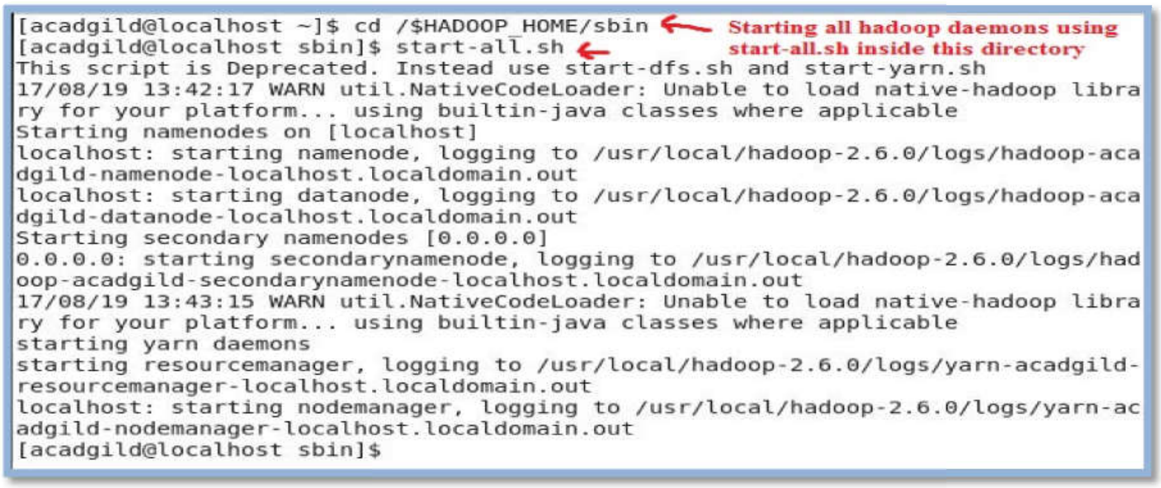
HBase - Assignment

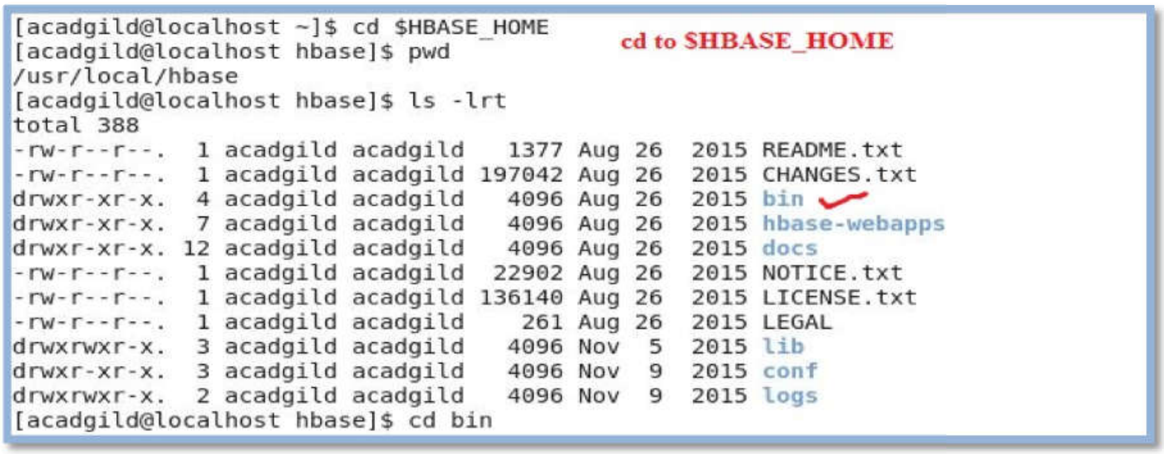
To launch HBase Shell, below steps are performed:

1. Started all hadoop daemons, using **start-all.sh** command inside **/$HADOOP_HOME/sbin** directory.
2. Started hbase daemon, using **start-hbase.sh** command inside **/$HBASE_HOME/bin**
3. Using jps, we can see all daemons have started or not.
4. Since, all daemons started, then launched hbase shell using **hbase shell** command.
   Refer below screenshots for above steps:

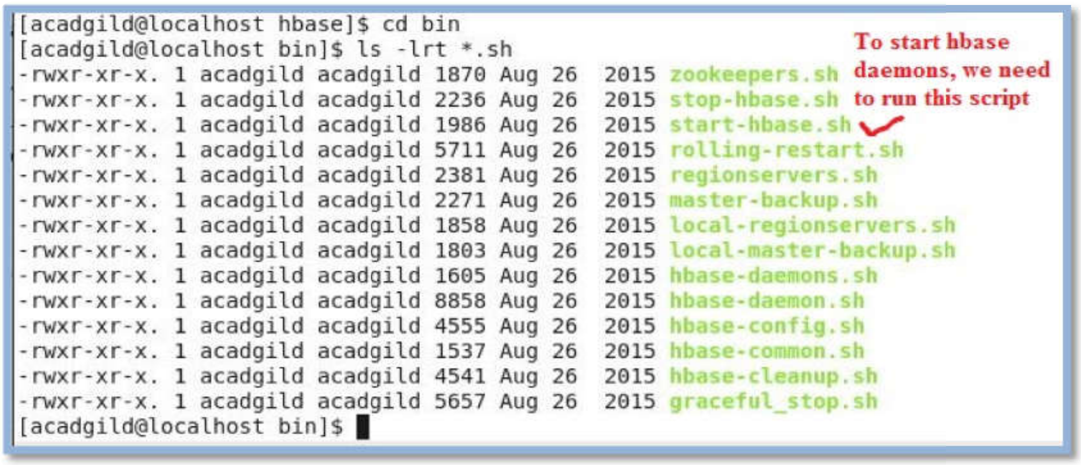


```
[acadgild@localhost ~]$ cd /$HADOOP_HOME/sbin
[acadgild@localhost sbin]$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
17/08/19 13:42:17 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop-2.6.0/logs/hadoop-aca
dgild-namenode-localhost.localdomain.out
localhost: starting datanode, logging to /usr/local/hadoop-2.6.0/logs/hadoop-aca
dgild-datanode-localhost.localdomain.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop-2.6.0/logs/had
oop-acadgild-secondarynamenode-localhost.localdomain.out
17/08/19 13:43:15 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop-2.6.0/logs/yarn-acadgild-
resourcemanager-localhost.localdomain.out
localhost: starting nodemanager, logging to /usr/local/hadoop-2.6.0/logs/yarn-ac
adgild-nodemanager-localhost.localdomain.out
[acadgild@localhost sbin]$
```

