**Assignment 9.1:**

**Problem Statement:**

1. **What is NoSQL data base?**

* NoSQL is an approach to database design that can accomodate 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.
* **NoSQL database examples:**Dozens of NoSQL data stores are available; the following are among the most popular:
* **MongoDB**. Open-source document database.
* **CouchDB**. Database that uses JSON for documents, JavaScript for MapReduce queries, and regular HTTP for an API.
* **GemFire**. Distributed data management platform providing dynamic scalability, high performance, and database-like persistence.
* **Redis**. Data structure server wherein keys can contain strings, hashes, lists, sets, and sorted sets.
* **Cassandra**. Database that provides scalability and high availability without compromising performance.
* **memcached**. Open source high-performance, distributed-memory, and object-caching system.
* **Hazelcast**. Open source highly scalable data distribution platform.
* **HBase**. Hadoop database, a distributed and scalable big data store.
* **Mnesia**. Distributed database management system that exhibits soft real-time properties.
* **Neo4j**. Open source high-performance, enterprise-grade graph database.

1. **How does data get 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.

1. **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: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 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.

1. **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 two or three column families are recommended ( not more than three).

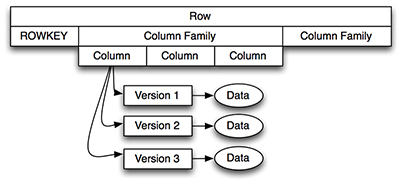
1. **Why columns are not defined at the time of table creation in HBase?**

* 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.

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

 HBase actually defines a four-dimensional data model and the following four coordinates define each cell (see Figure 1):

* Row Key: Each row has a unique row key; the row key does not have a data type and is treated internally as a byte array.
* Column Family: Data inside a row is organized into column families; each row has the same set of column families, but across rows, the same column families do not need the same column qualifiers. Under-the-hood, HBase stores column families in their own data files, so they need to be defined upfront, and changes to column families are difficult to make.
* Column Qualifier: Column families define actual columns, which are called column qualifiers. You can think of column qualifiers as the columns themselves.
* Version: Each column can have a configurable number of versions, and you can access the data for a specific version of a column qualifier.

[](javascript:popUp('/content/images/art_haines_hbases1_1/elementLinks/haines_hbase_1_1_01.jpg'))

**Figure 1.**  HBase Four-Dimensional Data Model

As shown in Figure 1, an individual row is accessible through its row key and is composed of one or more column families. Each column family has one or more column qualifiers (called “column” in Figure 1) and each column can have one or more versions. To access an individual piece of data, you need to know its row key, column family, column qualifier, and version.

1. **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 sorts the versions of a cell from newest to oldest, by sorting the timestamps lexicographically.