Session 10:
HBASE BASICS
Assignment 1

Task 1

Answer in your own words with example.

1. What is NoSQL database?

   NoSQL database is a framework of database that allows for high performance and dynamic processing of massive data. It is a solution to process big data through low latency queries.

   Unlike relational databases, NoSQL databases are unstructured in nature that mainly focuses on high speed instead of consistency. As it is unstructured in nature, NoSQL is suitable to store in distributed environment. NoSQL can scale horizontally, i.e. when the volume of data increases, we can just add more storage hardware without impacting the performance of databases. In Relational databases, when the volume of data increases, it directly impacts the performance of SQL queries.

   There are 4 types of NoSQL databases,
   - Columnar – suitable for processing structured data. Ex: HBase and Casandra
   - Document – suitable for processing semi-structured data. Ex: MongoDB
   - Memory – suitable for processing data in temporary distributed memory in real time. Ex: RedisDB
   - Graph – suitable for processing and representing data in graphical format

2. How does data get stored in NoSQL database?

   In NoSQL, data are stored in different ways. It depends on the type of data getting stored. There are 4 types of NoSQL databases,
   - Columnar – suitable for processing structured data. In columnar database table, the data is stored along with row id and column instead of rows. This type of storage eliminates the strict consistency in table there-by brings in dynamic.
     Ex: HBase and Casandra
   - Document – suitable for processing semi-structured data. In this type, data is stored in key value pairs.
     Ex: MongoDB where data is stored in json format
   - Memory – suitable for processing data in temporary distributed memory in real time.
     Ex: RedisDB
   - Graph – suitable for processing and representing data in graphical format. It is used to store information about network of data.
     Ex: Neo4J

3. What is column family in HBase?

   Column family is a logical and physical grouping of columns that represent specific attributes of data. The columns in one column family are stored separately from the columns in another family. It is always a good practice to group all non-frequent accessing columns in single column family.

4. How many maximum numbers of columns can be added in HBase table?

   Currently there is no limit in number of columns in HBase table. However, since the column qualifiers are being stored along with values to uniquely identify the value, it is always good practice to keep column qualifier name as small as possible.

   Ex: hbase> put 'employees',1,'prop:FN','saravanan'

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

   As HBase is columnar database, the column qualifier is getting stored along with values. So, column qualifier will be dynamic based on what type of data is expected to store in table. Thereby it eliminates the strict consistency in maintain same columns.

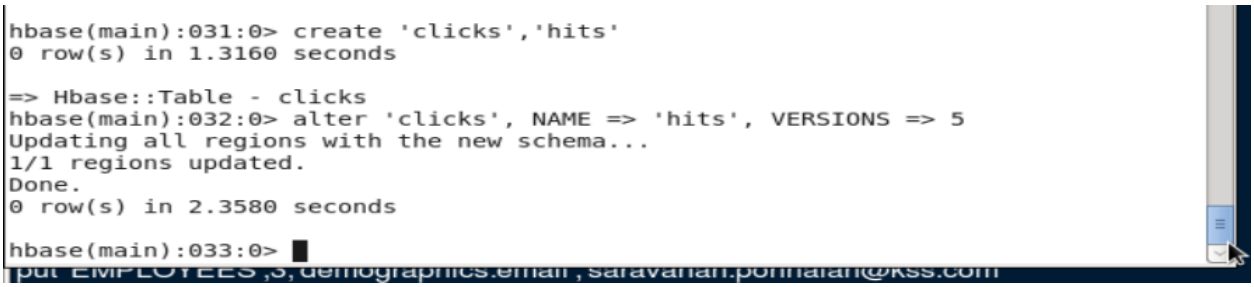
6. How does data get managed in HBase?

   HBase is a column-oriented database and data is stored in tables. The tables are sorted by RowId. HBase has RowId, which is the collection of several column families which-in-which, there are multiple column qualifiers. HBase table act as a map in which each cell can be uniquely identified by a rowid and column qualifier.

7. What happens internally when new data gets inserted into HBase table?
   When a new data is inserted in to HBase table, it stores value with rowid & column qualifier along with timestamp. This will act as an index and also maintains history of data. When the specific value is updated, it actually insert a new record with updated value for specific rowid, column qualifier along with timestamp.