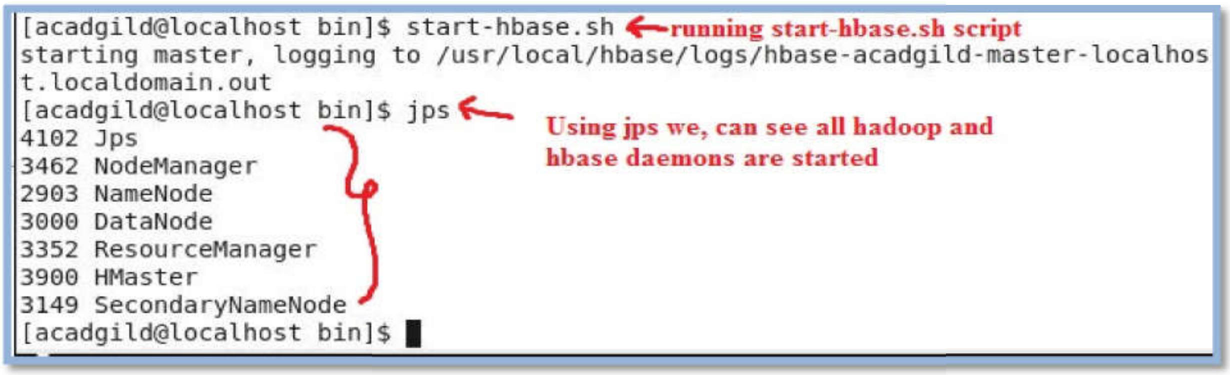
Starting all hadoop daemons using
start-all.sh inside this directory



```
[acadgild@localhost ~]$ cd $HBASE_HOME
[acadgild@localhost hbase]$ pwd
/usr/local/hbase
[acadgild@localhost hbase]$ ls -lrt
total 388
-rw-r--r--.  1 acadgild acadgild   1377 Aug 26  2015 README.txt
-rw-r--r--.  1 acadgild acadgild 197042 Aug 26  2015 CHANGES.txt
drwxr-xr-x.  4 acadgild acadgild   4096 Aug 26  2015 bin
drwxr-xr-x.  7 acadgild acadgild   4096 Aug 26  2015 hbase-webapps
drwxr-xr-x. 12 acadgild acadgild   4096 Aug 26  2015 docs
-rw-r--r--.  1 acadgild acadgild  22902 Aug 26  2015 NOTICE.txt
-rw-r--r--.  1 acadgild acadgild 136140 Aug 26  2015 LICENSE.txt
-rw-r--r--.  1 acadgild acadgild    261 Aug 26  2015 LEGAL
drwxrwxr-x.  3 acadgild acadgild   4096 Nov  5  2015 lib
drwxr-xr-x.  3 acadgild acadgild   4096 Nov  9  2015 conf
drwxrwxr-x.  2 acadgild acadgild   4096 Nov  9  2015 logs
[acadgild@localhost hbase]$ cd bin
```

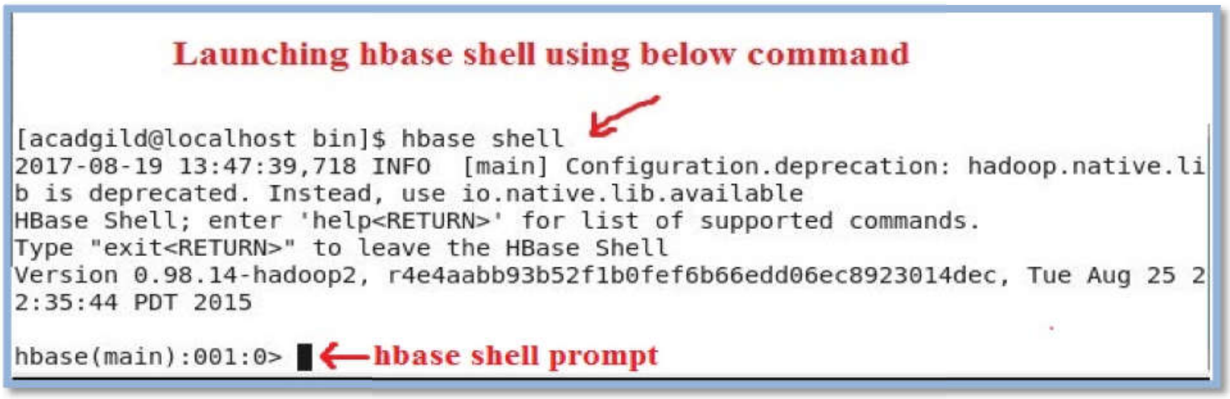


```
[acadgild@localhost hbase]$ cd bin
[acadgild@localhost bin]$ ls -lrt *.sh
-rwxr-xr-x. 1 acadgild acadgild 1870 Aug 26  2015 zookeepers.sh
-rwxr-xr-x. 1 acadgild acadgild 2236 Aug 26  2015 stop-hbase.sh
-rwxr-xr-x. 1 acadgild acadgild 1986 Aug 26  2015 start-hbase.sh
-rwxr-xr-x. 1 acadgild acadgild 5711 Aug 26  2015 rolling-restart.sh
-rwxr-xr-x. 1 acadgild acadgild 2381 Aug 26  2015 regionservers.sh
-rwxr-xr-x. 1 acadgild acadgild 2271 Aug 26  2015 master-backup.sh
-rwxr-xr-x. 1 acadgild acadgild 1858 Aug 26  2015 local-regionservers.sh
-rwxr-xr-x. 1 acadgild acadgild 1803 Aug 26  2015 local-master-backup.sh
-rwxr-xr-x. 1 acadgild acadgild 1605 Aug 26  2015 hbase-daemons.sh
-rwxr-xr-x. 1 acadgild acadgild 8858 Aug 26  2015 hbase-daemon.sh
-rwxr-xr-x. 1 acadgild acadgild 4555 Aug 26  2015 hbase-config.sh
-rwxr-xr-x. 1 acadgild acadgild 1537 Aug 26  2015 hbase-common.sh
-rwxr-xr-x. 1 acadgild acadgild 4541 Aug 26  2015 hbase-cleanup.sh
-rwxr-xr-x. 1 acadgild acadgild 5657 Aug 26  2015 graceful_stop.sh
[acadgild@localhost bin]$
```



