

What is No Sql database?

Is a type of database mechanism which enables storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases. *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.

How does data get stored in NoSQL database?

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.

What is a column family in HBase?

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 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 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.

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

Millions. Any number of columns can be added.

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

HBase is schema-less, it doesn't have the concept of fixed columns schema. Because column families are defined and 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.

How does data get managed in HBase?

HBase is a sparse, distributed, persistent multidimensional sorted map database. A subset of the *data* of each table *is handled* by each region server. ... Clients *do* not read HFiles directly but go through region servers to *get* to the *data*. The region files share a common key corresponding to the column family. When accessing data clients communicate directly with the region servers. Hbase master process Zookeeper maintains a live cluster state necessary for region assignment and create and delete operations. There is a special HBase Catalog table called the META table, which holds the location of the regions in the cluster. ZooKeeper stores the location of the META table.

This is what happens the first time a client reads or writes to HBase:

1. The client gets the Region server that hosts the META table from ZooKeeper.
2. The client will query the .META. server to get the region server corresponding to the row key it wants to access. The client caches this information along with the META table location.
3. It will get the Row from the corresponding Region Server.

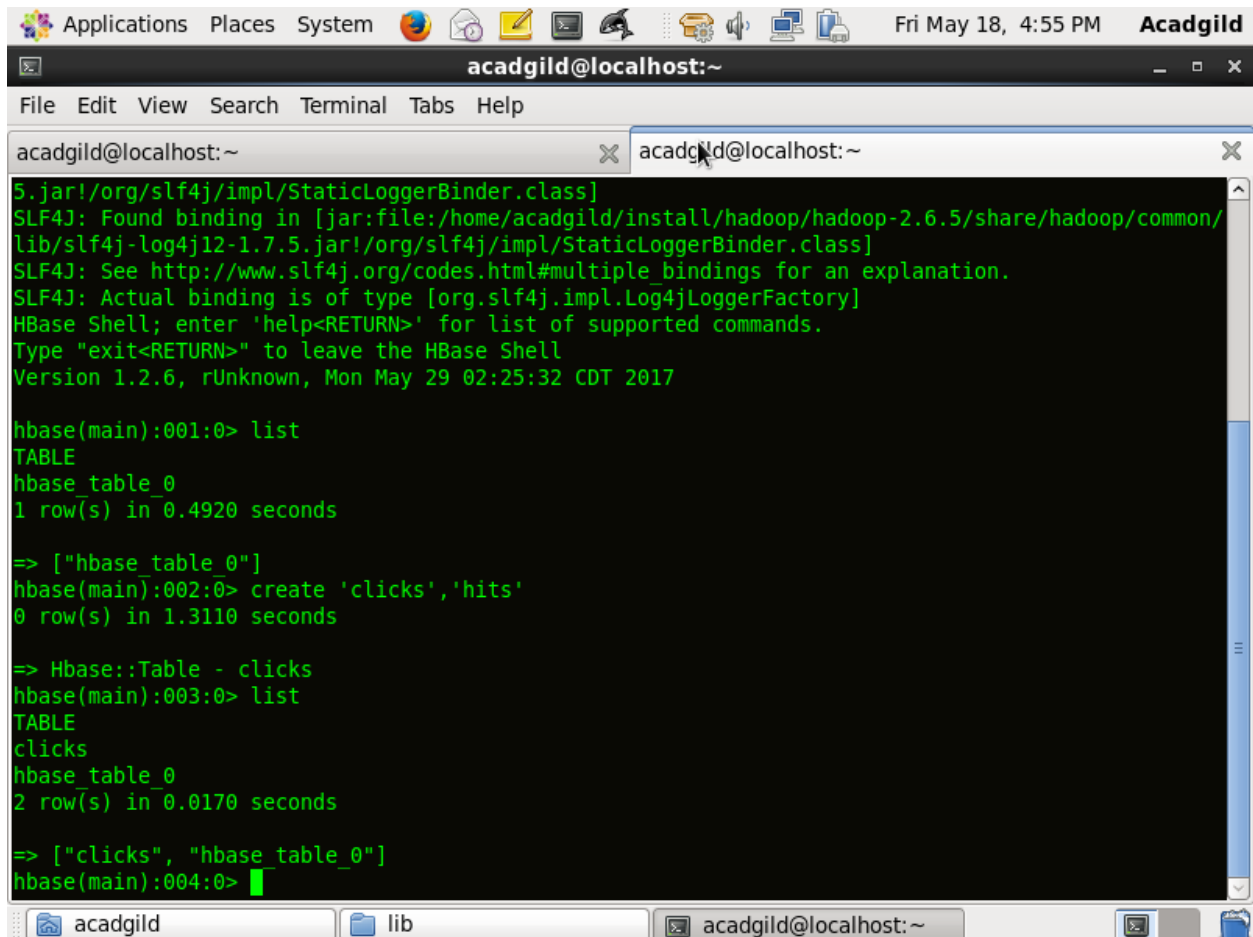
For future reads, the client uses the cache to retrieve the META location and previously read row keys. Over time, it does not need to query the META table, unless there is a miss because a region has moved; then it will re-query and update the cache.

