UNIVERSITI MALAYA

WQD 7005:
DATA MINING

SUBMISSION OF:
MILESTONE 2 (GROUP)

STORE DATA INTO HIVE DATA WAREHOUSE

STUDENTS:
SITI AZIRA BINTI SUHOT (WQD170083)
MOHAMED FIRDAUS BIN SHAJAHAN (WQD170070)

LECTURER:
DR. TEH YING WAH

SUBMISSION DATE:
13TH OF OCTOBER 2019

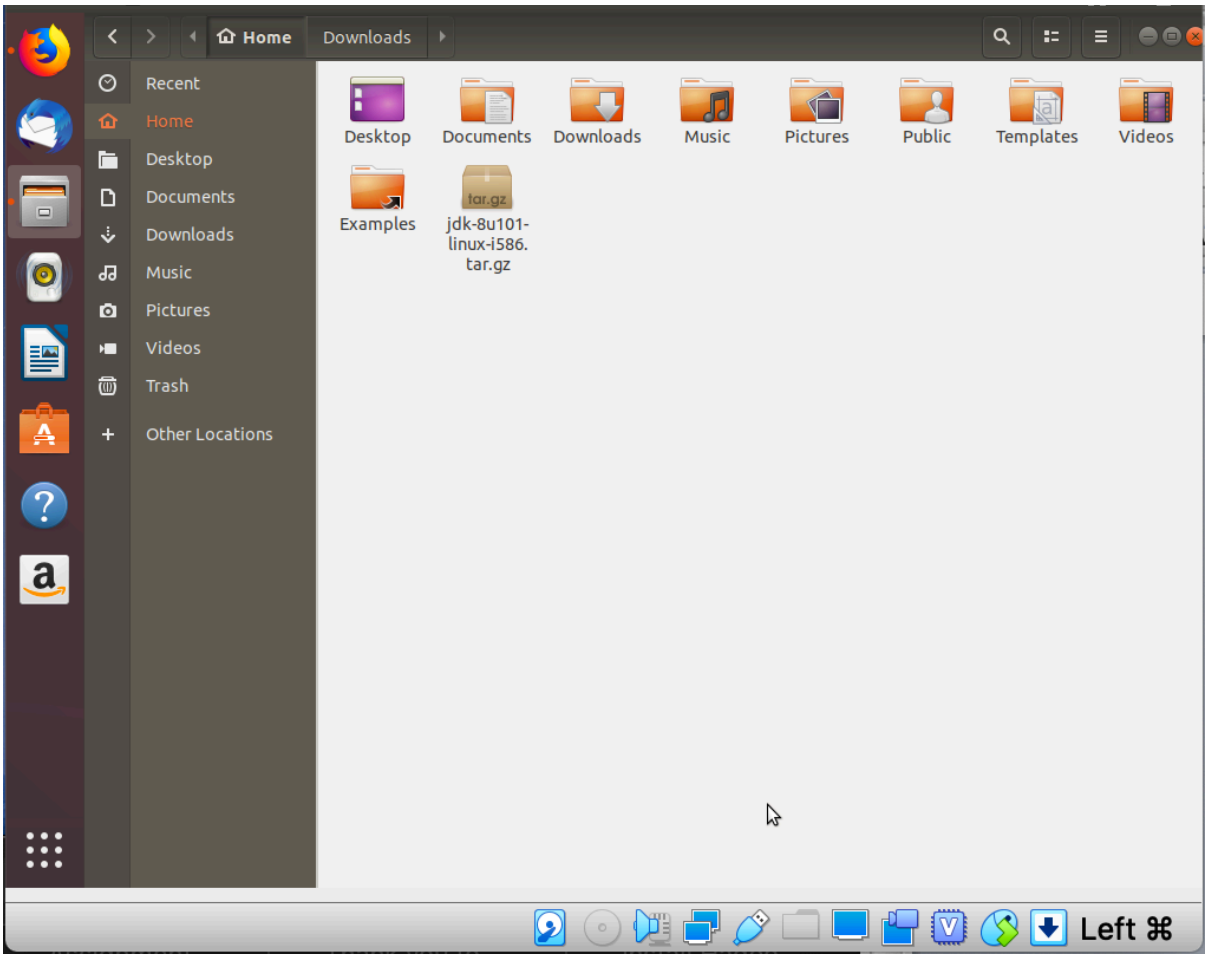
TABLE OF CONTENTS

1.0 Installation of Hive Data Warehouse

Step 1: Virtual Box has been installed in the notebook and ready for further installation; Java, Hadoop and Hive.

1.1 Java and Hadoop installation

Step 1: Download Java 8 package from https://drive.google.com/file/d/0BwIBPXSoIx_FTk1RN2IzMU9zMm8/view and save the file in home directory.



Step 2: Extract the Java Tar file.
Command: tar -xvf jdk-8u101-linux-i586.tar.gz

```
azira@azira-VirtualBox:~$ tar -xvf jdk-8u101-linux-i586.tar.gz
jdk1.8.0_101/
jdk1.8.0_101/THIRDPARTYLICENSEREADME.txt
jdk1.8.0_101/src.zip
jdk1.8.0_101/man/
jdk1.8.0_101/man/ja_JP.UTF-8/
jdk1.8.0_101/man/ja_JP.UTF-8/man1/
jdk1.8.0_101/man/ja_JP.UTF-8/man1/jjs.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/jstatd.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/javadoc.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/javaws.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/jrunscript.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/jvisualvm.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/extcheck.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/pack200.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/rmiregistry.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/unpack200.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/schemagen.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/appletviewer.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/jstat.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/wsgen.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/jmc.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/jdb.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/jsadebugd.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/javah.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/native2ascii.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/servertool.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/serialver.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/rmid.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/orbd.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/jps.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/xjc.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/jstack.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/javafxpackager.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/jinfo.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/idlj.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/jhat.1
jdk1.8.0_101/man/ja_JP.UTF-8/man1/jarsigner.1
```