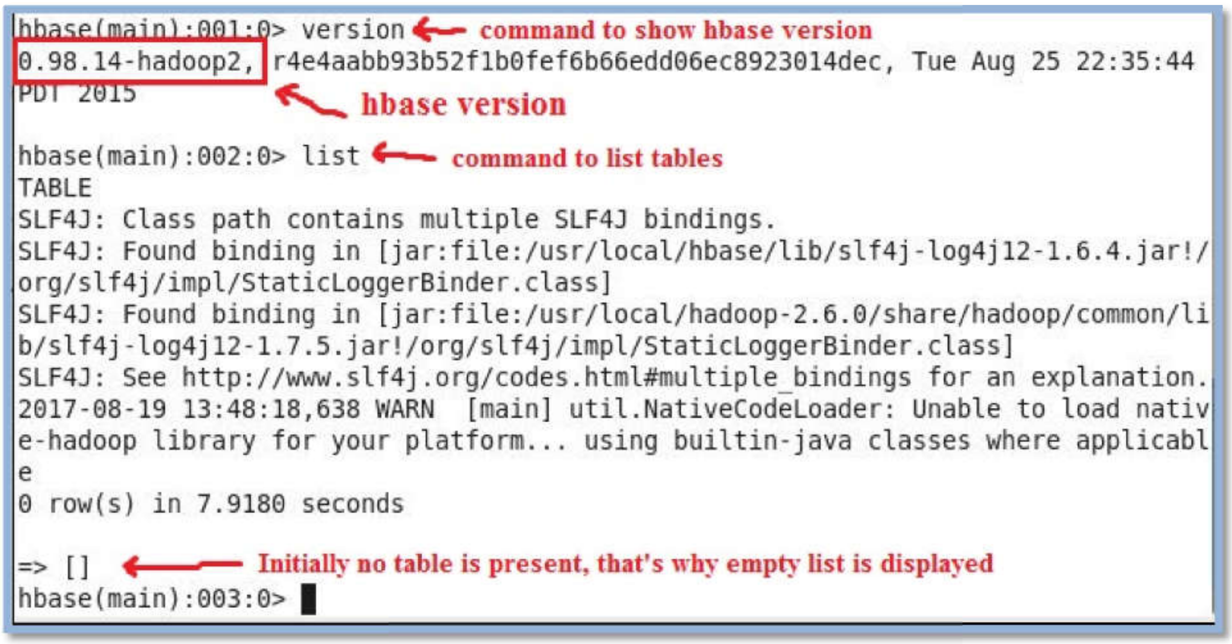
```
[acadgild@localhost bin]$ start-hbase.sh
starting master, logging to /usr/local/hbase/logs/hbase-acadgild-master-localhos
t.localdomain.out
[acadgild@localhost bin]$ jps
4102 Jps
3462 NodeManager
2903 NameNode
3000 DataNode
3352 ResourceManager
3900 HMaster
3149 SecondaryNameNode
[acadgild@localhost bin]$
```

running start-hbase.sh script
Using jps we, can see all hadoop and hbase daemons are started



```
[acadgild@localhost bin]$ hbase shell
2017-08-19 13:47:39,718 INFO  [main] Configuration.deprecation: hadoop.native.li
b is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.98.14-hadoop2, r4e4aabb93b52f1b0fef6b66edd06ec8923014dec, Tue Aug 25 2
2:35:44 PDT 2015

hbase(main):001:0>
```

Since hbase shell prompt has appeared, now we can proceed with operations on hbase tables



```
hbase(main):001:0> version
0.98.14-hadoop2, r4e4aabb93b52f1b0fef6b66edd06ec8923014dec, Tue Aug 25 22:35:44
PDT 2015

hbase(main):002:0> list
TABLE
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/
org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/li
b/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2017-08-19 13:48:18,638 WARN  [main] util.NativeCodeLoader: Unable to load nativ
e-hadoop library for your platform... using builtin-java classes where applicabl
e
0 row(s) in 7.9180 seconds

=> []
hbase(main):003:0>
```

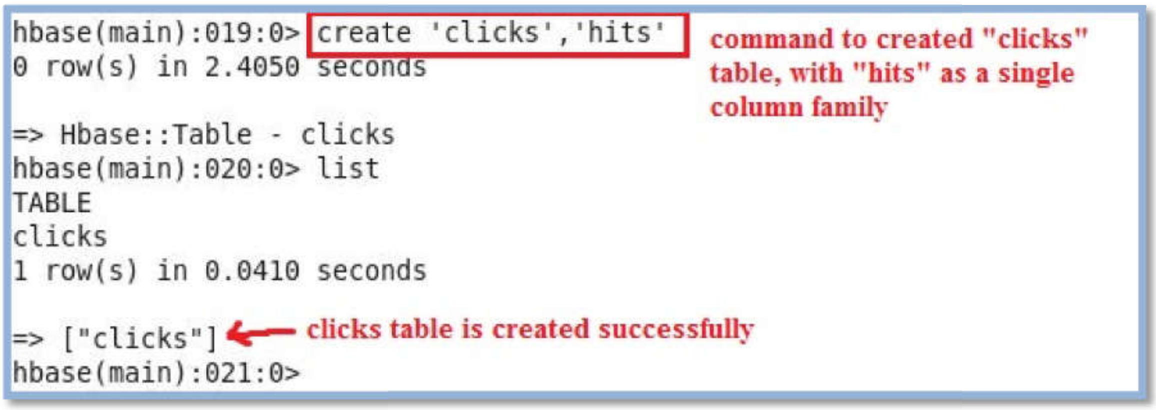
command to show hbase version
hbase version
command to list tables
Initially no table is present, that's why empty list is displayed

Problem 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.

Below steps are followed to accomplish above task:

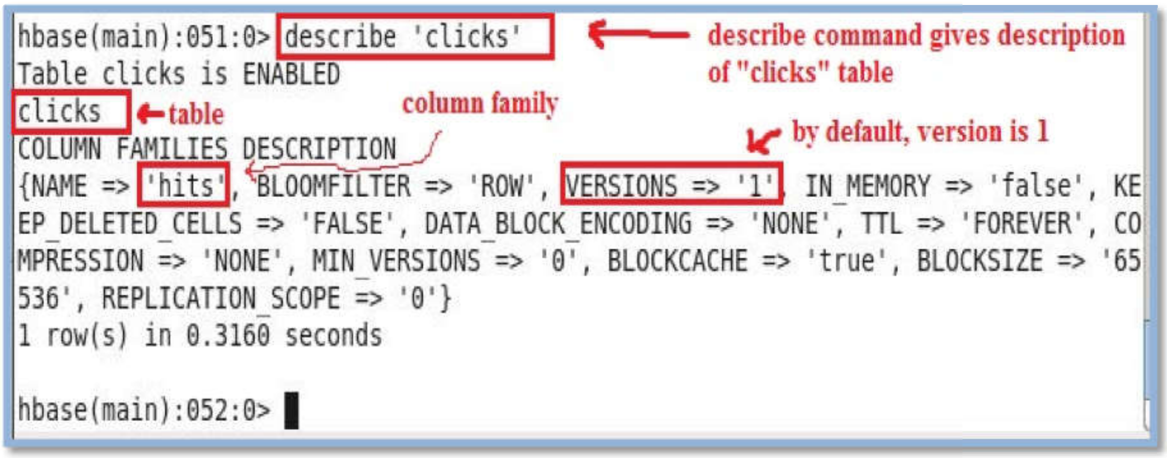
Step 1: Create table "clicks" with single column family "hits"



```
hbase(main):019:0> create 'clicks','hits'
0 row(s) in 2.4050 seconds

=> Hbase::Table - clicks
hbase(main):020:0> list
TABLE
clicks
1 row(s) in 0.0410 seconds

=> ["clicks"]
hbase(main):021:0>
```

command to created "clicks" table, with "hits" as a single column family
clicks table is created successfully

Step 2: Describe "clicks" table



```
hbase(main):051:0> describe 'clicks'
Table clicks is ENABLED
clicks
COLUMN FAMILIES DESCRIPTION
{NAME => 'hits', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KE
EP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', CO
MPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65
536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.3160 seconds

hbase(main):052:0>
```

