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

Definition:
Hive is a data warehousing service built on top of hadoop.It provides an SQL like interface to user to query hadoop distributed file system.

Present Version:
3.1.3

2. Is Hive suitable to be used for OLTP systems? Why?
Hive is not suitable for OLTP systems, as hive is built for OLAP processing on large amounts of data,also it does not provide insert and update on row level.

3. How is HIVE different from RDBMS? Does hive support ACID transactions. If not then give the proper reason.

HIVE	RDBMS
It is a data warehousing service	It is a database system.
It is used for OLAP processing.	It is used for OLTP processing.
It is schema on Read only	It is a schema on read and write.

Higher versions of hive do support acid transactions on tables stored in ORC format.
However it does not support acid transactions on external tables.

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

Hive architecture consists of following components:
1.Hive clients
Thrift - Hive uses thrift to server the requests from clients.
JDBC and ODBC connectors:
To connect to hive programmatically we need JDBC or ODBC connectors.
CLI:
Hive also provides a command line interface to query data.

2.Hive Services
Driver - It acts as a controller for HQL statements.
Create sessions for HQL queries
Maintain life cycle for HQL.
Maintain metadata for execution.
Collects output and display.

Compile - Hive compiler parses the query.
It performs syntax and semantic checks.
Creates the execution plan

Optimizer:Compare execution plans
Calculate cost of execution
It puts transformations together

Execution engine:Execution engine after the compilation and optimization steps,executes the execution plan created by the compiler.

Hive Metastore-It is a repository which stores metadata of tables,about columns,data types, partitions.By default it uses derby db for metastore.

5.Mention what Hive query processor does? And Mention what are the components of a Hive query processor?
Hive query processor convert graph of MapReduce jobs with the execution time framework. So that the jobs can be executed in the order of dependencies.
The components of a Hive query processor are:
Parse and Semantic analysis
Optimizer
Plan Components
MetaData Layer
Map/Reduce Execution Engine
Sessions
Type interfaces
UDF's and UDAF's

6. What are the three different modes in which we can operate Hive?
Embedded Metastore Mode
Local Metastore Mode
Remote Metastore mode

7.Features and Limitations of Hive.
Features:
1.Can process large datasets of petabytes in size.
2.HQL is supported by hive so knowledge of SQL is enough to query data.
3.Supports multiple file formats like ORC,Text,Parquet,Avro etc.
4.Supports partitioning and bucketing which improves query performance.
5.It is schema on read so less time for loading data.

Limitations:
1. Hive is not designed for the OLTP (Online transaction processing). We can use it for OLAP.
2. It does not offer real-time queries.
3. It provides limited subquery support.
4. Latency of Hive is generally very high.

8. How to create a Database in HIVE?
   Create database database_name;

   Eg: create database Employees;

9.How to create a table in hive?

For Internal table:

```
Create table Employee(
 Id  int,
 Dept_id int,
 Salary float )
Row Format Delimited
Fields terminated by ‘,’
Stored as ORC;
```

For External Table:

```
Create External table Employee(
  Id  int,
   Dept_id int,
   Salary float )
Row Format Delimited
Fields terminated by ‘,’
Stored as Textfile
Location ‘home/cloudera/’;
```

10.What do you mean by describe and describe extended and describe formatted with respect to database and table

For Database:

Describe database shows the database name,its root location on hdfs and comment.

Describe extended will list the properties of database in hive along with database name,its location and comment

For Table:

Describe shows the columns of the table along with data types.

describe extended command will show the detailed information of the table such as list of columns , data type of the columns,table type,location of the table,table size and so on.

