EXPERIMENT NO: 3

AIM: Programming exercises on HBASE.

Theory:

HBase, short for Hadoop Database, is an open-source, distributed, and scalable NoSQL database that is designed to handle large volumes of sparse data. It is part of the Hadoop ecosystem and is often used for real-time, read and write access to big data. Here's a comprehensive overview of HBase:

Key Characteristics and Concepts:

1. **NoSQL Database**: HBase is a NoSQL database, which means it is not based on traditional relational database models. Instead, it uses a schema-less design, allowing for flexible data storage.
2. **Distributed and Scalable**: HBase is built on the Hadoop Distributed File System (HDFS) and is designed to scale horizontally, making it capable of handling petabytes of data spread across a cluster of commodity hardware.
3. **Columnar Storage**: HBase stores data in tables, similar to traditional databases. However, the data within tables is stored in a columnar format, which enables efficient data retrieval and compression.
4. **Sparse Data Model**: HBase is optimized for sparse data, meaning that it is well-suited for situations where data is missing or not present in every record. This makes it particularly useful for handling semi-structured or rapidly changing data.
5. **Strong Consistency**: HBase provides strong consistency guarantees, which means that data is consistent and up-to-date across the entire cluster.
6. **High Write Throughput**: It excels in write-heavy workloads, making it ideal for applications with frequent data inserts and updates.

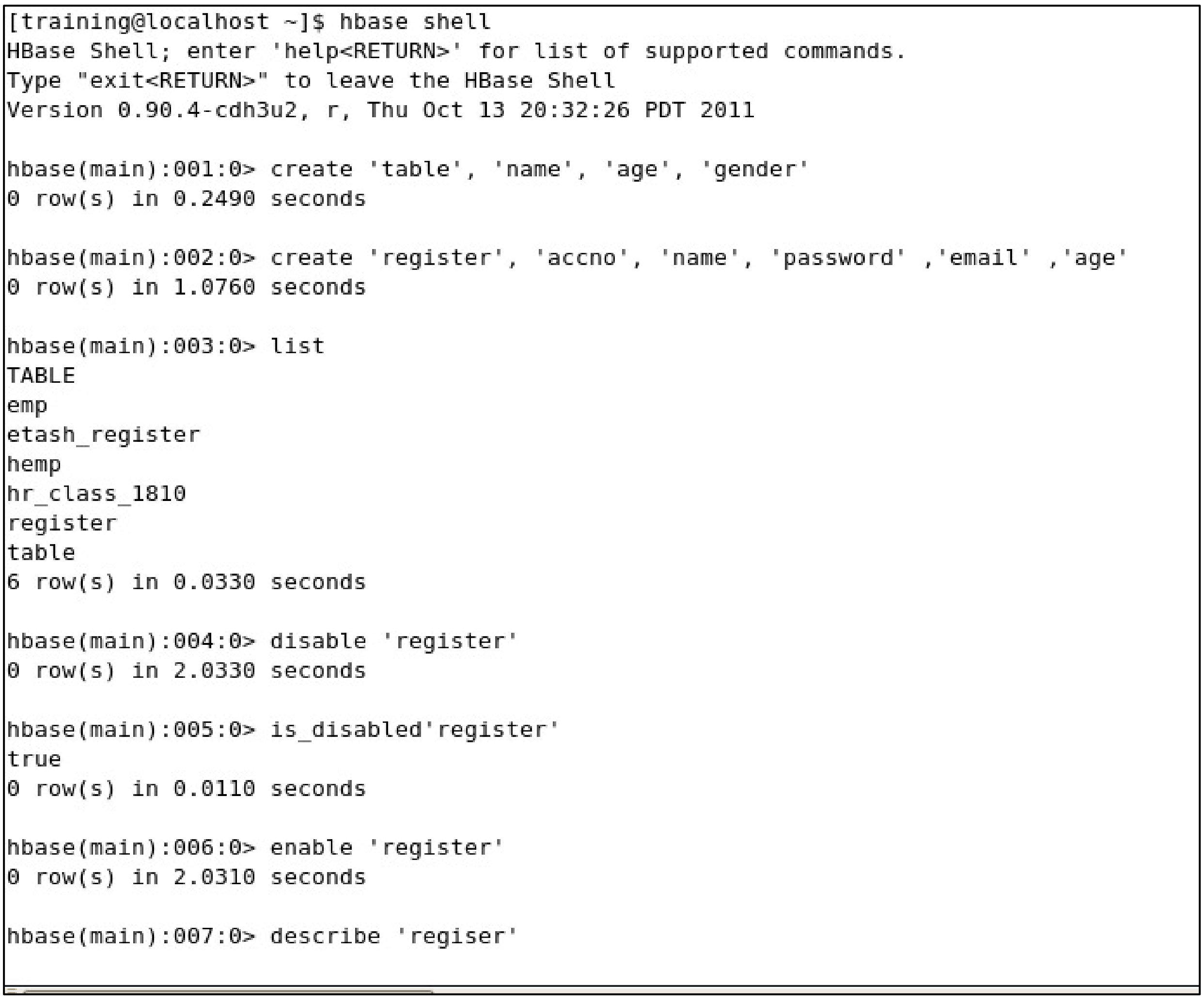
HBase is a powerful NoSQL database designed for handling large-scale, sparse data and real-time access. It plays a significant role in the Hadoop ecosystem, complementing other tools for big data processing and analytics. Its distributed architecture, scalability, and flexibility make it a valuable choice for various applications, particularly those requiring high write throughput and real-time data access.

