

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA HỆ THỐNG THÔNG TIN**

---



**Bài tập lớn 2**

**MÔN HỌC: Cơ sở dữ liệu phân tán**

**Tên đề tài: Tìm hiểu và cài đặt Apache Hbase**

*Giảng viên hướng dẫn:*

**Nguyễn Minh Nhựt**

*Sinh viên thực hiện:*

**Nguyễn Hoàng Long – 19521788**

**Trần Nguyễn Hạnh Nguyên – 19521923**

**Lê Thế Tiệm – 19522330**

**Huỳnh Quốc Khánh – 19521677**

*TP HCM, Ngày 01 tháng 01 năm 2022*

## LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các thầy cô của trường Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP.HCM, đặc biệt là quý thầy cô khoa Hệ thống thông tin của trường đã giúp cho chúng em trang bị các kiến thức cơ bản, các kỹ năng thực tế và tạo điều kiện để chúng em có thể hoàn thành bài tập lớn môn học của mình.

Đặc biệt chúng em xin chân thành cảm ơn thầy Nguyễn Minh Nhựt đã nhiệt tình hướng dẫn hướng dẫn, quan tâm truyền đạt những kiến thức và kinh nghiệm, trực tiếp hướng dẫn tận tình, sửa chữa và đóng góp ý kiến quý báu cho chúng em trong suốt thời gian học tập để chúng em có thể hoàn thành tốt môn học này.

Trong thời gian một học kỳ thực hiện đề tài, nhóm tác giả đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó, nhóm tác giả vận dụng tối đa những gì đã thu thập được để hoàn thành một báo cáo đồ án tốt nhất. Tuy nhiên, trong quá trình thực hiện, nhóm tác giả không tránh khỏi những thiếu sót. Chính vì vậy, nhóm tác giả rất mong nhận được những sự góp ý từ phía Thầy/Cô nhằm hoàn thiện những kiến thức mà nhóm tác giả đã học tập và là hành trang để nhóm tác giả thực hiện tiếp các đề tài khác trong tương lai.

Nhóm chúng em xin chân thành cảm ơn!

*Nhóm thực hiện*

[illegible]

# MỤC LỤC

LỜI CẢM ƠN .....	1
CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI .....	4
1.1 Lý do chọn đề tài .....	4
1.2 Công cụ sử dụng.....	4
CHƯƠNG 2: APACHE HBASE .....	5
2.1 Giới thiệu.....	5
2.2 Tính năng của Apache HBase .....	5
2.3 Kiến trúc của Apache HBase .....	6
2.4 Lưu trữ dữ liệu trong Apache HBase .....	7
2.5 Data Flow trong Apache HBase.....	10
CHƯƠNG 3: CẤU HÌNH VÀ CÀI ĐẶT.....	11
3.1 Yêu cầu.....	11
3.2 Cài đặt .....	11
3.2.1 Cài đặt Hadoop .....	11
3.2.2 Cài đặt ZooKeeper.....	16
3.2.3 Cài đặt HBase .....	18
CHƯƠNG 4: THỰC HÀNH .....	22
4.1 Tạo Table và ColumnFamily.....	23
4.2 Insert dữ liệu.....	23
4.3 Update dữ liệu .....	24
4.4 Delete dữ liệu .....	25
4.4 Truy xuất dữ liệu .....	26
4.4.1 Key Only Filter.....	26
4.4.2 Prefix Filter.....	26
4.4.3 Column Prefix Filter và Multiple Column Prefix Filter .....	27
4.4.4 Value Filter .....	27

---

# CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

---

## 1.1 Lý do chọn đề tài

Cùng với tốc độ phát triển vô cùng nhanh của công nghệ thông tin hiện nay, hàng loạt những vấn đề mang tính cần thiết về nơi lưu trữ dữ liệu cũng như tốc độ truy xuất trên những cơ sở dữ liệu này được đặt ra. Lượng dữ liệu ngày càng nở ra một cách nhanh chóng khiến những cách lưu trữ dữ liệu trên những cơ sở dữ liệu truyền thống bị quá tải, để đáp ứng nhu cầu lưu trữ, xử lý trên lượng dữ liệu khổng lồ đó một cách nhanh chóng và hiệu quả, một khái niệm cơ sở dữ liệu mới được ra đời đó là NoSQL. Mục tiêu bài tìm hiểu này nhằm hướng tới việc tìm hiểu các khái niệm liên quan tới Apache HBase, một dạng cơ sở dữ liệu NoSQL hiện đang rất phổ biến và sử dụng rộng rãi.

## 1.2 Công cụ sử dụng

Trong quá trình thực hiện đề tài, nhóm tác giả đã sử dụng một số công cụ hỗ trợ xây dựng và thử nghiệm sau:

1. Java 8
2. Hadoop 2.7.7
3. Apache ZooKeeper 3.6.3
4. Apache Hbase 1.3.5
5. VirtualBox 6.1.30
6. Ubuntu server 18.04

---

## CHƯƠNG 2: APACHE HBASE

---

### 2.1 Giới thiệu

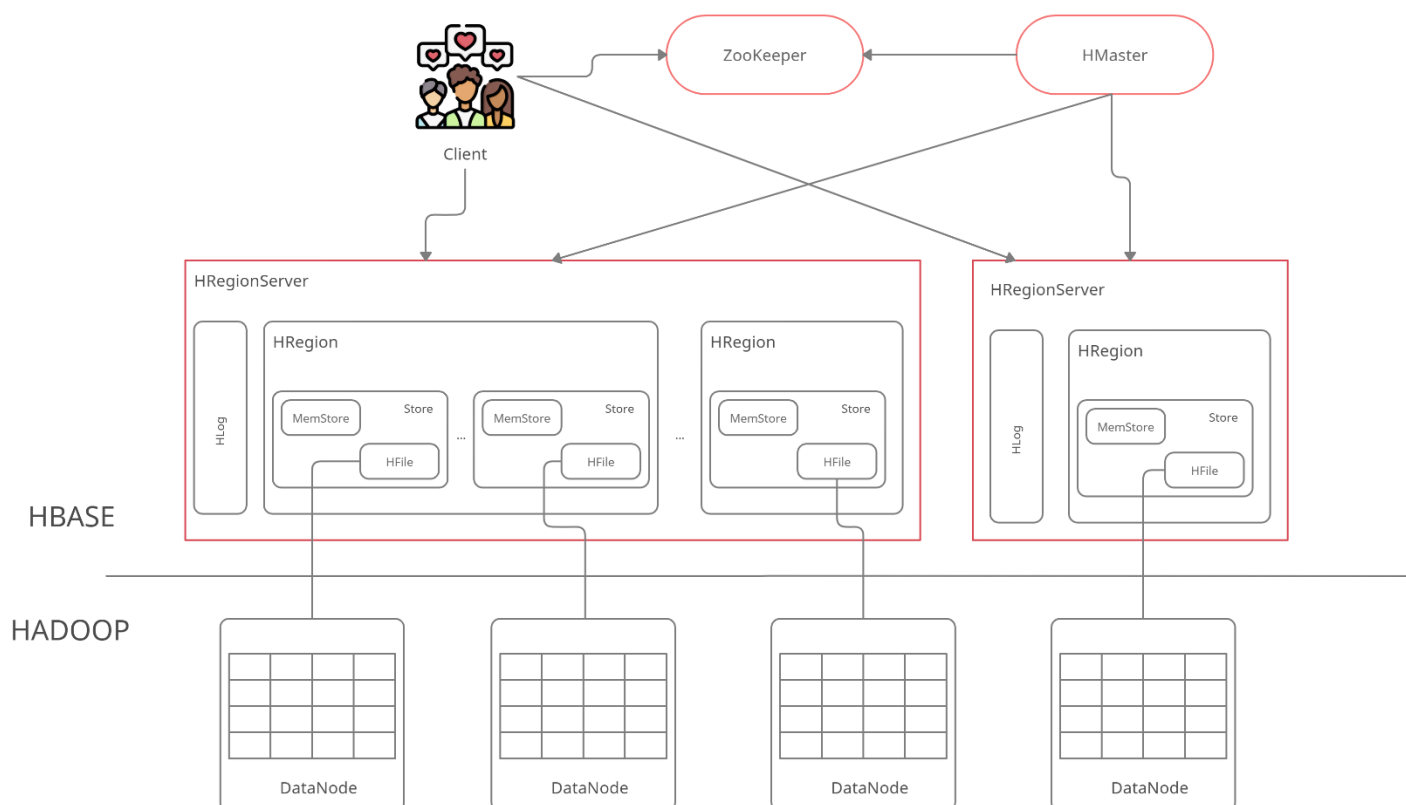
Apache HBase là một non-relational distributed database và là một dự án mã nguồn mở. Apache HBase lần đầu tiên được công bố vào năm 2008 bởi công ty Powerset, phục vụ cho những dự án cần xử lý và lưu trữ một lượng lớn dữ liệu. Sau này, Apache HBase được Apache Software Foundation đảm nhiệm tiếp để phát triển. Ban đầu nó được gọi là Google Big Table, sau đó được đổi thành HBase và được viết chủ yếu bằng ngôn ngữ Java và trở thành cơ sở dữ liệu phổ biến cho nhiều hệ thống xử lý Big Data

