WQD 7005 DATA MINING

MILESTONE 2

STORING DATA INTO HIVE DATA WAREHOUSE

STUDENTNAME: AIMI HASSIN
MATRIC NO. : 171988011
INSTRUCTOR : PROF. DR. TEH YING WAH

OBJECTIVES

1

HADOOP INSTALLATION

2

HIVE INSTALLATION

3

STORING DATA INTO HIVE DATA WAREHOUSE

4

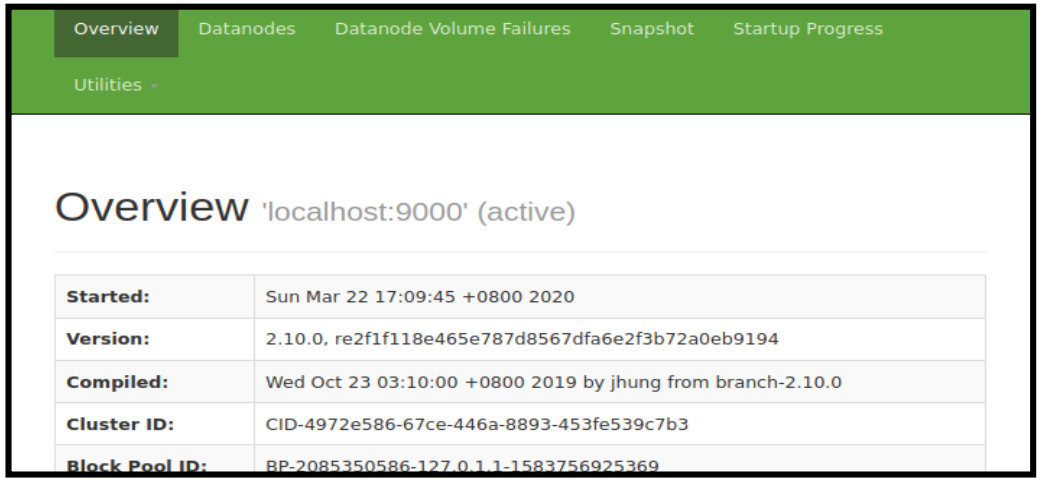
ACCESS THE STORED DATA

HADOOP INSTALLATION

Step-by-step installation can be referred here:

https://www.edureka.co/blog/install-hadoop-single-node-hadoop-cluster

Successful installation will be shown as below



HIVE INSTALLATION

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

Step 2: Extract the **tar** file.

- **Command:** tar -xzf apache-hive-2.1.0-bin.tar.gz
- **Command:** ls

Step 3: Edit the "**.bashrc**" file to update the environment variables for user.

- **Command:** sudo gedit .bashrc
- Add the following at the end of the file:

  # Set HIVE_HOME

  export HIVE_HOME=/home/amyh/apache-hive-2.1.0-bin
  export PATH=$PATH:/home/amyh/apache-hive-2.1.0-bin/bin
- **Command:** source .bashrc

Step 4: Check hive version.

- **Command:** $ hive --version

Step 5: Create **Hive** directories within **HDFS**. The directory 'warehouse' is the location to store the table or data related to hive.

Command:

- hdfs dfs -mkdir -p /user/hive/warehouse
- hdfs dfs -mkdir /tmp

Step 6: Set read/write permissions for table.

Command:

In this command, we are giving write permission to the group:

- hdfs dfs -chmod g+w /user/hive/warehouse
- hdfs dfs -chmod g+w /tmp

Step 7: Set Hadoop path in hive-env.sh

- **Command:** cd apache-hive-2.1.0-bin/
- **Command:** gedit conf/hive-env.sh

HIVE INSTALLATION

Set the parameters as below

 # Set HADOOP_INSTALL to point to a specific Hadoop install directory

export HADOOP_INSTALL=/usr/local/hadoop/hadoop-2.10.0

export HADOOP_HEAPSIZE=512

#Hive Configuration Directory can be controlled by:

Export HIVE_CONF_DIR=/home/amyh/apache-hive-2.1.0-bin/conf

Step 8: Edit hive-site.xml

- **Command:** gedit conf/hive-site.xml

<?xml version="1.0" encoding="UTF-8" standalone="no"?> <?xml-stylesheet type="text/xsl" href="configuration.xsl"?><!-- Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You 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. --> <configuration> <property> <name>javax.jdo.option.ConnectionURL</name> <value>jdbc:derby:;databaseName=/home/edureka/apache-hive-2.1.0-bin/metastore_db;create=true</value> <description> JDBC connect string for a JDBC metastore. To use SSL to encrypt/authenticate the connection, provide database-specific SSL flag in the connection URL. For example, jdbc:postgresql://myhost/db?ssl=true for postgres database. </description> </property> <property> <name>hive.metastore.warehouse.dir</name> <value>/user/hive/warehouse</value> <description>location of default database for the warehouse</description> </property> <property> <name>hive.metastore.uris</name> <value/> <description>Thrift URI for the remote metastore. Used by metastore client to connect to remote metastore.</description> </property> <property> <name>javax.jdo.option.ConnectionDriverName</name> <value>org.apache.derby.jdbc.EmbeddedDriver</value> <description>Driver class name for a JDBC metastore</description> </property> <property> <name>javax.jdo.PersistenceManagerFactoryClass</name> <value>org.datanucleus.api.jdo.JDOPersistenceManagerFactory</value> <description>class implementing the jdo persistence</description> </property> </configuration>

HIVE INSTALLATION

Step 9: By default, Hive uses Derby database. Initialize Derby database.

- **Command:** bin/schematool -initSchema -dbType derby

Step 10: Launch **Hive**.

- **Command:** hive

Step 11: Run few queries in Hive shell.

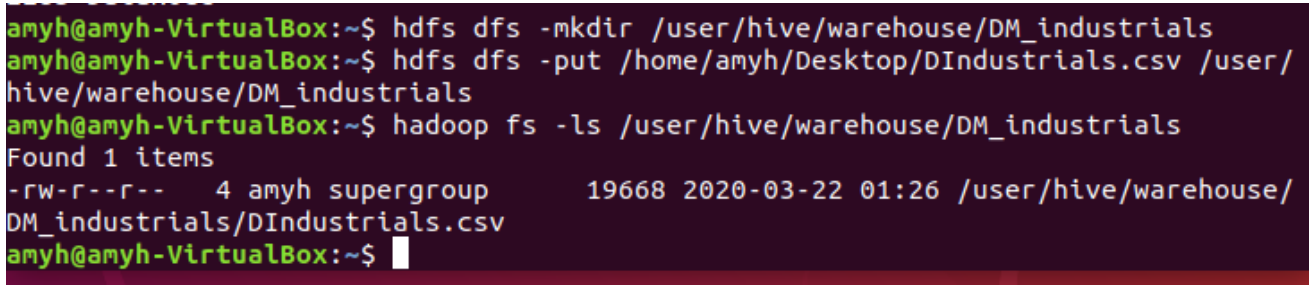
- **Command:** show databases;
- **Command:** create table employee (id string, name string, dept string) row format delimited fields terminated by ‘ ‘ stored as textfile;
- **Command:** show tables;

Step 12: To exit from **Hive:**

- **Command:** exit;

STORING DATA INTO HIVE DATA WAREHOUSE

1. Start all Hadoop daemons and make sure all localhosts are connected.
   - $ start-dfs.sh
   - $ start-yarn.sh
   - $ jps
2. Create a directory in HDFS
   - $ hdfs dfs –mkdir /user/hive/warehouse/DM_industrials
3. Move CSV file from local file system into HDFS directory
   - $ hdfs dfs –put /home/amyh/Desktop/Dindustrials.csv /user/hive/warehouse/DM_industrials
4. To check whether the file is available in HDFS or not
   - $ Hadoop fs –ls /user/hive/warehouse/DM_industrials



STORING DATA INTO HIVE DATA WAREHOUSE

```
hive> CREATE SCHEMA IF NOT EXISTS industrials;
OK
Time taken: 2.116 seconds
hive> CREATE EXTERNAL TABLE IF NOT EXISTS industrials_table
    > (`Date` STRING,
    > `Adj Close_BA` STRING,  `Adj Close_DAL` STRING,  `Adj Close_HON` STRING,
`Adj Close_TRI` STRING,  `Adj Close_UNP` STRING,
    > `Close_BA` STRING, `Close_DAL` STRING, `Close_HON` STRING, `Close_TRI` STR
ING, `Close_UNP` STRING,
    > `High_BA` STRING, `High_DAL` STRING, `High_HON` STRING, `High_TRI` STRING,
 `High_UNP` STRING,
    > `Low_BA` STRING, `Low_DAL` STRING, `Low_HON` STRING, `Low_TRI` STRING, `Lo
w_UNP` STRING,
    > `Open_BA` STRING, `Open_DAL` STRING, `Open_HON` STRING, `Open_TRI` STRING,
 `Open_UNP` STRING,
    > `Volume_BA` STRING, `Volume_DAL` STRING, `Volume_HON` STRING, `Volume_TRI`
 STRING, `Volume_UNP` STRING)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > STORED AS TEXTFILE
    > LOCATION '/user/hive/warehouse/DM_industrials';
OK
Time taken: 1.536 seconds
hive> Select * from industrials_table limit 5;
```

