**Hive Interview Tips:**

**----------------------------**

**What is Hive?**

[Hive](http://hive.apache.org/) is a data warehouse system for Hadoop that facilitates easy data summarization, ad-hoc queries, and the analysis of large datasets stored in Hadoop compatible file systems.

Hive was originally developed at Facebook. It’s now a [Hadoop](http://bigdataanalyticsnews.com/hadoop-interview-questions-setting-hadoop-cluster/" \o "Hadoop Interview Questions – Setting up Hadoop Cluster!) subproject with many contributors. Users need to concentrate only on the top level hive language rather than java map reduce programs. One of the main advantages of Hive is its SQLish nature. Thus it leverages the usability to a higher extend.

A hive program will be automatically compiled into map-reduce jobs executed on Hadoop. In addition, HiveQL supports custom map-reduce scripts to be plugged into queries.

Hive example:

selecting the employee names whose salary more than 100 dollars from a hive table called tbl\_employee.

SELECT employee\_name FROM tbl\_employee WHERE salary > 100;

Users are excited to use Hive since it is very similar to SQL.

**What are the types of tables in Hive?**

There are two types of tables.

1. Managed tables.

2. External tables.

Only the drop table command differentiates managed and external tables. Otherwise, both types of tables are very similar.

**Does Hive support record level Insert, delete or update?**

[Hive](http://en.wikipedia.org/wiki/Apache_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.

**What kind of datawarehouse 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

1) Relatively static data is analyzed,

2) Fast response times are not required, and

3) When the data is not changing rapidly.

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.

**How can the columns of a table in hive be written to a file?**

By using awk command in shell, the output from HiveQL (Describe) can be written to a file.

hive -S -e “describe table\_name;” | awk -F” ” ’{print 1}’ > ~/output.

**CONCAT function in Hive with Example?**

CONCAT function will concat the input strings. You can specify any number of strings separated by comma.

Example:

CONCAT (‘Hive’,'-’,'performs’,'-’,'good’,'-’,'in’,'-’,'Hadoop’);

Output:

Hive-performs-good-in-Hadoop

So, every time you 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.

CONCAT\_WS (‘-’,'Hive’,'performs’,'good’,'in’,'Hadoop’);

Output: Hive-performs-good-in-Hadoop

**REPEAT function in Hive with example?**

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

Example:

REPEAT(‘Hadoop’,3);

 Output:

HadoopHadoopHadoop.

Note: You can add a space with the input string also.

**TRIM function in Hive with example?**

TRIM function will remove the spaces associated with a string.

Example:

TRIM(‘  Hadoop  ‘);

Output:

Hadoop.

Note: If you want to remove only leading or trialing spaces then you can specify the below commands respectively.

LTRIM(‘  Hadoop’);

RTRIM(‘Hadoop  ‘);

REVERSE function in Hive with example?

REVERSE function will reverse the characters in a string.

**Difference between external table and internal table in HIVE ?**

Hive has a relational database on the master node it uses to keep track of state. For instance, when you CREATE TABLE FOO(foo string) LOCATION 'hdfs://tmp/';, this table schema is stored in the database. If you have a partitioned table, the partitions are stored in the database(this allows hive to use lists of partitions without going to the filesystem and finding them, etc). These sorts of things are the 'metadata'.  
When you drop an internal table, it drops the data, and it also drops the metadata.  
When you drop an external table, it only drops the meta data. That means hive is ignorant of that data now. It does not touch the data itself.

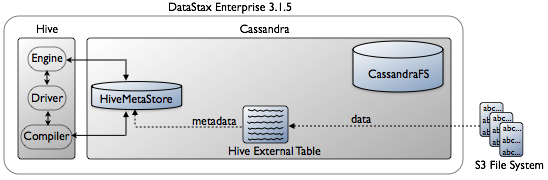
**What is the difference between Cassandra, Pig and Hive?**

Cassandra can work together with Hive and they are not complementary. Cassandra is good for streaming data.

Pig is good for ETL job and great for unstructured data. If you are doing lots of statistical analysis and operations, Hive score over Pig. Hive is good for structured data and, in general compete with SQL.

