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

A)

--> Hive is a data warehouse system that is used to analyze structured data. It is built on the top of Hadoop. It was developed by Facebook.

--> Hive provides the functionality of reading, writing, and managing large datasets residing in distributed storage. It runs SQL like queries called HQL (Hive query language) which gets internally converted to MapReduce jobs.

--> Using Hive, we can skip the requirement of the traditional approach of writing complex MapReduce programs. Hive supports Data Definition Language (DDL), Data Manipulation Language (DML), and User Defined Functions (UDF).

The present version of hive is 3.1.3 / April 8, 2022

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

A) No, it is not suitable for OLTP system since it does not offer insert and update at the row level.

3. How is HIVE different from RDBMS? Does hive support ACID

A)

RDBMS Hive

I) It is used to maintain database. It is used to maintain data warehouse.

II) It uses SQL (Structured Query Language). It uses HQL (Hive Query Language).

III) The schema is fixed in RDBMS. Schema varies in it.

IV) Normalized data is stored. Normalized and de-normalized both type of data is stored.

V) Tables in rdms are sparse. The tablele in hive are dense.

VI) It doesn’t support partitioning. It supports automation partition.

VII) No partition method is used. Sharding method is used for partition.

VIII) transactions. If not then give the proper reason.

--> Starting Version 0.14, Hive supports all ACID properties which enable us to use transactions, create transactional tables, and run queries like Insert, Update, and Delete on tables.

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

architecture?

A)

Components of Hive:-

User Interface (UI) –

As the name describes User interface provide an interface between user and hive. It enables user to submit queries and other operations to the system. Hive web UI, Hive command line, and Hive HD Insight (In windows server) are supported by the user interface.

Hive Server –

It is referred to as Apache Thrift Server. It accepts the request from different clients and provides it to Hive Driver.

Driver –

Queries of the user after the interface are received by the driver within the Hive. Concept of session handles is implemented by driver. Execution and Fetching of APIs modelled on JDBC/ODBC interfaces is provided by the user.

Compiler –

Queries are parses, semantic analysis on the different query blocks and query expression is done by the compiler. Execution plan with the help of the table in the database and partition metadata observed from the metastore are generated by the compiler eventually.

Metastore –

All the structured data or information of the different tables and partition in the warehouse containing attributes and attributes level information are stored in the metastore. Sequences or de-sequences necessary to read and write data and the corresponding HDFS files where the data is stored. Hive selects corresponding database servers to stock the schema or Metadata of databases, tables, attributes in a table, data types of databases, and HDFS mapping.

Execution Engine –

Execution of the execution plan made by the compiler is performed in the execution engine. The plan is a DAG of stages. The dependencies within the various stages of the plan is managed by execution engine as well as it executes these stages on the suitable system components.

5. Mention what the Hive query processor does? And Mention what are the

components of a Hive query processor?

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

The components of a Hive query processor include,

Logical Plan Generation

Physical Plan Generation

Execution Engine

Operators

UDF’s and UDAF’s

Optimizer

Parser

Semantic Analyzer

Type Checking

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

A)

Hadoop Mainly works on 3 different Modes:

-->Standalone Mode.

-->Pseudo-distributed Mode.

-->Fully-Distributed Mode.

7. Features and Limitations of Hive.

A) Features of Hive

Features Explanation

--> Supported Computing Engine Hive supports MapReduce, Tez, and Spark computing engine.

--> Framework Hive is a stable batch-processing framework built on top of the Hadoop Distributed File system and can work as a data warehouse.

--> Easy To Code Hive uses HIVE query language to query structure data which is easy to code. The 100 lines of java code we use to query a structure data can be minimized to 4 lines with HQL.

--> Declarative HQL is a declarative language like SQL means it is non-procedural.

--> Structure Of Table The table, structure is similar to the RDBMS. It also supports partitioning and bucketing.

--> Supported data structures Partition, Bucket, and tables are the 3 data structures that hive supports.

--> Supports ETL Apache hive supports ETL i.e. Extract Transform and Load. Before Hive python is used for ETL.

--> Storage Hive supports users to access files from HDFS, Apache HBase, Amazon S3, etc.

--> Capable Hive is capable to process very large datasets of Petabytes in size.

--> Helps in processing unstructured data We can easily embed custom MapReduce code with Hive to process unstructured data.

--> Drivers JDBC/ODBC drivers are also available in Hive.

--> Fault Tolerance Since we store Hive data on HDFS so fault tolerance is provided by Hadoop.

--> Area of uses We can use a hive for data mining, predictive modeling, and document indexing.

Apache Hive Limitations

Limitation Explanation

--> Does not support OLTP Apache Hive doesn’t support online transaction processing (OLTP) but Online Analytical Processing(OLAP) is supported.

--> Doesn’t support subqueries Subqueries are not supported.

--> Latency The latency in the apache hive query is very high.

--> Only non-real or cold data is supported Hive is not used for real-time data querying since it takes a while to produce a result.

