HADOOP DISTRIBUTED FILE SYSTEM (HDFS)

Before we begin our Hadoop Multi-node Installation, **IT'S IMPORTANT** to follow these two steps: (1) Set Up SSH and (2) Single-node Installation. If you skip these steps, you **WILL FAIL**.

In this tutorial, I will be using VMWorkstation Player 15 (https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html) on my Windows 10 and install Linux Ubuntu (https://ubuntu.com/download/desktop).

STEP 1: SSH SET-UP

SHORTCUT NOTE
To close file edit in the terminal is ESC, and save configuration by :wq press ENTER
To open terminal is CTRL+ALT+T

1. LOG IN AS ROOT

```
$sudo su
#whoami --should give root
```

2. ADDING A DEDICATED HADOOP SYSTEM USER CALLED 'HDUSER'

We will use a dedicated Hadoop user account for running Hadoop. While that's required it is recommended because it helps to separate the Hadoop installation from other software applications and user accounts running on the same machine (think: security, permissions, backups, etc).

3. CREATE A GROUP CALLED HADOOP

```
#sudo addgroup hadoop
```

4. CREATE AN USER HDUSER

```
#sudo adduser hduser
```

It will ask you to enter password 2 times followed by some details, just press enter and Yes. For password, enter your password that you've set up.

5. ADD HDUSER TO HADOOP GROUP

```
#sudo adduser hduser hadoop
```

**You can combine online for 4 & 5.

```
#sudo adduser -ingroup Hadoop hduser
```

6. ADD THE 'HDUSER' TO SUDOERS LIST SO THAT HDUSER CAN DO ADMIN TASKS

```
$sudo visudo
```

Add a line under ##Allow member of group sudo to execute any command anywhere in the format.

```
hduser ALL=(ALL)ALL
```

Press CTRL+X, Y enter and enter

This will add the user hduser and the group hadoop to your local machine.

7. LOG OUT FROM YOUR SYSTEM (RESTART) & LOG IN AGAIN AS HDUSER

8. CONFIGURING SSH

Hadoop requires SSH access to manage its nodes, i.e. remote machines plus your local machine if you want to use Hadoop on it.

```
$sudo apt-get install openssh-server
```

Enter password that you've set up and Y to continue

9. GENERATE SSH FOR COMMUNICATION

```
$ssh-keygen
```

Just press enter for whatever is asked

10. COPY PUBLIC KEY TO AUTHORIZED_KEY FILE & EDIT THE PERMISSION

Copy this public key to the authorized_keys file, so that ssh should not require password every time

```
$cat~/.ssh/id_rsa.pub>>/.ssh/authorized_keys
```

#Change of permission of the authorized_keys file to have all permission for hduser:

```
$chmod 700~/.ssh/authorized_keys
```

11. START SSH

```
$sudo /etc/init.d/ssh restart
```

12. TEST YOUR SSH CONNECTIVITY

```
$ssh localhost
```

Type 'YES', when asked for. You should be able to connect without password. If you're asked to enter password here, then something went wrong. Please check your steps.

13. DISABLE IPV6

Hadoop and IPV6 do not agree on the meaning of 0.0.0.0 address; thus, it is advisable to disable IPV6 adding the following lines at the end of /etc/sysct1.conf

```
$sudo vim /etc/sysctl.conf
```

Add this on the very last line:

```
#disable IPV6
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
net.ipv6.conf.lo.disable_ipv6=1
```

14. CHECK IF IPV6 IS DISABLED

After a system reboot the output of should be 1, meaning that IPV5 is actually disabled. Without reboot, it would still show you 0.

```
$cat/proc/sys/net/ipv6/conf/all/disable_ipv6
```

STEP 2: HADOOP SINGLE-NODE INSTALLATION

Please make sure that you have Java on your computer, execute this command on Terminal **java -version**. If you have not, execute this command **sudo apt-get install openjdk-8-jdk**

1. **DOWNLOAD HADOOP (https://www.apache.org/dist/hadoop/common/hadoop-2.8.5/) AND SAVE IT TO hduser/Desktop**

2. **MOVE THE ZIP FILE TO THE /USR/LOCAL/**
   Open Terminal, we will be executing some queries…

   #This will move Hadoop folder to your /usr/local/ as it will not allow you to do manually.

```
$sudo mv ~/Desktop/hadoop-2.8.5.tar.gz /usr/local/
$cd /usr/local
```

   #Extract Hadoop folder

```
$sudo tar -vxf hadoop-2.8.5.tar.gz
```

   #Removing the tar.gz folder since we have extracted the folder

```
$sudo rm hadoop-2.8.5.tar.gz
```

   #Create a shortcut folder 'hadoop' instead of typing 'hadoop-2.8.5' every time. This will save so much time as we move forward

```
$sudo in -s hadoop-2.8.5 hadoop
```

   #Change the owner of Hadoop folder into hduser: hadoop instead of root root

```
$sudo chown -R hduser:hadoop hadoop-2.8.5
```

   **To check if root if changed to hduser: hadoop, please run this code

```
$ls -ltr
$sudo chmod 777 hadoop-2.8.5 #Give all permissions to hadoop-2.8.5
```

3. **EDIT HADOOP-ENV.SH AND CONFIGURE JAVA**
   Add the following to /usr/local/hadoop/etc/hadoop/hadoop-env.sh by removing:

```
export JAVA_HOME = ${JAVA_HOME}
$sudo vim /usr/local/hadoop/etc/hadoop/hadoop-env.sh
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
export HADOOP_HOME_WARN_SUPPRESS="TRUE"
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/          (Check your java path file by #sudo update-
alternative --config java)
```

4. **UPDATE $HOME/.bashrc**
   Add the following lines to the end of the $HOME/.bashrc file of user nail. If you use a shell other than bash, you should of course update its appropriate configuration files instead of .bashrc

```
$vim ~/.bashrc
```

```
#Set Hadoop-related environment variables
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_PREFIX=/usr/local/hadoop
export HADOOP_MAPRED_HOME=${HADOOP_HOME}
export HADOOP_COMMON_HOME=${HADOOP_HOME}
export HADOOP_HDFS_HOME=${HADOOP_HOME}
export HADOOP_ YARN_HOME=${HADOOP_HOME}
export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop
#Native Path
export HADOOP_COMMON_LIB_NATIVE_DIR=${HADOOP_PREFIX}/lib/native
export HADOOP_OPTS=”-Djava.library.path=$HADOOP_PREFIX/lib”

#Set JAVA_HOME
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
```

5. **CREATE A TEMPROARY DIRECTORY WHICH WILL BE USE AS BASE LOCATION FOR DFS**

Now we create the directory and the required ownerships and permissions:

```
$sudo mkdir -p /app/hadoop/tmp
$sudo chown -R hduser:Hadoop /app/hadoop/tmp
$sudo chmod -R /app/hadoop/tmp
```

6. **EDIT CORE-SITE.XML**

```
$vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

Add the following snippets between the <configuration> ………… </configuration> tags

```
<property>
    <name>hadoop.tmp.dir</name>
    <value>/app/hadoop/tmp</value>
</property>
<property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000</value>
</property>
```

7. **UPDATE YARN-SITE.XML**

```
$vim /usr/local/hadoop/etc/hadoop/yarn-site.xml
```

Add the following snippets between the <configuration> ………… </configuration> tags

```
<property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
</property>
<property>
```

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

8. CREATE MAPRED-SITE.XML FILE FROM MAPRED-SITE.XML.TEMPLATE

```
$cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

Add the following snippets between the <configuration> ………… </configuration> tags

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

9. CREATE DIRECTORY WHERE HADOOP WILL STORE ITS WORK AND GIVE GOOD PERMISSION TO IT. ALSO CHANGE THE OWNER OF THOSE TWO DIRECTORIES TO hduser:hadoop UserName:groupName

```
$sudo mkdir -p /usr/local/hadoop/yarn_data/hdfs/namenode
$sudo mkdir -p /usr/local/hadoop/yarn_data/hdfs/datanode

$sudo chmod 777 /usr/local/hadoop/yarn_data/hdfs/namenode
$sudo chmod 777 /usr/local/hadoop/yarn_data/hdfs/datanode

$sudo chown -R hduser:hadoop /usr/local/hadoop/yarn_data/hdfs/namenode
$sudo chown -R hduser:hadoop /usr/local/hadoop/yarn_data/hdfs/datanode
```

10. EDIT HDFS-SITE.XML

```
$vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

Add the following snippets between the <configuration> ………… </configuration> tags

```
<property>
      <name>dfs.replication</name>
      <value>1</value>
</property>
<property>
      <name>dfs.namenode.name.dir</name>
      <value>file:/usr/local/hadoop/yarn_data/hdfs/namenode</value>
</property>
<property>
      <name>dfs.datanode.data.dir</name>
      <value>file:/usr/local/hadoop/yarn_data/hdfs/datanode</value>
</property>
```

11. FORMAT YOUR NODE

Open a new Terminal as the Hadoop command will not work.

Format HDFS cluster with command below:

```
$hadoop namenode -format
```

If the format is not working, double check your entries in .bashrc file. The .bashrc updating come into force only if you have opened a new terminal.

12. STARTING YOUR SINGLE-NODE CLUSTER

Congratulations! Your Hadoop single-node cluster is ready to use. Test your cluster by running the following commands.