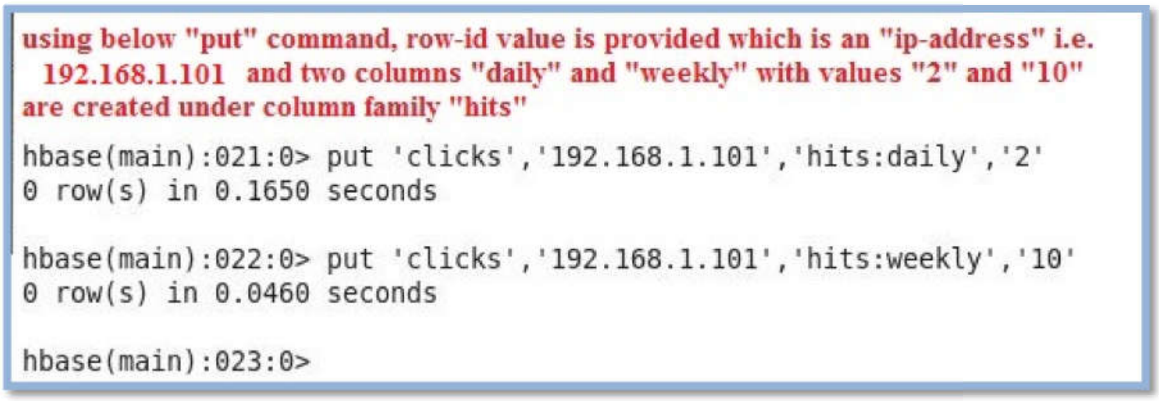
NOTE: Version information is altered/ updated to 5, while accomplishing Task 2.

Problem 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.

Below steps are followed to accomplish above task:

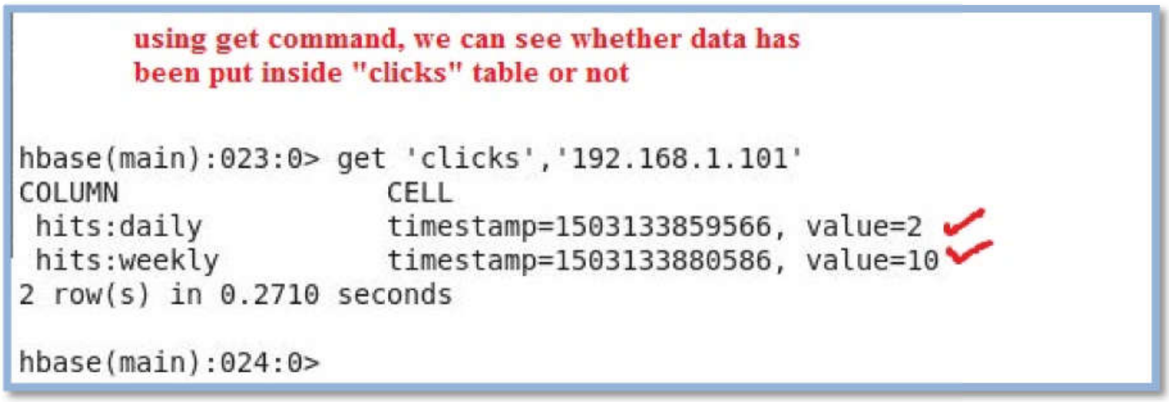
Step 1: **Insert data inside "click" table**



using below "put" command, row-id value is provided which is an "ip-address" i.e. 192.168.1.101 and two columns "daily" and "weekly" with values "2" and "10" are created under column family "hits"

```
hbase(main):021:0> put 'clicks','192.168.1.101','hits:daily','2'
0 row(s) in 0.1650 seconds

hbase(main):022:0> put 'clicks','192.168.1.101','hits:weekly','10'
0 row(s) in 0.0460 seconds

hbase(main):023:0>
```

Step 2: **Check whether data has been put inside "clicks" table or not, using get command**



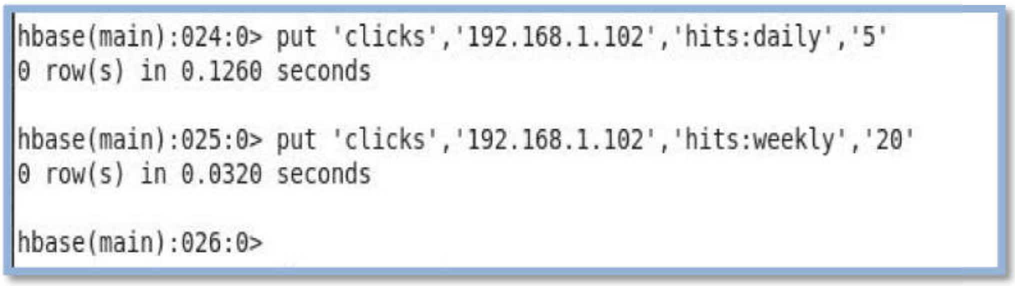
using get command, we can see whether data has been put inside "clicks" table or not

```
hbase(main):023:0> get 'clicks','192.168.1.101'
COLUMN                CELL
 hits:daily           timestamp=1503133859566, value=2 ✓
 hits:weekly          timestamp=1503133880586, value=10✓
2 row(s) in 0.2710 seconds

hbase(main):024:0>
```

Above "get" command displays column-family:columns, timestamp and values

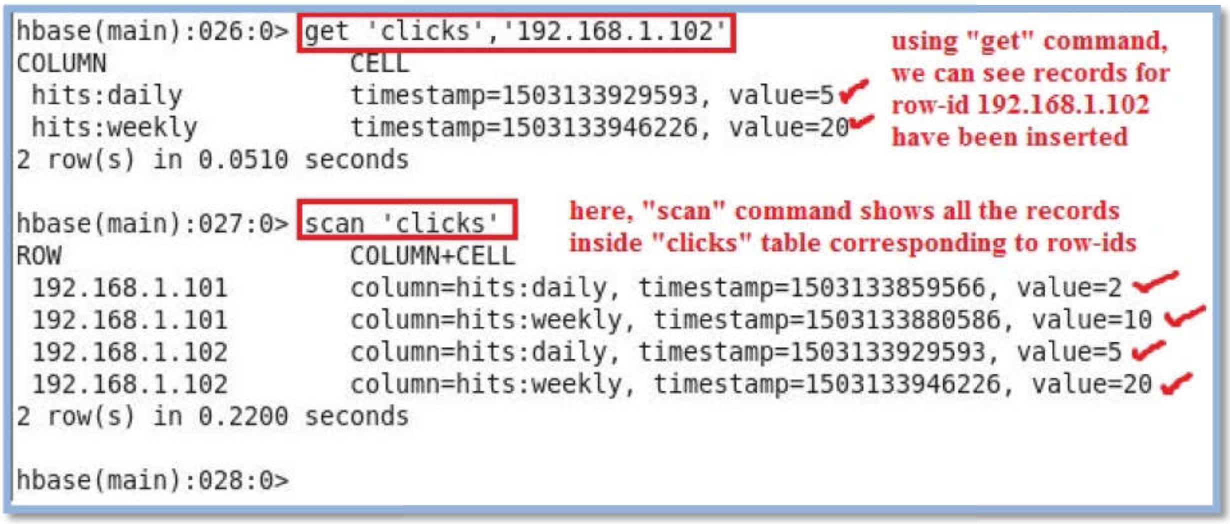
NOTE: Atleast one column-family needs to be specified along with table name in get command, else error with suggestion would be returned.