Apache HBase có thể chạy và cộng tác trực tiếp trên Hadoop Distributed File System (HDFS) để gia tăng tốc độ truy vấn dữ liệu, và còn là cơ sở dữ liệu đặc lực cho các hệ thống lưu trữ dữ liệu lớn, real-time queries

### 2.2 Tính năng của Apache HBase

1. Được xây dựng để giải quyết những vấn đề về độ trễ cao
2. Được sử dụng rộng rãi cho các ứng dụng có phương thức đọc hoặc viết ngẫu nhiên
3. Có khả năng lưu trữ một lượng dữ liệu rất lớn (lên tới hàng tỷ dòng)
4. Cung cấp khả năng mở rộng khả năng lưu trữ thông qua các cụm
5. Người dùng dễ dàng sử dụng Java API để thao tác trên HBase
6. Tự động hoặc tự cài đặt các phương thức sharding tables
7. Tự động hỗ trợ các Region Servers trên các cụm khi gặp lỗi

## 2.3 Kiến trúc của Apache HBase



### 1. HMaster

HMaster là thành phần trung tâm trong kiến trúc HBase. HMaster giám sát tất cả các Region Servers trong cụm, trong môi trường bao gồm nhiều cụm như Hadoop thì HMaster sẽ nằm ở NameNode, mọi thay đổi liên quan đến metadata đều phải thông qua HMaster, cụ thể:

- Cung cấp quyền admin, tính toán đến các Region Servers
- Gán các Regions cho Region Servers
- HMaster cũng đảm nhận nhiệm vụ cân bằng tải hoặc xử lý lỗi ở các node con trong 1 cụm
- Những thao tác liên quan đến metadata hoặc DDL tới cơ sở dữ liệu HBase

### 2. HBase Region Server hay HRegionServer

Nhận trực tiếp yêu cầu DML (read, write) từ Client mà không cần thông qua HMaster. Khi HRegionServer nhận yêu cầu từ người dùng, nó thực hiện các yêu cầu này cho các

Regions tương ứng, HRegionServer còn chứa HLog dùng để chứa mọi log files. Trong môi trường bao gồm nhiều cụm như Hadoop thì HRegionServers sẽ nằm trên các DataNode, HMaster sẽ liên lạc với HRegionServers khi có các thao tác sau:

- Quản lý các Regions
- Phân phối các Regions tự động
- Nhận các lệnh DML
- Liên lạc trực tiếp với client

### **3. HBase Region hay Region**

Regions chứa vô số các Stores, mỗi Store bao gồm 2 thành phần chính Memstore và Hfile. Memstore giống như một bộ nhớ cache, data đi vào đầu tiên sẽ nằm ở Memstore, được sắp xếp lại và cuối cùng là được đưa vào HFile, nếu ta sử dụng Apache HBase trên một hệ thống Hadoop cluster, các HFile này sẽ được lưu trữ vào trong Hadoop Distributed File System (HDFS)

### **4. HBase ZooKeeper hay ZooKeeper**

ZooKeeper là trung tâm điều khiển của HBase, nó mang nhiệm vụ quan trọng là duy trì những thông tin cấu hình, cung cấp cơ chế đồng bộ phân tán cho toàn cơ sở dữ liệu. Cơ chế đồng bộ phân tán là việc truy cập phân tán đến các cụm đang chạy với nhiệm vụ điều phối nhiệm vụ giữa các node một cách chính xác, tránh xảy ra lỗi. Nếu Client muốn giao tiếp với Regions thì phải thông qua ZooKeeper trước, cụ thể của ZooKeeper như sau:

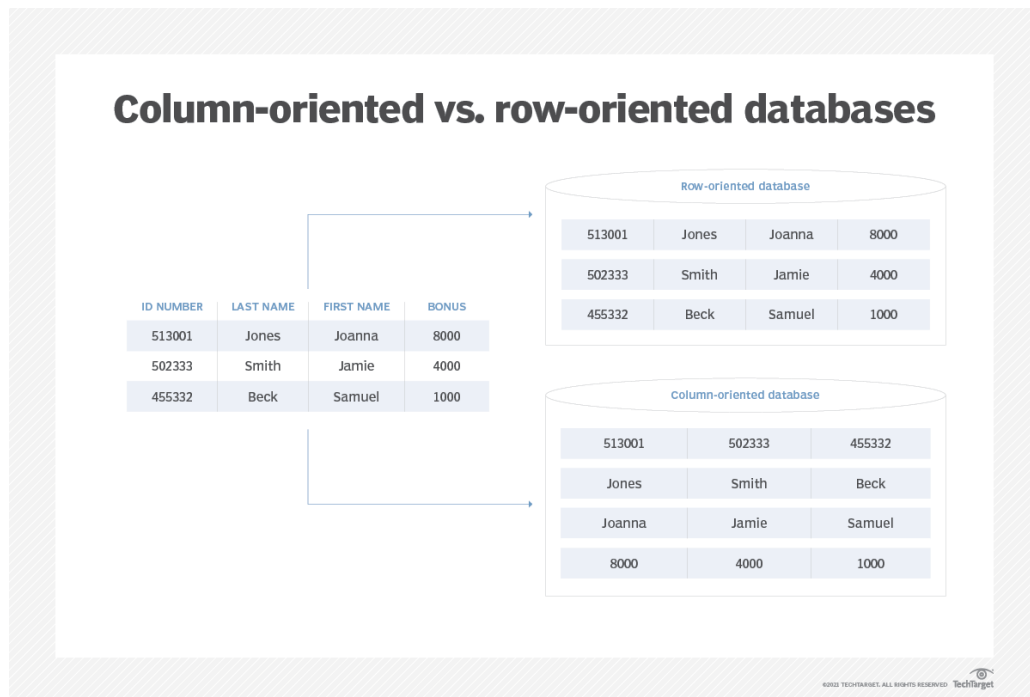
- Duy trì thông tin thiết lập
- Cung cấp cơ chế đồng bộ phân tán
- Thiết lập kết nối giữa Client với HRegionServers
- Kiểm tra liên tục các lỗi xảy ra với các cụm

