vs-flume/)

**9. What is difference between Sqoop and Flume?**

|  |  |
| --- | --- |
| **Sqoop** | **Flume** |
| Used for importing data from structured data sources like RDBMS. | Used for moving bulk streaming data into HDFS. |
| It has a connector based architecture. | It has a agent based architecture. |
| Data import in sqoop is not event driven. | Data load in flume is event driven |
| HDFS is the destination for importing data. | Data flows into HDFS through one or more channels. |
| **Sqoop vs Flume** | |

### 10. **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.

all the records will be represent as the output files with delimited characters between rows and columns.

#### **ii) Sence 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.

### **11. 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

Then specifying the exclude-tables option with it as follows-

sqoop import-all-tables

--connect –username –password

--exclude-tables Table298, Table 123, Table 299

**12. How can you import large objects like BLOB and CLOB in Sqoop?**  
The direct import function is not supported by Sqoop in case of CLOB and BLOB objects. Hence, if you have to import large purposes, you can use JDBC based imports. This can be done without introducing the direct argument of the import utility.

**13. What is the default database of Apache Sqoop?**  
The default database of Apache Sqoop is MySQL.

**14. Describe the process of executing a free-form SQL query to import rows**  
To achieve a free-form SQL query, you have to use the –m1 option. This would create only one Mapreduce task. This would then import the rows directly.

**15. What is the meaning of Free form import in Sqoop?**  
With the use of Sqoop, one can import the relational database ry. This can be done using column and table name parameters.

**16. Describe the importance of using –compress-codec parameter  
The –compress-codec parameter can be used to get export file of the Sqoop import in the mentioned formats.**

**17. What is the significance of Eval tool?**  
Sqoop Eval would help you to make use of the sample SQL ries. This can be against the database as it can preview the results that are displayed on the console. Interestingly, with the help of the Eval tool, you would be well aware of the fact that the desired data can be imported correctly or not.

18.**Shed light on the advantage of utilizing –password-file rather than –P option**  
The –password-file option is usually used inside the Sqoop script file. On the other hand, the –P option is able to read the standard input along with the column name parameters.

**19. Is the JDBC driver fully capable to connect Sqoop on the databases?**  
The JDBC driver is not capable to connect Sqoop on the databases. This is the reason that Sqoop requires both the connector and JDBC driver.

**20.  What is the meaning of Input Split in Hadoop?**  
Input Split is that kind of a function which is associated with splitting the input files into various chunks. These chunks can also assign each split to a mapper in the ongoing process of data correction.

* 1. **Illustrate the utility of the Help Command in Sqoop**

The help command in Sqoop can be utilized to list the various available commands.

* 1. **Shed light on the service of Codegen command in Sqoop**

The Codegen command is associated with the generation of code so that it can appropriately interact with the database records.

* 1. **Describe the procedure involved in executing an incremental data load in Sqoop**

You should be well aware of the fact that in Sqoop, the process of performing additional data load is to update the uploaded data. This data is often referred to as delta data. In Sqoop, this delta data can be altered with the use of incremental load command. Additionally, it can be said that with the help of Sqoop, the import command can also perform additional load. By loading the data into the hive without overwriting it, its efficiency can be maintained in a significant manner. This is possible only with the help of incremental data load.

It is also essential for you to illustrate the various types of incremental data load. They are as follows:  
Progressive Mode: This variety usually determines the number of new rows. Moreover, it also possesses a value that can best resemble the Append functions.

**Value:** This denotes the maximum amount that is derived from the check column from the previous import operation.

**The Check Column feature:** This function is helpful in specifying the number of columns that should be assessed to determine the number of rows to be imported.

**25. What are the limitations of importing the RDBMS tables into the Hcatalog directly?**  
In order to import the tables into the Hcatalog in a direct manner, you have to make sure that you are using the –Hcatalog database option. However, in this process, you would face a limitation of importing the tables. It is in the form of the fact that this option do not supports a plethora of arguments like –direct, –as-Avro file and -export-dir.

**28. Define Sqoop metastore**  
It is also known as a shared metadata repository with the help of which the local users can execute and define various types of list tables. In order to connect to the metastore, you have to make changes to the Sqoop –site.xml.

**29. Does Sqoop uses the maps reduce function? If it does then shed light on the reasons**  
Apache Sqoop also uses the Map-Reduce function of Hadoop to obtain data from the relational databases. During the process of importing data, Sqoop controls the mappers and their numbers. The mappers who access RDBMS come across denial of service attacks. Hence, it can be said that with the help of Sqoop, big data can be efficiently managed.

30.**Describe the practicality of opting for Sqoop nowadays**  
Apache Sqoop is regarded as an excellent help for those individuals who face challenges in transferring data out of the data warehouse. It is also used for importing data from RDBMS to HDFS. With the help of Sqoop, the users can also import more than one table. Interestingly, with the use of Apache Sqoop, the data selected columns can be easily exported. Furthermore, Sqoop is also compatible with a majority of JDBC databases. Here is the list of stions which would help you to crack the Sqoop interview.

**31. Name a few import control commands. How can Sqoop handle large objects?**

Import control commands are used to import RDBMS data

**Append:** Append data to an existing dataset in HDFS. –append

**Columns:** columns to import from the table. –columns  
<col,col……> • Where: where clause to use during import. —

where The common large objects are Blog and Clob.Suppose the object is less than 16 MB, it is stored inline with the rest of the data. If there are big objects, they are temporarily stored in a subdirectory with the name \_lob. Those data are then materialized in memory for processing. If we set lob limit as ZERO (0) then it is stored in external memory.

**32. How can we import data from particular row or column? What is the destination types allowed in Sqoop import command?**

Sqoop allows to Export and Import the data from the data table based on the where clause. The syntax is

--columns

<col1,col2……> --where

--ry

Example:

sqoop import –connect jdbc:mysql://db.one.com/corp --table INTELLIPAAT\_EMP --where “start\_date> ’2016-07-20’ ”

sqoopeval --connect jdbc:mysql://db.test.com/corp --ry “SELECT \* FROM intellipaat\_emp LIMIT 20”

sqoop import –connect jdbc:mysql://localhost/database --username root --password aaaaa –columns “name,emp\_id,jobtitle”

Sqoop supports data imported into following services:

* HDFS
* Hive
* Hbase
* Hcatalog
* Accumulo

Bottom of Form

**33. Role of JDBC driver in sqoop setup? Is the JDBC driver enough to connect the sqoop to the database?**

Sqoop needs a connector to connect the different relational databases. Almost all Database vendors make a JDBC connector available specific to that Database, Sqoop needs a JDBC driver of the database for interaction.  
No, Sqoop needs JDBC and a connector to connect a database.

**34. How will you update the rows that are already exported? Write sqoop command to show all the databases in MySQL server.**

By using the parameter – update-key we can update existing rows. Comma-separated list of columns is used which unily identifies a row. All of these columns are used in the WHERE clause generated UPDATE ry. All other table columns will be used in the SET part of the ry.  
The command below is used to show all the databases in MySQL server.

$ sqoop list –databases –connect jdbc:mysql://database.test.com/

**35. The purpose of sqoop-merge is:**  
This tool combines 2 datasets where entries in one dataset overwrite entries of an older dataset preserving only the new version of the records between both the data sets.

**36. Explain the saved job process in Sqoop.**

Sqoop allows us to define saved jobs which make this process simple. A saved job records the configuration information required to execute a Sqoop command at a later time. sqoop-job tool describes how to create and work with saved jobs. Job descriptions are saved to a private repository stored in $HOME/.sqoop/.

We can configure Sqoop to instead use a shared metastore, which makes saved jobs offered to multiple users across a shared cluster. Starting the metastore is covered by the section on the sqoop-metastore tool.

**37. How to enter into Mysql prompt, and explain the command parameter indicates?**

The command for entering into Mysql prompt is “mysql –u root –p”  
-u indicatesthe user  
Root indicates username  
-p indicates password.

**38. I am getting connection failure exception during connecting to Mysql through Sqoop, what is the root cause and fix for this error scenario?**

This will happen when there is lack of permissions to access our Mysql database over the network. We can try the below command to confirm the connect to Mysql database from aSqoop client machine.  
$ mysql –host=MySqlnode> –database=test –user= –password=  
We can grant the permissions with below commands.

mysql> GRANT ALL PRIVILEGES ON \*.\* TO ‘%’@’localhost’;

mysql> GRANT ALL PRIVILEGES ON \*.\* TO ‘ ’@’localhost’;

Give your career a big boost by going through our Apache [Sqoop Training Course](https://intellipaat.com/big-data-hadoop-training/#sample-video) now!

**39. I am getting java.lang.IllegalArgumentException: during importing tables from oracle database.what might be the root cause and fix for this error scenario?**

Sqoop commands are case- sensitive of table names and user names.  
By specifying the above two values in UPPER case, it will resolve the issue.  
In case, the source table is created under different user namespace,then table name should be like USERNAME.TABLENAME as shown below  
sqoop import  
–connect jdbc:oracle:thin:@intellipaat.testing.com/INTELLIPAAT  
–username SQOOP  
–password sqoop  
–table COMPANY.EMPLOYEES

**40. How can you list all the columns of a table using Apache sqoop?**

There is no straight way to list all the columns of a table in Apache Sqoop like sqoop-list-columns, so first we should retrieve the columns of the particular table and transform to a file containing the column names of particular table.Syntax is:

Sqoop import –m1 –connect ‘jdbc:sqlserver://servername;database=databasename;

Username-DeZyre;password=mypassword’ –ry “SELECT column\_name,DATA\_TYPE FROM INFORMATION\_SCHEMA columns WHEREtable\_name=’mytableofinterest’ AND \$CONDITIONS” –target-dir ‘mytableofinterest\_column\_name’.

**41. How to create a table in Mysql and how to insert the values into the table ?**

To create a table in mysql using the below command

mysql> create table tablename( col1 datatype, col2 datatype,…………);

Example –

mysql> create table INTELLIPAAT(emp\_idint,emp\_namevarchar(30),emp\_salint);

Insert the values into the table

mysql> insert into table name(value1,value2,value3,………);

Example-

mysql> insert into INTELLIPAAT(1234,’aaa’,20000);

mysql> insert into INTELLIPAAT(1235,’bbb’,10000);

mysql> insert into INTELLIPAAT(1236,’ccc’,15000);

**42. Is sqoop same as to distcp in hadoop?**

No. Because the only distcp import command is same as Sqoop import command and both the commands submit parallel map-only jobs but both command functions are different. Distcp is used to copy any type of files from Local filesystem to HDFS and Sqoop is used for transferring the data records between RDBMS and Hadoop eco- system service.

**43. For each sqoop copying into HDFS how many MapReduce jobs and tasks will be submitted?**

There are 4 jobs that will be submitted to each Sqoop copying into HDFS and no reduce tasks are scheduled.

**44. How can Sqoop be used in Java programs?**

In the Java code Sqoop jar is included in the classpath. The required parameters are created to Sqoop programmatically like for CLI (command line interface). Sqoop.runTool() method also invoked in Java code.