STEP 1:

Starting with HBase shell, creating ‘register’ and learning how to disable and enable the table.

syntax

Create ‘table’ ‘columns’, ‘columns’

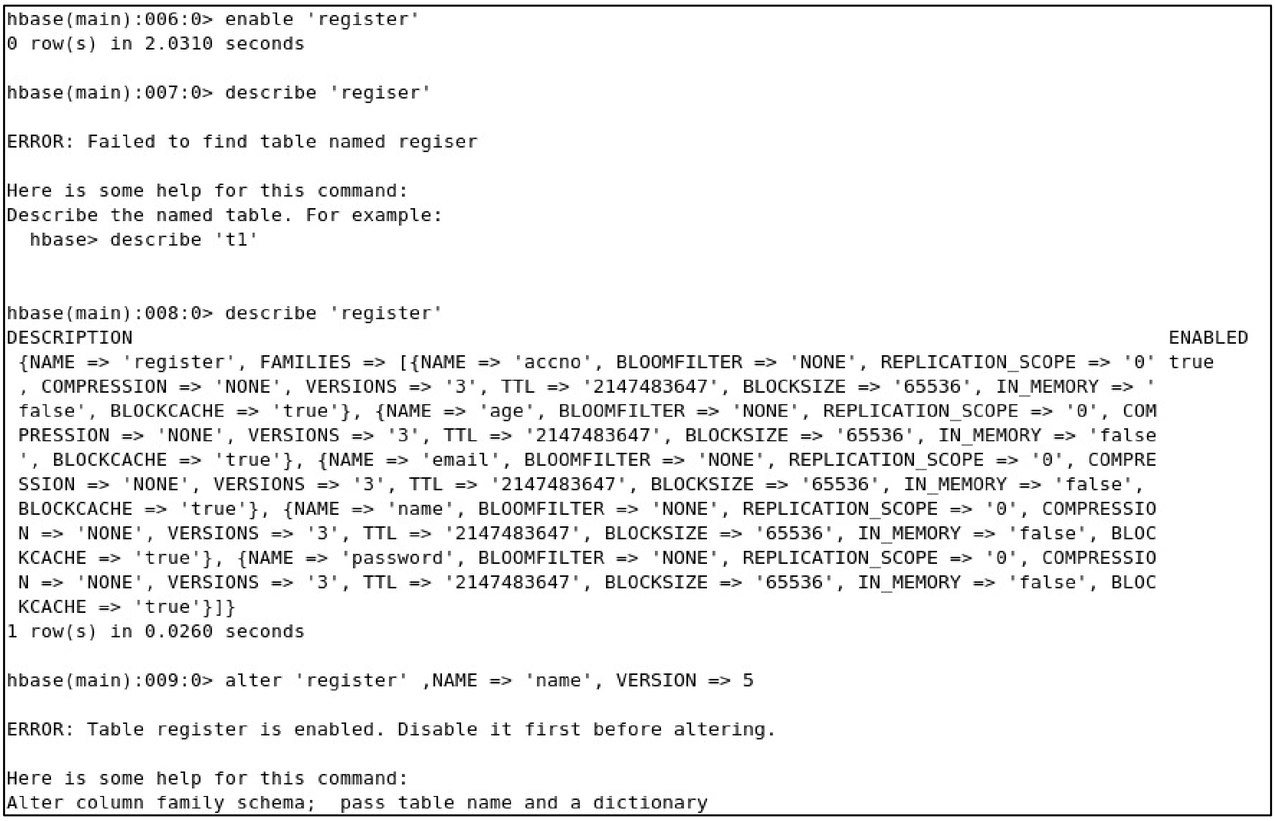


```
[training@localhost ~]$ hbase shell
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.90.4-cdh3u2, r, Thu Oct 13 20:32:26 PDT 2011

hbase(main):001:0> create 'table', 'name', 'age', 'gender'
0 row(s) in 0.2490 seconds

hbase(main):002:0> create 'register', 'accno', 'name', 'password' ,'email' ,'age'
0 row(s) in 1.0760 seconds

hbase(main):003:0> list
TABLE
emp
etash_register
hemp
hr_class_1810
register
table
6 row(s) in 0.0330 seconds

hbase(main):004:0> disable 'register'
0 row(s) in 2.0330 seconds

hbase(main):005:0> is_disabled'register'
true
0 row(s) in 0.0110 seconds

hbase(main):006:0> enable 'register'
0 row(s) in 2.0310 seconds

hbase(main):007:0> describe 'regiser'
```