5. Load the data as an external Hive table

hive > CREATE SCHEMA IF NOT EXISTS industrials;

hive > CREATE EXTERNAL TABLE IF NOT EXISTS industrials_table

> (`Date` STRING,

> Adj Close_BA STRING, Adj Close_DAL STRING, Adj Close_HON STRING, Adj Close_TRI STRING, Adj Close_UNP STRING,

> Close_BA STRING, Close_DAL STRING, Close_HON STRING, Close_TRI STRING, Close_UNP STRING,

> High_BA STRING, High_DAL STRING, High_HON STRING, High_TRI STRING, High_UNP STRING,

> Low_BA STRING, Low_DAL STRING, Low_HON STRING, Low_TRI STRING, Low_UNP STRING,

> Open_BA STRING, Open_DAL STRING, Open_HON STRING, Open_TRI STRING, Open_UNP STRING,

> Volume_BA STRING, Volume_DAL STRING, Volume_HON STRING, Volume_TRI STRING, Volume_UNP STRING)

> ROW FORMAT DELIMITED

> FIELDS TERMINATED BY ‘,’

> STORED AS TEXTFILE

> LOCATION ‘/user/hive/warehouse/DM_industrials’;

STORING DATA INTO HIVE DATA WAREHOUSE

6. To verify the data
   - hive > Select * from industrials_table limit 5;

```
80.1499939      330.6300049     59.29999924     177     71.87000275     181.3800
049     325.7099915     58.24000168     175.7599945     71.09999847     179.6000
061     330.5   59.25   176.5599976     71.86000061     181.1499939     45255003
779300  1670100 289800  1493000
2019-12-31      323.833313      58.07914734     176.0471344     71.25269318     1
79.7319794      325.7600098     58.47999954     177     71.59999847     180.7899
933     326.5700073     58.72999954     177.0800018     71.66000366     181.2599
945     323.3200073     58.34000015     175.4600067     71.05000305     179.5399
933     325.4100037     58.49000168     176.4600067     71.37999725     179.9600
067     4958800 2917300 1728900 312900  1588800
2020-01-02      331.3485718     58.63530731     179.8167267     71.81993103     1
81.2033234      333.3200073     59.04000092     180.7899933     72.16999817     1
82.2700043      333.3500061     59.38999939     180.8000031     72.18000031     1
82.3800049      327.7000122     58.45000076     177.1399994     71.55999756     1
79.9700012      328.5499878     58.93000031     177.5   71.61000061     180.9499
969     4544400 4459200 2857400 293200  2444800
2020-01-03      330.7919006     57.66202545     177.8971252     72.24784088     1
79.9208679      332.7600098     58.06000137     178.8600006     72.59999847     1
80.9799957      334.8900146     58.11999893     179.8300018     72.94000244     1
81.1900024      330.2999878     56.90999985     177.4100037     71.55999756     1
78.2899933      330.6300049     57.5    178.3399963     71.62000275     179.5399
933     3875900 9078100 2805200 286500  2344200
Time taken: 4.029 seconds, Fetched: 5 row(s)
hive>
```

STORING DATA INTO HIVE DATA WAREHOUSE

7. Create an internal Hive table.
    - There are a few options of STORED AS format;
        - Text file—All data are stored as raw text using the Unicode standard.
        - Sequence file—The data are stored as binary key/value pairs.
        - RCFile—All data are stored in a column optimized format (instead of row optimized).
        - ORC—An optimized row columnar format that can significantly improve Hive performance.
        - Parquet—A columnar format that provides portability to other Hadoop tools including Hive, Drill, Impala, Crunch, and Pig.

- Creating internal Hive table

  - hive > CREATE EXTERNAL TABLE IF NOT EXISTS industrial

    > (`Date` STRING,

    > Adj Close_BA STRING, Adj Close_DAL STRING, Adj Close_HON STRING, Adj Close_TRI STRING, Adj Close_UNP STRING,

    > Close_BA STRING, Close_DAL STRING, Close_HON STRING, Close_TRI STRING, Close_UNP STRING,

    > High_BA STRING, High_DAL STRING, High_HON STRING, High_TRI STRING, High_UNP STRING,

