APACHE HBASE ON DOCKER

BIG DATA ANALYTICS

FINAL PROJECT

NAME: SYED ASAD RIZVI

ERP ID: 25365

PROGRAM: MS – DATA SCIENCE

Dataset Used: E-commerce Behavior Data – Multi Category Store

Implementation Steps:

HBase Container Setup:

1) C:\Users\HR Computers>f:
   F:\>cd big.data

- *For container setup, first create a folder 'big.data' in F: drive and navigate to it in the command prompt.*

2) F:\big.data>git clone https://github.com/big-data-europe/docker-hbase.git

- *Clone the docker-hbase folder by from the mentioned link by executing the 'git clone' command.*

```
Cloning into 'docker-hbase'...
remote: Enumerating objects: 146, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (19/19), done.
Receiving objects:  63% (92/146)used 23 (delta 23), pack-reused 104Receiving objects:  57% (84/146)
Receiving objects: 100% (146/146), 25.96 KiB | 604.00 KiB/s, done.
Resolving deltas: 100% (82/82), done.
```

3) F:\big.data>cd docker-hbase
   F:\big.data\docker-hbase>docker-compose -f docker-compose-distributed-local.yml up -d

- *Navigate to the created folder 'docker-hbase' and run the docker-compose command to finally create the 'HBase container' in the distributed mode.*

```
……………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………
[+] Running 12/12
 - Network docker-hbase_default            Created
1.0s
 - Volume "docker-hbase_hadoop_historyserver"  Created
0.1s
```

```
 - Volume "docker-hbase_hadoop_namenode"      Created
0.1s
 - Volume "docker-hbase_hadoop_datanode"      Created
0.1s
 - Container zoo                        Started                                                              22.4s
 - Container datanode                   Started                                                                18.7s
 - Container hbase-regionserver         Started                                                                  16.2s
 - Container nodemanager                Started                                                                  23.4s
 - Container namenode                   Started                                                                 12.7s
 - Container resourcemanager            Started                                                                  23.5s
 - Container historyserver              Started                                                                 17.3s
 - Container hbase-master               Started                                                                 14.1s
```

Creating Bridge Network (Multi-node):

1) F:\big.data\docker-hbase>docker network ls

- *Upon creating the HBase container, a docker bridge network named 'docker-base_default' is created automatically.*

```
NETWORK ID     NAME                  DRIVER    SCOPE
4c3f1ab6301a   bridge                bridge    local
e25448a9d187   docker-hbase_default  bridge    local
b8de1f6c79f3   host                  host      local
8b9a8940e314   none                  null      local
```

2) F:\big.data\docker-hbase>docker network inspect docker-hbase_default

- *The bridge network 'docker-hbase_default' contains all the containers running inside HBase (such as datanode, namenode etc.), which are connected like a bridge, so that they can communicate with each other.*

```
[
  {
    "Name": "docker-hbase_default",
    "Id": "acdf6f4810f937dd6dd72acb0a7786c609ab08f2a4c64f13552c5ccfd2b0e923",
    "Created": "2022-05-27T13:48:37.5925704Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
"Containers": {
  "2721065a329f9fa7a98a7a4274c7150a929d2bb40e671620a63375603753ceb1": {
    "Name": "nodemanager",
    "EndpointID": "316feae367fe7dac3d777d0b73f5074ec151fbe1db84138a7a6c7a788cf8358c",
    "MacAddress": "02:42:ac:12:00:08",
    "IPv4Address": "172.18.0.8/16",
    "IPv6Address": ""
  },
  "77bb5c2f0f9bd5e57d5df03e8a29dab662bafddc8444d80512437721c97d20ae": {
    "Name": "historyserver",
    "EndpointID": "5d7b2ea9f4bd5e53669a5e32b46bedd7137bc331f1cb9155aa2e1cc2b4d975d1",
    "MacAddress": "02:42:ac:12:00:05",
    "IPv4Address": "172.18.0.5/16",
    "IPv6Address": ""
  },
  "9f611b702bb3ad392b9ef008ed9d9e7b2bfbc2e68eafe171d77b833fb42115b3": {
    "Name": "hbase-regionserver",
    "EndpointID": "23788a15404c14d31a45ea5ec452e100c905b74a17ef8a28bfeaa3010f6e080b",
    "MacAddress": "02:42:ac:12:00:04",
    "IPv4Address": "172.18.0.4/16",
    "IPv6Address": ""
  },
  "b2e026235fda84eb19d7a737e3651c6063ee1ff3a861f6bdafd0cea49afbedac": {
    "Name": "resourcemanager",
    "EndpointID": "45d76d8877722b83afa6a0cc1d2f0af2cf1568c68f2392fc50d4b6a2ee510fad",
    "MacAddress": "02:42:ac:12:00:09",
    "IPv4Address": "172.18.0.9/16",
    "IPv6Address": ""
  },
  "d18db1728f63db4d917a48b90856eca279c1e69ef26eb15593cf1885ed06bdfa": {
    "Name": "zoo",
    "EndpointID": "16406f811a9c1086dd7e79dfccad1e565213fb555dab03f7c1d5cfd66761a0c6",
    "MacAddress": "02:42:ac:12:00:07",
    "IPv4Address": "172.18.0.7/16",
    "IPv6Address": ""
  },
  "f0cd77662772e61da0cd7c9359147aeac271af5df7b492325a16687d4c909a4f": {
    "Name": "namenode",
    "EndpointID": "00aa495441a4e13b8f6c78d0ef5b8906db16fc65df576749e70b10d0f799f9eb",
    "MacAddress": "02:42:ac:12:00:02",
    "IPv4Address": "172.18.0.2/16",
    "IPv6Address": ""
  },
  "f9216f7f4f5c322e9a370dddcb9e6a9b4597878dc1de6bf1889cf778b761d13e": {
    "Name": "hbase-master",
    "EndpointID": "2c5ebbb94e66afa15a4dcf46ec051bcda3c736e09d31c8ad9db89e8ea1e7d9ce",
    "MacAddress": "02:42:ac:12:00:03",
    "IPv4Address": "172.18.0.3/16",
    "IPv6Address": ""
```

```
            },
            "fbfd495a4588a8f44835b81e3628042a2f69718a1fc074f54f36503d53930abb": {
                "Name": "datanode",
                "EndpointID": "24602bbe2dbcab193acd00b7bbd1fde6cbaaea92802d1a11715932f03abdb7fe",
                "MacAddress": "02:42:ac:12:00:06",
                "IPv4Address": "172.18.0.6/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {
            "com.docker.compose.network": "default",
            "com.docker.compose.project": "docker-hbase",
            "com.docker.compose.version": "2.2.3"
        }
    }
]
```

Importing Dataset:

- **Importing CSV from local system to Hadoop container.**

1) F:\big.data\docker-hbase>docker cp ecommerce_data.csv f0cd77662772:/hadoop-data

2) F:\big.data\docker-hbase>docker exec -it f0cd77662772 /bin/bash

3) root@f0cd77662772:/# cd hadoop-data
   root@f0cd77662772:/hadoop-data# ls

```
ecommerce_data.csv
```

- **Copying CSV from container to HDFS.**

1) root@f0cd77662772:/hadoop-data# hadoop fs -mkdir -p /bda/hbasedata

2) root@f0cd77662772:/hadoop-data# hadoop fs -copyFromLocal /hadoop-data/*.csv /bda/hbasedata

3) root@f0cd77662772:/hadoop-data# hadoop fs -ls /bda/hbasedata

```
Found 1 items
-rw-r--r--   3 root supergroup 5668612855 2022-05-27 14:14 /bda/hbasedata/ecommerce_data.csv
```

- **Importing CSV from local system to HBase distributed container.**

1) F:\big.data\docker-hbase>docker cp ecommerce_data.csv f9216f7f4f5c:/hadoop-data

2) F:\big.data\docker-hbase>docker exec -it f9216f7f4f5c /bin/bash

3) root@hbase-master:/# cd hadoop-data
   root@hbase-master:/hadoop-data# ls

```
ecommerce_data.csv
```

- **Entering in Hbase container’s command line/shell.**

4) root@hbase-master:~# hbase shell

```
2022-05-27 14:30:24,073 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
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
```

Table Creation & Data Mapping:

1) hbase(main):006:0> hbase(main):009:0> create 'ecommerce2','event_time','cf1','cf2','cf3','cf4','cf5','cf6','cf7','cf8'

- *Creates a table named 'ecommerce2' with **event_time** as row key, along with eight column families (cf1 to cf8).*

```
0 row(s) in 4.6490 seconds

=> Hbase::Table - ecommerce2
```

2) hbase(main):003:0> create 'hbase:labels', 'f'

- *Creates another table to avoid a recurring error of HBase.*

```
0 row(s) in 4.5770 seconds

=> Hbase::Table - hbase:labels
```

3) root@hbase-master:/# hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator=',' -Dimporttsv.columns=HBASE_ROW_KEY,cf1:event_type,cf1:product_id,cf2:category_id,cf2:category_code,cf3:brand,cf3:price,cf4:user_id,cf4:user_session ecommerce /hadoop-data/ecommerce_data.csv

