Interview Questions for HIVE

**Difference between Hive and Hbase?**

**Hive** is query engine that whereas **HBase** is a data storage particularly for unstructured data. Apache **Hive** is mainly used for batch processing i.e. ... Unlike **Hive**, operations in **HBase** are run in real-time on the database instead of transforming into mapreduce jobs

**How to print column name of a table in hive query result?**

set hive.cli.print.header=true

What is HCatalog?

What is WebHCat?

7. How do we import JSON data into Hive?

**250)What is bucketing in Hive?**

**(251)What is Clustring in Hive?**

**(287)How to create table in hive for a json input file.**

**(290)How to resolve the following error while running a query in hive: Error in metadata: Cannot validate serde**

**ADD JAR /<local-path>/json-serde-1.0.jar;**

**(292)What is Bucketing and Clustering in Hive?**

1. What is your Pig/Hive programming level (1- 10)? (Almost all interviewers asked this.)
2. Hive – What type of data stored?
3. **(234)what is ODBC and JDBC connectivity in Hive?**
4. **(235)What is Derby database?**
5. meta store configured for hive
6. **(232)Hive for Hadoop - Give some points?**

1. Does hive support record level operations?

* Hive doesn't support record level update, insert, and deletion operations on table, but Hbase can do it. Hive is a Data warehouse framework where as Hbase is a NoSQL database.

2. In hive table can we change string DT to Int DT?

* cast(str\_column as int)

Converts the results of the expression expr to <type>. For example, cast('1' as BIGINT) will convert the string '1' to its integral representation. A null is returned if the conversion does not succeed. If cast(expr as boolean) Hive returns true for a non-empty string.

3. Can we rename a Table in Hive? if Yes, How?

* Yeah we can do that. You just need to follow below three commands in sequence.
* Lets say you have a external table test\_1 in hive. And you want to rename it test\_2 which should point test\_2 location not test\_1. Then you need to convert this table into Managed table using below command. test\_1 -> pointing to test\_1 location
* ALTER TABLE db\_name.test\_1 SET TBLPROPERTIES('EXTERNAL'='FALSE');
* Rename the table name.
* ALTER TABLE db\_name.test\_1 RENAME TO db\_name.test\_2;
* Again convert the managed table after renaming to external table.
* ALTER TABLE db\_name.test\_2 SET TBLPROPERTIES('EXTERNAL'='TRUE');
* db\_name.test\_2 table will point the test\_2 location. If we do it without making the managed table it will point the test\_1 location.

4. What is metastore? how will you start the service?

* Metastore is the central repository of Apache Hive metadata. It stores metadata for Hive tables (like their schema and location) and partitions in a relational database. It provides client access to this information by using metastore service API.
* To start HiveServer:
* $ sudo service hive-server2 start
* To stop HiveServer:
* $ sudo service hive-server2 stop

5. How do we import XML data into Hive?

Steps: Example to load xml file

1. add jar hivexmlserde-1.0.5.3jar

2.create table (Include all sub fields in xml file)

row format serde 'com.ibm.spss.hive.serde2.xml.xmlserde'

WITH SERDE PROPERTIES

("column.xpath.subfield"="/main\_field/sub\_field/text();

for all fields)

stored as

inputformat 'com.ibm.spss.hive.serde2.xml.xmlInputFormat'

outputformat

'org.apache.hadoop.hive.ql.io.IgnorekeyTextoutputFormat'

TBLPROPERTIES(

"xmlinput.start"="Main\_field",

"xmlinput.end"="/Main\_field"

);

3. Then load the data on to table using .xml file

6. How do we import CSV data into Hive?

1.create table

row format delimited fields terminated by ','

2. load data [local] inpath 'path of the file' into table table\_name

7 What are dynamic partitions?

partitions are nothing but splitting of the data into separate directories based on a particular column unique values

there are two types of partitions

1. dynamic

2. static(default)

Steps to do dynamic partition

1. set hive.exec.partition=true;

set hive.exec.dynamic.partition.mod=nonstrict;

2.create external table (fields ) partitoned by (column)

row format -------------------

location----------------------

3.insert overwrite table table\_name partition(column)

select all fields last partition field from managed table

distributed by (column);

3. fire select query with where clause

8 Does Hive have an ODBC driver?

> Yes cloudera provides ODBC drivers for Hiveserver

9.Is HiveQL case sensitive?

>no

10. Does Hive support Unicode?

A) Yes we can use Unicode string on data/comments, but cannot use for database/table/column

name.

11. Does Hive have a JDBC Driver?

A) Yes, The driver is ‘org.apache.hadoop.hive.jdbc.HiveDriver’.

It supports two modes: a local mode and a remote one.

In the remote mode it connects to the hive server through its Thrift API. The JDBC url to use should be of the form: ‘jdbc:hive://hostname:port/databasename’. In the local mode Hive is embedded. The JDBC url to use should be ‘jdbc:hive://’.

12.When running a JOIN query, what is the idea to solve out-of-memory errors.

A)This is usually caused by the order of JOIN tables. Instead of “FROM tableA a JOIN tableB b ON …”, try “FROM tableB b JOIN tableA a ON …”. NOTE that if we are using LEFT OUTER JOIN, we might want to change to RIGHT OUTER JOIN. This trick usually solve the problem – the rule of thumb is, always put the table with a lot of rows having the same value in the join key

on the rightmost side of the JOIN.

13. Can a Hive table contain data in more than one format?

* Yes, We can achieve this through a combination of different external tables.
* Because, different SerDes with different specifications for how to read columns in a different files will be needed, you will need to create one external table per type of file(and table). The data from each of these external tables can then be combined into a view with UNION. The view can could then be used for reading from these,and you could e.g. insert the data into a managed table.

14. How do we import JSON data into Hive?

* You can import JSON into Hive by implementing the HiveSerDe.
* This link serves as a sample implementation.

<https://github.com/rcongiu/Hive-JSON-Serde>

15. How do I import Avro data into Hive?

* If you follow the hive convention in your hdfs folder and create the hive table pointing to the table location, you should run the msck repair table statement.

For example

**CREATE TABLE table\_name**

**PARTITIONED BY (t string, y string, m string, d string, h string, hh string)**

**ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'**

**STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'**

**OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat'**

**TBLPROPERTIES (**

**'avro.schema.url'='hdfs://location/schema/schema.avsc')**

**location "hdfs:///location/data;**

**and load the data like**

/location/data/y=2016/m=02/d=03/h=03/hh=12/data.avro /location/data/y=2016/m=02/d=03/h=03/hh=13/data2.avro

in that way you will be able to load the data with the following statement because Hive will recognize the partitions

msck repair table table\_name;

You can also follow the link to check the example for this:

<https://www.cloudera.com/documentation/enterprise/5-6-x/topics/cdh_ig_hive.html>

16.How many times Tez engine runs faster than MR engine in Hive?

* Tez is a DAG (Directed acyclic graph) architecture. A typical Map reduce job has following steps:

1. Read data from file -->one disk access

2. Run mappers

3. Write map output --> second disk access

4. Run shuffle and sort --> read map output, third disk access

5. write shuffle and sort --> write sorted data for reducers --> fourth disk access

6. Run reducers which reads sorted data --> fifth disk output

7. Write reducers output -->sixth disk access

Tez works very similar to Spark (Tez was created by Hortonworks well before Spark):

1. Execute the plan but no need to read data from disk.

2. Once ready to do some calculations (similar to actions in spark), get the data from disk and perform all steps and produce output.

Only one read and one write.

Notice the efficiency introduced by not going to disk multiple times. Intermediate results are stored in memory (not written to disks). On top of that there is vectorization (process batch of rows instead of one row at a time). All this adds to efficiencies in query time.

Now to answer your question on why Tez queries fail but executed in MR. This should not happen. Possible bugs or sometimes people working with Hive have used MapReduce for a while and know how to make things work but not as familiar with Tez. I think, Tez queries should not fail any more than Map Reduce.

17.How much time each Tez session will be active?

* Consider a set up where you have started the hivecli with an engine such as TEZ. For each hivecli session there will be an associated AM(application master) created with job name same as hive session number. This session holds good (is alive) until the there is inactivity on that session which later leads to timeout causing the TEZ session to terminate. AM job related to the same TEZ session will be complete.

If a user is actively running the queries below in the session, then the TEZ session will continue to be active in the same AM.

1)Start the hivecli and set the execution engine as tez ( set hive.execution.engine=tez)

2)Set the tez queue (set [tez.queue.name](http://tez.queue.name/)=root.mapr)

3)Run the hive query hive>select count(\*) from test;

**Some more difference in short about Tez and MR which will help you to understand deeply**

Tez simplifies processing for both small scale (low-latency) and large-scale (high throughput) workloads. The more complex query is the more benefit from TEZ. For simple queries consisting of single map step it will be most probably no difference at all because there is nothing to optimize. TEZ represents query as a DAG (directed acyclic graph) for a single job and eliminates unnecessary steps like read/write to durable storage, sort of the output from each Map, also enables containers reuse. Tez is always the best choice, for simple queries it will work not worse than MR and much better for complex queries. And consider this: For MR and for TEZ you have to tune different sets of configuration parameters, there are a lot of TEZ-specific and a lot of MR-specific. Choose TEZ and you will simplify you life even in cases when there is nothing to optimize.

1. 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:  
  
    The Hive SET command  
    The command line -hiveconf option  
    hive-site.xml  
    hive-default.xml  
    hadoop-site.xml (or, equivalently, core-site.xml, hdfs-site.xml, and mapred-site.xml)  
    hadoop-default.xml (or, equivalently, core-default.xml, hdfs-default.xml, and mapred-default.xml)  
  
2. How do change settings within Hive Session?  
We can change settings from within a session, too, using the SET command. This is useful for changing Hive or MapReduce job settings for a particular query. For example, the following command ensures buckets are populated according to the table definition.  
hive> SET hive.enforce.bucketing=true;  
   
hive> SET hive.enforce.bucketing=true;  
  
To see the current value of any property, use SET with just the property name:  
hive> SET hive.enforce.bucketing;  
hive.enforce.bucketing=true  
   
hive> SET hive.enforce.bucketing;  
hive.enforce.bucketing=true  
  
By itself, SET will list all the properties and their values set by Hive. This list will not include Hadoop defaults, unless they have been explicitly overridden in one of the ways covered in the above answer. Use SET -v to list all the properties in the system, including Hadoop defaults.  
3. How to print header on Hive query results?  
  