Step 3: **Putting new records for row-id "192.168.1.102"**



```
hbase(main):024:0> put 'clicks','192.168.1.102','hits:daily','5'
0 row(s) in 0.1260 seconds

hbase(main):025:0> put 'clicks','192.168.1.102','hits:weekly','20'
0 row(s) in 0.0320 seconds

hbase(main):026:0>
```

Step 4: Till now we have created four records, two records for row-id "192.168.1.101" and two records for row-id "192.168.1.102". Using get and scan command we can see the inserted records

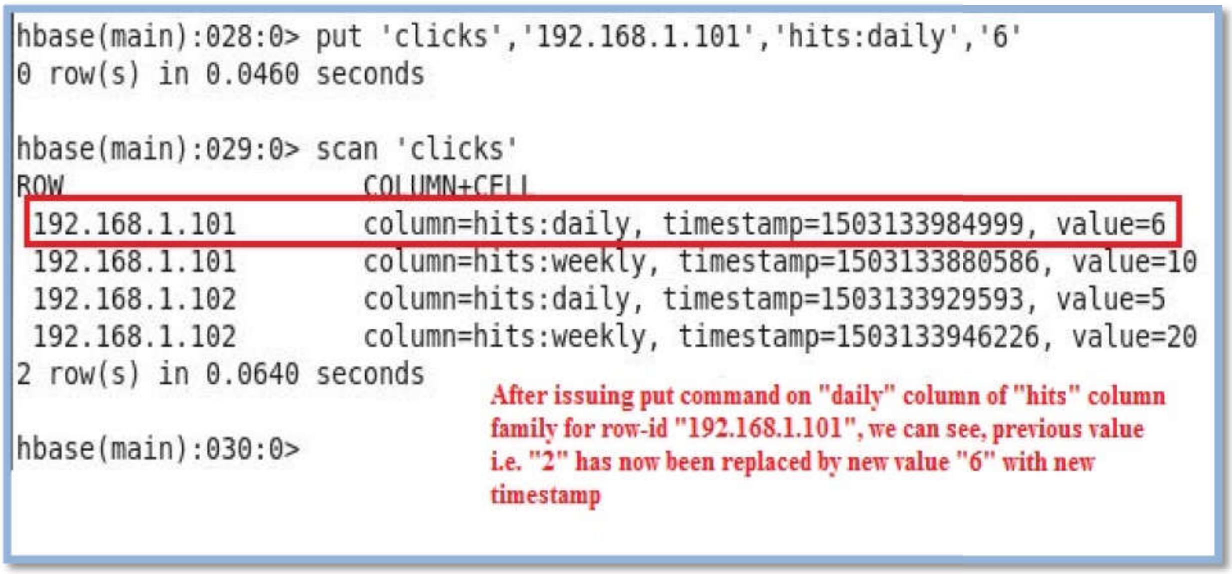


```
hbase(main):026:0> get 'clicks','192.168.1.102'
COLUMN                CELL
 hits:daily           timestamp=1503133929593, value=5
 hits:weekly          timestamp=1503133946226, value=20
2 row(s) in 0.0510 seconds

hbase(main):027:0> scan 'clicks'
ROW                   COLUMN+CELL
 192.168.1.101        column=hits:daily, timestamp=1503133859566, value=2
 192.168.1.101        column=hits:weekly, timestamp=1503133880586, value=10
 192.168.1.102        column=hits:daily, timestamp=1503133929593, value=5
 192.168.1.102        column=hits:weekly, timestamp=1503133946226, value=20
2 row(s) in 0.2200 seconds

hbase(main):028:0>
```

using "get" command, we can see records for row-id 192.168.1.102 have been inserted

here, "scan" command shows all the records inside "clicks" table corresponding to row-ids

Step 5: Updating value of column "daily" for column-family "hits" for row-id "192.168.1.101"



```
hbase(main):028:0> put 'clicks','192.168.1.101','hits:daily','6'
0 row(s) in 0.0460 seconds

hbase(main):029:0> scan 'clicks'
ROW                   COLUMN+CELL
 192.168.1.101        column=hits:daily, timestamp=1503133984999, value=6
 192.168.1.101        column=hits:weekly, timestamp=1503133880586, value=10
 192.168.1.102        column=hits:daily, timestamp=1503133929593, value=5
 192.168.1.102        column=hits:weekly, timestamp=1503133946226, value=20
2 row(s) in 0.0640 seconds

hbase(main):030:0>
```

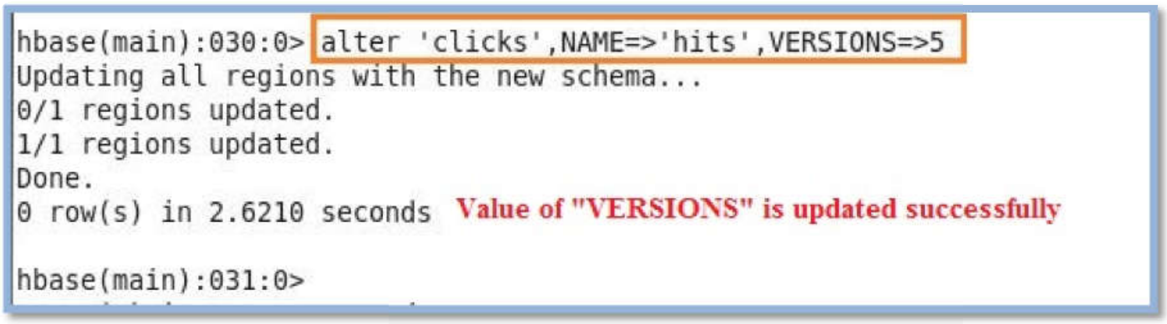
After issuing put command on "daily" column of "hits" column family for row-id "192.168.1.101", we can see, previous value i.e. "2" has now been replaced by new value "6" with new timestamp

Explanation of above screenshot:

Here, old value is replaced by new value, because by default "VERSIONS" is set to 1. So if we want that five versions of column-family "hits" must be maintained, then we need to change value of "VERSIONS" to 5 explicitly.

Refer next step for altering the value of VERSIONS to 5.