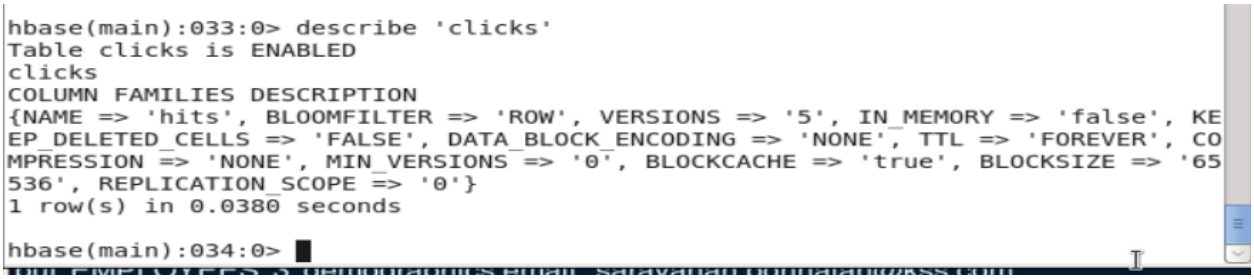
Task 2

1. 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 command creates a new table called ‘clicks’ with one column family as ‘hits’

   - Alter command updates table schema in such a way that it will allow to store maximum of 5 versions of cell values only.



2. Add few records in the table and update some of them. Use IP Address as row-key. Scan the table to view if all the previous versions are getting displayed.

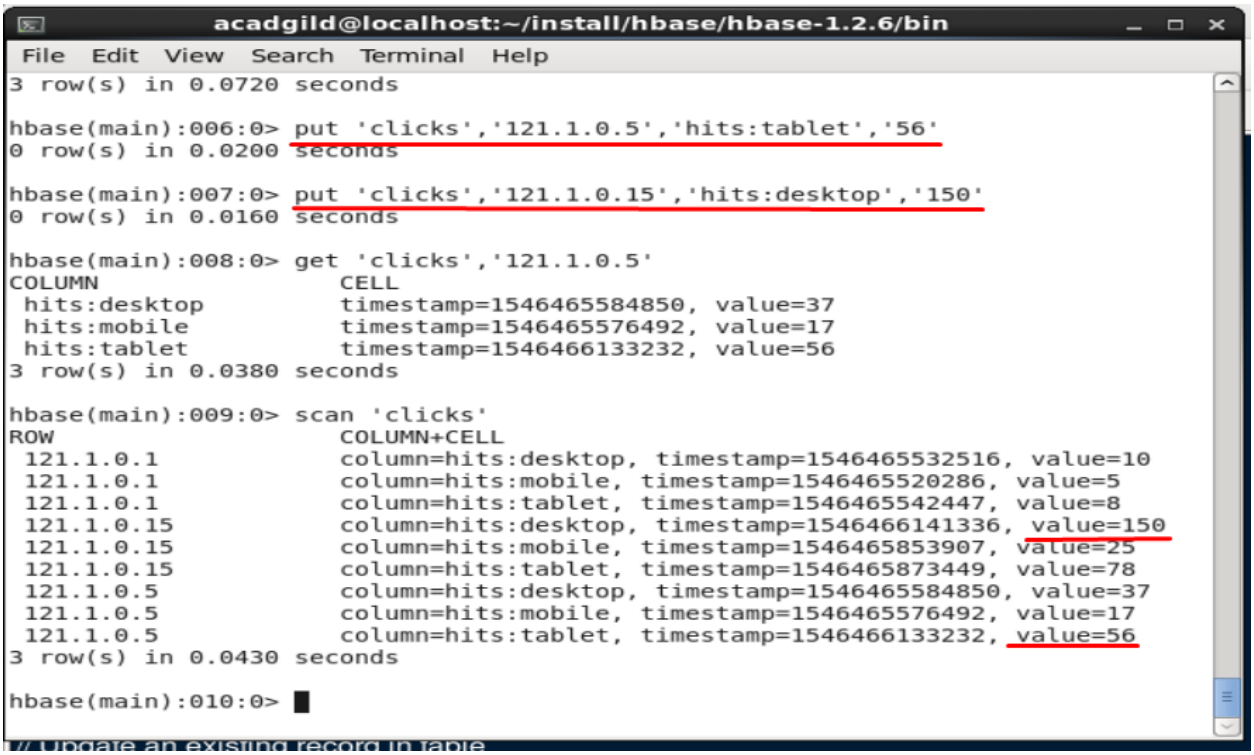
‘put’ command is executed to insert data in to ‘clicks’ table.

```
hbase(main):005:0> scan 'clicks'
ROW                     COLUMN+CELL
 121.1.0.1              column=hits:desktop, timestamp=1546465532516, value=10
 121.1.0.1              column=hits:mobile, timestamp=1546465520286, value=5
 121.1.0.1              column=hits:tablet, timestamp=1546465542447, value=8
 121.1.0.15             column=hits:desktop, timestamp=1546465864630, value=100
 121.1.0.15             column=hits:mobile, timestamp=1546465853907, value=25
 121.1.0.15             column=hits:tablet, timestamp=1546465873449, value=78
 121.1.0.5              column=hits:desktop, timestamp=1546465584850, value=37
 121.1.0.5              column=hits:mobile, timestamp=1546465576492, value=17
 121.1.0.5              column=hits:tablet, timestamp=1546465844674, value=43
3 row(s) in 0.0720 seconds

hbase(main):006:0>
```