--> Transaction processing is not supported HQL does not support the Transaction processing feature.

8. How to create a Database in HIVE?

A) hive> create database database\_name;

9. How to create a table in HIVE?

A) hive> create table table\_name;

10.What do you mean by describe and describe extended and describe

formatted with respect to database and table?

A)

Describe - If you want to see the primary information of the Hive table such as only the list of columns and its data types,the describe command will help you on this.

Describe extended - This will show table columns, data types,table type,location of the table,table size and other details of the table. Other details will be displayed in single line.

Describe formatted - The describe formatted command returns the detailed table information in a clean manner. The results are easy to understand.The complete table information is displayed in a clean manner by describe formatted command

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

A) using the below command we can skip the header row:

hive> tblproperties("skip.header.line.count"="1");

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

A) The HiveQL operators facilitate to perform various arithmetic and relational operations.

There are four types of operators in Hive:

--> Relational Operators

--> Arithmetic Operators

--> Logical Operators

--> Complex Operators

13.Explain about the Hive Built-In Functions?

A) The Hive provides various in-built functions to perform mathematical and aggregate type operations.The are,

a. Collection Functions (such as size(Map<K.V>),Sort\_array(Array<T>) ....)

b. Hive Date Functions (such as To\_date(string timestamp),hour(string date),year(string date) ....)

c. Mathematical Functions (such as round(DOUBLE X),floor(DOUBLE X) ...)

d. Conditional Functions (such as ISNOTNULL(X ),ISNULL( X) ...)

e. Hive String Functions (such as reverse(string X),rtrim(string X),space(INT n) ...)

14. Write hive DDL and DML commands.

A)

# The various Hive DML commands are:

--> LOAD

--> SELECT

--> INSERT

--> DELETE

--> UPDATE

--> EXPORT

--> IMPORT

# The several types of Hive DDL commands are:

--> CREATE

--> SHOW

--> DESCRIBE

--> USE

--> DROP

--> ALTER

--> TRUNCATE

15.Explain about SORT BY, ORDER BY, DISTRIBUTE BY and

CLUSTER BY in Hive.

A)

# SORT BY

The SORT by clause sorts the data per reducer. As a result, if we have N number of reducers, we will have N number of sorted files in the output. These files can have overlapping data ranges.

Also, the output data is not globally sorted because the hive sorts the rows before feeding them to reducers based on the key columns used in the SORT BY clause.

The syntax of the SORT BY clause is as below:

SELECT Col1, Col2,……ColN FROM TableName SORT BY Col1 <ASC | DESC>, Col2 <ASC | DESC>, …. ColN <ASC | DESC>

# ORDER BY

ORDER BY clause orders the data globally. Because it ensures the global ordering of the data, all the data need to be passed from a single reducer only. As a result, the order by clause outputs one single file only.

Bringing all the data on one single reducer can become a performance killer, especially if our output dataset is significantly large. So, we should always avoid the ORDER BY clause in the hive queries. However, if we need to enforce a global ordering of the data, and the output dataset is not that big, we can use this hive clause to order the final dataset globally.

The syntax of the ORDER BY clause in hive is as below:

SELECT Col1, Col2,……ColN FROM TableName ORDER BY Col1 <ASC | DESC>, Col2 <ASC | DESC>, …. ColN <ASC | DESC>

# DISTRIBUTE BY

DISTRIBUTE BY clause is used to distribute the input rows among reducers. It ensures that all rows for the same key columns are going to the same reducer. So, if we need to partition the data on some key column, we can use the DISTRIBUTE BY clause in the hive queries.

However, the DISTRIBUTE BY clause does not sort the data either at the reducer level or globally. Also, the same key values might not be placed next to each other in the output dataset.

As a result, the DISTRIBUTE BY clause may output N number of unsorted files where N is the number of reducers used in the query processing. But, the output files do not contain overlapping data ranges.

The syntax of the DISTRIBUTE BY clause in hive is as below:

SELECT Col1, Col2,……ColN FROM TableName DISTRIBUTE BY Col1, Col2, ….. ColN

# CLUSTER BY

CLUSTER BY clause is a combination of DISTRIBUTE BY and SORT BY clauses together. That means the output of the CLUSTER BY clause is equivalent to the output of DISTRIBUTE BY + SORT BY clauses. The CLUSTER BY clause distributes the data based on the key column and then sorts the output data by putting the same key column values adjacent to each other.

So, the output of the CLUSTER BY clause is sorted at the reducer level. As a result, we can get N number of sorted output files where N is the number of reducers used in the query processing. Also, the CLUSTER by clause ensures that we are getting non-overlapping data ranges into the final outputs. However, if the query is processed by only one reducer the output will be equivalent to the output of the ORDER BY clause.

The syntax of the CLUSTER BY clause is as below:

SELECT Col1, Col2,……ColN FROM TableName CLUSTER BY Col1, Col2, ….. ColN

16.Difference between "Internal Table" and "External Table" and Mention

when to choose “Internal Table” and “External Table” in Hive?