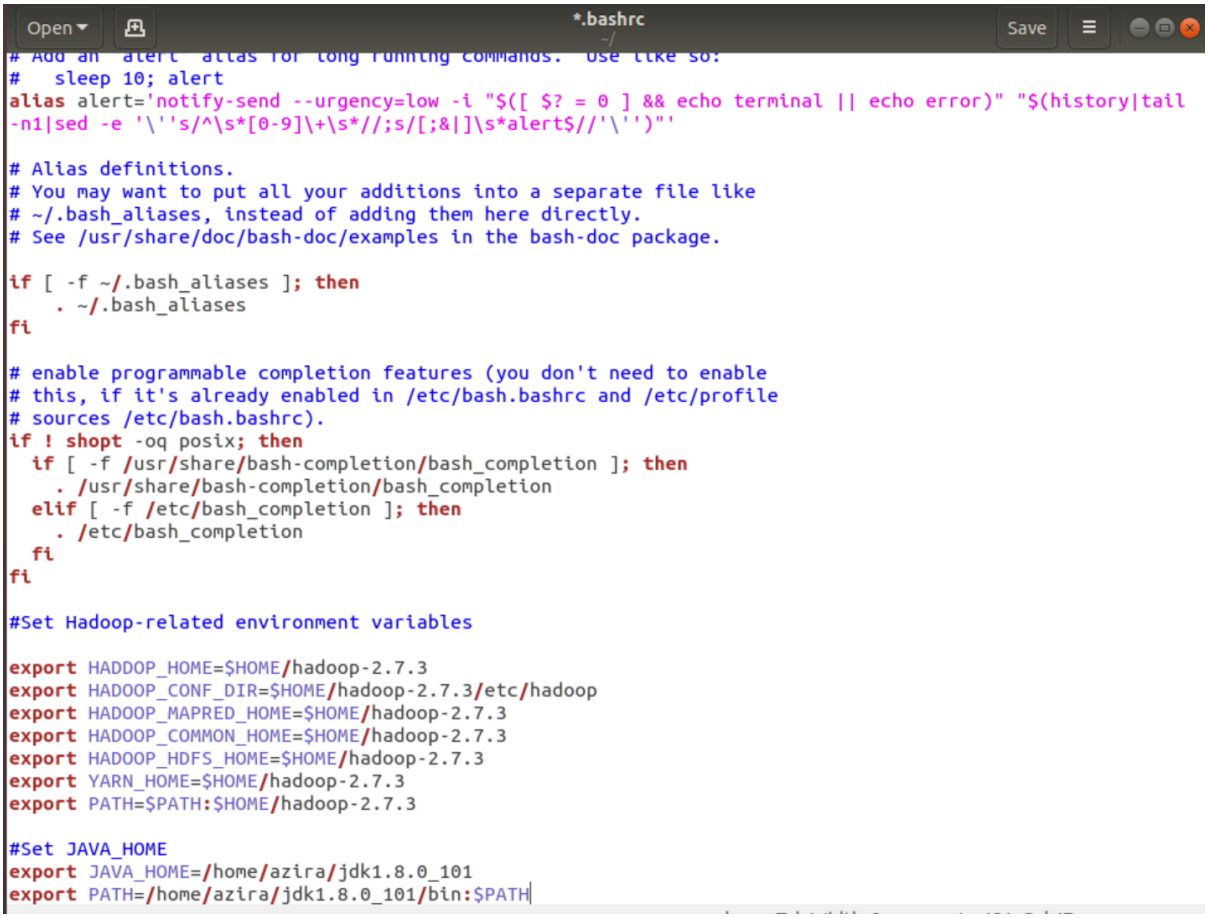
Step 3: Download Hadoop 2.7.3 package.
Command: wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz

```
azira@azira-VirtualBox:~$ wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar
.gz
--2019-10-06 17:31:51--  https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz
Resolving archive.apache.org (archive.apache.org)... 163.172.17.199
Connecting to archive.apache.org (archive.apache.org)|163.172.17.199|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 214092195 (204M) [application/x-gzip]
Saving to: ‘hadoop-2.7.3.tar.gz’

hadoop-2.7.3.tar.gz       100%[===================================>] 204.17M  2.81MB/s    in 1m 51s

2019-10-06 17:33:44 (1.84 MB/s) - ‘hadoop-2.7.3.tar.gz’ saved [214092195/214092195]
```

Step 4: Extract the Hadoop tar file.
Command: tar –xvf hadoop-2.7.3.tar.gz

```
azira@azira-VirtualBox:~$ tar -xvf hadoop-2.7.3.tar.gz
hadoop-2.7.3/
hadoop-2.7.3/bin/
hadoop-2.7.3/bin/hadoop
hadoop-2.7.3/bin/hadoop.cmd
hadoop-2.7.3/bin/rcc
hadoop-2.7.3/bin/hdfs
hadoop-2.7.3/bin/hdfs.cmd
hadoop-2.7.3/bin/container-executor
hadoop-2.7.3/bin/test-container-executor
hadoop-2.7.3/bin/yarn
hadoop-2.7.3/bin/yarn.cmd
```

Step 5: Add the Java and Hadoop paths in the bash file (.bashrc).
Step 5.1: Open .bashrc file.
Command: gedit .bashrc