We need to use following set command before our query to show column headers in STDOUT.  
hive> set hive.cli.print.header=true;  
  
hive> set hive.cli.print.header=true;  
  
4. How to get detailed description of a table in Hive?  
  
Use below hive command to get a detailed description of a hive table.  
hive> describe extended <tablename>;

difference between describe and descrine extended

5. How to access sub directories 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.

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. This is suitable for a scenario where the directory structure is as following: /data/country/state/city

6. How to skip header rows from a table in Hive?  
  
Suppose while processing some log files, we may find header records.  
  
System=….  
Version=…  
Sub-version=….  
  
Like above, It may have 3 lines of headers that we do not want to include in our Hive query. To skip header lines from our tables in Hive we can set a table property that will allow us to skip the header lines.  
CREATE EXTERNAL TABLE userdata (  
name STRING,  
job STRING,  
dob STRING,  
id INT,  
salary INT)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘ ‘ STORED AS TEXTFILE  
LOCATION ‘/user/data’  
TBLPROPERTIES("skip.header.line.count"="3");  
   
CREATE EXTERNAL TABLE userdata (  
name STRING,  
job STRING,  
dob STRING,  
id INT,  
salary INT)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘ ‘ STORED AS TEXTFILE  
LOCATION ‘/user/data’  
TBLPROPERTIES("skip.header.line.count"="3");  
  
7. 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 querying, if we specify any column other than above list, will result in NULL values.

8. What is the maximum size of string data type supported by Hive?  
  
Maximum size is 2 GB.

9. What are the Binary Storage formats supported in Hive? 

By default Hive supports text file format, however hive also supports below binary formats.  
  
Sequence Files, Avro Data files, RCFiles, ORC files, Parquet files  
  
Sequence files: General binary format. splittable, compressible and row oriented. a typical example can be. if we have lots of small file, we may use sequence 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 Sequence 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 subsequently.  
  
ORC Files: Optimized Record Columnar files

10. is HQL case sensitive?  
  
HQL is not case sensitive.

11. Describe CONCAT function in Hive with Example?  
  
CONCAT function will concatenate the input strings. We can specify any number of strings separated by comma.  
  
Example: CONCAT (‘Hive’,’-‘,’is’,’-‘,’a’,’-‘,’data warehouse’,’-‘,’in Hadoop’);  
Output: Hive-is-a-data warehouse-in Hadoop  
  
So, every time we delimit the strings by ‘-‘. If it is common for all the strings, then Hive provides another command CONCAT\_WS. Here you have to specify the delimit operator first.  
  
Syntax: CONCAT\_WS (‘-‘,’Hive’,’is’,’a’,’data warehouse’,’in Hadoop’);  
Output: Hive-is-a-data warehouse-in Hadoop

12. 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.  
13. Describe REVERSE function in Hive with example?  
  
REVERSE function will reverse the characters in a string.  
  
Example: REVERSE(‘Hive’);  
Output: eviH

14. Describe TRIM function in Hive with example?  
  
TRIM function will remove the spaces associated with a string.  
  
Example: TRIM(‘ Hadoop ‘);  
Output: Hadoop.  
  
If we want to remove only leading or trailing spaces then we can specify the below commands respectively.  
  
LTRIM(‘ Hadoop’);  
RTRIM(‘Hadoop ‘);

15. Describe RLIKE in Hive with an example?  
  
RLIKE (Right-Like) is a special function in Hive where if any substring of A matches with B then it evaluates to true. It also obeys Java regular expression pattern. Users don’t need to put % symbol for a simple match in RLIKE.  
  
Examples: ‘Express’ RLIKE ‘Exp’ –> True  
‘Express’ RLIKE ‘^E.\*’ –> True (Regular expression)

select \* from personal\_d

> where id rlike '[0-9]';

Moreover, RLIKE will come handy when the string has some spaces. Without using TRIM function, RLIKE satisfies the required scenario. Suppose if A has value ‘Express ‘ (2 spaces additionally) and B has value ‘Express’. In these situations, RLIKE will work better without using TRIM.  
  
‘Express ‘ RLIKE ‘Express’ –> True  
  
Note: RLIKE evaluates to NULL if A or B is NULL.  
------------------------------------------------------------------------------------------------------------------------------------  
------------------------------------------------------------------------------------------------------------------------------------  
Hive Interview Questions and Answers for experienced:  
1. How to start Hive metastore service as a background process?  
  
We can start hive metastore service as a background process with below command.  
$ hive --service metastore &  
  
By using kill -9 <process id> we can stop this service.  
  
2. How to configure hive remote metastore in hive-site.xml file?

/usr/lib/hive/conf/hive-site.xml

We can configure remote metastore in hive-site.xml file with the below property.

<property>  
   <name>hive.metastore.uris</name>  
   <value>thrift://node1(or IP Address):9083</value>  
  <description>IP address (or fully-qualified domain name) and port of the metastore host</description>  
</property>  
  
<property>  
   <name>hive.metastore.uris</name>  
   <value>thrift://node1(or IP Address):9083</value>  
   <description>IP address (or fully-qualified domain name) and port of the metastore host</description>  
</property>  
  
3. What is the need for partitioning in Hive?  
  
To get quick response on hive queries

Partitioning is mainly intended for quick turn around time for queries on hive tables.  
  
4. We have already 3 tables named US,UK,IND in Hive. Now we have one more JPN created using hadoop fs -mkdir JPN. Can we move the content in IND to JPN directly?  
  
Yes, we can copy contents from hive warehouse directory table IND into JPN.

5. Now we have to display the contents in US,UK,IND,JPN. By using SELECT \* FROM TABLES is it possible to display?  
  
No, Because JPN is created by using fs -mkdir command. It is not part of metadata.

6. Is it possible to use same metastore by multiple users, in case of embedded hive?  
  
No, it is not possible to use metastore in sharing mode. It is recommended to use  
standalone “real" database like MySQL or PostGreSQL.

7. What is HCatalog and how to use it?  
  
HCatalog is a Table and Storage Management tool to Hadoop/HDFS. In MR, we use it by specifying InputOutput Formats i.e. HCatInputFormat and HCatOutputFormat.  
  
In Pig, we use it by specifying Storage types i.e HCatLoader and HCatStorer.  
  
8. If we run hive as a server, what are the available mechanisms for connecting it from application?  
  
Below are following ways by which we can connect with the Hive Server:  
  
    Thrift Client: Using thrift we can call hive commands from a various programming  
    languages e.g: Java, PHP, Python and Ruby.  
    JDBC Driver : It supports the Type 4 (pure Java) JDBC Driver  
    ODBC Driver: It supports ODBC protocol.  
  
9. Is multi line comment supported in Hive Script ?  
  
No.  
  
10. What is SerDe in Apache Hive?  
  
A SerDe is a Serializer Deserializer. Hive uses SerDe to read and write data from tables. An important concept behind Hive is that it DOES NOT own the Hadoop File System (HDFS) format that data is stored in. Users are able to write files to HDFS with whatever tools/mechanism takes their fancy(“CREATE EXTERNAL TABLE” or “LOAD DATA INPATH,” ) and use Hive to correctly “parse” that file format in a way that can be used by Hive. A SerDe is a powerful and customizable mechanism that Hive uses to “parse” data stored in HDFS to be used by Hive.  
  
11. Which classes are used by the Hive to Read and Write HDFS Files?  
  
Following classes are used by Hive to read and write HDFS files  
  
    TextInputFormat/HiveIgnoreKeyTextOutputFormat: These 2 classes read/write data in plain text file format.  
    SequenceFileInputFormat/SequenceFileOutputFormat: These 2 classes read/write data in hadoop SequenceFile format.  
  
12. What are the examples of the SerDe classes which hive uses to Serialize and Deserialize data?  
  
Hive currently use below 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 run time. Also it supports a lot of different protocols, including TBinaryProtocol, TJSONProtocol, TCTLSeparatedProtocol  
  
13. How do we write our 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.  
  
14. What is the functionality of Query Processor in Apache Hive ?  
  
This component implements the processing framework for converting SQL to a graph of map/reduce jobs and the execution time framework to run those jobs in the order of dependencies.  
  
15. What is ObjectInspector functionality ?  
  
Hive uses ObjectInspector to analyze the internal structure of the row object and also  
the structure of the individual columns. ObjectInspector provides a uniform way to access complex objects that can be stored in multiple formats in the memory, including:  
  
    Instance of a Java class (Thrift or native Java)  
    A standard Java object (we use java.util.List to represent Struct and Array, and use  
    java.util.Map to represent Map)  
    A lazily-initialized object (For example, a Struct of string fields stored in a single Java string object with starting offset for each field)  
    A complex object can be represented by a pair of ObjectInspector and Java Object.  
    The ObjectInspector not only tells us the structure of the Object, but also gives us ways to access the internal fields inside the Object.  
  
--------------------------------------------------------------------------------------------------------------------------------------  
--------------------------------------------------------------------------------------------------------------------------------------  
Hive Interview Questions and Answers  
1. What are the types of tables in Hive?  
  
There are two types of tables.  
  
    Managed tables  
    External tables  
  
Only while dropping tables these two differentiates. Otherwise both type of tables are very similar.  
2. What kind of data warehouse application is suitable for Hive?  
  