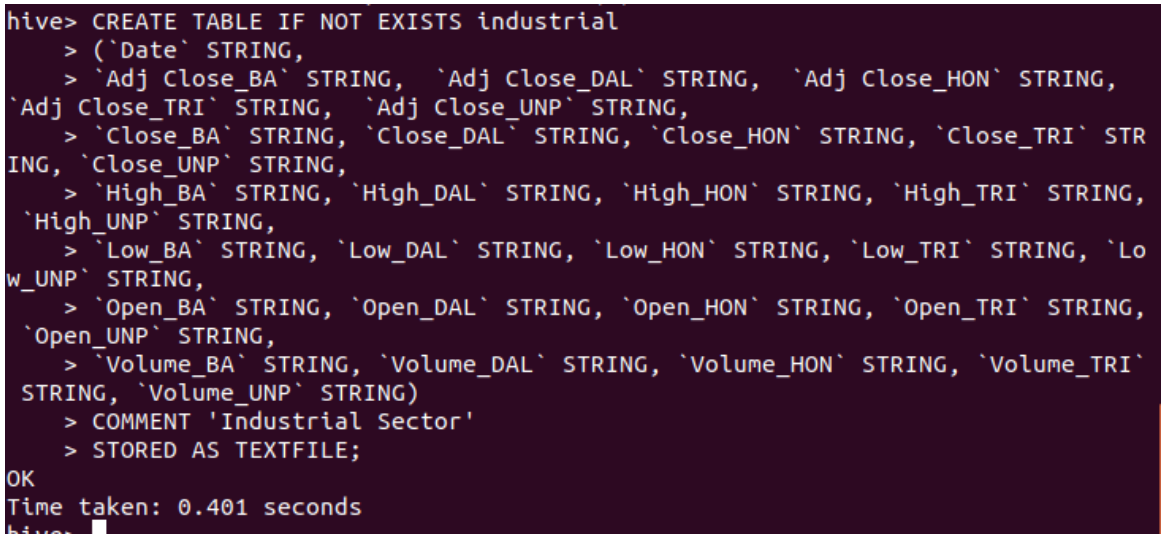
    > Low_BA STRING, Low_DAL STRING, Low_HON STRING, Low_TRI STRING, Low_UNP STRING,

    > Open_BA STRING, Open_DAL STRING, Open_HON STRING, Open_TRI STRING, Open_UNP STRING,

    > Volume_BA STRING, Volume_DAL STRING, Volume_HON STRING, Volume_TRI STRING, Volume_UNP STRING)

    > COMMENT 'Industrial Sector'

    > STORED AS TEXTFILE;



STORING DATA INTO HIVE DATA WAREHOUSE

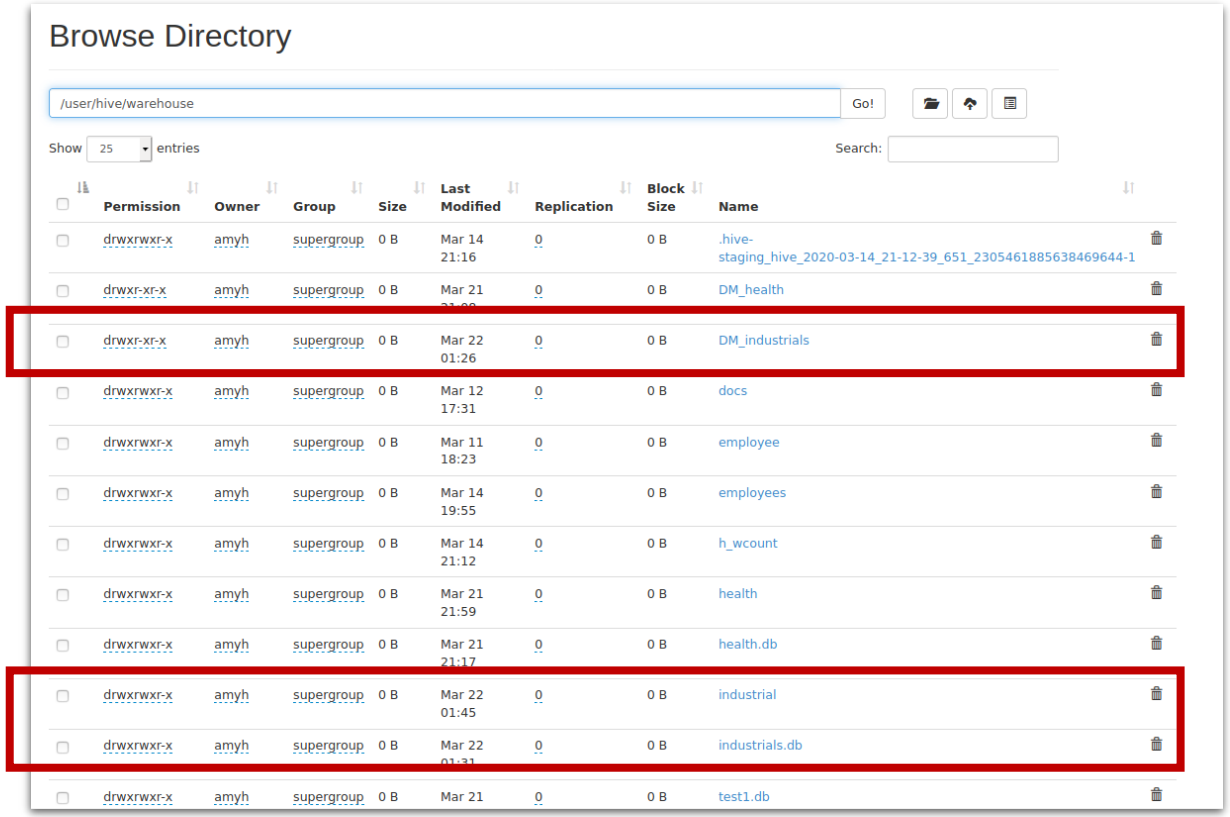
8. Move the data from the external table to the internal table
   - hive > INSERT OVERWRITE TABLE industrial SELECT * FROM industrials_table;