Step 2: accessing various columns and describing table:

Syntax:

Describe ‘register’



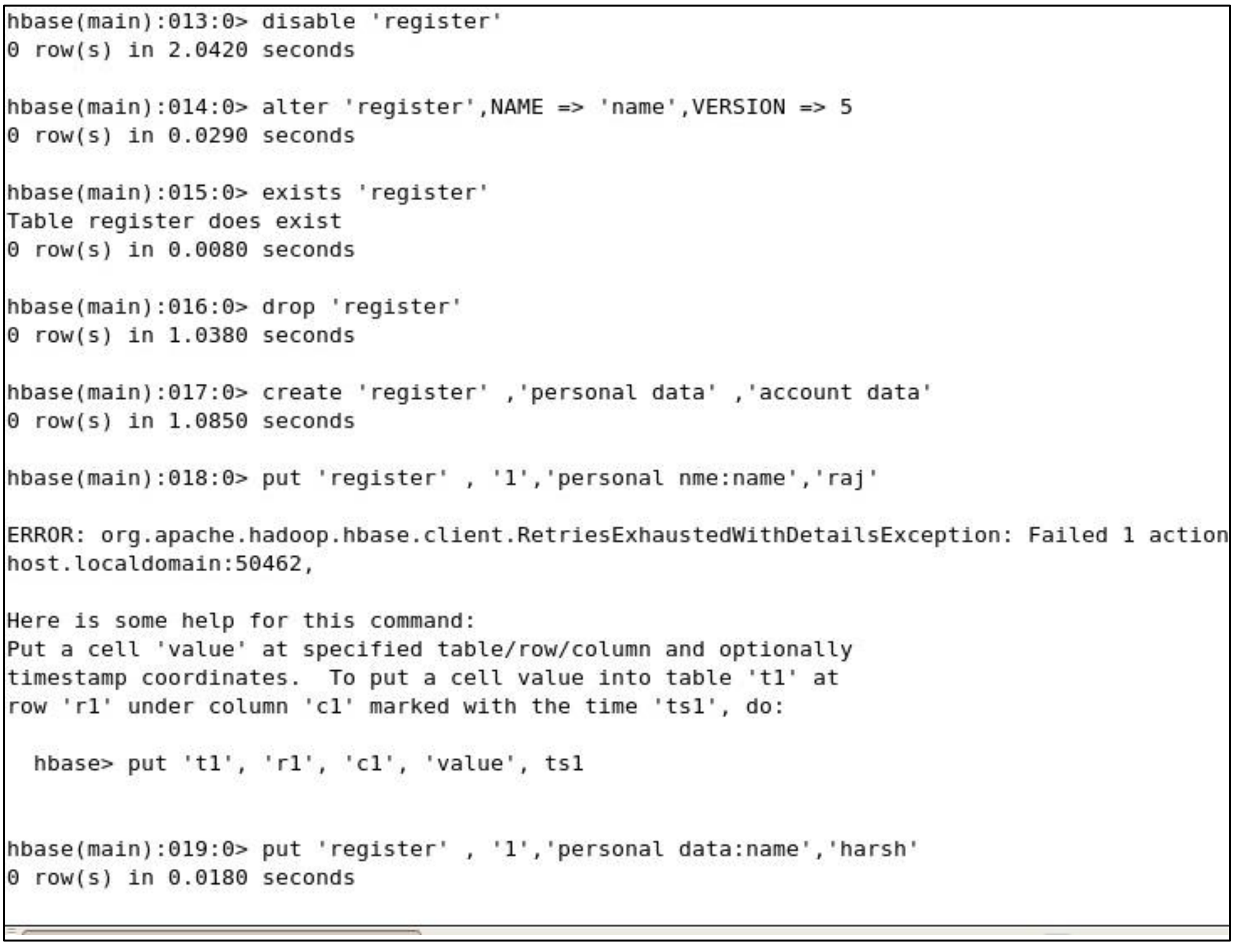
```
hbase(main):006:0> enable 'register'
0 row(s) in 2.0310 seconds

hbase(main):007:0> describe 'regiser'

ERROR: Failed to find table named regiser

Here is some help for this command:
Describe the named table. For example:
  hbase> describe 't1'


hbase(main):008:0> describe 'register'
DESCRIPTION                                                                                          ENABLED
 {NAME => 'register', FAMILIES => [{NAME => 'accno', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0' true
 , COMPRESSION => 'NONE', VERSIONS => '3', TTL => '2147483647', BLOCKSIZE => '65536', IN_MEMORY => '
 false', BLOCKCACHE => 'true'}, {NAME => 'age', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', COM
 PRESSION => 'NONE', VERSIONS => '3', TTL => '2147483647', BLOCKSIZE => '65536', IN_MEMORY => 'false
 ', BLOCKCACHE => 'true'}, {NAME => 'email', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', COMPRE
 SSION => 'NONE', VERSIONS => '3', TTL => '2147483647', BLOCKSIZE => '65536', IN_MEMORY => 'false',
 BLOCKCACHE => 'true'}, {NAME => 'name', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', COMPRESSIO
 N => 'NONE', VERSIONS => '3', TTL => '2147483647', BLOCKSIZE => '65536', IN_MEMORY => 'false', BLOC
 KCACHE => 'true'}, {NAME => 'password', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', COMPRESSIO
 N => 'NONE', VERSIONS => '3', TTL => '2147483647', BLOCKSIZE => '65536', IN_MEMORY => 'false', BLOC
 KCACHE => 'true'}]}
1 row(s) in 0.0260 seconds

hbase(main):009:0> alter 'register' ,NAME => 'name', VERSION => 5

ERROR: Table register is enabled. Disable it first before altering.

Here is some help for this command:
Alter column family schema;  pass table name and a dictionary
```

Step 3: working on various commands

Alter’table_name’