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

When data is updated it is first written to a commit log, called a write-ahead log (WAL) in HBase, and then stored in the in-memory memstore. Once the data in memory has exceeded a given maximum value, it is flushed as an HFile to disk. After the flush, the commit logs can be discarded up to the last unflushed modification.

Creating table with name 'clicks' with one column family name 'hits'

Using list command to see the list of tables. You can see your newly created table 'clicks'



The screenshot shows a terminal window titled 'acadgild@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Tabs, Help). The terminal displays the following text:

```
5.jar!/org.slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/
lib/slf4j-log4j12-1.7.5.jar!/org.slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

hbase(main):001:0> list
TABLE
hbase_table_0
1 row(s) in 0.4920 seconds

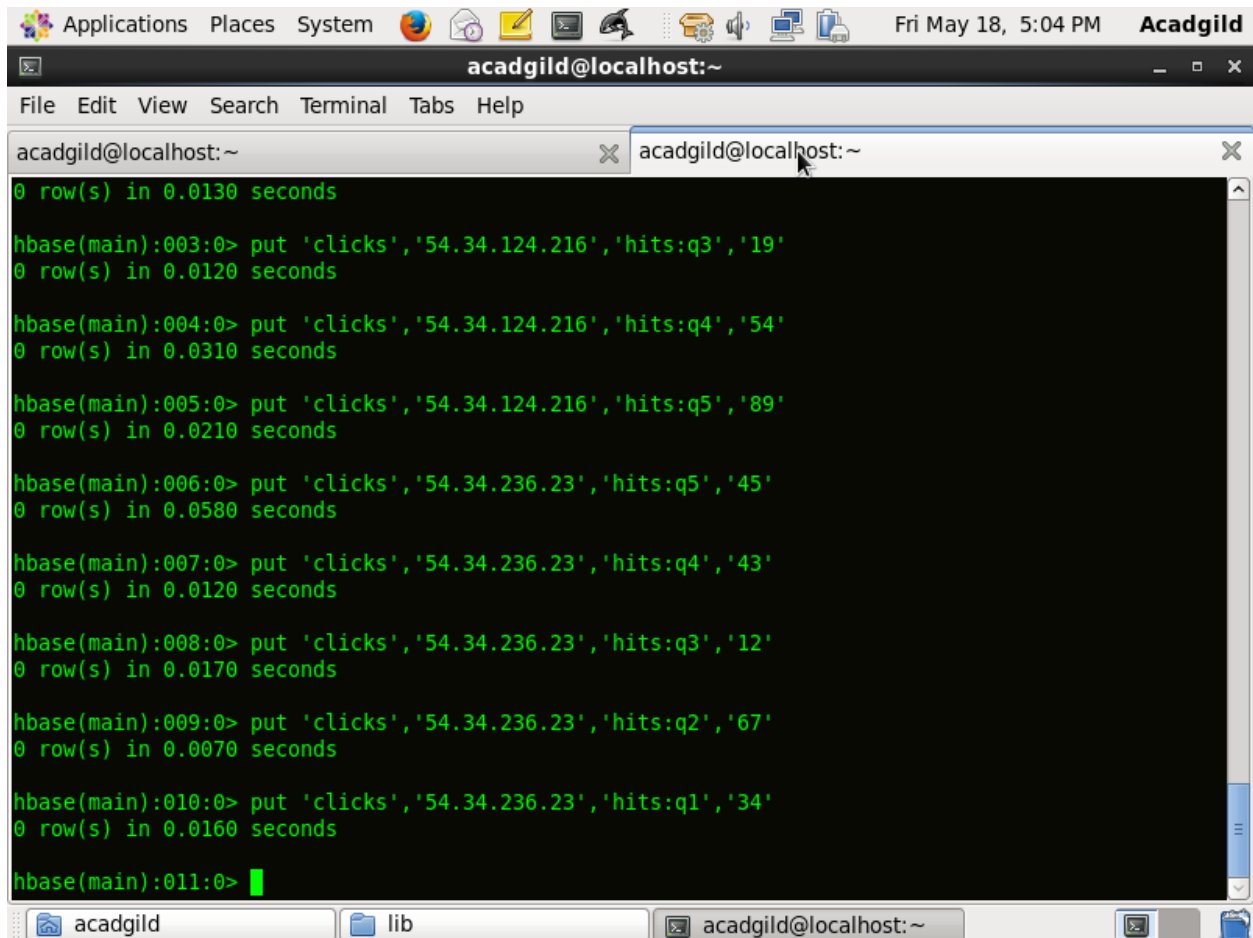
=> ["hbase_table_0"]
hbase(main):002:0> create 'clicks','hits'
0 row(s) in 1.3110 seconds

=> Hbase::Table - clicks
hbase(main):003:0> list
TABLE
clicks
hbase_table_0
2 row(s) in 0.0170 seconds

=> ["clicks", "hbase_table_0"]
hbase(main):004:0> █
```