**45. I am having around 500 tables in a database. I want to import all the tables from the database except the tables named Table 498, Table 323, and Table 199. How can we do this without having to import the tables one by one?**

This can be proficient 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 Table498, Table 323, Table 199

**46. Explain the significance of using –split-by clause in Apache Sqoop?**

split-by is a clause, it is used to specify the columns of the table which are helping to generate splits for data imports during importing the data into the Hadoop cluster. This clause specifies the columns and helps to improve the performance via greater parallelism. And also it helps to specify the column that has an even distribution of data to create splits,that data is imported.

command line.

**48. What is the significance of using –compress-codec parameter?**

Answer: To get the out file of a sqoop import in formats other than .gz like .bz2 we use the –compress -code parameter.

**50. What is a disadvantage of using –direct parameter for faster data load by sqoop?**

Answer: The native utilities used by databases to support faster load do not work for binary data formats like Sence File.

**51. Is it possible to do an incremental import using Sqoop?**

Answer: Yes, Sqoop supports two types of incremental imports-

* Append
* Last Modified

To insert only rows Append should be used in import command and for inserting the rows and also updating Last-Modified should be used in the import command.

**52. How can you check all the tables present in a single database using Sqoop?**

Answer: The command to check the list of all tables present in a single database using Sqoop is as follows-

Sqoop list-tables –connect jdbc: mysql: //localhost/user;

**53. What is the standard location or path for Hadoop Sqoop scripts?**

Answer: /usr/bin/Hadoop Sqoop.

**54. How can we import a subset of rows from a table without using the where clause?**

Answer: We can run a filtering ry on the database and save the result to a temporary table in database.

Then use the sqoop import command without using the –where clause.

**57. Tell few import control commands:**

Answer: –Append

–Columns

–Where

These command are most frently used to import RDBMS Data.

**59. How can you see the list of stored jobs in sqoop metastore?**

Answer: sqoop job –list

**70. What type of databases Sqoop can support?**

Answer: MySQL, Oracle, PostgreSQL, IBM, Netezza and Teradata. Every database connects through jdbc driver.

**74. what are the majorly used commands in sqoop?**

Answer: In Sqoop Majorly Import and export command are used. But below commands are also useful sometimes. codegen, eval, import-all-tables, job, list-database, list-tables, merge, metastore.

**75. What is the usefulness of the options file in sqoop.**

Answer: The options file is used in sqoop to specify the command line values in a file and use it in the sqoop commands.

For example the –connect parameter’s value and –user name value scan be stored in a file and used again and again with different sqoop commands.

**76. what are the common delimiters and escape character in sqoop?**

Answer: The default delimiters are a comma(,) for fields, a newline(\n) for records

Escape characters are \b,\n,\r,\t,\”, \\’,\o etc

**77. while loading table from MySQL into HDFS, if we need to copy tables with maximum possible speed, what can you do?**

Answer: We need to use -direct argument in import command to use direct import fast path and this -direct can be used only with MySQL and PostGreSQL as of now.

**78. How can you sync a exported table with HDFS data in which some rows are deleted?**

Answer: Truncate the target table and load it again.

**79. Differentiate between Sqoop and distCP.**

Answer: DistCP utility can be used to transfer data between clusters whereas Sqoop can be used to transfer data only between Hadoop and RDBMS.

**80. How can you import only a subset of rows form a table?**

Answer: By using the WHERE clause in the sqoop import statement we can import only a subset of rows.

**81. How do you clear the data in a staging table before loading it by Sqoop?**

Answer: By specifying the –clear-staging-table option we can clear the staging table before it is loaded. This can be done again and again till we get proper data in staging.

**83. How can you export only a subset of columns to a relational table using sqoop?**

Answer: By using the –column parameter in which we mention the required column names as a comma separated list of values.

**84. Which database the sqoop metastore runs on?**

Answer: Running sqoop-metastore launches a shared HSQLDB database instance on the current machine.

**85. How will you update the rows that are already exported?**

Answer: The parameter –update-key can be used to update existing rows. In it a comma-separated list of columns is used which unily identifies a row. All of these columns is used in the WHERE clause of the generated UPDATE ry. All other table columns will be used in the SET part of the ry.

**86. You have a data in HDFS system, if you want to put some more data to into the same table, will it append the data or overwrite?**

Answer: No it can’t overwrite, one way to do is copy the new file in HDFS.

**87. Where can the metastore database be hosted?**

Answer: The metastore database can be hosted anywhere within or outside of the Hadoop cluster.

**88. Which is used to import data in Sqoop ?**

Answer: In SQOOP import command is used to import RDBMS data into HDFS. Using import command we can import a particular table into HDFS.

**89. What is the role of JDBC driver in a Sqoop set up?**

Answer: To connect to different relational databases sqoop needs a connector. Almost every DB vendor makes this connecter available as a JDBC driver which is specific to that DB. So Sqoop needs the JDBC driver of each of the database it needs to interact with.

**90. How to import only the updated rows form a table into HDFS using sqoop assuming the source has last update timestamp details for each row?**

Answer: By using the lastmodified mode. Rows where the check column holds a timestamp more recent than the timestamp specified with –last-value are imported.

**91. What is InputSplit in Hadoop?**

Answer: When a hadoop job is run, it splits input files into chunks and assign each split to a mapper to process. This is called Input Split.

**93. How can you schedule a sqoop job using Oozie?**

Answer: Oozie has in-built sqoop actions inside which we can mention the sqoop commands to be executed.

**94. Is JDBC driver enough to connect sqoop to the databases?**

No. Sqoop needs both JDBC and connector to connect to a database.

When to use --target-dir and when to use --warehouse-dir while importing data?How can you import only a subset of rows form a table?

By using the WHERE clause in the sqoop import statement we can import only a subset of rows.

**96. What is the advantage of using --password-file rather than -P option while preventing the display of password in the sqoop import statement?**

The --password-file option can be used inside a sqoop script while the -P option reads from standard input , preventing automation.

**97. How can you avoid importing tables one-by-one when importing a large number of tables from a database?**

Using the command

sqoop import-all-tables

--connect

--usrename

--password

--exclude-tables table1,table2 ..

This will import all the tables except the ones mentioned in the exclude-tables clause.

**98.When the source data keeps getting updated frently, what is the approach to keep it in sync with the data in HDFS imported by sqoop?**

sqoop can have 2 approaches.

**a** − To use the --incremental parameter with append option where value of some columns are checked and only in case of modified values the row is imported as a new row.

**b** − To use the --incremental parameter with lastmodified option where a date column in the source is checked for records which have been updated after the last import.

* 1. **Is it possible to add a parameter while running a saved job?**

Yes, we can add an argument to a saved job at runtime by using the --exec option

sqoop job --exec jobname -- -- newparameter

* 1. **How do you fetch data which is the result of join between two tables?**

By using the --ry parameter in place of --table parameter we can specify a sql ry. The result of the ry will be imported.

* 1. **How can we slice the data to be imported to multiple parallel tasks?**

Using the --split-by parameter we specify the column name based on which sqoop will divide the data to be imported into multiple chunks to be run in parallel.

* 1. **How can we load to a column in a relational table which is not null but the incoming value from HDFS has a null value?**

By using the –input-null-string parameter we can specify a default value and that will allow the row to be inserted into the target table.

* 1. **What does this sqoop command achieve?**

$ sqoop import --connnect <connect-str> --table foo --target-dir /dest \

**111. What happens when a table is imported into a HDFS directory which already exists using the –apend parameter?**

**112. How can you control the mapping between SQL data types and Java types?**

**113. How to import only the updated rows form a table into HDFS using sqoop assuming the source has last update timestamp details for each row?**

**114. What does the following ry do?**

$ sqoop import --connect jdbc:mysql://host/dbname --table EMPLOYEES \

--where "start\_date > '2012-11-09'

**115. Give a Sqoop command to import all the records from employee table divided into groups of records by the values in the column department\_id.What does the following ry do?**

$ sqoop import --connect jdbc:mysql://db.foo.com/somedb --table sometable \

--where "id > 1000" --target-dir /incremental\_dataset --append

**116. Give a sqoop command to import data from all tables in the MySql DB DB1.Give a command to execute a stored procedure named proc1 which exports data to from MySQL db named DB1 into a HDFS directory named Dir1.What is a sqoop metastore?**

It is a tool using which Sqoop hosts a shared metadata repository. Multiple users and/or remote users can define and execute saved jobs (created with sqoop job) defined in this metastore.

Clients must be configured to connect to the metastore in sqoop-site.xml or with the --meta-connect argument.

**118.Give the sqoop command to see the content of the job named myjob?**

Sqoop job –show myjob

###### **122. What are the destination types allowed in Sqoop Import command?**

Currently Sqoop Supports data imported into below services.

* HDFS
* Hive
* HBase
* HCatalog
* Accumulo

###### **124. How Many Mapreduce jobs and Tasks will be submitted for Sqoop copying into HDFS?**

For each sqoop copying into HDFS only one mapreduce job will be submitted with **4 map tasks**. There will not be any reduce tasks scheduled.

###### **125. How can we control the parallel copying of RDBMS tables into hadoop ?**

We can control/increase/decrease speed of copying by configuring the number of map tasks to be run for each sqoop copying process. We can do this by providing argument **-m 10 or  –num-mappers 10 argument**to sqoop import command. If we specify **-m 10**then it will submit 10 map tasks parallel at a time. Based on our requirement we can increase/decrease this number to control the copy speed.

###### **126. What is the criteria for specifying parallel copying in Sqoop with multiple parallel map tasks?**

To use multiple mappers in Sqoop, RDBMS table must have one **primary key column** (if present) in a table and the same will be used as split-by column in Sqoop process. If primary key is not present, we need to provide any uni key column or set of columns to form uni values and these should be provided to **-split-by** column argument.

.

###### **128. What is the example connect string for Oracle database to import tables into HDFS?**

###### **129. While connecting to MySQL through Sqoop, I am getting Connection Failure exception what might be the root cause and fix for this error scenario?**

This might be due to insufficient permissions to access your MySQL database over the network. To confirm this we can try the below command to connect to MySQL database from Sqoop’s client machine.

|  |  |
| --- | --- |
| 1  2 | $ mysql --host=MySql node&gt; --database=test --user= --password= |

If this is the case then we need grant permissions user @ sqoop client machine as per the answer to stion 6 in this post.

###### **130. While importing tables from Oracle database, Sometimes I am getting java.lang.IllegalArgumentException: Attempted to generate class with no columns! or NullPointerException what might be the root cause and fix for this error scenario?**

**While dealing with Oracle database from Sqoop, Case sensitivity of table names and user names matters highly. Most probably by specifying these two values in UPPER case will solve the issue unless actual names are mixed with Lower/Upper cases. If these are mixed, then we need to provide them within double quotes.**

**In case, the source table is created under different user namespace, then we need to provide table name as USERNAME.TABLENAME as shown below.**

MySQL

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | sqoop import \  --connect jdbc:oracle:thin:@oracle.example.com/ORACLE \  --username SQOOP \  --password sqoop \  --table SIVA.EMPLOYEES |

**133. You successfully imported a table using Apache Sqoop to HBase but when you ry 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.