```
$start-dfs.sh          --Type YES if anything asked for
$start-yarn.sh
```

13. CHECK IF ALL THE NECESSARY HADOOP DAEMON IS RUNNING OUT

```
$jps
NameNode
ResourceManager
Jps
Secondary NameNode
Node Manager
DataNode
```

14. CHECK IF HOME FOLDER IS CREATED OR NOT IN HDFS

```
$hadoop fs -ls
```

```
19/11/01 11:57:28 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

** If you get the error above, this means your Hadoop home directory was not created successfully.

Please execute this command below:

```
$hdfs dfs -mkdir -p /user/hduser
```

Now you should not get error with below command. For the first time you will not get any output as the hdfs home folder is empty.

15. CHECK IF THE HADDOP IS ACCESSIBLE THROUGH BROWSER BY HITTING THE BELOW URLs

NameNode	http://localhost:50070
ResourceManager	http://localhost:8088

STEP 3: HADOOP MULTI-NODE INSTALLATION

1. CREATE 2 NODES ON VIRTUAL MACHINE

We will create 2 new slaves nodes called Data1 and Data2. We will repeat installation PART 1 & 2.

2. CHECK IF NODES ARE REACHABLE

Find the IP Address of all 3 systems and try to ping each other.

```
$ifconfig
```

For example, these are 3 IPs in my VM:

```
Master 192.168.211.131
Data1 192.168.211.129
Data2 192.168.211.130
```

To stop ping, CTRL+C

```
Master hduser@ubuntu
$ping 192.168.211.129          // Master pinging slave1
$ping 192.168.211.130          // Master pinging slave2

Data1 hduser@ubuntu
Master 192.168.211.131         // Data1 pinging master
Data2 192.168.211.130          // Data2 pinging data2
Data2 hduser@ubuntu
Master 192.168.211.131         // Data2 pinging master
Data1 192.168.211.129          // Data2 pinging data1
```

3. CHANGE THE HOSTNAME OF ALL 3 SYSTEMS

```
Master VM
$sudo vim /etc/hostname
```

Press i on the keyboard and write ‘master’ by deleting Ubuntu

Press ESC on the keyboard

Save the configuration by :wq ENTER

Repeat the steps above with Data1 and Data2 (It’s recommended to write in low caps)

4. UPDATE THE HOSTS ON ALL 3 NODES

```
Master VM:
$sudo vim /etc/hosts

127.0.0.1      localhost             #Don’t REMOVE this line
127.0.0.1      master   #REMOVE this line
192.168.211.131 master
192.168.211.129 data1
192.168.211.130 data2
```

5. CONFIRM THE HOSTNAME OF ALL 3 NODES

Executing the below command on each VM.

```
$hostname
```

It should print master, data1, data2 in 3 machines respectively.

6. PING EACH OTHER USING HOSTNAME

Start pinging each other system again using the hostname instead of IPAddress.

```
Master
$ping data1
$ping data2

Data1
$ping master
$ping data2

Data2
$ping master
$ping data1
```

7. TEST SSH CONNECTIVITY

Test the SSH connectivity by doing the following. It will ask for yes or no and you should type ‘yes’ Perform SSH master/data1/data2 on each of the node to verify the connectivity

8. UPDATE CORE-SITE.XML

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

9. UPDATE HDFS-SITE.XML

```
<property>
        <name>dfs.replication</name>
        <value>2</value>
</property>
<property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
</property>
```

10. UPDATE YARN-SITE.XML

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

<property>
```

```
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>master:8025</value>
</property>
<property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>master:8030</value>
</property>

<property>
    <name>yarn.resourcemanager.address</name>
    <value>master:8050</value>
</property>
```

11. UPDATE MAPRED-SITE.XML

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

<property>
    <name>mapreduce.jobhistory.address</name>
    <value>master:10020</value>
</property>
```

MASTER ONLY CONFIGURATION

12. UPDATE MASTER AND SLAVES (DATA) FILES (Master Node only)

If you see any entry related to localhost, feel free to delete it. This file is just helper file are used by Hadoop scripts to start appropriate services on master and slave nodes.

```
$sudo vim /usr/local/hadoop/etc/hadoop/slaves
data1
data2

$sudo vim /usr/local/hadoop/etc/hadoop/masters
Master
```

Note: You don’t need to configure them in slave nodes

13. RECREATE NAMENODE FOLDER (MASTER ONLY)

```
sudo rm -rf /usr/local/hadoop_tmp
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
sudo chmod 777 /usr/local/hadoop_tmp/hdfs/namenode
```

14. RECREATE DATANODE FOLDER (ALL DATA NODES ONLY)

```
sudo rm -rf /usr/local/hadoop_tmp
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
sudo chmod 777 /usr/local/hadoop_tmp/hdfs/datanode
```

15. FORMAT THE NAMENODE (MASTER ONLY)

Before starting the cluster, we need to format the Namenode. Using the following command only on master node:

```
$hdfs namenode -format
```

16. START THE DFS & YARN (MASTER ONLY)

```
$start-dfs.sh
$start-yarn.sh
```

You should observe that it tries to start data node on slave nodes one by one.

Once it is started, do a JPS on master and slaves.

Jps on Master node

```
hduser@master$ jps
3379 NameNode                    #because of start-dfs.sh
3175 ScondayNameNode             #because of start-dfs.sh
3539 ResourceManager             #because of start-yarn.sh
```

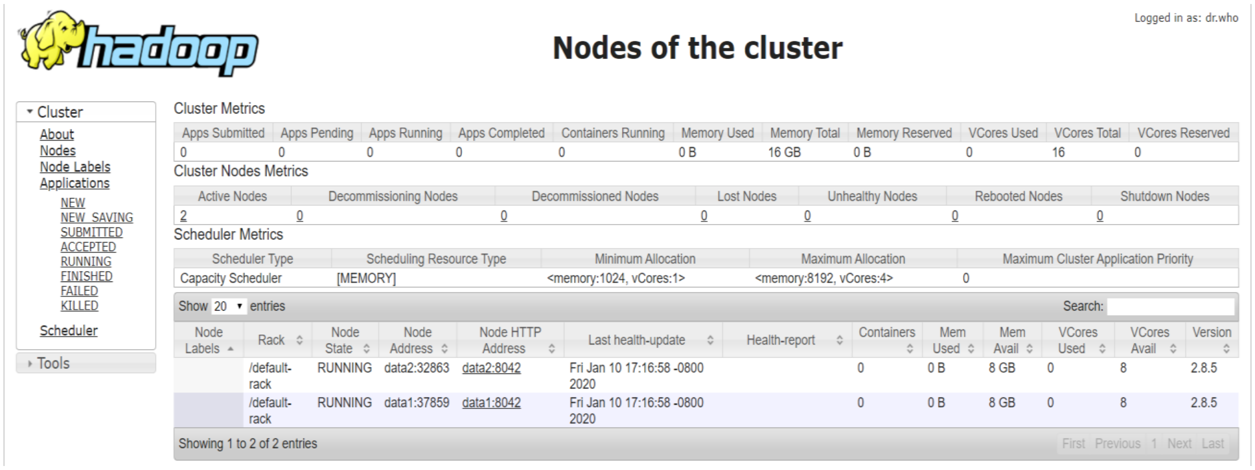
Jps on slave nodes (data1 and data2)

```
hduser@slave1$ jps
2484 DataNode                    #because of start-dfs.sh
2607 NodeManager                 #because of start-yarn.sh
```

17. REVIEW YARN CONSOLE

If all the services started successfully on all nodes, then you should see all your nodes listed under Yarn Nodes. You can hit the following url on your browser ad very that:

http://master:8088/cluster



Nodes of the cluster

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
0	0	0	0	0	0 B	16 GB	0 B	0	16	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
2	0	0	0	0	0	0

Scheduler Metrics

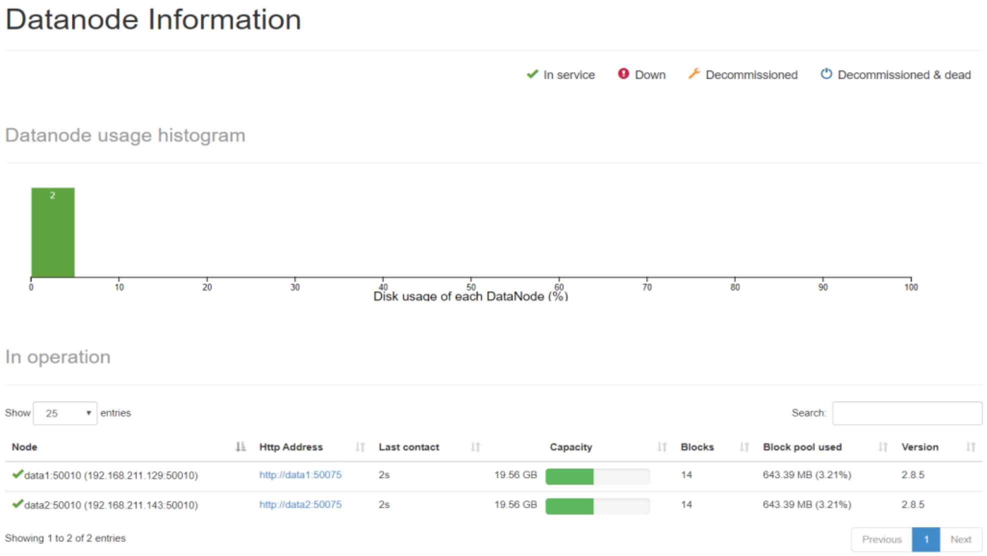
Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries Search:

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	VCores Used	VCores Avail	Version
	/default-rack	RUNNING	data2:32863	data2:8042	Fri Jan 10 17:16:58 -0800 2020		0	0 B	8 GB	0	8	2.8.5
	/default-rack	RUNNING	data1:37859	data1:8042	Fri Jan 10 17:16:58 -0800 2020		0	0 B	8 GB	0	8	2.8.5

Showing 1 to 2 of 2 entries First Previous 1 Next Last

http://master:50070 #can show live node count and info about each live node.