A) For External Tables, Hive stores the data in the LOCATION specified during creation of the table(generally not in warehouse directory). If the external table is dropped, then the table metadata is deleted but not the data.

For Internal tables, Hive stores data into its warehouse directory. If the table is dropped then both the table metadata and the data will be deleted.

#Use INTERNAL tables when:

The data is temporary. You want Hive to completely manage the lifecycle of the table and data.

#Use EXTERNAL tables when:

-->The data is also used outside of Hive. For example, the data files are read and processed by an existing program that doesn’t lock the files.

-->Data needs to remain in the underlying location even after a DROP TABLE. This can apply if you are pointing multiple schema (tables or views) at a single data set or if you are iterating through various possible schema.

-->Hive should not own data and control settings, directories, etc., you may have another program or process that will do those things.

-->You are not creating table based on existing table (AS SELECT).

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

A) Hive stores tables files by default at /user/hive/warehouse location on HDFS file system. You need to create these directories on HDFS before you use Hive. On this location,

you can find the directories for all databases you create and subdirectories with the table name you use.

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

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

19.What is a metastore in Hive? What is the default database provided by

Apache Hive for metastore?

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

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

A) Hive stores metadata information in the metastore using RDBMS instead of HDFS. The reason for choosing RDBMS is to achieve low latency as HDFS read/write operations are time consuming processes.

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

Hive?

A)The partitioning in Hive means dividing the table into some parts based on the values of a particular column like date, course, city or country.

The advantage of partitioning is that since the data is stored in slices, the query response time becomes faster.

22.What is the difference between dynamic partitioning and static

partitioning?

A) In static partitioning we need to specify the partition column value in each and every LOAD statement.

Dynamic partition allow us not to specify partition column value each time.

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

A) By using below command:

SHOW PARTITIONS table\_name

PARTITION(partitioned\_column=’partition\_value’)

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

A) By using the ENABLE OFFLINE clause with ALTER TABLE satatement.

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

A) The bucketing in Hive is a data organizing technique. It is similar to partitioning in Hive with an added functionality that it divides large datasets into more manageable parts known as buckets.

So, we can use bucketing in Hive when the implementation of partitioning becomes difficult.

The concept of bucketing is based on the hashing technique. Here, modules of current column value and the number of required buckets is calculated (let say, F(x) % 3). Now,

based on the resulted value, the data is stored into the corresponding bucket.

26.In Hive, how can you enable buckets?

A) The command to enable bucketing is:

set hive. enforce. bucketing = true;

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

A) Bucketing is an optimization technique in Apache Spark SQL. Data is allocated among a specified number of buckets, according to values derived from one or more bucketing columns.

Bucketing improves performance by shuffling and sorting data prior to downstream operations such as table joins.

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

A) There are several Hive optimization techniques to improve its performance which we can implement when we run our hive queries, thereby focusing on the Performance Tuning as well:

a) Avoid locking of tables

It is extremely important to make sure that the tables are being used in any Hive query as sources are not being used by another process. This can lead to locking of the table and our query can be stuck for an unknown time.

We can use the parameters below for making sure that the tables are not being locked:

set hive.support.concurrency=false;

set hive.txn.manager=org.apache.hadoop.hive.ql.lockmgr.DummyTxnManager;

set hive.bin.strict.locking.mode=false;

b) Use the Hive execution engine as TEZ

While executing Hive queries, TEZ execution engine is the preferred choice because it eliminates unnecessary disk access. It takes data from disk once, performs calculations, and produces output, thus saving us from multiple disk traversals. We can consider TEZ to be a much more flexible and powerful successor to the map-reduce framework.

We can set the parameter below for using TEZ engine:

set hive.execution.engine=tez;

c) Use Hive Cost Based Optimizer (CBO)

Apache Hive provides a cost-based optimizer to improve performance. It generates efficient execution plans like how to order joins, which type of join to perform, the degree of parallelism etc. by examining the query cost. These decisions are collected by ANALYZE statements or the metastore itself, ultimately cutting down on query execution time and reducing resource utilization.

Become a Full-Stack Data Scientist

Date: SUNDAY, 26 Feb 2023 Time: 1 PM – 2 PM IST

We can set the parameter using :

set hive.cbo.enable=true;

d) Parallel execution at a Mapper & Reducer level

We can improve the performance of aggregations, filters, and joins of our hive queries by using vectorized query execution, which means scanning them in batches of 1024 rows at once instead of single row each time.

We should explore the below parameters which will help to bring in more parallelism and which significantly improves query execution time:

set hive.vectorized.execution.enabled=true;

set hive.exec.parallel=true;

For example:

Select a.\*, b.\* from

(select \* from table1 ) a

Join

(select \* from table2 ) b

On a.id=b.id

;

As we can see the two subqueries are independent so this might increase efficiency.

One important thing to note is, parallel execution will increase cluster utilization. If the cluster utilization of a cluster is already very high, parallel execution will not help much.

e) Use STREAMTABLE option

When we are joining multiple tables, we can use STREAMTABLE option. By default, the right-most table gets streamed.