**136. How can Sqoop be used in Java programs?,**  
In the Java code Sqoop jar is included in the classpath. The required parameters are created to Sqoop programmatically like for CLI (command line interface). Sqoop.runTool() method also invoked in Java code.

**137. 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

**139. I am having around 500 tables in a database. I want to import all the tables from the database except the tables named Table498, Table 323, and Table199. How can we do this without having to import the tables one by one?**  
This can be proficient 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 Table498, Table 323, Table 199

**140. You use –split-by clause but it still does not give optimal performance then how will you improve the performance further.**  
Using the –boundary-ry clause. Generally, sqoop uses the SQL ry select min (), max () from to find out the boundary values for creating splits. However, if this ry is not optimal then using the –boundary-ry argument any random ry can be written to generate two numeric columns.

**141. 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.

**143. Give a sqoop command to import data from all tables in the MySql DB DB1.**  
sqoop –tables –connect jdbc:mysql://host/DB1

**145. The first and best function of Sqoop?**

Sqoop can import individual tables or entire databases into HDFS. The data is stored in the native directories and files in the HDFS file system.

**146. What is export in Sqoop?**

Sqoop can export data directly from HDFS into a relational database using a

target table definition based on the specifics of the target database.

**147. Why Sqoop uses mapreduce in import/export operations?**

 Sqoop uses MapReduce to import and export the data, which provides parallel operation as well as fault tolerance.

**148. What does it mean non-Hadoop data?**

It is one of the most popular Big Data tools that leverages the competency to haul out data from a non-Hadoop data store by transforming information into a form that can be easily accessed and used by Big Data Hadoop

**149. What are the underlying technologies in Sqoop?**

It has a high degree of dependence on underlying technologies like HDFS and MapReduce.

**150. What is the data loading or import in Sqoop?**

Load directly into Hive tables, creating HDFS files in the background and the Hive metadata automatically

**151. Sqoop imports data into three kinds of data storage what are those?**

Hive Tables

HDFS files

Hbase (HBase is an open-source, distributed, versioned, non-relational database modeled after Google’s Bigtable)

**152. The nine functions of Sqoop?**

* + 1. Full Load
    2. Incremental Load
    3. Parallel import/export
    4. Import results of SQL ry
    5. Compression
    6. Connectors for all major RDBMS Databases
    7. Kerberos Security Integration
    8. Load data directly into Hive/Hbase
    9. Support for Accumulo

**154. Whether Sqoop will do aggregations?**

Sqoop just imports and exports the data; it does not do any aggregations.

**155. How to check sqoop version?**

The Sqoop version can be checked using the following command in the command line:

sqoop version

For example,  numb-mappers 10.

**157. How to delete sqoop job?**

A job can be deleted in Sqoop by the following syntax:

--delete <job id>

This is used to delete save d jobs.

**159. However, to use the tool successfully, the following conditions have to be met:**

* The tables should have a single column primary key.
* All the columns of each table must be imported.
* Using conditions having a WHERE clause and non-default splitting columns must be avoided.

**165. What is accumulo in sqoop?**

The Apache Acccumulo is a key value store that is developed over the Hadoop File system. The handling of larger sets of data is easier with Accumulo. The data in Accumulo is stored in the Haddop File system. In Sqoop, there is a functionality that allows users to directly import a table from the Accumulo data store.

**168. How much memory does a sqoop client require to initialize the job?**

A Sqoop client may require a minimum of 1Gb of memory to initialize a job.

**169. How to pass the schema name in the sqoop?**

The schema name can be provided by using the following syntax:

$ **sqoop** import ... --table custom\_table -- --schema custom\_schema

**170. What is the syntax to see db in sqoop?**

The databases can be viewed by using the following command:

$sqoop  **list**-databases

**SPARK**

### **1. Compare Hadoop and Spark.**

We will compare Hadoop MapReduce and Spark based on the following aspects:

|  |  |  |
| --- | --- | --- |
| **Feature Criteria** | **Apache Spark** | **Hadoop** |
| **Speed** | 100 times faster than Hadoop | Slower speed |
| **Difficulty** | Easy to learn | Tough to learn |
| **Interactivity** | Has interactive modes | No interactive mode except Pig & Hive |
| **Activities** | Multitasking - like batch processing, Steaming, Machine learning, Interactive SQL queries. | Only batch processing. |

### **2. What is Apache Spark?**

* [**Apache Spark**](https://www.edureka.co/blog/spark-tutorial/) is fast in-memory open-source cluster computing framework for big data and real-time processing.

### **3. Explain the key features of Apache Spark.**

The following are the key features of Apache Spark:

**Polyglot**

**Scalable**

**Speed**

**Multiple Formats**

**Powerful caching**

**Real Time Computation**

**Lazy Evaluation**

**deployment**

**Hadoop Integration**

**Machine Learning**

**Easy programming –** Spark has 100 ‘s operator for programming.

### **4. What are the languages supported by Apache Spark and which is the most popular one?**

Apache Spark supports the following four languages: Scala, Java, Python and R. Among these languages, Scala and Python have interactive shells for Spark.

### **5. What is YARN?**

YARN is yet another resource negotiator is features of spark and Hadoop useful for

### resource management and job scheduling.

Resource maneger consist of Nodemanager ,container and application master.

YARN is useful for real time processing, in –memory execution

### **6. Do you need to install Spark on all nodes of YARN cluster?**

No, Spark runs independently from its installation.  when dispatching jobs to the cluster, there are some configurations to run YARN eg. master, deploy-mode, driver-memory, executor-memory, executor-cores, and ue.

### **Is there any benefit of learning MapReduce if Spark is better than MapReduce?**

Spark in not cost efficient so when batch processing-slow response is required we can use map-reduce

### **Explain the concept of Resilient Distributed Dataset (RDD).**

Immutable collection of objects which is distributed among different nodes of cluster.

RDD can be created in two ways , by reffering external dataset or parallelizing dataset.

Partition can stored in memory or disks.

RDD has following properties:-

**Immutable**  :- read only datasets

**Partitioned:-**

**Coarse grained operations** :- it means operations are applied on all objects in datasets.

Eg. A map , filter, groupBy

Fault Tolerance –

Since RDDs are created over a set of transformations , it logs those transformations

**Lazy evaluations:**

When action is invoked then only memory is allocated allocated to RDD.

**Caching and Persistance –**

We can cache rdd’s which we are reffered in multiple actions or transformation.

### **9. How do we create RDDs in Spark?**

Spark provides two methods to create RDD:

1. By parallelizing dataset in your Driver program using sc.parallelize.

3. By loading an dataset from external storage like HDFS, HBase, shared file system.

### **10. What is Executor Memory in a Spark application?**

Every spark application has same fixed heap size and fixed number of cores for a spark executor. The heap size is what referred to as the Spark executor memory which is controlled with the spark.executor.memory property of the **–executor-memory** flag. Every spark application will have one executor on each worker node. The executor memory is basically a measure on how much memory of the worker node will the application utilize.

### **11. Define Partitions in Apache Spark.**

Partition is a smaller and logical division of data to speed up process.  Spark use partitions to keep datasets.

### **12. List the functions of Spark SQL.**

Spark SQL is capable of:

1. Loading data from a variety of structured sources.
2. Querying data using SQL statements, both inside a Spark program and from external tools that connect to Spark SQL through standard database connectors (JDBC/ODBC). For instance, using business intelligence tools like Tableau.
3. Providing rich integration between SQL and regular Python/Java/Scala code, including the ability to join RDDs and SQL tables, expose custom functions in SQL, and more.

### **13. What operations does RDD support?**

RDDs support two types of operations: transformations and actions.

Transformations: Transformations are functions applied existing RDD to create new RDD it is lazily computed ie it does not occure until action occurs. eg. map, reduceByKey and filter,

Actions: Actions is results/outputs of transformation.

### **14. What do you understand by Transformations in Spark?**

Transformations: Transformations are functions applied to existing RDD to create new RDD it is lazily computed ie it does not occure until action occurs. eg. map, reduceByKey and filter saw,

### **15. Define Actions in Spark.**

An action’s execution is the result /output of all transformations. Actions uses DAG graph to load data into original RDD, carry out all intermediate transformations and return final results to Driver program or file system.

take()is action takes all the values from RDD to a local node.

|  |  |
| --- | --- |
|  | saveAsTextFile(“MoviesData.txt”) save RDD into a text file. |

### **16. Define functions of SparkCore.**

Spark Core is the base engine for large-scale parallel and distributed data processing and can be written in java, Scala, and Python APIs offe. SparkCore performs various important functions like

memory management,

job scheduling

monitoring jobs,

fault-tolerance,

interaction with storage systems.

Further, additional libraries, allow workloads for streaming, SQL, and machine learning.

### **17. What do you understand by Pair RDD?**

pairRDD operations are applied on each key/value element in parallel.its data tuple.

Eg. reduceByKey() method that collects data based each key and

a join() method that combines different RDDs together, based on same key.

### **18. Name the components of Spark Ecosystem.**

1. **Spark Core**: performs large-scale parallel and distributed data processing
2. **Spark Streaming**: performs real-time streaming
3. **Spark SQL**: Access and process data of the different databases.
4. **GraphX**: perform graph-parallel computation
5. **MLlib**: Performs machine learning

### **19. How is Streaming implemented in Spark? Explain with examples.**

Spark Streaming is series of RDDs (Resilient Distributed Datasets) to process the real-time data,

It is a use with core Spark API.

High-throughput and fault-tolerant processing .

Data from different sources like kafka,Flume, HDFS,S3,twitter,stock market goes to streaming engine and then streamed data goes to live dashboards and databases.



### **20. Is there a module to implement SQL in Spark? How does it work?**

Spark SQL supports querying data via SQL or HQL.