Datanode Information

Datanode usage histogram

In operation

Show 25 entries Search:

Node	Http Address	Last contact	Capacity	Blocks	Block pool used	Version
✓data1:50010 (192.168.211.129:50010)	http://data1:50075	2s	19.56 GB	14	643.39 MB (3.21%)	2.8.5
✓data2:50010 (192.168.211.143:50010)	http://data2:50075	2s	19.56 GB	14	643.39 MB (3.21%)	2.8.5

Showing 1 to 2 of 2 entries Previous 1 Next

You can also get the report of your cluster by issuing the below commands:

```
hduser@master$ hdfs dfsadmin -report
```

STEP 4: HIVE INSTALLATION

1. DOWNLOAD HIVE

You can directly use ‘wget’ command also to download hive from your home directory.

```
$mkdir/home/hduser/ecosystem
$cd /home/hduser/ecosystem
$wget http://apache.mesi.com.ar/hive/hive-2.1.0/apache-hive-2.1.0-bin.tar.gz
```

2. EXTRACT THE TAR.GZ FILE

```
$tar -xfz apache-hive-2.1.0-bin.tar.gz
```

3. CREATE A SYMBOLIC LINK FOR HIVE

```
$ln -s apache-hive-2.1.0-bin hive
```

4. **SET HIVE_HOME & PATH POINTING TO HIVE INSTALLATION DIRECTORY IN.BASHRC FILE**

```
$vim /home/hduser/.bashrc
```

Add the below snipped at the end of the line:

```
Export HIVE_HOME=/home/hduser/ecosystem/hive
Export PATH=$PATH:$HIVE_HOME/bin/
```

5. **HIVE USES HADOOP, SO MAKE SURE**
   a. You must have Hadoop in your path or
   b. Export HADOOP_HOME=<hadoop-install-dir>

6. **MAKE SURE YOUR HADOOP IS IN RUNNING MODE**

```
$start-dfs.sh
$start-yarn.sh
```

Jps should give all the daemons running as shown below

```
3123 NodeManager
2615 DataNode
3000 ResourceManager
2490 NameNode
3468 Jps
2783 SecondaryNameNode
```

7. **CREATE TEMPORARY DIRECTORY AND WAREHOUSE DIRECTORY IN HDFS WITH PROPER PERMISSIONS**

These directories will be used by HIVE

```
$hdfs dfs -mkdir -p /user/hive/warehouse
$hdfs dfs -mkdir -p /tmp/hive
$hdfs dfs -chmod 777 /tmp
$hdfs dfs -chmod 777 /user/hive/warehouse
$hdfs dfs -chmod 777 /tmp/hive
```

8. **DELETE THE ABSOLUTE LOG4J-SLF4J-IMPL.JAR AS WE HAVE SAME JAR FILE PROVIDED BY HADOOP AND WE WILL USE HADOOP GIVER JAR**

```
$rm /home/hduser/ecosystem/hive/lib/log4j-slf4j-impl-2.4.1.jar
```

9. **INITILIAZE THE DATABASE TO BE USED WITH HIVE**

```
$schematool -initSchema -dbType derby
```

The above command will create a metastore_db folder with proper initialization and then when we launch Hive we will not get any problem.

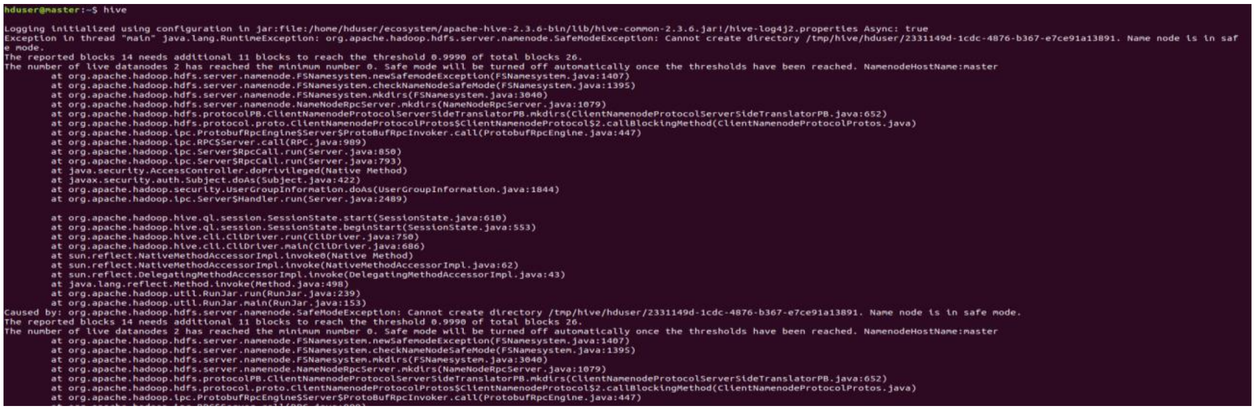
If you get the below error:
FUNCTION 'NUCLEUS_ASCII' already exist, delete metastore_Db from current folder and re-execute schematool command. You should get the successful execution as shown below.

10. **LOG IN TO HIVE**

Open a new terminal (ALT+CTRL+T) and issue the below command. You will get Hive terminal where you can write SQL query.

```
$hive
```

If you are getting this error on your Hive,



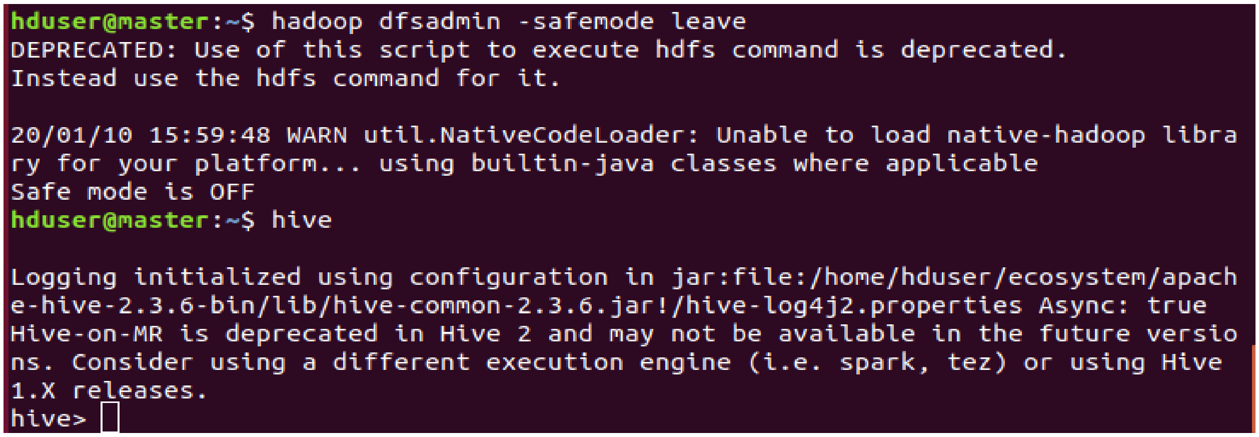
```
hduser@master:~$ hive

Logging initialized using configuration in jar:file:/home/hduser/ecosystem/apache-hive-2.3.6-bin/lib/hive-common-2.3.6.jar!/hive-log4j2.properties Async: true
Exception in thread "main" java.lang.RuntimeException: org.apache.hadoop.hdfs.server.namenode.SafeModeException: Cannot create directory /tmp/hive/hduser/2331149d-1cdc-4876-b367-e7ce91a13891. Name node is in saf
e mode.
The reported blocks 14 needs additional 11 blocks to reach the threshold 0.9990 of total blocks 26.
The number of live datanodes 2 has reached the minimum number 0. Safe mode will be turned off automatically once the thresholds have been reached. NamenodeHostName:master
        at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.newSafemodeException(FSNamesystem.java:1407)
        at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkNameNodeSafeMode(FSNamesystem.java:1395)
        at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirs(FSNamesystem.java:3040)
        at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.mkdirs(NameNodeRpcServer.java:1079)
        at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.mkdirs(ClientNamenodeProtocolServerSideTranslatorPB.java:652)
        at org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol$2.callBlockingMethod(ClientNamenodeProtocolProtos.java)
        at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:447)
        at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:989)
        at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:850)
        at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:793)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1844)
        at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2489)

        at org.apache.hadoop.hive.ql.session.SessionState.start(SessionState.java:610)
        at org.apache.hadoop.hive.ql.session.SessionState.beginStart(SessionState.java:553)
        at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:750)
        at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:686)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:239)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:153)
Caused by: org.apache.hadoop.hdfs.server.namenode.SafeModeException: Cannot create directory /tmp/hive/hduser/2331149d-1cdc-4876-b367-e7ce91a13891. Name node is in safe mode.
The reported blocks 14 needs additional 11 blocks to reach the threshold 0.9990 of total blocks 26.
The number of live datanodes 2 has reached the minimum number 0. Safe mode will be turned off automatically once the thresholds have been reached. NamenodeHostName:master
        at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.newSafemodeException(FSNamesystem.java:1407)
        at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkNameNodeSafeMode(FSNamesystem.java:1395)
        at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirs(FSNamesystem.java:3040)
        at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.mkdirs(NameNodeRpcServer.java:1079)
        at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.mkdirs(ClientNamenodeProtocolServerSideTranslatorPB.java:652)
        at org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol$2.callBlockingMethod(ClientNamenodeProtocolProtos.java)
        at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:447)
```