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

To skip header row use tblproperties("skip.header.line.count" = "1") while creating table

```
Create table Employee(
  Id  int,
  Dept_id int,
  Salary float )
Row Format Delimited
Fields terminated by ‘,’
Stored as ORC
tblproperties("skip.header.line.count" = "1") ;
```

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

Hive Operators allow to perform various arithmetic and relational operations

Different types of operators:
Relational Operators - =,<,>,<=,>=,!=
Arithmetic Operators - +,-,*,/,%
Logical Operators- and,or,not
Complex Operators-array[n],map[key]

13.Explain about the Hive Built-In Functions

Mathematical Functions:

round(num):It returns the BIGINT for the rounded value of DOUBLE num.
floor(num):It returns the largest BIGINT that is less than or equal to num.
ceil(num), ceiling(DOUBLE num):It returns the smallest BIGINT that is greater than or equal to num.
exp(num):It returns exponential of num.
ln(num):It returns the natural logarithm of num.
og10(num):It returns the base-10 logarithm of num.
sqrt(num):It returns the square root of num.
abs(num):It returns the absolute value of num.
sin(d):It returns the sin of num, in radians.
asin(d):It returns the arcsin of num, in radians.
cos(d):It returns the cosine of num, in radians.

acos(d):It returns the arccosine of num, in radians.
tan(d):It returns the tangent of num, in radians.

Aggregate Functions:
count(*):It returns the count of the number of rows present in the file.
sum(col):It returns the sum of values.
sum(DISTINCT col):It returns the sum of distinct values.
avg(col):It returns the average of values.
avg(DISTINCT col):It returns the average of distinct values.
min(col):It compares the values and returns the minimum one from it.
max(col):It compares the values and returns the maximum one from it.

String Functions:
length(str):It returns the length of the string.
reverse(str):It returns the string in reverse order.
concat(str1, str2, ...):It returns the concatenation of two or more strings.
substr(str, start_index):It returns the substring from the string based on the provided starting index.
substr(str, int start, int length):It returns the substring from the string based on the provided starting index and length.
upper(str):It returns the string in uppercase.
lower(str):It returns the string in lowercase.
trim(str):It returns the string by removing whitespaces from both the ends.
ltrim(str):It returns the string by removing whitespaces from left-hand side.
rtrim(str):It returns the string by removing whitespaces from right-hand side.

14. Write hive DDL and DML commands.
DDL:

```
Create table Employee(
 Id  int,
 Dept_id int,
 Salary float )
Row Format Delimited
Fields terminated by ‘,’
Stored as ORC;
```

DML:
Load:
For loading data from local file system:
Load data local inpath ‘file:///file_path’ into table table_name.
For loading data from hdfs:
Load data inpath filepath into table table_name

Insert:
Insert into table_name select from statement;

Insert overwrite:
For overwriting existing data:
Insert into table_name select from statement;

Delete:
DELETE FROM tablename [WHERE expression];

Truncate:
Truncate table tablename

15.Explain about SORT BY, ORDER BY, DISTRIBUTE BY and CLUSTER BY in Hive.
Sort By:
The SORT by clause sorts the data per reducer. So if we have N reducers we have N files in the output and the final output is not globally sorted.

Order by:
The order by clause sorts data globally as it ensures all data need to be passed to a single reducer only.So order by outputs a single file only.

Distribute By:
DISTRIBUTE BY clause is used to distribute the input rows among reducers.It does not sort the data.If we need to partition the data on some key column we can use Distribute By.

Cluster By:
CLUSTER BY clause is a combination of DISTRIBUTE BY and SORT BY clauses. So 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 per reducer.

16.Difference between "Internal Table" and "External Table" and Mention when to choose “Internal Table” and “External Table” in Hive

Internal Table	External Table
Hive owns the data of internal tables.	Hive does not manage data on external table
internal table will be created in a folder path similar to **/user/hive/warehouse/db/** directory of HDFS	External tables are stored outside the warehouse directory. They can access data stored in sources such as remote HDFS locations or Azure Storage Volumes.
If we drop the internal table the table data and the metadata associated with that table will be deleted from the HDFS.	Whenever we drop the external table, then only the metadata associated with the table will get deleted, the table data remains.

We can use the internal table in cases:

- When generating temporary tables.
- When required that Hive should manage the lifecycle of the table.
- And when we don't want table data after deletion.

We can use the external table in cases:

- When we are not creating the table based on the existing table.
- When required to use data outside of Hive. For example, the data files are read and processed by an existing program that does not lock the files.
- When we don't want to delete the table data completely even after DROP.
- When the data should not be owned by Hive.