The terminal window has a taskbar at the bottom with icons for 'acadgild', 'lib', and 'acadgild@localhost:~'.

Using command : put '<table name>','row1','<colfamily:colname>','<value>'



The screenshot shows a terminal window titled 'acadgild@localhost:~'. The window contains a series of HBase 'put' commands. Each command is followed by a confirmation message indicating the number of rows inserted and the time taken. The commands are as follows:

```
hbase(main):003:0> put 'clicks','54.34.124.216','hits:q3','19'
0 row(s) in 0.0120 seconds

hbase(main):004:0> put 'clicks','54.34.124.216','hits:q4','54'
0 row(s) in 0.0310 seconds

hbase(main):005:0> put 'clicks','54.34.124.216','hits:q5','89'
0 row(s) in 0.0210 seconds

hbase(main):006:0> put 'clicks','54.34.236.23','hits:q5','45'
0 row(s) in 0.0580 seconds

hbase(main):007:0> put 'clicks','54.34.236.23','hits:q4','43'
0 row(s) in 0.0120 seconds

hbase(main):008:0> put 'clicks','54.34.236.23','hits:q3','12'
0 row(s) in 0.0170 seconds

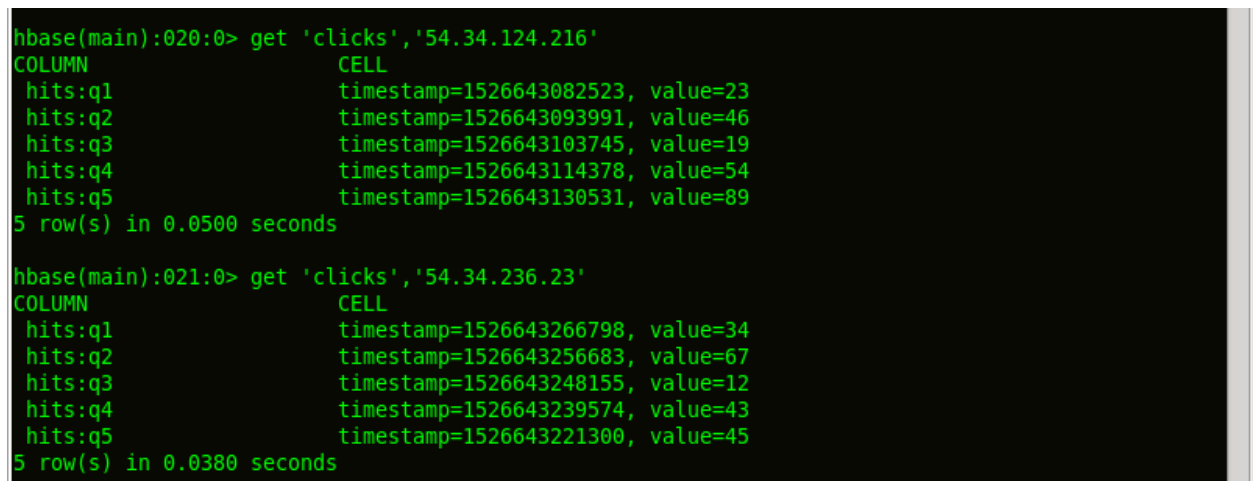
hbase(main):009:0> put 'clicks','54.34.236.23','hits:q2','67'
0 row(s) in 0.0070 seconds

hbase(main):010:0> put 'clicks','54.34.236.23','hits:q1','34'
0 row(s) in 0.0160 seconds

hbase(main):011:0>
```