You can use the following command:

$hadoop dfsadmin -safemode leave



```
hduser@master:~$ hadoop dfsadmin -safemode leave
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

20/01/10 15:59:48 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Safe mode is OFF
hduser@master:~$ hive

Logging initialized using configuration in jar:file:/home/hduser/ecosystem/apach
e-hive-2.3.6-bin/lib/hive-common-2.3.6.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versio
ns. Consider using a different execution engine (i.e. spark, tez) or using Hive
1.X releases.
hive> 
```

Congratulations! You have successfully set up Hive! Now, it is time to run the query that you have.

STEP 5: HIVE QUERIES

According to Data Flair (2018), Hive has two types of tables which are as follows:

- **Internal Table (Managed Table).** It is also known as internal table. When creating a table in Hive, it by default manages the data. This means that Hive moves the data into its warehouse directory.
- **External Table.** We can create an external table. It tells Hive to refer to the data that is at an existing location outside the warehouse directory.

<u>When to use internal and external table?</u>

- **Internal Table.** Data is temporary, and we want Hive to completely manage the lifecycle of the data and table.

- **External Table.** Data is outside of Hive. We are not creating a table based on the existing table and need data to remain in the underlying location even after a **DROP TABLE**. This may apply if we are pointing multiple schemas at a single data set.

For this project, I will be querying with external table and using **"US Accidents"** that I obtained from Kaggle. It has one table with 49 columns and intentionally broke it down into four tables: accident, address, detail, and weather.

Here are my Hive scripts:

Table 1: Accident

```
CREATE EXTERNAL TABLE accident
(adID STRING, source STRING, tmc INT, severity INT, start_time STRING, end_time STRING, start_latitude
DECIMAL(8,4), start_longitude DECIMAL(8,4), distance DECIMAL(8,4))
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',' STORED AS TEXTFILE ;

load data local inpath '/home/hduser/hproject/accident/accident.csv' into table accident;
```

Table 2: Address

```
CREATE EXTERNAL TABLE address
(adID STRING, number INT, street STRING, side STRING, city STRING, county STRING, state STRING, zipcode STRING,
country STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',' STORED AS TEXTFILE ;

load data local inpath '/home/hduser/hproject/address/address.csv' into table address;
```

Table 3: Detail

```
CREATE EXTERNAL TABLE detail
(detID STRING, source STRING, description STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',' STORED AS TEXTFILE ;

load data local inpath '/home/hduser/hproject/detail/detail.csv' into table detail;
```

Table 4: Weather

```
CREATE EXTERNAL TABLE weather
(weatherID STRING, weather_timestamp STRING, temperature DECIMAL(4,2), wind_chill DECIMAL(4,2), humidity
INT, pressure DECIMAL(4,2), visibility INT, wind_direction STRING, wind_speed DECIMAL(4,2), precipitation
DECIMAL(4,2), weather_condition STRING )
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',' STORED AS TEXTFILE ;

load data local inpath '/home/hduser/hproject/weather/weather.csv' into table weather;
```

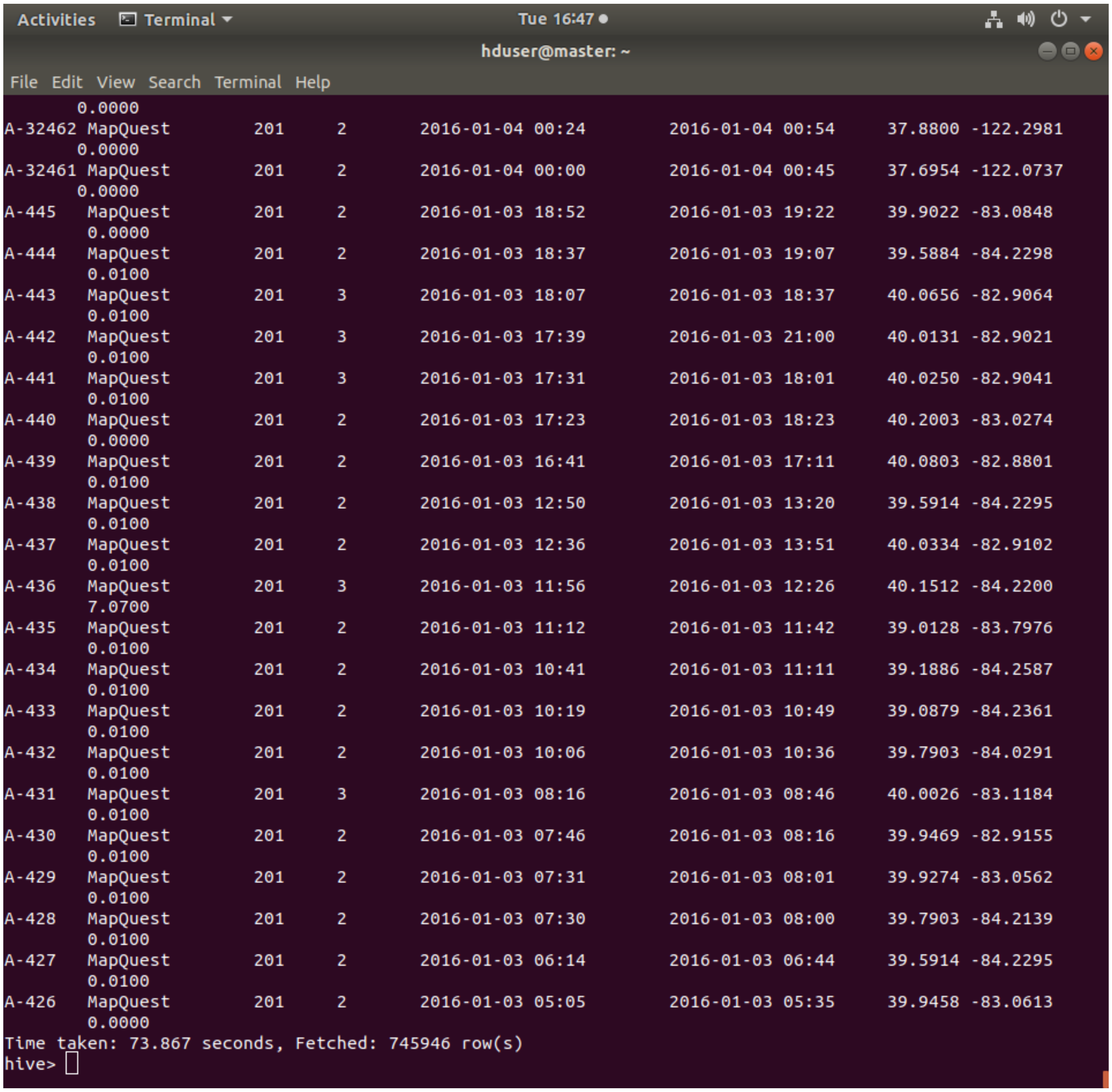
NOTES:

- To check table, use **show tables;**
- To describe table, use **describe (tablename);**
- To drop table, use **drop (tablename);**