## **2.4 Lưu trữ dữ liệu trong Apache HBase**

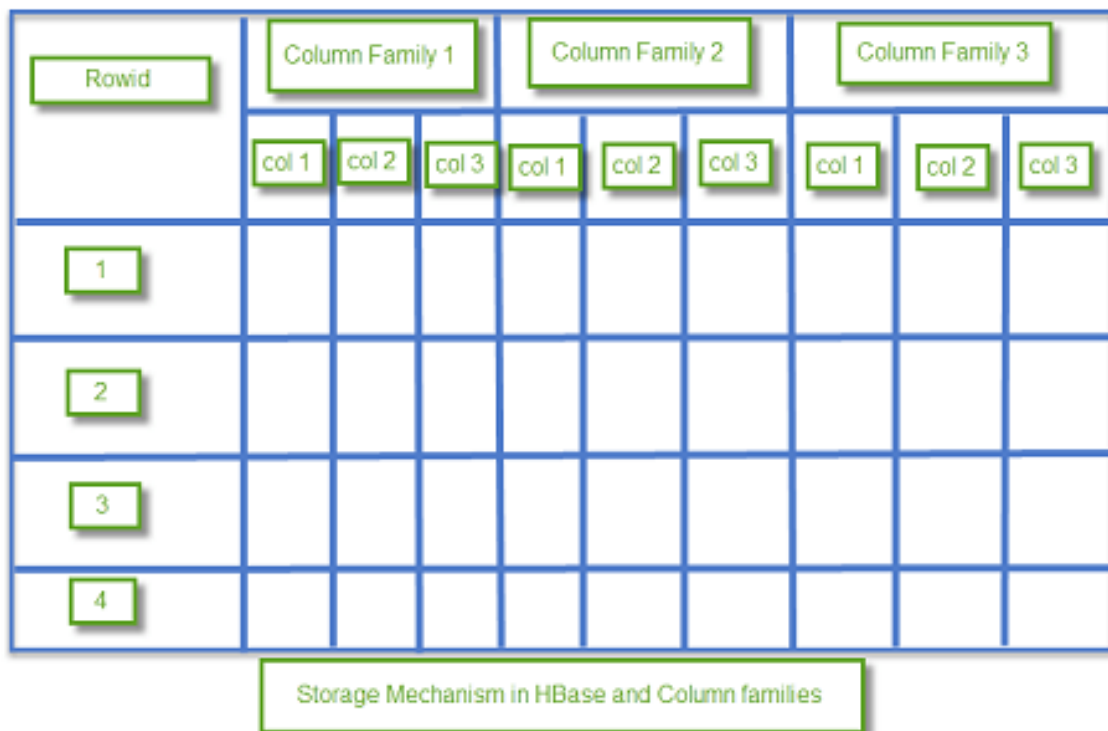
HBase lưu trữ dữ liệu theo mô hình wide column store hay còn gọi là column - oriented (một dạng của NoSQL database). Điểm khác nhau ở cách tiếp cận column – oriented và row – oriented đó là với mỗi điểm dữ liệu, row – oriented sẽ lưu trữ thông tin theo mỗi hàng, còn đối với column – oriented sẽ lưu trữ thông tin theo mỗi cột. Đây cũng là điểm



giúp columnar database có thể cải thiện hiệu năng so với các database truyền thống lưu trữ theo hàng bởi vì khi truy xuất đến columnar database, chỉ những cột cần thiết sẽ được lấy ra, và nó cũng mang đến khả năng mở rộng và xử lý một lượng lớn data



Trong HBase, các bảng được sắp xếp bởi RowID và là tập hợp của vô số các column families. Column Families trong lược đồ được thể hiện ở dạng key-value, mỗi column family gom nhóm nhiều columns khác có liên quan đến nhau. Column values sẽ được lưu trữ trong ổ đĩa



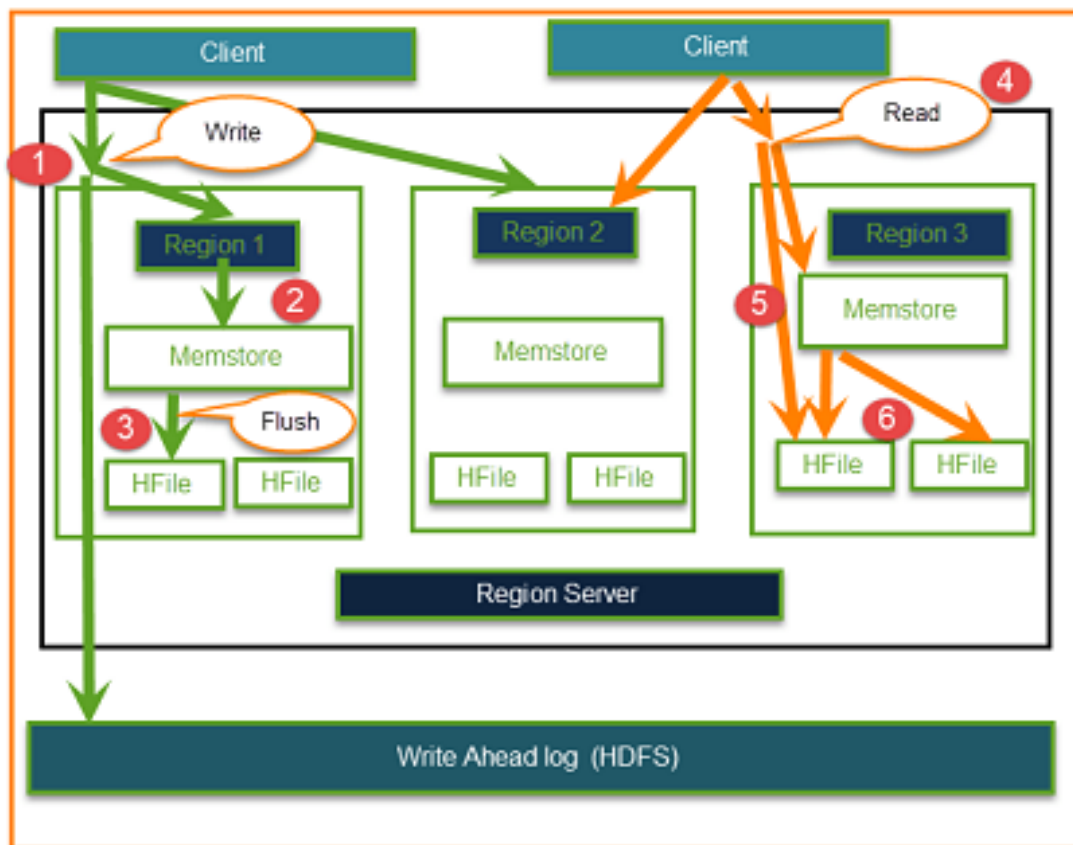
Các thành phần:

- Table: Tập hợp tất cả các rows
- Row: Tập hợp tất cả các column families
- Column Family: Tập hợp tất cả các columns
- Column hay Qualifier: Tập hợp các cặp key – value

HBase sẽ bao gồm tập các Table như trên với một số điểm chính sau:

- Mỗi table gồm nhiều rows
- Mỗi row được xác định bởi một RowID hay gọi là RowKey (tương đương primary key) duy nhất
- Mọi truy cập vào HBase đều phải sử dụng RowID
- Các row trong table luôn được sắp xếp theo thứ tự từ điển dựa vào RowID
- Mỗi row là tập hợp nhiều column hay qualifier khác nhau
- Những column (qualifier) liên quan đến nhau sẽ được gom lại thành một Column Family

## 2.5 Data Flow trong Apache HBase



### Write Operation

Bước 1: Client muốn write data phải tạo kết nối lần đầu tiên với Region Servers sau đó là Regions

Bước 2: Regions liên lạc với Memstore, tại đây data sẽ được sắp xếp lại bởi vì ta đã biết dữ liệu trong HBase luôn được sắp xếp theo RowID

Bước 3: Memstore đóng vai trò như cache trong máy tính, sau khi sắp xếp data, Memstore thực hiện đẩy data vào HFile và HFile này sẽ được lưu trữ trong HDFS

### Read Operation

Bước 4: Client yêu cầu đọc dữ liệu từ Regions

Bước 5: Lúc này Client có thể yêu cầu trực tiếp đến Memstore để truy vấn dữ liệu

Bước 6: Client truy cập data từ HFile và lấy data cần thiết

---

## CHƯƠNG 3: CẤU HÌNH VÀ CÀI ĐẶT

---

### 3.1 Yêu cầu

Các phiên bản sau được sử dụng xuyên suốt quá trình cài đặt:

7. Java 8
8. Hadoop 2.7.7
9. Apache ZooKeeper 3.6.3
10. Apache Hbase 1.3.5
11. VirtualBox 6.1.30
12. Ubuntu server 18.04

### 3.2 Cài đặt

Sử dụng VirtualBox tạo ra 3 máy ảo sử dụng Ubuntu server 18.04 bao gồm: Masternode, Slavenode1, Slavenode2

#### 3.2.1 Cài đặt Hadoop

1. Thực hiện lệnh sau (trên cả 3 máy):

```
$ sudo apt update
$ sudo apt install openjdk-8-jdk
```

2. Lấy đường dẫn java (trên cả 3 máy):

```
$ sudo update-alternatives --config java
```

3. Add đường dẫn java vào **/etc/environment** (trên cả 3 máy):

```
export JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre"
```

4. Thực hiện lệnh sau (trên cả 3 máy):

```
source /etc/environment
```

5. Thực hiện lệnh sau (trên masternode):

```
$ cd ~  
$ wget https://archive.apache.org/dist/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz
```

6. Thực hiện copy qua các slavenode (trên masternode):

```
$ scp hadoop-2.7.7.tar.gz slavenode1@thay_ip_slavenode1:/home/slavenode1  
$ scp hadoop-2.7.7.tar.gz slavenode2@thay_ip_slavenode2:/home/slavenode2
```

7. Giải nén file cài và move vào folder **/usr/local/hadoop** (trên cả 3 máy):

```
$ tar -zxvf hadoop-2.7.7.tar.gz  
$ sudo mv hadoop-2.7.7 /usr/local/hadoop
```

8. Add đường dẫn hadoop vào **/etc/environment** (trên cả 3 máy):

```
export HADOOP_HOME="/usr/local/hadoop"
```

9. Thực hiện lệnh sau (trên cả 3 máy):

```
source /etc/environment
```

10. Thực hiện chỉnh sửa **/etc/hosts** (trên cả 3 máy):

```
# comment lại những dòng liên quan tới 127.0.x.x  
# thêm những dòng sau đây  
  
ip_masternode masternode  
ip_slavenode1 slavenode1  
ip_slavenode2 slavenode2
```

11. Thực hiện thêm user hadoop bằng các lệnh sau (trên cả 3 máy):

```
# adduser hadoop  
# usermod -aG hadoop hadoop  
# chown hadoop:root -R /usr/local/hadoop  
# chmod g+rwX -R /usr/local/hadoop
```

12. Đăng nhập vào user hadoop và generate SSH key (trên masternode):

```
# su - hadoop
# ssh-keygen -t rsa
```

13. Copy key này sang các node slave (trên masternode):

```
# su - hadoop
$ ssh-copy-id hadoop@masternode
$ ssh-copy-id hadoop@slavenode1
$ ssh-copy-id hadoop@slavenode2
```

14. Add những dòng sau vào file ~/.bashrc (trên cả 3 máy):

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre
export PATH=$PATH:$JAVA_HOME/bin

export HADOOP_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_INSTALL=$HADOOP_HOME
```

15. Thực hiện lệnh sau (trên cả 3 máy):

```
source /etc/environment
```

16. Thực hiện cd vào folder Hadoop (trên masternode):

```
$ cd $HADOOP_HOME/etc/hadoop
```

17. Chỉnh sửa file **core-site.xml** (trên masternode):

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://masternode:9000</value>
  </property>
</configuration>
```

18. Chỉnh sửa file **hdfs-site.xml** (trên masternode):

```
<configuration>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/usr/local/hadoop/data/Namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/usr/local/hadoop/data/Datanode</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
</configuration>
```

19. Chỉnh sửa file **yarn-site.xml** (trên masternode):

```
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>masternode</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

20. Chỉnh sửa file **mapred-site.xml** (trên masternode):

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

21. Chỉnh sửa file **workers** (trên masternode):

```
slavenode1
slavenode2
```

22. Chỉnh sửa file **slaves** (trên masternode):