For example: If we are joining 2 tables ‘huge\_table’ join ‘small\_table’, by default ‘small\_table’ gets streamed as it is the rightmost table. In this case, ‘huge\_table’ , being the bigger table, will try to get buffered into memory and might cause java heap space issues or the job might run longer. In this case, what we can do is add /\*+ STREAMTABLE(‘huge\_table’) \*/ and it will make ‘huge\_table’ to be streamed rather than coming into memory.

Hence, in this way, we can be free of remembering the order of joining tables.

f) Use Map Side JOIN Option

If one of the tables in the join is a small table and can be loaded into memory, we can force a MAPSIDE join like shown below:

Select /\*+ MAPJOIN(small\_table) \*/ large\_table.col1,large\_table.col2 from large\_table join small\_table on large\_table.col1 = small\_table.col1;

A map side join can be performed within a mapper without using a Map/Reduce step.

Also, We can let the execution engine take care of this by setting auto.convert.join as True.

set auto.convert.join = True :

g) Avoid Calculated Fields in JOIN and WHERE clause

We should avoid using any calculated fields in JOIN and WHERE clauses as they take a long time to run the Hive query. We can use CTE(Create table expression) to handle those functionalities and can optimize our queries.

For example:

Original query:

select a.coll, b.col2 from table1 as a join table2 as b on (a.coll +50 = b.col2);

Optimized query:

with CTE as

(select a.col1 + 50 as C1 FROM table1 )

select CTE.C1, b.col2 from CTE join table2 b on (CTE.C1 = b.col2);

h) Use SORT BY instead of ORDER BY

Hive supports both ORDER BY and SORT BY causes. ORDER BY works on a single reducer and it causes a performance bottleneck. But, SORT BY orders the data only within each reducer and performs a local ordering where each reducer’s output will be sorted ensuring better performance.

i) Select columns which are needed

While we are using Hive, If we need only a few columns from a table, avoid using SELECT \* FROM as it adds unnecessary time to the execution.

j) Suitable Input format Selection

Using appropriate file formats on the basis of data can significantly increase our query performance. Hive comes with columnar input formats like RCFile, ORC, etc. On comparing to Text, Sequence, and RC file formats, ORC shows better performance because Hive has a vectorized ORC reader which allows reducing the read operations in analytics queries by allowing each column to be accessed individually.

k) Limit (Filter) the data as early as possible

This is the fundamental principle for any tuning where we filter or drop records ahead in the process so that we can avoid dealing with long-running of queries and dealing with unnecessary results.

For example :

a join b where a.col !=null

can be written as

(select \* from a where a.col!=null) join b

l) Use Multi Query Inserts

Here we will have a common dataset (Superset) and then we will use that to insert into multiple tables based on specific conditions specified in the WHERE clause. This helps to load multiple tables in parallel when they have the common superset of data.

m) Modularize the code into logical pieces

This helps in terms of the maintenance of the code. Also helps in restarting the jobs in scenarios of failures. This can be used to achieve parallelism by running modules that are independent of each other thus saving time.

29. What is the use of Hcatalog?

A) HCatalog is a tool that allows you to access Hive metastore tables within Pig, Spark SQL, and/or custom MapReduce applications. HCatalog has a REST interface and command line client that allows you to create tables or do other operations.

You then write your applications to access the tables using HCatalog libraries.

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

A) There are different types of joins given as follows:

--> JOIN.

--> LEFT OUTER JOIN.

--> RIGHT OUTER JOIN.

--> FULL OUTER JOIN.

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

A) Yes, Hive supports cross joins. A cross join is a type of join that produces all possible combinations of rows from the two joined tables. It is also referred to as a Cartesian product or full join.

32.Explain the SMB Join in Hive?

A) Sort merge bucket (SMB) join:

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

We need to enable the following properties to use SMB:

> SET hive.input.format=> org.apache.hadoop.hive.ql.io.BucketizedHiveInputFormat;

> SET hive.auto.convert.sortmerge.join=true;

> SET hive.optimize.bucketmapjoin=true;

> SET hive.optimize.bucketmapjoin.sortedmerge=true;

> SET hive.auto.convert.sortmerge.join.noconditionaltask=true;

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

A) Order by and Sort by are two similarfunctions in Hive which are used to sort data. The difference is that Order by sorts data in ascending or descending order,

while Sort by only sorts data in ascending order. When deciding which one to use, it depends on the sorting requirements of the data.

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

A) DISTRIBUTE BY clause in Hive is used to define the distribution of rows of a table among the different reducers. This clause helps to optimize the query performance as it reduces the need for data shuffling. By defining the distribution of rows among reducers,

Hive can minimize the amount of data being shuffled.

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

A) transfer from HDFS to Hive happens Data through HiveQL queries. HiveQL queries allow users to read, write, and manage large datasets stored in HDFS. With these queries,

users can move data to and from Hive tables, as well as between Hive tables and HDFS. Additionally, using HiveQL, users can join, filter, and aggregate data stored in HDFS.