Step 5.2: Save the bash file and close it.
Step 5.3: Save the changes to the current terminal.
Command: source .bashrc

```
azira@azira-VirtualBox:~$ source .bashrc
```

Step 6: Verify Java installation.
Command: java -version

```
azira@azira-VirtualBox:~$ java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) Client VM (build 25.101-b13, mixed mode)
```

Step 7: Verify Hadoop installation.
Command: hadoop version

```
azira@azira-VirtualBox:~$ hadoop version
Hadoop 2.7.3
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled by root on 2016-08-18T01:41Z
Compiled with protoc 2.5.0
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4
This command was run using /home/azira/hadoop-2.7.3/share/hadoop/common/hadoop-common-2.7.3.jar
```

Step 8: Edit the Hadoop configuration files to ensure all the Hadoop configuration files are located in hadoop-2.7.2/etc/hadoop directory.
Command: cd hadoop-2.7.3/etc/hadoop/
Command: ls

```
azira@azira-VirtualBox:~/hadoop-2.7.3/etc/hadoop$ ls
capacity-scheduler.xml      hadoop-policy.xml          kms-log4j.properties        ssl-client.xml.example
configuration.xsl           hdfs-site.xml              kms-site.xml                ssl-server.xml.example
container-executor.cfg      httpfs-env.sh              log4j.properties            yarn-env.cmd
core-site.xml               httpfs-log4j.properties    mapred-env.cmd              yarn-env.sh
hadoop-env.cmd              httpfs-signature.secret    mapred-env.sh               yarn-site.xml
hadoop-env.sh               httpfs-site.xml            mapred-queues.xml.template
hadoop-metrics2.properties  kms-acls.xml               mapred-site.xml.template
hadoop-metrics.properties   kms-env.sh                 slaves
```

Step 9: Open core-site.xml and edit the property inside configuration tag.
core-site.xml informs Hadoop daemon where NameNode runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS and MapReduce.
Command: gedit core-site.xml

```
azira@azira-VirtualBox:~/hadoop-2.7.3/etc/hadoop$ gedit core-site.xml
```

Open ▾ | *core-site.xml | ~/hadoop-2.7.3/etc/hadoop | Save

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

Step 10: Edit hdfs-site.xml and edit the property inside configuration tag.
hdfs-site.xml contains configuration settings of HDFS daemons (i.e. NameNode, DataNode, Secondary NameNode). It also includes the replication factor and block size of HDFS.
Command: gedit hdfs-site.xml

```
azira@azira-VirtualBox:~/hadoop-2.7.3/etc/hadoop$ gedit hdfs-site.xml
```

Open ▾ *hdfs-site.xml ~/hadoop-2.7.3/etc/hadoop Save

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.permission</name>
<value>false</value>
</property>
</configuration>
```

Step 11: Edit mapred-site.xml file and edit the property inside configuration tag.
mapred-site.xml contains configuration settings of MapReduce application like number of JVM that can run in parallel, the size of mapper and the reducer process, CPU cores available for a process, etc.
In some cases, mapred-site.xml is not available. Hence, need to create the mapred-site.xml file using mapred.site.xml template.
Command: cp mapred-site.xml.template mapred-site.xml
Command: gedit mapred-site.xml

```
azira@azira-VirtualBox:~/hadoop-2.7.3/etc/hadoop$ cp mapred-site.xml.template mapred-site.xml
azira@azira-VirtualBox:~/hadoop-2.7.3/etc/hadoop$ gedit mapred-site.xml
```

Open ▾ *mapred-site.xml ~/hadoop-2.7.3/etc/hadoop Save

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

Step 12: Edit yarn-site.xml file and edit the property inside configuration tag.
yarn-site.xml contains configuration settings of ResourceManager and NodeManager like application memory management size, the operation needed on program and algorithm, etc.
Command: gedit yarn-site.xml

```
azira@azira-VirtualBox:~/hadoop-2.7.3/etc/hadoop$ gedit yarn-site.xml
```

Open ▾ *yarn-site.xml ~/hadoop-2.7.3/etc/hadoop Save

```
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<!-- Site specific YARN configuration properties -->

<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>
```

Step 13: Edit hadoop-env.sh and add the Java path.
Hadoop-env.sh contains the environment variables that are used in the script to run Hadoop like Java home path, etc.
Command: gedit hadoop-env.sh

```
azira@azira-VirtualBox:~/hadoop-2.7.3/etc/hadoop$ gedit hadoop-env.sh
```

Open ▾ *hadoop-env.sh ~/hadoop-2.7.3/etc/hadoop Save

```
# On secure datanodes, user to run the datanode as after dropping privileges.
# This **MUST** be uncommented to enable secure HDFS if using privileged ports
# to provide authentication of data transfer protocol.  This **MUST NOT** be
# defined if SASL is configured for authentication of data transfer protocol
# using non-privileged ports.
export HADOOP_SECURE_DN_USER=${HADOOP_SECURE_DN_USER}

# Where log files are stored.  $HADOOP_HOME/logs by default.
#export HADOOP_LOG_DIR=${HADOOP_LOG_DIR}/$USER

# Where log files are stored in the secure data environment.
export HADOOP_SECURE_DN_LOG_DIR=${HADOOP_LOG_DIR}/${HADOOP_HDFS_USER}

###
# HDFS Mover specific parameters
###
# Specify the JVM options to be used when starting the HDFS Mover.
# These options will be appended to the options specified as HADOOP_OPTS
# and therefore may override any similar flags set in HADOOP_OPTS
#
# export HADOOP_MOVER_OPTS=""

