1.What is NoSQL data base?

A **NoSQL** (originally referring to "non SQL", "non relational" or "not only SQL")**database** provides a mechanism for storage and retrieval of **data** which is modeled in means other than the tabular relations used in relational **databases**. ... **NoSQL databases** are increasingly used in big **data** and real-time web applications.

The system of engagement would need to be extremely dynamic. A traditional**database** product would prefer more predictable, structured data. A relational**database** may require vertical and, sometimes horizontal expansion of servers, to expand as data or processing requirements grow. ... **NoSQL** is not a relational**database.**

2. How does data get stored in NoSQl database?

There are 4 basic types of NoSQL databases:

1. **Key-Value Store** – It has a Big Hash Table of keys & values {Example- Riak, Amazon S3 (Dynamo)}
2. **Document-based** **Store- It**stores documents made up of tagged elements. {Example- CouchDB}
3. **Column-based Store-**Each storage block contains data from only one column, {Example- HBase, Cassandra}
4. **Graph-based**-A network database that uses edges and nodes to represent and store data. {Example- Neo4J}

3. What is a column family in HBase?

Column families are the base storage mechanism in HBase.   A HBase table is comprised of one or more column families,  each of which is stored in a separate set of regionfiles sharing a common key. A column family defines shared features to all columns that are created within them (think of it almost as a sub-table within your larger table).  
To express it in terms of an RDBMS, a column family is roughly analogous to a RDBMS table with the rowkey as a clustered primary key index.

4.How many maximum number of columns can be added to HBase table?

No Limit

5.Why columns are not defined at the time of table creation in HBase?

1. How does data get managed in HBase?

HBase contains two primary services:

**Master server**

The master server co-ordinates the cluster and performs administrative operations, such as assigning regions and balancing the loads.

**Region server**

The region servers do the real work. A subset of the data of each table is handled by each region server. Clients talk to region servers to access data in HBase.

**Regions**

Region servers manage a set of regions.

An HBase table is made up of a set of regions. Regions are the basic unit of work in HBase. It is what is used as a split by the map reduce framework. The region contains store objects that correspond to column families. There is one store instance for each column family. Store objects create one or more *StoreFiles*, which are wrappers around the actual storage file called the *HFile*.

The region also contains a **MemStore**, which is in-memory storage and is used as a write cache. Rows are written to the **MemStore**. The data in the **MemStore** is ordered. If the **MemStore** becomes full, it is persisted to an HFile on disk

To improve performance, it is important to get an even distribution of data among regions, which ensures the best parallelism in map tasks.

**HFiles**

HFiles are the physical representation of data in HBase. Clients do not read HFiles directly but go through region servers to get to the data.

HBase internally puts the data in indexed *StoreFiles* that exist on HDFS for high-speed lookups.

Everything in HBase is stored as bytes and there are no types. There is no schema since each row in HBase can have a different set of columns.

An HBase table contains *column families*, which are the logical and physical grouping of columns. There are *column qualifiers* inside of a column family, which are the columns. Column families contain columns with time stamped versions. Columns only exist when they are inserted, which makes HBase a sparse database. All column members of the same column family have the same column family prefix. Each column value is identified by a key. The *row key* is the implicit primary key. Rows are sorted by the *row key*. An HBase column can be specified by using the following format:

hbase-family:hbase-column-name

7. What happens internally when new data gets inserted into HBase table

the row key does not have a **data** type and is treated **internally** as a byte array.