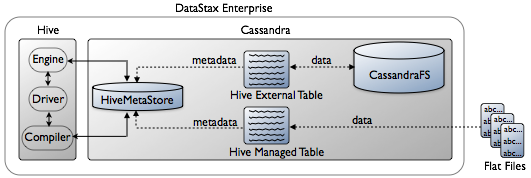
**What is the difference between External and Internal Table in Hive?**

Hive has a relational database on the master node it uses to keep track of state. For instance, when you CREATE TABLE FOO(foo string) LOCATION 'hdfs://tmp/';, this table schema is stored in the database. If you have a partitioned table, the partitions are stored in the database(this allows hive to use lists of partitions without going to the filesystem and finding them, etc). These sorts of things are the 'metadata'.  
When you drop an internal table, it drops the data, and it also drops the metadata.  
When you drop an external table, it only drops the meta data. That means hive is ignorant of that data now. It does not touch the data itself.

[](http://www.datastax.com/docs/_images/hive_tableS3.png)

Use EXTERNAL tables when:

* The data is also used outside of Hive. For example, the data files are read and processed by an existing program that doesn't lock the files. Data needs to remain in the underlying location even after a DROP TABLE. This can apply if you are pointing multiple schemas (tables or views) at a single data set or if you are iterating through various possible schemas.
* You want to use a custom location such as ASV,
* Hive should not own data and control settings, dirs, etc., you have another program or process that will do those things.
* You are not creating table based on existing table (AS SELECT).

[](http://www.datastax.com/docs/_images/hive_tables.png)

Use INTERNAL tables when:

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

**What is sample Query in Hive?**

hive> Show Tables;

hive> Show databases;

hive> create table sample (firstName STRING, lastName STRING , id INT)

ROW FORMAT

DELIMITED FIELDS

TERMINATED BY ""

LINES TERMINATED BY "\n'

STORED AS TEXTFILE;

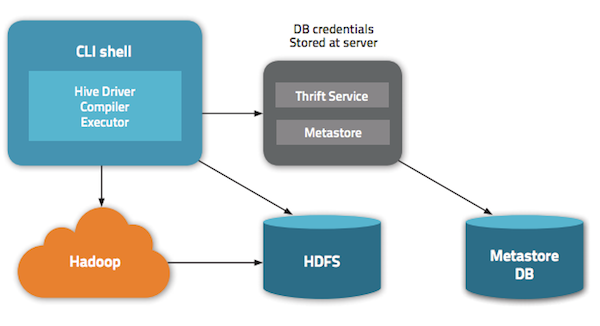
hive> Describe sample;

**What is Hive Metastore?**

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. By default, it is stored in the local machine in the derby database (namenode). It can be kept on some shared machine if multiple users are accessing it.

[](http://blog.cloudera.com/wp-content/uploads/2013/07/hiveserver2.png)

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

Whenever you 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.

**What are the different collection type in Hive?**

Database: It separates the table and other data unit from the naming confliction.  
Table: Homogenious data which has the same schema Partition: Each table can have one ore more partition key wihch determines how the data is stored. It also allows the user to efficiently identify the rows that satisfy a certain criteria. Therefore if you run analysis, you run the query on certain partition resulting higher latency. Also, partitition is based on certain words so it is the duty of the user that the relevant data goes to relevant partition.   
Buckets:Data in each partition can be divided into buckets. This is for efficient sampling. For example, in page-view table, there could be bucket or cluster based on the user# other than the partition column. This is good for the efficient sampling of the data.

**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 PostGresSQL.

**Is multiline comment supported in Hive Script ?**

No.

**If you run hive as a server, what are the available mechanism for connecting it from application?**

There are following ways by which you can connect with the Hive Server:  
1. Thrift Client: Using thrift you can call hive commands from a various programming languages e.g. C++, Java, PHP, Python and Ruby.  
2. JDBC Driver : It supports the Type 4 (pure Java) JDBC Driver  
3. ODBC Driver: It supports ODBC protocol.

**What is SerDe in Apache Hive ?**

A SerDe is a short name for a Serializer Deserializer. Hive uses SerDe (and FileFormat) 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.