'scan' command is executed to display all data in 'clicks' table

```
acadgild@localhost:~/install/hbase/hbase-1.2.6/bin
File  Edit  View  Search  Terminal  Help

hbase(main):004:0> put 'clicks','121.1.0.15','hits:tablet','78'
0 row(s) in 0.0220 seconds

hbase(main):005:0> scan 'clicks'
ROW                     COLUMN+CELL
 121.1.0.1              column=hits:desktop, timestamp=1546465532516, value=10
 121.1.0.1              column=hits:mobile, timestamp=1546465520286, value=5
 121.1.0.1              column=hits:tablet, timestamp=1546465542447, value=8
 121.1.0.15             column=hits:desktop, timestamp=1546465864630, value=100
 121.1.0.15             column=hits:mobile, timestamp=1546465853907, value=25
 121.1.0.15             column=hits:tablet, timestamp=1546465873449, value=78
 121.1.0.5              column=hits:desktop, timestamp=1546465584850, value=37
 121.1.0.5              column=hits:mobile, timestamp=1546465576492, value=17
 121.1.0.5              column=hits:tablet, timestamp=1546465844674, value=43
3 row(s) in 0.0720 seconds

hbase(main):006:0> put 'clicks','121.1.0.5','hits:tablet','56'
0 row(s) in 0.0200 seconds

hbase(main):007:0> put 'clicks','121.1.0.15','hits:desktop','150'
0 row(s) in 0.0160 seconds

hbase(main):008:0>
```

'put' command is executed to update existing records marked above and scan the table to check the data is updated,


acadgild@localhost:~/install/hbase/hbase-1.2.6/bin
File Edit View Search Terminal Help
3 row(s) in 0.0720 seconds
hbase(main):006:0> put 'clicks','121.1.0.5','hits:tablet','56'
0 row(s) in 0.0200 seconds
hbase(main):007:0> put 'clicks','121.1.0.15','hits:desktop','150'
0 row(s) in 0.0160 seconds
hbase(main):008:0> get 'clicks','121.1.0.5'
COLUMN CELL
hits:desktop timestamp=1546465584850, value=37
hits:mobile timestamp=1546465576492, value=17
hits:tablet timestamp=1546466133232, value=56
3 row(s) in 0.0380 seconds
hbase(main):009:0> scan 'clicks'
ROW COLUMN+CELL
121.1.0.1 column=hits:desktop, timestamp=1546465532516, value=10
121.1.0.1 column=hits:mobile, timestamp=1546465520286, value=5
121.1.0.1 column=hits:tablet, timestamp=1546465542447, value=8
121.1.0.15 column=hits:desktop, timestamp=1546466141336, value=150
121.1.0.15 column=hits:mobile, timestamp=1546465853907, value=25
121.1.0.15 column=hits:tablet, timestamp=1546465873449, value=78
121.1.0.5 column=hits:desktop, timestamp=1546465584850, value=37
121.1.0.5 column=hits:mobile, timestamp=1546465576492, value=17
121.1.0.5 column=hits:tablet, timestamp=1546466133232, value=56
3 row(s) in 0.0430 seconds
hbase(main):010:0>
// Update an existing record in table