- *After table creation and data preparation, now mapping and loading the data from HDFS to HBase.*

```
……………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………
2022-05-27 17:47:26,802 INFO  [LocalJobRunner Map Task Executor #0]
client.ConnectionManager$HConnectionImplementation: Closing zookeeper sessionid=0x18105c71a2e02ac
2022-05-27 17:47:26,957 INFO  [LocalJobRunner Map Task Executor #0-EventThread] zookeeper.ClientCnxn:
EventThread shut down
2022-05-27 17:47:26,957 INFO  [LocalJobRunner Map Task Executor #0] zookeeper.ZooKeeper: Session:
0x18105c71a2e02ac closed
2022-05-27 17:47:26,967 INFO  [LocalJobRunner Map Task Executor #0] mapred.Task:
Task:attempt_local330161248_0001_m_000168_0 is done. And is in the process of committing
2022-05-27 17:47:26,968 INFO  [LocalJobRunner Map Task Executor #0] mapred.LocalJobRunner: map
```

```
2022-05-27 17:47:26,968 INFO  [LocalJobRunner Map Task Executor #0] mapred.Task: Task
'attempt_local330161248_0001_m_000168_0' done.
2022-05-27 17:47:26,968 INFO  [LocalJobRunner Map Task Executor #0] mapred.LocalJobRunner: Finishing
task: attempt_local330161248_0001_m_000168_0
2022-05-27 17:47:26,968 INFO  [Thread-33] mapred.LocalJobRunner: map task executor complete.
2022-05-27 17:47:27,256 INFO  [main] mapreduce.Job: Job job_local330161248_0001 completed successfully
2022-05-27 17:47:27,413 INFO  [main] mapreduce.Job: Counters: 24
	File System Counters
		FILE: Number of bytes read=486465058211
		FILE: Number of bytes written=4425411185
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=0
		HDFS: Number of bytes written=0
		HDFS: Number of read operations=0
		HDFS: Number of large read operations=0
		HDFS: Number of write operations=0
	Map-Reduce Framework
		Map input records=42448765
		Map output records=42448765
		Input split bytes=17069
		Spilled Records=0
		Failed Shuffles=0
		Merged Map outputs=0
		GC time elapsed (ms)=24817
		CPU time spent (ms)=0
		Physical memory (bytes) snapshot=0
		Virtual memory (bytes) snapshot=0
		Total committed heap usage (bytes)=34267856896
	ImportTsv
		Bad Lines=0
	File Input Format Counters
		Bytes Read=5669300983
	File Output Format Counters
		Bytes Written=0
```

Table Management Commands:

1) hbase(main):007:0> list

- *Lists all the tables in HBase to check if the table 'ecommerce2' is present.*

```
TABLE
ecommerce2
1 row(s) in 0.6370 seconds

=> ["ecommerce2"]
```

2) hbase(main):002:0> describe 'ecommerce2'

- *Gives information about the column families present in the table 'ecommerce2'.*

```
Table ecommerce2 is ENABLED
ecommerce2
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf1', 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 => 'cf2', 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 => 'cf3', 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 => 'cf4', 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 => 'cf5', 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 => 'cf6', 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 => 'cf7', 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 => 'cf8', 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 => 'event_time', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false',
KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION =>
'NONE', MIN_VERSIO

NS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}

9 row(s) in 0.1680 seconds
```

3) hbase(main):008:0> is_enabled 'ecommerce2'

- *Checks whether the table 'ecommerce2' is enabled or not.*

```
true
0 row(s) in 0.0160 seconds
```

4) hbase(main):009:0> exists 'ecommerce2'

- *Checks the existence of the table 'ecommerce2'.*

```
Table ecommerce2 does exist
    0    row(s) in 0.0130 seconds
```

5) hbase(main):011:0> is_disabled 'ecommerce2'

- *Checks whether the table 'ecommerce2' is disabled or not.*

```
false
0 row(s) in 0.0630 seconds
```

6) hbase(main):003:0> alter 'ecommerce2', NAME => 'cf8', VERSIONS => 5
   hbase(main):004:0> describe 'ecommerce2'

- *Changing the 'cf8' column family in table 'ecommerce2' from the current value to keep a maximum of 5 cell VERSIONS.*

```
Updating all regions with the new schema...
0/1 regions updated.
1/1 regions updated.
Done.
0 row(s) in 4.1910 seconds

Table ecommerce2 is ENABLED
ecommerce2
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf1', 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 => 'cf2', 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 => 'cf3', 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 => 'cf4', 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 => 'cf5', 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 => 'cf6', 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 => 'cf7', 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 => 'cf8', BLOOMFILTER => 'ROW', VERSIONS => '5', 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 => 'event_time', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false',
KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION =>
'NONE', MIN_VERSIO
NS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
9 row(s) in 0.0430 seconds
```

7) hbase(main):001:0> alter_status 'ecommerce2'

- *Indicating the number of regions of the table 'ecommerce2' which have received the updated schema.*

```
1/1 regions updated.
Done.
```

8) hbase(main):006:0> locate_region 'ecommerce2', '2019-10-01 00:00:30 UTC'

- *Locate the region of table 'ecommerce2' and the row key '2019-10-01 00:00:30 UTC'.*

```
HOST                          REGION
 hbase-region:16020                 {ENCODED => 294ce23b90015bd3414f11c78d906d45, NAME =>
'ecommerce2,,1653667447076.294ce23b90015bd3414f11c78d906d45.', STARTKEY => '', ENDKEY => ''}
1 row(s) in 0.0240 seconds
```

9) hbase(main):006:0> disable 'ecommerce_data'

- *Disables the table 'ecommerce_data'. If the table needs to be dropped, it has to disable first.*

```
0 row(s) in 4.5210 seconds
```

10) hbase(main):007:0> drop 'ecommerce_data'

- *Deletes the table 'ecommerce_data' present in HBase.*

```
0 row(s) in 4.4260 seconds
```

Data Manipulation Commands:

1) hbase(main):020:0> append 'ecommerce2', '2019-10-01 00:00:08 UTC', 'cf4:category_code', 'appliances.kitchen.mixer'

   hbase(main):022:0> get 'ecommerce2', '2019-10-01 00:00:08 UTC', {COLUMN => 'cf4'}

- *Inserts a new value of 'appliances.kitchen.mixer' in the column family 'category_code' against row key '2019-10-01 00:00:08 UTC', and display it.*

```
0 row(s) in 0.1940 seconds

COLUMN                    CELL
 cf4:category_code            timestamp=1654002775428, value=appliances.kitchen.mixer
1 row(s) in 0.0420 seconds
```

2) hbase(main):001:0> count 'ecommerce2', CACHE => 10000000

- *Count the number of rows in the table 'ecommerce2'. The current count is displayed every 1000 rows by default.*

```
Current count: 1000, row: 2019-10-01 02:19:13 UTC
Current count: 2000, row: 2019-10-01 02:35:54 UTC
Current count: 3000, row: 2019-10-01 02:52:34 UTC
Current count: 4000, row: 2019-10-01 03:09:14 UTC
Current count: 5000, row: 2019-10-01 03:25:54 UTC
Current count: 6000, row: 2019-10-01 03:42:34 UTC
Current count: 7000, row: 2019-10-01 03:59:14 UTC
Current count: 8000, row: 2019-10-01 04:15:54 UTC
Current count: 9000, row: 2019-10-01 04:32:34 UTC
Current count: 10000, row: 2019-10-01 04:49:14 UTC
Current count: 11000, row: 2019-10-01 05:05:54 UTC
Current count: 12000, row: 2019-10-01 05:22:34 UTC
………………………………………………………………………………………………………………
………………………………………………………………………………………………………………
………………………………………………………………………………………………………………
………………………………………………………………………………………………………………
Current count: 2621000, row: 2019-10-31 23:49:49 UTC
2621539 row(s) in 69.1020 seconds
```

3) hbase(main):002:0> get 'ecommerce2', '2019-10-01 00:00:22 UTC'

- *Get the columns and their values present against the row key '2019-10-01 00:00:22 UTC' in the table 'ecommerce2'.*

```
COLUMN                    CELL
 cf1:event_type            timestamp=1653667708084, value=view
 cf2:product_id            timestamp=1653667708084, value=1480714
 cf3:category_id           timestamp=1653667708084, value=2053013561092866779
 cf4:category_code           timestamp=1653667708084, value=computers.desktop
 cf5:brand                timestamp=1653667708084, value=pulser
 cf6:price               timestamp=1653667708084, value=921.49
 cf7:user_id              timestamp=1653667708084, value=512742880
 cf8:user_session            timestamp=1653667708084, value=0d0d91c2-c9c2-4e81-90a5-
86594dec0db9
8 row(s) in 0.1640 seconds
```

4) hbase(main):004:0> get 'ecommerce2', '2019-10-01 03:57:38 UTC', {COLUMN => 'cf6'}

- *Get the column family 'cf6' and its value present against the row key '2019-10-01 03:57:38 UTC' in the table 'ecommerce2'.*

```
COLUMN                    CELL
 cf6:price               timestamp=1653667708084, value=609.72
1 row(s) in 0.0120 seconds
```

5) hbase(main):006:0> get 'ecommerce2', '2019-10-01 05:50:41 UTC', {COLUMN => ['cf3','cf4','cf5']}

- *Get the column families 'cf3, cf4, cf5' and their respective values present against the row key '2019-10-01 05:50:41 UTC' in the table 'ecommerce2'.*

```
COLUMN                    CELL
 cf3:category_id           timestamp=1653667708084, value=2053013561092866779
 cf4:category_code           timestamp=1653667708084, value=computers.desktop
 cf5:brand                timestamp=1653667708084, value=pulser
3 row(s) in 0.0190 seconds
```

6) hbase(main):014:0> get 'ecommerce2', '2019-10-01 00:00:23 UTC', {FILTER => "ValueFilter(=, 'binary:midea')"}

- *Using ValueFilter, get the value 'midea' present against the row key '2019-10-01 05:50:41 UTC' in the table 'ecommerce2'.*

```
COLUMN                    CELL
 cf5:brand               timestamp=1653667708084, value=midea
1 row(s) in 0.0100 seconds
```

7) hbase(main):010:0> get_splits 'ecommerce2'

- *Returns the split points for the table 'ecommerce2'.*

```
Total number of splits = 1

=> []
```

8) hbase(main):032:0> put 'ecommerce2', '2019-10-01 00:00:15 UTC', 'cf4:category_code', 'apparel.perfumes.men'
hbase(main):040:0> get 'ecommerce2', '2019-10-01 00:00:15 UTC'

- *Put the value 'apparel.perfumes.men' in the column 'category_code' of family 'cf4' against the row key '2019-10-01 00:00:15 UTC', and display the contents.*

```
0 row(s) in 0.0700 seconds

COLUMN                    CELL
 cf1:event_type            timestamp=1653667708084, value=view
 cf2:product_id            timestamp=1653667708084, value=44600062
 cf3:category_id           timestamp=1653667708084, value=2103807459595387724
 cf4:category_code           timestamp=1653816014112, value=apparel.perfumes.men
 cf5:brand               timestamp=1653667708084, value=shiseido
 cf6:price              timestamp=1653667708084, value=35.79
 cf7:user_id              timestamp=1653667708084, value=541312140
 cf8:user_session            timestamp=1653667708084, value=72d76fde-8bb3-4e00-8c23-
a032dfed738c
8 row(s) in 0.0420 seconds
```

