# Assignment- 10 Hbase Basics

**Ques 1.1: What is 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, 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.

**Ques 1.2: 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:

- **Key-value data stores:** Key-value NoSQL databases emphasize simplicity and are very useful in accelerating an application to support high-speed read and write processing of non-transactional data. Stored values can be any type of binary object (text, video, JSON document, etc.) and are accessed via a key. The application has complete control over what is stored in the value, making this the most flexible NoSQL model. Data is partitioned and replicated across a cluster to get scalability and availability. For this reason, key value stores often do not support transactions. However, they are highly effective at scaling applications that deal with high-velocity, non-transactional data.
- **Document stores:** Document databases typically store self-describing JSON, XML, and BSON documents. They are similar to key-value stores, but in this case, a value is a single document that stores all data related to a specific key. Popular fields in the document can be indexed to provide fast retrieval without knowing the key. Each document can have the same or a different structure.
- **Wide-column stores:** Wide-column NoSQL databases store data in tables with rows and columns similar to RDBMS, but names and formats of columns can vary from row to row across the table. Wide-column databases group columns of related data together. A query can retrieve related data in a single operation because only the columns associated with the query are retrieved. In an RDBMS, the data would be in different rows stored in different places on disk, requiring multiple disk operations for retrieval.
- **Graph stores:** A graph database uses graph structures to store, map, and query relationships. They provide index-free adjacency, so that adjacent elements are linked together without using an index.

**Ques 1.3: 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.

**Ques 1.4: How many maximum number of columns can be added to HBase table?**

**Ans:** We can add n numbers of columns in Hbase Table.

**Ques 1.5: 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 and running.

Physically, all column family members are stored together on the filesystem. Because tunings and storage specifications are done at the column family level, it is advised that all column family members have the same general access pattern and size characteristics.

**Ques 1.6: How does data get managed in HBase?**

**Ans:** Data in Hbase is organized into tables. Any characters that are legal in file paths are used to name tables. Tables are further organized into rows that store data. Each row is identified by a unique row key which does not belong to any data type but is stored as a bytearray. Column families are further used to group data in rows. Column families define the physical structure of data so they are defined upfront and their modification is difficult. Each row in a table has same column families. Data in a column family is addressed using a column qualifier. It is not necessary to specify column qualifiers in advance and there is no consistency requirement between rows. No data types are specified for column qualifiers, as such they are just stored as bytearrays. A unique combination of row key, column family and column qualifier forms a cell. Data contained in a cell is referred to as cell value. There is no concept of data type when referring to cell values and they are stored as bytearrays. Versioning happens to cell values using a timestamp of when the cell was written.

**Ques 1.7: What happens internally when new data gets inserted into HBase table?**

**Ans:** The below mentioned steps gives us the good idea how the write process happened in HBase:

- 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**

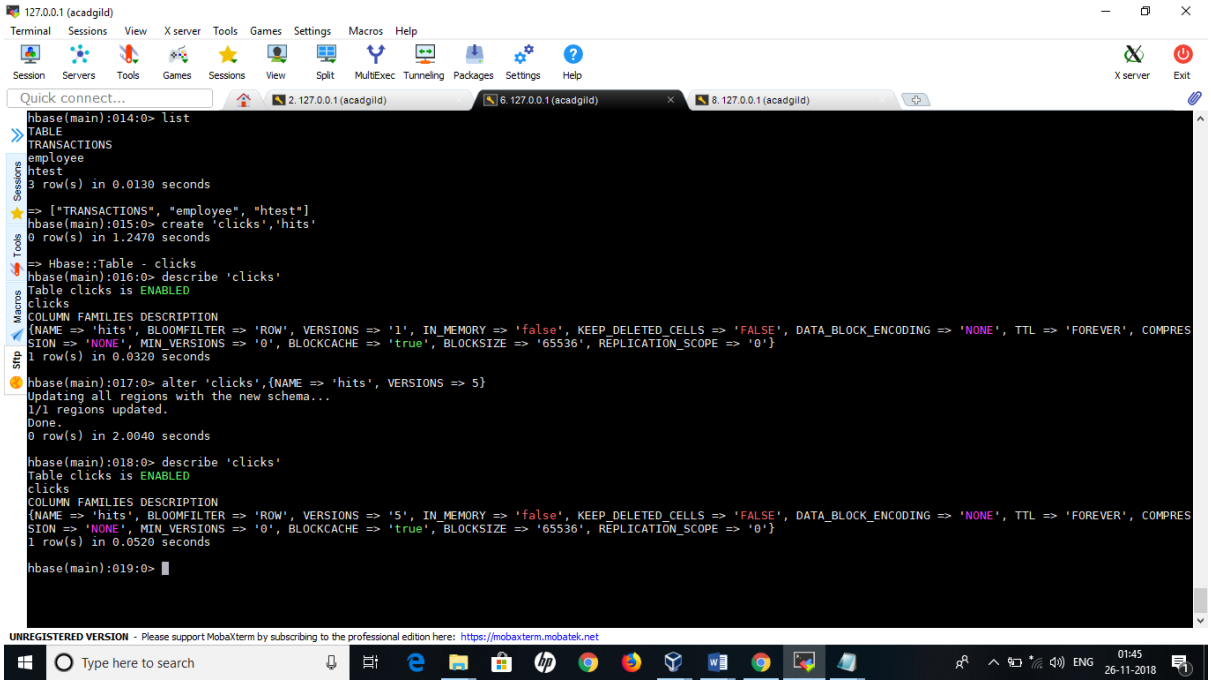
1. Create an HBase table named 'clicks' with a column family 'hits' such that it should be

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

Solution: *create 'clicks', 'hits'*

*describe 'clicks'*

*alter 'clicks', {NAME => 'hits', VERSIONS => 5}*




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.