17.Where does the data of a Hive table get stored?
Hive table data is stored in hdfs path **user/hive/warehouse/db/**

18.Is it possible to change the default location of a managed table?
Yes
To change default location we need to specify location using location construct

```
Create table employee(
 Id int,
 Salary int
)
Location '/user/hive/warehouse/employeedetails/'
```

19.What is a metastore in Hive? What is the default database provided by Apache Hive for metastore?
Hive metastore stores metadata of tables,about columns,data types,partitions.By default it uses derby db for metastore.

20.Why does Hive not store metadata information in HDFS?
Hive stores metadata information in the metastore using RDBMS instead of HDFS, 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
The partitioning in Hive means dividing the table into 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 divided in partitions, the query response time becomes faster.

22.What is the difference between dynamic partitioning and static partitioning?

Static Partitioning	Dynamic Partitioning
We need to manually create each partition before inserting data into a partition	In dynamic partitioning hive creates partitions automatically based on the values in partition column.
We need to know all partitions in advance. So it is suitable for use cases where partitions are defined well ahead and are small in number	Dynamic partitions are suitable when we have lot of partitions and we can not predict in advance new partitions ahead of time
Loading data is fast in static partitioning	Loading is slower than static partitioning

23.How do you check if a particular partition exists?
Show partitions table_name;

24.How can you stop a partition from being queried?
To stop a partition from being queried set the partition as offline.
Alter table table set partition partition(PARTITION_SPEC) enable offline.

25.Why do we need buckets? How Hive distributes the rows into buckets?
We need buckets:
When partitioning a table is not effective
When there are several map side joins in queries then bucketing improves the performance
Of query.

Hive distributes the rows into buckets based on hash value of a column generated by hashing Algorithm.

26.In Hive, how can you enable buckets?
set hive.enforce.bucketing = true;

27.How does bucketing help in the faster execution of queries?
Bucketed tables allow faster execution of map side joins, as data is stored in equal-sized buckets. Also, efficient sampling happens for bucketed tables when compared to non-bucketed ones. It also improves performance by shuffling and sorting data prior to downstream operations such as table joins.

28.How to optimize Hive Performance? Explain in very detail.
Use of Suitable file format:
if we use appropriate file format on the basis of data. It will drastically increase our query performance.ORC file format gives best performance in hive.It compresses the data so data processed is less

Hive Partitioning:
Partitioning divides the table into multiple parts based on values in the columns.
So when querying only the required partitions are queried.

Hive Bucketing:
When after partitioning also the size of file is quite large then bucketing is useful.
it divides the data into more manageable parts which improves the query performance.

Vectorization in Hive:
To improve the performance of operations such as scans, aggregations, filters, and joins. It happens by performing them in batches of 1024 rows at once instead of single row each time.

Hive Indexing:
for the original table use of indexing will create a separate called index table which acts as a reference.
it will take a large amount of time if we want to perform queries only on some columns without indexing. Because queries will be executed on all the columns present in the table.
There is no need for the query to scan all the rows in the table while we perform a query on a table that has an index

29. What is the use of Hcatalog?
HCatalog is a table storage management tool for Hadoop that exposes the tabular data of Hive metastore to other Hadoop applications.
For example:
Suppose I want to load a table from hive into pig for some transformations it can be done using Hcatalog.

30. Explain about the different types of join in Hive
Inner Join:
Hive inner join is used to return the rows of multiple tables where the join condition satisfies.

Left Outer Join:
Left outer join returns all the rows from left table and only matching rows from right table.
It assigns nulls for the columns of right table where join condition is not satisfied

Right Outer Join:
Right Outer join returns all the row from right table and only matching row from left table.
It assigns nulls for the columns of left table where join condition is not satisfied

Full Outer Join:
full outer join returns all the records from both the tables. It assigns Null for missing records in either table.

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

32.Explain the SMB Join in Hive?
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.
in Mapper, only Join is done. all the buckets are joined with each other at the mapper which are corresponding.