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

A) The reason why Hive creates a new metastore\_db every time you run it in a different directory is because Hive is designed to look for a metastore\_db in the current directory it is running in. If it does not find one, it will create a new one. This is done to prevent the user from accidentally overwriting existing metastore\_db or other data.

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

A) If you have not issued the command 'SET hive.enforce.bucketing=true;' before bucketing a table in Hive, any attempts to load data into buckets of the table using LOAD DATA or INSERT statements will fail.

38.Can a table be renamed in Hive?

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

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

A)

ALTER TABLE table\_name

CHANGE COLUMN new\_col INT

BEFORE x\_col

40.What is serde operation in HIVE?

A) SerDe stands for Serial izer/Deserializer. It is a framework that is used to read and write data from a variety of input and output sources. In HIVE, it is used to read and write data from HIVE tables by deserializing and serializing the data.

41.Explain how Hive Deserializes and serialises the data?

A)

--> Serializing data means converting a complex data structure into a flat sequence of bytes that can be written to a file or sent over a network. Deserializing data means converting the sequence of bytes back into the original data structure.

--> Hive uses a serialization/deserialization framework called Apache Thrift to serialize and deserialize data. Thrift is a cross-language framework for handling remote procedure calls and serialization of structured data. It allows for efficient and flexible serialization and deserialization of data structures.

--> When a user creates a table in Hive, they specify the schema of the table, which includes the names and data types of each column. Hive then uses Thrift to generate a Java class that represents the schema of the table. This Java class is used to serialize and deserialize data in that table.

--> When data is inserted into a table in Hive, it is first converted into the Java class that represents the table schema. This process is called serialization. The serialized data is then written to HDFS. When data is read from a table in Hive, it is first read from HDFS as a sequence of bytes. This sequence of bytes is then converted back into the Java class that represents the table schema. This process is called deserialization.

--> Hive also supports different serialization and deserialization formats, such as Apache Avro, JSON, and ORC. These formats can be used instead of Thrift serialization and deserialization to improve performance or to support different data types.

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

A)

a)TextInputFormat/HiveIgnoreKeyTextOutputFormat

b)SequenceFileInputFormat/SequenceFileOutputFormat

c)MetadataTypedColumnsetSerDe

d)LazySimpleSerDe

e)ALTER TABLE person SET SERDEPROPERTIES (‘serialization.encoding’=’GBK’)

f)Thrift SerDe

g)Dynamic SerDe

43.What is the need of custom Serde?

A) Despite Hive SerDe users want to write a Deserializer in most cases. It is because users just want to read their own data format instead of writing to it

By using the configuration parameter ‘regex’, the RegexDeserializer will deserialize the data, and possibly a list of column names (see serde2.MetadataTypedColumnsetSerDe).

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

A)Hive supports several complex data types for collections. Some examples include:

Array: An ordered collection of elements of the same type. For example, an array of integers can be declared as follows:

array<int>

Map: A collection of key-value pairs where each key maps to a value. For example, a map of string to integer can be declared as follows:

map<string, int>

Struct: A collection of named fields where each field has a specified data type. For example, a struct with two fields, name of type string and age of type int, can be declared as follows:

struct<name:string, age:int>

These complex data types can be used in Hive tables and queries to represent nested or hierarchical data structures.

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

A) Yes, Hive queries can be executed from script files. There are several ways to do this, but one common method is to use the Hive command line interface (CLI) in combination with shell scripts.

Here's an example of how to execute a Hive query from a script file using the CLI and shell scripting:

Create a script file with your Hive query, let's call it "myquery.hql":

SELECT column1, column2

FROM mytable

WHERE column3 = 'some\_value';

Create a shell script file, let's call it "run\_myquery.sh", that calls the Hive CLI and passes the script file as an argument:

#!/bin/bash

hive -f myquery.hql

Make the shell script executable:

chmod +x run\_myquery.sh

Run the script:

./run\_myquery.sh

This will execute the Hive query in "myquery.hql" and output the results to the console. You can also redirect the output to a file by adding "> output.txt" to the end of the Hive CLI command in the shell script.

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

A)In Hive, the default record delimiter used for text files is the newline character (\n). The default field delimiter used for text files is a tab character (\t).

However, these delimiters can be customized by specifying different delimiter characters using the ROW FORMAT DELIMITED FIELDS TERMINATED BY and LINES TERMINATED BY clauses in the Hive table definition.

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

A)To list all databases in Hive whose name starts with "s", you can use the following command in the Hive CLI:

SHOW DATABASES LIKE 's\*';

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

A)In Hive, both LIKE and RLIKE are comparison operators used to match string patterns. The main difference between them is the syntax used to define the pattern and the complexity of the pattern matching.