Let’s run some queries now! I would like to explore…

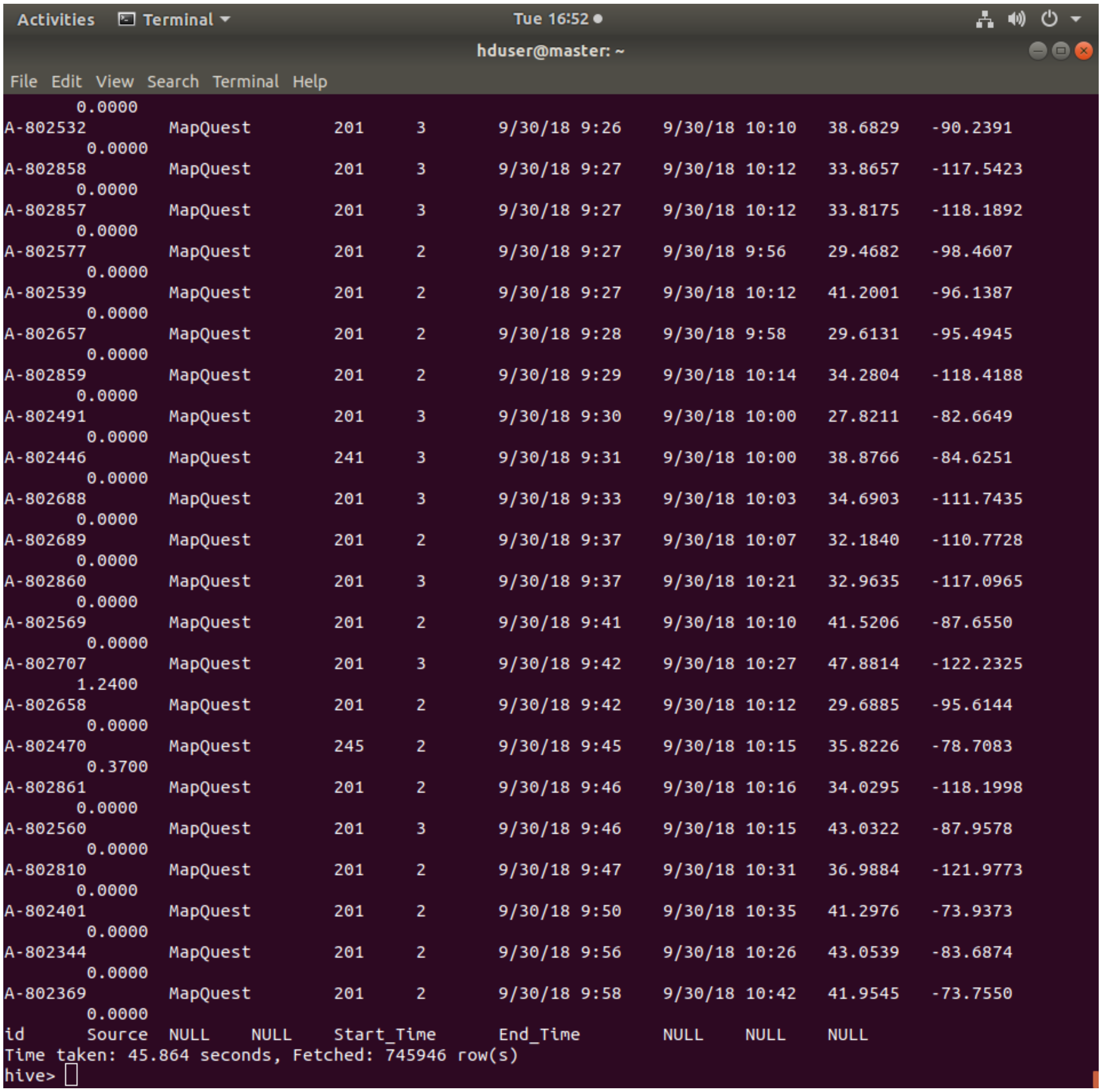
1. **All accidents data above 2016**

   SELECT * FROM accident WHERE start_time > ‘2016-01-01’
   SORT BY start_time DESC;



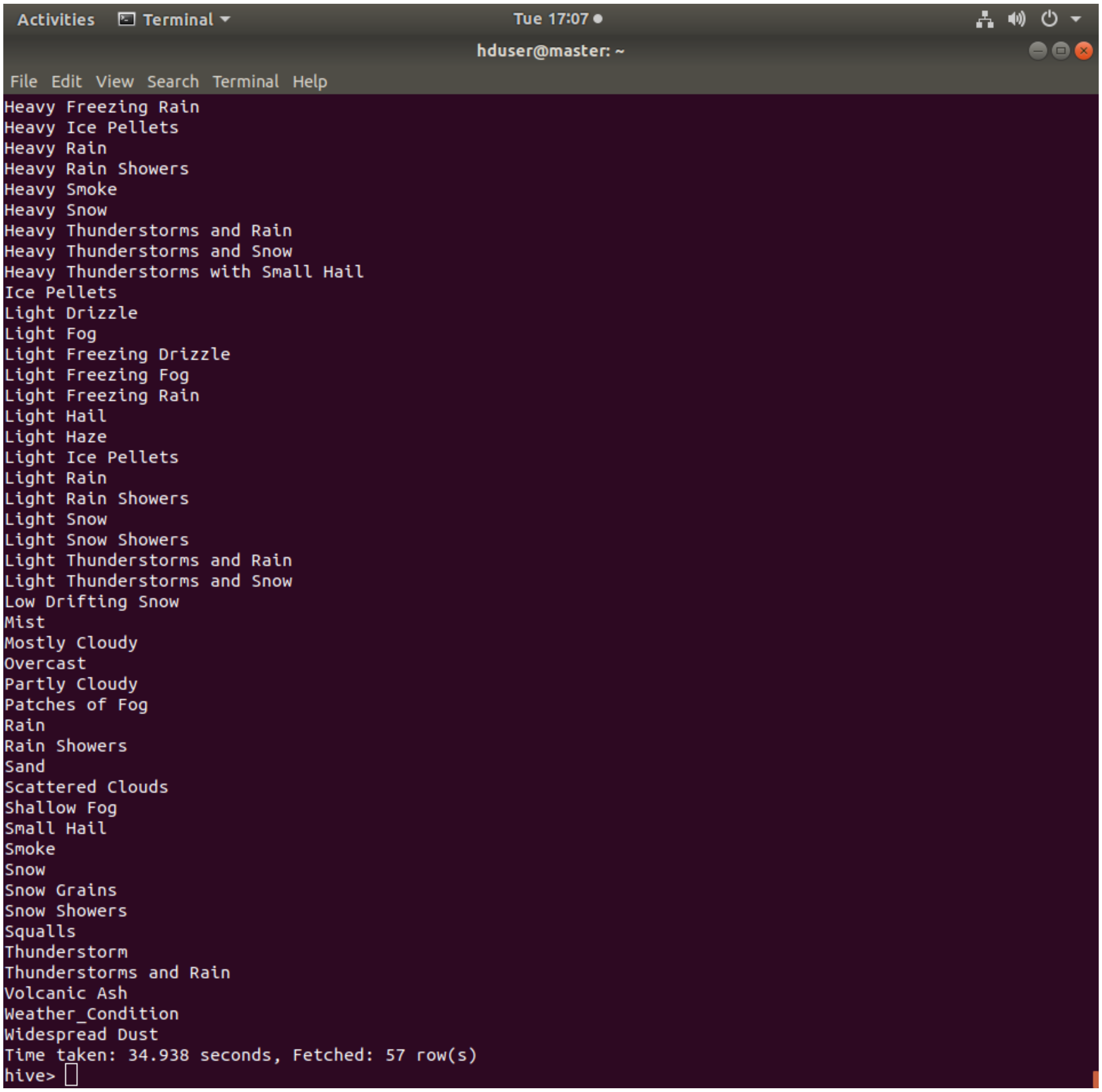
```
       0.0000
A-32462 MapQuest        201     2       2016-01-04 00:24        2016-01-04 00:54     37.8800 -122.2981
       0.0000
A-32461 MapQuest        201     2       2016-01-04 00:00        2016-01-04 00:45     37.6954 -122.0737
       0.0000
A-445   MapQuest        201     2       2016-01-03 18:52        2016-01-03 19:22     39.9022 -83.0848
        0.0000
A-444   MapQuest        201     2       2016-01-03 18:37        2016-01-03 19:07     39.5884 -84.2298
        0.0100
A-443   MapQuest        201     3       2016-01-03 18:07        2016-01-03 18:37     40.0656 -82.9064
        0.0100
A-442   MapQuest        201     3       2016-01-03 17:39        2016-01-03 21:00     40.0131 -82.9021
        0.0100
A-441   MapQuest        201     3       2016-01-03 17:31        2016-01-03 18:01     40.0250 -82.9041
        0.0100
A-440   MapQuest        201     2       2016-01-03 17:23        2016-01-03 18:23     40.2003 -83.0274
        0.0000
A-439   MapQuest        201     2       2016-01-03 16:41        2016-01-03 17:11     40.0803 -82.8801
        0.0100
A-438   MapQuest        201     2       2016-01-03 12:50        2016-01-03 13:20     39.5914 -84.2295
        0.0100
A-437   MapQuest        201     2       2016-01-03 12:36        2016-01-03 13:51     40.0334 -82.9102
        0.0100
A-436   MapQuest        201     3       2016-01-03 11:56        2016-01-03 12:26     40.1512 -84.2200
        7.0700
A-435   MapQuest        201     2       2016-01-03 11:12        2016-01-03 11:42     39.0128 -83.7976
        0.0100
A-434   MapQuest        201     2       2016-01-03 10:41        2016-01-03 11:11     39.1886 -84.2587
        0.0100
A-433   MapQuest        201     2       2016-01-03 10:19        2016-01-03 10:49     39.0879 -84.2361
        0.0100
A-432   MapQuest        201     2       2016-01-03 10:06        2016-01-03 10:36     39.7903 -84.0291
        0.0100
A-431   MapQuest        201     3       2016-01-03 08:16        2016-01-03 08:46     40.0026 -83.1184
        0.0100
A-430   MapQuest        201     2       2016-01-03 07:46        2016-01-03 08:16     39.9469 -82.9155
        0.0100
A-429   MapQuest        201     2       2016-01-03 07:31        2016-01-03 08:01     39.9274 -83.0562
        0.0100
A-428   MapQuest        201     2       2016-01-03 07:30        2016-01-03 08:00     39.7903 -84.2139
        0.0100
A-427   MapQuest        201     2       2016-01-03 06:14        2016-01-03 06:44     39.5914 -84.2295
        0.0100
A-426   MapQuest        201     2       2016-01-03 05:05        2016-01-03 05:35     39.9458 -83.0613
        0.0000
Time taken: 73.867 seconds, Fetched: 745946 row(s)
hive> 
```

Now, let’s try with ascending!