Step 6: **Update value of VERSIONS to 5**

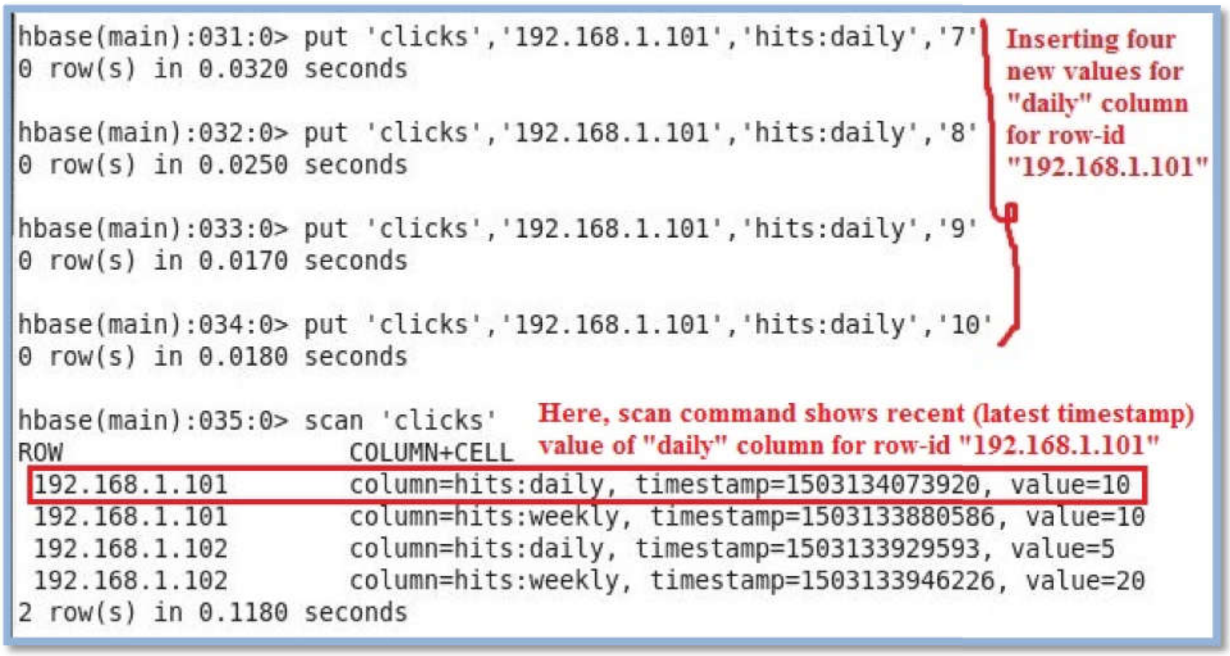


```
hbase(main):030:0> alter 'clicks',NAME=>'hits',VERSIONS=>5
Updating all regions with the new schema...
0/1 regions updated.
1/1 regions updated.
Done.
0 row(s) in 2.6210 seconds

hbase(main):031:0>
```

Since, value of "VERSIONS" is updated for column-family "hits", now by inserting new values for same column for same row-id, we can check whether different versions are maintained or not.

Refer next step for this.

Step 7: **Insert new values for "daily" column for row-id "192.168.1.101", and using "scan" check whether different versions for "daily" column of "hits" column-family are maintained or not.**



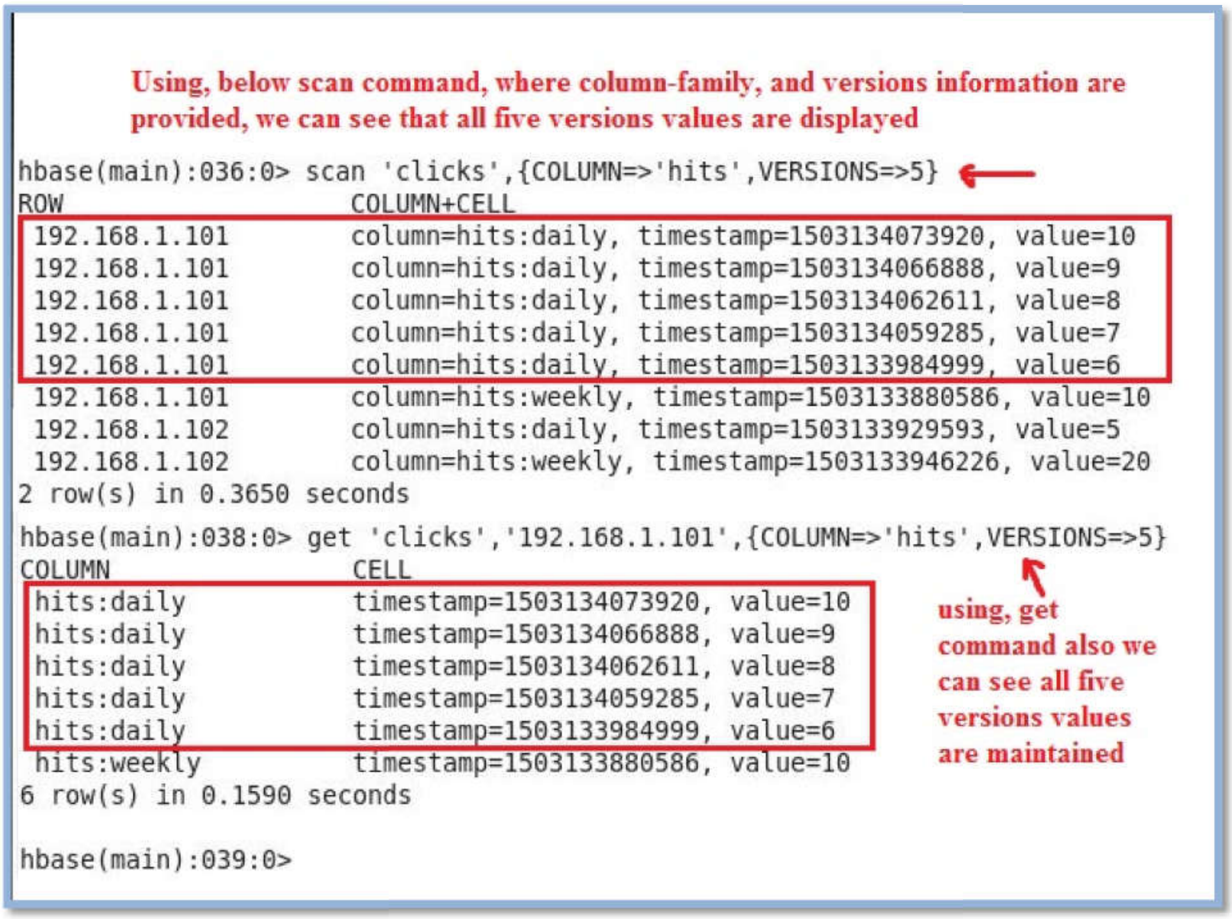
```
hbase(main):031:0> put 'clicks','192.168.1.101','hits:daily','7'
0 row(s) in 0.0320 seconds

hbase(main):032:0> put 'clicks','192.168.1.101','hits:daily','8'
0 row(s) in 0.0250 seconds

hbase(main):033:0> put 'clicks','192.168.1.101','hits:daily','9'
0 row(s) in 0.0170 seconds

hbase(main):034:0> put 'clicks','192.168.1.101','hits:daily','10'
0 row(s) in 0.0180 seconds

hbase(main):035:0> scan 'clicks'
ROW                   COLUMN+CELL
 192.168.1.101        column=hits:daily, timestamp=1503134073920, value=10
 192.168.1.101        column=hits:weekly, timestamp=1503133880586, value=10
 192.168.1.102        column=hits:daily, timestamp=1503133929593, value=5
 192.168.1.102        column=hits:weekly, timestamp=1503133946226, value=20
2 row(s) in 0.1180 seconds
```

Inserting four new values for "daily" column for row-id "192.168.1.101"
Here, scan command shows recent (latest timestamp) value of "daily" column for row-id "192.168.1.101"

In above screenshot, we can see, "scan" command displays only latest value 10 for "daily" column of "hits" column-family for row-id "192.168.1.101".