It works by using four libraries-

1. Data Source API-

Structured data is loaded from different data sources eg. , mysql,db2,Hbase,Hadoop,Hive, RDBMS, oracle, Teradata

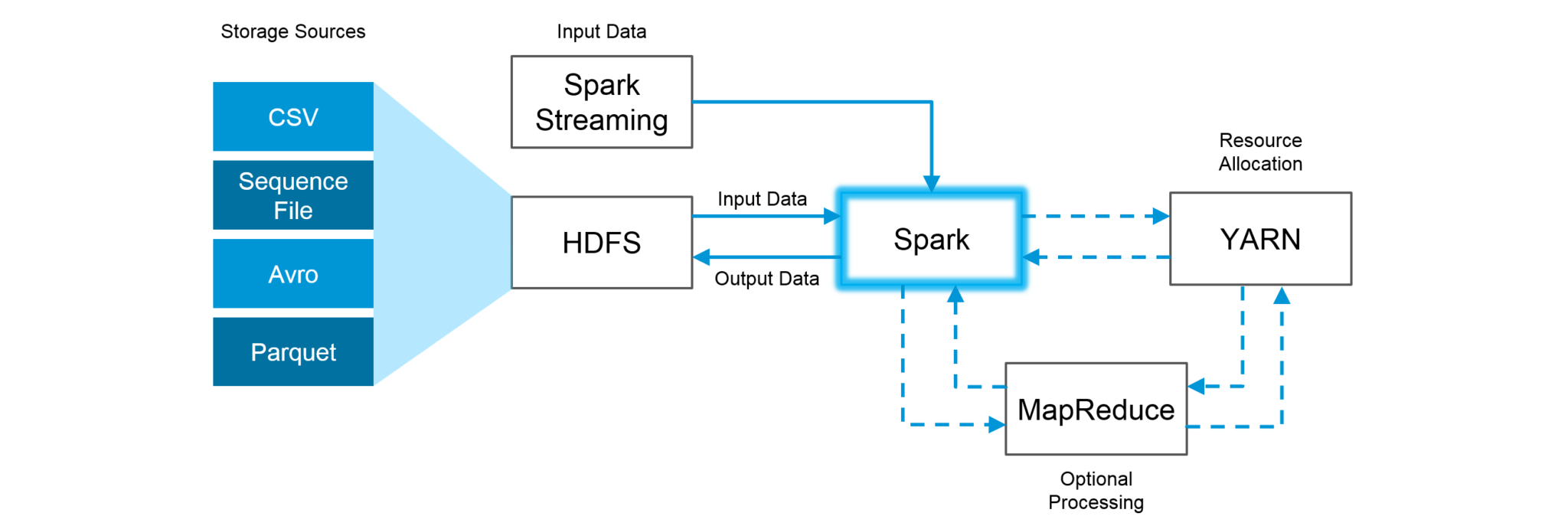
1. Into named columns called as DataFrame or Dataset API- using

Schema RDD and then data passes to

1. Interpreter & Optimizer where it get parse and then goes to
2. SQL Service where it get process and finally we get results.

### Spark SQL - Spark Interview Questions - Edureka

### **21. How can Apache Spark be used alongside Hadoop?**

**Fig**

Hadoop components can be used alongside Spark in the following ways:

1. **HDFS**: Spark can run on top of HDFS to use distributed and replicated storage.
2. **MapReduce** and **Spark** - MapReduce and Spark can be use together where MapReduce is used for batch processing and Spark for real-time processing.

### **22. What is RDD Lineage?**

Spark does not support data replication hence if any data lost occur, data partitions rebuilding occurs using RDD lineage.

**23. What are the advantages of Scala?**

Among various other benefits of the language, here are a few:

highly scalable

highly testable

highly maintainable and productive

concurrent programming

both object-oriented and functional

It has no boilerplate code

Singleton objects are a cleaner solution than static

Scala arrays use regular generics

Scala has native tuples and concise code

**24. Do real-time data processing is possible with Spark SQL?**

Not directly but we can register existing RDD as a SQL table and trigger the SQL queries on priority.

### **25. How can you trigger automatic clean-ups in Spark to handle accumulated metadata?**

by setting the parameter ‘spark.cleaner.ttl’

**26. Who designed Scala? Which is the latest version?**

Martin Odersky, a German computer scientist, began designing it in 2001 at EPFL, Switzerland. latest version of scala is 2.12.10

### **27. What is Spark Driver?**

Spark Driver declares transformations and actions on RDD data and submits RDD graph to cluster manager for that spark driver creates SparkContext which can connects to cluster manager.

### **28. What file systems does Spark support?**

The following three file systems are supported by Spark:

1. Hadoop Distributed File System (HDFS).
2. Local File system.
3. Amazon S3

### **29. What are the various levels of persistence in Apache Spark?**

1. **MEMORY\_ONLY:**

Data is cached in memory in non-serialized format.

1. **MEMORY\_AND\_DISK:**

Data is cached in memory. If memory is not available, remaining blocks from memory are serialized to disk.

1. **DISK\_ONLY:**

Data is cached on disk only in serialized format.

1. **OFF\_HEAP:** store the data in off-heap memory.

### **30. What is Spark Executor?**

When SparkContext connects to a cluster manager, it run executors on clusternodes. Executors are Spark processes that run computations and store the data on the worker node.

### **31. Name types of Cluster Managers in Spark.**

The Spark framework supports three major types of Cluster Managers:

1. **Standalone**: A basic cluster manager.
2. **Apache Mesos**: commonly-used cluster manager, which runs Spark, Hadoop MapReduce and other applications.
3. **YARN**: do resource management

### **32. What do you understand by worker node?**

Worker node is slave node. Master node assigns work and worker node performs the assigned tasks with the help of data stored on the clusternode and report the resources to the master. Based on the resource availability, the master schedule tasks.

### **33. Illustrate some demerits of using Spark.**

The following are some of the demerits of using Apache Spark:

1. Spark utilizes more storage space compared to Hadoop and MapReduce.
2. Spark’s “in-memory” capability is expensive
3. **Not purely Real Time:-**

In [Spark Streaming](http://data-flair.training/blogs/apache-spark-streaming-comprehensive-guide/), live stream of data is divided into batches of the pre-defined time intervals and each batch is considered as [Spark Resilient Distributed Database (RDD).](http://data-flair.training/blogs/rdd-in-apache-spark/)Then these RDD is processed using map, reduce, join operations etc. Again transformation results ie actions are returned in batches. Thus, it is not real time processing but Spark is near real-time processing of live data.

1. **Latency**

Apache Spark has higher latency as compared to [Apache Flink](http://data-flair.training/blogs/apache-flink-big-data-unified-platform/).

1. **No File Management System**

Spark relies on HDFS for file storage

1. **Manual Optimization**

we need to manually optimize spark jobs for that we need to use partitioning and caching techniques.

1. **Back pressure is build up of data**:-

When the buffer overflows , data is not a transferred until the buffer get empty.

We can overcome these limitations of Spark by using [Apache Flink – 4G of Big Data](http://data-flair.training/blogs/big-data-getting-matured-with-unified-platform-apache-flink/).

### **34. List some use cases where Spark outperforms Hadoop in processing.**

1. **Sensor Data Processing**: Apache Spark’s “In-memory” computing works best here.
2. **Real Time Processing**: Spark is preferred over Hadoop for real-time querying of data. e.g. Stock Market Analysis, Banking, Healthcare, Telecommunications, etc.
3. **Stream Processing**: For processing logs and detecting frauds in live streams for alerts, Apache Spark is the best solution.
4. **Big Data Processing**:Spark runs upto 100 times faster than Hadoop when it comes to processing medium and large-sized datasets.

### **35. Can you use Spark to access and analyze data stored in Cassandra databases?**

Yes, it is possible if you use Spark Cassandra Connector.

### **36. Is it possible to run Apache Spark on Apache Mesos?**

Yes , here apache mesos act as cluster manager and allocate tasks to the worker nodes also do resource management.

### **37. How can you minimize data transfers when working with Spark?**

1. Using Broadcast Variable- increase efficiency of joins of small and large RDDs.
2. Using Accumulators –update variable values while executing.

### **38. Explain accumulators in Apache Spark.**

Accumulators are variables that are used for aggregating operations from executors. It is applicable to commutative and associative operations. Eg. Sum and Max

Eg. Val max = Sc.accumulator(99,”max\_value”)

### **39. What is a DStream in Apache Spark?**

**Discretized Stream** is a continuous streams of data received from various data sources like Apache Kafka, HDFS, and Apache Flume.

It is continuous series of RDDs with predefined time interval.

DStreams have two operations:

1. Transformations- computations on existing Dstream to produce new Dstream.
2. Output operations that write data to an external system.

### **=======================================================**

### **40. Does Apache Spark provide checkpoints?**

Checkpoints are similar to checkpoints in gaming. They make it run 24/7 and make it resilient to failures unrelated to the application logic.

**Figure:**

Lineage graphs are always useful to recover RDDs from a failure but this is generally time-consuming if the RDDs have long lineage chains. Spark has an API for checkpointing i.e. a REPLICATE flag to persist. However, the decision on which data to checkpoint – is decided by the user. Checkpoints are useful when the lineage graphs are long and have wide dependencies.

for messaging between the workers and masters.

**41. What is File System API?**

FS API can read data from different storage devices like HDFS, S3 or local FileSystem. Spark uses FS API to read data from different storage engines.

**42. Why Partitions are immutable?**

Every transformation generates new partition.  Partitions use HDFS API so that partition is immutable, distributed and fault tolerance. Partition also aware of data locality.

**43. What is Map and flatMap in Spark?**

The map is a specific line or row to process that data. In FlatMap each input item can be mapped to multiple output items (so the function should return a Seq rather than a single item). So most frently used to return Array elements.

**44. How RDD persist the data?**

There are two methods to persist the data, such as persist() to persist permanently and cache() to persist temporarily in the memory. Different storage level options there such as MEMORY\_ONLY, MEMORY\_AND\_DISK, DISK\_ONLY and many more. Both persist() and cache() uses different options depends on the task.

**45. When do you use apache spark? OR  What are the benefits of Spark over Mapreduce?**

* Spark is really fast. As per their claims, it runs programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk. It aptly utilizes RAM to produce the faster results.
* In map reduce paradigm, you write many Map-reduce tasks and then tie these tasks together using Oozie/shell script. This mechanism is very time consuming and the map-reduce task has heavy latency.
* And quite often, translating the output out of one MR job into the input of another MR job might require writing another code because Oozie may not suffice.
* In Spark, you can basically do everything using single application/console (pyspark or scala console) and get the results immediately. Switching between ‘Running something on cluster’ and ‘doing something locally’ is fairly easy and straightforward. This also leads to less context switch of the developer and more productivity.
* Spark kind of equals to MapReduce and Oozie put together.

### **46. What do you understand by Lazy Evaluation?**

Spark is intellectual in the manner in which it operates on data. When you tell Spark to operate on a given dataset, it heeds the instructions and makes a note of it, so that it does not forget – but it does nothing, unless asked for the final result. When a transformation like map() is called on an RDD, the operation is not performed immediately. Transformations in Spark are not evaluated till you perform an action. This helps optimize the overall data processing workflow.

### **47. What do you understand by SchemaRDD in Apache Spark RDD?**

SchemaRDD is an RDD that consists of row objects (wrappers around the basic string or integer arrays) with schema information about the type of data in each column.

SchemaRDD was designed as an attempt to make life easier for developers in their daily routines of code debugging and unit testing on SparkSQL core module. The idea can boil down to describing the data structures inside RDD using a formal description similar to the relational database schema. On top of all basic functions provided by common RDD APIs, SchemaRDD also provides some straightforward relational ry interface functions that are realized through SparkSQL.

Now, it is officially renamed to DataFrame API on Spark’s latest trunk.

