Problem Statement:

Explain in brief

● Differences between HBASE and HDFS.

● List and explain the main components of HBASE.

● Does Hbase support sql?

1. Differences between HBASE and HDFS:

Hadoop is, essentially, HDFS (Hadoop Distributed File System) and MapReduce. HDFS is meant for storing massive amounts of data across a distributed system. Technically speaking, your question should be on the difference between HBase and HDFS.

HDFS is a distributed file system and has the following properties:   
1. It is optimized for streaming access of large files. You would typically store files that are in the 100s of MB upwards on HDFS and access them through MapReduce to process them in batch mode.  
2. HDFS is optimized for use cases where you write once and read many times like in the case of production logs. You can append to files in some of the recent versions but that is not a feature that is very commonly used. There is no concept of random writes.  
3. HDFS doesn’t do random reads very well.

HBase is a non-relational database that can run *on top of Hadoop* and provides you random data access/querying capabilities. HDFS, by itself has no support for reads/writes at random location.

HBase on the other hand is a distributed column oriented database. The filesystem of choice typically is HDFS owing to the tight integration between HBase and HDFS. Having said that, it doesn’t mean that HBase can’t work on any other filesystem. It’s just not proven in production and at scale to work with anything except HDFS.  
HBase provides you with the following:  
1. It gives you the ability to do random read/writes on your data which HDFS doesn’t allow you to.  
2. HBase stores data in the form of key value pairs in a columnar fashion. HBase provides a flexible data model.  
3. Fast scans across tables.  
4. Scale in terms of writes as well as total volume of data.

Another primary difference would be the way data is stored in the two. HBase stores data as key/value pairs as in a column database (something similar to Cassandra DB) while data, in HDFS is stored as flat files.

To put it simply, HBase is an extension for the Hadoop environment that allows you to quickly read/write data.

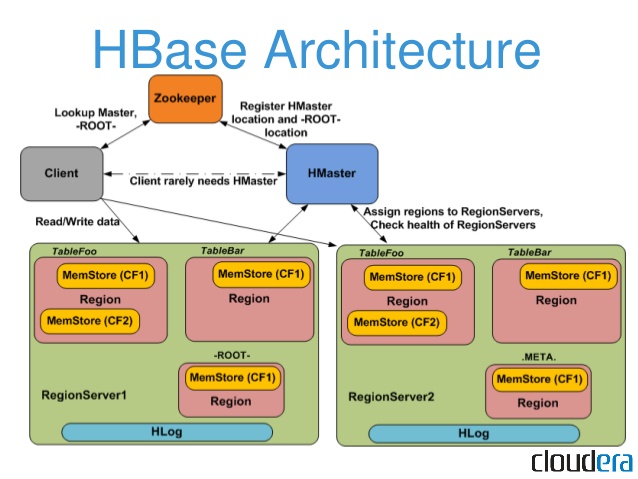
2. List and explain the main components of HBASE:

**Need for HBase**

Apache Hadoop has gained popularity in the big data space for storing, managing and processing big data as it can handle high volume of multi-structured data. However, Hadoop cannot handle high velocity of random writes and reads and also cannot change a file without completely rewriting it. HBase is a NoSQL, column oriented database built on top of hadoop to overcome the drawbacks of HDFS as it allows fast random writes and reads in an optimized way. Also, with exponentially growing data, relational databases cannot handle the variety of data to render better performance. HBase provides scalability and partitioning for efficient storage and retrieval.

“Anybody who wants to keep data within an HDFS environment and wants to do anything other than brute-force reading of the entire file system [with MapReduce] needs to look at HBase.

HBase provides low-latency random reads and writes on top of HDFS. In HBase, tables are dynamically distributed by the system whenever they become too large to handle (Auto Sharding). The simplest and foundational unit of horizontal scalability in HBase is a Region. A continuous, sorted set of rows that are stored together is referred to as a region (subset of table data).  HBase architecture has a single HBase master node (HMaster) and several slaves i.e. region servers. Each region server (slave) serves a set of regions, and a region can be served only by a single region server. Whenever a client sends a write request, HMaster receives the request and forwards it to the corresponding region server.



HBase can be run in a multiple master setup, wherein there is only single active master at a time. HBase tables are partitioned into multiple regions with every region storing multiple table’s rows.

**Components of Apache HBase Architecture**

HBase architecture has 3 important components- HMaster, Region Server and ZooKeeper.

1. **HMaster**

HBase HMaster is a lightweight process that assigns regions to region servers in the Hadoop cluster for load balancing. Responsibilities of HMaster –

* Manages and Monitors the Hadoop Cluster
* Performs Administration (Interface for creating, updating and deleting tables.)
* Controlling the failover
* DDL operations are handled by the HMaster
* Whenever a client wants to change the schema and change any of the metadata operations, HMaster is responsible for all these operations.

1. **Region Server**

These are the worker nodes which handle read, write, update, and delete requests from clients. Region Server process, runs on every node in the hadoop cluster. Region Server runs on HDFS DataNode and consists of the following components –

* Block Cache – This is the read cache. Most frequently read data is stored in the read cache and whenever the block cache is full, recently used data is evicted.
* MemStore- This is the write cache and stores new data that is not yet written to the disk. Every column family in a region has a MemStore.
* Write Ahead Log (WAL) is a file that stores new data that is not persisted to permanent storage.
* HFile is the actual storage file that stores the rows as sorted key values on a disk.

1. **Zookeeper**

HBase uses ZooKeeper as a distributed coordination service for region assignments and to recover any region server crashes by loading them onto other region servers that are functioning. ZooKeeper is a centralized monitoring server that maintains configuration information and provides distributed synchronization. Whenever a client wants to communicate with regions, they have to approach Zookeeper first. HMaster and Region servers are registered with ZooKeeper service, client needs to access ZooKeeper quorum in order to connect with region servers and HMaster. In case of node failure within an HBase cluster, ZKquoram will trigger error messages and start repairing failed nodes.

ZooKeeper service keeps track of all the region servers that are there in an HBase cluster- tracking information about how many region servers are there and which region servers are holding which DataNode. HMaster contacts ZooKeeper to get the details of region servers. Various services that Zookeeper provides include –

* Establishing client communication with region servers.
* Tracking server failure and network partitions.
* Maintain Configuration Information
* Provides ephemeral nodes, which represent different region servers.

3. Does Hbase support sql?

Apache HBase is a column-oriented database management system that runs on top of HDFS. It is well suited for sparse data sets, which are common in many big data use cases. Unlike relational database systems, HBase does not support a structured query language like SQL; in fact, HBase isn’t a relational data store at all. HBase applications are written in Java much like a typical MapReduce application. HBase does support writing applications in Avro, REST, and Thrift.

An HBase system comprises a set of tables. Each table contains rows and columns, much like a traditional database. Each table must have an element defined as a Primary Key, and all access attempts to HBase tables must use this Primary Key.