Use Case 1:
Creating a table with name Table01 and adding a column family Location which contains country, state and city columns

> create 'Table01', 'Location'
> put 'Table01', '1', 'Location:country' ,'USA'
> put 'Table01', '1', 'Location:state' , 'Illinois'
> put 'Table01', '1', 'Location:City', 'Chicago'
> put 'Table01', '2', 'Location:country' ,'USA'
> put 'Table01', '2', 'Location:state' , 'Missouri'
> put 'Table01', '2', 'Location:City', 'Kansas City'
> put 'Table01', '3', 'Location:country' ,'India'
> put 'Table01', '3', 'Location:state' , 'AP'
> put 'Table01', '3', 'Location:City', 'Vizag'

scan 'Table1'
describe 'Table01'
count 'Table01' Data Manipulation Command used to count no of rows in a table
get 'Table01','1' Data Manipulation Command used to get the all the column values of a particular row

```
t to table 't1', the corresponding command would be:

  hbase> t.put 'r1', 'c1', 'value', ts1, {ATTRIBUTES=>{'mykey'=>'myvalue'}}


hbase(main):004:0> put 'Table01', '1', 'Location:country' ,'USA'
0 row(s) in 0.0200 seconds

hbase(main):005:0> put 'Table01', '1', 'Location:state' , 'Illinois'
0 row(s) in 0.0160 seconds

hbase(main):006:0> put 'Table01', '1', 'Location:City', 'Chicago'
0 row(s) in 0.0230 seconds

hbase(main):007:0> put 'Table01', '2', 'Location:country' ,'USA'
0 row(s) in 0.0140 seconds

hbase(main):008:0> put 'Table01', '2', 'Location:state' , 'Missouri'
0 row(s) in 0.0140 seconds

hbase(main):009:0> put 'Table01', '2', 'Location:City', 'Kansas City'
0 row(s) in 0.0130 seconds

hbase(main):010:0> put 'Table01', '3', 'Location:country' ,'India'
0 row(s) in 0.0140 seconds

hbase(main):011:0> put 'Table01', '3', 'Location:state' , 'AP'
0 row(s) in 0.0150 seconds

hbase(main):012:0> put 'Table01', '3', 'Location:City', 'Vizag'
0 row(s) in 0.0130 seconds

hbase(main):013:0> scan 'Table01'
ROW                   COLUMN+CELL
 1                    column=Location:City, timestamp=1529188115842, value=Chica
                      go
 1                    column=Location:country, timestamp=1529188011814, value=US
                      A
 1                    column=Location:state, timestamp=1529188105655, value=Illi
                      nois
 2                    column=Location:City, timestamp=1529188157008, value=Kansa
                      s City
 2                    column=Location:country, timestamp=1529188138357, value=US
                      A
 2                    column=Location:state, timestamp=1529188148186, value=Miss
                      ouri
 3                    column=Location:City, timestamp=1529188186058, value=Vizag
 3                    column=Location:country, timestamp=1529188168328, value=In
                      dia
 3                    column=Location:state, timestamp=1529188176612, value=AP
3 row(s) in 0.0330 seconds

hbase(main):014:0> describe 'Table01'
Table Table01 is ENABLED
Table01
COLUMN FAMILIES DESCRIPTION
{NAME => 'Location', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false'
, KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER'
, COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE =>
'65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.0320 seconds

hbase(main):015:0> count 'Table01'
3 row(s) in 0.0340 seconds

=> 3
hbase(main):016:0> get 'Table01','1'
COLUMN                CELL
 Location:City        timestamp=1529188115842, value=Chicago
 Location:country     timestamp=1529188011814, value=USA
 Location:state       timestamp=1529188105655, value=Illinois
3 row(s) in 0.0260 seconds

hbase(main):017:0>
```

