1. No, Its not possible to use same embedded metastore by multiple users because embedded metastore database is mainly used for unit test and only one process can be able to access and connect the metastore database at a time.

2: Serde is a library which is built in to the Hadoop API. Hive uses Files systems like HDFS or any other storage (FTP) to store data, data here is in the form of tables (which has rows and columns).SerDe - Serializer, Deserializer instructs hive on how to process a record (Row). Hive enables semi-structured (XML, Email, etc) or unstructured records (Audio, Video, etc) to be processed also. For Example If there is 1000 GB worth of RSS Feeds (RSS XMLs). In order to ingest those to a location in HDFS, we need to write a custom SerDe based on XML structure so that Hive knows how to load XML files to Hive tables or other way around.

3: 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.

4: If we set the property hive.exec.mode.local.auto to true then hive will avoid mapreduce to fetch query results.

5: Two tables in hive:

a. Internal tables

b. External table

6. By default hive does not support row level update, insert and delete. But in later version of Hive 1.4, these supports are available.

To achieve this in 1.4 version, default values must be changed like:

hive.enforce.bucketing-true

hive.support.concurrency-true

hive.exec.dynamic.partition.mode-nonstrict

hive.txn.manager-org.apache.hadoop.hive.ql.lockmgr.DbTxnManager

hive.compactor.initiator.on-true

hive.compactor.worker.thread- a positive number on at least one instance of thrift metastore service

7: Different binary storage format supported in Hive are:

Text file

Sequence File

RC file

ORC file

8: For External Tables -

External table stores files on the HDFS server but tables are not linked to the source file completely.When we delete an external table the file still remains on the HDFS server.As an example if we create an external table called “table\_test” in HIVE using HIVE-QL and link the table to file “file”, then deleting “table\_test” from HIVE will not delete “file” from HDFS. External table files are accessible to anyone who has access to HDFS file structure and therefore security needs to be managed at the HDFS file/folder level. Meta data is maintained on master node and deleting an external table from HIVE, only deletes the metadata not the data/file.

For Internal Tables-

Stored in a directory based on settings in hive.metastore.warehouse.dir, by default internal tables are stored in the following directory “/user/hive/warehouse” you can change it by updating the location in the config file . Deleting the table deletes the metadata & data from master-node and HDFS respectively. Internal table file security is controlled solely via HIVE. Security needs to be managed within HIVE, probably at the schema level (depends on organisation to organisation).