33.What is the difference between order by and sort by which one we should use?
Order by:
Order by orders the entire data.It passes all the data to a single reducer.When dataset is not that large the we can go with order by.

Sort by:
Sort by orders the data per reducer so the entire is not sorted.
When dataset is huge we can go with sort by.

34.What is the usefulness of the DISTRIBUTED BY clause in Hive?
Hive uses the columns in Distribute By to distribute the rows among reducers. All rows with the same Distribute By columns will go to the same reducer. However, Distribute By does not guarantee clustering or sorting properties on the distributed keys.

35.How does data transfer happen from HDFS to Hive?
There is no data transfer from HDFS to hive.In case of an internal table, you load the data.
It means you tell HIVE where you source file is in HDFS and HIVE moves that file from that HDFS location to its own warehouse directory which is also present in HDFS.
In case of external table hive points to the directory of the file and uses the schema to query Files.

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

37.What will happen in case you have not issued the command: ‘SET hive.enforce.bucketing=true;’ before bucketing a table in Hive?
The command’ allows one to have the correct number of reducers while using ‘CLUSTER BY’ clause for bucketing a column. If it is not set to true then there may be number of files that will be generated in the table directory to be not equal to the number of buckets.

38.Can a table be renamed in Hive?
Yew we can use alter command to rename the table in hive.
Alter table table_name rename to new
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)

Alter table table_name add columns (new_col int);
Alter table table_name change column new_col new_col before x_col;

40.What is serde operation in HIVE?
SerDe means Serializer and Deserializer. Hive uses SerDe and FileFormat to read and write table rows. Main use of SerDe interface is for IO operations. A SerDe allows hive to read the data from the table and write it back to the HDFS in any custom format.

41.Explain how Hive Deserializes and Serializes the data?
Serializer serialization: the process of converting an object into a sequence of bytes
Deserializer deserialization: it is the process of converting byte sequence into object

The process of serialization and deserialization is as follows:

Serializer serialization: Row object - > serialization - > outputfileformat - > HDFS file

Deserializer deserialization: HDFS file - > inputfileformat - > deserialization - > Row obje

42.Write the name of the built-in serde in hive.
Avro,ORC,regex,thrift,Parquet,CSV,JSON are built in serdes in hive.

43.What is the need of custom Serde?
Hive custom Serde is used to read data in any custom format.
Suppose you need to query a xml file in hive then you have to write a custom serde.

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

45.Can hive queries be executed from script files? How?
Yes
Hive> source /path/to/file/file_with_query.hql

46.What are the default record and field delimiter used for hive text files?
The default record delimiter is - \n
And the field delimiters are - \001,\002,\003

47.How do you list all databases in Hive whose name starts with s?
Show databases like ‘s*’;

48.What is the difference between LIKE and RLIKE operators in Hive?
Like -
Like is used for pattern matching in the string.It always matches the entire string.

Rlike-RLIKE function is an advanced version of LIKE operator in Hive. It is used to search the advanced Regular expression pattern on the columns. If the given pattern matches with any substring of the column, the function returns TRUE. otherwise FALSE.

49.How to change the column data type in Hive?
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?
Select cast(‘51.2’ as FLOAT)

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

52.What does the following query do? a. INSERT OVERWRITE TABLE employees b. PARTITION (country, state) c. SELECT ..., se.cnty, se.st d. FROM staged_employees se
It creates dynamic partition on table employees with partition values coming from the columns in the select clause. It is insert into the dynamic partitions.

53.Write a query where you can overwrite data in a new table from the existing table
From existing_table_name insert overwrite table new_table_name;

54.What is the maximum size of a string data type supported by Hive? Explain how Hive supports binary formats.
Maximum size of a string data type is 2GB.
BINARY is an array of Bytes and similar to VARBINARY in many RDBMSs. BINARY columns are stored within the record, not separately like BLOBs.

55. What File Formats and Applications Does Hive Support?
Supported File Formats: Textfile,ORC,Parquet,Avro,RC
Applications-PHP,Python,Java,CPP,Ruby.