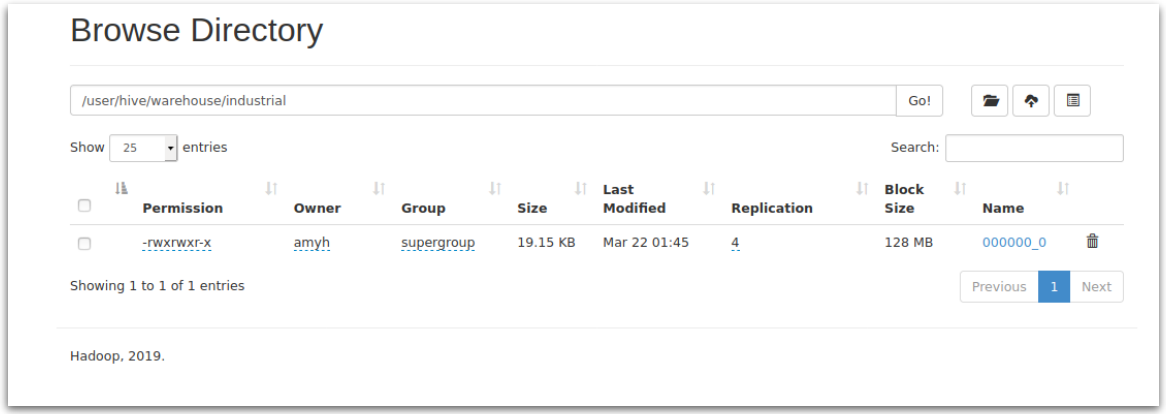
```
hive> INSERT OVERWRITE TABLE industrial SELECT * FROM industrials_table;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
Query ID = amyh_20200322014443_24edd67a-0443-4131-8b5e-3cbcc0a80118
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1584784780100_0003, Tracking URL = http://amyh-VirtualBox:808
8/proxy/application_1584784780100_0003/
Kill Command = /usr/local/hadoop/hadoop-2.10.0/bin/hadoop job  -kill job_1584784
780100_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-03-22 01:45:03,467 Stage-1 map = 0%,  reduce = 0%
2020-03-22 01:45:16,650 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.38 se
c
MapReduce Total cumulative CPU time: 2 seconds 380 msec
Ended Job = job_1584784780100_0003
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/industrial/.h
ive-staging_hive_2020-03-22_01-44-43_771_1261638295482724564-1/-ext-10000
Loading data to table default.industrial
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.38 sec   HDFS Read: 29588 HDFS Write:
19687 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 380 msec
OK
Time taken: 36.355 seconds
hive>
```

