Aim : "Crud operations and more using hive query language."

In Hadoop Eco System, out of many, one more matured component is Hive. Hive is mainly used for performing SQL on data residing distributed.

For the workforce having skillset SQL and not programming, Hive is a handy tool to perform data analytics with Hadoop Framework. The query language supported by Hive is also sometimes referred as HiveQL or HQL.

Start Hadoop framework first because hive writes the relations in Hadoop Distributed File System which is also called 'Warehouse' location.

Below required steps are already performed as part of hive setup on lab computers.

Download and extract hive tarbal to /opt/hive.

Create below required folder for hive within hdfs and make group writable:

hadoop fs -mkdir -p /tmp

hadoop fs -mkdir -p /user/hive/warehouse

hadoop fs -chmod g+w /tmp

hadoop fs -chmod g+w /user/hive/warehouse

add HIVE_HOME=<hive-install-dir> into bash profile ~/.bash_profile

Hive is internally bundled with derby db jar in lib folder. The durbyDB is used for metadata storage. If you want to configure mysql or other perform appropriate hive-site.xml configuration.

To be able to run hive from any locations on terminal and still use the same metadata, let us fix the metadata location by configuring as below:

/opt/hive/conf/hive-site.xml

```
<configuration>
<property>
<name>javax.jdo.option.ConnectionURL</name>
<value>jdbc:derby:;databaseName=/opt/hive/metastore_db;create=true</value>
</property>
</configuration>
```

Set Hadoop and java home within conf/hive-env.sh

i.e.

HADOOP_HOME=/opt/hadoop

JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_151.jdk/Contents/Home

Run below command to instatiate derby db default:

schematool -dbType derby -initSchema

Run Hive CLI

1. Start HDFS.

start-dfs.sh

2. Start Yarn.

start-yarn.sh

3. Start History Manager

start-historyserver.sh

Verify using jps command.

Run hive command, type in 'show tables;' If it show OK, you are Successful.

Note that RunJar named process is representing the Hive.

```
Jigar-Pandyas-MacBook:hadoop JigarPandya$ jps
4594 NameNode
7714 JobHistoryServer
6389 RunJar
7639 NodeManager
7544 ResourceManager
4681 DataNode
10319 Jps
4783 SecondaryNameNode
Jigar-Pandyas-MacBook:hadoop JigarPandya$
```

Note that if not chosen any particular database, Hive does use default database.

```
hive (default)> create database try1;
OK
Time taken: 0.375 seconds
hive (default)> use try1;
OK
Time taken: 0.05 seconds
hive (try1)>
```

Type 'quit;' to exit.

Hive internally generates Map-Reduce jobs for queries. Follow through below to practice the same.

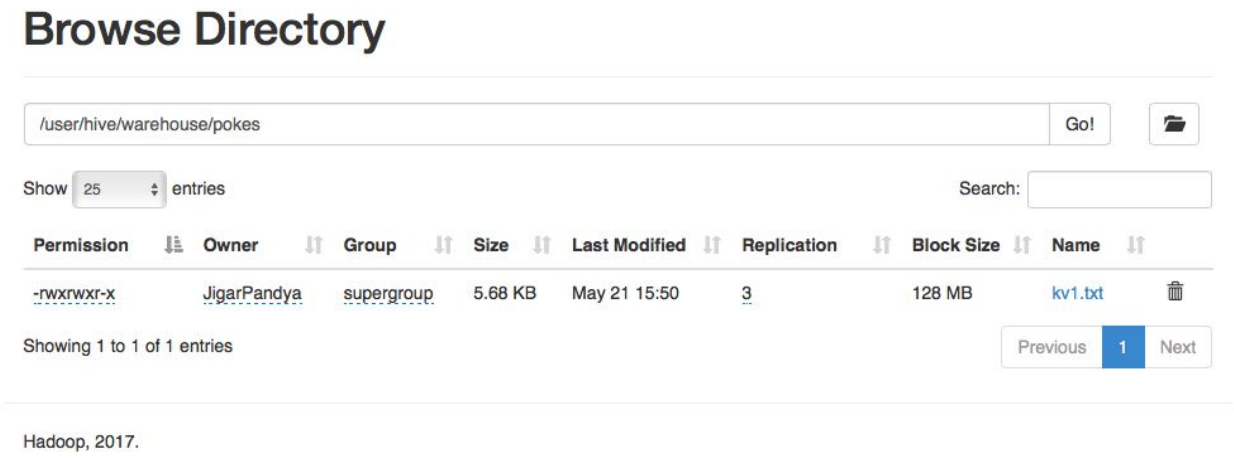
1. create, load ad select

hive (default)> CREATE TABLE pokes (foo INT, bar STRING);

hive (default)>

LOAD DATA LOCAL INPATH '/opt/hive/examples/files/kv1.txt' OVERWRITE INTO TABLE pokes;

```
hive (default)> select * from pokes;
OK
238     val_238
86      val_86
311     val_311
27      val_27
165     val_165
409     val_409
```



http://localhost:50070/explorer.html#/user/hive/warehouse/pokes

Note that the file is stored from native file system to Hive Warehouse which is within HDFS.