### **48. How is Spark SQL different from HQL and SQL?**

Spark SQL is a special component on the Spark Core engine that supports SQL and Hive query Language without changing any syntax. It is possible to join SQL table and HQL table to Spark SQL.

#### **[49.Repartition and coalesce difference?](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)**

a. Using repartition spark can increase/decrease number of partitions of data.  
b. Using coalesce spark only can reduce the number of partitions of input data  
c. Reparition is not efficient than coalesce.

**A.**Both coalesce and repartition are used to modify the number of partitions in an RDD but Coalesce avoids full shuffle.

If you go from 1000 partitions to 100 partitions, there will not be a shuffle, instead each of the 100 new partitions will claim 10 of the current partitions and this does not require a shuffle.

Repartition performs a coalesce with shuffle. Repartition will result in the specified number of partitions with the data distributed using a hash practitioner.

**50**[**.How to create a stream in spark**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

a. Dstream  
b. Structured stream.  
c. DirectStream

#### [**51.how to handle data shuffle in spark?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

Using map partition and foreachpartition to replace all the collect methods in spark.

#### [**52. what are all the file formats supported by spark ?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

Avro, parst, json, xml, csv, tsv, snappy, orc, rc are the file formats supported by spark.  
Raw files as well as the structured file formats also supported by spark for efficient reading.

#### [**53. What are all the internal daemons used in spark?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

ACLs, BlockManager, Memestore, DAGScheduler, SparkContext, Driver, Worker,Executor, Tasks.

#### [**54. What is SPARK UI how to monitor a spark job?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

Jobs- to view all the spark jobs  
Stages- to check the DAGs in spark  
Storages- to check all the cached RDDs  
Streaming- to check the cached RDDs  
Spark history server- to check all the logs of finished spark jobs

#### [**55. how to submit a spark job?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

Using spark-submit and just follow the following program?  
spark-submit –class org.apache.spark.examples.ClassJobName –master yarn –deploy-mode client –driver-memory 4g –num-executors 2 –executor-memory 2g –executor-cores 10  
in the above sample  
–master is a cluster manager  
driver-memory is the actual memory size of the driver  
executor-memory is the actual memory size of the executor  
–num-executors is the total number of executors which are running at the worker nodes.  
–executor-cores number of individual processes that the executor memory can take up.

#### [**56. what is the difference between dataframe and dataset?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

Dataframe is untyped (throw an exception at runtime in case of any error in the schema mismatch)  
Dataset is typed(throw an exception at compile time in case of any error in the schema mismatch)

#### [**57. spark history server how to start?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

./sbin/start-history-server.sh –properties-file history.properties  
Once you successfully start this server then you can check all the logs of all the containers in spark jobs.

#### **58**[**. how to join two dataframes in spark?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

Df1.join(df2).where(df1.col1==df2.col1).where(df1.col1==df2.col1)

**59. What is Distributed?**

RDD can automatically the data is distributed across different parallel computing nodes.

**60. What is Lazy evaluated?**

If you execute a bunch of programs, it’s not mandatory to evaluate immediately. Especially in Transformations, this Laziness is a trigger.

**61. What is Catchable?**

Keep all the data in-memory for computation, rather than going to the disk. So Spark can catch the data 100 times faster than Hadoop.

**62. What is Spark engine responsibility?**

Spark responsible for scheduling, distributing, and monitoring the application across the cluster.

**64. How spark partition the data?**

Spark use map-reduce API to do the partition the data. In Input format we can create number of partitions. By default HDFS block size is partition size (for best performance), but its’ possible to change partition size like Split.

#### [**65. code sample to read a data from text file?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

from pyspark import SparkContext  
SparkContext.stop(sc)  
sc = SparkContext(“local”,”besant”) sqlContext = SQLContext(sc)  
sc.textFile(filename)

#### [**66. code sample to read a data from mysql ?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

spark.read.format(‘jdbc’).options(driver=’com.mysql.jdbc.Driver’,url=”””jdbc:mysql://<host>:3306/<>db?user=<usr>&password=<pass>”””,dbtable=’besant’,numPartitions=4 ).load()

#### [**67. Why Partitions are constant?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

Each change creates new portion. Distributions use HDFS API so fragment is perpetual, flowed and adjustment to inner disappointment. Portion moreover aware of data region.

points.

#### [**68. How RDD hang on the data?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

There are two strategies to bear the data, for instance, hang on() to drive forward forever and hold() to proceed quickly in the memory. Unmistakable limit level decisions there, for instance, MEMORY\_ONLY, MEMORY\_AND\_DISK, DISK\_ONLY and some more. Both endure() and hold() uses assorted choices depends upon the task.

#### [**69. What is DAG and Stage in spark processing?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

FYI the above program, the overall execution plan is as per the DAG scheduler.

For each and every method execution is optimized as per the stages.

**70. What is Immutable?**

Once created and assign a value, it’s not possible to change, this property is called Immutability. Spark is by default immutable, it does not allow updates and modifications. Please note data collection is not immutable, but data value is immutable.

**71. Is it mandatory to start Hadoop to run spark application?**

No not mandatory, but there is no separate storage in Spark, so it use local file system to store the data. You can load data from local system and process it, Hadoop or HDFS is not mandatory to run spark application.

**72. What is SparkCore functionalities?**

SparkCore is a base engine of apache spark framework. Memory management, fault tolarance, scheduling and monitoring jobs, interacting with store systems are primary functionalities of Spark.

#### [**73.code sample to read a data from mysql ?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

spark.read.format(&#39;jdbc&#39;).options(driver=&#39;com.mysql.jdbc.Driver&#39;,url=&quot;&quot;&quot;jdbc:mysql://&lt;host&gt;:3306/&lt;&gt;db

?user=&lt;usr&gt;&amp;password=&lt;pass&gt;&quot;&quot;&quot;,dbtable=’besant&#39;,numPartitions=4 ).load()

#### [**75. Difference among cache() and persist()?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

With cache(), you utilize just the default stockpiling level MEMORY\_ONLY. With persist(), you can indicate which stockpiling level you want.So ache() is equivalent to calling hold on() with the default stockpiling level.Spark has numerous dimensions of determination to browse dependent on what our objectives are.The default continue() will store the information in the JVM pile as unserialized objects. When we work information out to circle, that information is additionally dependably serialized.Different dimensions of tirelessness are MEMORY\_ONLY, MEMORY\_ONLY\_SER, MEMORY\_AND\_DISK, MEMORY\_AND\_DISK\_SER, DISK\_ONLY.

#### [**76. What is reduce() action?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

It takes a capacity that works on two components of the sort in your RDD and returns another component of a similar kind. A straightforward case of such a capacity is +, which we can use to whole our RDD. With lessen(), we can undoubtedly entirety the components of our RDD, tally the quantity of components, and perform different sorts of conglomerations.

### **77. How do you specify the number of partitions while creating an RDD? What are the functions?**

**A.**You can specify the number of partitions while creating a RDD either by using the sc.textFile or by using parallelize functions as follows:

Val rdd = sc.parallelize(data,4)

val data = sc.textFile(“path”,4)

### **78. What are the optimizations that developer can make while working with spark?**

**A.**Spark is memory intensive, whatever you do it does in memory.

Firstly, you can adjust how long spark will wait before it times out on each of the phases of data locality (data local –> process local –> node local –> rack local –> Any).

Filter out data as early as possible. For caching, choose wisely from various storage levels.

Tune the number of partitions in spark.

### **79. How can you connect Hive to Spark SQL?**

**A.**The first important thing is that you have to place hive-site.xml file in conf directory of Spark.

Then with the help of Spark session object we can construct a data frame as,

result = spark.sql(“select \* from <hive\_table>”)

**80. What are client mode and cluster mode?**

Each application has a driver process which coordinates its execution. This process can run in the foreground (**client mode**) or in the background (**cluster mode**). Client mode is a little simpler, but cluster mode allows you to easily log out after starting a Spark application without terminating the application.

**81. How to run spark in Standalone client mode?**

* spark-submit \
* class org.apache.spark.examples.SparkPi \
* deploy-mode client \
* master spark//$SPARK\_MASTER\_IP:$SPARK\_MASTER\_PORT \
* $SPARK\_HOME/examples/lib/spark-examples\_version.jar 10

**82. How to run spark in Standalone cluster mode?**

* spark-submit \
* class org.apache.spark.examples.SparkPi \
* deploy-mode cluster \
* master spark//$SPARK\_MASTER\_IP:$SPARK\_MASTER\_PORT \
* $SPARK\_HOME/examples/lib/spark-examples\_version.jar 10

**83. How to run spark in YARN client mode?**

* spark-submit \
* class org.apache.spark.examples.SparkPi \
* deploy-mode client \
* master yarn \
* $SPARK\_HOME/examples/lib/spark-examples\_version.jar 10

**84. How to run spark in YARN cluster mode?**

* spark-submit \
* class org.apache.spark.examples.SparkPi \
* deploy-mode cluster \
* master yarn \
* $SPARK\_HOME/examples/lib/spark-examples\_version.jar 10

===========================================================================over mental problesms.

**85. What is a schema RDD/DataFrame?**

A SchemaRDD is an RDD composed of Row objects with additional schema information of the types in each column. Row objects are just wrappers around arrays of basic types (e.g., integers and strings).

1. **What are the types of Transformations on DStreams?**

* In **stateless transformations** the processing of each batch does not depend on the data of its previous batches. They include the common RDD transformations like map(), filter(), and reduceByKey().

• **Stateful transformations**, in contrast, use data or intermediate results from previous batches to compute the results of the current batch. They include transformations based on sliding windows and on tracking state across time.

**87. What is Receiver in Spark Streaming?**

Every input DStream is associated with a **Receiver**object which receives the data from a source and stores it in Spark’s memory for processing.

**88. How Spark achieves fault tolerance?**

Spark stores data in-memory whereas Hadoop stores data on disk. Hadoop uses replication to achieve fault tolerance whereas Spark uses different data storage model, RDD. RDDs achieve fault tolerance through a notion of lineage: if a partition of an RDD is lost, the RDD has enough information to rebuild just that partition.This removes the need for replication to achieve fault tolerance.

**89. How can you achieve high availability in Apache Spark?**

Implementing single node recovery with local file system Using StandBy Masters with Apache ZooKeeper.

**90. Hadoop uses replication to achieve fault tolerance. How is this achieved in Apache Spark?**

Data storage model in Apache Spark is based on RDDs. RDDs help achieve fault tolerance through lineage. RDD always has the information on how to build from other datasets. If any partition of a RDD is lost due to failure, lineage helps build only that particular lost partition.

.

#### **91. Do private members of Companion classes can be accessed through companion objects in Scala?**

Yes, it is possible that private members of Companion classes can be accessed through companion objects in Scala.

#### **92. How will you explain yield keyword in Scala?**

Yield keyword can be used either before or after expressions. It is taken more useful when declared before expression. The return value from every expression will be stored as the collection. The returned value can either be used as a normal collection or iterate in another loop.