###
# Advanced Users Only!
###

# The directory where pid files are stored. /tmp by default.
# NOTE: this should be set to a directory that can only be written to by
#       the user that will run the hadoop daemons.  Otherwise there is the
#       potential for a symlink attack.
export HADOOP_PID_DIR=${HADOOP_PID_DIR}
export HADOOP_SECURE_DN_PID_DIR=${HADOOP_PID_DIR}

# A string representing this instance of hadoop. $USER by default.
export HADOOP_IDENT_STRING=$USER

#The Java implementation to use
export JAVA_HOME=/home/azira/jdk1.8.0_101
```

Step 14: Go to Hadoop home directory and format the NameNode.
Command: cd
Command: cd hadoop-2.7.3
Command: bin/hadoop namenode -format

```
azira@azira-VirtualBox:~/hadoop-2.7.3/etc/hadoop$ cd
azira@azira-VirtualBox:~$ cd hadoop-2.7.3
azira@azira-VirtualBox:~/hadoop-2.7.3$ bin/hadoop namenode -format
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

19/10/06 19:12:33 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = azira-VirtualBox/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.7.3
STARTUP_MSG:   classpath = /home/azira/hadoop-2.7.3/etc/hadoop:/home/azira/hadoop-2.7.3/share/hadoop/com
mon/lib/commons-cli-1.2.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/lib/netty-3.6.2.Final.jar:/home
/azira/hadoop-2.7.3/share/hadoop/common/lib/commons-beanutils-1.7.0.jar:/home/azira/hadoop-2.7.3/share/h
adoop/common/lib/jackson-core-asl-1.9.13.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/lib/commons-co
dec-1.4.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/lib/commons-digester-1.8.jar:/home/azira/hadoop
-2.7.3/share/hadoop/common/lib/commons-lang-2.6.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/lib/jac
kson-mapper-asl-1.9.13.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/lib/jsch-0.1.42.jar:/home/azira/
hadoop-2.7.3/share/hadoop/common/lib/commons-net-3.1.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/li
b/apacheds-kerberos-codec-2.0.0-M15.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/lib/slf4j-log4j12-1
.7.10.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/lib/jersey-server-1.9.jar:/home/azira/hadoop-2.7.
3/share/hadoop/common/lib/curator-framework-2.7.1.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/lib/j
ava-xmlbuilder-0.4.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/lib/stax-api-1.0-2.jar:/home/azira/h
adoop-2.7.3/share/hadoop/common/lib/commons-io-2.4.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/lib/
commons-collections-3.2.2.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/lib/guava-11.0.2.jar:/home/az
ira/hadoop-2.7.3/share/hadoop/common/lib/jsp-api-2.1.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/li
b/httpclient-4.2.5.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/lib/jettison-1.1.jar:/home/azira/had
oop-2.7.3/share/hadoop/common/lib/junit-4.11.jar:/home/azira/hadoop-2.7.3/share/hadoop/common/lib/common
```

This formats the HDFS via NameNode. This command is only executed for the first time. Formatting the file system means initializing the directory specified by the dfs.name.dir variable.

Step 15: Start NameNode. The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files stored in the HDFS and tracks all the file stored across the cluster.
Command: cd sbin/
Command: ./hadoop-daemon.sh start namenode

```
azira@azira-VirtualBox:~/hadoop-2.7.3$ cd sbin/
azira@azira-VirtualBox:~/hadoop-2.7.3/sbin$ /hadoop-daemon.sh start namenode
bash: /hadoop-daemon.sh: No such file or directory
azira@azira-VirtualBox:~/hadoop-2.7.3/sbin$ ./hadoop-daemon.sh start namenode
starting namenode, logging to /home/azira/hadoop-2.7.3/logs/hadoop-azira-namenode-azira-VirtualBox.out
azira@azira-VirtualBox:~/hadoop-2.7.3/sbin$
```

Step 16: Start DataNode. A DataNode connects to the NameNode and it responds to the request from the NameNode for different operations.
Command: ./hadoop-daemon.sh start datanode

```
azira@azira-VirtualBox:~/hadoop-2.7.3/sbin$ ./hadoop-daemon.sh start datanode
starting datanode, logging to /home/azira/hadoop-2.7.3/logs/hadoop-azira-datanode-azira-VirtualBox.out
azira@azira-VirtualBox:~/hadoop-2.7.3/sbin$ jps
2163 DataNode
2243 Jps
2068 NameNode
```

Step 17: Start ResourceManager. ResourceManager is the master that arbitrates all the available cluster resources and thus helps in managing the distributed applications running on the YARN system. Its work is to manage each NodeManagers and each application's ApplicationMaster.
Command: ./yarn-daemon.sh start resourcemanager

```
azira@azira-VirtualBox:~/hadoop-2.7.3/sbin$ ./yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /home/azira/hadoop-2.7.3/logs/yarn-azira-resourcemanager-azira-Virt
ualBox.out
Java HotSpot(TM) Client VM warning: You have loaded library /home/azira/hadoop-2.7.3/lib/native/libhadoo
p.so.1.0.0 which might have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noex
ecstack'.
azira@azira-VirtualBox:~/hadoop-2.7.3/sbin$ jps
2272 ResourceManager
2163 DataNode
2068 NameNode
2492 Jps
```

Step 18: Start NodeManager. The NodeManager in each machine framework is the agent which is responsible for managing containers, monitoring their resource usage and reporting the same to the ResourceManager.
Command: ./yarn-daemon.sh start nodemanager