56.How do ORC format tables help Hive to enhance its performance?
The ORC file format provides the following advantages:

- **Efficient compression**: Stored as columns and compressed, which leads to smaller disk reads. The columnar format is also ideal for vectorization optimizations in Tez.
- **Fast reads**: ORC has a built-in index, min/max values, and other aggregates that cause entire stripes to be skipped during reads. In addition, predicate pushdown pushes filters into reads so that minimal rows are read. And Bloom filters further reduce the number of rows that are returned.
- **Proven in large-scale deployments**: Facebook uses the ORC file format for a 300+ PB deployment.

57.How can Hive avoid mapreduce while processing the query?
If the property hive.exec.mode.local.auto is set to true then hive will avoid mapreduce to fetch query results.

58.What is view and indexing in hive?

Views:
Views are similar to tables, which are generated based on the requirements.
We can save any result set data as a view in Hive
Usage is similar to as views used in SQL
All type of DML operations can be performed on a view

Create view view_name as select statement;

Indexing:
The goal of Hive indexing is to improve the speed of query lookup on certain columns of a table. Without an index, queries with predicates like 'WHERE tab1.col1 = 10' load the entire table or partition and process all the rows. But if an index exists for col1, then only a portion of the file needs to be loaded and processed.
Create INDEX index_name ON TABLE tablename(column-name);

59.Can the name of a view be the same as the name of a hive table?
No,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?
Indexes occupies space and there is a processing cost in arranging the values of the column on which index is created.

61.Give the command to see the indexes on a table.
SHOW INDEX ON table_name

62. Explain the process to access subdirectories recursively in Hive queries.
To process directories recursively in Hive, we need to set below two commands in hive session. These two parameters work in conjunction.

```
Shell
hive> Set mapred.input.dir.recursive=true;
hive> Set hive.mapred.supports.subdirectories=true;
```

Now hive tables can be pointed to the higher level directory.

63.If you run a select * query in Hive, why doesn't it run MapReduce?
When we run a "select * from <tablename>", Hive fetches the whole data from file as a FetchTask rather than a mapreduce task which just dumps the data as it is without doing anything on it.

64.What are the uses of Hive Explode?
Explode is a User Defined Table generating Function(UDTF) in Hive. It takes an array or a map as an input and outputs the elements of the array or a map as separate rows.

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

- ODBC Driver-This supports the ODBC protocol
- JDBC Driver- This supports the JDBC protocol
- Thrift Client- This client can be used to make calls to all hive commands using different programming language like PHP, Python, Java, C++ and Ruby.

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

```
Create table table_name(
Col1 datatype,
Col2 datatype
..
)
Row format delimited
Fields terminated by ‘,’
Location ‘/new_file_path/’;
```

67.What is the Hive ObjectInspector function?

In generic UDFs, all objects are passed around using the Object type. Hive is structured this way so that all code handling records and cells is generic, and to avoid the costs of instantiating and deserializing objects when it's not needed.

Therefore, all interaction with the data passed in to UDFs is done via ObjectInspectors. They allow you to read values from an UDF parameter, and to write output values.

Object Inspectors belong to one of the following categories:

- Primitive, for primitive types (all numerical types, string, boolean, …)
- List, for Hive arrays
- Map, for Hive maps
- Struct, for Hive structs

When Hive analyses the query, it computes the actual types of the parameters passed in to the UDF, and calls

public ObjectInspector initialize(ObjectInspector[] args) throws UDFArgumentException;

The method receives one object inspector for each of the arguments of the query, and must return an object inspector for the return type. Hive then uses the returned ObjectInspector to know what the UDF returns and to continue analyzing the query.

68.What is UDF in Hive?
User Defined Functions, also known as UDF, we can create custom functions to process records or groups of records when Hive built in functions do not satisfy our requirements.

69.Write a query to extract data from hdfs to hive.
Load data inpath '/hdfs_file_path' into table tablename.