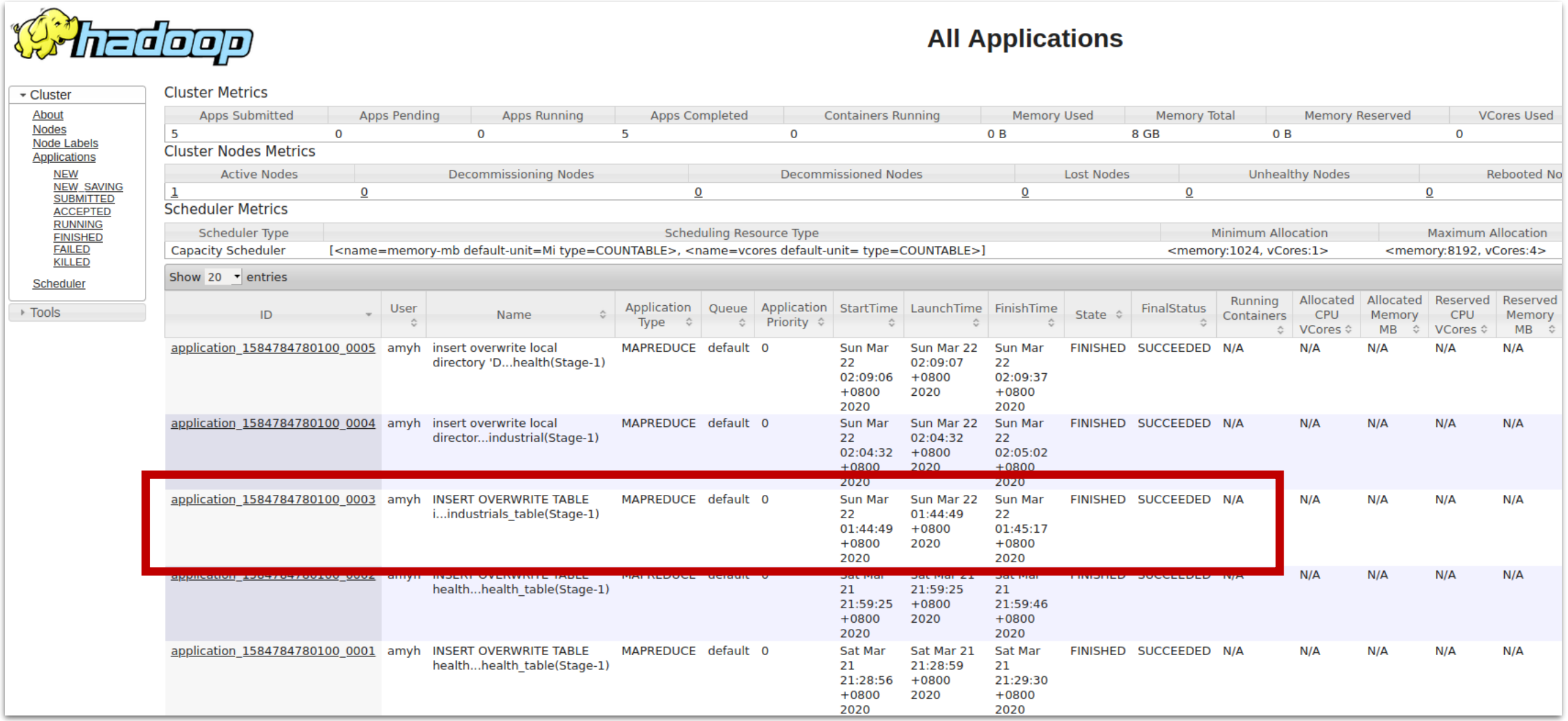
STORING DATA INTO HIVE DATA WAREHOUSE

9. To verify the moved contents;
   - hive > Select * from industrial limit 5;

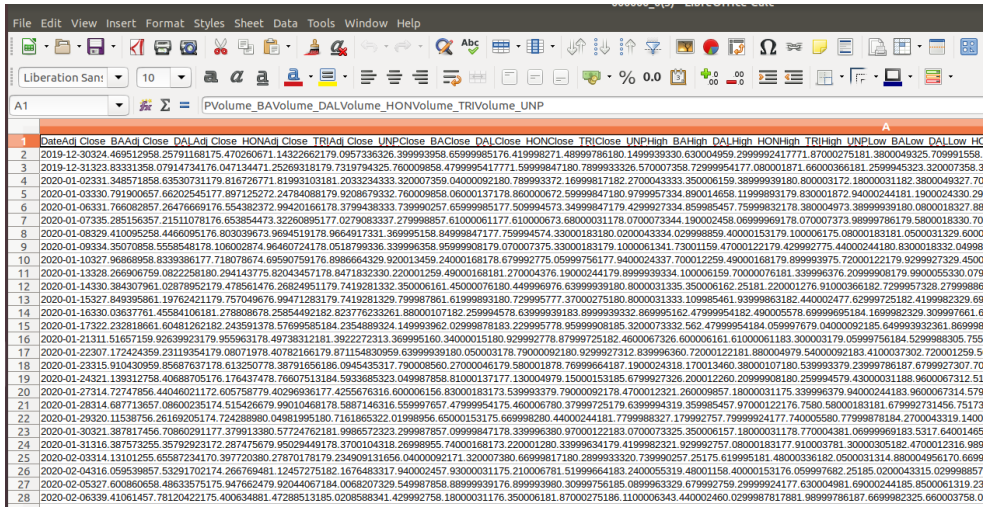
```
hive> Select * from industrial limit 5;
OK
Date    Adj Close_BA    Adj Close_DAL   Adj Close_HON   Adj Close_TRI   Adj Close_UNP
   Close_BA        Close_DAL       Close_HON       Close_TRI       Close_UNP       High
_BA High_DAL        High_HON        High_TRI        High_UNP        Low_BA   Low_DAL Low
_HON Low_TRI Low_UNP Open_BA Open_DAL        Open_HON       Open_TRI Open_UNP        Vo
lume_BA       Volume_DAL      Volume_HON      Volume_TRI      Volume_UNP
2019-12-30      324.4695129     58.25791168     175.4702606     71.14322662     179.0957
336     326.3999939     58.65999985     176.4199982     71.48999786     180.1499939
      330.6300049     59.29999924     177     71.87000275     181.3800049     325.70999
15     58.24000168     175.7599945     71.09999847     179.6000061     330.5   59.25
176.5599976     71.86000061     181.1499939     45255003779300  1670100 289800  1493000
2019-12-31      323.833313      58.07914734     176.0471344     71.25269318     179.7319
794     325.7600098     58.47999954     177     71.59999847     180.7899933     326.57
00073     58.72999954     177.0800018     71.66000366     181.2599945     323.3200073
     58.34000015     175.4600067     71.05000305     179.5399933     325.4100037     58
.49000168     176.4600067     71.37999725     179.9600067     4958800 2917300 1728900 3
12900  1588800
2020-01-02      331.3485718     58.63530731     179.8167267     71.81993103     181.2033
234     333.3200073     59.04000092     180.7899933     72.16999817     182.2700043
      333.3500061     59.38999939     180.8000031     72.18000031     182.3800049     3