Hive is not a full database. The design constraints and limitations of Hadoop and HDFS  
impose limits on what Hive can do.  
Hive is most suited for data warehouse applications, where  
  
    Relatively static data is analyzed,  
    Fast response times are not required, and  
    When the data is not changing rapidly.  
  
3. Does Hive provide OLTP or OLAP?  
  
Hive doesn’t provide crucial features required for OLTP, Online Transaction Processing.  
It’s closer to being an OLAP tool, Online Analytic Processing. So, Hive is best suited for  
data warehouse applications, where a large data set is maintained and mined for insights, reports, etc.  
  
4. Does Hive support record level Insert, delete or update?  
  
No. Hive does not provide record-level update, insert, or delete. Henceforth, Hive does not  
provide transactions too. However, users can go with CASE statements and built in functions of Hive to satisfy the above DML operations. Thus, a complex update query in  
a RDBMS may need many lines of code in Hive.  
5. How can we change a column data type in Hive?  
  
We can use below command to alter data type of a column in hive.  
Shell  
ALTER TABLE table\_name CHANGE column\_name column\_name new\_datatype;  
   
ALTER TABLE table\_name CHANGE column\_name column\_name new\_datatype;  
  
Example: If we want to change the data type of empid column from integer to bigint in a  
table called employee.  
ALTER TABLE employee CHANGE empid empid BIGINT;  
  
ALTER TABLE employee CHANGE empid empid BIGINT;  
  
6. How can we copy the columns of a hive table into a file?  
  
By using awk command in shell, the output from HiveQL Describe command can be written to a file.  
$ hive -S -e "describe table\_name;" | awk -F" " ’{print 1}’ > ~/output.  
   
$ hive -S -e "describe table\_name;" | awk -F" " ’{print 1}’ > ~/output.  
  
7. How to rename a table in Hive?  
  
Using ALTER command with RENAME, we can rename a table in Hive.  
ALTER TABLE hive\_table\_name RENAME TO new\_name;  
  
ALTER TABLE hive\_table\_name RENAME TO new\_name;  
  
8. Is there any alternative way to rename a table without ALTER command?  
  
By using Import and export options we can be rename a table as shown below. Here we are saving the hive data into HDFS and importing back to new table like below.  
EXPORT TABLE tbl\_name TO 'HDFS\_location';  
IMPORT TABLE new\_tbl\_name FROM 'HDFS\_location';  
   
EXPORT TABLE tbl\_name TO 'HDFS\_location';  
IMPORT TABLE new\_tbl\_name FROM 'HDFS\_location';  
  
If we prefer to just preserve the data, we can create a new table from old table like  
below.  
CREATE TABLE new\_tbl\_name AS SELECT \* FROM old\_tbl\_name;  
DROP TABLE old\_tbl\_name;  
   
CREATE TABLE new\_tbl\_name AS SELECT \* FROM old\_tbl\_name;  
DROP TABLE old\_tbl\_name;  
  
9. What is the difference between order by and sort by in hive?  
  
    SORT BY will sort the data within each reducer. We can use any number of reducers  
    for SORT BY operation.  
    ORDER BY will sort all of the data together, which has to pass through one reducer.  
    Thus, ORDER BY in hive uses single reducer.  
    ORDER BY guarantees total order in the output while SORT BY only guarantees  
    ordering of the rows within a reducer. If there is more than one reducer, SORT BY may give partially ordered final results  
  
10. What is Double data type in Hive?  
  
Double data type in Hive will present the data differently unlike RDBMS.  
See the double type data below:  
14324.0  
342556.0  
1.28893E4  
  
E4 represents 10^4 here. So, the value1.28893E4 represents 12889.3. All the  
calculations will be accurately performed using double type  
  
It is crucial while exporting the double type data to any RDBMS since the type may be  
wrongly interpreted. So, it is advised to cast the double type into appropriate type before  
exporting.  
  
  
1. What is Metadata?  
Data about Data.

2. What is Hive?  
  
Hive is one of the important tool in Hadoop eco system and it provides an SQL like dialect to Hadoop distributed file system.

3. What are the features of Hive?  
  
Hive provides,  
  
    Tools to enable easy data extract/transform/load (ETL)  
    A mechanism to project structure on a variety of data formats  
    Access to files stored either directly in HDFS or other data storage systems as HBase  
    Query execution through MapReduce jobs.  
    SQL like language called HiveQL that facilitates querying and managing large data sets residing in hadoop.  
  
4. What are the limitations of Hive?  
  
Below are the limitations of Hive:  
  
    Hive is best suited for data warehouse applications, where a large data set is maintained and mined for insights, reports, etc.  
    Hive does not provide record-level update, insert, nor delete.  
    Hive queries have higher latency than SQL queries, because of start-up overhead for MapReduce jobs submitted for each hive query.   
  
     As Hadoop is a batch-oriented system, Hive doesn’t support OLTP (Online Transaction Processing).  
    Hive is close to OLAP (Online Analytic Processing) but not ideal since there is significant latency between issuing a query and receiving a reply, both due to the overhead of Mapreduce jobs and due to the size of the data sets Hadoop was designed to serve.  
    If we need OLAP, we need to use NoSQL databases like HBase that can be integrated with Hadoop.  
  
5. What is the differences Between Hive and HBase?  
  
Hive is not a database but a data warehousing frame work. Hive doesn’t provide record level operations on tables.  
  
    HBase is a NoSQL Database and it provides record level updates, inserts and deletes to the table data.  
    HBase doesn’t provide a query language like SQL, but Hive is now integrated with  
    HBase.  
  
6. What is Hive Metastore?  
  
The metastore is the central repository of Hive metadata. The metastore is divided into two pieces: a service and the backing store for the data. By default, the metastore is run in the same process as the Hive service. Using this service, it is possible to run the metastore as a standalone (remote) process. Set the METASTORE\_PORT environment variable to specify the port the server will listen on.

7. Wherever (Different Directory) we run hive query, it creates new metastore\_db, please explain the reason for it?  
  
Whenever we run the hive in embedded mode, it creates the local metastore. And  
before creating the metastore it looks whether metastore already exist or not. This property is defined in configuration file hive-site.xml.  
  
Property is “javax.jdo.option.ConnectionURL" with default value “jdbc:derby:;databaseName=metastore\_db;create=true".  
  
So to change the behavior change the location to absolute path, so metastore will be used from that location.

8. What are the different types of Hive Metastore?  
  
Below are three different types of metastore.  
  
    Embedded Metastore  
    Local Metastore  
    Remote Metastore  
  
9. What is the default Hive warehouse directory?  
  
It is /user/hive/warehouse directory in local file system.

10. How to start Hive Thrift server?  
  
We can issue below command from terminal to start Hive thrift server.  
  
$ hive –service hiveserver

|  |  |
| --- | --- |
|  |  |

It is a platform used to develop SQL type scripts to do MapReduce operations.  
PARTITIONING  
  
Partition tables changes how HIVE structures the data storage  
\*Used for distributing load horizantally  
  
ex: PARTITIONED BY (country STRING, state STRING);  
A subset of a table’s data set where one column has the same value for all records in the subset.  
In Hive, as in most databases that support partitioning, each partition is stored in a physically separate location—in Hive’s case, in a subdirectory of the root directory for the table.  
Partitions have several advantages. The column value corresponding to a partition doesn’t have to be repeated  
in every record in the partition, saving space, and queries with WHERE clauses that restrict the result set to specific values for the partition columns can perform more quickly, because they avoid scanning the directories of non matching partition values.  
  
 Partitioned by(country string,dept string)  
  
Limitations:  
A design that creates too many partitions is the large no of Hadoop files and directories that are created unnecessarily  
Too many partitions may optimize some quieries but be detrimental  
partitioned by(country string,state string);  
  
 hive> select \* from partitionemployees where country=’CANADA’;  
OK  
  
Dynamic ParttioningThis technique is convenient for  
partitioning a query result into a potentially large number of partitions in a new table, without having to write a separate query for each partition column value.  
SET hive.exec.dynamic.partition = true;  
2.  
SET hive.exec.dynamic.partition.mode = nonstrict  
BUCKETING  
  
IT is a technique for decomposing datasets into more manageable parts.  
if we bucket table and use column as the bucketing column, the value of this column will be hashed by a user-defined number into buckets. Records with the same column will always be stored in the same bucket.  
  
Decomposing datasets into more manageable parts while creating table clusteredby(uniqueid) into noof buckets No of buckets is fixed.  
If 2 tables are bucketed by empid hive can create a logically correct [sampling.it](http://sampling.it/) is also used for effiecient map side joins  
  
External Partition Tables:  
create external table if not exists customers.externalpartitionemployees(  
name String comment 'employee name'  
,salary float comment 'employee salary',  
subordinates Array<string> comment 'name of subordinates',  
deductions map<String,float> comment 'keys are deduction names values are percentages',  
address struct<street:String,city:String,state:string,zipcode:int> comment 'Home address')  
partitioned by(country string,universe string,galaxy string);  
alter table externalpartitionemployees add partition(country='US',universe='earth',galaxy='sun')  
location '/user/cloudera/usearthsun';  
  
load data local inpath '/home/cloudera/Desktop/externalpart' into table externalpartitionemployees partition(country='US',universe='earth',galaxy='sun');  
to archive data to s3 storage  
alter table externalpartitionemployees partition(country='US',universe='earth',galaxy='sun')  
set location '/user/cloudera/usearthsunarchive';  
  
  
FEATURES OF HIVE  
  
    It stores schema in a database and processed data into HDFS.  
    It is designed for OLAP.  
    It provides SQL type language for querying called HiveQL or HQL.  
    It is familiar, fast, scalable, and extensible.  
  
HIVE INTERFACES  
  
    Hive interfaces include WEB UI,  
    HIVE Command Line,  
    HD Insight(Windows Server)  
  
TYPES OF TABLES in Hive  
  
External Table  
Internal Table(Mangaed Table)  
External Table  
A table using a storage location and contents that are outside of Hive’s control.  
It is convenient for sharing data with other tools,  
when an external table is created, Hive does not create the external directory (or directories for partitioned tables), nor are the directory and data files deleted when an external table is  
dropped.  
  