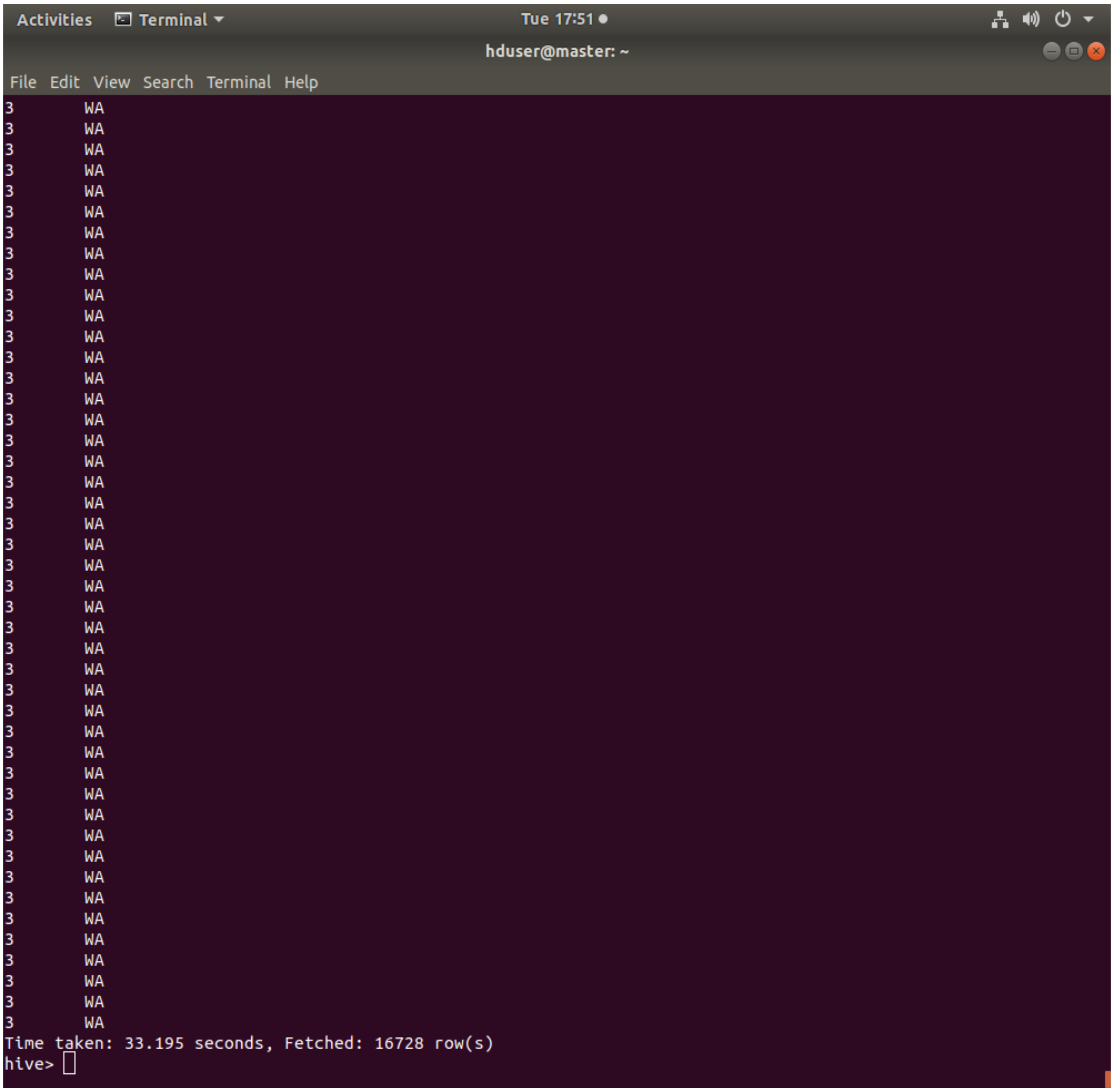
```
       0.0000
A-802532        MapQuest        201     3       9/30/18 9:26    9/30/18 10:10   38.6829   -90.2391
        0.0000
A-802858        MapQuest        201     3       9/30/18 9:27    9/30/18 10:12   33.8657   -117.5423
       0.0000
A-802857        MapQuest        201     3       9/30/18 9:27    9/30/18 10:12   33.8175   -118.1892
        0.0000
A-802577        MapQuest        201     2       9/30/18 9:27    9/30/18 9:56    29.4682   -98.4607
        0.0000
A-802539        MapQuest        201     2       9/30/18 9:27    9/30/18 10:12   41.2001   -96.1387
        0.0000
A-802657        MapQuest        201     2       9/30/18 9:28    9/30/18 9:58    29.6131   -95.4945
        0.0000
A-802859        MapQuest        201     2       9/30/18 9:29    9/30/18 10:14   34.2804   -118.4188
       0.0000
A-802491        MapQuest        201     3       9/30/18 9:30    9/30/18 10:00   27.8211   -82.6649
        0.0000
A-802446        MapQuest        241     3       9/30/18 9:31    9/30/18 10:00   38.8766   -84.6251
       0.0000
A-802688        MapQuest        201     3       9/30/18 9:33    9/30/18 10:03   34.6903   -111.7435
       0.0000
A-802689        MapQuest        201     2       9/30/18 9:37    9/30/18 10:07   32.1840   -110.7728
       0.0000
A-802860        MapQuest        201     3       9/30/18 9:37    9/30/18 10:21   32.9635   -117.0965
       0.0000
A-802569        MapQuest        201     2       9/30/18 9:41    9/30/18 10:10   41.5206   -87.6550
        0.0000
A-802707        MapQuest        201     3       9/30/18 9:42    9/30/18 10:27   47.8814   -122.2325
       1.2400
A-802658        MapQuest        201     2       9/30/18 9:42    9/30/18 10:12   29.6885   -95.6144
        0.0000
A-802470        MapQuest        245     2       9/30/18 9:45    9/30/18 10:15   35.8226   -78.7083
        0.3700
A-802861        MapQuest        201     2       9/30/18 9:46    9/30/18 10:16   34.0295   -118.1998
       0.0000
A-802560        MapQuest        201     3       9/30/18 9:46    9/30/18 10:15   43.0322   -87.9578
        0.0000
A-802810        MapQuest        201     2       9/30/18 9:47    9/30/18 10:31   36.9884   -121.9773
       0.0000
A-802401        MapQuest        201     2       9/30/18 9:50    9/30/18 10:35   41.2976   -73.9373
        0.0000
A-802344        MapQuest        201     2       9/30/18 9:56    9/30/18 10:26   43.0539   -83.6874
        0.0000
A-802369        MapQuest        201     2       9/30/18 9:58    9/30/18 10:42   41.9545   -73.7550
        0.0000
id      Source  NULL    NULL    Start_Time      End_Time        NULL    NULL    NULL
Time taken: 45.864 seconds, Fetched: 745946 row(s)
hive> 
```

2. **Type of Weather Conditions**
   SELECT DISTINCT weather_condition from weather;



