



### Assignment No. 3

Title: Write an application using HiveQL for flight information system which will include

- a. creating, dropping and altering Database tables.
- b. Creating an external Hive table.
- c. load table with data, insert new values and field in the table, join table with Hive.
- d. create index on Flight Information table.
- e. Find the average departure delay per day in 2008.



### Assignment No. 3

Performance (03)	understanding (01)	Regularity (01)	Total (05)	Dated sign of subject teacher
02	01	01	04	<u>S. H. Jadhav</u>

Date of Performance:

Aim: Write an application using HiveQL for flight information system which will include,

- a. creating, dropping and altering database
- b. tables creating an external hive table.
- c. load table with data, insert new values & field in the table, join table with hive.
- d. create Index on flight information table.
- e. find the average departure delay per day 2008

Objective: to learn HiveQL database management system.



## Theory:

Hive: Hive is a data warehouse software project built on top of apache Hadoop. for providing data query and analysis. Hive gives a SQL-like interface to query data stored in various databases and file systems that integrate with hadoop. Traditional SQL queries must be implemented in the Map-Reduce Java API to executes. SQL applications. and queries over distributed data. Hive provides the necessary SQL abstraction. to integrate. SQL like queries into the underlying Java without the need to implement queries. in the low-level Java API. Since most data warehousing applications work with SQL-based querying languages, Hive did portability of SQL-based application to Hadoop.

## Pseudo code:

### Hive Installation steps:

Prerequisites - Hadoop.

sudo su hduser

cd

-- download hive tar and copy to /home/hduser.  
cp /home/student/Desktop/apache-hive-2.1.1.tar.gz  
/home/hduser.

tar -xvzf apache-hive-2.1.1.tar.gz

hduser1@Student-HP-Pro-3330-MT: ~\$ ls.

analog example.desktop inpa0 MUSIC



apache-hive-2.1.1-bin hbase-1.4.1-bin.tar.gz inp40  
apache-hive-2.1.1-bin.tar.gz inp inpg public  
Desktop. inpl input Templates.  
Documents inp2 Manifest.txt videos.  
Downloads inp3 mapred.

) Move hive to /usr/local/hive Folder.

hduser1 @ Student-HP-Pro-3330-MT: ~\$ sudo mv apache-hive-2.1.1-bin /usr/local/hive.

[sudo] password for hduser1:

hduser1 @ student-HP-Pro-3330-MT: ~\$ pwd.

/home/hduser1

hduser1 @ student-HP-Pro-3330-MT: ~\$ sudo gedit ~/.bashrc  
~~export HIVE\_HOME=/usr/local/hive.~~

hduser1 @ student-HP-Pro-3330-MT: ~\$ source ~/.bashrc

Start hadoop

hduser1 @ Student-HP-Pro-3330-MT: ~\$ start.dfs.sh.

18/3/20 13:22:24 WARN util.NativeCodeLoader  
Unable to load native-hadoop library for your platform.  
Using built-in Java classes where applicable

Starting namenodes on [localhost]



localhost : Starting namenode , logging to usr/local/hadoop  
- hduser1 - namenode - student - HP - Pro - 3330 - MT.out  
localhost : Starting datanode, logging to usr/local/  
hadoop/logs/hadoop-hduser1 - standalone - student.  
HP - Pro - 3330 - MT.out.

Starting Secondary namenode [0.0.0.0]

0.0.0.0 : Starting Secondary namenode, logging  
to usr/local/hadoop/logs/hadoop-hduser1 -  
Secondarynamenode - student - HP - Pro - 3330 - MT.out

18/3/20 13:22:43 WARN Util.NativeCodeLoader  
Unable to load Native.-hadoop library for your  
platform . Using built - java . classes where applicable

hduser1@student - HP - Pro - 3330 - MT : ~\$ start yarn.sh  
starting yarn daemons.

Starting resource manager , logging to usr/local/  
hadoop/logs/yarn - hduser1 - resource manager -  
student - HP - Pro - 3330 - MT.out.

localhost : Starting nodemanager, logging to usr/local/  
hadoop/logs/yarn - hduser1 - nodemanager - student -  
HP - Pro - 3330 - MT.out.

hduser1@Student - HP - Pro - 3330 - MT : ~\$ jps

7785 ResourceManager .

7593 Secondary Name Node .

7930 Node Manager

8253 Jps



389 DataNode

245 NameNode.

) Create directories for hive on hadoop.

hduser1@Student-HP-Pro-3330-MT: ~ \$ cd /usr/local/hadoop

hduser1@Student-HP-Pro-3330-MT: /usr/local/hadoop

\$ cd bin

hduser1@Student-HP-Pro-3330-MT: /usr/local/hadoop/bin\$ hdfs dfs -mkdir -p /user/hive/warehouse.

18/03/20 13:24:17 WARN util.NativeCodeLoader:

unable to load native-hadoop library for your platform. Using built-in java classes where applicable

1) Give permission for the directories for hive on hadoop.

hduser1@Student-HP-Pro-3330-MT: /usr/local/hadoop/bin\$ hdfs dfs -chmod -R 777 /tmp.

18/03/20 13:33:09 WARN util.NativeCodeLoader:

unable to load native-hadoop library for your platform using built-in-javaclasses where applicable.

hduser1@Student-HP-Pro-3330-MT: /usr/local/hadoop/bin\$ hdfs dfs -chmod 777 /user/hive/warehouse.