Data can be used outside of hive Hive cannot own control [data.it](http://data.it/) delets only metadata Internal Tabe(or Manageable) Managed tables store their data inside the warehouse directory. Managed tables are less convenient for sharing with other tools. when you want full control over the table When u drp deletes data and schema in table Less convenient for sharing other tools.  
Collection Data types  
  
SRTUCT,MAP,ARRAY  
  
STRUCT  
STRUCT {first STRING; last STRING}, then  
the first name field can be referenced using name.first.  
struct(‘John’, ‘Doe’)  
  
arrays–similar data types–access by index  
map-key value pairs–access by column[“federal”]  
structs-different data types–access by dot  
MAP A collection of key-value tuples, where the fields are accessed  
using array notation (e.g., [‘key’]). For example, if a column  
name is of type MAP with key?value pairs  
‘first’?’John’ and ‘last’?’Doe’, then the last  
name can be referenced using name[‘last’].  
map(‘first’, ‘John’,’last’, ‘Doe’)  
  
ARRAY Ordered sequences of the same type that are indexable using zero-based integers. For example, if a column name is of type ARRAY of strings with the value [‘John’, ‘Doe’], then the second element can be referenced using name[1].  
  
CREATE TABLE employees (  
name STRING,  
salary FLOAT,  
subordinates ARRAY<STRING>,  
deductions MAP<STRING, FLOAT>,  
address STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>);  
  
The name is a simple string and for most employees, a float is large enough for the  
Primitive data types  
  
Tinyint  
Smallint  
Int  
Bigint  
Boolean  
Float  
Double  
String  
Timestamp  
Binary  
Complex struct  
Map  
Array  
User Defined Aggregate Functions  
  
User-defined functions that take multiple rows (or columns from multiple rows) and return a single “aggregation" of the data, such as a count of the rows, a sum or average of number values, etc.

User Defined Table generating Functions  
  
User-defined functions that take a column from a single record and expand it into multiple rows.  
Example includes converting a map into rows of key and value fields  
UDF  
  
Create new class extends udf Complie this class and add it to jar Put jar to hive classpath Register function  
Indexes  
  
For speedy access to columns in your database Create index <INDEX\_NAME> on table <TABLE\_NAME> as name  
Select \* from <INDEX\_NAME> where a = 1;  
objectinspector  
  
Hive uses ObjectInspector to analyze the internal structure of the row object and also the structure of the individual columns. ObjectInspector provides a uniform way to access complex objects that can be stored in  
multiple formats in the memory, including:  
• Instance of a Java class (Thrift or native Java)  
• A standard Java object (we use java.util.List to represent Struct and Array, and use  
java.util.Map to represent Map)  
• A lazily-initialized object (For example, a Struct of string fields stored in a single Java string object with starting offset for each field) A complex object can be represented by a pair of ObjectInspector and Java Object. The ObjectInspector not only tells us the structure of the Object, but also gives us ways to access  
the internal fields inside the Object.  
  
create table:  
  
create table if not exists customers.employees(  
name String comment 'employee name'  
,salary float comment 'employee salary',  
subordinates Array<string> comment 'name of subordinates',  
deductions map<String,float> comment 'keys are deduction names values are percentages',  
address struct<street:String,city:String,state:string,zipcode:int> comment 'Home address')  
comment 'Description of the table'  
tblproperties('creator'='me','created at'='01-20-15 2:59:00');  
   
load data local inpath '/home/cloudera/Desktop/employees' into table customers.employees;  
  
Copy Schema  
  
create table if not exists customers.employeescopy like customers.employees;  
  
–schema will be copied  
  
SerDe  
  
The Serializer/Deserializer or SerDe for short is used to parse the bytes of a record into columns or fields, the deserialization process. It is also used to create those record bytes (i.e., serialization)  
  
Built-in SerDes  
Avro (Hive 0.9.1 and later)  
ORC (Hive 0.11 and later)  
RegEx  
Thrift  
Parquet (Hive 0.13 and later)  
CSV (Hive 0.14 and later)  
  
Serde is a library built on top of Hadoop API. Serializer, Deserializer instructs HIVE on how to process a record.  
HIVE enables semi-structured or unstructured records to be processed also.  
Serde will deseralize data from file to object so that it can be queried using SQL syntax.  
  
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 customserde  
  
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  
  
    An RPC(Remote Procedure Call) system integrated into Hive.  
    Remote processes can send Hive statements to Hive through Thrift.  
    HIVE thrift allows access to HIVE over a single port.  
    Thrift is a software framework for scalable cross language services development.  
    Thrift allows clients using other languages to programmatically access HIVE remotely.  
    A Thrift service provides remote access from other processes.  
    Access using JDBC and ODBC are provided, too. They are implemented on top of the Thrift service  
  
MetaStore  
  
The service that maintains “metadata" information, such as table schemas.  
  
Hive requires this service to be running.  
  
support. By default, it uses a built-in Derby SQL server, which provides limited, single-process SQL for hive production environment,metastore service should run in an isolated jvm.  
Hive processes can communicate with metastore services using thrift Hive metastore data is persisted in acid database such as db/mysql In hive-default.xml.template file we can set name to store in local or warehouse  
<name> hive.metastore.warehouse.dir</name> <name>hive.metastore.local</name>  
  
Hive metastore is a database that stores metadata about your Hive tables (eg. table name, column names and types, table location, storage handler being used, number of buckets in the table, sorting columns if any, partition columns if any, etc.). When you create a table, this metastore gets updated with the information related to the new table which gets queried when you issue queries on that table.  
  
Avro  
  
Avro is a serialization format developed to address some of the common problems associated with evolving other serialization formats.  
  
Some of the benefits are:  
rich data structures, fast binary format, support for remote procedure calls, and built-in schema evolution  
JSON  
JSON (JavaScript Object Notation) is a lightweight data serialization format used commonly in web-based applications.  
HDFS  
(HDFS) A distributed, resilient file system for data storage (optimized for scanning large contiguous blocks of data on hard disks.) Distribution across a cluster provides horizontal scaling of data storage.  
  
Blocks of HDFS files are replicated across the cluster (by default, three times) to prevent data loss when hard drives or whole servers fail.  
Modes  
Strict—these are used to protect large data in partitioned tables.Instead of extracting all data from partitioned table strict mode is safety measure which allows queries with where cause only on partitones  
  
Set hive.mapred.mode=strict;  
Select \* from employee where Nonstrict  
   
Hive commands  
-d a=b –define a=b  
-e –sql from command line  
Hive –e “select \* from"  
-f<filename>-sql from files  
-H –help  
-h<hostname>connects to hive server on remote host  
--hiveconf<property=value>  
-i<filename> initialization sql file  
-p<port> connecting to hive server on port number  
-s/--silent mode in interactive shell  
-v verbose  
  
Job  
In the Hadoop context, a job is a self-contained workflow submitted to MapReduce.  
  
It encompasses all the work required to perform a complete calculation, from reading input to generating output.  
  
Job Tracker  
The MapReduce JobTracker will decompose the job into one or more tasks for distribution and execution around the cluster.  
  
The top-level controller of all jobs using Hadoop’s MapReduce.  
The JobTracker accepts job submissions, determines what tasks to run and where to run them, monitors their execution, restarts failed tasks as needed, and provides a web console for monitoring job and task execution,  
viewing logs, etc  
Task  
  
In the MapReduce context, a task is the smallest unit of work performed on a single cluster node, as part of an overall job.  
By default each task involves a separate JVM process.  
Each map and reduce invocation will have its own task.  
  
Input Format  
  
The input format determines how input streams, usually from files, are split into records.  
  
A SerDe handles parsing the record into columns. A custom input format can be specified when creating a table using the INPUTFORMAT clause.  
  
Output Format  
  
The output format determines how records are written to output streams, usually to files.  
A SerDe handles serialization of each record into an appropriate byte stream.  
  
A custom output format can be specified when creating a table using the OUTPUTFORMAT clause.  
  
The output format for the default STORED AS TEXTFILE specification is implemented by the Java object named org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat  
Partitioner  
  
Partitioner controls the partitioning of the keys of the intermediate map-outputs. The key (or a subset of the key) is used to derive the partition, typically by a hash function. The total number of partitions is the same as the number of reduce tasks for the job.  
Metastore  
  
hive stores schema and other Hive uses to store table schemas and other metadata for hive production environment,metastore service should run in an isolated jvm.  
Hive processes can communicate with metastore services using thrift Hive metastore data is persisted in acid database such as db/mysql In hive-default.xml.template file we can set name to store in local or warehouse  
<name> hive.metastore.warehouse.dir</name>  
<name>hive.metastore.local</name>  
  
storage formats  
Text/sequencefile/rc file  
Rc-recordcolumn file  
Crate table xx(a int) stored as rc file;  
--compression  
Set hive.exec.compress.output=true  
Set mapred.output.compression.codec=;  
--import  
Insert overwrite table xx Select \* from  
For compression and serialization we use rcfile and sequencefile  
  
Following classes are used by Hive to read and write HDFS files  
•TextInputFormat/HiveIgnoreKeyTextOutputFormat: These 2 classes read/write data in plain  
text file format.  
•SequenceFileInputFormat/SequenceFileOutputFormat: These 2 classes read/write data in  
hadoop SequenceFile format.  
  
