ASSIGNMENT

TASK 1:

1.What is a NoSQL database?

Ans. 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 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.

Features:

- Generic data model
- Data structure can be of any form
- No primary and foreign key concept
- Data stored in single table

2.How does data get stored in NoSQL database?

Ans. Data is partitioned and replicated across a cluster to get scalability and availability.

3.What is a column family in Hbase?

Ans. 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.

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(Its a write cache which stores new data before writing it into Hfiles) per Column Family, when one is full, they all flush.

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

Ans. The column families are part of the schema and would require a schema change is that they profoundly impact the way the data is stored, both on disk and in memory. Each column family has its own set of HFiles, and its own set of data structures in memory of the RegionServer. It would be pretty expensive to dynamically create or start using new column families

Task 2:

1:create:

```
hbase(main):005:0> create 'clicks', 'hits'
0 row(s) in 2.2860 seconds

=> Hbase::Table - clicks
```

This command creates a table n hbase with the table name as ‘clicks’ and column name as ‘hits’.

2.put:

```
hbase(main):007:0> put 'clicks','rw001','hits:name','mahesh'
0 row(s) in 0.6690 seconds

hbase(main):008:0> put 'clicks','rw001','hits:name','rahul'
0 row(s) in 0.0290 seconds

hbase(main):009:0> put 'clicks','rw002','hits:name','mahesh'
0 row(s) in 0.0270 seconds

hbase(main):010:0> put 'clicks','rw003','hits:name','kishore'
0 row(s) in 0.0280 seconds

hbase(main):011:0> put 'clicks','rw004','hits:name','shruti'
0 row(s) in 0.0290 seconds

hbase(main):012:0> put 'clicks','rw005','hits:name','kabir'
0 row(s) in 0.0440 seconds

hbase(main):013:0> put 'clicks','rw006','hits:name','rohan'
0 row(s) in 0.0260 seconds
```

This command is used to insert data into hbase table.

3.scan:

```
hbase(main):014:0> scan 'clicks'
ROW                             COLUMN+CELL
 rw001                          column=hits:name, timestamp=1547398751202, value=rahul
 rw002                          column=hits:name, timestamp=1547398764921, value=mahesh
 rw003                          column=hits:name, timestamp=1547398775067, value=kishore
 rw004                          column=hits:name, timestamp=1547398791387, value=shruti
 rw005                          column=hits:name, timestamp=1547398806404, value=kabir
 rw006                          column=hits:name, timestamp=1547398820348, value=rohan
6 row(s) in 0.5420 seconds
```

This command is used to scan the table i.e. view the table contents.