--> The LIKE operator performs a simple pattern matching operation based on wildcards. It allows for the use of two wildcards: the percentage sign (%) and the underscore (\_). The percentage sign represents zero, one or multiple characters, while the underscore represents exactly one character. For example, to find all the strings that start with "a" and end with "e", you can use the following query:

SELECT \* FROM my\_table WHERE my\_column LIKE 'a%e';

--> The RLIKE operator, on the other hand, supports regular expressions, which are a more powerful way to define patterns. Regular expressions allow for more complex pattern matching operations, such as searching for strings that contain a specific set of characters in a specific order or matching a string against a complex pattern. For example, to find all the strings that contain the word "cat" followed by any number of characters, you can use the following query:

SELECT \* FROM my\_table WHERE my\_column RLIKE 'cat.\*';

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

A) By using this command below one can change the column data type:

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

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

A) By using the below command:

SELECT CAST(column\_name AS FLOAT) FROM table\_name;

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

A) If you try to cast the string 'abc' to an integer, you will get a ValueError because 'abc' is not a valid integer representation.

52.What does the following query do?

a. INSERT OVERWRITE TABLE employees

The given query performs an insert operation on a Hive table called "employees" and overwrites any existing data in it.

b. PARTITION (country, state)

The data is inserted into the "employees" table partitioned by two columns: "country" and "state".

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

it selects the "cnty" and "st" columns from the "staged\_employees" table

d. FROM staged\_employees se;

The data to be inserted is selected from a temporary/staged table named "staged\_employees".

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

A) To overwrite data in a new table from an existing table in Hive, you can use the INSERT OVERWRITE TABLE command. Here's an example query:

INSERT OVERWRITE TABLE new\_table

SELECT column1, column2, column3

FROM existing\_table

WHERE column4 = 'some\_value';

54.What is the maximum size of a string data type supported by Hive?

Explain how Hive supports binary formats.

A)

-->The maximum size of a string data type supported by Hive depends on the version of Hive being used.

In Hive 1.x and 2.x, the maximum size of a string data type is 2GB.

In Hive 3.x, the maximum size of a string data type is 1TB.

--> Hive supports binary formats through custom SerDe classes and storage handlers. Users can write their own SerDe class or storage handler to handle custom binary formats, which allows Hive to interact with a wide range of data sources.

55. What File Formats and Applications Does Hive Support?

A)

# File Formats: Hive supports various file formats, including:

--> Text Files: Hive can read and write data in plain text files, such as CSV, TSV, and JSON.

--> Sequence Files: Hive can read and write data in Hadoop Sequence Files.

--> ORC Files: ORC (Optimized Row Columnar) files are columnar storage files optimized for Hive queries. Hive supports reading and writing data in ORC files.

--> Parquet Files: Parquet is a columnar storage file format that is optimized for query performance. Hive supports reading and writing data in Parquet files.

# Applications: Hive integrates with various applications, including:

--> Apache Hadoop: Hive is designed to work with Hadoop Distributed File System (HDFS) and MapReduce. Hive can access data stored in HDFS and process it using MapReduce.

--> Apache Spark: Hive can also integrate with Spark, allowing users to query data stored in Spark RDDs (Resilient Distributed Datasets) and DataFrames.

--> JDBC/ODBC Drivers: Hive provides JDBC and ODBC drivers, allowing users to connect to Hive using popular BI tools such as Tableau, Excel, and Power BI.

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

A) Using the ORC format leads to a reduction in the size of the data stored, as this file format has high compression ratios. As the data size is reduced, the time to read and write the data is also reduced.

The ORC format improves query performance also by the way it stores data in a file.

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

A) You can make Hive avoid MapReduce to return query results by setting the hive.exec.mode.local.auto property to true.

58.What is view and indexing in hive?

A)

--> In Hive, a view is a logical abstraction on top of a table or tables that allows you to query the data without actually creating a new table. Views can be used to simplify complex queries, to enforce data security by limiting access to certain columns, and to hide the underlying complexity of the data model.

--> Indexing in Hive refers to the process of creating an index on a table column to improve query performance. Hive supports two types of indexing: Bitmap Index and Compact Index. Bitmap Index is created on columns with low cardinality (few distinct values) while Compact Index is used for columns with high cardinality.

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

A)The name of a view must be unique, and it cannot be the same as any table or database or view's name.

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

A)When creating indexes on Hive tables, there are several types of costs that can be associated, including:

--> Computation Cost: Indexing involves additional computation, which can result in increased CPU utilization and longer processing times.

--> Storage Cost: Indexing requires additional storage space to store the index data, which can increase the storage requirements of the table.

--> Maintenance Cost: Indexes require maintenance to ensure that they remain up-to-date with the underlying data in the table. This can involve additional CPU and I/O resources, as well as increased network traffic if the index is distributed across multiple nodes.

--> Query Performance Cost: While indexes can improve query performance by allowing for faster data access and filtering, they can also have a negative impact on query performance if they are not designed and used properly. For example, if the index is not selective enough, it may not improve query performance, or if the index is overused, it can result in increased query execution times due to the additional processing overhead.