Using get command to view the contents of row : get '<table name>','row1'

Here I have two row ids which has 5 columns: q1, q2, q3, q4, q5



The screenshot shows a terminal window with HBase 'get' commands and their output. The output for each command shows the column names and the corresponding values, along with timestamps.

```
hbase(main):020:0> get 'clicks','54.34.124.216'
COLUMN          CELL
hits:q1          timestamp=1526643082523, value=23
hits:q2          timestamp=1526643093991, value=46
hits:q3          timestamp=1526643103745, value=19
hits:q4          timestamp=1526643114378, value=54
hits:q5          timestamp=1526643130531, value=89
5 row(s) in 0.0500 seconds

hbase(main):021:0> get 'clicks','54.34.236.23'
COLUMN          CELL
hits:q1          timestamp=1526643266798, value=34
hits:q2          timestamp=1526643256683, value=67
hits:q3          timestamp=1526643248155, value=12
hits:q4          timestamp=1526643239574, value=43
hits:q5          timestamp=1526643221300, value=45
5 row(s) in 0.0380 seconds
```

Performing updation of values

```
hbase(main):022:0> put 'clicks','54.34.236.23','hits:q4','99'
0 row(s) in 0.0110 seconds

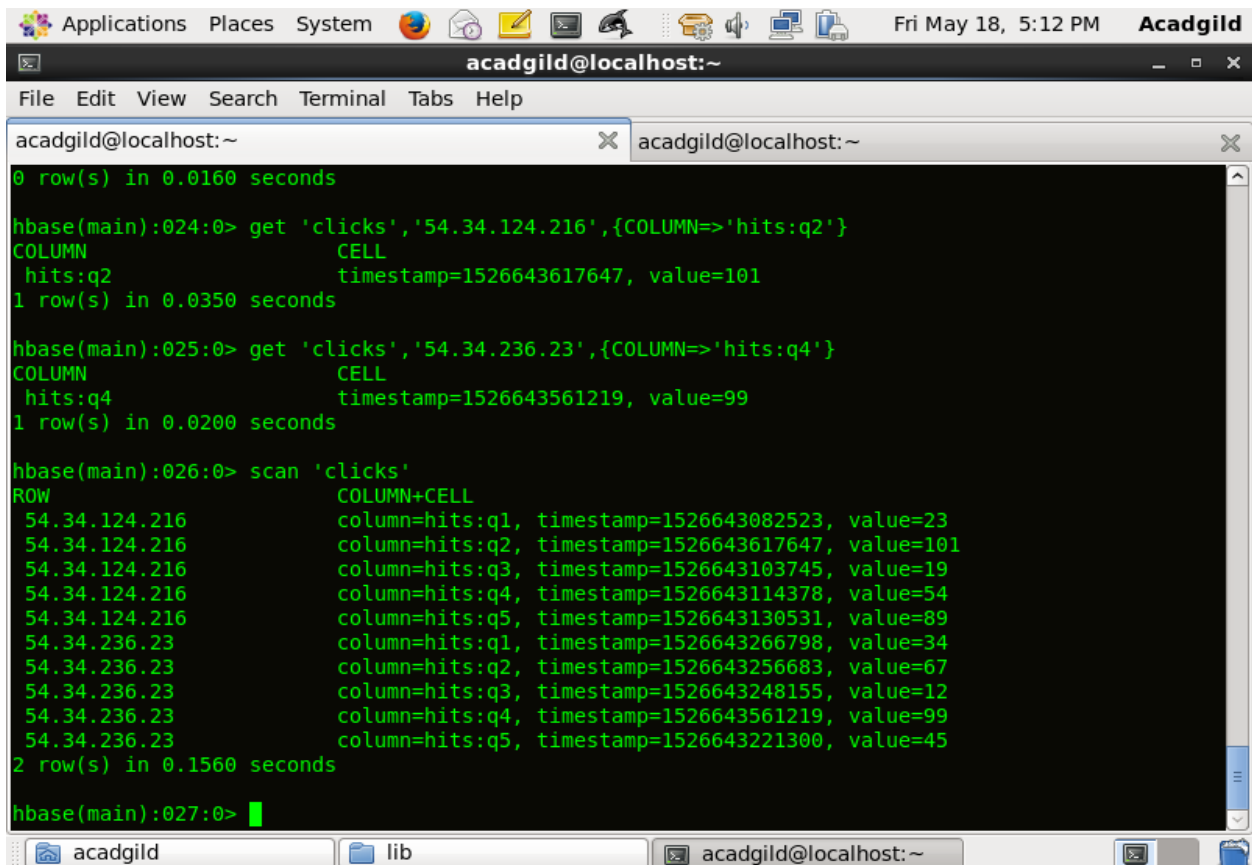
hbase(main):023:0> put 'clicks','54.34.124.216','hits:q2','101'
0 row(s) in 0.0160 seconds
```

Checking Updated values

```
hbase(main):024:0> get 'clicks','54.34.124.216',{COLUMN=>'hits:q2'}
COLUMN          CELL