Use Case 2:
Creating a table with name Table2 and adding a column family Student and Courses which contains Student details and courses enrolled by each student

```
> create 'Table2','Student', 'Courses'
> put 'Table2', '1', 'Student:Name','Lalitha'
> put 'Table2', '1', 'Student:SEX', 'Female'
> put 'Table2', '1', 'Student:AGE', '20'
> put 'Table2', '2', 'Student:Name','Vinay'
> put 'Table2', '2', 'Student:SEX', 'Male'
> put 'Table2', '2', 'Student:AGE', '20'
> put 'Table2', '3', 'Student:Name','Raj'
> put 'Table2', '3', 'Student:SEX', 'Male'
> put 'Table2', '3', 'Student:AGE', '20'
> put 'Table2', '1', 'Courses:Course1','ISL'
> put 'Table2', '1', 'Courses:Course2', 'Big Data'
> put 'Table2', '2', 'Courses:Course1','Python'
> put 'Table2', '2', 'Courses:Course2', 'ASE'
> put 'Table2', '3', 'Courses:Course1','Big Data'
> put 'Table2', '3', 'Courses:Course2', 'ISL'
scan 'Table2'
describe 'Table2'
count 'Table2' Data Manipulation Command used to count no of rows in a table
get 'Table2','2' Data Manipulation Command used to get the all the column values of a particular row
```

```
hbase(main):017:0> create 'Table2','Student','Courses'
0 row(s) in 2.2670 seconds

=> Hbase::Table - Table2
hbase(main):018:0> put 'Table2', '1', 'Student:Name','Lalitha'
0 row(s) in 0.0180 seconds

hbase(main):019:0> put 'Table2', '1', 'Student:SEX', 'Female'
0 row(s) in 0.0150 seconds

hbase(main):020:0> put 'Table2', '1', 'Student:AGE', '20'
0 row(s) in 0.0160 seconds

hbase(main):021:0> put 'Table2', '2', 'Student:Name','Vinay'
0 row(s) in 0.0140 seconds

hbase(main):022:0> put 'Table2', '2', 'Student:SEX', 'Male'
0 row(s) in 0.0160 seconds

hbase(main):023:0> put 'Table2', '2', 'Student:AGE', '20'
0 row(s) in 0.0150 seconds

hbase(main):024:0> put 'Table2', '3', 'Student:Name','Raj'
0 row(s) in 0.0140 seconds

hbase(main):025:0> put 'Table2', '3', 'Student:SEX', 'Male'
0 row(s) in 0.0140 seconds

hbase(main):026:0> put 'Table2', '3', 'Student:AGE', '20'
0 row(s) in 0.0160 seconds

hbase(main):027:0> put 'Table2', '1', 'Courses : Course1','ISL'

ERROR: Unknown column family! Valid column names: Courses:*, Student:*

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 'ns1:t1' or 't1'
at row 'r1' under column 'c1' marked with the time 'ts1', do:

  hbase> put 'ns1:t1', 'r1', 'c1', 'value'
  hbase> put 't1', 'r1', 'c1', 'value'
  hbase> put 't1', 'r1', 'c1', 'value', ts1
  hbase> put 't1', 'r1', 'c1', 'value', {ATTRIBUTES=>{'mykey'=>'myvalue'}}
  hbase> put 't1', 'r1', 'c1', 'value', ts1, {ATTRIBUTES=>{'mykey'=>'myvalue'}}
  hbase> put 't1', 'r1', 'c1', 'value', ts1, {VISIBILITY=>'PRIVATE|SECRET'}

The same commands also can be run on a table reference. Suppose you had a reference
t to table 't1', the corresponding command would be:

  hbase> t.put 'r1', 'c1', 'value', ts1, {ATTRIBUTES=>{'mykey'=>'myvalue'}}


hbase(main):028:0> put 'Table2', '1', 'Courses:Course1','ISL'
0 row(s) in 0.0180 seconds

hbase(main):029:0> put 'Table2', '1', 'Courses:Course2', 'Big Data'
0 row(s) in 0.0130 seconds

hbase(main):030:0> put 'Table2', '2', 'Courses:Course1','Python'
0 row(s) in 0.0140 seconds

hbase(main):031:0> put 'Table2', '2', 'Courses:Course2', 'ASE'
0 row(s) in 0.0130 seconds

hbase(main):032:0> put 'Table2', '3', 'Courses:Course1','Big Data'
0 row(s) in 0.0140 seconds

hbase(main):033:0> put 'Table2', '3', 'Courses:Course2', 'ISL'
0 row(s) in 0.0120 seconds

hbase(main):034:0> scan 'Table2'
ROW                                                   COLUMN+CELL
```

```
Return a counter cell value at specified table/row/column coordinates.
A counter cell should be managed with atomic increment functions on HBase
and the data should be binary encoded (as long value). Example:

  hbase> get_counter 'ns1:t1', 'r1', 'c1'
  hbase> get_counter 't1', 'r1', 'c1'

The same commands also can be run on a table reference. Suppose you had a reference
t to table 't1', the corresponding command would be:

  hbase> t.get_counter 'r1', 'c1'


hbase(main):038:0> get_counter 'Table2','1','Student:Name'

ERROR: offset (0) + length (8) exceed the capacity of the array: 7

Here is some help for this command:
Return a counter cell value at specified table/row/column coordinates.
A counter cell should be managed with atomic increment functions on HBase
and the data should be binary encoded (as long value). Example:

  hbase> get_counter 'ns1:t1', 'r1', 'c1'
  hbase> get_counter 't1', 'r1', 'c1'

The same commands also can be run on a table reference. Suppose you had a reference
t to table 't1', the corresponding command would be:

  hbase> t.get_counter 'r1', 'c1'


hbase(main):039:0> scan 'Table2'
ROW                                                   COLUMN+CELL
 1                                                    column=Courses:Course1, timestamp=1529188699534, value=ISL
 1                                                    column=Courses:Course2, timestamp=1529188710790, value=Big Data
 1                                                    column=Student:AGE, timestamp=1529188596361, value=20
 1                                                    column=Student:Name, timestamp=1529188577965, value=Lalitha
 1                                                    column=Student:SEX, timestamp=1529188586521, value=Female
 2                                                    column=Courses:Course1, timestamp=1529188718759, value=Python
 2                                                    column=Courses:Course2, timestamp=1529188726893, value=ASE
 2                                                    column=Student:AGE, timestamp=1529188622320, value=20
 2                                                    column=Student:Name, timestamp=1529188605600, value=Vinay
 2                                                    column=Student:SEX, timestamp=1529188613722, value=Male
 3                                                    column=Courses:Course1, timestamp=1529188736532, value=Big Data
 3                                                    column=Courses:Course2, timestamp=1529188745330, value=ISL
 3                                                    column=Student:AGE, timestamp=1529188446379, value=20
 3                                                    column=Student:Name, timestamp=1529188631565, value=Raj
 3                                                    column=Student:SEX, timestamp=1529188639025, value=Male
3 row(s) in 0.0420 seconds

hbase(main):040:0> describe 'Table2'
Table Table2 is ENABLED
Table2
COLUMN FAMILIES DESCRIPTION
{NAME => 'Courses', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE =
> '65536', REPLICATION_SCOPE => '0'}
{NAME => 'Student', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE =
> '65536', REPLICATION_SCOPE => '0'}
2 row(s) in 0.0250 seconds

hbase(main):041:0> count 'Table2'
3 row(s) in 0.0160 seconds

=> 3
hbase(main):042:0> get 'Table2','2'
COLUMN                                                CELL
 Courses:Course1                                      timestamp=1529188718759, value=Python
 Courses:Course2                                      timestamp=1529188726893, value=ASE
 Student:AGE                                          timestamp=1529188622320, value=20
 Student:Name                                         timestamp=1529188605600, value=Vinay
 Student:SEX                                          timestamp=1529188613722, value=Male
5 row(s) in 0.0140 seconds

hbase(main):043:0>
```