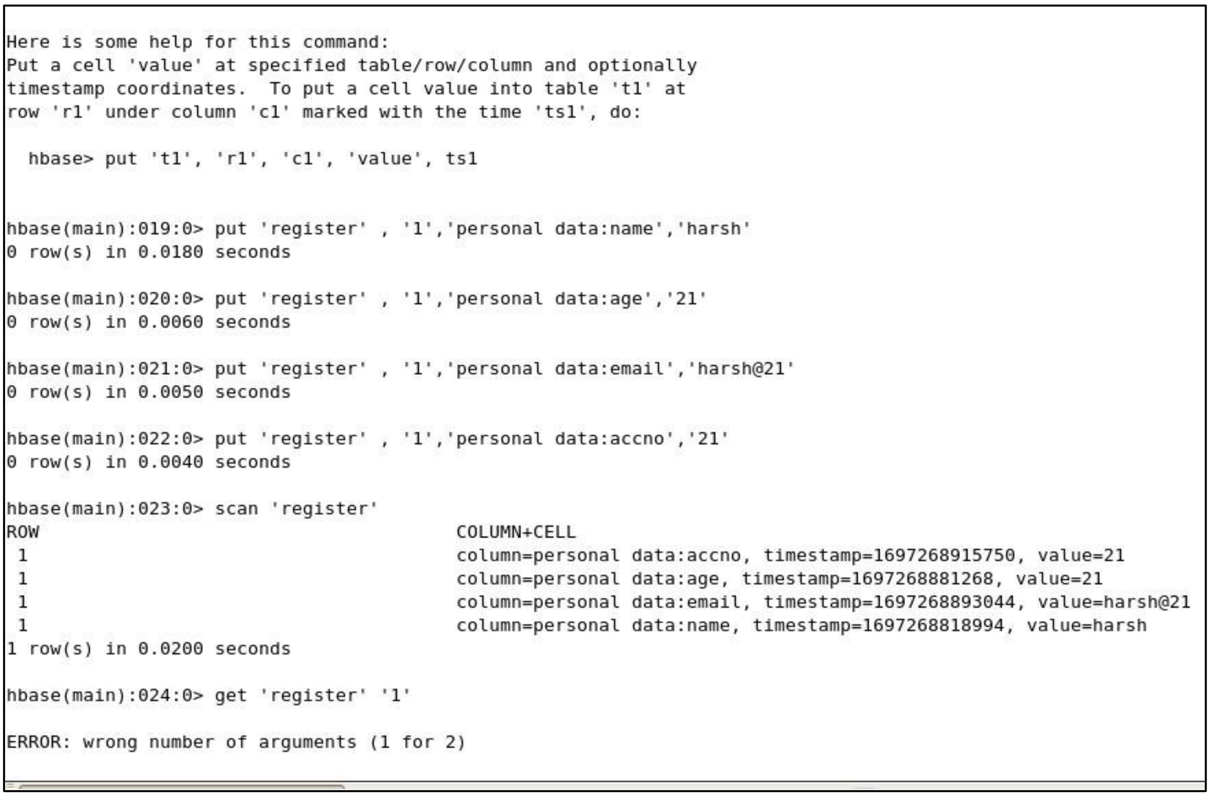
Drop”column_name”

Put “table_name’ , ‘data’