```
slavenode1
slavenode2
```

23. Thực hiện copy toàn bộ folder hadoop đã chỉnh sửa sang các slaves (trên masternode):

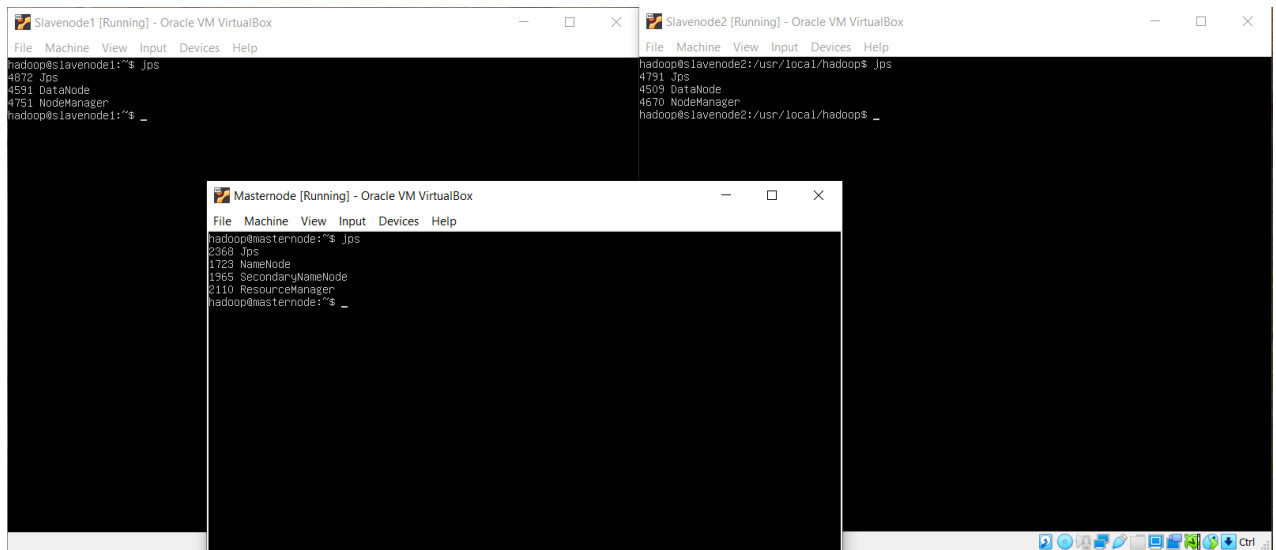
```
# scp /usr/local/hadoop/etc/hadoop/* slavenode1:/usr/local/hadoop/etc/hadoop/
# scp /usr/local/hadoop/etc/hadoop/* slavenode2:/usr/local/hadoop/etc/hadoop/
```

24. Thực hiện chạy hadoop (trên masternode):

```
$ cd ~
$ hadoop namenode -format
$ start-all.sh
```



## 25. Kiểm tra các tiến trình hoàn tất



### 3.2.2 Cài đặt ZooKeeper

1. Thực hiện lệnh sau (trên masternode):

```
$ cd ~  
$ wget https://dlcdn.apache.org/zookeeper/zookeeper-3.6.3/apache-zookeeper-3.6.3-bin.tar.gz
```

2. Thực hiện copy qua các slavenode (trên masternode):

```
$ scp apache-zookeeper-3.6.3-bin slavenode1@thay_ip_slavenode1:/home/slavenode1  
$ scp apache-zookeeper-3.6.3-bin slavenode2@thay_ip_slavenode2:/home/slavenode2
```

3. Giải nén file cài và move vào folder **/usr/local/zookeeper** (trên cả 3 máy):

```
$ tar -zxvf apache-zookeeper-3.6.3-bin.tar.gz  
$ sudo mv apache-zookeeper-3.6.3-bin /usr/local/zookeeper
```

4. Add đường dẫn hadoop vào **/etc/environment** (trên cả 3 máy):

```
$ export ZOOKEEPER_HOME="/usr/local/zookeeper"
```

5. Thực hiện lệnh sau (trên cả 3 máy):

```
source /etc/environment
```

6. Thực hiện chuỗi lệnh sau (trên cả 3 máy):

```
# chown hadoop:root -R /usr/local/zookeeper
# chmod g+rx -R /usr/local/zookeeper
# sudo su - hadoop
```

7. Add những dòng sau vào file ~/.bashrc (trên cả 3 máy):

```
export ZOOKEEPER_HOME=/usr/local/zookeeper
export PATH=$PATH:$ZOOKEEPER_HOME/bin
```

8. Thực hiện lệnh sau (trên cả 3 máy):

```
source /etc/environment
```

9. Thực hiện cd vào folder Zookeeper (trên cả 3 máy):

```
$ cd $ZOOKEEPER_HOME/conf
```

10. Thực hiện lệnh copy zoo\_sample.cfg (trên cả 3 máy):

```
$ cp zoo_sample.cfg zoo.cfg
```

11. Thực hiện chỉnh sửa file **zoo.cfg** (trên cả 3 máy), lưu ý, tại node nào thì node đó sẽ chỉnh sửa thành 0.0.0.0, ví dụ chỉnh sửa trên masternode = server.1:

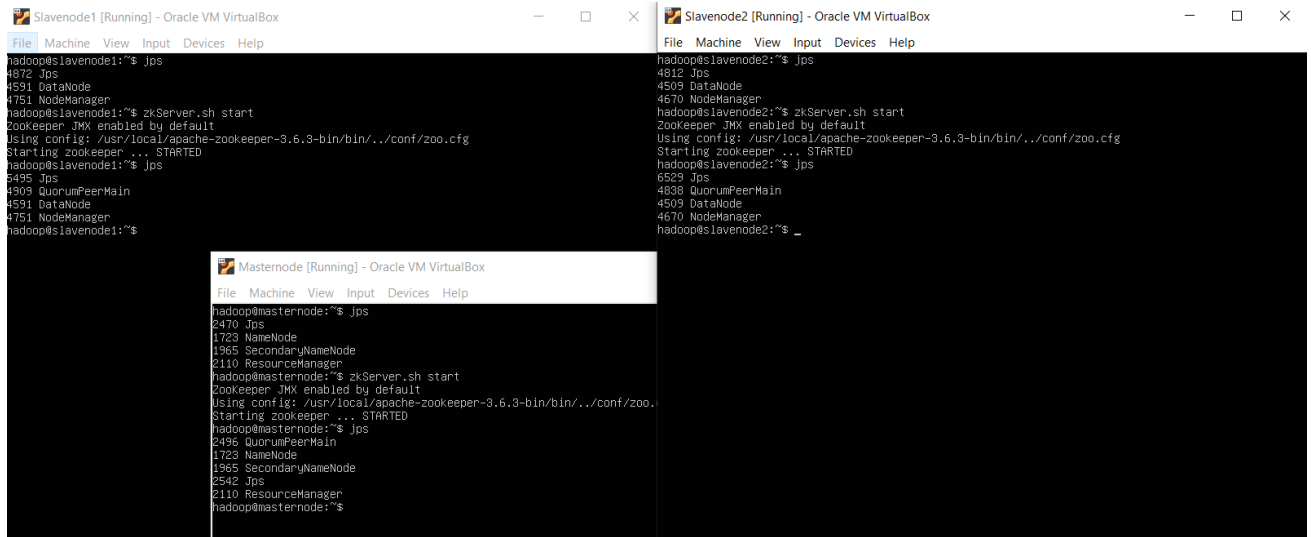
```
dataDir=/usr/local/zookeeper/data # nếu chưa có folder này phải thêm bước tạo folder data
```

```
server.1=0.0.0.0:2888:3888
server.2=slavenode1:2888:3888|
server.3=slavenode2:2888:3888
```

12. Chạy Zookeeper trên cả 3 máy

```
$ zkServer.sh start
```

### 13. Kiểm tra các tiến trình hoàn tất



### 3.2.3 Cài đặt HBase

1. Thực hiện lệnh sau (trên masternode):

```
$ cd ~  
$ wget https://archive.apache.org/dist/hbase/1.3.5/hbase-1.3.5-bin.tar.gz
```

2. Thực hiện copy qua các slavenode (trên masternode):

```
$ scp hbase-1.3.5-bin slavenode1@thay_ip_slavenode1:/home/slavenode1  
$ scp hbase-1.3.5-bin slavenode2@thay_ip_slavenode2:/home/slavenode2
```

3. Giải nén file cài và move vào folder **/usr/local/hbase** (trên cả 3 máy):