Creating a table with name Table2b and adding a column family CourseInfo and Student which contains Course details and Students enrolled in each course

```
> create 'Table2b','CourseInfo', 'Student'
> put 'Table2b', '1', 'CourseInfo:Title','ISL'
> put 'Table2b', '1', 'CourseInfo:Intro', 'Statistical Learning'
> put 'Table2b', '1', 'CourseInfo:Teacher', 'Deep Medhi'
> put 'Table2b', '2', 'CourseInfo:Title','Python'
> put 'Table2b', '2', 'CourseInfo:Intro', 'Deep Learning'
> put 'Table2b', '2', 'CourseInfo:Teacher', 'Lee'
> put 'Table2b', '3', 'CourseInfo:Title','Big Data'
> put 'Table2b', '3', 'CourseInfo:Intro', 'Hadoop and spark'
> put 'Table2b', '3', 'CourseInfo:Teacher', 'Lee'

> put 'Table2b', '1', 'Student:Stu1','Lalitha'
> put 'Table2b', '1', 'Student:Stu2', 'Vardhini'
> put 'Table2b', '1', 'Student:Stu3', 'Sparshita'
> put 'Table2b', '2', 'Student:Stu1','Vardhini'
> put 'Table2b', '2', 'Student:Stu2', 'Sanjana'
> put 'Table2b', '2', 'Student:Stu3', 'Anjana'
> put 'Table2b', '3', 'Student:Stu1','Vardhini'
> put 'Table2b', '3', 'Student:Stu2', 'Anjana'
> put 'Table2b', '3', 'Student:Stu3', 'Vardhini’

scan 'Table2b'
count 'Table2b'
get 'Table2b','2'
```

```
hbase(main):043:0> create 'Table2b','CourseInfo','Student'
0 row(s) in 2.2660 seconds

=> Hbase::Table - Table2b
hbase(main):044:0> put 'Table2b', '1', 'CourseInfo:Title','ISL'
0 row(s) in 0.0190 seconds

hbase(main):045:0> put 'Table2b', '1', 'CourseInfo:Intro', 'Statistical Learning'
0 row(s) in 0.0150 seconds

hbase(main):046:0> put 'Table2b', '1', 'CourseInfo:Teacher', 'Deep Medhi'
0 row(s) in 0.0130 seconds

hbase(main):047:0> put 'Table2b', '2', 'CourseInfo:Title','Python'
0 row(s) in 0.0140 seconds

hbase(main):048:0> put 'Table2b', '2', 'CourseInfo:Intro', 'Deep Learning'
0 row(s) in 0.0140 seconds

hbase(main):049:0> put 'Table2b', '2', 'CourseInfo:Teacher', 'Lee'
0 row(s) in 0.0200 seconds

hbase(main):050:0> put 'Table2b', '3', 'CourseInfo:Title','Big Data'
0 row(s) in 0.0140 seconds

hbase(main):051:0> put 'Table2b', '3', 'CourseInfo:Intro', 'Hadoop and spark'
0 row(s) in 0.0140 seconds

hbase(main):052:0> put 'Table2b', '3', 'CourseInfo:Teacher', 'Lee'
0 row(s) in 0.0390 seconds

hbase(main):053:0> put 'Table2b', '1', 'Student:Stu1','Lalitha'
0 row(s) in 0.0140 seconds

hbase(main):054:0> put 'Table2b', '1', 'Student:Stu2', 'Vardhini'
0 row(s) in 0.0130 seconds

hbase(main):055:0> put 'Table2b', '1', 'Student:Stu3', 'Sparshita'
0 row(s) in 0.0170 seconds

hbase(main):056:0> put 'Table2b', '2', 'Student:Stu1','Vardhini'
0 row(s) in 0.0130 seconds

hbase(main):057:0> put 'Table2b', '2', 'Student:Stu2', 'Sanjana'
0 row(s) in 0.0140 seconds

hbase(main):058:0> put 'Table2b', '2', 'Student:Stu3', 'Anjana'
0 row(s) in 0.0130 seconds

hbase(main):059:0> put 'Table2b', '3', 'Student:Stu1','Vardhini'
0 row(s) in 0.0140 seconds

hbase(main):060:0> put 'Table2b', '3', 'Student:Stu2', 'Anjana'
0 row(s) in 0.0130 seconds

hbase(main):061:0> put 'Table2b', '3', 'Student:Stu3', 'Vardhini'
0 row(s) in 0.0120 seconds

hbase(main):062:0> scan 'Table2b'
ROW                                                 COLUMN+CELL
 1                                                  column=CourseInfo:Intro, timestamp=1529189469867, value=Statistical Learning
 1                                                  column=CourseInfo:Teacher, timestamp=1529189478052, value=Deep Medhi
 1                                                  column=CourseInfo:Title, timestamp=1529189460879, value=ISL
 1                                                  column=Student:Stu1, timestamp=1529189543580, value=Lalitha
 1                                                  column=Student:Stu2, timestamp=1529189550746, value=Vardhini
 1                                                  column=Student:Stu3, timestamp=1529189558569, value=Sparshita
 2                                                  column=CourseInfo:Intro, timestamp=1529189499026, value=Deep Learning
 2                                                  column=CourseInfo:Teacher, timestamp=1529189506746, value=Lee
 2                                                  column=CourseInfo:Title, timestamp=1529189486021, value=Python
 2                                                  column=Student:Stu1, timestamp=1529189565718, value=Vardhini
 2                                                  column=Student:Stu2, timestamp=1529189573088, value=Sanjana
 2                                                  column=Student:Stu3, timestamp=1529189581732, value=Anjana
 3                                                  column=CourseInfo:Intro, timestamp=1529189524753, value=Hadoop and spark
```

```
hbase(main):050:0> put 'Table2b', '3', 'CourseInfo:Title','Big Data'
0 row(s) in 0.0140 seconds

hbase(main):051:0> put 'Table2b', '3', 'CourseInfo:Intro', 'Hadoop and spark'
0 row(s) in 0.0140 seconds

hbase(main):052:0> put 'Table2b', '3', 'CourseInfo:Teacher', 'Lee'
0 row(s) in 0.0390 seconds

hbase(main):053:0> put 'Table2b', '1', 'Student:Stu1','Lalitha'
0 row(s) in 0.0140 seconds

hbase(main):054:0> put 'Table2b', '1', 'Student:Stu2', 'Vardhini'
0 row(s) in 0.0130 seconds

hbase(main):055:0> put 'Table2b', '1', 'Student:Stu3', 'Sparshita'
0 row(s) in 0.0170 seconds

hbase(main):056:0> put 'Table2b', '2', 'Student:Stu1','Vardhini'
0 row(s) in 0.0130 seconds

hbase(main):057:0> put 'Table2b', '2', 'Student:Stu2', 'Sanjana'
0 row(s) in 0.0140 seconds

hbase(main):058:0> put 'Table2b', '2', 'Student:Stu3', 'Anjana'
0 row(s) in 0.0130 seconds

hbase(main):059:0> put 'Table2b', '3', 'Student:Stu1','Vardhini'
0 row(s) in 0.0140 seconds

hbase(main):060:0> put 'Table2b', '3', 'Student:Stu2', 'Anjana'
0 row(s) in 0.0130 seconds

hbase(main):061:0> put 'Table2b', '3', 'Student:Stu3', 'Vardhini'
0 row(s) in 0.0120 seconds

hbase(main):062:0> scan 'Table2b'
ROW                                                 COLUMN+CELL
 1                                                  column=CourseInfo:Intro, timestamp=1529189469867, value=Statistical Learning
 1                                                  column=CourseInfo:Teacher, timestamp=1529189478052, value=Deep Medhi
 1                                                  column=CourseInfo:Title, timestamp=1529189460879, value=ISL
 1                                                  column=Student:Stu1, timestamp=1529189543580, value=Lalitha
 1                                                  column=Student:Stu2, timestamp=1529189550746, value=Vardhini
 1                                                  column=Student:Stu3, timestamp=1529189558569, value=Sparshita
 2                                                  column=CourseInfo:Intro, timestamp=1529189499026, value=Deep Learning
 2                                                  column=CourseInfo:Teacher, timestamp=1529189506746, value=Lee
 2                                                  column=CourseInfo:Title, timestamp=1529189486021, value=Python
 2                                                  column=Student:Stu1, timestamp=1529189565718, value=Vardhini
 2                                                  column=Student:Stu2, timestamp=1529189573088, value=Sanjana
 2                                                  column=Student:Stu3, timestamp=1529189581732, value=Anjana
 3                                                  column=CourseInfo:Intro, timestamp=1529189524753, value=Hadoop and spark
 3                                                  column=CourseInfo:Teacher, timestamp=1529189532954, value=Lee
 3                                                  column=CourseInfo:Title, timestamp=1529189514700, value=Big Data
 3                                                  column=Student:Stu1, timestamp=1529189589384, value=Vardhini
 3                                                  column=Student:Stu2, timestamp=1529189596133, value=Anjana
 3                                                  column=Student:Stu3, timestamp=1529189605011, value=Vardhini
3 row(s) in 0.0410 seconds

hbase(main):063:0> count 'Table2b'
3 row(s) in 0.0170 seconds

=> 3
hbase(main):064:0> get 'Table2b','2'
COLUMN                                              CELL
 CourseInfo:Intro                                   timestamp=1529189499026, value=Deep Learning
 CourseInfo:Teacher                                 timestamp=1529189506746, value=Lee
 CourseInfo:Title                                   timestamp=1529189486021, value=Python
 Student:Stu1                                       timestamp=1529189565718, value=Vardhini
 Student:Stu2                                       timestamp=1529189573088, value=Sanjana
 Student:Stu3                                       timestamp=1529189581732, value=Anjana
6 row(s) in 0.0140 seconds

hbase(main):065:0>
```