27.7000122     58.45000076     177.1399994     71.55999756     179.9700012     328.5499
878     58.93000031     177.5   71.61000061     180.9499969     4544400 4459200 2857400
 293200  2444800
2020-01-03      330.7919006     57.66202545     177.8971252     72.24784088     179.9208
679     332.7600098     58.06000137     178.8600006     72.59999847     180.9799957
      334.8900146     58.11999893     179.8300018     72.94000244     181.1900024     3
30.2999878     56.90999985     177.4100037     71.55999756     178.2899933     330.6300
049     57.5    178.3399963     71.62000275     179.5399933     3875900 9078100 2805200
 286500  2344200
Time taken: 0.476 seconds, Fetched: 5 row(s)
hive>
```


Browse Directory
/user/hive/warehouse
Go!
Show 25 entries
Search:
Permission Owner Group Size Last Modified Replication Block Size Name
drwxrwxr-x amyh supergroup 0 B Mar 14 21:16 0 0 B .hive-staging_hive_2020-03-14_21-12-39_651_2305461885638469644-1
drwxr-xr-x amyh supergroup 0 B Mar 21 0 0 B DM_health
drwxr-xr-x amyh supergroup 0 B Mar 22 01:26 0 0 B DM_industrials
drwxrwxr-x amyh supergroup 0 B Mar 12 17:31 0 0 B docs
drwxrwxr-x amyh supergroup 0 B Mar 11 18:23 0 0 B employee
drwxrwxr-x amyh supergroup 0 B Mar 14 19:55 0 0 B employees
drwxrwxr-x amyh supergroup 0 B Mar 14 21:12 0 0 B h_wcount
drwxrwxr-x amyh supergroup 0 B Mar 21 21:59 0 0 B health
drwxrwxr-x amyh supergroup 0 B Mar 21 21:17 0 0 B health.db
drwxrwxr-x amyh supergroup 0 B Mar 22 01:45 0 0 B industrial
drwxrwxr-x amyh supergroup 0 B Mar 22 0 0 B industrials.db
drwxrwxr-x amyh supergroup 0 B Mar 21 0 0 B test1.db


Browse Directory
/user/hive/warehouse/industrial
Go!
Show 25 entries
Search:
Permission Owner Group Size Last Modified Replication Block Size Name
-rwxrwxr-x amyh supergroup 19.15 KB Mar 22 01:45 4 128 MB 000000_0
Showing 1 to 1 of 1 entries
Previous 1 Next
Hadoop, 2019.