#### **93. What do you mean by Scala Traits and how it can be used in Scala programming language?**

Scala trait is an advanced class in Scala that enables the use of multiple inheritances and it can be extended to multiple classes together. In other words, one class can have multiple Scala traits based on requirement.

Traits are used commonly when you need dependency injection. You just need to initiate class with Scala traits and dependency will be injected immediately.

#### **94. How are Monads useful for Scala developers?**

If you want to understand Monads in simple words then it would not be wrong comparing them with a wrapper. As wrappers are used to protect any product and to make it attractive, Monads are used for the same purpose in Scala. They are used to wrap objects together and perform two important functions further. These functions are –

* Identity through “unit” in Scala
* Bind through “flatMap” in Scala

**95. What is Shark?**

Most of the data users know only SQL and are not good at programming. Shark is a tool, developed for people who are from a database background - to access Scala MLib capabilities through Hive like SQL interface. Shark tool helps data users run Hive on Spark - offering compatibility with Hive metastore, ries and data.

**96. Name a few companies that use Apache Spark in production.**

Pinterest, Conviva, Shopify, Open Table

**97. Which spark library allows reliable file sharing at memory speed across different cluster frameworks?**

Tachyon

**98. What are the common mistakes developers make when running Spark applications?**

Developers often make the mistake of-

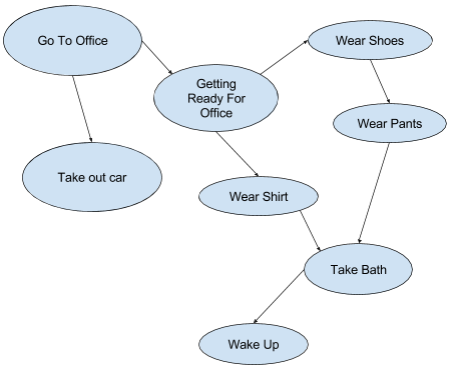
* Hitting the web service several times by using multiple clusters.
* Run everything on the local node instead of distributing it.

Developers need to be careful with this, as Spark makes use of memory for processing.

**99. What is DAG – Directed Acyclic Graph?**

Directed Acyclic Graph – DAG is a graph data structure having edges which are directional and do not have any loops or cycles.

People use DAG almost all the time. Let’s take an example of getting ready for office.

Spark Interview stions – DAG

DAG is a way of representing dependencies between objects. It is widely used in computing. The examples where it is used in computing are:

1. Build tools such Apache Ant, Apache Maven, make, sbt
2. Tasks Dependencies in project management – Microsoft Project
3. The data model of Git

**100. If there is certain data that we want to use again and again in different transformations, what should improve the performance?**

RDD can be persisted or cached. There are various ways in which it can be persisted: in-memory, on disc etc. So, if there is a dataset that needs a good amount computing to arrive at, you should consider caching it. You can cache it to disc if preparing it again is far costlier than just reading from disc or it is very huge in size and would not fit in the RAM. You can cache it to memory if it can fit into the memory.

[Watch this video](https://cloudxlab.com/assessment/slide/adv-spark-programming/553/adv-spark-programming-understanding-persistence) to learn more about persisting Spark RDD.

**101. What happens to RDD when one of the nodes on which it is distributed goes down?**

Since Spark knows how to prepare a certain data set because it is aware of various transformations and actions that have lead to the dataset, it will be able to apply the same transformations and actions to prepare the lost partition of the node which has gone down.

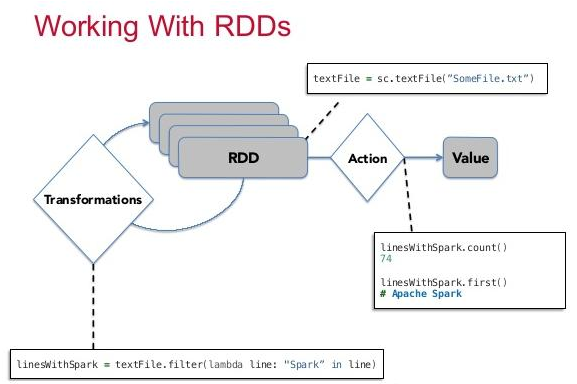
**102. How to save RDD?**

There are few methods provided by Spark:

* **saveAsTextFile**: Write the elements of the RDD as a text file (or set of text files) to the provided directory. The directory could be in the local filesystem, HDFS or any other file system. Each element of the dataset will be converted to text using toString() method on every element. And each element will be appended with newline character “\n”
* **saveAsSenceFile:** Write the elements of the dataset as a Hadoop SenceFile. This works only on the key-value pair RDD which implement Hadoop’s Writeable interface. You can load sence file using sc.senceFile().
* **saveAsObjectFile:** This simply saves data by serializing using standard java object serialization.

**103. When creating an RDD, what goes on internally?**

There are two ways to create RDD. One while loading data from a source. Second, by operating on existing RDD. And an action causes the computation from an RDD to yield the result. The diagram below shows the relationship between RDD, transformations, actions and value/result.

Spark Interview stions – Working With RDD

* **While loading Data from Source –**When an RDD is prepared by loading data from some source (HDFS, Cassandra, in-memory), the machines which exist nearer to the data are assigned for the creation of partitions. These partitions would hold the parts of mappings or pointers to the actual data. When we are loading data from the memory (for example, by using parallelize), the partitions would hold the actual data instead of pointers or mapping
* **By converting an in-memory array of objects –**An in-memory object can be converted to an RDD using parallelize.
* **By operating on existing RDD –**An RDD is immutable. We can’t change an existing RDD. We can only form a new RDD based on the previous RDD by operating on it. When operating on existing RDD, a new RDD is formed. These operations are also called transformations. The operation could also result in shuffling – moving data across the nodes. Some operations that do not cause shuffling: map, flatMap and filter. Examples of the operations that could result in shuffling are groupByKey, repartition, sortByKey, aggregateByKey, reduceByKey, distinct.Spark maintains the relationship between the RDD in the form of a DAG (Directed Acyclic Graph). When an action such reduce() or saveAsTextFile() is called, the whole graph is evaluated and the result is returned to the driver or saved to the location such as HDFS.

**104. What Makes Apache Spark Good At Low-latency Workloads Like Graph Processing And Machine Learning?**

**Answer :**

Apache Spark stores data in-memory for faster model building and training. Machine learning algorithms require multiple iterations to generate a resulting optimal model and similarly graph algorithms traverse all the nodes and edges.These low latency workloads that need multiple iterations can lead to increased performance. Less disk access and  controlled network traffic make a huge difference when there is lots of data to be processed.

#### **105. Give some example for Narrow transformation?**

Answer: Map and Filter.

#### **106. Give some example of wide transformations?**

Answer: GroupByKey and ReduceByKey

#### **107. What are the limitations of Data frame?**

Answer: Data Frame does not have provision for compile-time type safety.

#### **108. Give some example of Action in Spark?**

Answer: Count(),Collect() and reduce(func).

#### **109. What collect does in Spark?**

Answer: It returns all the elements in the RDD to Driver.

#### **110. What is the default storage level in Spark?**

Answer:  Memory\_Only

#### **111. What are the deployment modes in Spark?**

Answer: Cluster mode and Client Mode

#### **112. What are the two types of shared variable available in Apache Spark?**

Answer: Broadcast Variable and Accumulator.

#### **113. What is map transformation in Spark?**

Answer: Map transformation takes a function as input after applying that function to each RDD return another RDD. Its return type can be different from its input type.

#### **114. What is a flatmap transformation in Spark?**

Answer: Flatmap is used when we want to produce multiple elements for each input element. The output of the flatmap is a List of the element through which we can iterate.

#### **115. What is MLib in Spark?**

Answer: MLlib is a distributed machine learning framework built on top of Spark.

#### **116. What is spark shell?**

Answer: Spark Shell is a Spark Application which is written in Scala. It offers a command line environment with auto-completion.

which is helpful in developing our own Standalone Spark Application.

#### **117. What is Caching in Spark Streaming?**

Answer: Caching Streaming is storing streaming RDD in memory. It is a mechanism to speed up applications that access the same RDD multiple times.

#### **118. Write some input sources for Spark Streaming.**

Answer: TCP Sockets, Stream of files, Apache Kafka, Apache Flume, Kinesis etc.

#### **119. Can we use Hive on Spark?**

Answer: Yes, by creating Hive context

#### **120. What is a pipe () operation in Spark?**

Answer: Spark is using Scala, Java, and Python to write the program. However, if one wants to pipe (inject) the data

which is written in other languages Spark provides a general mechanism for that in the form of pipe() method.

#### **121. What are the types of Partitioning in Apache Spark?**

Answer: The types of portioning in Apache Spark are as follows:

* Hash Partitioning
* Range Partitioning

#### **122.** [**What is the other notable feature of RDD and ways to create the RDD?**](https://www.knowledgehut.com/interview-questions/spark#collapse-beginner-950)

* **In-Memory:**Ability to perform operation in the primary memory not in the disk
* **Immutable or Read-Only:**Emphasize in creating the immutable data set.
* **Lazy evaluated:**Spark computing the record when the action is going to perform, not in transformation level.
* **Cacheable:**We can cache the record, for faster processing.
* **Parallel:**Spark has an ability to parallelize the operation on data, saved in     RDD.
* **Partitioned of records:**Spark has ability to partition the record, by default its support 128 MB of partition.
* **Parallelizing:** an existing collection in your driver program.
* **Referencing a dataset** in an external storage system, such as a shared file system, HDFS, HBase

#### **123. Write some function of Spark Context?**

Answer: Used to create Spark RDDs, accumulators, and broadcast variables, access all Spark services and run jobs also to get the status of spark application. Starting and cancelling of Job etc.

#### **124. Write some function of Spark Executor?**

Answer: To run a task that makeup application and to return the result to Driver. It Provides in-memory storage for RDDs cached by user.

**125. Suppose that there is an RDD named DeZyrerdd that contains a huge list of numbers.  The following spark code is written to calculate the average -**

**def DeZyreAvg(x, y):  
return (x+y)/2.0;  
avg = DeZyrerdd.reduce(DeZyreAvg);**

**126. What is wrong with the above code and how will you correct it ?**

Average function is neither commutative nor associative. The best way to compute average is to first sum it and then divide it by count as shown below -

def sum(x, y):  
return x+y;  
total =DeZyrerdd.reduce(sum);  
avg = total / DeZyrerdd.count();

However, the above code could lead to an overflow if the total becomes big. So, the best way to compute average is divide each number by count and then add up as shown below -

cnt = DeZyrerdd.count();  
def divideByCnt(x):  
return x/cnt;  
myrdd1 = DeZyrerdd.map(divideByCnt);  
avg = DeZyrerdd.reduce(sum);

127**. What is ofDim in Scala?**

ofDim() is a method in Scala that lets us create multidimensional arrays. Since these let us store data in more than one dimension, we can store data like in a matrix. Let’s take an example.

scala> import Array.ofDim

import Array.ofDim

scala> var a=ofDim[Int](3,3)

a: Array[Array[Int]] = Array(Array(0, 0, 0), Array(0, 0, 0), Array(0, 0, 0))

scala> var k=1

k: Int = 1

scala> for(i<-0 to 2){

    | for(j<-0 to 2){

    | a(i)(j)={i+k}

    | k+=1

    | }

    | k-=1

    | }

scala> a

res12: Array[Array[Int]] = Array(Array(1, 2, 3), Array(4, 5, 6), Array(7, 8, 9)).