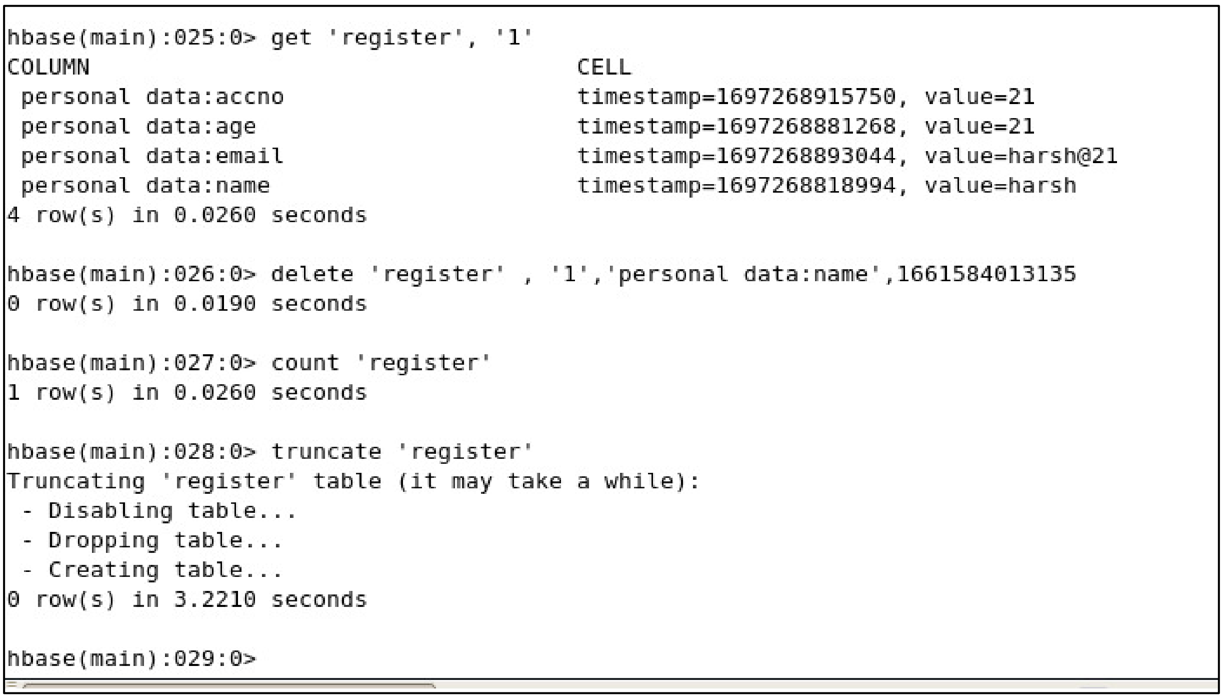


```
hbase(main):013:0> disable 'register'
0 row(s) in 2.0420 seconds

hbase(main):014:0> alter 'register',NAME => 'name',VERSION => 5
0 row(s) in 0.0290 seconds

hbase(main):015:0> exists 'register'
Table register does exist
0 row(s) in 0.0080 seconds

hbase(main):016:0> drop 'register'
0 row(s) in 1.0380 seconds

hbase(main):017:0> create 'register' ,'personal data' ,'account data'
0 row(s) in 1.0850 seconds

hbase(main):018:0> put 'register' , '1','personal nme:name','raj'

ERROR: org.apache.hadoop.hbase.client.RetriesExhaustedWithDetailsException: Failed 1 action
host.localdomain:50462,

Here is some help for this command:
Put a cell 'value' at specified table/row/column and optionally
timestamp coordinates.  To put a cell value into table 't1' at
row 'r1' under column 'c1' marked with the time 'ts1', do:

  hbase> put 't1', 'r1', 'c1', 'value', ts1


hbase(main):019:0> put 'register' , '1','personal data:name','harsh'
0 row(s) in 0.0180 seconds
```

Step 4: truncating table and other commands :



```
Here is some help for this command:
Put a cell 'value' at specified table/row/column and optionally
timestamp coordinates.  To put a cell value into table 't1' at
row 'r1' under column 'c1' marked with the time 'ts1', do:

  hbase> put 't1', 'r1', 'c1', 'value', ts1


hbase(main):019:0> put 'register' , '1','personal data:name','harsh'
0 row(s) in 0.0180 seconds

hbase(main):020:0> put 'register' , '1','personal data:age','21'
0 row(s) in 0.0060 seconds

hbase(main):021:0> put 'register' , '1','personal data:email','harsh@21'