```
$ tar -zxvf hbase-1.3.5-bin.tar.gz  
$ sudo mv hbase-1.3.5-bin /usr/local/hbase
```

4. Add đường dẫn hadoop vào **/etc/environment** (trên cả 3 máy):

```
$ export HBASE_HOME="/usr/local/hbase"
```

5. Thực hiện lệnh sau (trên cả 3 máy):

```
source /etc/environment
```

6. Thực hiện chuỗi lệnh sau (trên cả 3 máy):

```
# chown hadoop:root -R /usr/local/hbase
# chmod g+rx -R /usr/local/hbase
# sudo su - hadoop
```

7. Add những dòng sau vào file **~/.bashrc** (trên cả 3 máy):

```
export HBASE_HOME=/usr/local/hbase
export PATH=$PATH:$HBASE_HOME/bin
```

8. Thực hiện lệnh sau (trên cả 3 máy):

```
source /etc/environment
```

9. Thực hiện cd vào folder Zookeeper (trên cả 3 máy):

```
$ cd $HBASE_HOME/conf
```

10. Chỉnh sửa file **hbase-env.sh** (trên cả 3 máy):

```
export JAVA_HOME=${JAVA_HOME}
export HBASE_PID_DIR=/usr/local/hbase/pids # nếu chưa có folder này thì phải tạo thêm
export HBASE_MANAGES_ZK=false
```

11. Chỉnh sửa file **hbase-site.xml** (trên máy masternode):

```
<property>
  <name>hbase.rootdir</name>
  <value>hdfs://masternode:9000/hbase</value>
</property>
<property>
  <name>hbase.master.info.port</name>
  <value>60010</value>
</property>
<property>
  <name>hbase.cluster.distributed</name>
  <value>true</value>
</property>
<property>
  <name>hbase.zookeeper.quorum</name>
  <value>masternode,slavenode1,slavenode2</value>
</property>
<property>
  <name>hbase.tmp.dir</name>
  <value>/usr/local/hbase-1.3.5/pids</value>
</property>
<property>
  <name>hbase.zookeeper.property.dataDir</name>
  <value>/usr/local/apache-zookeeper-3.6.3-bin</value>
</property>
```

12. Chỉnh sửa file **hbase-site.xml** (trên 2 máy slaves):

```
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>hdfs://masternode:9000/hbase</value>
  </property>
  <property>
    <name>hbase.master.info.port</name>
    <value>60010</value>
  </property>
  <property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
  </property>
</configuration>
```

13. Chỉnh sửa file **regionservers** (trên máy masternode):

```
slavenode1
slavenode2
```

#### 14. Chỉnh sửa file **regionservers** (trên máy slaves):

slavenode1:

```
slavenode1
```

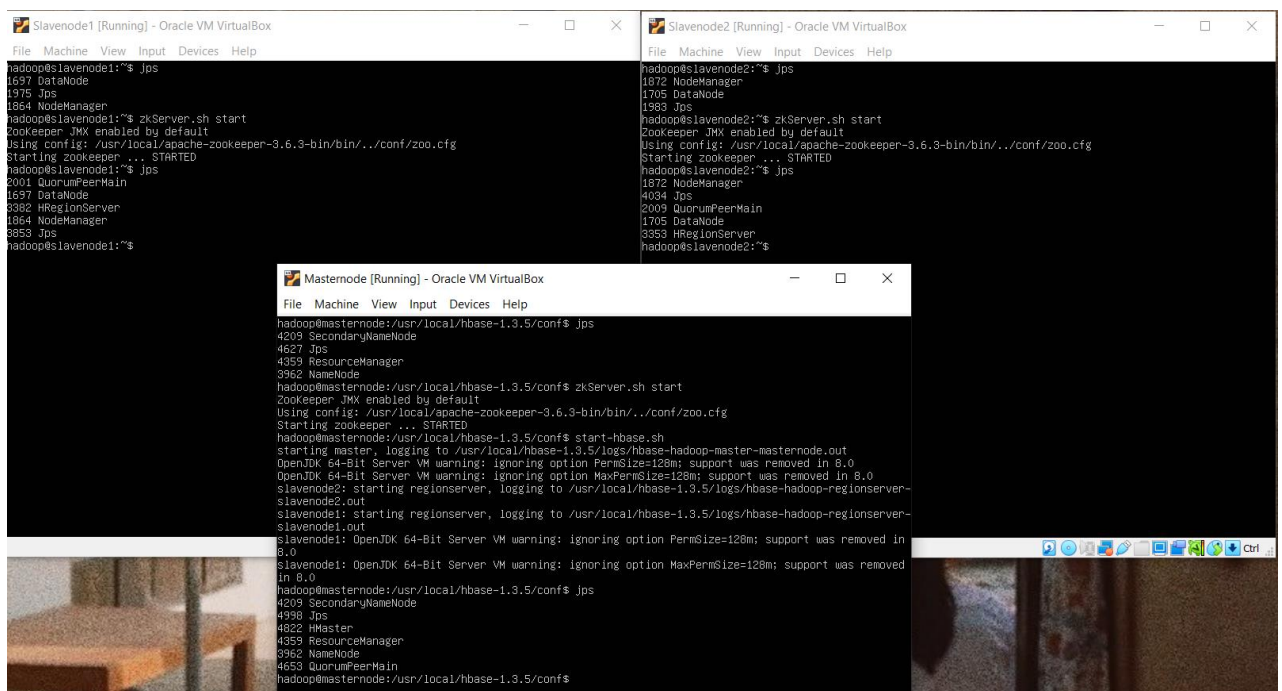
lavenode2:

```
slavenode2
```

#### 15. Khởi chạy Hbase (trên máy masternode):

```
$ start-hbase.sh
```

#### 16. Kiểm tra tiến trình hoàn tất



```
Slavenode1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
hadoop@slavenode1:~$ jps
1697 DataNode
1975 Jps
1684 NodeManager
hadoop@slavenode1:~$ zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /usr/local/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
hadoop@slavenode1:~$ jps
2001 QuorumPeerMain
1697 DataNode
3382 HRegionServer
1864 NodeManager
3853 Jps
hadoop@slavenode1:~$

Slavenode2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
hadoop@slavenode2:~$ jps
1872 NodeManager
1705 DataNode
1983 Jps
hadoop@slavenode2:~$ zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /usr/local/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
hadoop@slavenode2:~$ jps
1872 NodeManager
4034 Jps
2009 QuorumPeerMain
1705 DataNode
3353 HRegionServer
hadoop@slavenode2:~$