Usecase 3:

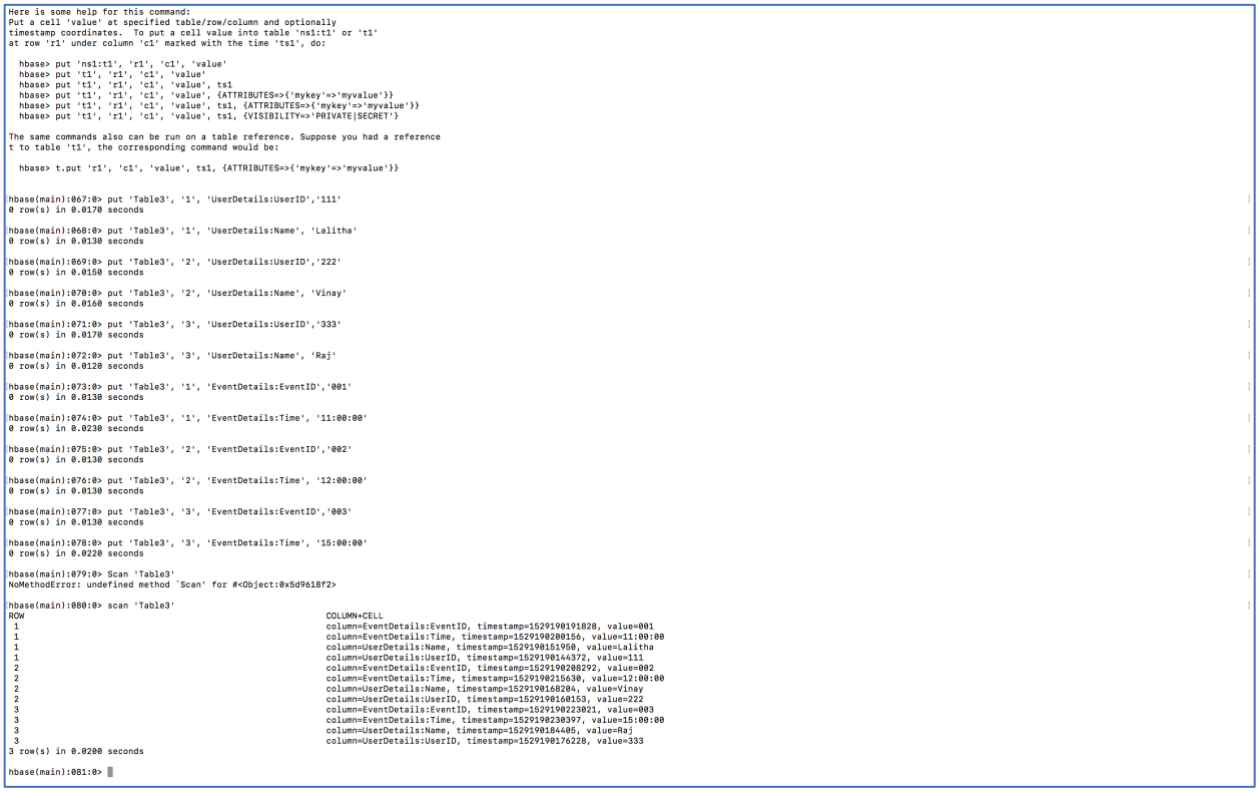
Creating a table with name Table3 and adding a column family UserDetails and EventDetails

create 'Table3','UserDetails', 'EventDetails'

> put 'Table3', '1', 'UserDetails:UserID','111'
> put 'Table3', '1', 'UserDetails:Name', 'Lalitha'
> put 'Table3', '2', 'UserDetails:UserID','222'
> put 'Table3', '2', 'UserDetails:Name', 'Vinay'
> put 'Table3', '3', 'UserDetails:UserID','333'
> put 'Table3', '3', 'UserDetails:Name', 'Raj'

> put 'Table3', '1', 'EventDetails:EventID','001'
> put 'Table3', '1', 'EventDetails:Time', '11:00:00'
> put 'Table3', '2', 'EventDetails:EventID','002'
> put 'Table3', '2', 'EventDetails:Time', '12:00:00'
> put 'Table3', '3', 'EventDetails:EventID','003'
> put 'Table3', '3', 'EventDetails:Time', '15:00:00'