hadoop
All Applications
Cluster
About
Nodes
Node Labels
Applications
NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler
Tools
Cluster Metrics
Apps Submitted | Apps Pending | Apps Running | Apps Completed | Containers Running | Memory Used | Memory Total | Memory Reserved | VCores Used
5 | 0 | 0 | 5 | 0 | 0 B | 8 GB | 0 B | 0
Cluster Nodes Metrics
Active Nodes | Decommissioning Nodes | Decommissioned Nodes | Lost Nodes | Unhealthy Nodes | Rebooted No
1 | 0 | 0 | 0 | 0 | 0
Scheduler Metrics
Scheduler Type | Scheduling Resource Type | Minimum Allocation | Maximum Allocation
Capacity Scheduler | [<name=memory-mb default-unit=Mi type=COUNTABLE>, <name=vcores default-unit= type=COUNTABLE>] | <memory:1024, vCores:1> | <memory:8192, vCores:4>
Show 20 entries
ID | User | Name | Application Type | Queue | Application Priority | StartTime | LaunchTime | FinishTime | State | FinalStatus | Running Containers | Allocated CPU VCores | Allocated Memory MB | Reserved CPU VCores | Reserved Memory MB
application_1584784780100_0005 | amyh | insert overwrite local directory 'D...health(Stage-1) | MAPREDUCE | default | 0 | Sun Mar 22 02:09:06 +0800 2020 | Sun Mar 22 02:09:07 +0800 2020 | Sun Mar 22 02:09:37 +0800 2020 | FINISHED | SUCCEEDED | N/A | N/A | N/A | N/A | N/A
application_1584784780100_0004 | amyh | insert overwrite local director...industrial(Stage-1) | MAPREDUCE | default | 0 | Sun Mar 22 02:04:32 +0800 2020 | Sun Mar 22 02:04:32 +0800 2020 | Sun Mar 22 02:05:02 +0800 2020 | FINISHED | SUCCEEDED | N/A | N/A | N/A | N/A | N/A
application_1584784780100_0003 | amyh | INSERT OVERWRITE TABLE i...industrials_table(Stage-1) | MAPREDUCE | default | 0 | Sun Mar 22 01:44:49 +0800 2020 | Sun Mar 22 01:44:49 +0800 2020 | Sun Mar 22 01:45:17 +0800 2020 | FINISHED | SUCCEEDED | N/A | N/A | N/A | N/A | N/A
health...health_table(Stage-1) | Sat Mar 21 21:59:25 +0800 2020 | Sat Mar 21 21:59:25 +0800 2020 | Sat Mar 21 21:59:46 +0800 2020 | N/A | N/A | N/A | N/A
application_1584784780100_0001 | amyh | INSERT OVERWRITE TABLE health...health_table(Stage-1) | MAPREDUCE | default | 0 | Sat Mar 21 21:28:56 +0800 2020 | Sat Mar 21 21:28:59 +0800 2020 | Sat Mar 21 21:29:30 +0800 2020 | FINISHED | SUCCEEDED | N/A | N/A | N/A | N/A | N/A

ACCESS THE STORED DATA IN HIVE



1. To access the stored data in HDFS;
   - hive > insert overwrite local directory ‘Desktop/industrial_output.csv’
     > row format delimited
     > fields terminated by ‘,’
     > select * from industrial;

```
hive> insert overwrite local directory 'Desktop/industrial_output.csv'
    > row format delimited
    > fields terminated by ','
    > select * from industrial;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future vers
ions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X r
eleases.
Query ID = amyh_20200322020429_b2adf154-4baa-419e-96d7-74c8e7db1a3e
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1584784780100_0004, Tracking URL = http://amyh-VirtualBox:8088/proxy
/application_1584784780100_0004/
Kill Command = /usr/local/hadoop/hadoop-2.10.0/bin/hadoop job  -kill job_1584784780100_
0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-03-22 02:04:48,388 Stage-1 map = 0%,  reduce = 0%
2020-03-22 02:05:02,832 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.97 sec
MapReduce Total cumulative CPU time: 2 seconds 970 msec
Ended Job = job_1584784780100_0004
Moving data to local directory Desktop/industrial_output.csv
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.97 sec   HDFS Read: 28245 HDFS Write: 19610 S
UCCESS
Total MapReduce CPU Time Spent: 2 seconds 970 msec
OK
Time taken: 35.597 seconds
hive>
```

REFERENCE

- https://www.edureka.co/blog/install-hadoop-single-node-hadoop-cluster
- https://www.edureka.co/blog/apache-hive-installation-on-ubuntu
- https://lws-abt5wcf.netdna-ssl.com/blogs/wp-content/uploads/2014/10/Web-Scraping-and-Web-Data-Mining-2.jpg

THANK YOU!