Hive natively supports text file format, however hive also has support for other binary formats. Hive supports Sequence, Avro, RCFiles.  
========================================================================

Read it from here

==========================================================================  
Sequence files :-General binary format. splittable, compressible and row oriented.a typical example can be. if we have lots of small file, we may use sequence 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 Sequence 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 subsequently..  
Joins  
  
Innerjoins-returns all rows when there is atleast one match in both tables  
  
Leftouterjoin-returns all rows from left table and matched rows from right table  
  
Right join-returns all rows from right table and matched rows from left  
  
Full join-returns all rows  
  
Map join-join can be performed as map only job  
One of the table being joined should be small  
  
Select /\*mapjoin\*/ a.key,a.value from a join b on a.key=b.key  
  
Oredrby/sortby/distributedby/clusterby  
Orderby-ascending or desc pushing all dat through one reducer  
  
Sortby-orders data at each of n reducers but each reducer can receive overlapping ranges of data.you end up with one or more sotrted files with overlapping ranges  
  
Distributeby-ensures each of n reducers gets non overlapping ranges of x,but doesnot sort the o/p of each reducer.u end with n or more unsorted files with non overlapping ranges  
  
Clusterby-ensures each of n reducers gets non overlapping ranges,then sorts by those changes at reducers.this gives global ordering and is same as doing distributed by and sort by.you end up with n or more sorted files with non overlapping ranges  
  
Cluster by is more scalable than order by sampling  
Tablesample sample ur table It can return subsets of buckets bucket smaplingOr hdfs blocks block sampling or only first n elements from each ip split compression To optimized memory streams are compressed using CODEC which is specified as table property.  
RC file, ORC, Sequence, LZO, can be compressed.  
  
security  
Unmask value. P  
<property>  
<name> hive.files.umask.value</name>  
<description> the dfs umask value for hive created folder </description>  
</property>  
Hcatalog  
  
It is a metadata and table management system for Hadoop platform. It enables storage of data in any format regardless of any structure.  
It supports reading and writing files in any format for which a hive SerDe can be written .  
By default, HCatalog supports RCFile, CSV, JSON and Sequence File formats.

To use a custom format you must provide input format, output format and SerDe.  
HCatalog is built on top a HIVE metastore and incorporates components from the HIVE DDL.  
HCatalog provides read and write interfaces for PIG and mapreduce and used HIVE Command line interface for issuing data definition and metadata exploration commands.  
It also presents a REST interface to allow external tools to access to HIVE DDL operations, such as “create table" and “describle table".  
HCatalog presents a relational view of Data  
Data is stored in tables and these tables can be places into databases.  
Tables can also be partitioned on one more keys.  
For a given value of a key there will be one partition that contains all rows with that value  
  
InputFormat  
  
The InputFormat class is one of the fundamental classes of the MapReduce framework. It is responsible for defining two main things: InputSplit and RecordReader  
  
Input Split  
  
InputSplit defines both the size of individual map tasks (and, consequently, the number of map tasks) and its “ideal" execution server (locality).  
  
Record Reader  
  
The RecordReader is responsible for actually reading records from the input file, and submitting them (as key/value pairs) to the mapper.  
  
Counters  
Hadoop provides some inbuilt counters for every job, for purposes like counting the number of records or bytes processed  
Map Reduce  
  
MapReduce is designed to scale computation horizontally by decomposing map and reduce steps into tasks and distributing those tasks across a cluster.  
The MapReduce runtime provided by Hadoop handles decomposition of a job into tasks, distribution around the cluster, movement of a particular task to the machine that holds the data for the task , movement of data  
  
to tasks (as needed), and automated reexecution of failed tasks and other error recovery and logging services.  
Streaming  
  
Streaming offers an alternative way to transform data.  
During a streaming job, the Hadoop Streaming API opens an I/O pipe to an external process.  
Data is then passed to the process, which operates on the data it reads from the standard input and writes the results out through the standard output, and back to the Streaming API job.  
  
Hive provides several clauses to use streaming:  
MAP(), REDUCE(), and TRANSFORM().

Speculative Exection  
  
Speculative execution is a feature of Hadoop that launches a certain number of duplicate tasks.  
While this consumes more resources computing duplicate copies of data  
that may be discarded, the goal of this feature is to improve overall job progress by getting individual task results faster, and detecting then black-listing slow-running TaskTrackers.  
  
Distributed Cache  
  
Distributed Cache is a facility provided by the MapReduce framework to cache files (text, archives, jars and so on) needed by applications during execution of the job.  
The framework will copy the necessary files to the slave node before any tasks for the job are executed on that node.  
  
WebDav

WebDAV is an Internet-based open standard that enables editing Web sites over HTTP and HTTPS connections. WebDAV yields several advantages over the File Transfer Protocol (FTP), the most notable advantages are more security options and the ability to use a single TCP port for all communication.  
Fair Scheduler  
  
In FairScheduler, Jobs, which are submitted to queues, are placed into pools.  
Each pool is assigned a number of task slots based on a number of factors including the total slot capacity of the cluster, the current demand (where “demand" is the number of tasks in a pool) on other pools, minimum slot guarantees, and available slot capacity.  
Pools may optionally have minimum slot guarantees. These pools are said to have an SLA, with the minimum number of slots providing the vehicle for ensuring task scheduling within a given period of time.  
Beyond the minimum slot guarantees, each pool gets an equal number of the remaining available slots on the cluster; this is where the “fair share" portion of the name comes from  
Oozie  
Oozie is a workflow/coordination system that you can use to manage Apache Hadoop jobs.It implements a set of remote Web Services APIs that can be invoked from Oozie client components and third-party applications.  
  
A single Oozie server implements all four functional Oozie components:  
Oozie Workflow This component provides support for defining and executing a controlled sequence of MapReduce, Hive, and Pig jobs.  
? Oozie Coordinator — This provides support for the automatic execution of Workflows  
based on the events and the presence of system resources.  
? Oozie Bundles — This engine enables you to define and execute a “bundle" of applications, thus providing a way to batch together a set of Coordinator applications that can be managed together.  
? Oozie Service Level Agreement (SLA) — This provides support for tracking the execution of Workflow applications.  
  
Common join versus map-side join  
“common join” to refer to a join operation where  
one or more reducers are required to physically join the table rows. Map-side joins, as the  
name would suggest, perform the join across parallel map tasks and eliminate the need  
for the reduce phase.  
Kerberos  
  
“Kerberos Authentication,"  
describes the Kerberos protocol, and discusses how Hadoop uses Kerberos authentication for Remote Procedure Calls (RPCs), and how Hadoop web consoles can be protected with Kerberos authentication using HTTP Simple and Protected Negotiation Mechanism (SPNEGO).  
  
Hadoop uses the Kerberos protocol to authenticate users to Hadoop, and to authenticate Hadoop services to each other.  
  
Kerberos relies on the concept of “tickets" in order to work. In Kerberos, three parties are involved:  
  
    A client (which can be a user or service) that requests access to a resource.  
    A requested resource (which is typically a service).  
    The Kerberos Key Distribution Center (KDC), which includes an Authentication Service (AS) and a Ticket Granting Service (TGS). The KDC is the main hub of all communications.  
  
Apache Knox Gateway  
Knox provides perimeter security for Hadoop clusters  
  
Provide security to all of Hadoop’s REST & HTTP services  
  
Support for REST APIs for Apache Ambari, Apache Falcon and Apache Ranger  
  
Expose Hadoop’s REST/HTTP services without revealing network details, with SSL provided out of box

/\*  
How to handle blank, empty string and make it as NULL  
\*/  
/\*DATA emp file  
John,,200  
Albert,HR<1900  
Mark,,10000  
Frank,TP,1150  
Loopa,HR,1100  
Lut,,1300  
Lesa,TP,900  
Pars,HR,800  
leo,HR,700  
lock,,650  
Bhut,TP,800  
Lio,TP,500  
\*/

Hive Interview Questions and Answers for experienced  
1. 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:  
  
    The Hive SET command  
    The command line -hiveconf option  
    hive-site.xml  
    hive-default.xml  
    hadoop-site.xml (or, equivalently, core-site.xml, hdfs-site.xml, and mapred-site.xml)  
    hadoop-default.xml (or, equivalently, core-default.xml, hdfs-default.xml, and mapred-default.xml)  
  
2. How do change settings within Hive Session?  
We can change settings from within a session, too, using the SET command. This is useful for changing Hive or MapReduce job settings for a particular query. For example, the following command ensures buckets are populated according to the table definition.  
hive> SET hive.enforce.bucketing=true;  
   
hive> SET hive.enforce.bucketing=true;  
  
To see the current value of any property, use SET with just the property name:  
hive> SET hive.enforce.bucketing;  
hive.enforce.bucketing=true  
   
hive> SET hive.enforce.bucketing;  
hive.enforce.bucketing=true  
  
By itself, SET will list all the properties and their values set by Hive. This list will not include Hadoop defaults, unless they have been explicitly overridden in one of the ways covered in the above answer. Use SET -v to list all the properties in the system, including Hadoop defaults.  
3. How to print header on Hive query results?  
  
We need to use following set command before our query to show column headers in STDOUT.  
hive> set hive.cli.print.header=true;  
  
hive> set hive.cli.print.header=true;  
  
4. How to get detailed description of a table in Hive?  
  
Use below hive command to get a detailed description of a hive table.  
hive> describe extended <tablename>;  
   
hive> describe extended <tablename>;  
  
5. How to access sub directories 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.  
hive> Set mapred.input.dir.recursive=true;  
hive> Set hive.mapred.supports.subdirectories=true;      
   
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. This is suitable for a scenario where the directory structure is as following: /data/country/state/city  
6. How to skip header rows from a table in Hive?  
  
Suppose while processing some log files, we may find header records.  
  
System=….  
Version=…  
Sub-version=….  
  
Like above, It may have 3 lines of headers that we do not want to include in our Hive query. To skip header lines from our tables in Hive we can set a table property that will allow us to skip the header lines.  
CREATE EXTERNAL TABLE userdata (  
name STRING,  
job STRING,  
dob STRING,  
id INT,  
salary INT)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘ ‘ STORED AS TEXTFILE  
LOCATION ‘/user/data’  
TBLPROPERTIES("skip.header.line.count"="3");  
   
CREATE EXTERNAL TABLE userdata (  
name STRING,  
job STRING,  
dob STRING,  
id INT,  
salary INT)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘ ‘ STORED AS TEXTFILE  
LOCATION ‘/user/data’  
TBLPROPERTIES("skip.header.line.count"="3");  
  
7. 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 querying, if we specify any column other than above list, will result in NULL values.  
8. What is the maximum size of string data type supported by Hive?  
  