Scan 'Table3'



```
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 'ns1:t1' or 't1'
at row 'r1' under column 'c1' marked with the time 'ts1', do:

  hbase> put 'ns1:t1', 'r1', 'c1', 'value'
  hbase> put 't1', 'r1', 'c1', 'value'
  hbase> put 't1', 'r1', 'c1', 'value', ts1
  hbase> put 't1', 'r1', 'c1', 'value', {ATTRIBUTES=>{'mykey'=>'myvalue'}}
  hbase> put 't1', 'r1', 'c1', 'value', ts1, {ATTRIBUTES=>{'mykey'=>'myvalue'}}
  hbase> put 't1', 'r1', 'c1', 'value', ts1, {VISIBILITY=>'PRIVATE|SECRET'}

The same commands also can be run on a table reference. Suppose you had a reference
t to table 't1', the corresponding command would be:

  hbase> t.put 'r1', 'c1', 'value', ts1, {ATTRIBUTES=>{'mykey'=>'myvalue'}}


hbase(main):067:0> put 'Table3', '1', 'UserDetails:UserID','111'
0 row(s) in 0.0170 seconds

hbase(main):068:0> put 'Table3', '1', 'UserDetails:Name', 'Lalitha'
0 row(s) in 0.0130 seconds

hbase(main):069:0> put 'Table3', '2', 'UserDetails:UserID','222'
0 row(s) in 0.0150 seconds

hbase(main):070:0> put 'Table3', '2', 'UserDetails:Name', 'Vinay'
0 row(s) in 0.0160 seconds

hbase(main):071:0> put 'Table3', '3', 'UserDetails:UserID','333'
0 row(s) in 0.0170 seconds

hbase(main):072:0> put 'Table3', '3', 'UserDetails:Name', 'Raj'
0 row(s) in 0.0120 seconds

hbase(main):073:0> put 'Table3', '1', 'EventDetails:EventID','001'
0 row(s) in 0.0130 seconds

hbase(main):074:0> put 'Table3', '1', 'EventDetails:Time', '11:00:00'
0 row(s) in 0.0230 seconds

hbase(main):075:0> put 'Table3', '2', 'EventDetails:EventID','002'
0 row(s) in 0.0130 seconds

hbase(main):076:0> put 'Table3', '2', 'EventDetails:Time', '12:00:00'
0 row(s) in 0.0130 seconds

hbase(main):077:0> put 'Table3', '3', 'EventDetails:EventID','003'
0 row(s) in 0.0130 seconds

hbase(main):078:0> put 'Table3', '3', 'EventDetails:Time', '15:00:00'
0 row(s) in 0.0220 seconds

hbase(main):079:0> Scan 'Table3'
NoMethodError: undefined method `Scan' for #<Object:0x5d961bf2>

hbase(main):080:0> scan 'Table3'
ROW                                                   COLUMN+CELL
 1                                                    column=EventDetails:EventID, timestamp=1529190191828, value=001
 1                                                    column=EventDetails:Time, timestamp=1529190200156, value=11:00:00
 1                                                    column=UserDetails:Name, timestamp=1529190151950, value=Lalitha
 1                                                    column=UserDetails:UserID, timestamp=1529190144372, value=111
 2                                                    column=EventDetails:EventID, timestamp=1529190208292, value=002
 2                                                    column=EventDetails:Time, timestamp=1529190215630, value=12:00:00
 2                                                    column=UserDetails:Name, timestamp=1529190168204, value=Vinay
 2                                                    column=UserDetails:UserID, timestamp=1529190160153, value=222
 3                                                    column=EventDetails:EventID, timestamp=1529190223021, value=003
 3                                                    column=EventDetails:Time, timestamp=1529190230397, value=15:00:00
 3                                                    column=UserDetails:Name, timestamp=1529190184405, value=Raj
 3                                                    column=UserDetails:UserID, timestamp=1529190176228, value=333
3 row(s) in 0.0200 seconds

hbase(main):081:0>
```

Use Case 4:
Creating a table with name Table3 and adding a column family User and Friends

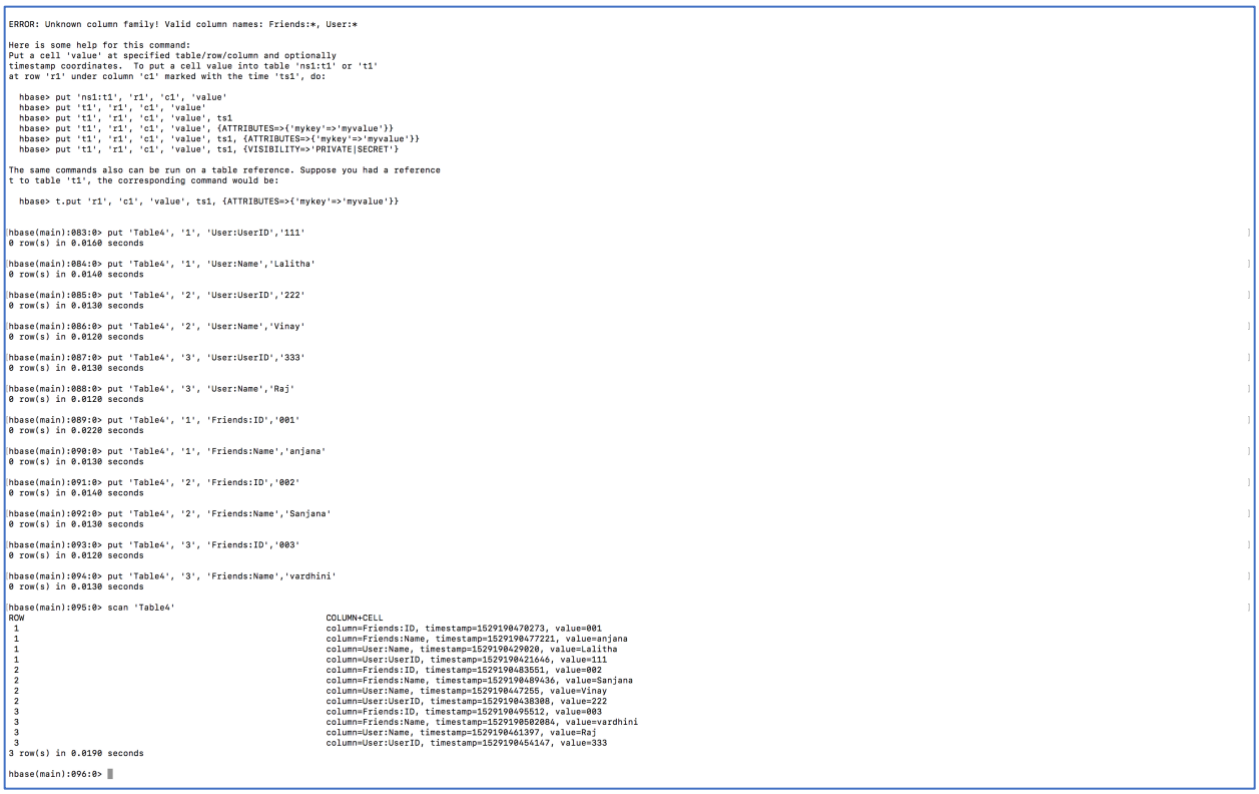
create 'Table4','User', 'Friends'

