1. Give a brief 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 files are write once files. You can append to files in some of the recent versions but that is not a feature that is very commonly used. Consider HDFS files as write-once and read-many files. There is no concept of random writes.

3. HDFS doesn't do random reads very well. HBase on the other hand is a database that stores it's data in a distributed filesystem. 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. Low latency access to small amounts of data from within a large data set. You can access single rows quickly from a billion row table.

2. Flexible data model to work with and data is indexed by the row key.

3. Fast scans across tables.

4. Scale in terms of writes as well as total volume of data.

1. List the main components of HBASE.
   1. WAL: Write Ahead Log is a file on the distributed file system. ...
   2. BlockCache: is the read cache. ...
   3. MemStore: is the write cache. ...
   4. Hfiles store the rows as sorted KeyValues on disk.
   5. Zookeeper, Catalog Tables , Master , RegionServer , Region
2. Does Hbase support sql?

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. HBaseapplications are written in Java much like a typical MapReduce application.

1. When should we use HBASE, list some of the scenarios for the same.

Use HBase when you need fault-tolerant, random, real time read/write access to data stored in HDFS. Use HBase when you need strong data consistency. HBase provides Bigtable-like capabilities on top of Hadoop. HBase’s goal is the hosting of very large tables — billions of rows times millions of columns — atop clusters of commodity hardware.  
HBase manages structured data on top of HDFS for you, efficiently using the underlying replicated storage as backing store to gain the benefits of its fault tolerance and data availability and locality.

1. What are the different modes in which Hbase can be run?

Standalone HBase

This is the default mode.In standalone mode, HBase does not use HDFS -- it uses the local filesystem instead -- and it runs all HBase daemons and a local ZooKeeper all up in the same JVM. Zookeeper binds to a well known port so clients may talk to HBase.

Distributed

Distributed mode can be subdivided into distributed but all daemons run on a single node -- a.k.a *pseudo-distributed*-- and *fully-distributed* where the daemons are spread across all nodes in the cluster,Distributed modes require an instance of the *Hadoop Distributed File System* (HDFS). See the Hadoop [requirements and instructions](http://hadoop.apache.org/common/docs/r1.1.1/api/overview-summary.html#overview_description) for how to set up a HDFS. Before proceeding, ensure you have an appropriate, working HDFS.

1. Why is zookeeper needed in Hbase?

Zookeeper, which is part of HDFS, maintains a live cluster state. ... All HBase data is stored in HDFS files. Region Servers are collocated with the HDFS DataNodes, which enable data locality (putting the data close to where it is needed) for the data served by the RegionServers

7 Hbase is a schema less database, what does it mean?

 In HBase (and other BigTable implementations) data is labeled with its types. An analogy is CSV : RDBMS is to XML : HBase.

8 What is the minimum number of column family every Hbase table should have?

Two or three

9 What is the benefit of using connection pool in Hbase?

For applications which require high-end multi-threaded access(eg web-server or application servers that may serve many applications threads in a single JVM), you can precreate an Hconnection.