What is NoSQL data base?

NoSQL is an approach to database design that can accommodate a wide variety of data models, including key-value, document, columnar and graph formats. NoSQL, which stand for "not only SQL," is an alternative to traditional relational databases in which data is placed in tables and data schema is carefully designed before the database is built. NoSQL databases are especially useful for working with large sets of distributed data.

How does data get stored in NoSQl database?

**Document databases** pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.

**Graph stores** are used to store information about networks of data, such as social connections. Graph stores include Neo4J and Graph.

**Key-value stores** are the simplest NoSQL databases. Every single item in the database is stored as an attribute name (or 'key'), together with its value. Examples of key-value stores are Riak and Berkeley DB. Some key-value stores, such as Radis, allow each value to have a type, such as 'integer', which adds functionality.

**Wide-column stores** such as Cassandra and HBase are optimized for queries over large datasets, and store columns of data together, instead of rows

3. What is a column family in HBase?

In the HBase data model columns are grouped into column families, which must be defined up front during table creation. Column families are stored together on disk, which is why HBase is referred to as a column-oriented data store.

When creating a table in HBase, the developer or administrator is required to define one or more column families using printable characters.

Generally, column families remain fixed throughout the lifetime of an HBase table but new column families can be added by using administrative commands. The official recommendation for the number of column families per table is three or less.

In addition, you should store data with similar access patterns in the same column family — you wouldn’t want a customer’s middle name stored in a separate column family from the first or last name because you generally access all name data at the same time.

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

 There is no hard limit to number of columns in HBase , we can have more than 1 million columns but usually three column families are recommended ( not more than three).

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

We cannot provide column-qualifier during table creation. This is what makes HBase schema-less. The syntax for creating tables in HBase is as follows.

create ‘<table name>’,’<column family>’

Once we have created the table, we can add rows which should belong to at-least one column-family in a table and then can belong to any column-qualifier. If the column-qualifier is not already present, it will be created.

6. How does data get managed in HBase?

Data can be managed in HBase by using the create, get, put and scan commands from the HBase shell. Data is written to the database by using put and read by using get. The scan command is used to obtain data from multiple rows in a table. Data can also be managed using the HBase C# API, which provides a client library on top of the HBase REST API. An HBase database can also be queried by using Hive. Co-processors are also available, which allow data processing in the nodes that host the database.

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

Write operation

Step 1) Client wants to write data and in turn first communicates with Regions server and then regions

Step 2) Regions contacting memstore for storing associated with the column family

Step 3) First data stores into Memstore, where the data is sorted and after that it flushes into HFile. The main reason for using Memstore is to store data in Distributed file system based on Row Key. Memstore will be placed in Region server main memory while HFiles are written into HDFS.

Read operation

Step 4) Client wants to read data from Regions

Step 5) In turn Client can have direct access to Mem store, and it can request for data.

Step 6) Client approaches HFiles to get the data. The data are fetched and retrieved by the Client.