Maximum size is 2 GB.  
9. What are the Binary Storage formats supported in Hive?   
By default Hive supports text file format, however hive also supports below binary formats.  
  
Sequence Files, Avro Data files, RCFiles, ORC files, Parquet files  
  
Sequence files: General binary format. splittable, compressible and row oriented. a typical example can be. if we have lots of small file, we may use sequence 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 Sequence 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 subsequently.  
  
ORC Files: Optimized Record Columnar files  
10. is HQL case sensitive?  
  
HQL is not case sensitive.  
11. Describe CONCAT function in Hive with Example?  
  
CONCAT function will concatenate the input strings. We can specify any number of strings separated by comma.  
  
Example: CONCAT (‘Hive’,’-‘,’is’,’-‘,’a’,’-‘,’data warehouse’,’-‘,’in Hadoop’);  
Output: Hive-is-a-data warehouse-in Hadoop  
  
So, every time we delimit the strings by ‘-‘. If it is common for all the strings, then Hive provides another command CONCAT\_WS. Here you have to specify the delimit operator first.  
  
Syntax: CONCAT\_WS (‘-‘,’Hive’,’is’,’a’,’data warehouse’,’in Hadoop’);  
Output: Hive-is-a-data warehouse-in Hadoop  
12. 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.  
13. Describe REVERSE function in Hive with example?  
  
REVERSE function will reverse the characters in a string.  
  
Example: REVERSE(‘Hive’);  
Output: eviH  
14. Describe TRIM function in Hive with example?  
  
TRIM function will remove the spaces associated with a string.  
  
Example: TRIM(‘ Hadoop ‘);  
Output: Hadoop.  
  
If we want to remove only leading or trailing spaces then we can specify the below commands respectively.  
  
LTRIM(‘ Hadoop’);  
RTRIM(‘Hadoop ‘);  
15. Describe RLIKE in Hive with an example?  
  
RLIKE (Right-Like) is a special function in Hive where if any substring of A matches with B then it evaluates to true. It also obeys Java regular expression pattern. Users don’t need to put % symbol for a simple match in RLIKE.  
  
Examples: ‘Express’ RLIKE ‘Exp’ –> True  
‘Express’ RLIKE ‘^E.\*’ –> True (Regular expression)  
  
Moreover, RLIKE will come handy when the string has some spaces. Without using TRIM function, RLIKE satisfies the required scenario. Suppose if A has value ‘Express ‘ (2 spaces additionally) and B has value ‘Express’. In these situations, RLIKE will work better without using TRIM.  
  
‘Express ‘ RLIKE ‘Express’ –> True  
  
Note: RLIKE evaluates to NULL if A or B is NULL.  
------------------------------------------------------------------------------------------------------------------------------------  
------------------------------------------------------------------------------------------------------------------------------------  
Hive Interview Questions and Answers for experienced:  
1. How to start Hive metastore service as a background process?  
  
We can start hive metastore service as a background process with below command.  
$ hive --service metastore &  
  
By using kill -9 <process id> we can stop this service.  
  
2. How to configure hive remote metastore in hive-site.xml file?  
  
We can configure remote metastore in hive-site.xml file with the below property.  
<property>  
   <name>hive.metastore.uris</name>  
   <value>thrift://node1(or IP Address):9083</value>  
   <description>IP address (or fully-qualified domain name) and port of the metastore host</description>  
</property>  
  
<property>  
   <name>hive.metastore.uris</name>  
   <value>thrift://node1(or IP Address):9083</value>  
   <description>IP address (or fully-qualified domain name) and port of the metastore host</description>  
</property>  
  
3. What is the need for partitioning in Hive?  
  
Partitioning is mainly intended for quick turn around time for queries on hive tables.  
  
4. We have already 3 tables named US,UK,IND in Hive. Now we have one more JPN created using hadoop fs -mkdir JPN. Can we move the content in IND to JPN directly?  
  
Yes, we can copy contents from hive warehouse directory table IND into JPN.  
5. Now we have to display the contents in US,UK,IND,JPN. By using SELECT \* FROM TABLES is it possible to display?  
  
No, Because JPN is created by using fs -mkdir command. It is not part of metadata.  
6. Is it possible to use same metastore by multiple users, in case of embedded hive?  
  
No, it is not possible to use metastore in sharing mode. It is recommended to use  
standalone “real" database like MySQL or PostGreSQL.  
7. What is HCatalog and how to use it?  
  
HCatalog is a Table and Storage Management tool to Hadoop/HDFS. In MR, we use it by specifying InputOutput Formats i.e. HCatInputFormat and HCatOutputFormat.  
  
In Pig, we use it by specifying Storage types i.e HCatLoader and HCatStorer.  
  
8. If we run hive as a server, what are the available mechanisms for connecting it from application?  
  
Below are following ways by which we can connect with the Hive Server:  
  
    Thrift Client: Using thrift we can call hive commands from a various programming  
    languages e.g: Java, PHP, Python and Ruby.  
    JDBC Driver : It supports the Type 4 (pure Java) JDBC Driver  
    ODBC Driver: It supports ODBC protocol.  
  
9. Is multi line comment supported in Hive Script ?  
  
No.  
  
10. What is SerDe in Apache Hive?  
  
A SerDe is a Serializer Deserializer. Hive uses SerDe to read and write data from tables. An important concept behind Hive is that it DOES NOT own the Hadoop File System (HDFS) format that data is stored in. Users are able to write files to HDFS with whatever tools/mechanism takes their fancy(“CREATE EXTERNAL TABLE” or “LOAD DATA INPATH,” ) and use Hive to correctly “parse” that file format in a way that can be used by Hive. A SerDe is a powerful and customizable mechanism that Hive uses to “parse” data stored in HDFS to be used by Hive.  
  
11. Which classes are used by the Hive to Read and Write HDFS Files?  
  
Following classes are used by Hive to read and write HDFS files  
  
    TextInputFormat/HiveIgnoreKeyTextOutputFormat: These 2 classes read/write data in plain text file format.  
    SequenceFileInputFormat/SequenceFileOutputFormat: These 2 classes read/write data in hadoop SequenceFile format.  
  
12. What are the examples of the SerDe classes which hive uses to Serialize and Deserialize data?  
  
Hive currently use below 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 run time. Also it supports a lot of different protocols, including TBinaryProtocol, TJSONProtocol, TCTLSeparatedProtocol  
  
13. How do we write our 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.  
  
14. What is the functionality of Query Processor in Apache Hive ?  
  
This component implements the processing framework for converting SQL to a graph of map/reduce jobs and the execution time framework to run those jobs in the order of dependencies.  
  
15. What is ObjectInspector functionality ?  
  
Hive uses ObjectInspector to analyze the internal structure of the row object and also  
the structure of the individual columns. ObjectInspector provides a uniform way to access complex objects that can be stored in multiple formats in the memory, including:  
  
    Instance of a Java class (Thrift or native Java)  
    A standard Java object (we use java.util.List to represent Struct and Array, and use  
    java.util.Map to represent Map)  
    A lazily-initialized object (For example, a Struct of string fields stored in a single Java string object with starting offset for each field)  
    A complex object can be represented by a pair of ObjectInspector and Java Object.  
    The ObjectInspector not only tells us the structure of the Object, but also gives us ways to access the internal fields inside the Object.  
  
--------------------------------------------------------------------------------------------------------------------------------------  
--------------------------------------------------------------------------------------------------------------------------------------  
Hive Interview Questions and Answers  
1. What are the types of tables in Hive?  
  
There are two types of tables.  
  
    Managed tables  
    External tables  
  
Only while dropping tables these two differentiates. Otherwise both type of tables are very similar.  
2. What kind of data warehouse application is suitable for Hive?  
  
Hive is not a full database. The design constraints and limitations of Hadoop and HDFS  
impose limits on what Hive can do.  
Hive is most suited for data warehouse applications, where  
  
    Relatively static data is analyzed,  
    Fast response times are not required, and  
    When the data is not changing rapidly.  
  
3. Does Hive provide OLTP or OLAP?  
  
Hive doesn’t provide crucial features required for OLTP, Online Transaction Processing.  
It’s closer to being an OLAP tool, Online Analytic Processing. So, Hive is best suited for  
data warehouse applications, where a large data set is maintained and mined for insights, reports, etc.  
  
4. Does Hive support record level Insert, delete or update?  
  
No. Hive does not provide record-level update, insert, or delete. Henceforth, Hive does not  
provide transactions too. However, users can go with CASE statements and built in functions of Hive to satisfy the above DML operations. Thus, a complex update query in  
a RDBMS may need many lines of code in Hive.  
5. How can we change a column data type in Hive?  
  
We can use below command to alter data type of a column in hive.  
Shell  
ALTER TABLE table\_name CHANGE column\_name column\_name new\_datatype;  
   
ALTER TABLE table\_name CHANGE column\_name column\_name new\_datatype;  
  
Example: If we want to change the data type of empid column from integer to bigint in a  
table called employee.  
ALTER TABLE employee CHANGE empid empid BIGINT;  
  
ALTER TABLE employee CHANGE empid empid BIGINT;  
  
6. How can we copy the columns of a hive table into a file?  
  
By using awk command in shell, the output from HiveQL Describe command can be written to a file.  
$ hive -S -e "describe table\_name;" | awk -F" " ’{print 1}’ > ~/output.  
   
$ hive -S -e "describe table\_name;" | awk -F" " ’{print 1}’ > ~/output.  
  
