ACD\_BDD2\_Session\_17\_Assignment\_1

1. What is NoSQL data base?

Ans: NoSQL is a non-relational database management systems, different from traditional relational database management systems in some significant ways. It is designed for distributed data stores where very large scale of data storing needs (for example Google or Facebook which collects terabits of data every day for their users). These type of data storing may not require fixed schema, avoid join operations and typically scale horizontally.

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.

There are four general types of NoSQL databases:

Graph database – Based on graph theory, these databases are designed for data whose relations are well represented as a graph and has elements which are interconnected, with an undetermined number of relations between them. Examples include: Neo4j and Titan.

Key-Value store – we start with this type of database because these are some of the least complex NoSQL options. These databases are designed for storing data in a schema-less way. In a key-value store, all of the data within consists of an indexed key and a value, hence the name. Examples of this type of database include:Cassandra, DyanmoDB, Azure Table Storage (ATS), Riak, BerkeleyDB.

Column store – (also known as wide-column stores) instead of storing data in rows, these databases are designed for storing data tables as sections of columns of data, rather than as rows of data. While this simple description sounds like the inverse of a standard database, wide-column stores offer very high performance and a highly scalable architecture. Examples include: HBase, BigTable and HyperTable.

Document database – expands on the basic idea of key-value stores where “documents” contain more complex in that they contain data and each document is assigned a unique key, which is used to retrieve the document. These are designed for storing, retrieving, and managing document-oriented information, also known as semi-structured data. Examples include: MongoDB and CouchDB.

3. What is a column family in HBase?

Ans. Column families are the base storage mechanism in HBase. A HBase table is comprised of one or more column families, each of which is stored in a separate set of regionfiles sharing a common key.

To express it in terms of an RDBMS, a column family is roughly analogous to a RDBMS table with the rowkey as a clustered primary key index. A HBase table would then be a view which does a full outer join on a set of RDBMS tables which all share the same primary key (thus having a 1:1 relationship). In this analogy, HBase region files map to pages in an RDBMS.

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

Ans. There is a limit to the number of column families in HBase. There is one MemStore( it’s a write cache which stores new data before writing it into Hfiles) per Column Family, when one is full, they all flush.

The more you add column families there will be more MemStore created and Memstore flush will be more frequent. It will degrade the performance.

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

Ans. 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.

6. How does data get managed in HBase?

An HBase system comprises a set of tables. Each table contains rows and columns, much like a traditional database. Each table must have an element defined as a Primary Key, and all access attempts to HBase tables must use this Primary Key. An HBase column represents an attribute of an object; for example, if the table is storing diagnostic logs from servers in your environment, where each row might be a log record, a typical column in such a table would be the timestamp of when the log record was written, or perhaps the server name where the record originated. In fact, HBase allows for many attributes to be grouped together into what are known as column families, such that the elements of a column family are all stored together. This is different from a row-oriented relational database, where all the columns of a given row are stored together. With HBase you must predefine the table schema and specify the column families. However, it’s very flexible in that new columns can be added to families at any time, making the schema flexible and therefore able to adapt to changing application requirements.

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

Ans. While inserting data user will mention column family and col name, unique key and value. Hbase inserts the timestamp that's how it can handle multiple versions. Hbase provide very fast key access. Records are sorted in key based.

HBase Tables are divided horizontally by row key range into “Regions.”

• A region contains all rows in the table between the region’s start key and end key.

• Regions are assigned to the nodes in the cluster, called “RegionServers,” and these serve data for reads and writes.

• A region server can serve about 1,000 regions. Region assignment, DDL (create, delete tables) operations are handled by the HBase Master. A master is responsible for:

• Coordinating the region servers

• Assigning regions on startup

• Re-assigning regions for recovery or load balancing

• Monitoring all RegionServer instances in the cluster (listens for notifications from zookeeper)