**128. What do you have to say about exception propagation in Scala?**

When a function experiences an exception, it looks for a handler to deal with it. When it fails to find one, it searches for one in the caller method. Failing there, it looks for yet another in the next caller in the chain. Whenever it does find a handler, it makes it catch the exception. This is exception propagation.

**129. What is a BitSet?**

A bitset is a set of non-negative integers depicted as arrays. These arrays are variable in size and packed into 64-bit words. The largest number in a bitset determines its memory footprint. Let’s take an example.

scala> import scala.collection.immutable.\_

import scala.collection.immutable.\_

scala> var nums=BitSet(7,2,4,3,1)

nums: scala.collection.immutable.BitSet = BitSet(1, 2, 3, 4, 7)

scala> nums+=9  //Adding an element

scala> nums

res14: scala.collection.immutable.BitSet = BitSet(1, 2, 3, 4, 7, 9)

scala> nums-=4  //Deleting an element

scala> nums

res16: scala.collection.immutable.BitSet = BitSet(1, 2, 3, 7, 9)

scala> nums-=0  //Deleting an element that doesn’t exist

scala> nums

res18: scala.collection.immutable.BitSet = BitSet(1, 2, 3, 7, 9)

**130. What is a vector in Scala?**

A vector is a general-purpose data structure that is immutable. We can use it when we want to hold a huge number of elements and want random access to them. This data structure extends the trait IndexedSeq and the abstract class AbstractSeq.

scala> import scala.collection.immutable.\_

import scala.collection.immutable.\_

scala> var v1=Vector.empty

v1: scala.collection.immutable.Vector[Nothing] = Vector()

scala> var v2=Vector(7,2,4,3,1)

v2: scala.collection.immutable.Vector[Int] = Vector(7, 2, 4, 3, 1)

scala> var v3:Vector[Int]=Vector(8,2,6,5,9)

v3: scala.collection.immutable.Vector[Int] = Vector(8, 2, 6, 5, 9)

scala> v3=v3 :+7  //Adding a new element

v3: scala.collection.immutable.Vector[Int] = Vector(8, 2, 6, 5, 9, 7)

scala> v2++v3  //Merging two vectors

res19: scala.collection.immutable.Vector[Int] = Vector(7, 2, 4, 3, 1, 8, 2, 6, 5, 9, 7)

scala> v3.reverse  //Reversing a vector

res20: scala.collection.immutable.Vector[Int] = Vector(7, 9, 5, 6, 2, 8)

scala> v3.sorted  //Sorting a vector

res21: scala.collection.immutable.Vector[Int] = Vector(2, 5, 6, 7, 8, 9)

In results 20 and 21, we do not assign the expression to any variable, so not that this doesn’t change the original vectors.

## Cts spark interview stions

**131. How to create array in Spark scala?**

var sarray = Array(1,23,14,23,100,111);

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | sarray.size; | |  | // printing the distinct elements | |  | sarray.distinct; | |  | sarray.map(\_/2); | |  | sarray.filter(\_>3); | |  | sarray(0) | |  | sarray(1); | |  | sarray(0) =1; | |  | // we cannot update the array sarray(11) = 2; | |  | sarray(5) = 111; | |  | sarray.indexOf(3); | |  | (1 until 9).toArray | |  | ( 'a' to 'h').toArray; | |  | Array.range(1,99); | |  | Array.tabulate(4)(n => n \* n) |   **132. How to create arraybuffer in Spark scala?**   |  | | --- | | var empty\_arraybuffer = new ArrayBuffer[Int]()// | |  | empty\_arraybuffer += 3; | |  | empty\_arraybuffer += 4; |   133. What is syntax of the forloop in scala? | |
| package Loops |
|  |  |
|  | object ForLoop { |
|  | def main(args:Array[String]) |
|  | { |
|  | var a=0 |
|  | var b=0 |
|  | var c=0 |
|  | for(a<-1 to 10; b<-4 until 11;if a > b) |
|  | { |
|  | println(a + "|" + b); |
|  |  |
|  | } |
|  | } |
|  |  |
|  | } |

**Syntax of List , Map and Set**

**List:-**

|  |
| --- |
| var l = List(1 , 3 , 4 , 5, 6 ,7); |
|  |  |
|  | for (index <- 0 until l.length) |
|  | { |
|  | if (l(index)%2 != 0) |
|  | println("odd element : " + l(index)); |
|  | else |
|  | println("even element: " + l(index)); |
|  | } |
|  |  |
|  | for ( a <- l) |
|  | println("\t" + a) |
|  | // for populating the another list |
|  |  |
|  | val duplicate\_list = List.tabulate(9)(n=>n); |
|  | println("duplicate elements" + duplicate\_list); |

**Set :-**

|  |
| --- |
| package Collection |
|  | import Set.\_ |
|  |  |
|  | object Set\_Initial { |
|  | def main(args:Array[String]) { |
|  | var s = Set(11,23,13,14,15); |
|  | println("Values are:" + s); |
|  |  |
|  |  |
|  | } |
|  | } |

**Map**

|  |
| --- |
| var imap = Map(1 -> "Heena" , 2 -> "Ajinkya") |
|  | imap+=(3->"Shaikh") |
|  | imap+=(4->"collection") |
|  | println("imap"+ imap) |
|  |  |
|  | //Adding the values in empty map |
|  | //Syntax var map\_name:Map[type1,type2] = Map() |
|  | var empty\_map:Map[Any,Any] = Map(); |
|  | empty\_map+= (1 -> "Sourabh") |
|  | empty\_map+= ("Heena" -> "Salim") |
|  | empty\_map+= ( 2 -> "Shaikh") |
|  |  |

**Syntax of the tuple ?**

|  |
| --- |
| var sTuple = (1,2,3,5,6) |
|  |  |
|  | // var aTuple = scala.io.Source.fromFile("C:\\Users\\jabin\\Desktop\\sample\_test.txt"); |
|  |  |
|  | println(sTuple.\_2); |
|  | println(sTuple.\_3); |
|  |  |

**134. How will you handle error in spark?**

|  |
| --- |
| package spark\_Context\_Programs |
|  | import org.apache.spark.{SparkConf,SparkContext} |
|  | import scala.util.{Try ,Success,Failure} |
|  |  |
|  | object Error\_handling { |
|  |  |
|  | var total =0 |
|  | def main(args:Array[String]):Unit = { |
|  | var configuration = new SparkConf().setAppName("Exceptional Handling").setMaster("local") |
|  | var sc = new SparkContext(configuration); |
|  | var list = sc.parallelize(List(1,2,3,44,55)) |
|  |  |
|  |  |
|  | def calculate():Int = |
|  | { |
|  | if(list.isEmpty()) |
|  | throw new Exception("Elements not fond"); |
|  |  |
|  | total = list.fold(0)((a,b) => a +b) |
|  | return total |
|  |  |
|  | } |
|  |  |
|  | Try(calculate()) match |
|  | { |
|  | case Success(\_) => println(s"The script completed Successfully : $total") |
|  | case Failure(\_) => println ("The script has failed") |
|  |  |
|  | } |
|  |  |
|  | } |
|  | } |

**135. What is case classes and why we use case classes**

**136. What is syntax of the fold operation and what is fold operation?**

|  |
| --- |
| var l = List(2,4,5,6) |
|  | val sum = l.fold(0)(\_+\_) |
|  | val sum1 = l.fold(0){ (a,b) => (a + b) }; |
|  |  |

**137. Which are print operations in Spark ?**

Dump

Println

Take

First

Top

**138. Which operation we use for combining and subtracting items?**

**Union and Intersection**

**139. Explain syntax of the case statements:-**

case when import\_ind = '1' and location\_id != '8277' then '8277' else concat(trim(ship\_duns\_nbr),'\_S') end as source\_location\_id,"

|  |
| --- |
| "case when import\_ind = 1 then 'N' \n" + |
|  | "else\n" + |
|  | "case when vendor\_managed\_inventory\_cd is not null or trim(vendor\_managed\_inventory\_cd) != '' \n" + |
|  | "and vendor\_managed\_inventory\_cd = '5' or vendor\_managed\_inventory\_cd = '6' \n" + |
|  | "or vendor\_managed\_inventory\_cd = '7' or vendor\_managed\_inventory\_cd = '8' \n" + |
|  | "or vendor\_managed\_inventory\_cd = '5' or vendor\_managed\_inventory\_cd = '9' \n" + |
|  | "then 'N' \n" + |
|  | "else \n " + |
|  | "case when stock\_ind='N' and IsNull(dc\_handling\_cd != '') and dc\_handling\_cd is not null and dc\_handling\_cd ='CASE' then 'Y' else 'N' end \n" + |
|  | "end \n" + |
|  | "end as dc\_flowthru\_ind,dotcom\_orderable\_cd \n" + |

**140. What is registerTemp Table?**

#### **141**[**. What is a construction RDD/DataFrame?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

A SchemaRDD is a RDD made out of Row objects with extra construction data of the sorts in every segment. Column objects are only wrappers around varieties of fundamental sorts (e.g., whole numbers and strings).

Please highlight which part of the following code will be executed on the master, and which will be run on each worker node.

val formatter: DateTimeFormatter = DateTimeFormatter.ofPattern("yyyy/MM")

def getEventCountOnWeekdaysPerMonth(data: RDD[(LocalDateTime, Long)]): Array[(String, Long)] = {

val result = data

.filter(e => e.\_1.getDayOfWeek.getValue < DayOfWeek.SATURDAY.getValue)

.map(mapDateTime2Date)

.reduceByKey(\_ + \_)

.collect()

result

.map(e => (e.\_1.format(formatter), e.\_2))

}

private def mapDateTime2Date(v: (LocalDateTime, Long)): (LocalDate, Long) = {

(v.\_1.toLocalDate.withDayOfMonth(1), v.\_2)

}

Describe the following code and what the output will be.

case class User(userId: Long, userName: String)

case class UserActivity(userId: Long, activityTypeId: Int, timestampEpochSec: Long)

val LoginActivityTypeId = 0

val LogoutActivityTypeId = 1

private def readUserData(sparkSession: SparkSession): RDD[User] = {

sparkSession.sparkContext.parallelize(

Array(

User(1, "Doe, John"),

User(2, "Doe, Jane"),

User(3, "X, Mr."))