70.What is TextInputFormat and SequenceFileInputFormat in hive?
TextInputFormat :
This is very familiar input format in the Hadoop. The input will be given as key and value to Mapper, where key and value are generated in record reader. The record reader is just like a multiplexing.
For TextInputFormat, No need to create external record reader.
Key generated-----LongWritable(position ,generally "\n" or offset)
Value generated-----Text of each line.
The important point in the TextInputFormat is
a) Number of maps created is equal to number of files given in input path.
b) For each line, map method in mapper will be called.

SequenceFileInputFormat:
The sequence file is the file has lot of importnace in hadoop.It has equal importance as "AVRO file" .Hadoop is the technology best suited for file with huge size,than many number of files with small size.So in this case,One of the possible solution to process many number of files with

small size is "Merging of files".So we use sequence file (Binary key/value) to merge large number of small files as file name as key and total content as value.
By mentioning SequenceFileOutputFormat in mapreduce we can get sequence file ,Here key is file name and value is content of file.Inorder to use,SequenceFileInputFormat we must need sequencefile in hdfs.Otherwise we encounter with exception.so now we have key and values in sequencefile in hdfs and later sequencefileinputformat behaves exactly same as textinputformat.Map method will be called for every key/value in sequnce file same as map methods will be called for everyline in textflie in textinputformat.

71.How can you prevent a large job from running for a long time in a hive?
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?
When we need to convert array or map elements into separate rows then we use explode.

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

There are some specific file formats which Hive can handle such as:

• TEXTFILE

• SEQUENCEFILE

• RCFILE

• ORCFILE

TEXTFILE

TEXTFILE format is a famous input/output format used in Hadoop. In Hive if we define a table as TEXTFILE it can load data of from CSV (Comma Separated Values), delimited by Tabs, Spaces, and JSON data. This means fields in each record should be separated by comma or space or tab or it may be JSON(JavaScript Object Notation) data.By default, if we use TEXTFILE format then each line is considered as a record.The TEXTFILE input and TEXTFILE output format are present in the Hadoop package as shown below:

org.apache.hadoop.mapred.TextInputFormat

org.apache.hadoop.mapred.TextOutputFormat

SEQUENCEFILE

We know that Hadoop's performance is drawn out when we work with a small number of files with big size rather than a large number of files with small size. If the size of a file is smaller than the typical block size in Hadoop, we consider it as a small file. Due to this, a number of metadata increases which will become an overhead to the NameNode. To solve this problem sequence files are introduced in Hadoop. Sequence files act as a container to store the small files.

Sequence files are flat files consisting of binary key-value pairs. When Hive converts queries to MapReduce jobs, it decides on the appropriate key-value pairs to be used for a given record. Sequence files are in the binary format which can be split and the main use of these files is to club two or more smaller files and make them as a one sequence file.

In Hive we can create a sequence file by specifying STORED AS SEQUENCEFILE in the end of a CREATE TABLE statement.

There are three types of sequence files:

• Uncompressed key/value records.

• Record compressed key/value records - only 'values' are compressed here

• Block compressed key/value records - both keys and values are collected in 'blocks' separately and compressed. The size of the 'block' is configurable.

Hive has its own SEQUENCEFILE reader and SEQUENCEFILE writer libraries for reading and writing through sequence files.

In Hive we can create a sequence file format as follows:

create table table_name (schema of the table) row format delimited fileds terminated by ',' | stored as SEQUENCEFILE

Hive uses the SEQUENCEFILE input and output formats from the following packages:

org.apache.hadoop.mapred.SequenceFileInputFormat
org.apache.hadoop.hive.ql.io.HiveSequenceFileOutputFormat

RCFILE

RCFILE stands of Record Columnar File which is another type of binary file format which offers high compression rate on the top of the rows.RCFILE is used when we want to perform operations on multiple rows at a time.

RCFILEs are flat files consisting of binary key/value pairs, which shares many similarities with SEQUENCEFILE. RCFILE stores columns of a table in form of record in a columnar manner. It first partitions rows horizontally into row splits and then it vertically partitions each row split in a columnar way. RCFILE first stores the metadata of a row split, as the key part of a record, and

all the data of a row split as the value part. This means that RCFILE encourages column oriented storage rather than row oriented storage.