```
azira@azira-VirtualBox:~/hadoop-2.7.3/sbin$ ./yarn-daemon.sh start nodemanager
starting nodemanager, logging to /home/azira/hadoop-2.7.3/logs/yarn-azira-nodemanager-azira-VirtualBox.o
ut
azira@azira-VirtualBox:~/hadoop-2.7.3/sbin$ jps
2272 ResourceManager
2163 DataNode
2627 Jps
2068 NameNode
2521 NodeManager
```

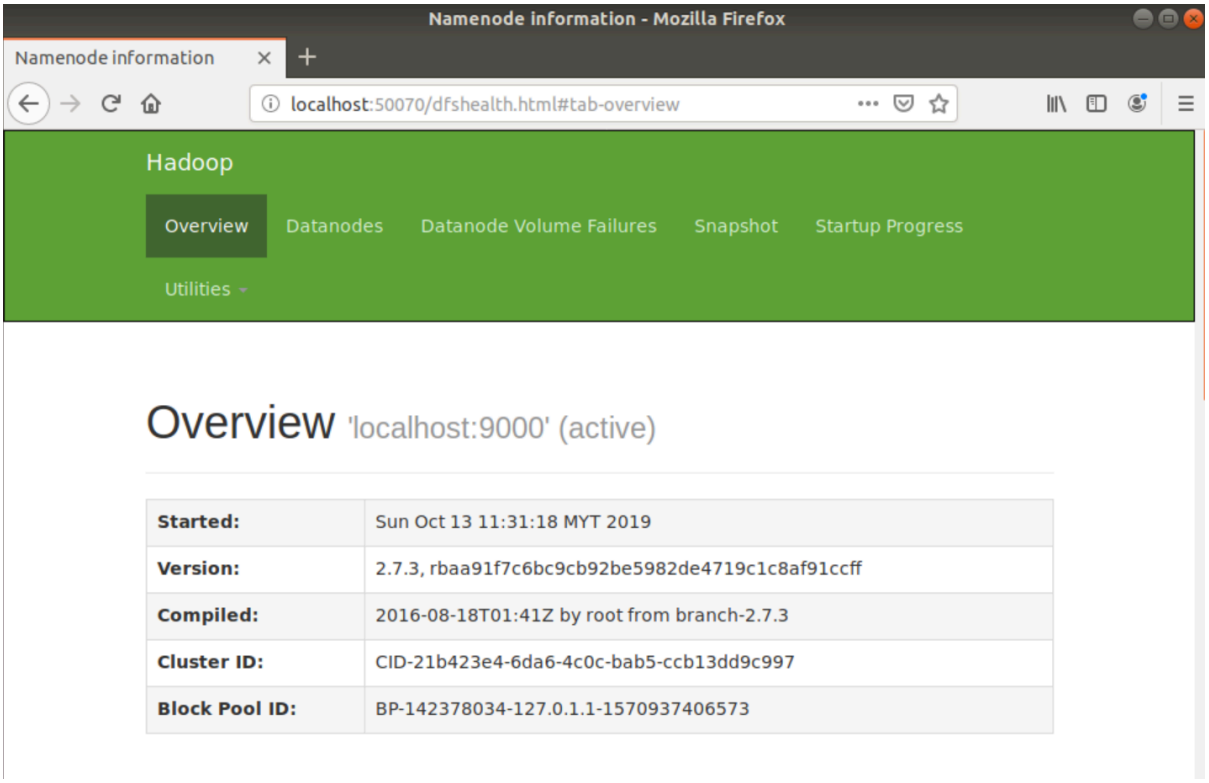
Step 19: Start JobHistoryServer. JobHistoyServer is responsible for servicing all job history related requests from client.
Command: ./mr-jobhistory-daemon.sh start historyserver

```
azira@azira-VirtualBox:~/hadoop-2.7.3/sbin$ ./mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/azira/hadoop-2.7.3/logs/mapred-azira-historyserver-azira-Virtua
lBox.out
Java HotSpot(TM) Client VM warning: You have loaded library /home/azira/hadoop-2.7.3/lib/native/libhadoo
p.so.1.0.0 which might have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noex
ecstack'.
azira@azira-VirtualBox:~/hadoop-2.7.3/sbin$ jps
2272 ResourceManager
2163 DataNode
2068 NameNode
2665 JobHistoryServer
2521 NodeManager
2716 Jps
```

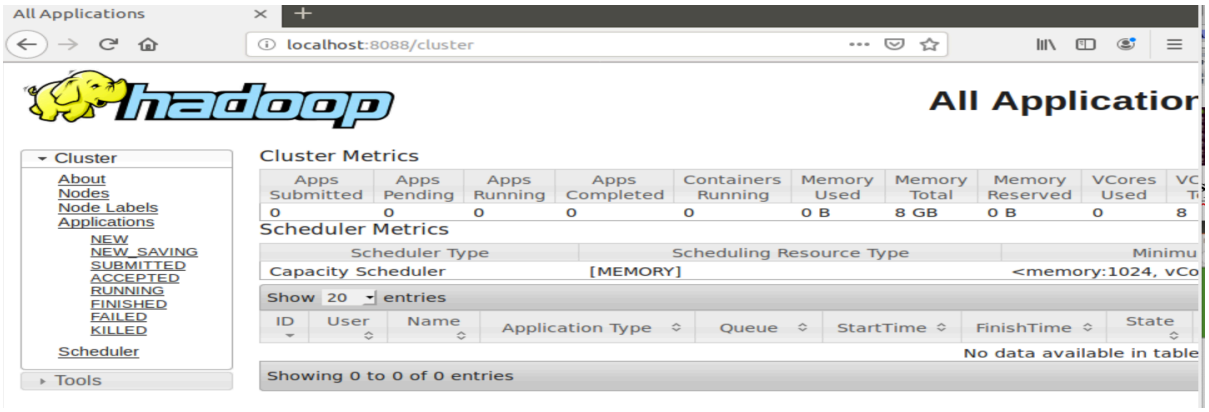
Step 20: Make sure that all the Hadoop services are up and running.
Command: jps