)

}

private def readUserActivityData(sparkSession: SparkSession): RDD[UserActivity] = {

sparkSession.sparkContext.parallelize(

Array(

UserActivity(1, LoginActivityTypeId, 1514764800L),

UserActivity(2, LoginActivityTypeId, 1514808000L),

UserActivity(1, LogoutActivityTypeId, 1514829600L),

UserActivity(1, LoginActivityTypeId, 1514894400L))

)

}

def calculate(sparkSession: SparkSession): Unit = {

val userRdd: RDD[(Long, User)] =

readUserData(sparkSession).map(e => (e.userId, e))

val userActivityRdd: RDD[(Long, UserActivity)] =

readUserActivityData(sparkSession).map(e => (e.userId, e))

val result = userRdd

.leftOuterJoin(userActivityRdd)

.filter(e => e.\_2.\_2.isDefined && e.\_2.\_2.get.activityTypeId == LoginActivityTypeId)

.map(e => (e.\_2.\_1.userName, e.\_2.\_2.get.timestampEpochSec))

.reduceByKey((a, b) => if (a < b) a else b)

result

.foreach(e => println(s"${e.\_1}: ${e.\_2}"))

}

**142. The following code provides two prepared dataframes with the following structure:**

**DF1: userId, userName**

**DF2: userId, pageId, timestamp, eventType**

**Add the code to join the two dataframes and count the number of events per userName. It should output in the format userName; totalEventCount and only for users that have events.**

def calculate(sparkSession: SparkSession): Unit = {

val UserIdColName = "userId"

val UserNameColName = "userName"

val CountColName = "totalEventCount"

val userRdd: DataFrame = readUserData(sparkSession)

val userActivityRdd: DataFrame = readUserActivityData(sparkSession)

val result = userRdd

.repartition(col(UserIdColName))

// ???????????????

.select(col(UserNameColName))

// ???????????????

result.show()

}

View the answer →

**143. You have a cluster of 10 nodes with 24 CPU cores available on each node.**

**The following code works but might crash on large data sets, or at least will not leverage the full processing power of the cluster. Which is the problematic part and how would you adapt it?**

def calculate(sparkSession: SparkSession): Unit = {

val NumNodes = 10

val userActivityRdd: RDD[UserActivity] =

readUserActivityData(sparkSession)

.repartition(NumNodes)

val result = userActivityRdd

.map(e => (e.userId, 1L))

.reduceByKey(\_ + \_)

result

.take(1000)

}

**144. The following code registers a user-defined function (UDF) and uses it in a ry. (The general business logic is irrelevant to the stion.) What’s problematic about the code such that it might tear down the whole cluster, and how can it be solved?**

(Hint: It has to do with the usage of the categoryNodesWithChildren Map variable.)

def calculate(sparkSession: SparkSession): Unit = {

val UserIdColumnName = "userId"

val CategoryIdColumnName = "categoryId"

val NumActionsColumnName = "numActions"

val OtherCategoryIdColumnName = "otherCategoryId"

val OtherNumActionsColumnName = "otherNumActions"

val categoryNodesWithChildren: Map[Int, Set[Int]] =

Map(0 -> Set(1, 2, 3),

1 -> Set(4, 5),

2 -> Set(6, 7),

3 -> Set(8),

7 -> Set(9, 10)

)

sparkSession.udf.register("isChildOf", (nodeId: Int, parentNodeId: Int) =>

nodeId != parentNodeId && categoryNodesWithChildren.getOrElse(nodeId, Set[Int]()).contains(parentNodeId))

val userCategoryActions = readUserCategoryActions(sparkSession)

val otherUserCategoryActions = userCategoryActions

.select(

col(UserIdColumnName),

col(CategoryIdColumnName).alias(OtherCategoryIdColumnName),

col(NumActionsColumnName).alias(OtherNumActionsColumnName)

)

val joinedUserActions = userCategoryActions

.join(otherUserCategoryActions, UserIdColumnName)

.where("!(isChildOf(categoryId,otherCategoryId) or isChildOf(otherCategoryId,categoryId))")

.groupBy(UserIdColumnName, CategoryIdColumnName, OtherCategoryIdColumnName)

.sum(OtherNumActionsColumnName)

.withColumnRenamed(s"sum($OtherNumActionsColumnName)", OtherNumActionsColumnName)

joinedUserActions.show()

}

?

**145. Complete the missing SQL ry to return the result as shown based on the example data:**

case class User(userId: Long, userName: String)

case class UserActivity(userId: Long, activityTypeId: Int, timestampEpochMs: Long)

val LoginActivityTypeId = 0

val LogoutActivityTypeId = 1

private def readUserData(sparkSession: SparkSession): DataFrame = {

sparkSession.createDataFrame(

sparkSession.sparkContext.parallelize(

Array(

User(1, "Doe, John"),

User(2, "Doe, Jane"),

User(3, "X, Mr."))

)

)

}

private def readUserActivityData(sparkSession: SparkSession): DataFrame = {

sparkSession.createDataFrame(

sparkSession.sparkContext.parallelize(

Array(

UserActivity(1, LoginActivityTypeId, 1514764800000L),

UserActivity(2, LoginActivityTypeId, 1514808000000L),

UserActivity(1, LogoutActivityTypeId, 1514829600000L),

UserActivity(1, LoginActivityTypeId, 1514894400000L))

)

)

}

def calculate(sparkSession: SparkSession): Unit = {

val UserTableName = "user"

val UserActivityTableName = "userActivity"

val userDf: DataFrame = readUserData(sparkSession)

val userActivityDf: DataFrame = readUserActivityData(sparkSession)

userDf.createOrReplaceTempView(UserTableName)

userActivityDf.createOrReplaceTempView(UserActivityTableName)

val result = sparkSession

.sql(s"SELECT ...")

result.show()

}

The output should be this:

| **userName** | **firstLogin** |
| --- | --- |
| Doe, John | 1514764800000 |
| Doe, Jane | 1514808000000 |

View the answer →

**146. Say I have a huge list of numbers in RDD(say myrdd). And I wrote the following code to compute average:**

def myAvg(x, y):  
return (x+y)/2.0;  
avg = myrdd.reduce(myAvg);

**147. What is wrong with it? And How would you correct it?**

The average function is not commutative and associative;  
I would simply sum it and then divide by count.  
def sum(x, y):  
return x+y;  
total = myrdd.reduce(sum);  
avg = total / myrdd.count();  
The only problem with the above code is that the total might become very big thus over flow. So, I would rather divide each number by count and then sum in the following way.  
cnt = myrdd.count();  
def devideByCnd(x):  
return x/cnt;  
myrdd1 = myrdd.map(devideByCnd);  
avg = myrdd.reduce(sum);

**148. Say I have a huge list of numbers in a file in HDFS. Each line has one number.And I want to compute the square root of sum of squares of these numbers. How would you do it?**

# We would first load the file as RDD from HDFS on spark  
numsAsText = sc.textFile(“hdfs://hadoop1.knowbigdata.com/user/student/sgiri/mynumbersfile.txt”);  
# Define the function to compute the squares  
def toSqInt(str):  
v = int(str);  
return v\*v;S  
#Run the function on spark rdd as transformation  
nums = numsAsText.map(toSqInt);  
#Run the summation as reduce action  
total = nums.reduce(sum)  
#finally compute the square root. For which we need to import math.  
import math;  
print math.sqrt(total);

**149. Is the following approach correct? Is the sqrtOfSumOfSq a valid reducer?**

numsAsText =sc.textFile(“hdfs://hadoop1.knowbigdata.com/user/student/sgiri/mynumbersfile.txt”);  
def toInt(str):  
return int(str);  
nums = numsAsText.map(toInt);  
def sqrtOfSumOfSq(x, y):  
return math.sqrt(x\*x+y\*y);  
total = nums.reduce(sum)  
import math;  
print math.sqrt(total);  
A: Yes. The approach is correct and **sqrtOfSumOfSq** is a valid reducer.

**150. Could you compare the pros and cons of the your approach (in stion 2 above) and my approach (in stion 3 above)?**

You are doing the square and square root as part of reduce action while I am squaring in map() and summing in reduce in my approach.  
My approach will be faster because in your case the reducer code is heavy as it is calling math.sqrt() and reducer code is generally executed approximately n-1 times the spark RDD.  
The only downside of my approach is that there is a huge chance of integer overflow because I am computing the sum of squares as part of map.

**151. If you have to compute the total counts of each of the uni words on spark, how would you go about it?**

#This will load the bigtextfile.txt as RDD in the spark lines = sc.textFile(“hdfs://hadoop1.knowbigdata.com/user/student/sgiri/bigtextfile.txt”);  
#define a function that can break each line into words  
def toWords(line):  
return line.split();  
# Run the toWords function on each element of RDD on spark as flatMap transformation.  
# We are going to flatMap instead of map because our function is returning multiple values.  
words = lines.flatMap(toWords);  
# Convert each word into (key, value) pair. Her key will be the word itself and value will be 1.  
def toTuple(word):  
return (word, 1);  
wordsTuple = words.map(toTuple);  
# Now we can easily do the reduceByKey() action.  
def sum(x, y):  
return x+y;  
counts = wordsTuple.reduceByKey(sum)  
# Now, print  
counts.collect()

**152. In a very huge text file, you want to just check if a particular keyword exists. How would you do this using Spark?**

lines = sc.textFile(“hdfs://hadoop1.knowbigdata.com/user/student/sgiri/bigtextfile.txt”);  
def isFound(line):  
if line.find(“mykeyword”) > -1:  
return 1;  
return 0;  
foundBits = lines.map(isFound);  
sum = foundBits.reduce(sum);  
if sum > 0:  
print “FOUND”;  
else:  
print “NOT FOUND”;

**153. Can you improve the performance of this code in previous answer?**

Yes. The search is not stopping even after the word we are looking for has been found. Our map code would keep executing on all the nodes which is very inefficient.  
We could utilize accumulators to report whether the word has been found or not and then stop the job. Something on these line:  
import thread, threading  
from time import sleep  
result = “Not Set”  
lock = threading.Lock()  
accum = sc.accumulator(0)  
def map\_func(line):  
#introduce delay to emulate the slowness  
sleep(1);  
if line.find(“Adventures”) > -1:  
accum.add(1);  
return 1;  
return 0;  
def start\_job():  
global result  
try:  
sc.setJobGroup(“job\_to\_cancel”, “some description”)  
lines = sc.textFile(“hdfs://hadoop1.knowbigdata.com/user/student/sgiri/wordcount/input/big.txt”);  
result = lines.map(map\_func);  
result.take(1);  
except Exception as e:  
result = “Cancelled”  
lock.release()  
def stop\_job():  
while accum.value < 3 :  
sleep(1);  
sc.cancelJobGroup(“job\_to\_cancel”)  
supress = lock.acquire()  
supress = thread.start\_new\_thread(start\_job, tuple())  
supress = thread.start\_new\_thread(stop\_job, tuple())  
supress = lock.acquire()  
[/tab]

#### [**154. Various ways to create contexts in spark ?**](https://www.besanttechnologies.com/apache-spark-interview-questions-and-answers)

a. Sparkconext  
b. Sqlcontext  
c. Sparksession  
d. Sqlcontext.sparkcontext