HBase Filters:

9) hbase(main):001:0> show_filters

- *Returns different filters that can be used to view the data.*

```
DependentColumnFilter
KeyOnlyFilter
ColumnCountGetFilter
SingleColumnValueFilter
PrefixFilter
SingleColumnValueExcludeFilter
FirstKeyOnlyFilter
ColumnRangeFilter
TimestampsFilter
FamilyFilter
QualifierFilter
ColumnPrefixFilter
RowFilter
MultipleColumnPrefixFilter
InclusiveStopFilter
PageFilter
ValueFilter
ColumnPaginationFilter
```

10) hbase(main):002:0> scan 'ecommerce2', {FILTER => "KeyOnlyFilter()"}

- *Returns the key portion i.e. column name of each key-value pair in the table ‘ecommerce2.*

```
ROW                        COLUMN+CELL
 2019-10-01 00:00:00 UTC           column=cf1:event_type, timestamp=1653667708084, value=
 2019-10-01 00:00:00 UTC           column=cf2:product_id, timestamp=1653667708084, value=
 2019-10-01 00:00:00 UTC           column=cf3:category_id, timestamp=1653667708084, value=
 2019-10-01 00:00:00 UTC           column=cf4:category_code, timestamp=1653667708084, value=
 2019-10-01 00:00:00 UTC           column=cf5:brand, timestamp=1653667708084, value=
 2019-10-01 00:00:00 UTC           column=cf6:price, timestamp=1653667708084, value=
 2019-10-01 00:00:00 UTC           column=cf7:user_id, timestamp=1653667708084, value=
 2019-10-01 00:00:00 UTC           column=cf8:user_session, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC          column=cf1:event_type, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC          column=cf2:product_id, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC          column=cf3:category_id, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC          column=cf4:category_code, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC          column=cf5:brand, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC          column=cf6:price, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC                column=cf7:user_id, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC                column=cf8:user_session, timestamp=1653667708084, value=
……………………………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………………………
```

11) hbase(main):001:0> scan 'ecommerce2', {FILTER => "FirstKeyOnlyFilter()"}

- *Returns the first key-value pair i.e. the first column and its value against each row key in the table 'ecommerce2'.*

```
ROW                              COLUMN+CELL
 2019-10-01 00:00:00 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:01 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:04 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:05 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:08 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:10 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:11 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:13 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:15 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:16 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:17 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:18 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:19 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
……………………………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………………………
```

12) hbase(main):001:0> scan 'ecommerce2', {FILTER => "PrefixFilter('1')"}

- *Returns only those key-values present in a row that starts with the row prefix '1'. Since there is no such row, hence it will return empty.*

```
ROW                              COLUMN+CELL
0 row(s) in 0.4750 seconds
```

13) hbase(main):002:0> scan 'ecommerce2', {FILTER => "PrefixFilter('2')"}

- *Returns only those key-values present in a row that starts with the row prefix '2'.*

```
ROW                              COLUMN+CELL
 2019-10-01 00:00:00 UTC              column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:00 UTC              column=cf2:product_id, timestamp=1653667708084, value=3900821
 2019-10-01 00:00:00 UTC              column=cf3:category_id, timestamp=1653667708084,
value=2053013552326770905
 2019-10-01 00:00:00 UTC              column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.water_heater
 2019-10-01 00:00:00 UTC              column=cf5:brand, timestamp=1653667708084, value=aqua
 2019-10-01 00:00:00 UTC              column=cf6:price, timestamp=1653667708084, value=33.20
 2019-10-01 00:00:00 UTC              column=cf7:user_id, timestamp=1653667708084, value=554748717
 2019-10-01 00:00:00 UTC              column=cf8:user_session, timestamp=1653667708084, value=9333dfbd-
b87a-4708-9857-6336556b0fcc
 2019-10-01 00:00:01 UTC             column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:01 UTC             column=cf2:product_id, timestamp=1653667708084, value=1307067
 2019-10-01 00:00:01 UTC             column=cf3:category_id, timestamp=1653667708084,
value=2053013558920217191
 2019-10-01 00:00:01 UTC             column=cf4:category_code, timestamp=1653667708084,
value=computers.notebook
 2019-10-01 00:00:01 UTC             column=cf5:brand, timestamp=1653667708084, value=lenovo
 2019-10-01 00:00:01 UTC             column=cf6:price, timestamp=1653667708084, value=251.74
 2019-10-01 00:00:01 UTC             column=cf7:user_id, timestamp=1653667708084, value=550050854
 2019-10-01 00:00:01 UTC             column=cf8:user_session, timestamp=1653667708084, value=7c90fc70-
0e80-4590-96f3-13c02c18c713
……………………………………………………………………………………………………………
……………………………………………………………………………………………………………
……………………………………………………………………………………………………………
```

14) hbase(main):001:0> scan 'ecommerce2', {FILTER => "ColumnPrefixFilter('u')"}

- *Returns only those key-values of a column that starts with the prefix 'u', which are 'user_id' and 'user_session'.*

```
ROW                              COLUMN+CELL
 2019-10-01 00:00:00 UTC              column=cf7:user_id, timestamp=1653667708084, value=554748717
 2019-10-01 00:00:00 UTC              column=cf8:user_session, timestamp=1653667708084, value=9333dfbd-
b87a-4708-9857-6336556b0fcc
 2019-10-01 00:00:01 UTC             column=cf7:user_id, timestamp=1653667708084, value=550050854
 2019-10-01 00:00:01 UTC             column=cf8:user_session, timestamp=1653667708084, value=7c90fc70-
0e80-4590-96f3-13c02c18c713
 2019-10-01 00:00:04 UTC              column=cf7:user_id, timestamp=1653667708084, value=535871217
 2019-10-01 00:00:04 UTC              column=cf8:user_session, timestamp=1653667708084, value=c6bd7419-
2748-4c56-95b4-8cec9ff8b80d
 2019-10-01 00:00:05 UTC              column=cf7:user_id, timestamp=1653667708084, value=512742880
 2019-10-01 00:00:05 UTC              column=cf8:user_session, timestamp=1653667708084, value=0d0d91c2-
c9c2-4e81-90a5-86594dec0db9
 2019-10-01 00:00:08 UTC              column=cf7:user_id, timestamp=1653667708084, value=550978835
 2019-10-01 00:00:08 UTC              column=cf8:user_session, timestamp=1653667708084, value=6280d577-
25c8-4147-99a7-abc6048498d6
 2019-10-01 00:00:10 UTC             column=cf7:user_id, timestamp=1653667708084, value=520571932
 2019-10-01 00:00:10 UTC             column=cf8:user_session, timestamp=1653667708084, value=ac1cd4e5-
a3ce-4224-a2d7-ff660a105880
 2019-10-01 00:00:11 UTC            column=cf7:user_id, timestamp=1653667708084, value=530282093
 2019-10-01 00:00:11 UTC            column=cf8:user_session, timestamp=1653667708084, value=50a293fb-
5940-41b2-baf3-17af0e812101
 2019-10-01 00:00:13 UTC             column=cf7:user_id, timestamp=1653667708084, value=555444559
 2019-10-01 00:00:13 UTC             column=cf8:user_session, timestamp=1653667708084, value=98b88fa0-
d8fa-4b9d-8a71-3dd403afab85
……………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………
```

15) hbase(main):001:0> scan 'ecommerce2', {FILTER => "MultipleColumnPrefixFilter('b','p')"}

- *Returns only those key-values present in a column that starts with any of the column prefixes 'b' and 'p', which are 'brand', 'price', and 'product_id'.*

```
ROW                          COLUMN+CELL
 2019-10-01 00:00:00 UTC              column=cf2:product_id, timestamp=1653667708084, value=3900821
 2019-10-01 00:00:00 UTC              column=cf5:brand, timestamp=1653667708084, value=aqua
 2019-10-01 00:00:00 UTC              column=cf6:price, timestamp=1653667708084, value=33.20
 2019-10-01 00:00:01 UTC             column=cf2:product_id, timestamp=1653667708084, value=1307067
 2019-10-01 00:00:01 UTC             column=cf5:brand, timestamp=1653667708084, value=lenovo
 2019-10-01 00:00:01 UTC             column=cf6:price, timestamp=1653667708084, value=251.74
 2019-10-01 00:00:04 UTC              column=cf2:product_id, timestamp=1653667708084, value=1004237
 2019-10-01 00:00:04 UTC              column=cf5:brand, timestamp=1653667708084, value=apple
 2019-10-01 00:00:04 UTC              column=cf6:price, timestamp=1653667708084, value=1081.98
 2019-10-01 00:00:05 UTC              column=cf2:product_id, timestamp=1653667708084, value=1480613
 2019-10-01 00:00:05 UTC              column=cf5:brand, timestamp=1653667708084, value=pulser
 2019-10-01 00:00:05 UTC              column=cf6:price, timestamp=1653667708084, value=908.62
 2019-10-01 00:00:08 UTC              column=cf2:product_id, timestamp=1653667708084, value=31500053
 2019-10-01 00:00:08 UTC              column=cf5:brand, timestamp=1653667708084, value=luminarc
 2019-10-01 00:00:08 UTC              column=cf6:price, timestamp=1653667708084, value=41.16
……………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………
```

16) hbase(main):001:0> scan 'ecommerce2', {FILTER => "ColumnCountGetFilter(4)"}

- *Returns the first 4 rows and their columns of the table 'ecommerce2'.*

```
ROW                              COLUMN+CELL
 2019-10-01 00:00:00 UTC              column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:00 UTC              column=cf2:product_id, timestamp=1653667708084, value=3900821
 2019-10-01 00:00:00 UTC              column=cf3:category_id, timestamp=1653667708084,
value=2053013552326770905
 2019-10-01 00:00:00 UTC              column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.water_heater
 2019-10-01 00:00:01 UTC              column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:01 UTC              column=cf2:product_id, timestamp=1653667708084, value=1307067
 2019-10-01 00:00:01 UTC              column=cf3:category_id, timestamp=1653667708084,
value=2053013558920217191
 2019-10-01 00:00:01 UTC              column=cf4:category_code, timestamp=1653667708084,
value=computers.notebook
 2019-10-01 00:00:04 UTC              column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:04 UTC              column=cf2:product_id, timestamp=1653667708084, value=1004237
 2019-10-01 00:00:04 UTC              column=cf3:category_id, timestamp=1653667708084,
value=2053013555631882655
 2019-10-01 00:00:04 UTC              column=cf4:category_code, timestamp=1653667708084,
value=electronics.smartphone
………………………………………………………………………………………………………………
………………………………………………………………………………………………………………
………………………………………………………………………………………………………………
```