18/03/20 13:33:51 WARN util.NativeCodeLoader: unable to load native-hadoop library for your platform. Using



builtin -java classes where applicable.

chmod: 'Use /hive/warehouse': No such file or directory  
hduser1@Student-HP-Pro-3330-MT:~\$ /usr/local/hadoop/bin\$ hdfs dfs -chmod 777 /tmp/hive  
WARN Util.NativeCodeLoader: Unable to load native-hadoop library for your platform, using  
builtin -java classes where applicable

### 5] Initialize SchemaTool.

hduser1@Student-HP-Pro-3330-MT:~\$ /usr/local/hadoop/bin\$ cd

hduser1@Student-HP-Pro-3330-MT:~\$ /usr/local/hadoop\$ cd

hduser1@Student-HP-Pro-3330-MT:~\$ /usr/local/hive\$ ls

bin example jdbc LICENSE README.txt scripts-  
conf hcatalog lib NOTICE RELEASE-NOTES.txt

hduser1@Student-HP-Pro-3330-MT:~\$ /usr/local/hive\$ cd bin

hduser1@Student-HP-Pro-3330-MT:~\$ /usr/local/hive/bin\$ ./schematool -initSchema -dbType derby



SLF4J: class path contains multiple SLF4J binding

SLF4J: Found binding in [for jar:file:/usr/local/hive/lib/log4j-slf4j-impl 2.4.1.jar!/org/slf4j/impl/staticLoggerBinder.class]

SLF4J: Found binding in jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/staticLoggerBinder.class]

SLF4J: See [http://www.slf4j.org/codes.html#multiple\\_binding](http://www.slf4j.org/codes.html#multiple_binding) for an explanation.

SLF4J: Actual binding of type [org.apache.logging.slf4j.Log4jLoggerFactory]

~~Metastore connection URL: jdbc:derby:;databaseName=metastore\_db;create=true.~~

Metastore connection Driver: org.apache.derby.jdbc.  
Embedded Driver

Metastore connection user: APP

Starting metastore schema initialization to 2.1.0

Initialization script hive-schema-2.1.0.derby.sql

Initialization script hive-6 completed.

SchemaTool completed.



nduser1@Student-HP-Pro-3330-MT: /usr/local/hive  
/bin\$ ls .

beeline ext hive-config.cmd hqlsql metatool  
beeline.cmd hive hive-config.sh hpsql.cmd schematool  
derby.log hive.cmd hiveserver2 metastore-dh

G) Start Hive

nduser1@Student-HP-Pro-3330-MT: /usr/local/hive  
bin\$. /hive

SLF4J: Class path contains multiple SLF4J binding.

SLF4J: Found binding in [jar:file:/usr/local/hivellib/  
log4j-SLF4J-impl-2.4.1.jar!/org/slf4j/impl/Static  
LoggerBinder.class]

SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/  
hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/  
impl/StaticLoggerBinder.class]

SLF4J: Found binding in [jar:1

SLF4J: See <http://www.slf4j.org/codes.html#multiple>  
bindings for an explanation.

SLF4J: Actual binding is of type [org.apache.logging.slf4j/  
.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/usr/



3) update bashrc

sudo gedit ~/.bashrc

11.bashrc

export HBASE\_HOME = /usr/local/hbase

export PATH = \$PATH : \$HBASE\_HOME/bin

7) check hosts folder

sudo gedit /etc/hosts

8) update env in conf

cd /usr/local/hbase/conf

sudo gedit hbase-env.sh

11 hbase-env.sh

export JAVA\_HOME = /usr/lib/jvm/java-7-openjdk-i386

9) update hbase-site.xml

sudo gedit hbase-site.xml

hbase-site.xml

<property>

<name> hbase.rootdir </name>

<value> hdfs://localhost:9000/hbase </value>

</property>

<property>

<name> hbase.rootdir </name>

<value> 6000 </value>



local/hive/lib/hive-common-2.1.1.jar!/hive-log4j2-  
properties Async: true.

Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine

hive>

hbase (NFS9) single node installation:

Prerequisite - Hadoop

1) Login with hduser

sudo su hduser

cd

2) Start Hadoop

start-dfs.sh

start-yarn.sh

jps

hadoop version

3) download hbase tar 1.4.1, copy to desktop and copy to /home/hduser

sudo cp hbase-1.4.1-bin.tar.gz /home/hduser/

4) Extract the HBase

tar -xvzf hbase-1.4.1-bin.tar.gz

ls

5) move hbase to /usr/local/hbase Folder.

sudo mv hbase-1.4.1 /usr/local/hbase



<property>

<property>

<name> hbase.cluster.distributed </name>

<value> true </value>

<property>

<property>

<name> hbase.zookeeper.property.datadir </name>

<value> /usr/local/zookeeper </value>

<property>

<property>

<name> hbase.zookeeper.property.maxClientCnxns

<name>

<value> 35 </value>

<property>

o) remove zookeeper

cd /usr/local

rm -rf zookeeper/

1) edit region servers

cd /usr/local/hbase/conf

Sudo gedit regionserver.

2) start hbase

cd ..

cd bin



1) start - hbase.sh

open your browser logon to  
localhost : 16010

3) start shell - table creation and data adding

./hbase .shel

hbase > create 'employee', 'salary'

hbase > list

hbase > put 'employee', '001', 'salary:name', 'RMB'

hbase > scan 'employee'

hbase > put 'employee', '001', 'salary:loc', 'Nashik'

hbase > scan 'employee'

hbase > exit.

4) stop hbase

./stop-hbase.sh

5) Stop all processes

stop-all.sh

Conclusion: Apache Hive helps with querying and managing large dataset real fast. It is an ETL tool for Hadoop ecosystem. HBase is the Hadoop database, a distributed, scalable, big data store, big data store, use apache HBase when you need random, real time read/write access to your Big Data.

MA