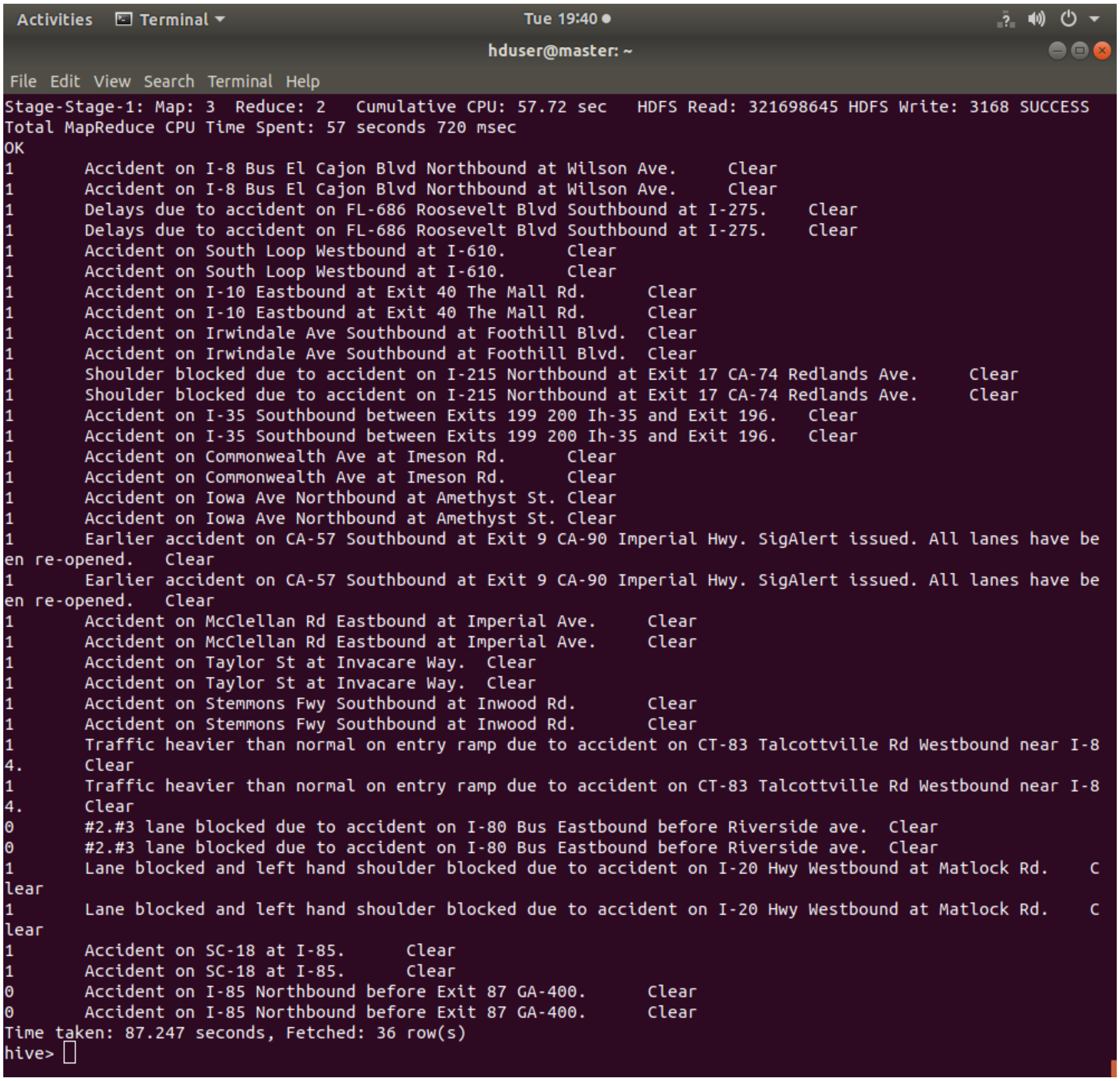
```
Heavy Freezing Rain
Heavy Ice Pellets
Heavy Rain
Heavy Rain Showers
Heavy Smoke
Heavy Snow
Heavy Thunderstorms and Rain
Heavy Thunderstorms and Snow
Heavy Thunderstorms with Small Hail
Ice Pellets
Light Drizzle
Light Fog
Light Freezing Drizzle
Light Freezing Fog
Light Freezing Rain
Light Hail
Light Haze
Light Ice Pellets
Light Rain
Light Rain Showers
Light Snow
Light Snow Showers
Light Thunderstorms and Rain
Light Thunderstorms and Snow
Low Drifting Snow
Mist
Mostly Cloudy
Overcast
Partly Cloudy
Patches of Fog
Rain
Rain Showers
Sand
Scattered Clouds
Shallow Fog
Small Hail
Smoke
Snow
Snow Grains
Snow Showers
Squalls
Thunderstorm
Thunderstorms and Rain
Volcanic Ash
Weather_Condition
Widespread Dust
Time taken: 34.938 seconds, Fetched: 57 row(s)
hive>
```

3. **<u>All accidents in Washington that the severity is 3</u>**
SELECT acc.severity , add.state
FROM accident acc JOIN address add ON (acc.adID = add.adID)
WHERE acc.severity > 2
AND add.state = 'WA';



```
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
3       WA
Time taken: 33.195 seconds, Fetched: 16728 row(s)
hive> 
```

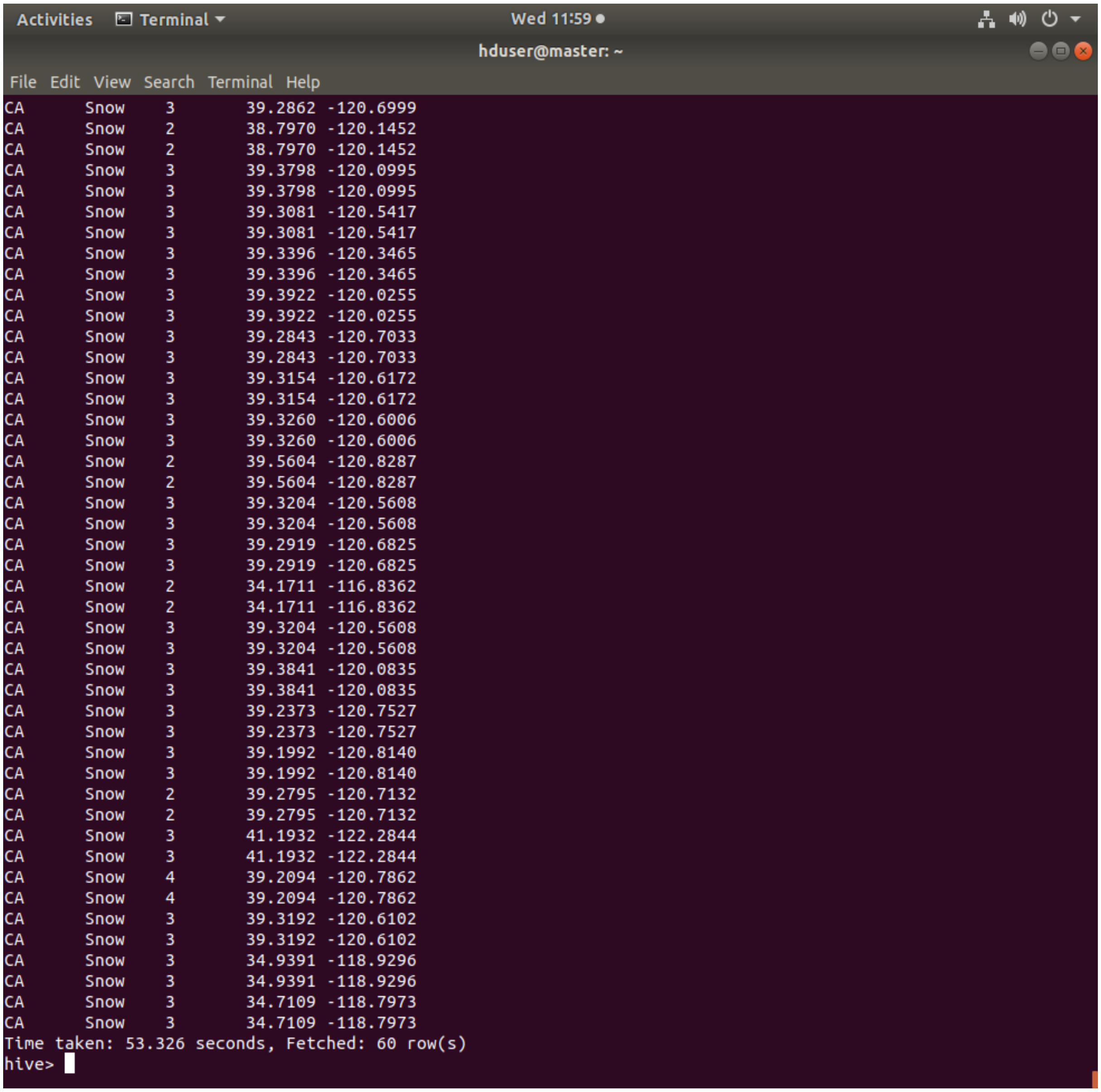
4. **<u>All accidents that happens in Interstate that severity is 1 when the weather condition is clear.</u>**
SELECT acc.severity, det.description, wea.weather_condition
FROM accident acc JOIN detail det ON (acc.adID = det.detID)
JOIN weather wea ON (det.detID = wea.weatherID)
WHERE acc.severity < 2
AND det.description LIKE '%I%'
AND wea.weather_condition = 'Clear';



```
Stage-Stage-1: Map: 3  Reduce: 2   Cumulative CPU: 57.72 sec   HDFS Read: 321698645 HDFS Write: 3168 SUCCESS
Total MapReduce CPU Time Spent: 57 seconds 720 msec
OK
1       Accident on I-8 Bus El Cajon Blvd Northbound at Wilson Ave.     Clear
1       Accident on I-8 Bus El Cajon Blvd Northbound at Wilson Ave.     Clear
1       Delays due to accident on FL-686 Roosevelt Blvd Southbound at I-275.   Clear
1       Delays due to accident on FL-686 Roosevelt Blvd Southbound at I-275.   Clear
1       Accident on South Loop Westbound at I-610.      Clear
1       Accident on South Loop Westbound at I-610.      Clear
1       Accident on I-10 Eastbound at Exit 40 The Mall Rd.     Clear
1       Accident on I-10 Eastbound at Exit 40 The Mall Rd.     Clear
1       Accident on Irwindale Ave Southbound at Foothill Blvd.  Clear
1       Accident on Irwindale Ave Southbound at Foothill Blvd.  Clear
1       Shoulder blocked due to accident on I-215 Northbound at Exit 17 CA-74 Redlands Ave.     Clear
1       Shoulder blocked due to accident on I-215 Northbound at Exit 17 CA-74 Redlands Ave.     Clear
1       Accident on I-35 Southbound between Exits 199 200 Ih-35 and Exit 196.   Clear
1       Accident on I-35 Southbound between Exits 199 200 Ih-35 and Exit 196.   Clear
1       Accident on Commonwealth Ave at Imeson Rd.      Clear
1       Accident on Commonwealth Ave at Imeson Rd.      Clear
1       Accident on Iowa Ave Northbound at Amethyst St. Clear
1       Accident on Iowa Ave Northbound at Amethyst St. Clear
1       Earlier accident on CA-57 Southbound at Exit 9 CA-90 Imperial Hwy. SigAlert issued. All lanes have be
en re-opened.   Clear
1       Earlier accident on CA-57 Southbound at Exit 9 CA-90 Imperial Hwy. SigAlert issued. All lanes have be
en re-opened.   Clear
1       Accident on McClellan Rd Eastbound at Imperial Ave.     Clear
1       Accident on McClellan Rd Eastbound at Imperial Ave.     Clear
1       Accident on Taylor St at Invacare Way.  Clear
1       Accident on Taylor St at Invacare Way.  Clear
1       Accident on Stemmons Fwy Southbound at Inwood Rd.       Clear
1       Accident on Stemmons Fwy Southbound at Inwood Rd.       Clear
1       Traffic heavier than normal on entry ramp due to accident on CT-83 Talcottville Rd Westbound near I-8
4.      Clear
1       Traffic heavier than normal on entry ramp due to accident on CT-83 Talcottville Rd Westbound near I-8
4.      Clear
0       #2.#3 lane blocked due to accident on I-80 Bus Eastbound before Riverside ave.  Clear
0       #2.#3 lane blocked due to accident on I-80 Bus Eastbound before Riverside ave.  Clear
1       Lane blocked and left hand shoulder blocked due to accident on I-20 Hwy Westbound at Matlock Rd.    C
lear
1       Lane blocked and left hand shoulder blocked due to accident on I-20 Hwy Westbound at Matlock Rd.    C
lear
1       Accident on SC-18 at I-85.      Clear
1       Accident on SC-18 at I-85.      Clear
0       Accident on I-85 Northbound before Exit 87 GA-400.      Clear
0       Accident on I-85 Northbound before Exit 87 GA-400.      Clear
Time taken: 87.247 seconds, Fetched: 36 row(s)
hive> 
```