17) hbase(main):001:0> scan 'ecommerce2', {FILTER => "ColumnPaginationFilter(2,4)"}

- *Takes two arguments, limit and offset. Returns the two columns which are present after the first four columns.*

```
ROW                              COLUMN+CELL
 2019-10-01 00:00:00 UTC              column=cf5:brand, timestamp=1653667708084, value=aqua
 2019-10-01 00:00:00 UTC              column=cf6:price, timestamp=1653667708084, value=33.20
 2019-10-01 00:00:01 UTC              column=cf5:brand, timestamp=1653667708084, value=lenovo
 2019-10-01 00:00:01 UTC              column=cf6:price, timestamp=1653667708084, value=251.74
 2019-10-01 00:00:04 UTC              column=cf5:brand, timestamp=1653667708084, value=apple
 2019-10-01 00:00:04 UTC              column=cf6:price, timestamp=1653667708084, value=1081.98
 2019-10-01 00:00:05 UTC              column=cf5:brand, timestamp=1653667708084, value=pulser
 2019-10-01 00:00:05 UTC              column=cf6:price, timestamp=1653667708084, value=908.62
 2019-10-01 00:00:08 UTC              column=cf5:brand, timestamp=1653667708084,
value=luminarc
 2019-10-01 00:00:08 UTC              column=cf6:price, timestamp=1653667708084, value=41.16
 2019-10-01 00:00:10 UTC              column=cf5:brand, timestamp=1653667708084, value=baden
 2019-10-01 00:00:10 UTC              column=cf6:price, timestamp=1653667708084, value=102.71
 2019-10-01 00:00:11 UTC              column=cf5:brand, timestamp=1653667708084,
value=samsung
 2019-10-01 00:00:11 UTC              column=cf6:price, timestamp=1653667708084, value=900.64
 2019-10-01 00:00:13 UTC                column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 00:00:13 UTC                column=cf6:price, timestamp=1653667708084, value=102.38
……………………………………………………………………………………………………………………
…………….……………………………………………………………………………………………………
……..………………………………………………………………………………………………………..
```

18) hbase(main):003:0> scan 'ecommerce2', {FILTER => "InclusiveStopFilter('2019-10-01 00:00:04 UTC')"}

- *Returns all key-values present in rows up to the row key '2019-10-01 00:00:04 UTC', including the specified row key itself.*

```
ROW                            COLUMN+CELL
 2019-10-01 00:00:00 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:00 UTC                column=cf2:product_id, timestamp=1653667708084, value=3900821
 2019-10-01 00:00:00 UTC                column=cf3:category_id, timestamp=1653667708084,
value=2053013552326770905
 2019-10-01 00:00:00 UTC                column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.water_heater
 2019-10-01 00:00:00 UTC                column=cf5:brand, timestamp=1653667708084, value=aqua
 2019-10-01 00:00:00 UTC                column=cf6:price, timestamp=1653667708084, value=33.20
 2019-10-01 00:00:00 UTC                column=cf7:user_id, timestamp=1653667708084, value=554748717
 2019-10-01 00:00:00 UTC                column=cf8:user_session, timestamp=1653667708084, value=9333dfbd-
b87a-4708-9857-6336556b0fcc
 2019-10-01 00:00:01 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:01 UTC                column=cf2:product_id, timestamp=1653667708084, value=1307067
 2019-10-01 00:00:01 UTC                column=cf3:category_id, timestamp=1653667708084,
value=2053013558920217191
 2019-10-01 00:00:01 UTC                column=cf4:category_code, timestamp=1653667708084,
value=computers.notebook
 2019-10-01 00:00:01 UTC                column=cf5:brand, timestamp=1653667708084, value=lenovo
 2019-10-01 00:00:01 UTC                column=cf6:price, timestamp=1653667708084, value=251.74
 2019-10-01 00:00:01 UTC                column=cf7:user_id, timestamp=1653667708084, value=550050854
 2019-10-01 00:00:01 UTC                column=cf8:user_session, timestamp=1653667708084, value=7c90fc70-
0e80-4590-96f3-13c02c18c713
 2019-10-01 00:00:04 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:04 UTC                column=cf2:product_id, timestamp=1653667708084, value=1004237
 2019-10-01 00:00:04 UTC                column=cf3:category_id, timestamp=1653667708084,
value=2053013555631882655
 2019-10-01 00:00:04 UTC                column=cf4:category_code, timestamp=1653667708084,
value=electronics.smartphone
 2019-10-01 00:00:04 UTC                column=cf5:brand, timestamp=1653667708084, value=apple
 2019-10-01 00:00:04 UTC                column=cf6:price, timestamp=1653667708084, value=1081.98
 2019-10-01 00:00:04 UTC                column=cf7:user_id, timestamp=1653667708084, value=535871217
 2019-10-01 00:00:04 UTC                column=cf8:user_session, timestamp=1653667708084, value=c6bd7419-
2748-4c56-95b4-8cec9ff8b80d
```

```
3 row(s) in 0.0590 seconds
```

19) hbase(main):001:0> scan 'ecommerce2', {FILTER => "PageFilter(2)"}

- *Returns the page size number of rows from the table. The page size selected here is '2'.*

```
ROW                              COLUMN+CELL
 2019-10-01 00:00:00 UTC                  column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:00 UTC                  column=cf2:product_id, timestamp=1653667708084, value=3900821
 2019-10-01 00:00:00 UTC                  column=cf3:category_id, timestamp=1653667708084,
value=2053013552326770905
 2019-10-01 00:00:00 UTC                  column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.water_heater
 2019-10-01 00:00:00 UTC                  column=cf5:brand, timestamp=1653667708084, value=aqua
 2019-10-01 00:00:00 UTC                  column=cf6:price, timestamp=1653667708084, value=33.20
 2019-10-01 00:00:00 UTC                  column=cf7:user_id, timestamp=1653667708084, value=554748717
 2019-10-01 00:00:00 UTC                  column=cf8:user_session, timestamp=1653667708084, value=9333dfbd-
b87a-4708-9857-6336556b0fcc
 2019-10-01 00:00:01 UTC                 column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:01 UTC                 column=cf2:product_id, timestamp=1653667708084, value=1307067
 2019-10-01 00:00:01 UTC                 column=cf3:category_id, timestamp=1653667708084,
value=2053013558920217191
 2019-10-01 00:00:01 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=computers.notebook
 2019-10-01 00:00:01 UTC                 column=cf5:brand, timestamp=1653667708084, value=lenovo
 2019-10-01 00:00:01 UTC                 column=cf6:price, timestamp=1653667708084, value=251.74
 2019-10-01 00:00:01 UTC                 column=cf7:user_id, timestamp=1653667708084, value=550050854
 2019-10-01 00:00:01 UTC                 column=cf8:user_session, timestamp=1653667708084, value=7c90fc70-
0e80-4590-96f3-13c02c18c713
2 row(s) in 0.7800 seconds
```

20) hbase(main):004:0> scan 'ecommerce2', {FILTER => "RowFilter(<=, 'binary:2019-10-01 00:00:05 UTC')"}

- *Returns all the key-value pairs present in and before the row key '2019-10-01 00:00:05 UTC'.*

```
ROW                              COLUMN+CELL
 2019-10-01 00:00:00 UTC                  column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:00 UTC                  column=cf2:product_id, timestamp=1653667708084, value=3900821
 2019-10-01 00:00:00 UTC                  column=cf3:category_id, timestamp=1653667708084,
value=2053013552326770905
 2019-10-01 00:00:00 UTC                  column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.water_heater
 2019-10-01 00:00:00 UTC                  column=cf5:brand, timestamp=1653667708084, value=aqua
 2019-10-01 00:00:00 UTC              column=cf6:price, timestamp=1653667708084, value=33.20
 2019-10-01 00:00:00 UTC              column=cf7:user_id, timestamp=1653667708084, value=554748717
 2019-10-01 00:00:00 UTC              column=cf8:user_session, timestamp=1653667708084, value=9333dfbd-
b87a-4708-9857-6336556b0fcc
 2019-10-01 00:00:01 UTC             column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:01 UTC             column=cf2:product_id, timestamp=1653667708084, value=1307067
 2019-10-01 00:00:01 UTC             column=cf3:category_id, timestamp=1653667708084,
value=2053013558920217191
 2019-10-01 00:00:01 UTC             column=cf4:category_code, timestamp=1653667708084,
value=computers.notebook
 2019-10-01 00:00:01 UTC             column=cf5:brand, timestamp=1653667708084, value=lenovo
 2019-10-01 00:00:01 UTC             column=cf6:price, timestamp=1653667708084, value=251.74
 2019-10-01 00:00:01 UTC             column=cf7:user_id, timestamp=1653667708084, value=550050854
 2019-10-01 00:00:01 UTC             column=cf8:user_session, timestamp=1653667708084, value=7c90fc70-
0e80-4590-96f3-13c02c18c713
 2019-10-01 00:00:04 UTC              column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:04 UTC              column=cf2:product_id, timestamp=1653667708084, value=1004237
 2019-10-01 00:00:04 UTC              column=cf3:category_id, timestamp=1653667708084,
value=2053013555631882655
 2019-10-01 00:00:04 UTC              column=cf4:category_code, timestamp=1653667708084,
value=electronics.smartphone
 2019-10-01 00:00:04 UTC              column=cf5:brand, timestamp=1653667708084, value=apple
 2019-10-01 00:00:04 UTC              column=cf6:price, timestamp=1653667708084, value=1081.98
 2019-10-01 00:00:04 UTC              column=cf7:user_id, timestamp=1653667708084, value=535871217
 2019-10-01 00:00:04 UTC              column=cf8:user_session, timestamp=1653667708084, value=c6bd7419-
2748-4c56-95b4-8cec9ff8b80d
 2019-10-01 00:00:05 UTC              column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:05 UTC              column=cf2:product_id, timestamp=1653667708084, value=1480613
 2019-10-01 00:00:05 UTC              column=cf3:category_id, timestamp=1653667708084,
value=2053013561092866779
 2019-10-01 00:00:05 UTC              column=cf4:category_code, timestamp=1653667708084,
value=computers.desktop
 2019-10-01 00:00:05 UTC              column=cf5:brand, timestamp=1653667708084, value=pulser
 2019-10-01 00:00:05 UTC              column=cf6:price, timestamp=1653667708084, value=908.62
 2019-10-01 00:00:05 UTC              column=cf7:user_id, timestamp=1653667708084, value=512742880
 2019-10-01 00:00:05 UTC              column=cf8:user_session, timestamp=1653667708084, value=0d0d91c2-
c9c2-4e81-90a5-86594dec0db9
4 row(s) in 38.1680 seconds
```