> put 'Table4', '1', 'User:UserID','111'
> put 'Table4', '1', 'User:Name','Lalitha'
> put 'Table4', '2', 'User:UserID','222'
> put 'Table4', '2', 'User:Name','Vinay'
> put 'Table4', '3', 'User:UserID','333'
> put 'Table4', '3', 'User:Name','Raj'

> put 'Table4', '1', 'Friends:ID','001'
> put 'Table4', '1', 'Friends:Name','anjana'
> put 'Table4', '2', 'Friends:ID','002'
> put 'Table4', '2', 'Friends:Name','Sanjana'
> put 'Table4', '3', 'Friends:ID','003'
> put 'Table4', '3', 'Friends:Name','vardhini'

Scan 'Table4'

```
hbase(main):081:0> create 'Table4','User', 'Friends'
0 row(s) in 1.2490 seconds

=> Hbase::Table - Table4
hbase(main):082:0> put 'Table4', '1', 'UserDetails:UserID','111'

ERROR: Unknown column family! Valid column names: Friends:*, User:*

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 'ns1:t1' or 't1'
at row 'r1' under column 'c1' marked with the time 'ts1', do:

  hbase> put 'ns1:t1', 'r1', 'c1', 'value'
  hbase> put 't1', 'r1', 'c1', 'value'
  hbase> put 't1', 'r1', 'c1', 'value', ts1
  hbase> put 't1', 'r1', 'c1', 'value', {ATTRIBUTES=>{'mykey'=>'myvalue'}}
  hbase> put 't1', 'r1', 'c1', 'value', ts1, {ATTRIBUTES=>{'mykey'=>'myvalue'}}
  hbase> put 't1', 'r1', 'c1', 'value', ts1, {VISIBILITY=>'PRIVATE|SECRET'}

The same commands also can be run on a table reference. Suppose you had a reference
t to table 't1', the corresponding command would be:

  hbase> t.put 'r1', 'c1', 'value', ts1, {ATTRIBUTES=>{'mykey'=>'myvalue'}}


hbase(main):083:0> put 'Table4', '1', 'User:UserID','111'
0 row(s) in 0.0160 seconds

hbase(main):084:0> put 'Table4', '1', 'User:Name','Lalitha'
0 row(s) in 0.0140 seconds

hbase(main):085:0> put 'Table4', '2', 'User:UserID','222'
0 row(s) in 0.0130 seconds

hbase(main):086:0> put 'Table4', '2', 'User:Name','Vinay'
0 row(s) in 0.0120 seconds

hbase(main):087:0> put 'Table4', '3', 'User:UserID','333'
0 row(s) in 0.0130 seconds

hbase(main):088:0> put 'Table4', '3', 'User:Name','Raj'
0 row(s) in 0.0120 seconds

hbase(main):089:0> put 'Table4', '1', 'Friends:ID','001'
0 row(s) in 0.0220 seconds

hbase(main):090:0> put 'Table4', '1', 'Friends:Name','anjana'
0 row(s) in 0.0130 seconds

hbase(main):091:0> put 'Table4', '2', 'Friends:ID','002'
0 row(s) in 0.0140 seconds

hbase(main):092:0> put 'Table4', '2', 'Friends:Name','Sanjana'
0 row(s) in 0.0130 seconds

hbase(main):093:0> put 'Table4', '3', 'Friends:ID','003'
0 row(s) in 0.0120 seconds

hbase(main):094:0> put 'Table4', '3', 'Friends:Name','vardhini'
0 row(s) in 0.0130 seconds

hbase(main):095:0> scan 'Table4'
ROW                                                   COLUMN+CELL
 1                                                    column=Friends:ID, timestamp=1529190470273, value=001
 1                                                    column=Friends:Name, timestamp=1529190477221, value=anjana
 1                                                    column=User:Name, timestamp=1529190429020, value=Lalitha
 1                                                    column=User:UserID, timestamp=1529190421646, value=111
 2                                                    column=Friends:ID, timestamp=1529190483651, value=002
 2                                                    column=Friends:Name, timestamp=1529190489436, value=Sanjana
 2                                                    column=User:Name, timestamp=1529190447255, value=Vinay
 2                                                    column=User:UserID, timestamp=1529190438308, value=222
 3                                                    column=Friends:ID, timestamp=1529190495512, value=003
```



```
ERROR: Unknown column family! Valid column names: Friends:*, User:*

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 'ns1:t1' or 't1'
at row 'r1' under column 'c1' marked with the time 'ts1', do:

  hbase> put 'ns1:t1', 'r1', 'c1', 'value'
  hbase> put 't1', 'r1', 'c1', 'value'
  hbase> put 't1', 'r1', 'c1', 'value', ts1
  hbase> put 't1', 'r1', 'c1', 'value', {ATTRIBUTES=>{'mykey'=>'myvalue'}}
  hbase> put 't1', 'r1', 'c1', 'value', ts1, {ATTRIBUTES=>{'mykey'=>'myvalue'}}
  hbase> put 't1', 'r1', 'c1', 'value', ts1, {VISIBILITY=>'PRIVATE|SECRET'}

The same commands also can be run on a table reference. Suppose you had a reference
t to table 't1', the corresponding command would be:

  hbase> t.put 'r1', 'c1', 'value', ts1, {ATTRIBUTES=>{'mykey'=>'myvalue'}}


hbase(main):083:0> put 'Table4', '1', 'User:UserID','111'
0 row(s) in 0.0160 seconds

hbase(main):084:0> put 'Table4', '1', 'User:Name','Lalitha'
0 row(s) in 0.0140 seconds

hbase(main):085:0> put 'Table4', '2', 'User:UserID','222'
0 row(s) in 0.0130 seconds

hbase(main):086:0> put 'Table4', '2', 'User:Name','Vinay'
0 row(s) in 0.0120 seconds

hbase(main):087:0> put 'Table4', '3', 'User:UserID','333'
0 row(s) in 0.0130 seconds

hbase(main):088:0> put 'Table4', '3', 'User:Name','Raj'
0 row(s) in 0.0120 seconds

hbase(main):089:0> put 'Table4', '1', 'Friends:ID','001'
0 row(s) in 0.0220 seconds

hbase(main):090:0> put 'Table4', '1', 'Friends:Name','anjana'
0 row(s) in 0.0130 seconds

hbase(main):091:0> put 'Table4', '2', 'Friends:ID','002'
0 row(s) in 0.0140 seconds

hbase(main):092:0> put 'Table4', '2', 'Friends:Name','Sanjana'
0 row(s) in 0.0130 seconds

hbase(main):093:0> put 'Table4', '3', 'Friends:ID','003'
0 row(s) in 0.0120 seconds

hbase(main):094:0> put 'Table4', '3', 'Friends:Name','vardhini'
0 row(s) in 0.0130 seconds

hbase(main):095:0> scan 'Table4'
ROW                                                   COLUMN+CELL
 1                                                    column=Friends:ID, timestamp=1529190470273, value=001
 1                                                    column=Friends:Name, timestamp=1529190477221, value=anjana
 1                                                    column=User:Name, timestamp=1529190429020, value=Lalitha
 1                                                    column=User:UserID, timestamp=1529190421646, value=111
 2                                                    column=Friends:ID, timestamp=1529190483551, value=002
 2                                                    column=Friends:Name, timestamp=1529190489436, value=Sanjana
 2                                                    column=User:Name, timestamp=1529190447255, value=Vinay
 2                                                    column=User:UserID, timestamp=1529190438308, value=222
 3                                                    column=Friends:ID, timestamp=1529190495512, value=003
 3                                                    column=Friends:Name, timestamp=1529190502084, value=vardhini
 3                                                    column=User:Name, timestamp=1529190461397, value=Raj
 3                                                    column=User:UserID, timestamp=1529190454147, value=333
3 row(s) in 0.0190 seconds

hbase(main):096:0>
```