**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.

 **Give examples of the SerDe classes whihc hive uses to Serializa and Deserilize data?**

Hive currently use these SerDe classes to serialize and deserialize data:  
• MetadataTypedColumnsetSerDe: This SerDe is used to read/write delimited records like CSV, tab-separated control-A separated records (quote is not supported yet.)  
• ThriftSerDe: This SerDe is used to read/write thrift serialized objects. The class file for the Thrift object must be loaded first.  
• DynamicSerDe: This SerDe also read/write thrift serialized objects, but it understands thrift DDL so the schema of the object can be provided at runtime. Also it supports a lot of different protocols, including TBinaryProtocol, TJSONProtocol, TCTLSeparatedProtocol (which writes data in delimited records).

**How do you write your own custom SerDe ?**

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

**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 inmultiple 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.

**What is the functionality of Query Processor in Apached 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.  
  
**What are some useful videos on Hive?**

[Rethinking the Data Warehouse with Hadoop and Hive - Ashish Thusoo](http://vimeo.com/7111203) from [Cloudera](http://vimeo.com/cloudera) on [Vimeo](https://vimeo.com/).

**SOME BASIC HIVE QUERIES**

**//////////////WIKI SQL** DROP TABLE IF EXISTS wiki; CREATE EXTERNAL TABLE wiki(projectname STRING,pagename STRING,pageview INT,pagesizeINT)

ROW FORMAT

DELIMITED FIELDS

TERMINATED BY ' '

LINES TERMINATED BY  '\n'

STORED AS TEXTFILE LOCATION '/user/training/wikidata';

DROP TABLE IF EXISTS wiki1;

CREATE  TABLE wiki1(pagename STRING,sum INT);

INSERT OVERWRITE TABLE wiki1

SELECT pagename,sum(pageview) FROM wiki

WHERE projectname='en' group by pagename; Select \* from wiki1 order by sum DESC limit 20

**////////////////WORD COUNT PROBLEM**  
DROP DATABASE  IF EXISTS documents CASCADE; CREATE DATABASE documents; USE documents; CREATE TABLE docs(words string); LOAD DATA LOCAL INPATH  '/home/training/training\_material/data/shakespeare'  INTO TABLE docs; CREATE TABLE word\_count AS

SELECT word, count(\*) AS count FROM

(SELECT explode(split(words, '\\W+')) AS word FROM docs) w

GROUP BY word; SELECT \* FROM word\_count limit 100;

**///////////CREATING EXTERNAL TABLES**

DROP TABLE IF EXISTS WIKI2; CREATE EXTERNAL TABLE location\_table( location\_code STRING,

location\_name STRING)

ROW FORMAT

DELIMITED FIELDS TERMINATED by ','

LINES TERMINATED by '\n'

LOCATION '/home/training/training\_material/data/ReduceAndMapSideJoin/us\_states.csv';  
  
**////////////////USING PARTITIONING**

DROP TABLE  IF EXISTS wikisample; CREATE TABLE wikisample ( project\_name STRING COMMENT 'Represents page name of the record',

page\_name STRING COMMENT 'Represents project name of the record',

page\_count INT COMMENT 'Number of times the page has been visited,

page\_size INT COMMENT " size of the page')

PARTITIONED BY (pagename STRING); LOAD DATA LOCAL INPATH '/home/training/training\_material/data/wikisample' INTO TABLE wikisample partition (projectname='en');

**///////////////DYNAMIC PARTITIONING INSERTS**

set hive.exec.dynamic.partition=true;

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

set hive.exec.max.dynamic.partitions.pernode=1000;

DROP TABLE  IF EXISTS wikisample; CREATE TABLE wikisample ( project\_name STRING COMMENT 'Represents page name of the record',

page\_name STRING COMMENT 'Represents project name of the record',

page\_count INT COMMENT 'Number of times the page has been visited',

page\_size INT COMMENT 'size of the page'); LOAD DATA LOCAL INPATH '/home/training/training\_material/data/wikisample' INTO TABLE wikisample; DROP TABLE IF EXISTS wiki; CREATE TABLE wiki(

project\_name STRING COMMENT 'Represents page name of the record',

page\_name STRING COMMENT 'Represents project name of the record',

page\_count INT COMMENT 'Number of times the page has been visited'

)

PARTITIONED BY (projectname STRING);

INSERT INTO TABLE wiki

PARTITION (projectname)

SELECT w.project\_name,w.page\_name,w.page\_count

FROM wikisample w;