21) hbase(main):006:0> scan 'ecommerce2', {FILTER => "FamilyFilter(<=, 'binaryprefix:cf3')"}

- *Returns all the key-value pairs of column family 'cf3' and its predecessors, such that it contains column families cf1, cf2, and cf3 itself.*

```
ROW                           COLUMN+CELL
 2019-10-01 00:00:00 UTC              column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:00 UTC              column=cf2:product_id, timestamp=1653667708084, value=3900821
 2019-10-01 00:00:00 UTC                column=cf3:category_id, timestamp=1653667708084,
value=2053013552326770905
 2019-10-01 00:00:01 UTC               column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:01 UTC               column=cf2:product_id, timestamp=1653667708084, value=1307067
 2019-10-01 00:00:01 UTC               column=cf3:category_id, timestamp=1653667708084,
value=2053013558920217191
 2019-10-01 00:00:04 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:04 UTC                column=cf2:product_id, timestamp=1653667708084, value=1004237
 2019-10-01 00:00:04 UTC                column=cf3:category_id, timestamp=1653667708084,
value=2053013555631882655
 2019-10-01 00:00:05 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:05 UTC                column=cf2:product_id, timestamp=1653667708084, value=1480613
 2019-10-01 00:00:05 UTC                column=cf3:category_id, timestamp=1653667708084,
value=2053013561092866779
 2019-10-01 00:00:08 UTC                column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:08 UTC                column=cf2:product_id, timestamp=1653667708084, value=31500053
 2019-10-01 00:00:08 UTC                column=cf3:category_id, timestamp=1653667708084,
value=2053013558031024687
…………………………………………………………………………………………………………………………………………….
…………………………………………………………………………………………………………………..…………………
…………………………………………………………………………………………………………………………………………….
```

22) hbase(main):001:0> scan 'ecommerce2', {FILTER => "QualifierFilter(=, 'substring:and')"}

- *Returns all the key-value pairs of the columns that matches its name with the substring ‘and’. So, the output column is ‘brand’.*

```
ROW                          COLUMN+CELL
 2019-10-01 00:00:00 UTC            column=cf5:brand, timestamp=1653667708084, value=aqua
 2019-10-01 00:00:01 UTC            column=cf5:brand, timestamp=1653667708084, value=lenovo
 2019-10-01 00:00:04 UTC            column=cf5:brand, timestamp=1653667708084, value=apple
 2019-10-01 00:00:05 UTC            column=cf5:brand, timestamp=1653667708084, value=pulser
 2019-10-01 00:00:08 UTC            column=cf5:brand, timestamp=1653667708084,
value=luminarc
 2019-10-01 00:00:10 UTC            column=cf5:brand, timestamp=1653667708084, value=baden
 2019-10-01 00:00:11 UTC            column=cf5:brand, timestamp=1653667708084,
value=samsung
 2019-10-01 00:00:13 UTC            column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 00:00:15 UTC            column=cf5:brand, timestamp=1653667708084, value=shiseido
 2019-10-01 00:00:16 UTC            column=cf5:brand, timestamp=1653667708084, value=brw
 2019-10-01 00:00:17 UTC            column=cf5:brand, timestamp=1653667708084, value=
 2019-10-01 00:00:18 UTC            column=cf5:brand, timestamp=1653667708084, value=bosch
 2019-10-01 00:00:19 UTC            column=cf5:brand, timestamp=1653667708084, value=apple
 2019-10-01 00:00:20 UTC            column=cf5:brand, timestamp=1653667708084, value=jbl
 2019-10-01 00:00:22 UTC            column=cf5:brand, timestamp=1653667708084, value=pulser
```

……
……………..………
………..……

23) hbase(main):001:0> scan 'ecommerce2', {FILTER => "QualifierFilter(=, 'substring:id')"}

- *Returns all the key-value pairs of the columns that matches its name with the substring 'id'. So, the output columns are 'product_id', 'category_id', and 'user_id'.*

```
ROW                          COLUMN+CELL
 2019-10-01 00:00:00 UTC            column=cf2:product_id, timestamp=1653667708084, value=3900821
 2019-10-01 00:00:00 UTC            column=cf3:category_id, timestamp=1653667708084,
value=2053013552326770905
 2019-10-01 00:00:00 UTC            column=cf7:user_id, timestamp=1653667708084, value=554748717
 2019-10-01 00:00:01 UTC            column=cf2:product_id, timestamp=1653667708084, value=1307067
 2019-10-01 00:00:01 UTC            column=cf3:category_id, timestamp=1653667708084,
value=2053013558920217191
 2019-10-01 00:00:01 UTC            column=cf7:user_id, timestamp=1653667708084, value=550050854
 2019-10-01 00:00:04 UTC            column=cf2:product_id, timestamp=1653667708084, value=1004237
 2019-10-01 00:00:04 UTC            column=cf3:category_id, timestamp=1653667708084,
value=2053013555631882655
 2019-10-01 00:00:04 UTC            column=cf7:user_id, timestamp=1653667708084, value=535871217
 2019-10-01 00:00:05 UTC            column=cf2:product_id, timestamp=1653667708084, value=1480613
 2019-10-01 00:00:05 UTC            column=cf3:category_id, timestamp=1653667708084,
value=2053013561092866779
 2019-10-01 00:00:05 UTC            column=cf7:user_id, timestamp=1653667708084, value=512742880
 2019-10-01 00:00:08 UTC            column=cf2:product_id, timestamp=1653667708084, value=31500053
 2019-10-01 00:00:08 UTC            column=cf3:category_id, timestamp=1653667708084,
value=2053013558031024687
 2019-10-01 00:00:08 UTC            column=cf7:user_id, timestamp=1653667708084, value=550978835
……………………………………………………………………………………………………………………………….
………………………………………………………………………………………………………………..…………
……………………………………………………………………………………………………………………………….
```

24) hbase(main):001:0> scan 'ecommerce2', {FILTER => "ValueFilter(=, 'binary:haier')"}

- *Returns key-value pairs against each row that matches with the string 'haier'.*

```
ROW                          COLUMN+CELL
 2019-10-01 00:00:13 UTC            column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 00:13:47 UTC            column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 02:41:31 UTC            column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 02:51:08 UTC            column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 02:55:39 UTC                column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 02:57:46 UTC                column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 02:58:39 UTC                column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 02:58:52 UTC                column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 03:03:11 UTC                column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 03:08:10 UTC                column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 03:43:14 UTC                column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 03:43:15 UTC                column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 03:53:48 UTC                column=cf5:brand, timestamp=1653667708084, value=haier
………………………………………………………………………………………………………………………………
………………………………………………………………………………………………………………………………
………………………………………………………………………………………………………………………………
```

25) hbase(main):001:0> scan 'ecommerce2', {FILTER => "DependentColumnFilter('cf4', 'category_code')"}

- *Returns all the key-value pairs that have the same timestamp as in the column family ‘cf4’ with column ‘category_code.*

```
ROW                          COLUMN+CELL
 2019-10-01 00:00:00 UTC              column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:00 UTC              column=cf2:product_id, timestamp=1653667708084, value=3900821
 2019-10-01 00:00:00 UTC              column=cf3:category_id, timestamp=1653667708084,
value=2053013552326770905
 2019-10-01 00:00:00 UTC              column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.water_heater
 2019-10-01 00:00:00 UTC              column=cf5:brand, timestamp=1653667708084, value=aqua
 2019-10-01 00:00:00 UTC              column=cf6:price, timestamp=1653667708084, value=33.20
 2019-10-01 00:00:00 UTC              column=cf7:user_id, timestamp=1653667708084, value=554748717
 2019-10-01 00:00:00 UTC              column=cf8:user_session, timestamp=1653667708084, value=9333dfbd-
b87a-4708-9857-6336556b0fcc
 2019-10-01 00:00:01 UTC              column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:01 UTC              column=cf2:product_id, timestamp=1653667708084, value=1307067
 2019-10-01 00:00:01 UTC              column=cf3:category_id, timestamp=1653667708084,
value=2053013558920217191
 2019-10-01 00:00:01 UTC              column=cf4:category_code, timestamp=1653667708084,
value=computers.notebook
 2019-10-01 00:00:01 UTC              column=cf5:brand, timestamp=1653667708084, value=lenovo
 2019-10-01 00:00:01 UTC              column=cf6:price, timestamp=1653667708084, value=251.74
 2019-10-01 00:00:01 UTC              column=cf7:user_id, timestamp=1653667708084, value=550050854
 2019-10-01 00:00:01 UTC              column=cf8:user_session, timestamp=1653667708084, value=7c90fc70-
0e80-4590-96f3-13c02c18c713
………………………………………………………………………………………………………………………………
………………………………………………………………………………………………………………………………
………………………………………………………………………………………………………………………………
```

26) hbase(main):001:0> scan 'ecommerce2', {FILTER => "DependentColumnFilter('cf1', 'event_type', true)"}