USECASE-5

Creating a table with name Table3 and adding a column family http and User

create 'Table5','http','User'

> put 'Table5', '1', 'http:IP','11.111.11.1'
> put 'Table5', '1', 'http:Domain', 'anjana'
> put 'Table5', '2', 'http:IP','22.222.22.2'
> put 'Table5', '2', 'http:Domain', 'Sanjana'
> put 'Table5', '3', 'http:IP','33.333.33.3'
> put 'Table5', '3', 'http:Domain', 'vardhini'

> put 'Table5', '1', 'User:UserID','111'
> put 'Table5', '1', 'User:Name', 'Lalitha'
> put 'Table5', '2', 'User:UserID','222'
> put 'Table5', '2', 'User:Name', 'Vinay'
> put 'Table5', '3', 'User:UserID','333'
> put 'Table4', '3', 'User:Name', 'Raj'

Scan ‘Table5’

```
hbase(main):096:0> create 'Table5','http','User'
0 row(s) in 2.2500 seconds

=> Hbase::Table - Table5
hbase(main):097:0> put 'Table5', '1', 'http:IP','11.111.11.1'
0 row(s) in 0.0180 seconds

hbase(main):098:0> put 'Table5', '1', 'http:Domain', 'anjana'
0 row(s) in 0.0140 seconds

hbase(main):099:0> put 'Table5', '2', 'http:IP','22.222.22.2'
0 row(s) in 0.0230 seconds

hbase(main):100:0> put 'Table5', '2', 'http:Domain', 'Sanjana'
0 row(s) in 0.0120 seconds

hbase(main):101:0>  put 'Table5', '3', 'http:IP','33.333.33.3'
0 row(s) in 0.0120 seconds

hbase(main):102:0> put 'Table5', '3', 'http:Domain', 'vardhini'
0 row(s) in 0.0130 seconds

hbase(main):103:0> put 'Table5', '1', 'User:UserID','111'
0 row(s) in 0.0130 seconds

hbase(main):104:0> put 'Table5', '1', 'UserDetails:Name', 'Lalitha'

ERROR: Unknown column family! Valid column names: User:*, http:*

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 'ns1:t1' or 't1'
at row 'r1' under column 'c1' marked with the time 'ts1', do:

  hbase> put 'ns1:t1', 'r1', 'c1', 'value'
  hbase> put 't1', 'r1', 'c1', 'value'
  hbase> put 't1', 'r1', 'c1', 'value', ts1
  hbase> put 't1', 'r1', 'c1', 'value', {ATTRIBUTES=>{'mykey'=>'myvalue'}}
  hbase> put 't1', 'r1', 'c1', 'value', ts1, {ATTRIBUTES=>{'mykey'=>'myvalue'}}
  hbase> put 't1', 'r1', 'c1', 'value', ts1, {VISIBILITY=>'PRIVATE|SECRET'}

The same commands also can be run on a table reference. Suppose you had a reference
t to table 't1', the corresponding command would be:

  hbase> t.put 'r1', 'c1', 'value', ts1, {ATTRIBUTES=>{'mykey'=>'myvalue'}}


hbase(main):105:0> put 'Table5', '1', 'User:Name', 'Lalitha'
0 row(s) in 0.0170 seconds

hbase(main):106:0> put 'Table5', '2', 'User:UserID','222'
0 row(s) in 0.0130 seconds

hbase(main):107:0> put 'Table5', '2', 'User:Name', 'Vinay'
0 row(s) in 0.0130 seconds

hbase(main):108:0>  put 'Table5', '3', 'User:UserID','333'
0 row(s) in 0.0120 seconds

hbase(main):109:0> put 'Table4', '3', 'User:Name', 'Raj'
0 row(s) in 0.0120 seconds

hbase(main):110:0> scan 'Table5'
ROW                                                 COLUMN+CELL
 1                                                  column=User:Name, timestamp=1529190817533, value=Lalitha
 1                                                  column=User:UserID, timestamp=1529190769515, value=111
 1                                                  column=http:Domain, timestamp=1529190733465, value=anjana
 1                                                  column=http:IP, timestamp=1529190725162, value=11.111.11.1
 2                                                  column=User:Name, timestamp=1529190830316, value=Vinay
 2                                                  column=User:UserID, timestamp=1529190823854, value=222
 2                                                  column=http:Domain, timestamp=1529190746472, value=Sanjana
 2                                                  column=http:IP, timestamp=1529190739956, value=22.222.22.2
 3                                                  column=User:UserID, timestamp=1529190837389, value=333
```

```
hbase(main):097:0> put 'Table5', '1', 'http:IP','11.111.11.1'
0 row(s) in 0.0180 seconds

hbase(main):098:0> put 'Table5', '1', 'http:Domain', 'anjana'
0 row(s) in 0.0140 seconds

hbase(main):099:0> put 'Table5', '2', 'http:IP','22.222.22.2'
0 row(s) in 0.0230 seconds

hbase(main):100:0> put 'Table5', '2', 'http:Domain', 'Sanjana'
0 row(s) in 0.0120 seconds

hbase(main):101:0>  put 'Table5', '3', 'http:IP','33.333.33.3'
0 row(s) in 0.0120 seconds

hbase(main):102:0> put 'Table5', '3', 'http:Domain', 'vardhini'
0 row(s) in 0.0130 seconds

hbase(main):103:0> put 'Table5', '1', 'User:UserID','111'
0 row(s) in 0.0130 seconds

hbase(main):104:0> put 'Table5', '1', 'UserDetails:Name', 'Lalitha'

ERROR: Unknown column family! Valid column names: User:*, http:*

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 'ns1:t1' or 't1'
at row 'r1' under column 'c1' marked with the time 'ts1', do:

  hbase> put 'ns1:t1', 'r1', 'c1', 'value'
  hbase> put 't1', 'r1', 'c1', 'value'
  hbase> put 't1', 'r1', 'c1', 'value', ts1
  hbase> put 't1', 'r1', 'c1', 'value', {ATTRIBUTES=>{'mykey'=>'myvalue'}}
  hbase> put 't1', 'r1', 'c1', 'value', ts1, {ATTRIBUTES=>{'mykey'=>'myvalue'}}
  hbase> put 't1', 'r1', 'c1', 'value', ts1, {VISIBILITY=>'PRIVATE|SECRET'}

The same commands also can be run on a table reference. Suppose you had a reference
t to table 't1', the corresponding command would be:

  hbase> t.put 'r1', 'c1', 'value', ts1, {ATTRIBUTES=>{'mykey'=>'myvalue'}}


hbase(main):105:0> put 'Table5', '1', 'User:Name', 'Lalitha'
0 row(s) in 0.0170 seconds

hbase(main):106:0> put 'Table5', '2', 'User:UserID','222'
0 row(s) in 0.0130 seconds

hbase(main):107:0> put 'Table5', '2', 'User:Name', 'Vinay'
0 row(s) in 0.0130 seconds

hbase(main):108:0>  put 'Table5', '3', 'User:UserID','333'
0 row(s) in 0.0120 seconds

hbase(main):109:0> put 'Table4', '3', 'User:Name', 'Raj'
0 row(s) in 0.0120 seconds

hbase(main):110:0> scan 'Table5'
ROW                                                 COLUMN+CELL
 1                                                  column=User:Name, timestamp=1529190817533, value=Lalitha
 1                                                  column=User:UserID, timestamp=1529190769515, value=111
 1                                                  column=http:Domain, timestamp=1529190733465, value=anjana
 1                                                  column=http:IP, timestamp=1529190725162, value=11.111.11.1
 2                                                  column=User:Name, timestamp=1529190830316, value=Vinay
 2                                                  column=User:UserID, timestamp=1529190823854, value=222
 2                                                  column=http:Domain, timestamp=1529190746472, value=Sanjana
 2                                                  column=http:IP, timestamp=1529190739956, value=22.222.22.2
 3                                                  column=User:UserID, timestamp=1529190837389, value=333
 3                                                  column=http:Domain, timestamp=1529190761161, value=vardhini
 3                                                  column=http:IP, timestamp=1529190754121, value=33.333.33.3
3 row(s) in 0.0190 seconds

hbase(main):111:0>
```

