**Assignment 9.1**

**1. What is NoSQL database?**

**2. How does data gets stored in NoSQL database?**

**3. What is column family in HBase?**

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

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

**6. How does data get managed in HBase?**

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

1. **What is NoSQL database?**

NoSQL is an approach to [database](http://searchsqlserver.techtarget.com/definition/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**](http://searchsqlserver.techtarget.com/definition/SQL)**,"** is an alternative to traditional relational databases in which data is placed in tables and data [schema](http://searchsqlserver.techtarget.com/definition/schema) is carefully designed before the database is built. NoSQL databases are especially useful for working with large sets of distributed data.

1. **How data does gets stored in NoSQL database?**

There are various NoSQL Databases. Each one uses a different method to store data. Some might use column store, some document, some graph, etc., Each database has its own unique characteristics.

MongoDB is a document store, where data is stored as ***Key: Value*** pairs in [JSON](http://json.org/) format.

An Example:

1. {
2. name: "sid",
3. phone: 1234567890,
4. address:
5. {
6. street: "1234 Some\_XYZ Pkwy" ,
7. Apt: 1001,
8. City: "Richardson",
9. State: "Texas"
10. }
11. }

[B-tree](https://en.wikipedia.org/wiki/B-tree) is used for indexing if you are using [MMAP storage Engine in Mongodb](http://stackoverflow.com/questions/29575420/mmap-storage-engine-in-mongodb) . And [B+ tree](https://en.wikipedia.org/wiki/B%2B_tree) if WiredTiger storage Engine is used.

1. **What is 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:math are both members of the courses column family. The colon character (:) delimits the column family from the . 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.

Physically, all column family members are stored together on the file system. 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.

1. **How many maximum number of columns are added to HBase table?**

* 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.
* When the MemStore accumulates enough data, the entire sorted set is written to a new HFile in HDFS. HBase uses multiple HFiles per column family, which contain the actual cells, or KeyValue instances. These files are created over time as KeyValue edits sorted in the MemStores are flushed as files to disk.
* Note that this is one reason why there is a limit to the number of column families in HBase. There is one MemStore per CF; when one is full, they all flush. It also saves the last written sequence number so the system knows what was persisted so far.
* The highest sequence number is stored as a meta field in each HFile, to reflect where persisting has ended and where to continue. On region startup, the sequence number is read, and the highest is used as the sequence number for new edits.

1. **Why column are not define at the time of table creation in HBase?**

Columns are not defined at the time of table creation in HBase because to give the specific names at the time of storing and further for accessing. Column families should not be changed often, nor should there be too many of them, so it is important to think carefully about what column families will be useful for our particular data. Each column family, however, can contain a very large number of columns. Columns are named using the format family: qualifier.

1. **How does data get managed at the Hbase table?**

The Data Model in HBase is designed to accommodate semi-structured data that could vary in field size, data type and columns. Additionally, the layout of the data model makes it easier to partition the data and distribute it across the cluster. The Data Model in HBase is made of different logical components such as Tables, Rows, Column Families, Columns, Cells and Versions.

*Tables* – The HBase Tables are more like logical collection of rows stored in separate partitions called Regions. As shown above, every Region is then served by exactly one Region Server. The figure above shows a representation of a Table.

*Rows* – A row is one instance of data in a table and is identified by a *rowkey*. Rowkeys are unique in a Table and are always treated as a byte[].

*Column Families* – Data in a row are grouped together as Column Families. Each Column Family has one more Columns and these Columns in a family are stored together in a low level storage file known as HFile. Column Families form the basic unit of physical storage to which certain HBase features like compression are applied. Hence it’s important that proper care be taken when designing Column Families in table.

The table above shows Customer and Sales Column Families. The Customer Column Family is made up 2 columns – Name and City, whereas the Sales Column Families is made up to 2 columns – Product and Amount.

*Columns* – A Column Family is made of one or more columns. A Column is identified by a Column Qualifier that consists of the Column Family name concatenated with the Column name using a colon – example: columnfamily:columnname. There can be multiple Columns within a Column Family and Rows within a table can have varied number of Columns.

*Cell* – A Cell stores data and is essentially a unique combination of *rowkey*, Column Family and the Column (Column Qualifier). The data stored in a Cell is called its value and the data type is always treated as byte[].

*Version* – The data stored in a cell is versioned and versions of data are identified by the timestamp. The number of versions of data retained in a column family is configurable and this value by default is 3.

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

When you put data into HBase, a timestamp is required. The timestamp can be generated automatically by the RegionServer or can be supplied by you. The timestamp must be unique per version of a given cell, because the timestamp identifies the version. To modify a previous version of a cell, for instance, you would issue a Put with a different value for the data itself, but the same timestamp.

HBase's behavior regarding versions is highly configurable. The maximum number of versions defaults to 1 in CDH 5, and 3 in previous versions. You can change the default value for HBase by configuring hbase.column.max.version in hbase-site.xml, either using an advanced configuration snippet if you use Cloudera Manager, or by editing the file directly otherwise.

We can also configure the maximum and minimum number of versions to keep for a given column, or specify a default time-to-live (TTL), which is the number of seconds before a version is deleted. The following examples all use alter statements in HBase Shell to create new column families with the given characteristics, but you can use the same syntax when creating a new table or to alter an existing column family. This is only a fraction of the options you can specify for a given column family.