- *Returns all the key-value pairs that have the same timestamp as in the column family 'cf1' with column 'event_type'. Family 'cf1' itself will not return because the Boolean argument 'dropDependentColumn' is set to 'true'.*

```
ROW                              COLUMN+CELL
 2019-10-01 00:00:00 UTC              column=cf2:product_id, timestamp=1653667708084, value=3900821
 2019-10-01 00:00:00 UTC              column=cf3:category_id, timestamp=1653667708084,
value=2053013552326770905
 2019-10-01 00:00:00 UTC              column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.water_heater
 2019-10-01 00:00:00 UTC              column=cf5:brand, timestamp=1653667708084, value=aqua
 2019-10-01 00:00:00 UTC              column=cf6:price, timestamp=1653667708084, value=33.20
 2019-10-01 00:00:00 UTC              column=cf7:user_id, timestamp=1653667708084, value=554748717
 2019-10-01 00:00:00 UTC              column=cf8:user_session, timestamp=1653667708084, value=9333dfbd-
b87a-4708-9857-6336556b0fcc
 2019-10-01 00:00:01 UTC              column=cf2:product_id, timestamp=1653667708084, value=1307067
 2019-10-01 00:00:01 UTC              column=cf3:category_id, timestamp=1653667708084,
value=2053013558920217191
 2019-10-01 00:00:01 UTC              column=cf4:category_code, timestamp=1653667708084,
value=computers.notebook
 2019-10-01 00:00:01 UTC              column=cf5:brand, timestamp=1653667708084, value=lenovo
 2019-10-01 00:00:01 UTC              column=cf6:price, timestamp=1653667708084, value=251.74
 2019-10-01 00:00:01 UTC              column=cf7:user_id, timestamp=1653667708084, value=550050854
 2019-10-01 00:00:01 UTC              column=cf8:user_session, timestamp=1653667708084, value=7c90fc70-
0e80-4590-96f3-13c02c18c713
………………………………………………………………………………………………………….
…………………………………………………………………………………………………..………………
………………………………………………………………………………………………………….
```

27) hbase(main):002:0> scan 'ecommerce2', {FILTER => "ColumnRangeFilter('brand', true, 'product_id', true)"}

- *Returns only those keys and columns that are between 'brand' and 'product_id'. The Boolean arguments 'true' shows that both the specified columns are inclusive.*

```
ROW                              COLUMN+CELL
 2019-10-01 00:00:00 UTC              column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:00 UTC              column=cf2:product_id, timestamp=1653667708084, value=3900821
 2019-10-01 00:00:00 UTC              column=cf3:category_id, timestamp=1653667708084,
value=2053013552326770905
```

```
 2019-10-01 00:00:00 UTC                    column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.water_heater
 2019-10-01 00:00:00 UTC                    column=cf5:brand, timestamp=1653667708084, value=aqua
 2019-10-01 00:00:00 UTC                    column=cf6:price, timestamp=1653667708084, value=33.20
 2019-10-01 00:00:01 UTC                    column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:01 UTC                    column=cf2:product_id, timestamp=1653667708084, value=1307067
 2019-10-01 00:00:01 UTC                    column=cf3:category_id, timestamp=1653667708084,
value=2053013558920217191
 2019-10-01 00:00:01 UTC                    column=cf4:category_code, timestamp=1653667708084,
value=computers.notebook
 2019-10-01 00:00:01 UTC                    column=cf5:brand, timestamp=1653667708084, value=lenovo
 2019-10-01 00:00:01 UTC                    column=cf6:price, timestamp=1653667708084, value=251.74
 2019-10-01 00:00:04 UTC                    column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:04 UTC                    column=cf2:product_id, timestamp=1653667708084, value=1004237
 2019-10-01 00:00:04 UTC                    column=cf3:category_id, timestamp=1653667708084,
value=2053013555631882655
 2019-10-01 00:00:04 UTC                    column=cf4:category_code, timestamp=1653667708084,
value=electronics.smartphone
 2019-10-01 00:00:04 UTC                    column=cf5:brand, timestamp=1653667708084, value=apple
 2019-10-01 00:00:04 UTC                    column=cf6:price, timestamp=1653667708084, value=1081.98
…………………………………………………………………………………………………………………………………….
…………………………………………………………………………………………………………………..………………
…………………………………………………………………………………………………………………………………….
```

28) hbase(main):003:0> scan 'ecommerce2', {FILTER => "PrefixFilter('1') AND KeyOnlyFilter()"}

- *Returns empty, since there is no row key that starts with '1'.*

```
ROW                         COLUMN+CELL
0 row(s) in 0.0530 seconds
```

29) hbase(main):004:0> scan 'ecommerce2', {FILTER => "PrefixFilter('2') AND KeyOnlyFilter()"}

- *Returns only keys of those rows that starts with '2'.*

```
ROW                         COLUMN+CELL
 2019-10-01 00:00:00 UTC            column=cf1:event_type, timestamp=1653667708084, value=
 2019-10-01 00:00:00 UTC            column=cf2:product_id, timestamp=1653667708084, value=
 2019-10-01 00:00:00 UTC            column=cf3:category_id, timestamp=1653667708084, value=
 2019-10-01 00:00:00 UTC            column=cf4:category_code, timestamp=1653667708084,
value=
 2019-10-01 00:00:00 UTC                column=cf5:brand, timestamp=1653667708084, value=
 2019-10-01 00:00:00 UTC                column=cf6:price, timestamp=1653667708084, value=
 2019-10-01 00:00:00 UTC                column=cf7:user_id, timestamp=1653667708084, value=
 2019-10-01 00:00:00 UTC                column=cf8:user_session, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC               column=cf1:event_type, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC               column=cf2:product_id, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC               column=cf3:category_id, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC               column=cf4:category_code, timestamp=1653667708084,
value=
 2019-10-01 00:00:01 UTC               column=cf5:brand, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC               column=cf6:price, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC               column=cf7:user_id, timestamp=1653667708084, value=
 2019-10-01 00:00:01 UTC               column=cf8:user_session, timestamp=1653667708084, value=
……………………………………………………………………………………………………………………………………………
………….….…………………………………………………………………………………………………………………………
………..….………………………………………………………………………………………………………………………..
```

30) hbase(main):005:0> scan 'ecommerce2', {FILTER => "RowFilter(<, 'binary:2019-10-01 00:00:13 UTC') AND QualifierFilter(=, 'substring:user')"}

- *Returns key-value pairs of columns that starts with the string 'user' against all rows that are present before '2019-10-01 00:00:13 UTC'.*

```
ROW                            COLUMN+CELL
 2019-10-01 00:00:00 UTC              column=cf7:user_id, timestamp=1653667708084, value=554748717
 2019-10-01 00:00:00 UTC              column=cf8:user_session, timestamp=1653667708084, value=9333dfbd-
b87a-4708-9857-6336556b0fcc
 2019-10-01 00:00:01 UTC             column=cf7:user_id, timestamp=1653667708084, value=550050854
 2019-10-01 00:00:01 UTC             column=cf8:user_session, timestamp=1653667708084, value=7c90fc70-
0e80-4590-96f3-13c02c18c713
 2019-10-01 00:00:04 UTC              column=cf7:user_id, timestamp=1653667708084, value=535871217
 2019-10-01 00:00:04 UTC              column=cf8:user_session, timestamp=1653667708084, value=c6bd7419-
2748-4c56-95b4-8cec9ff8b80d
 2019-10-01 00:00:05 UTC              column=cf7:user_id, timestamp=1653667708084, value=512742880
 2019-10-01 00:00:05 UTC              column=cf8:user_session, timestamp=1653667708084, value=0d0d91c2-
c9c2-4e81-90a5-86594dec0db9
 2019-10-01 00:00:08 UTC              column=cf7:user_id, timestamp=1653667708084, value=550978835
 2019-10-01 00:00:08 UTC              column=cf8:user_session, timestamp=1653667708084, value=6280d577-
25c8-4147-99a7-abc6048498d6
 2019-10-01 00:00:10 UTC             column=cf7:user_id, timestamp=1653667708084, value=520571932
 2019-10-01 00:00:10 UTC             column=cf8:user_session, timestamp=1653667708084, value=ac1cd4e5-
a3ce-4224-a2d7-ff660a105880
 2019-10-01 00:00:11 UTC            column=cf7:user_id, timestamp=1653667708084, value=530282093
 2019-10-01 00:00:11 UTC            column=cf8:user_session, timestamp=1653667708084, value=50a293fb-
5940-41b2-baf3-17af0e812101
```

31) hbase(main):001:0> scan 'ecommerce2', {FILTER => "ValueFilter(=, 'binary:purchase') OR ValueFilter(=, 'binary:ariston')"}

- *Returns key-value pairs against each row that has either the value 'purchase' or 'ariston' in it.*

```
ROW                          COLUMN+CELL
 2019-10-01 00:01:25 UTC          column=cf5:brand, timestamp=1653667708084, value=ariston
 2019-10-01 00:01:28 UTC          column=cf5:brand, timestamp=1653667708084, value=ariston
 2019-10-01 00:02:14 UTC          column=cf1:event_type, timestamp=1653667708084, value=purchase
 2019-10-01 00:09:26 UTC          column=cf1:event_type, timestamp=1653667708084, value=purchase
 2019-10-01 00:10:56 UTC          column=cf1:event_type, timestamp=1653667708084, value=purchase
 2019-10-01 00:12:14 UTC          column=cf1:event_type, timestamp=1653667708084, value=purchase
 2019-10-01 00:14:14 UTC          column=cf1:event_type, timestamp=1653667708084, value=purchase
 2019-10-01 00:14:20 UTC          column=cf5:brand, timestamp=1653667708084, value=ariston
 2019-10-01 00:16:40 UTC          column=cf5:brand, timestamp=1653667708084, value=ariston
 2019-10-01 00:17:53 UTC          column=cf5:brand, timestamp=1653667708084, value=ariston
 2019-10-01 02:19:12 UTC          column=cf1:event_type, timestamp=1653667708084, value=purchase
 2019-10-01 02:19:34 UTC          column=cf1:event_type, timestamp=1653667708084, value=purchase
……………………………………………………………………………………………………………
……………………………………………………………………………………………………………
……………………………………………………………………………………………………………
```

32) hbase(main):001:0> scan 'ecommerce2', {FILTER => "PageFilter(3) AND ColumnPrefixFilter('e')"}