7. How to rename a table in Hive?  
  
Using ALTER command with RENAME, we can rename a table in Hive.  
ALTER TABLE hive\_table\_name RENAME TO new\_name;  
  
ALTER TABLE hive\_table\_name RENAME TO new\_name;  
  
8. Is there any alternative way to rename a table without ALTER command?  
  
By using Import and export options we can be rename a table as shown below. Here we are saving the hive data into HDFS and importing back to new table like below.  
EXPORT TABLE tbl\_name TO 'HDFS\_location';  
IMPORT TABLE new\_tbl\_name FROM 'HDFS\_location';  
   
EXPORT TABLE tbl\_name TO 'HDFS\_location';  
IMPORT TABLE new\_tbl\_name FROM 'HDFS\_location';  
  
If we prefer to just preserve the data, we can create a new table from old table like  
below.  
CREATE TABLE new\_tbl\_name AS SELECT \* FROM old\_tbl\_name;  
DROP TABLE old\_tbl\_name;  
   
CREATE TABLE new\_tbl\_name AS SELECT \* FROM old\_tbl\_name;  
DROP TABLE old\_tbl\_name;  
  
9. What is the difference between order by and sort by in hive?  
  
    SORT BY will sort the data within each reducer. We can use any number of reducers  
    for SORT BY operation.  
    ORDER BY will sort all of the data together, which has to pass through one reducer.  
    Thus, ORDER BY in hive uses single reducer.  
    ORDER BY guarantees total order in the output while SORT BY only guarantees  
    ordering of the rows within a reducer. If there is more than one reducer, SORT BY may give partially ordered final results  
  
10. What is Double data type in Hive?  
  
Double data type in Hive will present the data differently unlike RDBMS.  
See the double type data below:  
14324.0  
342556.0  
1.28893E4  
  
E4 represents 10^4 here. So, the value1.28893E4 represents 12889.3. All the  
calculations will be accurately performed using double type  
  
It is crucial while exporting the double type data to any RDBMS since the type may be  
wrongly interpreted. So, it is advised to cast the double type into appropriate type before  
exporting.  
  
  
-----------------------------------------------------------------------------------------------------------------------------------------------  
-----------------------------------------------------------------------------------------------------------------------------------------------  
Hive Interview Questions and Answers  
1. What is Metadata?  
Data about Data.  
2. What is Hive?  
  
Hive is one of the important tool in Hadoop eco system and it provides an SQL like dialect to Hadoop distributed file system.  
3. What are the features of Hive?  
  
Hive provides,  
  
    Tools to enable easy data extract/transform/load (ETL)  
    A mechanism to project structure on a variety of data formats  
    Access to files stored either directly in HDFS or other data storage systems as HBase  
    Query execution through MapReduce jobs.  
    SQL like language called HiveQL that facilitates querying and managing large data sets residing in hadoop.  
  
4. What are the limitations of Hive?  
  
Below are the limitations of Hive:  
  
    Hive is best suited for data warehouse applications, where a large data set is maintained and mined for insights, reports, etc.  
    Hive does not provide record-level update, insert, nor delete.  
    Hive queries have higher latency than SQL queries, because of start-up overhead for MapReduce jobs submitted for each hive query.   
  
     As Hadoop is a batch-oriented system, Hive doesn’t support OLTP (Online Transaction Processing).  
    Hive is close to OLAP (Online Analytic Processing) but not ideal since there is significant latency between issuing a query and receiving a reply, both due to the overhead of Mapreduce jobs and due to the size of the data sets Hadoop was designed to serve.  
    If we need OLAP, we need to use NoSQL databases like HBase that can be integrated with Hadoop.  
  
5. What is the differences Between Hive and HBase?  
  
Hive is not a database but a data warehousing frame work. Hive doesn’t provide record level operations on tables.  
  
    HBase is a NoSQL Database and it provides record level updates, inserts and deletes to the table data.  
    HBase doesn’t provide a query language like SQL, but Hive is now integrated with  
    HBase.  
  
6. What is Hive Metastore?  
  
The metastore is the central repository of Hive metadata. The metastore is divided into two pieces: a service and the backing store for the data. By default, the metastore is run in the same process as the Hive service. Using this service, it is possible to run the metastore as a standalone (remote) process. Set the METASTORE\_PORT environment variable to specify the port the server will listen on.  
7. Wherever (Different Directory) we run hive query, it creates new metastore\_db, please explain the reason for it?  
  
Whenever we run the hive in embedded mode, it creates the local metastore. And  
before creating the metastore it looks whether metastore already exist or not. This property is defined in configuration file hive-site.xml.  
  
Property is “javax.jdo.option.ConnectionURL" with default value “jdbc:derby:;databaseName=metastore\_db;create=true".  
  
So to change the behavior change the location to absolute path, so metastore will be used from that location.  
8. What are the different types of Hive Metastore?  
  
Below are three different types of metastore.  
  
    Embedded Metastore  
    Local Metastore  
    Remote Metastore  
  
9. What is the default Hive warehouse directory?  
  
It is /user/hive/warehouse directory in local file system.  
10. How to start Hive Thrift server?  
  
We can issue below command from terminal to start Hive thrift server.  
  
$ hive –service hiveserver

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif

https://ssl.gstatic.com/ui/v1/icons/mail/profile_mask2.png

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Viraj.Hadoop** | | Sep 18, 2018, 12:40 PM (17 hours ago) |  | https://mail.google.com/mail/u/0/images/cleardot.gif  https://mail.google.com/mail/u/0/images/cleardot.gif |
| |  | | --- | | to Hadoop  https://mail.google.com/mail/u/0/images/cleardot.gif | | | |

Hive:  
Datawarehouse app built on top of hadoop(select,join,groupby…..)  
  
It is a platform used to develop SQL type scripts to do MapReduce operations.  
PARTITIONING  
  
Partition tables changes how HIVE structures the data storage  
\*Used for distributing load horizantally  
  
ex: PARTITIONED BY (country STRING, state STRING);  
A subset of a table’s data set where one column has the same value for all records in the subset.  
In Hive, as in most databases that support partitioning, each partition is stored in a physically separate location—in Hive’s case, in a subdirectory of the root directory for the table.  
Partitions have several advantages. The column value corresponding to a partition doesn’t have to be repeated  
in every record in the partition, saving space, and queries with WHERE clauses that restrict the result set to specific values for the partition columns can perform more quickly, because they avoid scanning the directories of non matching partition values.  
  
 Partitioned by(country string,dept string)  
  
Limitations:  
A design that creates too many partitions is the large no of Hadoop files and directories that are created unnecessarily  
Too many partitions may optimize some quieries but be detrimental  
partitioned by(country string,state string);  
  
 hive> select \* from partitionemployees where country=’CANADA’;  
OK  
  
Dynamic ParttioningThis technique is convenient for  
partitioning a query result into a potentially large number of partitions in a new table, without having to write a separate query for each partition column value.  
SET hive.exec.dynamic.partition = true;  
2.  
SET hive.exec.dynamic.partition.mode = nonstrict  
BUCKETING  
  
IT is a technique for decomposing datasets into more manageable parts.  
if we bucket table and use column as the bucketing column, the value of this column will be hashed by a user-defined number into buckets. Records with the same column will always be stored in the same bucket.  
  
Decomposing datasets into more manageable parts while creating table clusteredby(uniqueid) into noof buckets No of buckets is fixed.  
If 2 tables are bucketed by empid hive can create a logically correct [sampling.it](http://sampling.it/) is also used for effiecient map side joins  
  
External Partition Tables:  
create external table if not exists customers.externalpartitionemployees(  
name String comment 'employee name'  
,salary float comment 'employee salary',  
subordinates Array<string> comment 'name of subordinates',  
deductions map<String,float> comment 'keys are deduction names values are percentages',  
address struct<street:String,city:String,state:string,zipcode:int> comment 'Home address')  
partitioned by(country string,universe string,galaxy string);  
alter table externalpartitionemployees add partition(country='US',universe='earth',galaxy='sun')  
location '/user/cloudera/usearthsun';  
  
load data local inpath '/home/cloudera/Desktop/externalpart' into table externalpartitionemployees partition(country='US',universe='earth',galaxy='sun');  
to archive data to s3 storage  
alter table externalpartitionemployees partition(country='US',universe='earth',galaxy='sun')  
set location '/user/cloudera/usearthsunarchive';  
  
  
FEATURES OF HIVE  
  
    It stores schema in a database and processed data into HDFS.  
    It is designed for OLAP.  
    It provides SQL type language for querying called HiveQL or HQL.  
    It is familiar, fast, scalable, and extensible.  
  
HIVE INTERFACES  
  
    Hive interfaces include WEB UI,  
    HIVE Command Line,  
    HD Insight(Windows Server)  
  
TYPES OF TABLES in Hive  
  
External Table  
Internal Table(Mangaed Table)  
External Table  
A table using a storage location and contents that are outside of Hive’s control.  
It is convenient for sharing data with other tools,  
when an external table is created, Hive does not create the external directory (or directories for partitioned tables), nor are the directory and data files deleted when an external table is  
dropped.  
  
Data can be used outside of hive Hive cannot own control [data.it](http://data.it/) delets only metadata Internal Tabe(or Manageable) Managed tables store their data inside the warehouse directory. Managed tables are less convenient for sharing with other tools. when you want full control over the table When u drp deletes data and schema in table Less convenient for sharing other tools.  
Collection Data types  
  
SRTUCT,MAP,ARRAY  
  
STRUCT  
STRUCT {first STRING; last STRING}, then  
the first name field can be referenced using name.first.  
struct(‘John’, ‘Doe’)  
  
arrays–similar data types–access by index  
map-key value pairs–access by column[“federal”]  
structs-different data types–access by dot  
MAP A collection of key-value tuples, where the fields are accessed  
using array notation (e.g., [‘key’]). For example, if a column  
name is of type MAP with key?value pairs  
‘first’?’John’ and ‘last’?’Doe’, then the last  
name can be referenced using name[‘last’].  
map(‘first’, ‘John’,’last’, ‘Doe’)  
  
ARRAY Ordered sequences of the same type that are indexable using zero-based integers. For example, if a column name is of type ARRAY of strings with the value [‘John’, ‘Doe’], then the second element can be referenced using name[1].  
  