Solution: *put 'clicks', 'IP Address', 'hits:name', 'Gaurav Puri'*

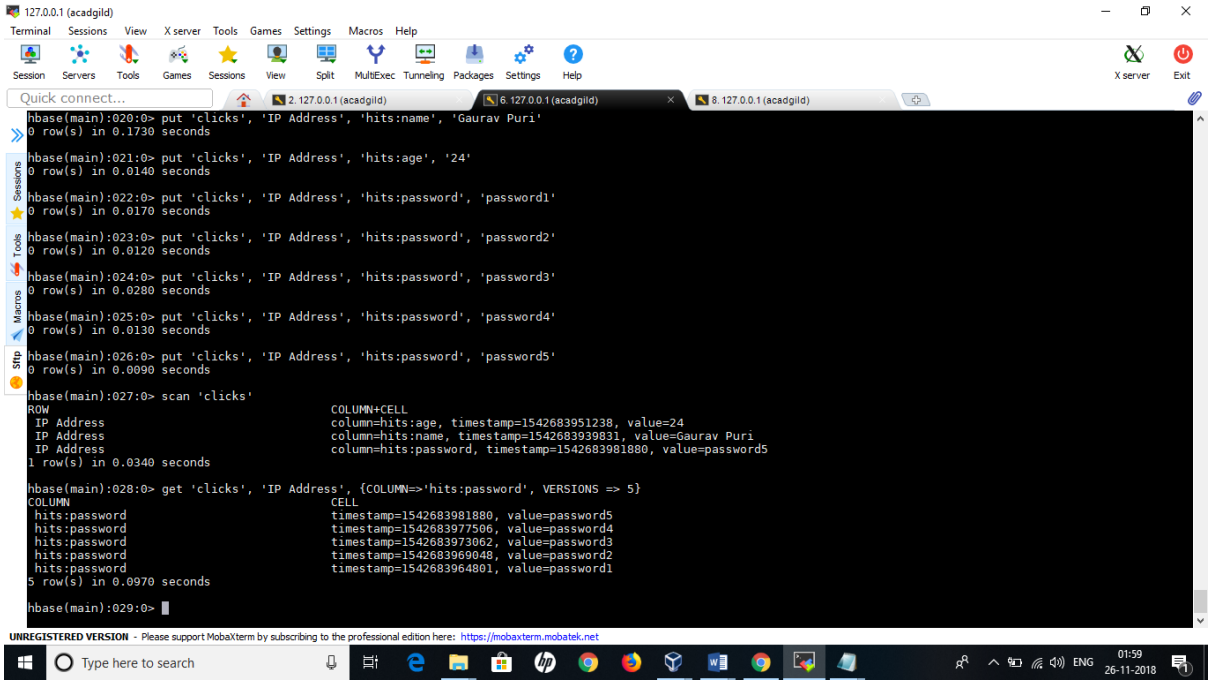
*put 'clicks', 'IP Address', 'hits:age', '24'*

*put 'clicks', 'IP Address', 'hits:password', 'password1'*

*put 'clicks', 'IP Address', 'hits:password', 'password2'*

*put 'clicks', 'IP Address', 'hits:password', 'password3'*

*get 'clicks', 'IP Address', {COLUMN=>'hits:password', VERSIONS => 5}*


127.0.0.1 (acadgild)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
2. 127.0.0.1 (acadgild)
6. 127.0.0.1 (acadgild)
8. 127.0.0.1 (acadgild)
Sessions Tools Macros Sftp
hbase(main):020:0> put 'clicks', 'IP Address', 'hits:name', 'Gaurav Puri'
0 row(s) in 0.1730 seconds
hbase(main):021:0> put 'clicks', 'IP Address', 'hits:age', '24'
0 row(s) in 0.0140 seconds
hbase(main):022:0> put 'clicks', 'IP Address', 'hits:password', 'password1'
0 row(s) in 0.0170 seconds
hbase(main):023:0> put 'clicks', 'IP Address', 'hits:password', 'password2'
0 row(s) in 0.0120 seconds
hbase(main):024:0> put 'clicks', 'IP Address', 'hits:password', 'password3'
0 row(s) in 0.0280 seconds
hbase(main):025:0> put 'clicks', 'IP Address', 'hits:password', 'password4'
0 row(s) in 0.0130 seconds
hbase(main):026:0> put 'clicks', 'IP Address', 'hits:password', 'password5'
0 row(s) in 0.0090 seconds
hbase(main):027:0> scan 'clicks'
ROW COLUMN+CELL
IP Address column=hits:age, timestamp=1542683951238, value=24
IP Address column=hits:name, timestamp=1542683939831, value=Gaurav Puri
IP Address column=hits:password, timestamp=1542683981880, value=password5
1 row(s) in 0.0340 seconds
hbase(main):028:0> get 'clicks', 'IP Address', {COLUMN=>'hits:password', VERSIONS => 5}
COLUMN CELL
hits:password timestamp=1542683981880, value=password5
hits:password timestamp=1542683977506, value=password4
hits:password timestamp=1542683973062, value=password3
hits:password timestamp=1542683969048, value=password2
hits:password timestamp=1542683964801, value=password1
5 row(s) in 0.0970 seconds
hbase(main):029:0>
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search
ENG
01:59
26-11-2018