**Session 9 Assignment 1**

1. **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. It is 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.

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

* There are 4 types of NoSQL databases in which each store the data as below,
  + Key-Value databases: It stores are the simplest NoSQL data stores to use from an API perspective. The client can either get the value for the key, put a value for a key, or delete a key from the data store. The value is a blob that the data store just stores, without caring or knowing what's inside; it's the responsibility of the application to understand what was stored. Since key-value stores always use primary-key access, they generally have great performance and can be easily scaled.
  + Document databases: Documents are the main concept in document databases. The database stores and retrieves documents, which can be XML, JSON, BSON, and so on. These documents are self-describing, hierarchical tree data structures which can consist of maps, collections, and scalar values. The documents stored are similar to each other but do not have to be exactly the same. Document databases store documents in the value part of the key-value store; think about document databases as key-value stores where the value is examinable.
  + Column family stores: Column families are groups of related data that is often accessed together. For a Customer, we would often access their Profile information at the same time, but not their Orders. Each column family can be compared to a container of rows in an RDBMS table where the key identifies the row and the row consists of multiple columns. When a column consists of a map of columns, then we have a super column. A super column consists of a name and a value which is a map of columns. Think of a super column as a container of columns.
  + Graph Databases: Graph databases allow you to store entities and relationships between these entities. Entities are also known as nodes, which have properties. Think of a node as an instance of an object in the application. Relations are known as edges that can have properties. Edges have directional significance; nodes are organized by relationships which allow you to find interesting patterns between the nodes. The organization of the graph lets the data to be stored once and then interpreted in different ways based on relationships.

1. **What is a column family in HBase?**

* Columns in HBase are grouped into column families. All column members of a column family have the same prefix. For example, the columns courses:java and courses:spark are 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 a 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 can be added to HBase table?**

* There is no 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.

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

* The column qualifiers (columns) do not have to be defined at schema definition time because each row in HBase can have a different set of columns so they can be added on the fly while the database is up and running.

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

* The main characteristics that make HBase an excellent data management platform are fault tolerance, speed and usability.
  + Fault tolerance is provided by automatic fail-over, automatically shared and load balanced tables, strong consistency in row level operations and replication.
  + Speed is provided by almost real time lookups, in memory caching and server side processing.
  + Usability is provided by a flexible data model that allows many uses, a simple Java API and ability to export metrics.

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 byte array. 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 byte arrays. 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 byte arrays. Versioning happens to cell values using a timestamp of when the cell was written.

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

* When the client issues a Put request, the first step is to write the data to the write-ahead log, the WAL:
  + Edits are appended to the end of the WAL file that is stored on disk.
  + The WAL is used to recover not-yet-persisted data in case a server crashes.

Once the data is written to the WAL, it is placed in the Mem Store. Then, the put request acknowledgement returns to the client. When we issue a Put command, the coordinates of the data are the row, the column, and the timestamp and when data is put into HBase, a timestamp is required. The timestamp can be generated automatically by the Region Server or can be supplied by us and must be a long integer. 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, we would issue a Put command with a different value for the data itself, but the same timestamp.