--> Data Consistency Cost: If the table is updated frequently, maintaining the consistency between the index and the table data can be challenging, which can result in additional overhead and complexity.

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

A) To see the index for a specific table use SHOW INDEX:

SHOW INDEX FROM table\_name;

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

A) We can use following commands in Hive to recursively access sub-directories:

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

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

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

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

A) whether or not a SELECT \* query in Hive runs a MapReduce job depends on a variety of factors, including the storage format of the data and the configuration of your Hive environment. Hive is designed to automatically choose the most efficient execution engine for a given query, based on these factors.

64.What are the uses of Hive Explode?

A) Hive Explode is a built-in function in Apache Hive, which is used for "exploding" or splitting a collection column into multiple rows. The collection column can be an array, a map, or a struct. When you apply the Explode function to a collection column, it will create a new row for each element of the collection, with the element's value in place of the collection column.

The main uses of Hive Explode include:

--> Flattening arrays: When you have an array column in your Hive table, you can use the Explode function to transform it into multiple rows, where each row represents an element in the array.

--> Unpacking maps: When you have a map column in your Hive table, you can use the Explode function to unpack it into multiple rows, where each row represents a key-value pair in the map.

--> Exploding nested data: When you have a struct column that contains

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

A) When running Hive as a server, there are several mechanisms available for connecting applications to Hive, including:

--> JDBC Driver: Hive provides a JDBC driver that allows Java applications to connect to Hive through JDBC. The JDBC driver can be used with any programming language that supports JDBC, such as Java, Scala, and Python.

--> ODBC Driver: Hive also provides an ODBC driver that allows applications to connect to Hive through ODBC. The ODBC driver can be used with any programming language that supports ODBC, such as C++, C#, and PHP.

--> Thrift API: Hive also provides a Thrift API that allows applications to connect to Hive through a remote procedure call (RPC) mechanism. The Thrift API can be used with any programming language that supports Thrift, such as Java, Python, and Ruby.

--> WebHCat API: WebHCat is a REST API for Hadoop that provides a mechanism for submitting Hive queries from a web interface. WebHCat can be used with any programming language that supports RESTful web services, such as Java, Python, and Ruby.

Overall, these mechanisms provide a range of options for connecting applications to Hive, depending on the programming language and technology stack being used.

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

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

67.What is the Hive ObjectInspector function?

A) The Hive ObjectInspector function is a built-in function in Apache Hive, which is a data warehouse infrastructure built on top of Hadoop.

In Hive, an ObjectInspector is a class that provides a way to inspect the internal structure of data in a Hive table. The Hive ObjectInspector function allows users to create an instance of an ObjectInspector for a given data type. This function is used to convert data in Hive tables from its internal binary representation into a more human-readable format.

The syntax of the Hive ObjectInspector function is as follows:

ObjectInspector getObjectInspector(typeInfo);

68.What is UDF in Hive?

A)

UDF stands for User-Defined Functions in Hive. A UDF is a way for users to define their own custom functions in Hive. These functions can be used in Hive queries just like any other built-in functions.

UDFs can be divided into three categories based on the input and output types:

--> UDF: User-defined functions that take one or more arguments and return a single value.

--> UDTF: User-defined table functions that take one or more arguments and return a table. UDTFs can be used in the FROM clause of a Hive query to generate a table.

--> UDAF: User-defined aggregation functions that take multiple input rows and return a single output value. UDAFs are used in conjunction with the GROUP BY clause to perform custom aggregations.

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

A) To extract data from HDFS to Hive, you can use the following query:

INSERT OVERWRITE TABLE hive\_table\_name

SELECT \*

FROM hdfs\_file\_path;

70.What is TextInputFormat and SequenceFileInputFormat in hive.

A)In Hive, TextInputFormat and SequenceFileInputFormat are input formats used for reading data from files.

--> TextInputFormat is the default input format in Hive. It is used to read text files where each line represents a record. In TextInputFormat, each line of the file is considered a separate record and is converted into a row in the Hive table.

The default delimiter used is the newline character (\n), but this can be changed by setting the "hive.delimiter" property.

--> SequenceFileInputFormat, on the other hand, is used to read data from binary sequence files. Sequence files are a Hadoop-specific file format that stores data in a binary format, making them more efficient for large-scale data processing. Each record in a sequence file consists of a key-value pair, and SequenceFileInputFormat reads data from these files by converting each key-value pair into a row in the Hive table.

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

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

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

72.When do we use explode in Hive?

A) if we have collection column such as array,map,list in a table that you want to split each element in a separate row then we can use the explode.

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

A)Hive is capable of processing various data formats, including structured, semi-structured, and unstructured data. Hive supports various file formats, such as Text, CSV, JSON, ORC, Parquet, Avro, Sequence, and RCFile. Hive can also process data stored in HBase and Cassandra databases.

The reason Hive can process different data formats is that it supports a variety of SerDe (Serializer/Deserializer) libraries. SerDe is a framework that enables Hive to read and write data in different formats. It helps to interpret the data's structure and convert it to the desired format. SerDe libraries are plug-ins that Hive uses to read and write data from various formats. Each SerDe library is designed to handle a specific data format, and Hive uses the appropriate SerDe library to process the data based on the file format.