Now to see five versions of column family, refer next step.

Step 8: Show all five versions values for “daily” column of “hit” column-family

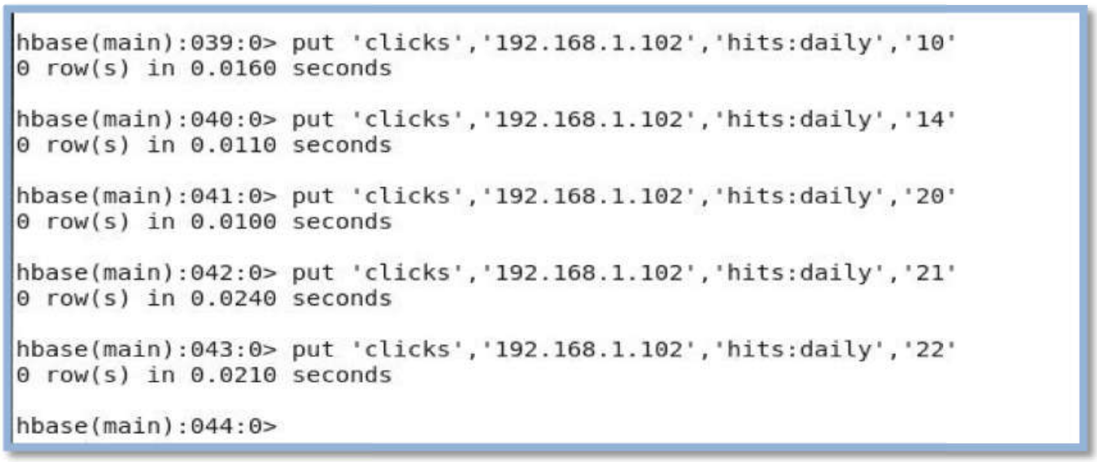


Using, below scan command, where column-family, and versions information are provided, we can see that all five versions values are displayed

```
hbase(main):036:0> scan 'clicks',{COLUMN=>'hits',VERSIONS=>5}
ROW                  COLUMN+CELL
 192.168.1.101       column=hits:daily, timestamp=1503134073920, value=10
 192.168.1.101       column=hits:daily, timestamp=1503134066888, value=9
 192.168.1.101       column=hits:daily, timestamp=1503134062611, value=8
 192.168.1.101       column=hits:daily, timestamp=1503134059285, value=7
 192.168.1.101       column=hits:daily, timestamp=1503133984999, value=6
 192.168.1.101       column=hits:weekly, timestamp=1503133880586, value=10
 192.168.1.102       column=hits:daily, timestamp=1503133929593, value=5
 192.168.1.102       column=hits:weekly, timestamp=1503133946226, value=20
2 row(s) in 0.3650 seconds
hbase(main):038:0> get 'clicks','192.168.1.101',{COLUMN=>'hits',VERSIONS=>5}
COLUMN               CELL
 hits:daily          timestamp=1503134073920, value=10
 hits:daily          timestamp=1503134066888, value=9
 hits:daily          timestamp=1503134062611, value=8
 hits:daily          timestamp=1503134059285, value=7
 hits:daily          timestamp=1503133984999, value=6
 hits:weekly         timestamp=1503133880586, value=10
6 row(s) in 0.1590 seconds

hbase(main):039:0>
```

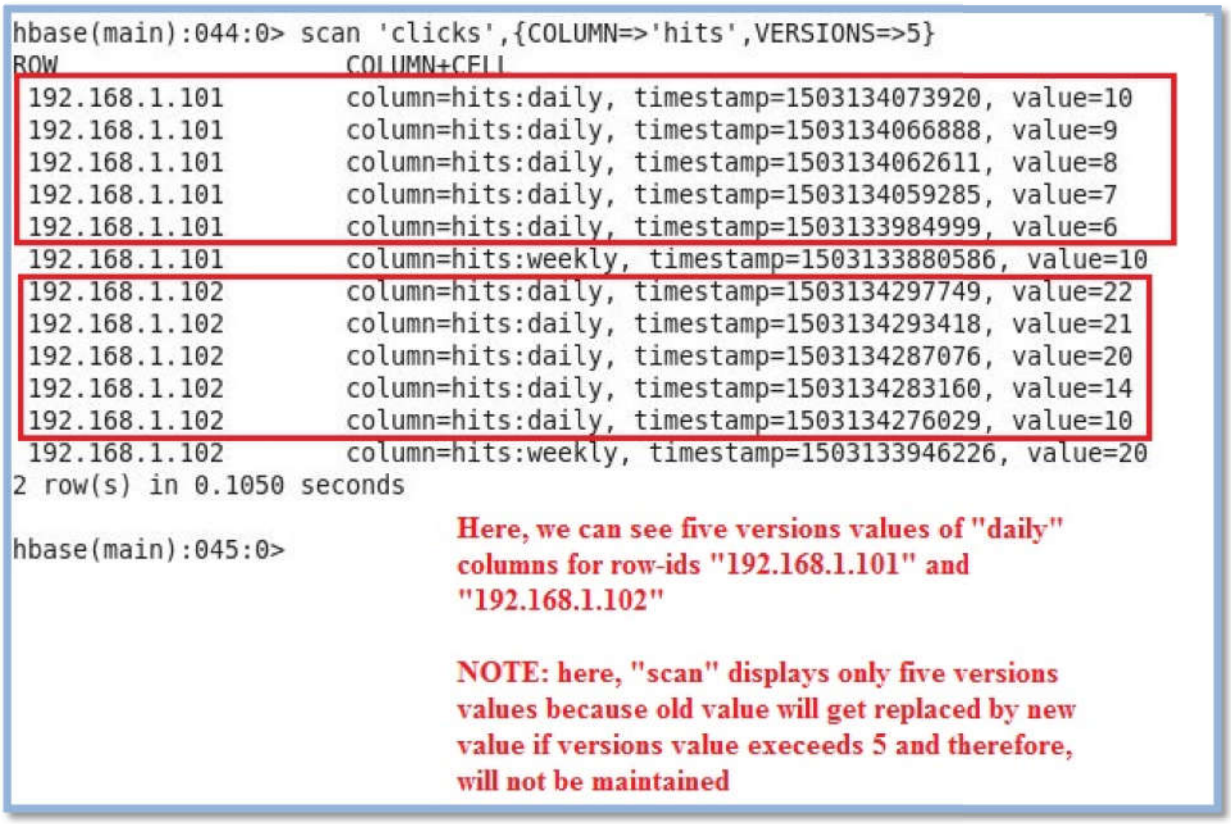
using, get command also we can see all five versions values are maintained

