Assignment 10

1.What is NoSQL data base?

NoSQL is an approach to database design that can accomodate a wide variety of data models, inclding various 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.

examples of SQL databases include MySQL, Oracle, PostgreSQL, and Microsoft SQL Server

2.How does data get stored in NoSQl database?

NoSQL databases are a speedier alternative because, for one, you don’t need to join tables in NoSQL. Every piece of data is stored in a JSON format. And the limitation of relational databases is that each item can only contain one attribute.

Here, each column is dedicated to just one measure or attribute.

3.What is a column family in HBase?

A column family is a database 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. A column family is as a "table", each key-value pair being a "row".

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

There is no limit on number of column families in HBase, in theory. In reality, there are several factors which can limit useable number of column families in HBase.

It depends on HBase Admin web UI usability.

Each column family has its own directory in HDFS and set of store files and, from performance point of view, the fewer directories (column families) we have the better performance for scan operations we get.

5.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.

Tables are declared up front at schema definition time. Row keys are arrays of bytes and they are lexicographically sorted with the lowest order appearing first.

HBASE returns the latest version of data by default but you can ask for multiple versions in your query. HBase returns data sorted first by the row key values, then by column family, column qualifier and finally by the timestamp value, with the most recent data returned first.

6.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 sharded 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.

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

Hmaster serves as the master which is responsible for creation of tables & region servers acts a slave which communicates with the master. Data is distributed horizontally within the data nodes across diffrent regions

When the client gives a command to Write, Instruction is directed to Write Ahead Log.

Once the log entry is done, the data to be written is forwarded to MemStore which is actually the RAM of the data node.

Data in the memstore is sorted in the same manner as data in a HFile.

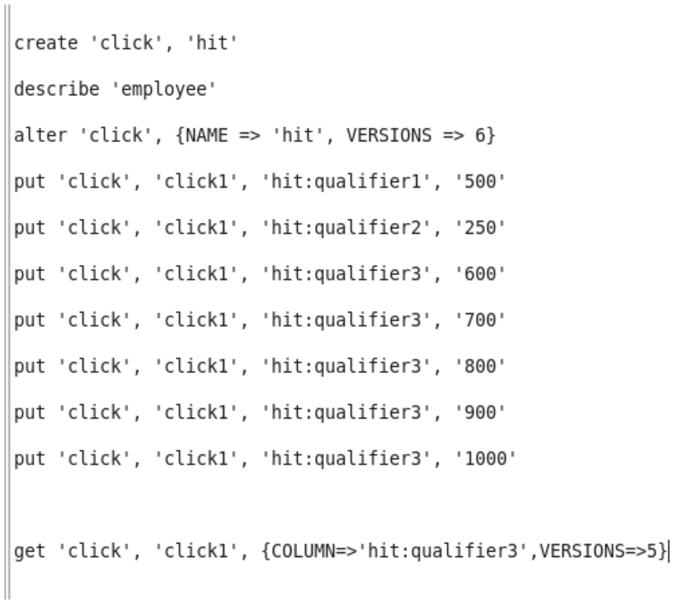
When the memstore accumulates enough data, the entire sorted set is written to a new HFile in HDFS.

Once writing data is completed, ACK (Acknowledgement) is sent to the client as a confirmation of task completed.

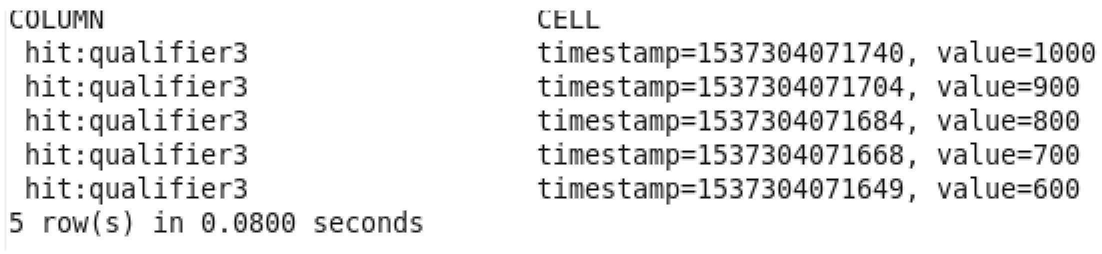
Task 2

Create an HBase table named 'clicks' with a column family 'hits' such that it should be

able to store last 5 values of qualifiers inside 'hits' column family.



```
create 'click', 'hit'

describe 'employee'

alter 'click', {NAME => 'hit', VERSIONS => 6}

put 'click', 'click1', 'hit:qualifier1', '500'

put 'click', 'click1', 'hit:qualifier2', '250'

put 'click', 'click1', 'hit:qualifier3', '600'

put 'click', 'click1', 'hit:qualifier3', '700'

put 'click', 'click1', 'hit:qualifier3', '800'

put 'click', 'click1', 'hit:qualifier3', '900'

put 'click', 'click1', 'hit:qualifier3', '1000'


get 'click', 'click1', {COLUMN=>'hit:qualifier3',VERSIONS=>5}
```



```
COLUMN                            CELL
 hit:qualifier3                   timestamp=1537304071740, value=1000
 hit:qualifier3                   timestamp=1537304071704, value=900
 hit:qualifier3                   timestamp=1537304071684, value=800
 hit:qualifier3                   timestamp=1537304071668, value=700
 hit:qualifier3                   timestamp=1537304071649, value=600
5 row(s) in 0.0800 seconds
```

Output of last 5 values.