1. What is the difference between memstore and hfile in HBase?

The MemStore stores updates in memory as sorted KeyValues, the same as it would be stored in an HFile. There is one MemStore per column family. The updates are sorted per column family.

Data is stored in an HFile which contains sorted key/values. When the MemStore accumulates enough data, the entire sorted KeyValue set is written to a new HFile in HDFS. This is a sequential write. It is very fast, as it avoids moving the disk drive head.

1. Describe compactions in HBase.

Hbase major compaction

HBase will automatically pick some smaller HFiles and rewrite them into fewer bigger Hfiles. This process is called minor compaction. Minor compaction reduces the number of storage files by rewriting smaller files into fewer but larger ones, performing a merge sort.

Hbase Minor compaction

Major compaction merges and rewrites all the HFiles in a region to one HFile per column family, and in the process, drops deleted or expired cells. This improves read performance; however, since major compaction rewrites all of the files, lots of disk I/O and network traffic might occur during the process. This is called write amplification.

1. List and explain the logical entities in HBase.

Searching for hbase schema design you will find materials on how you would design your schema to take advabtage of hbase including a pdf. hbase is mostly schema less but your application will manage what it knows to be a schema encoded as tables, column families, column naming, keys, versions, and values. after reading the top 3 links including the pdf, and if you have specific questions, add then as separate questions to discuss them.

1. What will happen if we do not create a row key while inserting the data?

HBase, values are always freighted with their coordinates; as a cell value passes through the system, it'll be accompanied by its row, column name, and timestamp - always.

You still need to put a lot of thought into what is the right row key for you. When you do this you need to take into account the access patterns you plan for your data and the way hbase tables are structured e.g. keys are ordered on disk lexicographically, there's a timestamp for each value that is stored, qualifiers are grouped into column families etc.

1. How can filters be applied in HBase and what are the benefits?

When reading data from HBase using Get or Scan operations, you can use custom filters to return a subset of results to the client. While this does not reduce server-side IO, it does reduce network bandwidth and reduces the amount of data the client needs to process. Filters are generally used using the Java API, but can be used from HBase Shell for testing and debugging purposes.

1. What are the data model operations in hBase?

The four primary data model operations are Get, Put, Scan, and Delete. Operations are applied via [HTable](http://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/HTable.html" \t "_top) instances.

* Get

[Get](http://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Get.html) returns attributes for a specified row. Gets are executed via [HTable.get](http://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/HTable.html" \l "get%28org.apache.hadoop.hbase.client.Get%29" \t "_top).

* Put

[Put](http://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Put.html) either adds new rows to a table (if the key is new) or can update existing rows (if the key already exists). Puts are executed via [HTable.put](http://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/HTable.html" \l "put%28org.apache.hadoop.hbase.client.Put%29" \t "_top) (writeBuffer) or [HTable.batch](http://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/HTable.html" \l "batch%28java.util.List%29" \t "_top)(non-writeBuffer).

* Scans

[Scan](http://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Scan.html) allow iteration over multiple rows for specified attributes.

### Delete

### [Delete](http://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Delete.html) removes a row from a table. Deletes are executed via [HTable.delete](http://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/HTable.html" \l "delete%28org.apache.hadoop.hbase.client.Delete%29" \t "_top).

1. How can MapReduce be used with HBase?

### The Default HBase MapReduce Splitter: When [TableInputFormat](http://hbase.apache.org/apidocs/org/apache/hadoop/hbase/mapreduce/TableInputFormat.html" \t "_top) is used to source an HBase table in a MapReduce job, its splitter will make a map task for each region of the table. Thus, if there are 100 regions in the table, there will be 100 map-tasks for the job - regardless of how many column families are selected in the Scan.

### Custom Splitters: For those interested in implementing custom splitters, see the method getSplits in [TableInputFormatBase](http://hbase.apache.org/apidocs/org/apache/hadoop/hbase/mapreduce/TableInputFormatBase.html" \t "_top). That is where the logic for map-task assignment resides.

1. What is regionserver?

RegionServers are the software processes (often called daemons) you activate to store and retrieve data in HBase (Hadoop Database). In production environments, each RegionServer is deployed on its own dedicated compute node. When you start using HBase, you create a table and then begin storing and retrieving your data.