0 row(s) in 0.0050 seconds

hbase(main):022:0> put 'register' , '1','personal data:accno','21'
0 row(s) in 0.0040 seconds

hbase(main):023:0> scan 'register'
ROW                                     COLUMN+CELL
 1                                      column=personal data:accno, timestamp=1697268915750, value=21
 1                                      column=personal data:age, timestamp=1697268881268, value=21
 1                                      column=personal data:email, timestamp=1697268893044, value=harsh@21
 1                                      column=personal data:name, timestamp=1697268818994, value=harsh
1 row(s) in 0.0200 seconds

hbase(main):024:0> get 'register' '1'

ERROR: wrong number of arguments (1 for 2)
```



```
hbase(main):025:0> get 'register', '1'
COLUMN                                  CELL
 personal data:accno                    timestamp=1697268915750, value=21
 personal data:age                      timestamp=1697268881268, value=21
 personal data:email                    timestamp=1697268893044, value=harsh@21
 personal data:name                     timestamp=1697268818994, value=harsh
4 row(s) in 0.0260 seconds

hbase(main):026:0> delete 'register' , '1','personal data:name',1661584013135
0 row(s) in 0.0190 seconds

hbase(main):027:0> count 'register'
1 row(s) in 0.0260 seconds

hbase(main):028:0> truncate 'register'
Truncating 'register' table (it may take a while):
 - Disabling table...
 - Dropping table...
 - Creating table...
0 row(s) in 3.2210 seconds

hbase(main):029:0>
```