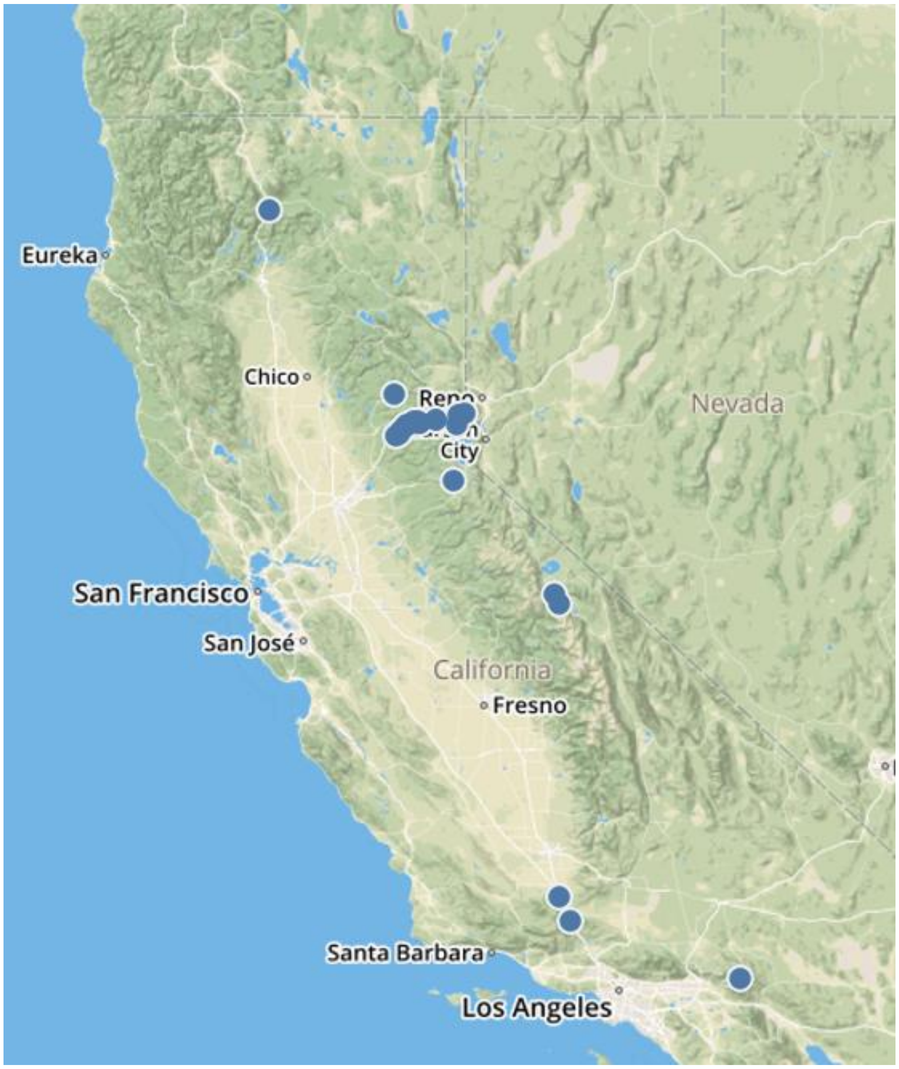
5. **<u>All accidents that happens in California</u>**

SELECT add.state, wea.weather_condition, acc.severity, acc.start_latitude, acc.start_longitude
FROM accident acc JOIN address add ON (acc.adID = add.adID)
JOIN weather wea ON (add.adID = wea.weatherID)
WHERE wea.weather_condition = 'Snow'
AND add.state = 'CA'



```
CA      Snow    3       39.2862 -120.6999
CA      Snow    2       38.7970 -120.1452
CA      Snow    2       38.7970 -120.1452
CA      Snow    3       39.3798 -120.0995
CA      Snow    3       39.3798 -120.0995
CA      Snow    3       39.3081 -120.5417
CA      Snow    3       39.3081 -120.5417
CA      Snow    3       39.3396 -120.3465
CA      Snow    3       39.3396 -120.3465
CA      Snow    3       39.3922 -120.0255
CA      Snow    3       39.3922 -120.0255
CA      Snow    3       39.2843 -120.7033
CA      Snow    3       39.2843 -120.7033
CA      Snow    3       39.3154 -120.6172
CA      Snow    3       39.3154 -120.6172
CA      Snow    3       39.3260 -120.6006
CA      Snow    3       39.3260 -120.6006
CA      Snow    2       39.5604 -120.8287
CA      Snow    2       39.5604 -120.8287
CA      Snow    3       39.3204 -120.5608
CA      Snow    3       39.3204 -120.5608
CA      Snow    3       39.2919 -120.6825
CA      Snow    3       39.2919 -120.6825
CA      Snow    2       34.1711 -116.8362
CA      Snow    2       34.1711 -116.8362
CA      Snow    3       39.3204 -120.5608
CA      Snow    3       39.3204 -120.5608
CA      Snow    3       39.3841 -120.0835
CA      Snow    3       39.3841 -120.0835
CA      Snow    3       39.2373 -120.7527
CA      Snow    3       39.2373 -120.7527
CA      Snow    3       39.1992 -120.8140
CA      Snow    3       39.1992 -120.8140
CA      Snow    2       39.2795 -120.7132
CA      Snow    2       39.2795 -120.7132
CA      Snow    3       41.1932 -122.2844
CA      Snow    3       41.1932 -122.2844
CA      Snow    4       39.2094 -120.7862
CA      Snow    4       39.2094 -120.7862
CA      Snow    3       39.3192 -120.6102
CA      Snow    3       39.3192 -120.6102
CA      Snow    3       34.9391 -118.9296
CA      Snow    3       34.9391 -118.9296
CA      Snow    3       34.7109 -118.7973
CA      Snow    3       34.7109 -118.7973
Time taken: 53.326 seconds, Fetched: 60 row(s)
hive> 
```

Let’s try to connect the results to Tableau… (Hint: Copy the results into excel and save it)



Wow! Isn’t cool? It looks like majority of the accidents happened in near Lake Tahoe!

During the process, you might encounter HDFS corrupt blocks that may interrupt the process.

To check if there is corrupted blocks

```
$hdfs dsck -list-corruptfileblocks
```

To delete the corrupted blocks

```
$hdfs fsck / -delete
```

STEP 6: EXIT HIVE

To stop Hive query, the command is **exit;** It is recommended to **stop-dfs.sh** and **stop-yarn.sh** after exiting from Hive. This will prevent error when logging back again.