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

Ans : No, it is not possible to use metastore in sharing mode. It is recommended to use standalone "real" database like MySQL or PostGresSQL.

**2. What is SerDe in Hive?**

Ans : Apache Hive uses SerDe (and FileFormat) to read and write data from tables.ASerDe is a short name for a Serializer Deserializer. An important concept behindHive is that it DOES NOT own the Hadoop File System (HDFS) format that data is stored in.

**3. What is the functionality of query processor in Apache Hive?**

Ans : The following are the main components of the Hive Query Processor:

* Parse and SemanticAnalysis (ql/parse) - This component contains the code for parsing SQL, converting it into Abstract Syntax Trees, converting the Abstract Syntax Trees into Operator Plans and finally converting the operator plans into a directed graph of tasks which are executed by Driver.java.
* Optimizer (ql/optimizer) - This component contains some simple rule based optimizations like pruning non referenced columns from table scans (column pruning) that the Hive Query Processor does while converting SQL to a series of map/reduce tasks.
* Plan Components (ql/plan) - This component contains the classes (which are called descriptors), that are used by the compiler (Parser, SemanticAnalysis and Optimizer) to pass the information to operator trees that is used by the execution code.
* MetaData Layer (ql/metadata) - This component is used by the query processor to interface with the MetaStore in order to retrieve information about tables, partitions and the columns of the table. This information is used by the compiler to compile SQL to a series of map/reduce tasks.
* Map/Reduce Execution Engine (ql/exec) - This component contains all the query operators and the framework that is used to invoke those operators from within the map/reduces tasks.
* Hadoop Record Readers, Input and Output Formatters for Hive (ql/io) - This component contains the record readers and the input, output formatters that Hive registers with a Hadoop Job.
* Sessions (ql/session) - A rudimentary session implementation for Hive.
* Type interfaces (ql/typeinfo) - This component provides all the type information for table columns that is retrieved from the MetaStore and the SerDes.
* Hive Function Framework (ql/udf) - Framework and implementation of Hive operators, Functions and Aggregate Functions. This component also contains the interfaces that a user can implement to create user defined functions.
* Tools (ql/tools) - Some simple tools provided by the query processing framework. Currently, this component contains the implementation of the lineage tool that can parse the query and show the source and destination tables of the query.

**4. How can Hive avoid MapReduce?**

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

**5. What are the types of table in Hive?**

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

**6. Does Hive support record level insert, delete or update?**

Ans : No

**7. What are the binary storage formats supported in Hive?**

Ans : RCFile and SequenceFile

**8. What is the difference between external table and internal table in Hive?**

Ans : For instance, when you CREATE TABLE FOO(foo string) LOCATION 'hdfs://tmp/'; , this table schema is stored in the database. ... 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.