Below are the components of HIVE which take part in execution of HIVE operations-

Mapreduce + HDFS

**Hadoop cluster**

EXECUTION ENGINE

METASTORE

COMPILER

DRIVER

**UI**

9

1 6

8 7

2 5 H

3

4

1. User Interface (UI) calls the execute interface to the Driver.
2. The driver creates a session handle for the query. Then it sends the query to the compiler to generate an execution plan.
3. The compiler needs the metadata. So it sends a request for getMetaData. It stores all the structure information of the various tables and partitions in the warehouse including column and column type information, the serializers and deserializers necessary to read and write data and the corresponding HDFS files where the data is stored. Thus receives the sendMetaData request from Metastore.
4. Now compiler uses this metadata to type check the expressions in the query. It also parses the query. The compiler generates the plan which is DAG(Directed Acyclic Graph) of stages with each stage being either a [map/reduce job](http://data-flair.training/blogs/hadoop-mapreduce-job-execution-flow/), a metadata operation or an operation on [HDFS](http://data-flair.training/blogs/apache-hadoop-hdfs-introduction-tutorial/). The plan contains map operator trees and a reduce operator tree for map/reduce stages.
5. Now execution engine submits these stages to appropriate components. After in each task the deserializer associated with the table or intermediate outputs is used to read the rows from HDFS files. Then pass them through the associated operator tree.
6. Once it generates the output, write it to a temporary HDFS file through the serializer/Execution engine. The Execution Engine also executes the execution plan created by the compiler. Now temporary file provides the subsequent map/reduce stages of the plan. Then move the final temporary file to the table’s location for DML operations. The execution engine manages the dependencies between these different stages of the plan and executes these stages on the appropriate system components.
7. Now for queries, execution engine directly read the contents of the temporary file from HDFS as part of the fetch call from the Driver.
8. After all above operations the data/output passes via DRIVER and gets thrown to UI.