Masternode [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
hadoop@masternode:~$ jps
4209 SecondaryNameNode
4627 Jps
4359 ResourceManager
3962 NameNode
hadoop@masternode:~$ zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /usr/local/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
hadoop@masternode:~$ start-hbase.sh
starting master, logging to /usr/local/hbase-1.3.5/logs/hbase-hadoop-master-masternode.out
OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was removed in 8.0
slavenode2: starting regionserver, logging to /usr/local/hbase-1.3.5/logs/hbase-hadoop-regionserver-slavenode2.out
slavenode1: starting regionserver, logging to /usr/local/hbase-1.3.5/logs/hbase-hadoop-regionserver-slavenode1.out
slavenode1: OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
slavenode1: OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was removed in 8.0
hadoop@masternode:~$ jps
4209 SecondaryNameNode
4998 Jps
4822 HMaster
4359 ResourceManager
3962 NameNode
4653 QuorumPeerMain
hadoop@masternode:~$
```

## Region Servers

Base StatsMemoryRequestsStorefilesCompactions

ServerName	Start time	Version	Requests Per Second	Num. Regions
masternode,16020,1640589992354	Mon Dec 27 07:26:32 UTC 2021	1.3.5	0	0
slavenode1,16020,1640589988749	Mon Dec 27 07:26:28 UTC 2021	1.3.5	0	1
slavenode2,16020,1640589988596	Mon Dec 27 07:26:28 UTC 2021	1.3.5	0	1
Total:3			0	2

## Backup Masters

ServerName	Port	Start Time
Total:0		

## Tables

User TablesSystem TablesSnapshots

# CHƯƠNG 4: THỰC HÀNH

## Mô hình chung cơ sở dữ liệu

	RowKey	self_info	account_info	buy_history	trans_info
Row	1710	name: NHL age: 20 phone: 09238232312 education: Dai hoc	username: lovemilk pwd: 10000 date_regis: 10/05/2019	mnt_gold: 300 mnt_btc: 10000 mnt_vic: 4000	date_ord: 17/12/2021 location_ord: Sai Gon
Row	1711	name: LTT age: 20 phone: 01123232345 education: Trung hoc	username: hnaykovui pwd: ngaymai date_regis: 10/12/2018	mnt_btc: 60000 mnt_hag: 8000	date_ord: 18/12/2021 location_ord: Dak Lak
Row	1712	name: TNHN age: 10 phone: 08632738232 education: Tieu hoc	username: kevotam pwd: kokoailove date_regis: 03/06/2017	mnt_shirt: 40000	date_ord: 20/12/2021 location_ord: Dak Lak

## 4.1 Tạo Table và ColumnFamily

Lệnh tạo Table và Column Family:

```
create <table>, <ColumnFamily1>, <ColumnFamily2>...<ColumnFamilyN>
```

Thực hiện:

```
hbase(main):004:0> create 'marketing_analytics', 'self_info', 'account_info', 'buy_history', 'trans_info'
0 row(s) in 1.5000 seconds

=> Hbase::Table - marketing_analytics
hbase(main):005:0> list
TABLE
marketing_analytics
1 row(s) in 0.0140 seconds

=> ["marketing_analytics"]
hbase(main):006:0>
```

Ta có thể check thông tin về Table trên Web UI:

APACHEHBASE

[Home](#) [Table Details](#) [Procedures](#) [Local Logs](#) [Log Level](#) [Debug Dump](#) [Metrics Dump](#) [Profiler](#) [HBase Configuration](#)

Table marketing\_analytics

Table Attributes

Attribute Name	Value	Description
Enabled	true	Is the table enabled
Compaction	NONE	Is the table compacting

Table Regions

Name	Region Server	Start Key	End Key	Locality	Requests
marketing_analytics,,1640609009801.1b841582cbda97bf411d5df987561b6.	<a href="#">slavenode2:16030</a>			0.0	0

Regions by Region Server

Region Server	Region Count
<a href="#">slavenode2:16030</a>	1

## 4.2 Insert dữ liệu

Lệnh thêm dữ liệu:

```
put <table>, <rowkey>, <CF:Qualifiers>, <value>
```



Thực hiện:

```
hbase(main):002:0> put 'marketing_analytics', '1710', 'self_info:name', 'NHL'
0 row(s) in 0.2150 seconds

hbase(main):003:0> scan 'marketing_analytics'
ROW                                COLUMN+CELL
 1710                               column=self_info:name, timestamp=1640609438881, value=NHL
1 row(s) in 0.0330 seconds
```

Script insert toàn bộ dữ liệu sẽ được đính kèm trong file báo cáo, kết quả thực hiện:

```
ROW                                COLUMN+CELL
 1710                               column=account_info:date_regis, timestamp=1640611049531, value=10/05/2019
 1710                               column=account_info:pwd, timestamp=1640611029356, value=10000
 1710                               column=account_info:username, timestamp=1640611005620, value=lovemilk
 1710                               column=buy_history:mnt_btc, timestamp=1640611319263, value=10000
 1710                               column=buy_history:mnt_gold, timestamp=1640611304298, value=300
 1710                               column=buy_history:mnt_vic, timestamp=1640611331038, value=4000
 1710                               column=self_info:age, timestamp=1640610905656, value=20
 1710                               column=self_info:education, timestamp=1640610973208, value=Dai hoc
 1710                               column=self_info:name, timestamp=1640609438881, value=NHL
 1710                               column=self_info:phone, timestamp=1640610954439, value=09238232312
 1710                               column=trans_info:date_ord, timestamp=1640611357653, value=17/12/2021
 1710                               column=trans_info:location_ord, timestamp=1640611376131, value=Sai Gon
 1711                               column=account_info:date_regis, timestamp=1640611531837, value=10/12/2018
 1711                               column=account_info:pwd, timestamp=1640611514578, value=ngaymai
 1711                               column=account_info:username, timestamp=1640611499088, value=hnaykovui
 1711                               column=buy_history:mnt_btc, timestamp=1640611559858, value=60000
 1711                               column=buy_history:mnt_hag, timestamp=1640611575094, value=8000
 1711                               column=self_info:age, timestamp=1640611438264, value=20
 1711                               column=self_info:education, timestamp=1640611482548, value=Trung hoc
 1711                               column=self_info:name, timestamp=1640611425914, value=LTT
 1711                               column=self_info:phone, timestamp=1640611464058, value=01123232345
 1711                               column=trans_info:date_ord, timestamp=1640611593095, value=18/12/2021
 1711                               column=trans_info:location_ord, timestamp=1640611613684, value=Dak Lak
 1712                               column=account_info:date_regis, timestamp=1640611774928, value=03/06/2017
 1712                               column=account_info:pwd, timestamp=1640611752263, value=kocoailove
 1712                               column=account_info:username, timestamp=1640611732793, value=kevotam
 1712                               column=buy_history:mnt_shirt, timestamp=1640611801113, value=40000
 1712                               column=self_info:age, timestamp=1640611655495, value=10
 1712                               column=self_info:education, timestamp=1640611712070, value=Tieu hoc
 1712                               column=self_info:name, timestamp=1640611638397, value=TNHN
 1712                               column=self_info:phone, timestamp=1640611677608, value=08632738232
 1712                               column=trans_info:date_ord, timestamp=1640611820903, value=20/12/2021
 1712                               column=trans_info:location_ord, timestamp=1640611832690, value=Dak Lak
3 row(s) in 0.0600 seconds
```

### 4.3 Update dữ liệu

Lệnh cập nhật dữ liệu (nếu ở chế độ distributed, mọi node phải cùng thực hiện lệnh xóa):

```
put <table>, <rowkey>, <CF:Qualifiers>, <value>
```

Thực hiện:

```
hbase(main):003:0> scan 'marketing_analytics'
ROW                                COLUMN+CELL
 1710                               column=self_info:name, timestamp=1640609438881, value=NHL
1 row(s) in 0.0330 seconds

hbase(main):004:0> put 'marketing_analytics', '1710', 'self_info:name', 'HQQ'
0 row(s) in 0.0240 seconds

hbase(main):005:0> get 'marketing_analytics', '1710', {COLUMN => 'self_info:name'}
COLUMN                                CELL
 self_info:name                       timestamp=1640612129223, value=HQQ
1 row(s) in 0.0250 seconds
```

## 4.4 Delete dữ liệu

Lệnh xóa dữ liệu (nếu ở chế độ distributed, mọi node phải cùng thực hiện lệnh xóa):

**delete <table>, <rowkey>, <CF:Qualifiers>**

Thực hiện:

