Exercise 1: Perform CRUD operations with HBase shell

The goal of this exercise is to create a table ‘customer’ with the data for Column Families address, order and columns city, state for as shown below using HBase shell commands.

Table: customer

Row-key userid	address		order	
	city	state	date	number
Ismith	nashville	tn	01/01/2015	12345
Biones	miami	fl	02/02/2015	56565

Use HBase shell

```
_$ hbase shell_
```

Once connected to the cluster and having started the HBase shell, explore the commands that you can perform: like ‘help’, ‘help “put”’ etc…

Note: if you are using a cluster, change user01 to your user, a text file with all of the commands is provided for your convenience in the LAB FILES directory.

Create a table in your home directory

```
create 'customer', {NAME=>'addr'}, {NAME=>'order'}
```

Use ‘describe’ to get the description of the table.

```
describe 'customer'
```

Execute the following statements to insert and get the records
Put some data into the table

```
put 'customer', 'jsmith', 'addr:city', 'nashville'
```

Use ‘get’ to retrieve the data for ‘jsmith’

```
get 'customer', 'jsmith'
```

Put more data into the table

```
put   'customer',  'jsmith',  'addr:state', 'TN'
put   'customer',  'jsmith',  'order:numb', '1234'
put   'customer',  'jsmith',  'order:date', '10-18-2014'
```

Use get to retrieve the data for ‘jsmith’

```
get 'customer', 'jsmith'
```

Note that this gets all the data for the row. How can we limit this to only one column family ?

```
get 'customer', 'jsmith', {COLUMNS=>['addr']}
```

How can we limit this to a specific column?

```
get 'customer', 'jsmith', {COLUMNS=>['order:numb']}
```

Alter table to store more versions in the order column family

```
alter 'customer' , NAME => 'order', VERSIONS => 5
```

Use ‘describe’ to get the description of the table.

```
describe 'customer'
```

put more order numbers

```
put   'customer',  'jsmith',  'order:numb', '1235'
put   'customer',  'jsmith',  'order:numb', '1236'
put   'customer',  'jsmith',  'order:numb', '1237'
put   'customer',  'jsmith',  'order:numb', '1238'
```

Get order number column cells

```
get 'customer', 'jsmith', {COLUMNS=>['order:numb']}
```

Note that you are getting the data for only one version per cell. How can you get more versions?

```
get 'customer', 'jsmith', {COLUMNS=>['order:numb'], VERSIONS => 5}
```

put more data for different rowkey userids

```
put   'customer',  'njones',  'addr:city', 'miami'
put   'customer',  'njones',  'addr:state', 'FL'
put   'customer',  'njones',  'order:numb', '5555'
put   'customer',  'tsimmons',  'addr:city', 'dallas'
put   'customer',  'tsimmons',  'addr:state', 'TX'
put   'customer',  'jsmith',  'addr:city', 'denver'
put   'customer',  'jsmith',  'addr:state', 'CO'
put   'customer',  'jsmith',  'order:numb', '6666'
put   'customer',  'njones',  'addr:state', 'TX'
put   'customer',  'amiller', 'addr:state', 'TX'
```

Use ‘scan’ to retrieve rows of data for the table

retrieve all rows, all columns

```
scan 'customer'
```

retrieve all rows, addr column family

```
scan 'customer', {COLUMNS=>['addr']}
```

retrieve all rows for order number column, 5 versions

```
scan 'customer', {COLUMNS=>['order:numb'], VERSIONS => 5}
```

retrieve rows with rowkey starting with 'njo', addr column family

```
scan 'customer', {STARTROW => 'njo', COLUMNS=>['addr'] }
```

retrieve rows with rowkey starting with 'j', stop before 't'

```
scan 'customer', { STARTROW => 'j', STOPROW => 't'}
```

retrieve rows with rowkey starting with 'a'

```
scan 'customer', { STARTROW => 'a'}
```

retrieve rows with rowkey starting with 'a'

```
scan 'customer', { STARTROW => 't'}>
```

Use ‘count’ to retrieve the number of rows in the table.

```
count 'customer'
```

Delete data from the table.

```
get 'customer', 'njones'
```

delete a column

```
delete 'customer', 'njones', 'addr:city'
```

delete a column family (This is not working). Instead you can give following command

alter <tablename>, {NAME => ‘Col_Family_Name’, METHOD => ‘delete’}

```
delete   'customer',  'jsmith',   'addr:'
get   'customer',  'jsmith'
```

delete a row

```
deleteall 'customer', 'jsmith'
```