ASSIGNMENT 9.1

Que 1: What is NoSQL data base?

Ans: A **NoSQL database** provides a mechanism for storage and retrieval of **data** that is modeled in means other than the tabular relations used in relational **databases**. **NoSQL databases** are increasingly used in big data and real-time web applications.

Que 2: How does data get stored in NoSQl database?

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

Que 3: What is column family in Hbase ?

Ans: A **column family** is a NoSQL object that contains **columns** of related data. It is a tuple (pair) that consists of a key-value pair, where the key is mapped to a value that is a set of **columns**. In analogy with relational databases, a **column family** is as a "table", each key-value pair being a "row".

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

Ans : 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).

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

Ans: An HBase table is made of column families which are the logical and physical grouping of column. The column is one family are stored separately from the column is another family.

A single column family contains one or more column , **Column families** must be defined at table creation time but column can be added dynamically after table creation .

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

Ans : 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.

Que 6: How does data get managed in HBase?

Ans : NoSQL databases are designed for scalability where unstructured data is spread across multiple nodes. When data volumes increase you just need to add another node to accommodate the growth. The lack of structure in NoSQL databases relaxes stringent requirements of consistency enforced in relational databases to improve speed and agility. Hbase, MongoDB and Cassandra are the three major options that provide NoSQL capabilities. The options differ in the features they provide, so the decision on which to use is informed by the workload that will be handled. The main difference between Hbase and Cassandra databases is the consistency model they implement. Cassandra implements eventual consistency which guarantees writes are available. This provides excellent write scaling but suffers a penalty when reading because for consistency in reads you have to read from many copies of data. On the other hand HBase provides a strong consistency model that excels at scaling reads but does not scale on writes as well as Cassandra does.

Hbase can run standalone on the local file system but this set up does not guarantee durability. Edits will be lost when daemons are not cleanly started and stopped. Such a set up is not suitable in a production environment but it provides a way of exploring how the database functions. Alternatively Hbase can be installed on a single or multi node cluster and use HDFS.

Que 6: What happens internally when new data gets inserted into HBase table?

Ans: Put a cell ‘value’ at specified table/row/column and optionally timestamp coordinates. To put a cell value into table ‘t1’ at row ‘r1’ under column ‘c1’ marked with the time ‘ts1’, do:

hbase> put ‘t1’, ‘r1’, ‘c1’, ‘value’, ts1
The same commands also can be run on a table reference. Suppose you had a reference t to table ‘t1’, the corresponding command would be:

hbase> t.put ‘r1’, ‘c1’, ‘value’, ts1
WAL allows updates of a database to be done in place. Another way to implement atomic updates is with shadow paging, which is not in-place. The main advantage of doing updates in-place is that it reduces the need to modify indexes and block lists

SUBMITTED BY : DIVYANI PAL