Section 2

General Hbase Commands

Table help command

```
hbase(main):124:0> table_help
Help for table-reference commands.

You can either create a table via 'create' and then manipulate the table via commands like 'put', 'get',
 etc.
See the standard help information for how to use each of these commands.

However, as of 0.96, you can also get a reference to a table, on which you can invoke commands.
For instance, you can get create a table and keep around a reference to it via:

   hbase> t = create 't', 'cf'

Or, if you have already created the table, you can get a reference to it:

   hbase> t = get_table 't'

You can do things like call 'put' on the table:

  hbase> t.put 'r', 'cf:q', 'v'

which puts a row 'r' with column family 'cf', qualifier 'q' and value 'v' into table t.

To read the data out, you can scan the table:

  hbase> t.scan

which will read all the rows in table 't'.

Essentially, any command that takes a table name can also be done via table reference.
Other commands include things like: get, delete, deleteall,
get_all_columns, get_counter, count, incr. These functions, along with
the standard JRuby object methods are also available via tab completion.

For more information on how to use each of these commands, you can also just type:

   hbase> t.help 'scan'

which will output more information on how to use that command.

You can also do general admin actions directly on a table; things like enable, disable,
flush and drop just by typing:

   hbase> t.enable
   hbase> t.flush
   hbase> t.disable
   hbase> t.drop

Note that after dropping a table, your reference to it becomes useless and further usage
is undefined (and not recommended).

hbase(main):125:0>
```

Status Command

```
hbase(main):123:0> status
1 active master, 0 backup masters, 1 servers, 1 dead, 8.0000 average load
```

Version Command

```
hbase(main):125:0> version
1.2.6.1, rUnknown, Sun Jun  3 23:19:26 CDT 2018

hbase(main):126:0>
```

Whoami Command

```
hbase(main):126:0> whoami
lalithajetty (auth:SIMPLE)
    groups: staff, com.apple.sharepoint.group.1, everyone, localaccounts, _apps
erverusr, admin, _appserveradm, _lpadmin, _appstore, _lpoperator, _developer, _
analyticsusers, com.apple.access_ftp, com.apple.access_screensharing, com.apple
.access_ssh

hbase(main):127:0>
```

Table Management Commands

List

```
hbase(main):111:0> list
TABLE
Table01
Table1
Table2
Table2b
Table3
Table4
Table5
table
table1
9 row(s) in 0.0350 seconds
```

Disable

```
hbase(main):111:0> list
TABLE
Table01
Table1
Table2
Table2b
Table3
Table4
Table5
table
table1
9 row(s) in 0.0350 seconds

=> ["Table01", "Table1", "Table2", "Table2b", "Table3", "Table4", "Table5", "table", "table1"]
hbase(main):112:0> disable 'table'
0 row(s) in 2.2740 seconds
```

Drop

```
hbase(main):111:0> list
TABLE
Table01
Table1
Table2
Table2b
Table3
Table4
Table5
table
table1
9 row(s) in 0.0350 seconds

=> ["Table01", "Table1", "Table2", "Table2b", "Table3", "Table4", "Table5", "table", "table1"]
hbase(main):112:0> disable 'table'
0 row(s) in 2.2740 seconds

hbase(main):113:0> list
TABLE
Table01
Table1
Table2
Table2b
Table3
Table4
Table5
table
table1
9 row(s) in 0.0110 seconds

=> ["Table01", "Table1", "Table2", "Table2b", "Table3", "Table4", "Table5", "table", "table1"]
hbase(main):114:0> drop 'table'
0 row(s) in 1.2620 seconds

hbase(main):115:0> list
TABLE
Table01
Table1
Table2
Table2b
Table3
Table4
Table5
table1
8 row(s) in 0.0110 seconds

=> ["Table01", "Table1", "Table2", "Table2b", "Table3", "Table4", "Table5", "table1"]
hbase(main):116:0>
```