# 1. What is NoSQL data base?

A *NoSQL* (originally referring to "non SQL" or "non relational") *database* provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational *databases*.

# 2. How does data get stored in NoSQl database?

It stores basically unstructured data like (articles, photos, social media data, videos, or content within a blog post) as a single document that can be easily found but isn’t necessarily categorized into fields like a relational database does.It uses Hadoop, an open-source computing and data analysis platform capable of processing huge amounts of data in the cloud.

# 3. What is a column family in HBase?

Columns in Apache HBase are grouped into column families. All column members of a column family have the same prefix. For example, the columns courses: history and courses: maths is both members of the courses column family. The colon character (:) delimits the column family from the column family qualifier. The column family prefix must be composed of printable characters. The qualifying tail, the column family qualifier, can be made of any arbitrary bytes. Column families must be declared up front at schema definition time whereas columns do not need to be defined at schema time but can be conjured on the fly while the table is up and running.  
  
Physically, all column family members are stored together on the filesystem. Because tunings and storage specifications are done at the column family level, it is advised that all column family members have the same general access pattern and size characteristics.

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

As per Apache HBase wiki Hbase will face performance issues more than 2 or 3 Column families. There is a limit to the number of column families in HBase. There is one MemStore(Its a write cache which stores new data before writing it into Hfiles) per Column Family, when one is full, they all flush.

The more you add column families there will be more MemStore created and Memstore flush will be more frequent. It will degrade the performance.

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

Columns in Apache HBase are grouped into column families. All column members of a column family have the same prefix. For example, the columns courses: history and courses: math is both members of the courses column family. The colon character (:) delimits the column family from the column family qualifier. The column family prefix must be composed of printable characters. The qualifying tail, the column family qualifier, can be made of any arbitrary bytes. Column families must be declared up front at schema definition time whereas columns do not need to be defined at schema time but can be conjured on the fly while the table is up an running.

# 6.How does data get managed in HBase?

Data in Hbase is organized into tables. Any characters that are legal in file paths are used to name tables. Tables are further organized into rows that store data. Each row is identified by a unique row key which does not belong to any data type but is stored as a bytearray. Column families are further used to group data in rows. Column families define the physical structure of data so they are defined upfront and their modification is difficult. Each row in a table has same column families. Data in a column family is addressed using a column qualifier. It is not necessary to specify column qualifiers in advance and there is no consistency requirement between rows. No data types are specified for column qualifiers; as such they are just stored as bytearrays. A unique combination of row key, column family and column qualifier forms a cell. Data contained in a cell is referred to as cell value. There is no concept of data type when referring to cell values and they are stored as bytearrays. Versioning happens to cell values using a timestamp of when the cell was written.

Indexing and sorting only happens on the row key. The concept of data types is absent and everything is stored as bytearray. Only row level atomicity is enforced so multi row transactions are not supported.

# 7.What happens internally when new data gets inserted into Hbase Table

This is what happens the first time a client reads or writes to HBase:

1. The client gets the Region server that hosts the META table from ZooKeeper.
2. The client will query the .META. Server to get the region server corresponding to the row key it wants to access. The client caches this information along with the META table location.
3. It will get the Row from the corresponding Region Server.

For future reads, the client uses the cache to retrieve the META location and previously read row keys. Over time, it does not need to query the META table, unless there is a miss because a region has moved; then it will re-query and update the cache.