```
azira@azira-VirtualBox:~/hadoop-2.7.3/sbin$ jps
2272 ResourceManager
2163 DataNode
2068 NameNode
2665 JobHistoryServer
2521 NodeManager
2746 Jps
```

Step 21: Open the Mozilla Firefox browser and go to localhost:50070/dfshealth.html to get Hadoop services on browser.



Step 22: Open the Mozilla Firefox browser and go to localhost:8088 to verify all applications for cluster.



1.2 Hive installation

Step 1: Verifying Java and Hadoop installation
Command: java –version
Command: hadoop version

```
azira@azira-VirtualBox:~$ java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) Client VM (build 25.101-b13, mixed mode)
azira@azira-VirtualBox:~$ hadoop version
Hadoop 2.7.3
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb92be5982de4719c1c8af91ccf
f
Compiled by root on 2016-08-18T01:41Z
Compiled with protoc 2.5.0
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4
This command was run using /home/azira/hadoop-2.7.3/share/hadoop/common/hadoop-common-2.7.3.jar
```

Step 1: Download Hive tar.
Command: wget http://archive.apache.org/dist/hive/hive-2.1.0/apache-hive-2.1.0-bin.tar.gz

```
azira@azira-VirtualBox:~$ wget http://archive.apache.org/dist/hive/hive-2.1.0/apache-hive-2.1.0-bin.tar.
gz
--2019-10-06 19:37:39--  http://archive.apache.org/dist/hive/hive-2.1.0/apache-hive-2.1.0-bin.tar.gz
Resolving archive.apache.org (archive.apache.org)... 163.172.17.199
Connecting to archive.apache.org (archive.apache.org)|163.172.17.199|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 149599799 (143M) [application/x-gzip]
Saving to: 'apache-hive-2.1.0-bin.tar.gz'

apache-hive-2.1.0-bin.tar 100%[===================================>] 142.67M  3.60MB/s    in 52s

2019-10-06 19:38:32 (2.72 MB/s) - 'apache-hive-2.1.0-bin.tar.gz' saved [149599799/149599799]
```

Step 2: Extract the tar file.
Command: tar –xzf apache-hive-2.1.0-bin.tar.gz
Command: ls

```
azira@azira-VirtualBox:~$ tar -xzf apache-hive-2.1.0-bin.tar.gz
azira@azira-VirtualBox:~$ ls
apache-hive-2.1.0-bin      Downloads            jdk1.8.0_101               Public
apache-hive-2.1.0-bin.tar.gz  examples.desktop  jdk-8u101-linux-i586.tar.gz  Templates
Desktop                    hadoop-2.7.3         Music                      Videos
Documents                  hadoop-2.7.3.tar.gz  Pictures
```

Step 3: Edit the .bashrc file to update the environment variables for user
Command: gedit .bashrc

```
#Set Hadoop-related environment variables
export HADOOP_HOME=$HOME/hadoop-2.7.3
export HADOOP_CONF_DIR=$HOME/hadoop-2.7.3/etc/hadoop
export HADOOP_MAPRED_HOME=$HOME/hadoop-2.7.3
export HADOOP_COMMON_HOME=$HOME/hadoop-2.7.3
export HADOOP_HDFS_HOME=$HOME/hadoop-2.7.3
export YARN_HOME=$HOME/hadoop-2.7.3
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"

#Set JAVA_HOME
export JAVA_HOME=/home/azira/jdk1.8.0_101
export PATH=/home/azira/jdk1.8.0_101/bin:$PATH

#Add Hadoop bin/ directory to PATH
export PATH=$PATH:/home/azira/hadoop-2.7.3/bin
export HADOOP_PID_DIR=/home/azira/hadoop-2.7.3/hadoop2_data/hdfs/pid

#Set HIVE_HOME
export HIVE_HOME=/home/azira/apache-hive-2.1.0-bin
export HIVE_CONF_DIR=/home/azira/apache-hive-2.1.0-bin/conf
export PATH=$HIVE_HOME/bin:$PATH
export CLASSPATH=$CLASSPATH:/home/azira/hadoop/lib/*:.
export CLASSPATH=$CLASSPATH:/home/azira/apache-hive-2.1.0-bin/lib/*:.
```

Step 4: Save the changes.
Command: source. Bashrc

Step 5: Create Hive directories within HDFS. The directory 'warehouse' is the location to store the table or data related to hive. Verify it. The directory warehouse is the location to store the table or data related to hive and the temporary directory tmp is the temporary location to store the intermediate result of processing.
Command: hdfs dfs –mkdir –p /user/hive/warehouse
Command: hdfs dfs –mkdir /tmp

Step 6: Set read/write permission for table
Command: hdfs dfs –chmod g+w /user/hive/warehouse
Command: hdfs dfs –chmod g+w /tmp

Hive installation is completed successfully. The next step is to acquire an external database server to configure Metastore using Apache Derby database.

1.3 Apache Derby installation

Step 1: Downloading Apache Derby
Command: wget http://archive.apache.org/dist/db/derby/db-derby-10.4.2.0/db-derby-10.4.2.0-bin.tar.gz
Command: ls

