## Is it possible to use same meta store by multiple users in case of embedded Hive, if no then why?

No, it is not possible to use metastore in sharing mode. It is recommended to use standalone “real” database like MySQL or PostGresSQL. Hive's Metastore is configured to store metadata locally in an embedded Apache Derby database. This configuration only allows a single user to access the Metastore at a time. An embedded metastore database is mainly used for unit tests. Only one process can connect to the metastore database at a time, so it is not really a practical solution but works well for unit tests.

## What is SerDe in 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.

## What is the functionality of query processor in Apache Hive?

Query processor in Apache Hive converts the SQL to a graph of MapReduce jobs with the execution time framework so that the jobs can be executed in the order of dependencies. The various components of a query processor are-

Parser

Semantic Analyser

Type Checking

Logical Plan Generation

Optimizer

Physical Plan Generation

Execution Engine

Operators

UDF’s and UDAF’s

## How can Hive avoid MapReduce?

select \* from table;

This query needs only read data from HDFS. So far neither requires any map or reduce phases.

## What are the types of table in Hive?

There are two types of tables in Hive, one is Managed table and second is external table.

## Does Hive support record level insert, delete or update?

With the addition of transactions in Hive 0.13 it is now possible to provide full ACID semantics at the row level, so that one application can add rows while another reads from the same partition without interfering with each other.

## What are the binary storage formats supported in Hive?

Among the different storage file formats that are used in hive, the default and simplest storage file format is the TEXTFILE.

TEXTFILE

The data in a TEXTFILE is stored as plain text, one line per record. The TEXTFILE is very useful for sharing data with other tools and also when you want to manually edit the data in the file. However the TEXTFILE is less proficient when compared to the other formats.

SEQUENCE FILE

In sequence files the data is stored in a binary storage format consisting of binary key value pairs. A complete row is stored as single binary value. Sequence files are more compact than text and fit well the map-reduce output format. Sequence files do support block compression and can be compressed on value, or block level, to improve its IO profile further.

SEQUENCEFILE is a standard format that is supported by Hadoop itself and is good choice for Hive table storage especially when you want to integrate Hive with other technologies in the Hadoop ecosystem.

RCFILE OR RECORD COLUMNAR FILE

The RCFILE stores columns of a table in a record columnar format rather than row oriented fashion and provides considerable compression and query performance benefits with highly efficient storage space utilization.

RC file format is more useful when tables have large number of columns but only few columns are typically retrieved.

The RCFile combines multiple functions to provide the following features

Fast data storing

Improved query processing,

Optimized storage space utilization

Dynamic data access patterns.

Compressed RCFile reduces the IO and storage significantly over text, sequence file, and row formats. Compression on a column base is more efficient here since it can take advantage of similarity of the data in a column.

ORC FILE OR OPTIMIZED ROW COLUMNAR FILE

ORCFILE stands for Optimized Row Columnar File and it’s a new Hive File Format that was created to provide many advantages over the RCFILE format while processing data. The ORC File format comes with the Hive 0.11 version and cannot be used with previous versions.

Lightweight indexes are included with ORC file to improve the performance.

Also it uses specific encoders for different column data types to improve compression further, e.g. variable length compression on integers

ORC stores collections of rows in one file and within the collection the row data is stored in a columnar format allowing parallel processing of row collections across a cluster.

ORC files compress better than RC files, enabling faster queries.

## What is the difference between external table and internal table in Hive?

The difference is, when you drop a table, if it is managed table hive deletes both data and Meta data, if it is external table Hive only deletes metadata.

Few things about these two:

1. Table Creation

By default, it is Managed table.

If you want to create an external table, you will use external keyword.

For example assume you have emp.csv file under directory /data/employee

To create a managed table we use normal syntax like below

create table managedemp(col1 datatype,col2 datatype, ....) row format delimited fields terminated by 'delimiter character' location '/data/employee'

But to create external table, we use external keyword like below

create external table managedemp(col1 datatype,col2 datatype, ....) row format delimited fields terminated by 'delimiter character' location '/data/employee'

2. Differentiation

How do you check whether existing table is managed or external table?

To check that we use describe command like below

describe formatted tablename;

It displays complete meta data of a table.you will see one row called table type which will display either MANAGED\_TABLE OR EXTERNAL\_TABLE

for example if it is managed table ,you will see

Table Type: MANAGED\_TABLE

if it is external table ,you will see

Table Type: EXTERNAL\_TABLE

3. Drop

As I already said if you drop a managed table both data and Meta data will be deleted

If you drop an external table only Meta data is deleted, external table is a way to protect data against accidental drop commands.

You can check this by below process.

use describe formatted tablename command and it gives location details like below.

Location: hdfs://namnodeip:portno/data/employee

After dropping the table if you use

hadoop fs -ls hdfs://namnodeip:portno/data/employee command

You should get no such file or directory exits in case of managed table.

Or you should get contents of that directory in case of external table.

And the last, try to use external table in your project, once you drop it, do not forget to remove directory if you do not need it anymore.