- *Returns the first three rows of the table containing only those key-values of a column that starts with 'e'.*

```
ROW                          COLUMN+CELL
 2019-10-01 00:00:00 UTC          column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:01 UTC          column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:04 UTC          column=cf1:event_type, timestamp=1653667708084, value=view
3 row(s) in 0.4050 seconds
```

33) hbase(main):001:0> scan 'ecommerce2', {COLUMNS => 'cf5', FILTER => "ValueFilter(=, 'regexstring:^d.*')"}

- *Returns key values of the column family 'cf5:brand' in which the name starts with the letter 'd'.*

```
ROW                          COLUMN+CELL
 2019-10-01 00:00:30 UTC         column=cf5:brand, timestamp=1653667708084, value=dauscher
 2019-10-01 00:04:01 UTC         column=cf5:brand, timestamp=1653667708084, value=dewalt
 2019-10-01 00:04:53 UTC         column=cf5:brand, timestamp=1653667708084, value=dior
 2019-10-01 00:09:04 UTC         column=cf5:brand, timestamp=1653667708084, value=delonghi
 2019-10-01 00:13:35 UTC         column=cf5:brand, timestamp=1653667708084, value=dauscher
 2019-10-01 00:18:58 UTC         column=cf5:brand, timestamp=1653667708084, value=doogee
 2019-10-01 00:19:01 UTC         column=cf5:brand, timestamp=1653667708084, value=doogee
 2019-10-01 00:19:05 UTC         column=cf5:brand, timestamp=1653667708084, value=doogee
 2019-10-01 00:58:07 UTC         column=cf5:brand, timestamp=1653667708084, value=dauscher
 2019-10-01 00:58:11 UTC         column=cf5:brand, timestamp=1653667708084, value=dauscher
 2019-10-01 01:14:09 UTC         column=cf5:brand, timestamp=1653667708084, value=dxracer
 2019-10-01 01:20:39 UTC         column=cf5:brand, timestamp=1653667708084, value=dior
 2019-10-01 02:17:12 UTC         column=cf5:brand, timestamp=1653667708084, value=dauscher
 2019-10-01 02:18:49 UTC         column=cf5:brand, timestamp=1653667708084, value=dewalt
………………………………………………………………………………………………………………………………………………
………………………………………………………………………………………………………………………………………………
………………………………………………………………………………………………………………………………………………
```

34) hbase(main):003:0> scan 'ecommerce2', {COLUMNS => 'cf5', FILTER => "ValueFilter(=, 'regexstring:el*be')"}

- *Returns key values of the column family 'cf5:brand' in which the name either contain the letters 'el' or 'be'.*

```
ROW                          COLUMN+CELL
 2019-10-01 02:34:08 UTC         column=cf5:brand, timestamp=1653667708084, value=febest
 2019-10-01 03:46:59 UTC         column=cf5:brand, timestamp=1653667708084, value=febest
 2019-10-01 03:52:58 UTC         column=cf5:brand, timestamp=1653667708084, value=febest
………………………………………………………………………………………………………………………………………………
………………………………………………………………………………………………………………………………………………
2019-10-01 04:32:30 UTC          column=cf5:brand, timestamp=1653667708084, value=wellberg
 2019-10-01 05:14:59 UTC         column=cf5:brand, timestamp=1653667708084, value=rebellion
………………………………………………………………………………………………………………………………………………
………………………………………………………………………………………………………………………………………………
2019-10-01 07:35:26 UTC          column=cf5:brand, timestamp=1653667708084, value=mebelev
 2019-10-01 08:12:42 UTC         column=cf5:brand, timestamp=1653667708084, value=febest
 2019-10-01 09:18:23 UTC         column=cf5:brand, timestamp=1653667708084, value=wellberg
 2019-10-01 09:31:20 UTC         column=cf5:brand, timestamp=1653667708084, value=wellberg
 2019-10-01 09:32:13 UTC         column=cf5:brand, timestamp=1653667708084, value=wellberg
 2019-10-01 10:26:20 UTC         column=cf5:brand, timestamp=1653667708084, value=febest
 2019-10-01 10:55:00 UTC         column=cf5:brand, timestamp=1653667708084, value=febest
 2019-10-01 10:57:23 UTC                    column=cf5:brand, timestamp=1653667708084, value=ostamebel
……………………………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………………………
```

35) hbase(main):001:0> scan 'ecommerce2', {COLUMNS => 'cf4', FILTER => "ValueFilter(=, 'binaryprefix:app')"}

- *Returns key values of the column family 'cf4:category_code' in which the name starts with the string 'app'.*

```
ROW                                     COLUMN+CELL
 2019-10-01 00:00:00 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.water_heater
 2019-10-01 00:00:10 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=apparel.shoes.keds
 2019-10-01 00:00:13 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.water_heater
 2019-10-01 00:00:18 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=appliances.kitchen.mixer
 2019-10-01 00:00:23 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.air_heater
 2019-10-01 00:00:28 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=apparel.shoes
 2019-10-01 00:00:30 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.vacuum
 2019-10-01 00:00:31 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=apparel.shoes.keds
 2019-10-01 00:00:33 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=apparel.shoes
 2019-10-01 00:00:35 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=appliances.kitchen.washer
 2019-10-01 00:01:02 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=appliances.kitchen.mixer
 2019-10-01 00:01:21 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=apparel.shoes.keds
 2019-10-01 00:01:23 UTC                 column=cf4:category_code, timestamp=1653667708084,
value=appliances.kitchen.mixer
……………………………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………………………
```

36) hbase(main):001:0> scan 'ecommerce2', {STARTROW => '2019-10-01 00:00:19 UTC'}

- *Using STARTROW, display only rows from '2019-10-01 00:00:19 UTC' onwards.*

```
ROW                         COLUMN+CELL
 2019-10-01 00:00:19 UTC            column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:19 UTC            column=cf2:product_id, timestamp=1653667708084, value=1005135
 2019-10-01 00:00:19 UTC            column=cf3:category_id, timestamp=1653667708084,
value=2053013555631882655
 2019-10-01 00:00:19 UTC            column=cf4:category_code, timestamp=1653667708084,
value=electronics.smartphone
 2019-10-01 00:00:19 UTC            column=cf5:brand, timestamp=1653667708084, value=apple
 2019-10-01 00:00:19 UTC            column=cf6:price, timestamp=1653667708084, value=1747.79
 2019-10-01 00:00:19 UTC            column=cf7:user_id, timestamp=1653667708084, value=535871217
 2019-10-01 00:00:19 UTC            column=cf8:user_session, timestamp=1653667708084, value=c6bd7419-
2748-4c56-95b4-8cec9ff8b80d
 2019-10-01 00:00:20 UTC            column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:20 UTC            column=cf2:product_id, timestamp=1653667708084, value=4803399
 2019-10-01 00:00:20 UTC            column=cf3:category_id, timestamp=1653667708084,
value=2053013554658804075
 2019-10-01 00:00:20 UTC            column=cf4:category_code, timestamp=1653667708084,
value=electronics.audio.headphone
 2019-10-01 00:00:20 UTC            column=cf5:brand, timestamp=1653667708084, value=jbl
 2019-10-01 00:00:20 UTC            column=cf6:price, timestamp=1653667708084, value=33.21
 2019-10-01 00:00:20 UTC            column=cf7:user_id, timestamp=1653667708084, value=555428858
 2019-10-01 00:00:20 UTC            column=cf8:user_session, timestamp=1653667708084, value=8a6afed4-
77f8-40c9-8e76-e062b28216ce
..............................................................................................
..............................................................................................
..............................................................................................
```

37) hbase(main):002:0> scan 'ecommerce2', {COLUMNS => ['cf2:product_id', 'cf4:category_code'], LIMIT => 3, STARTROW => '2019-10-01 03:10:28 UTC'}

- *Returns key-value pairs of column families 'cf2' and 'cf4' for three rows starting from row key '2019-10-01 03:10:28 UTC'.*

```
ROW                         COLUMN+CELL
 2019-10-01 03:10:28 UTC            column=cf2:product_id, timestamp=1653667708084, value=6200702
 2019-10-01 03:10:28 UTC            column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.air_heater
 2019-10-01 03:10:29 UTC            column=cf2:product_id, timestamp=1653667708084, value=1004240
 2019-10-01 03:10:29 UTC            column=cf4:category_code, timestamp=1653667708084,
value=electronics.smartphone
 2019-10-01 03:10:30 UTC            column=cf2:product_id, timestamp=1653667708084, value=1004856
 2019-10-01 03:10:30 UTC            column=cf4:category_code, timestamp=1653667708084,
value=electronics.smartphone
3 row(s) in 0.3530 seconds
```

38) hbase(main):003:0> scan 'ecommerce2', {COLUMNS => ['cf3:category_id', 'cf5:brand'], STARTROW => '2019-10-01 00:00:20 UTC', STOPROW => '2019-10-01 00:00:26 UTC'}

- *Returns key-value pairs of column families 'cf3' and 'cf5' starting from row key '2019-10-01 00:00:20 UTC' and stop before '2019-10-01 00:00:26 UTC' (exclusive).*

```
ROW                              COLUMN+CELL
 2019-10-01 00:00:20 UTC                  column=cf3:category_id, timestamp=1653667708084,
value=2053013554658804075
 2019-10-01 00:00:20 UTC                  column=cf5:brand, timestamp=1653667708084, value=jbl
 2019-10-01 00:00:22 UTC                  column=cf3:category_id, timestamp=1653667708084,
value=2053013561092866779
 2019-10-01 00:00:22 UTC                  column=cf5:brand, timestamp=1653667708084, value=pulser
 2019-10-01 00:00:23 UTC                  column=cf3:category_id, timestamp=1653667708084,
value=2053013552293216471
 2019-10-01 00:00:23 UTC                  column=cf5:brand, timestamp=1653667708084, value=midea
 2019-10-01 00:00:24 UTC                  column=cf3:category_id, timestamp=1653667708084,
value=2061717937420501730
 2019-10-01 00:00:24 UTC                  column=cf5:brand, timestamp=1653667708084, value=
 2019-10-01 00:00:25 UTC                  column=cf3:category_id, timestamp=1653667708084,
value=2053013557225718275
 2019-10-01 00:00:25 UTC                  column=cf5:brand, timestamp=1653667708084, value=gran-stone
5 row(s) in 0.1370 seconds
```