```
azira@azira-VirtualBox:~$ wget http://archive.apache.org/dist/db/derby/db-derby-10.4.2.0/db-derby-10.4.2
.0-bin.tar.gz
--2019-10-13 19:45:30--  http://archive.apache.org/dist/db/derby/db-derby-10.4.2.0/db-derby-10.4.2.0-bin
.tar.gz
Resolving archive.apache.org (archive.apache.org)... 163.172.17.199
Connecting to archive.apache.org (archive.apache.org)|163.172.17.199|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18692878 (18M) [application/x-gzip]
Saving to: 'db-derby-10.4.2.0-bin.tar.gz'

db-derby-10.4.2.0-bin.tar 100%[===================================>]  17.83M   244KB/s    in 24s

2019-10-13 19:45:55 (768 KB/s) - 'db-derby-10.4.2.0-bin.tar.gz' saved [18692878/18692878]

azira@azira-VirtualBox:~$ ls
apache-hive-2.1.0-bin          Documents         hadoop-2.7.3.tar.gz          Music      Videos
apache-hive-2.1.0-bin.tar.gz   Downloads         jdk1.8.0_101                 Pictures
db-derby-10.4.2.0-bin.tar.gz   examples.desktop  jdk-8u101-linux-i586.tar.gz  Public
Desktop                        hadoop-2.7.3      metastore_db                 Templates
```

Step 2: Extract and verify the Derby archive
Command: tar zxvf db-derby-10.4.2.0-bin.tar.gz
Command: ls

```
azira@azira-VirtualBox:~$ ls
apache-hive-2.1.0-bin          Desktop           hadoop-2.7.3                 metastore_db  Templates
apache-hive-2.1.0-bin.tar.gz   Documents         hadoop-2.7.3.tar.gz          Music         Videos
db-derby-10.4.2.0-bin          Downloads         jdk1.8.0_101                 Pictures
db-derby-10.4.2.0-bin.tar.gz   examples.desktop  jdk-8u101-linux-i586.tar.gz  Public
```

Step 3: Setting up environment for Derby and save the changes.
Command: gedit .bashrc
Command: source .bashrc

```
#Set DERBY_HOME
export DERBY_HOME=/home/azira/derby
export PATH=$PATH:$DERBY_HOME/bin
export CLASSPATH=$CLASSPATH:$DERBY_HOME/lib/derby.jar:$DERBY_HOME/lib/derbytools.jar
```

Step 4: Create a directory to store Metastore
Command: mkdir $DERBY_HOME/data

```
azira@azira-VirtualBox:~$ mkdir $DERBY_HOME/data
```

Apache Derby installation and environmental setup is completed.

1.4 Configuring Metastore of Hive

Step 1: Configuring Hive. To configure Hive with Hadoop, edit the hive-env.sh file which is placed in the $HIVE_HOME/conf directory.
Command: cd $HIVE_HOME/conf
Command: cp hive-env.sh.template hive-env.sh
Command: gedit hive-site.xml

```
azira@azira-VirtualBox:~$ cd $HIVE_HOME/conf
azira@azira-VirtualBox:~/apache-hive-2.1.0-bin/conf$ cp hive-default.xml.template hive-site.xml
azira@azira-VirtualBox:~/apache-hive-2.1.0-bin/conf$ gedit hive-site.xml
```

```
<property>
   <name>javax.jdo.option.ConnectionURL</name>
   <value>jdbc:derby://localhost:1527/metastore_db;create=true </value>
   <description>JDBC connect string for a JDBC metastore </description>
</property>
```

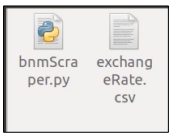
Step 2: Verifying Hive installation
Command: cd $HIVE_HOME
Command: BIN/HIVE

```
azira@azira-VirtualBox:~$ cd $HIVE_HOME
azira@azira-VirtualBox:~/apache-hive-2.1.0-bin$ bin/hive
```

1.5 Insert Data on Hive

Prepare csv data before insert

Run the python script to scrap the data and generate the csv file. Take note the path of the csv located.



In our case the file located in '*/home/student/wqd7005/hive/exchangeRate.csv*'

Setup Hive to insert data by load data from CSV

Step 1: Open Hive from Terminal by type *hive.* It will display as below.

```
student@student-VirtualBox:~/wqd7005/hive$ hive
ls: cannot access '/home/WQD7007/spark/lib/spark-assembly-*.jar': No such file or directory

Logging initialized using configuration in jar:file:/home/WQD7007/hive/lib/hive-common-1.2.2.jar!/hive-log4j.properties
hive>
```

Step 2: Create database and use the created database.

```
student@student-VirtualBox:~/wqd7005/hive$ hive
ls: cannot access '/home/WQD7007/spark/lib/spark-assembly-*.jar': No such file or directory

Logging initialized using configuration in jar:file:/home/WQD7007/hive/lib/hive-common-1.2.2.jar!/hive-log4j.properties
hive> CREATE DATABASE exchangeRate;
OK
Time taken: 2.1 seconds
hive> USE exchangeRate;
OK
Time taken: 0.089 seconds
hive> show tables;
OK
Time taken: 0.556 seconds
hive>
```

Enter Command ‘CREATE DATABASE exchangeRate;’

Followed by ‘USE exchangeRate;’

Step 3: Create table rate.

