Explain in brief

● Differences between HBASE and HDFS.

● List and explain the main components of HBASE.

● Does Hbase support sql?

**Differences between HBASE and HDFS:**

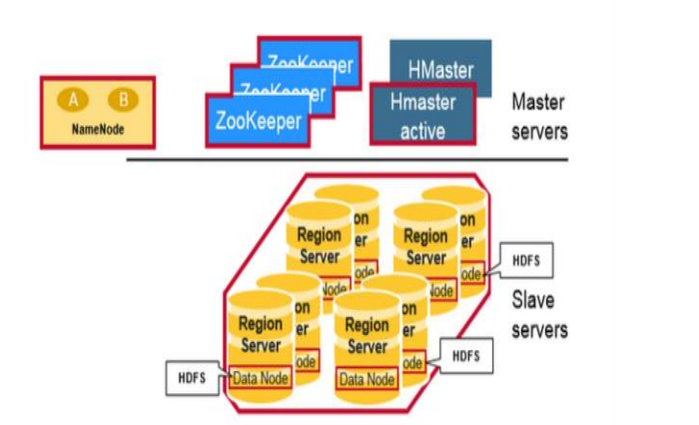
HDFS is a distributed file system and has the following properties:  
1. It is optimized for streaming access of large files. You would typically store files that are in the 100s of MB upwards on HDFS and access them through MapReduce to process them in batch mode.  
2. HDFS is optimized for use cases 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 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.

**List and explain the main components of HBASE:**

Main Components of HBase:

1. HMaster
2. Region Server
3. Zookeeper



**HMaster:**

* Assigns regions to the region servers and takes the help of Apache ZooKeeper for this task.
* Handles load balancing of the regions across region servers. It unloads the busy servers and shifts the regions to less occupied servers.
* Maintains the state of the cluster by negotiating the load balancing.
* Is responsible for schema changes and other metadata operations such as creation of tables and column families.

**Regions:**

Regions are nothing but tables that are split up and spread across the region servers

**Region Servers:**

The region servers have regions that -

* Communicate with the client and handle data-related operations.
* Handle read and write requests for all the regions under it.
* Decide the size of the region by following the region size thresholds.

When we take a deeper look into the region server, it contain regions and stores as shown below:



The store contains memory store and HFiles. Memstore is just like a cache memory. Anything that is entered into the HBase is stored here initially. Later, the data is transferred and saved in Hfiles as blocks and the memstore is flushed.

**Zookeeper:**

* Zookeeper is an open-source project that provides services like maintaining configuration information, naming, providing distributed synchronization, etc.
* Zookeeper has ephemeral nodes representing different region servers. Master servers use these nodes to discover available servers.
* In addition to availability, the nodes are also used to track server failures or network partitions.
* Clients communicate with region servers via zookeeper.
* In pseudo and standalone modes, HBase itself will take care of zookeeper.

**Does Hbase support sql?**

Hbase is a column-oriented database management system that runs on top of [HDFS](https://www.ibm.com/analytics/us/en/technology/hadoop/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. HBase isn’t a relational data store at all. HBase applications are written in Java much like a typical [MapReduce](https://www.ibm.com/analytics/us/en/technology/hadoop/mapreduce) application. HBase does support writing applications in [Avro](https://www.ibm.com/analytics/us/en/technology/hadoop/avro), REST, and Thrift.