```
hbase(main):009:0> delete 'marketing_analytics', '1710', 'self_info:name'
0 row(s) in 0.0210 seconds

hbase(main):010:0> get 'marketing_analytics', '1710'
COLUMN                                CELL
account_info:date_regis               timestamp=1640611049531, value=10/05/2019
account_info:pwd                      timestamp=1640611029356, value=10000
account_info:username                 timestamp=1640611005620, value=lovemilk
buy_history:mnt_btc                   timestamp=1640611319263, value=10000
buy_history:mnt_gold                  timestamp=1640611304298, value=300
buy_history:mnt_vic                   timestamp=1640611331038, value=4000
self_info:age                         timestamp=1640610905656, value=20
self_info:education                   timestamp=1640610973208, value=Dai hoc
self_info:phone                       timestamp=1640610954439, value=09238232312
trans_info:date_ord                   timestamp=1640611357653, value=17/12/2021
trans_info:location_ord               timestamp=1640611376131, value=Sai Gon
1 row(s) in 0.0240 seconds
```

## 4.4 Truy xuất dữ liệu

### 4.4.1 Key Only Filter

Dùng để lấy ra ColumnFamily và Qualifier của các records

```
hbase(main):014:0> scan 'marketing_analytics', {FILTER => "KeyOnlyFilter()"}_
```

Kết quả:

```
ROW          COLUMN+CELL
1710         column=account_info:date_regis, timestamp=1640611049531, value=
1710         column=account_info:pwd, timestamp=1640611029356, value=
1710         column=account_info:username, timestamp=1640611005620, value=
1710         column=buy_history:mnt_btc, timestamp=1640611319263, value=
1710         column=buy_history:mnt_gold, timestamp=1640611304298, value=
1710         column=buy_history:mnt_vic, timestamp=1640611331038, value=
1710         column=self_info:age, timestamp=1640610905656, value=
1710         column=self_info:education, timestamp=1640610973208, value=
1710         column=self_info:name, timestamp=1640613077759, value=
1710         column=self_info:phone, timestamp=1640610954439, value=
1710         column=trans_info:date_ord, timestamp=1640611357653, value=
1710         column=trans_info:location_ord, timestamp=1640611376131, value=
1711         column=account_info:date_regis, timestamp=1640611531837, value=
1711         column=account_info:pwd, timestamp=1640611514578, value=
1711         column=account_info:username, timestamp=1640611499088, value=
1711         column=buy_history:mnt_btc, timestamp=1640611559858, value=
1711         column=buy_history:mnt_hag, timestamp=1640611575094, value=
1711         column=self_info:age, timestamp=1640611438264, value=
1711         column=self_info:education, timestamp=1640611482548, value=
1711         column=self_info:name, timestamp=1640611425914, value=
1711         column=self_info:phone, timestamp=1640611464058, value=
1711         column=trans_info:date_ord, timestamp=1640611593095, value=
1711         column=trans_info:location_ord, timestamp=1640611613684, value=
1712         column=account_info:date_regis, timestamp=1640611774928, value=
1712         column=account_info:pwd, timestamp=1640611752263, value=
1712         column=account_info:username, timestamp=1640611732793, value=
1712         column=buy_history:mnt_shirt, timestamp=1640611801113, value=
1712         column=self_info:age, timestamp=1640611655495, value=
1712         column=self_info:education, timestamp=1640611712070, value=
1712         column=self_info:name, timestamp=1640611638397, value=
1712         column=self_info:phone, timestamp=1640611677608, value=
1712         column=trans_info:date_ord, timestamp=1640611820903, value=
1712         column=trans_info:location_ord, timestamp=1640611832690, value=
3 row(s) in 0.1510 seconds
```

### 4.4.2 Prefix Filter

Dùng để filter các records dựa trên RowKey

```
hbase(main):015:0> scan 'marketing_analytics', {FILTER => "PrefixFilter('1710')"}
ROW          COLUMN+CELL
1710         column=account_info:date_regis, timestamp=1640611049531, value=10/05/2019
1710         column=account_info:pwd, timestamp=1640611029356, value=10000
1710         column=account_info:username, timestamp=1640611005620, value=lovemilk
1710         column=buy_history:mnt_btc, timestamp=1640611319263, value=10000
1710         column=buy_history:mnt_gold, timestamp=1640611304298, value=300
1710         column=buy_history:mnt_vic, timestamp=1640611331038, value=4000
1710         column=self_info:age, timestamp=1640610905656, value=20
1710         column=self_info:education, timestamp=1640610973208, value=Dai hoc
1710         column=self_info:name, timestamp=1640613077759, value=NLH
1710         column=self_info:phone, timestamp=1640610954439, value=09238232312
1710         column=trans_info:date_ord, timestamp=1640611357653, value=17/12/2021
1710         column=trans_info:location_ord, timestamp=1640611376131, value=Sai Gon
1 row(s) in 0.0240 seconds
```

### 4.4.3 Column Prefix Filter và Multiple Column Prefix Filter

Dùng để Filter các Qualifiers

```
hbase(main):016:0> scan 'marketing_analytics', {FILTER => "ColumnPrefixFilter('n')"}
ROW          COLUMN+CELL
 1710         column=self_info:name, timestamp=1640613077759, value=NHL
 1711         column=self_info:name, timestamp=1640611425914, value=LTT
 1712         column=self_info:name, timestamp=1640611638397, value=TNHN
3 row(s) in 0.0460 seconds
```

#### *Column Prefix Filter*

```
hbase(main):017:0> scan 'marketing_analytics', {FILTER => "MultipleColumnPrefixFilter('n','p')"}
ROW          COLUMN+CELL
 1710         column=account_info:pwd, timestamp=1640611029356, value=10000
 1710         column=self_info:name, timestamp=1640613077759, value=NHL
 1710         column=self_info:phone, timestamp=1640610954439, value=09238232312
 1711         column=account_info:pwd, timestamp=1640611514578, value=ngaymai
 1711         column=self_info:name, timestamp=1640611425914, value=LTT
 1711         column=self_info:phone, timestamp=1640611464058, value=01123232345
 1712         column=account_info:pwd, timestamp=1640611752263, value=kocailove
 1712         column=self_info:name, timestamp=1640611638397, value=TNHN
 1712         column=self_info:phone, timestamp=1640611677608, value=08632738232
3 row(s) in 0.0460 seconds
```

#### *Multiple Column Prefix Filter*

### 4.4.4 Value Filter

Dùng để Filter các records theo value

```
hbase(main):018:0> scan 'marketing_analytics', {COLUMNS => 'self_info:name',FILTER => "ValueFilter(=, 'binary:NHL')"}
ROW          COLUMN+CELL
 1710         column=self_info:name, timestamp=1640613077759, value=NHL
1 row(s) in 0.0680 seconds
```

Link video báo cáo:

<https://drive.google.com/file/d/18kUSoFx82vPHRnGyY1rb4m9EN9Ar9Y3F/view?usp=sharing>