For example, if we have a data file in JSON format, we can instruct Hive to use the "org.apache.hadoop.hive.serde2.JsonSerDe" SerDe library to interpret and process the data. Similarly, if we have a data file in ORC format, we can instruct Hive to use the "org.apache.hadoop.hive.ql.io.orc.OrcSerde" SerDe library to process the data.

In addition to the SerDe libraries, Hive also supports user-defined functions (UDFs) that enable users to write custom functions to process data. UDFs can be written in Java, Python, or any other programming language supported by Hive. Users can create their own UDFs to process the data in the desired format.

In conclusion, Hive is capable of processing various data formats due to its support for SerDe libraries and UDFs. SerDe libraries enable Hive to interpret and convert the data into the desired format, while UDFs allow users to write custom functions to process the data. Hive's ability to process various data formats makes it a powerful tool for batch processing and data analysis tasks.

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

A) The metastore\_db is created when you first set up Hive and it is used to store metadata information about Hive tables, such as their schema, location, and storage format.

When you run a Hive query, the metastore\_db is not created anew each time. Instead, Hive uses the existing metastore\_db to store metadata information about any new tables created by the query or any modifications made to existing tables.

It's possible that you are seeing multiple metastore\_db directories because you are running Hive queries on different clusters or environments, each of which has its own metastore\_db. It's also possible that you are seeing multiple directories because you are running multiple instances of the Hive metastore service, each of which has its own metastore\_db.

In any case, the metastore\_db is an important component of Hive that is used to store metadata information about tables. It is not created anew each time a query is run, but is instead used to store metadata for any new tables or modifications made to existing tables.

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

A)Yes, we can change the data type of a column in a Hive table. We can achieve this by using the ALTER TABLE statement in Hive. Here is a complete query to change the data type of a column in a Hive table:

ALTER TABLE <table\_name>

CHANGE COLUMN <column\_name> <new\_column\_name> <new\_data\_type>;

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

A)When using the LOAD DATA clause to load data into a Hive table, you can specify whether the source file is in the local file system or in the Hadoop Distributed File System (HDFS) by specifying the LOCATION parameter.

If the source file is in the HDFS, you should specify the HDFS path in the LOCATION parameter. For example, if your file is located in /user/hive/data/file.txt in HDFS, you can load it into a Hive table by running the following command:

LOAD DATA INPATH '/user/hive/data/file.txt' INTO TABLE my\_table;

Note that the INPATH parameter is used to specify the source file in HDFS.

If the source file is in the local file system, you should use the LOCAL keyword in the LOAD DATA command. For example, if your file is located in /home/user/data/file.txt in the local file system, you can load it into a Hive table by running the following command:

LOAD DATA LOCAL INPATH '/home/user/data/file.txt' INTO TABLE my\_table;

Note that the LOCAL keyword is used to indicate that the source file is in the local file system.

77.What is the precedence order in Hive configuration?

A) In Hive we can use following precedence order to set the configurable properties.

--> Hive SET command has the highest priority

--> -hiveconf option from Hive Command Line

--> hive-site.xml file

--> hive-default.xml file

--> hadoop-site.xml file

--> hadoop-default.xml file

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

A) WebHCat API web interface can be used for Hive commands. It is a REST API that allows applications to make HTTP requests to access the Hive metastore (HCatalog DDL).

It also enables users to create and queue Hive queries and commands.

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

A) Yes, it is possible to compress JSON data in a Hive external table. Hive provides various compression codecs that can be used to compress data in external tables, including Gzip, Snappy, and LZO, among others.

To compress JSON data in a Hive external table, you can use one of the compression codecs in the table definition. Here's an example of how to create an external table with JSON data compressed using Gzip:

CREATE EXTERNAL TABLE my\_table (

column1 string,

column2 string,

column3 int

)

ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'

WITH SERDEPROPERTIES (

'serialization.format' = '1'

)

STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat'

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

LOCATION '/path/to/my\_table'

TBLPROPERTIES (

'serialization.null.format'='',

'mapreduce.job.output.compress'='true',

'mapreduce.output.fileoutputformat.compress.type'='BLOCK',

'mapreduce.output.fileoutputformat.compress.codec'='org.apache.hadoop.io.compress.GzipCodec'

);

In this example, the table is created with a JSON serde, and the output is compressed using the Gzip codec. You can use a different compression codec by changing the mapreduce.output.fileoutputformat.compress.codec property in the table properties.

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

A)

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

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

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

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

82.What is DBPROPERTY in Hive?

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

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

A)Local mode is actually a local simulation of MapReduce in Hadoop's LocalJobRunner class.

MapReduce mode (also known as Hadoop mode): Pig is executed on the Hadoop cluster. In this case, the Pig Script gets converted into a series of MapReduce jobs that are then run on the Hadoop cluster.