This column oriented storage is very useful while performing analytics. It is easy to perform analytics when we “hive’ a column oriented storage type.

Facebook uses RCFILE as its default file format for storing of data in their data warehouse as they perform different types of analytics using Hive.

In Hive we can create a RCFILE format as follows:

create table table_name (schema of the table) row format delimited fields terminated by ',' | stored as RCFILE

Hive has its own RCFILE Input format and RCFILE output format in its default package:

org.apache.hadoop.hive.ql.io.RCFileInputFormat
org.apache.hadoop.hive.ql.io.RCFileOutputFormat

ORCFILE

ORC stands for Optimized Row Columnar which means it can store data in an optimized way than the other file formats. ORC reduces the size of the original data up to 75%(eg: 100GB file will become 25GB). As a result the speed of data processing also increases. ORC shows better performance than Text, Sequence and RC file formats.

An ORC file contains rows data in groups called as Stripes along with a file footer. ORC format improves the performance when Hive is processing the data.

In Hive we can create a RCFILE format as follows:

create table table_name (schema of the table) row format delimited fields terminated by ',' | stored as ORC

Hive has its own ORCFILE Input format and ORCFILE output format in its default package:

Org.apache.hadoop.hive.ql.io.orc

74.Whenever we run a Hive query, a new metastore_db is created. Why?
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.

75.Can we change the data type of a column in a hive table? Write a complete query
Yew we can change data type of column in hive table
ALTER TABLE table_name CHANGE column_name column_name new_datatype;

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?

When loading local file if we use local switch then it specifies we are loading data from local file
eg:Load data local inpath 'file:///file_path/' into table table_name

If we do not use local switch in load command then hive understands that we are loading data From hdfs
eg:Load data inpath '/file_path/' into table table_name

77.What is the precedence order in Hive configuration?

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

78.Which interface is used for accessing the Hive metastore?
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 ?
Yew we need to gzip json files and put them as is (*.gz) into the table location.

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

In local metastore configuration, the metastore service runs in the same JVM in which the Hive service is running and connects to a database running in a separate JVM, either on the same machine or on a remote machine.

Remote Metastore:

In the remote metastore configuration, the metastore service runs on its own separate JVM and not in the Hive service JVM. Other processes communicate with the metastore server using Thrift Network APIs. You can have one or more metastore servers in this case to provide more availability.

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

HDFS is designed such that the number of files affects the memory consumption in the namenode.Memory usage may hit the limits of accessible memory on a machine when there millions of files.

The use of Hadoop Archives is one approach to reducing the number of files in partitions. Hive has built-in support to convert files in existing partitions to a Hadoop Archive (HAR) so that a partitionthat may once have consisted of 100's of files can occupy just ~3 files (depending on settings).
The queries may be slower due to the additional overhead in reading from the HAR.

82.What is DBPROPERTY in Hive?
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 it easy for the other user to recognize the database and use it according to the requirement.

```
CREATE DATABASE IF NOT EXISTS db_name
WITH DBPROPERTIES(
'Date' = '2022-09-16',
'Creator' = ‘Prajwal',
'Email' = 'prajwal@test.com'
);
```

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

MapReduce mode:

In MapReduce mode, Hive script is executed on Hadoop cluster. The Hive scripts are converted into MapReduce jobs and then executed on Hadoop cluster (hdfs).By default hive runs in mapreduce mode.

- If Hadoop is having multiple data nodes and data is distributed across different node we use Hive in this mode
- It will perform on large amount of data sets and query going to execute in parallel way
- Processing of large data sets with better performance can be achieved through this mode

Local Mode:

In this mode, Hive script runs on a Single machine without the need of Hadoop cluster or hdfs. Local mode is used for development purpose to see how the script would behave in an actual Environment.

- If the Hadoop installed under pseudo mode with having one data node we use Hive in this mode
- If the data size is smaller in term of limited to single local machine, we can use this mode
- Processing will be very fast on smaller data sets present in the local machine

To set hive in local mode:
SET mapred.job.tracker=local;