39) hbase(main):006:0> delete 'ecommerce2', '2019-10-01 00:00:13 UTC', 'cf2:product_id', 1653667708084
hbase(main):007:0> scan 'ecommerce2', {FILTER => "PageFilter(8)"}

- *Deletes the column family 'cf2:product_id' present against the row key '2019-10-01 00:00:13 UTC' in the timestamp 1653667708084, and display the first eight rows to confirm it.*

```
0 row(s) in 0.0800 seconds

ROW                              COLUMN+CELL
……………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………
2019-10-01 00:00:13 UTC                  column=cf1:event_type, timestamp=1653667708084, value=view
 2019-10-01 00:00:13 UTC                  column=cf3:category_id, timestamp=1653667708084,
value=2053013552326770905
 2019-10-01 00:00:13 UTC                  column=cf4:category_code, timestamp=1653667708084,
value=appliances.environment.water_heater
 2019-10-01 00:00:13 UTC                  column=cf5:brand, timestamp=1653667708084, value=haier
 2019-10-01 00:00:13 UTC                column=cf6:price, timestamp=1653667708084, value=102.38
 2019-10-01 00:00:13 UTC                column=cf7:user_id, timestamp=1653667708084, value=555444559
 2019-10-01 00:00:13 UTC                column=cf8:user_session, timestamp=1653667708084, value=98b88fa0-
d8fa-4b9d-8a71-3dd403afab85
8 row(s) in 0.2050 seconds
```

40) hbase(main):008:0> deleteall 'ecommerce2', '2019-10-01 00:00:16 UTC'
hbase(main):001:0> scan 'ecommerce2', {FILTER => "RowFilter(=, 'binary:2019-10-01 00:00:16 UTC')"}

- *Deletes all the column families (cf1 to cf8) against the row key '2019-10-01 00:00:16 UTC', and display the specified row key to confirm.*

```
0 row(s) in 0.0210 seconds
ROW                       COLUMN+CELL
0 row(s) in 29.7740 seconds
```

41) hbase(main):041:0> scan 'ecommerce2', {COLUMNS => ['cf7','cf8'], LIMIT => 6}

- *Returns the column families 'cf7' and 'cf8' present against the starting 6 row keys.*

```
ROW                             COLUMN+CELL
 2019-10-01 00:00:00 UTC                column=cf7:user_id, timestamp=1653667708084, value=554748717
 2019-10-01 00:00:00 UTC                column=cf8:user_session, timestamp=1653667708084, value=9333dfbd-
b87a-4708-9857-6336556b0fcc
 2019-10-01 00:00:01 UTC                column=cf7:user_id, timestamp=1653667708084, value=550050854
 2019-10-01 00:00:01 UTC                column=cf8:user_session, timestamp=1653667708084, value=7c90fc70-
0e80-4590-96f3-13c02c18c713
 2019-10-01 00:00:04 UTC                column=cf7:user_id, timestamp=1653667708084, value=535871217
 2019-10-01 00:00:04 UTC                column=cf8:user_session, timestamp=1653667708084, value=c6bd7419-
2748-4c56-95b4-8cec9ff8b80d
 2019-10-01 00:00:05 UTC                column=cf7:user_id, timestamp=1653667708084, value=512742880
 2019-10-01 00:00:05 UTC                column=cf8:user_session, timestamp=1653667708084, value=0d0d91c2-
c9c2-4e81-90a5-86594dec0db9
 2019-10-01 00:00:08 UTC                column=cf7:user_id, timestamp=1653667708084, value=550978835
 2019-10-01 00:00:08 UTC                column=cf8:user_session, timestamp=1653667708084, value=6280d577-
25c8-4147-99a7-abc6048498d6
 2019-10-01 00:00:10 UTC                column=cf7:user_id, timestamp=1653667708084, value=520571932
 2019-10-01 00:00:10 UTC                column=cf8:user_session, timestamp=1653667708084, value=ac1cd4e5-
a3ce-4224-a2d7-ff660a105880
6 row(s) in 0.0590 seconds
```

42) hbase(main):043:0> scan 'ecommerce2', {ROWPREFIXFILTER => '2', FILTER => "QualifierFilter(=, 'binary:price')"}

- *Returns the key values of column 'price' present against each row key which has a prefix of '2'.*

```
ROW                          COLUMN+CELL
 2019-10-01 00:00:00 UTC            column=cf6:price, timestamp=1653667708084, value=33.20
 2019-10-01 00:00:01 UTC            column=cf6:price, timestamp=1653667708084, value=251.74
 2019-10-01 00:00:04 UTC            column=cf6:price, timestamp=1653667708084, value=1081.98
 2019-10-01 00:00:05 UTC            column=cf6:price, timestamp=1653667708084, value=908.62
 2019-10-01 00:00:08 UTC            column=cf6:price, timestamp=1653667708084, value=41.16
 2019-10-01 00:00:10 UTC            column=cf6:price, timestamp=1653667708084, value=102.71
 2019-10-01 00:00:11 UTC            column=cf6:price, timestamp=1653667708084, value=900.64
 2019-10-01 00:00:13 UTC            column=cf6:price, timestamp=1653667708084, value=102.38
 2019-10-01 00:00:15 UTC            column=cf6:price, timestamp=1653667708084, value=35.79
 2019-10-01 00:00:17 UTC            column=cf6:price, timestamp=1653667708084, value=357.79
 2019-10-01 00:00:18 UTC            column=cf6:price, timestamp=1653667708084, value=58.95
 2019-10-01 00:00:19 UTC            column=cf6:price, timestamp=1653667708084, value=1747.79
 2019-10-01 00:00:20 UTC            column=cf6:price, timestamp=1653667708084, value=33.21
 2019-10-01 00:00:22 UTC            column=cf6:price, timestamp=1653667708084, value=921.49
 2019-10-01 00:00:23 UTC            column=cf6:price, timestamp=1653667708084, value=47.62
 2019-10-01 00:00:24 UTC            column=cf6:price, timestamp=1653667708084, value=151.87
………………………………………………………………………………………………………………………
………………………………………………………………………………………………………………………
………………………………………………………………………………………………………………………
```

Snapshot Shell Commands:

1) hbase(main):004:0> snapshot 'ecommerce2', 'ecomm_data'

- *Creates a snapshot named 'ecomm_data' of the specified table 'ecommerce2'.*

```
0 row(s) in 2.2580 seconds
```

2) hbase(main):007:0> list_snapshots

- *List all the snapshots taken. The only snapshot 'ecomm_data' will appear.*

```
SNAPSHOT                      TABLE + CREATION TIME
 ecomm_data                    ecommerce2 (Sun May 29 13:22:13 +0000 2022)
1 row(s) in 0.0420 seconds

=> ["ecomm_data"]
```

3) hbase(main):005:0> clone_snapshot 'ecomm_data', 'ecommerce_data'
   hbase(main):006:0> list

- *Creates a new table 'ecommerce_data' and clone the snapshot content into it.*

```
0 row(s) in 8.4430 seconds

TABLE
ecommerce2
ecommerce_data
2 row(s) in 0.0180 seconds
```

4) hbase(main):008:0> delete_snapshot 'ecomm_data'
   hbase(main):009:0> list_snapshots

- *Deletes the snapshot 'ecomm_data' which was created earlier.*

```
0 row(s) in 0.0980 seconds

SNAPSHOT                      TABLE + CREATION TIME
0 row(s) in 0.0110 seconds

=> []
```

Replication Commands:

1) hbase(main):002:0> add_peer '1', "server1.cie.com:2181:/hbase"
   hbase(main):001:0> add_peer '2', "zk1,zk2:2182:/hbase-prod"

- *Adding peers '1' and '2' to cluster to replicate the data.*

```
0 row(s) in 0.2920 seconds

0 row(s) in 0.3860 seconds
```

2) hbase(main):002:0> list_peers

- *List all replication peer clusters.*

```
PEER_ID CLUSTER_KEY STATE TABLE_CFS
 1 server1.cie.com:2181:/hbase ENABLED
 2 zk1,zk2:2182:/hbase-prod ENABLED
2 row(s) in 0.0510 seconds
```

3) hbase(main):014:0> list_replicated_tables

- *List all the tables and column families replicated from the cluster.*

```
TABLE:COLUMNFAMILY                ReplicationType
 ecommerce2:cf1                     GLOBAL
 ecommerce2:cf2                     GLOBAL
 ecommerce2:cf3                     GLOBAL
 ecommerce2:cf4                     GLOBAL
 ecommerce2:cf5                     GLOBAL
 ecommerce2:cf6                     GLOBAL
 ecommerce2:cf7                     GLOBAL
 ecommerce2:cf8                     GLOBAL
 ecommerce2:event_time              GLOBAL
9 row(s) in 0.0330 seconds
```

4) hbase(main):027:0> disable_peer '1'
   hbase(main):028:0> list_peers

- *Stops the replication stream to the specified peer cluster '1', and returns the list to confirm.*

```
0 row(s) in 0.0580 seconds

PEER_ID CLUSTER_KEY STATE TABLE_CFS
 1 server1.cie.com:2181:/hbase DISABLED
 2 zk1,zk2:2182:/hbase-prod ENABLED
2 row(s) in 0.0300 seconds
```

5) hbase(main):029:0> enable_peer '1'
   hbase(main):030:0> list_peers

- *Restarts the replication stream to the specified peer cluster '1', and returns the list to confirm.*

```
0 row(s) in 0.0910 seconds

PEER_ID CLUSTER_KEY STATE TABLE_CFS
 1 server1.cie.com:2181:/hbase ENABLED
 2 zk1,zk2:2182:/hbase-prod ENABLED
2 row(s) in 0.0160 seconds
```

6) hbase(main):031:0> remove_peer '2'
   hbase(main):032:0> list_peers

- *Stops the specified replication stream '2' and deletes all the meta information kept about it. Returns the list to confirm.*

```
0 row(s) in 0.0720 seconds

PEER_ID CLUSTER_KEY STATE TABLE_CFS
 1 server1.cie.com:2181:/hbase ENABLED
1 row(s) in 0.0090 seconds
```