CREATE TABLE employees (  
name STRING,  
salary FLOAT,  
subordinates ARRAY<STRING>,  
deductions MAP<STRING, FLOAT>,  
address STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>);  
  
The name is a simple string and for most employees, a float is large enough for the  
Primitive data types  
  
Tinyint  
Smallint  
Int  
Bigint  
Boolean  
Float  
Double  
String  
Timestamp  
Binary  
Complex struct  
Map  
Array  
User Defined Aggregate Functions  
  
User-defined functions that take multiple rows (or columns from multiple rows) and return a single “aggregation" of the data, such as a count of the rows, a sum or average of number values, etc.  
User Defined Table generating Functions  
  
User-defined functions that take a column from a single record and expand it into multiple rows.  
Example includes converting a map into rows of key and value fields  
UDF  
  
Create new class extends udf Complie this class and add it to jar Put jar to hive classpath Register function  
Indexes  
  
For speedy access to columns in your database Create index <INDEX\_NAME> on table <TABLE\_NAME> as name  
Select \* from <INDEX\_NAME> where a = 1;  
objectinspector  
  
Hive uses ObjectInspector to analyze the internal structure of the row object and also the structure of the individual columns. ObjectInspector provides a uniform way to access complex objects that can be stored in  
multiple formats in the memory, including:  
• Instance of a Java class (Thrift or native Java)  
• A standard Java object (we use java.util.List to represent Struct and Array, and use  
java.util.Map to represent Map)  
• A lazily-initialized object (For example, a Struct of string fields stored in a single Java string object with starting offset for each field) A complex object can be represented by a pair of ObjectInspector and Java Object. The ObjectInspector not only tells us the structure of the Object, but also gives us ways to access  
the internal fields inside the Object.  
  
create table:  
  
create table if not exists customers.employees(  
name String comment 'employee name'  
,salary float comment 'employee salary',  
subordinates Array<string> comment 'name of subordinates',  
deductions map<String,float> comment 'keys are deduction names values are percentages',  
address struct<street:String,city:String,state:string,zipcode:int> comment 'Home address')  
comment 'Description of the table'  
tblproperties('creator'='me','created at'='01-20-15 2:59:00');  
   
load data local inpath '/home/cloudera/Desktop/employees' into table customers.employees;  
  
Copy Schema  
  
create table if not exists customers.employeescopy like customers.employees;  
  
–schema will be copied  
  
SerDe  
  
The Serializer/Deserializer or SerDe for short is used to parse the bytes of a record into columns or fields, the deserialization process. It is also used to create those record bytes (i.e., serialization)  
  
Built-in SerDes  
Avro (Hive 0.9.1 and later)  
ORC (Hive 0.11 and later)  
RegEx  
Thrift  
Parquet (Hive 0.13 and later)  
CSV (Hive 0.14 and later)  
  
Serde is a library built on top of Hadoop API. Serializer, Deserializer instructs HIVE on how to process a record.  
HIVE enables semi-structured or unstructured records to be processed also.  
Serde will deseralize data from file to object so that it can be queried using SQL syntax.  
  
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 customserde  
  
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  
  
    An RPC(Remote Procedure Call) system integrated into Hive.  
    Remote processes can send Hive statements to Hive through Thrift.  
    HIVE thrift allows access to HIVE over a single port.  
    Thrift is a software framework for scalable cross language services development.  
    Thrift allows clients using other languages to programmatically access HIVE remotely.  
    A Thrift service provides remote access from other processes.  
    Access using JDBC and ODBC are provided, too. They are implemented on top of the Thrift service  
  
MetaStore  
  
The service that maintains “metadata" information, such as table schemas.  
  
Hive requires this service to be running.  
  
support. By default, it uses a built-in Derby SQL server, which provides limited, single-process SQL for hive production environment,metastore service should run in an isolated jvm.  
Hive processes can communicate with metastore services using thrift Hive metastore data is persisted in acid database such as db/mysql In hive-default.xml.template file we can set name to store in local or warehouse  
<name> hive.metastore.warehouse.dir</name> <name>hive.metastore.local</name>  
  
Hive metastore is a database that stores metadata about your Hive tables (eg. table name, column names and types, table location, storage handler being used, number of buckets in the table, sorting columns if any, partition columns if any, etc.). When you create a table, this metastore gets updated with the information related to the new table which gets queried when you issue queries on that table.  
  
Avro  
  
Avro is a serialization format developed to address some of the common problems associated with evolving other serialization formats.  
  
Some of the benefits are:  
rich data structures, fast binary format, support for remote procedure calls, and built-in schema evolution  
JSON  
JSON (JavaScript Object Notation) is a lightweight data serialization format used commonly in web-based applications.  
HDFS  
(HDFS) A distributed, resilient file system for data storage (optimized for scanning large contiguous blocks of data on hard disks.) Distribution across a cluster provides horizontal scaling of data storage.  
  
Blocks of HDFS files are replicated across the cluster (by default, three times) to prevent data loss when hard drives or whole servers fail.  
Modes  
Strict—these are used to protect large data in partitioned tables.Instead of extracting all data from partitioned table strict mode is safety measure which allows queries with where cause only on partitones  
  
Set hive.mapred.mode=strict;  
Select \* from employee where Nonstrict  
   
Hive commands  
-d a=b –define a=b  
-e –sql from command line  
Hive –e “select \* from"  
-f<filename>-sql from files  
-H –help  
-h<hostname>connects to hive server on remote host  
--hiveconf<property=value>  
-i<filename> initialization sql file  
-p<port> connecting to hive server on port number  
-s/--silent mode in interactive shell  
-v verbose  
  
Job  
In the Hadoop context, a job is a self-contained workflow submitted to MapReduce.  
  
It encompasses all the work required to perform a complete calculation, from reading input to generating output.  
  
Job Tracker  
The MapReduce JobTracker will decompose the job into one or more tasks for distribution and execution around the cluster.  
  
The top-level controller of all jobs using Hadoop’s MapReduce.  
The JobTracker accepts job submissions, determines what tasks to run and where to run them, monitors their execution, restarts failed tasks as needed, and provides a web console for monitoring job and task execution,  
viewing logs, etc  
Task  
  
In the MapReduce context, a task is the smallest unit of work performed on a single cluster node, as part of an overall job.  
By default each task involves a separate JVM process.  
Each map and reduce invocation will have its own task.  
  
Input Format  
  
The input format determines how input streams, usually from files, are split into records.  
  
A SerDe handles parsing the record into columns. A custom input format can be specified when creating a table using the INPUTFORMAT clause.  
  
Output Format  
  
The output format determines how records are written to output streams, usually to files.  
A SerDe handles serialization of each record into an appropriate byte stream.  
  
A custom output format can be specified when creating a table using the OUTPUTFORMAT clause.  
  
The output format for the default STORED AS TEXTFILE specification is implemented by the Java object named org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat  
Partitioner  
  
Partitioner controls the partitioning of the keys of the intermediate map-outputs. The key (or a subset of the key) is used to derive the partition, typically by a hash function. The total number of partitions is the same as the number of reduce tasks for the job.  
Metastore  
  
hive stores schema and other Hive uses to store table schemas and other metadata for hive production environment,metastore service should run in an isolated jvm.  
Hive processes can communicate with metastore services using thrift Hive metastore data is persisted in acid database such as db/mysql In hive-default.xml.template file we can set name to store in local or warehouse  
<name> hive.metastore.warehouse.dir</name>  
<name>hive.metastore.local</name>  
  
storage formats  
Text/sequencefile/rc file  
Rc-recordcolumn file  
Crate table xx(a int) stored as rc file;  
--compression  
Set hive.exec.compress.output=true  
Set mapred.output.compression.codec=;  
--import  
Insert overwrite table xx Select \* from  
For compression and serialization we use rcfile and sequencefile  
  
Following classes are used by Hive to read and write HDFS files  
•TextInputFormat/HiveIgnoreKeyTextOutputFormat: These 2 classes read/write data in plain  
text file format.  
•SequenceFileInputFormat/SequenceFileOutputFormat: These 2 classes read/write data in  
hadoop SequenceFile format.  
  
Hive natively supports text file format, however hive also has support for other binary formats. Hive supports Sequence, Avro, RCFiles.  
  
Sequence files :-General binary format. splittable, compressible and row oriented.a typical example can be. if we have lots of small file, we may use sequence 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 Sequence 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 subsequently..  
Joins  
  
Innerjoins-returns all rows when there is atleast one match in both tables  
  
Leftouterjoin-returns all rows from left table and matched rows from right table  
  
Right join-returns all rows from right table and matched rows from left  
  
Full join-returns all rows  
  
Map join-join can be performed as map only job  
One of the table being joined should be small  
  
Select /\*mapjoin\*/ a.key,a.value from a join b on a.key=b.key  
  
Oredrby/sortby/distributedby/clusterby  
Orderby-ascending or desc pushing all dat through one reducer  
  
Sortby-orders data at each of n reducers but each reducer can receive overlapping ranges of data.you end up with one or more sotrted files with overlapping ranges  
  
Distributeby-ensures each of n reducers gets non overlapping ranges of x,but doesnot sort the o/p of each reducer.u end with n or more unsorted files with non overlapping ranges  
  
Clusterby-ensures each of n reducers gets non overlapping ranges,then sorts by those changes at reducers.this gives global ordering and is same as doing distributed by and sort by.you end up with n or more sorted files with non overlapping ranges  
  
Cluster by is more scalable than order by sampling  
Tablesample sample ur table It can return subsets of buckets bucket smaplingOr hdfs blocks block sampling or only first n elements from each ip split compression To optimized memory streams are compressed using CODEC which is specified as table property.  
RC file, ORC, Sequence, LZO, can be compressed.