hits:q2          timestamp=1526643617647, value=101
1 row(s) in 0.0350 seconds

hbase(main):025:0> get 'clicks','54.34.236.23',{COLUMN=>'hits:q4'}
COLUMN          CELL
hits:q4          timestamp=1526643561219, value=99
1 row(s) in 0.0200 seconds
```

Using scan command



```
0 row(s) in 0.0160 seconds

hbase(main):024:0> get 'clicks','54.34.124.216',{COLUMN=>'hits:q2'}
COLUMN          CELL
hits:q2          timestamp=1526643617647, value=101
1 row(s) in 0.0350 seconds

hbase(main):025:0> get 'clicks','54.34.236.23',{COLUMN=>'hits:q4'}
COLUMN          CELL
hits:q4          timestamp=1526643561219, value=99
1 row(s) in 0.0200 seconds

hbase(main):026:0> scan 'clicks'
ROW              COLUMN+CELL
54.34.124.216    column=hits:q1, timestamp=1526643082523, value=23
54.34.124.216    column=hits:q2, timestamp=1526643617647, value=101
54.34.124.216    column=hits:q3, timestamp=1526643103745, value=19
54.34.124.216    column=hits:q4, timestamp=1526643114378, value=54
54.34.124.216    column=hits:q5, timestamp=1526643130531, value=89
54.34.236.23     column=hits:q1, timestamp=1526643266798, value=34
54.34.236.23     column=hits:q2, timestamp=1526643256683, value=67
54.34.236.23     column=hits:q3, timestamp=1526643248155, value=12
54.34.236.23     column=hits:q4, timestamp=1526643561219, value=99
54.34.236.23     column=hits:q5, timestamp=1526643221300, value=45
2 row(s) in 0.1560 seconds

hbase(main):027:0>
```