```
hive> CREATE TABLE rate (datecur STRING,usd STRING,gbp STRING,eur STRING,jpy100 STRING,chf STRING,aud STRING,cad STRING,sgd STRING,hkd100 STRING)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > STORED AS TEXTFILE;
OK
Time taken: 0.953 seconds
hive>
```

CREATE TABLE rate (dateCur STRING, usd STRING, gbp STRING, eur STRING, jpy100 STRING, chf STRING, aud STRING, cad STRING, sgd STRING, hkd100 STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

Step 4: Load CSV Data into Hive

```
hive> LOAD DATA LOCAL INPATH '/home/student/wqd7005/hive/exchangeRate.csv' INTO TABLE rate;
Loading data to table exchangerate.rate
Table exchangerate.rate stats: [numFiles=1, totalSize=58511]
OK
Time taken: 1.172 seconds
hive>
```

LOAD DATA LOCAL INPATH '/home/student/wqd7005/hive/exchangeRate.csv' INTO TABLE rate;

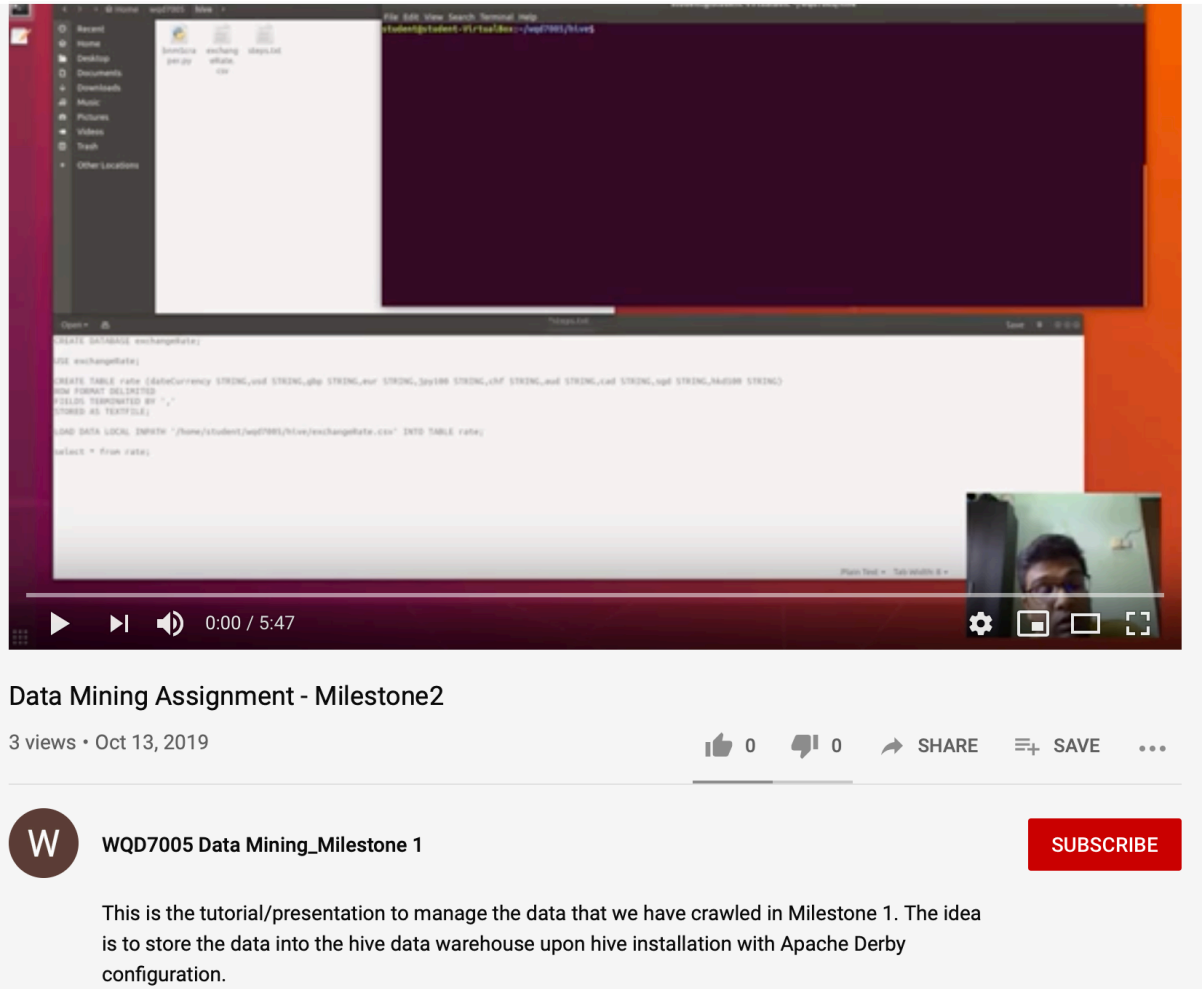
Step 5: View inserted data

```
hive> select * from rate;
```

```
24/9/2019       0.0180  0.1025  2.6278  0.2775  5.8987  113.7063        2.6521  3.6802  0.2555
25/9/2019       0.0180  0.1016  2.6420  0.2734  5.8903  113.8696        2.6559  3.6816  0.2557
26/9/2019       0.0181  0.1023  2.6298  0.2740  5.9053  114.1418        2.6622  3.6891  0.2570
27/9/2019       0.0181  0.1014  2.6349  0.2740  5.9018  114.0329        2.6597  3.6934  0.2561
Time taken: 0.207 seconds, Fetched: 779 row(s)
hive>
```

SELECT * FROM rate;

2.0 YouTube Presentation



2.1 YouTube Link

https://youtu.be/JR932MTR-aA

3.0 Data Storing – Store Data into Datawarehouse

3.1 GitHub Link

We have collaborated in this GitHub platform to work together on this milestone

https://github.com/firdausshajahan/WQD7005DATAMINING/tree/master/Milestone2