Step 9: Now insert 5 new records for “daily” column of row-id “192.168.1.102”



```
hbase(main):039:0> put 'clicks','192.168.1.102','hits:daily','10'
0 row(s) in 0.0160 seconds

hbase(main):040:0> put 'clicks','192.168.1.102','hits:daily','14'
0 row(s) in 0.0110 seconds

hbase(main):041:0> put 'clicks','192.168.1.102','hits:daily','20'
0 row(s) in 0.0100 seconds

hbase(main):042:0> put 'clicks','192.168.1.102','hits:daily','21'
0 row(s) in 0.0240 seconds

hbase(main):043:0> put 'clicks','192.168.1.102','hits:daily','22'
0 row(s) in 0.0210 seconds

hbase(main):044:0>
```

Step 10: **Check using scan, different versions values of "daily" column for both row-ids**

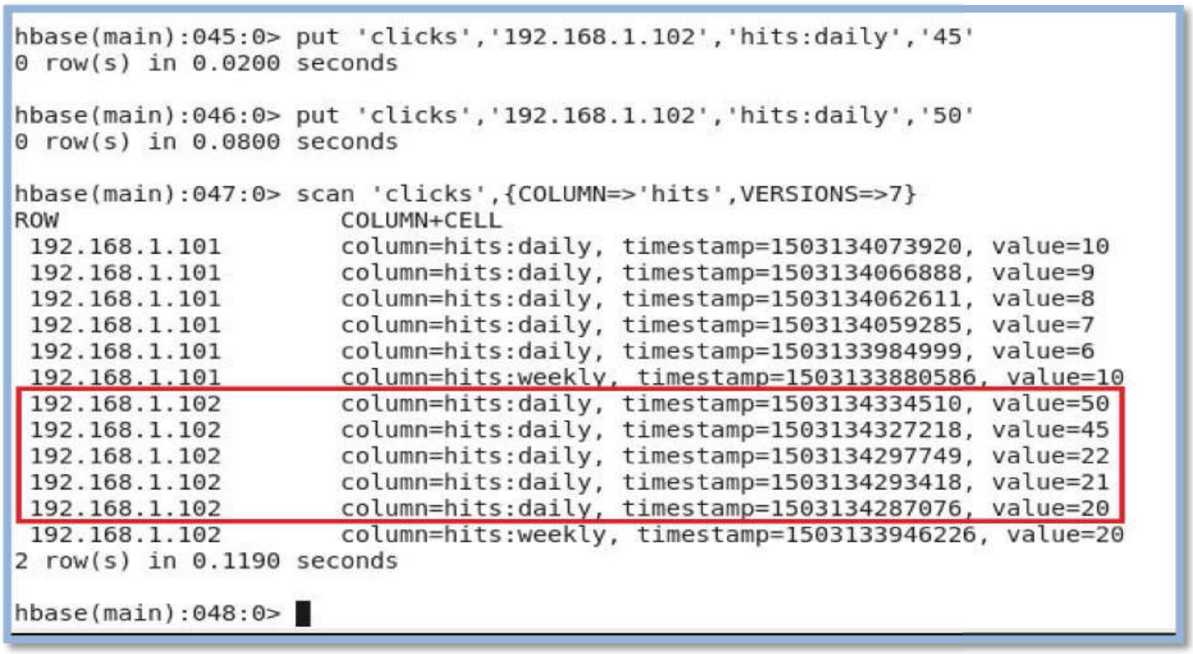


```
hbase(main):044:0> scan 'clicks',{COLUMN=>'hits',VERSIONS=>5}
ROW                         COLUMN+CELL
 192.168.1.101              column=hits:daily, timestamp=1503134073920, value=10
 192.168.1.101              column=hits:daily, timestamp=1503134066888, value=9
 192.168.1.101              column=hits:daily, timestamp=1503134062611, value=8
 192.168.1.101              column=hits:daily, timestamp=1503134059285, value=7
 192.168.1.101              column=hits:daily, timestamp=1503133984999, value=6
 192.168.1.101              column=hits:weekly, timestamp=1503133880586, value=10
 192.168.1.102              column=hits:daily, timestamp=1503134297749, value=22
 192.168.1.102              column=hits:daily, timestamp=1503134293418, value=21
 192.168.1.102              column=hits:daily, timestamp=1503134287076, value=20
 192.168.1.102              column=hits:daily, timestamp=1503134283160, value=14
 192.168.1.102              column=hits:daily, timestamp=1503134276029, value=10
 192.168.1.102              column=hits:weekly, timestamp=1503133946226, value=20
2 row(s) in 0.1050 seconds

hbase(main):045:0>
```

Here, we can see five versions values of "daily" columns for row-ids "192.168.1.101" and "192.168.1.102"

NOTE: here, "scan" displays only five versions values because old value will get replaced by new value if versions value execeeds 5 and therefore, will not be maintained

Step 11: **Again insert few records for daily column for row-id "192.168.1.102", and try scan command with "VERSION=>7"**



```
hbase(main):045:0> put 'clicks','192.168.1.102','hits:daily','45'
0 row(s) in 0.0200 seconds

hbase(main):046:0> put 'clicks','192.168.1.102','hits:daily','50'
0 row(s) in 0.0800 seconds

hbase(main):047:0> scan 'clicks',{COLUMN=>'hits',VERSIONS=>7}
ROW                         COLUMN+CELL
 192.168.1.101              column=hits:daily, timestamp=1503134073920, value=10
 192.168.1.101              column=hits:daily, timestamp=1503134066888, value=9
 192.168.1.101              column=hits:daily, timestamp=1503134062611, value=8
 192.168.1.101              column=hits:daily, timestamp=1503134059285, value=7
 192.168.1.101              column=hits:daily, timestamp=1503133984999, value=6
 192.168.1.101              column=hits:weekly, timestamp=1503133880586, value=10
 192.168.1.102              column=hits:daily, timestamp=1503134334510, value=50
 192.168.1.102              column=hits:daily, timestamp=1503134327218, value=45
 192.168.1.102              column=hits:daily, timestamp=1503134297749, value=22
 192.168.1.102              column=hits:daily, timestamp=1503134293418, value=21
 192.168.1.102              column=hits:daily, timestamp=1503134287076, value=20
 192.168.1.102              column=hits:weekly, timestamp=1503133946226, value=20
2 row(s) in 0.1190 seconds

hbase(main):048:0> █
```

Explanation of above screenshot:

In above screenshot, we can see, after inserting two new values for “daily” column for row-id “192.168.1.102”, only 5 records are shown not 7, however in “scan” command VERSIONS=>7 is mentioned.

This is because, initially using “alter” command, we set VERSIONS=>5, so only 5 versions are maintained, therefore if new values come after five versions values, then old value is replaced by new value.

In Step 10 Screenshot, we can see values for “daily” column for row-id “192.168.1.102” are “22,21,20,14,10” and now values are “50,45,22,21,20”, values “14 and 10” are discarded because their timestamp values are old.