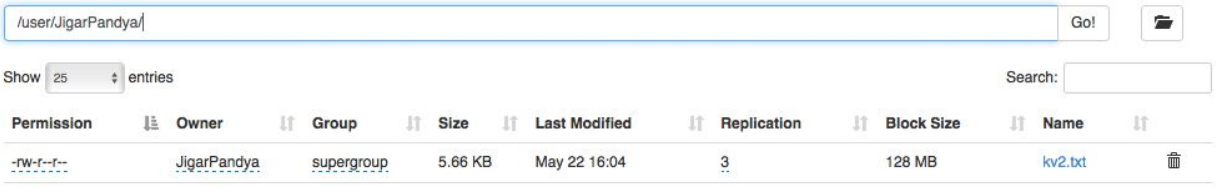
2. Analytics

2.1 hive (default)> CREATE TABLE invites (foo INT, bar STRING);

2.2 Put file from local file system to HDFS using terminal command:

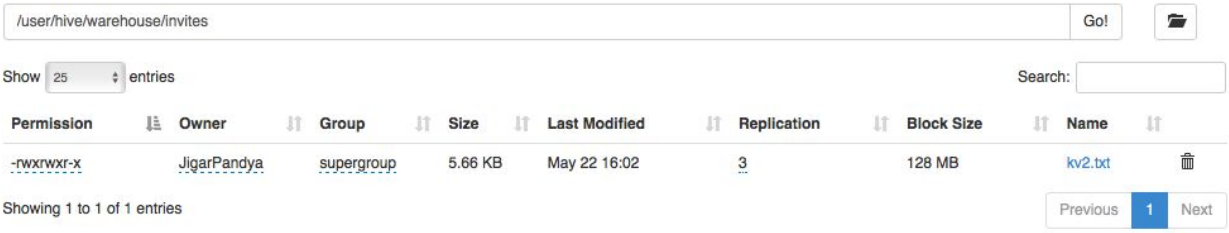
hadoop dfs -put /opt/hive/examples/files/kv2.txt /user/hadoop/kv2.txt

http://localhost:50070/explorer.html#/user/hadoop/



2.3 hive (default)> LOAD DATA INPATH '/user/hadoop/kv2.txt' OVERWRITE INTO TABLE invites;

http://localhost:50070/explorer.html#/user/hive/warehouse/invites



2.4

Important: Loading data from HDFS to Hive WareHouse (Again internal to HDFS) will result in moving the file/directory within HDFS. You might want to copy if original file/directory also needed to be retained at original place.

http://localhost:50070/explorer.html#/user/hadoop/

file named kv2.txt is no more at this location /user/hadoop within HDFS.

```
hive (default)> SELECT a.bar, count(*) FROM invites a WHERE a.foo > 0 GROUP BY a.bar;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using
 execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = JigarPandya_20190522161807_ba1a2638-adb0-4543-90b7-1288b35668bd
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1558435464009_0005, Tracking URL = http://localhost:8088/proxy/application_1558435464009
Kill Command = /opt/hadoop/bin/hadoop job  -kill job_1558435464009_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2019-05-22 16:25:44,730 Stage-1 map = 0%,  reduce = 0%
2019-05-22 16:26:24,872 Stage-1 map = 100%,  reduce = 0%
2019-05-22 16:26:41,264 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_1558435464009_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   HDFS Read: 14553 HDFS Write: 7121 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
val_100 1
val_101 2
val_102 1
val_103 1
val_105 3
val_106 2
val_107 1
val_11  1
```

2.5

hive (default)> SELECT a.bar, count(*) FROM invites a WHERE a.foo > 0 GROUP BY a.bar;

```
val_90  3
val_92  1
val_94  3
val_95  1
val_98  2
Time taken: 516.016 seconds, Fetched: 320 row(s)
hive (default)>
```

References :

1. https://hive.apache.org

2. https://cwiki.apache.org/confluence/display/Hive/GettingStarted

Exercise:

1. Retrieve the data set. (http://files.grouplens.org/datasets/movielens/ml-100k.zip)

2. Run below HQL

```
hive (default)>CREATE TABLE u_data (
userid INT,
movieid INT,
rating INT,
unixtime STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE;
```

3. Load the data from local file u.data to table u_data.

hive (default)> LOAD DATA LOCAL INPATH '<path>/u.data' OVERWRITE INTO TABLE u_data;

4. Find total number of records u_data table.

hive (default)> Select count(*) from u_data;

```
hive (default)> select count(*) from u_data;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
 execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = JigarPandya_20190521160647_534e4964-0291-4545-801c-65308b58e603
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1558435464009_0001, Tracking URL = http://10.10.25.171:8088/proxy/application_1558435464009_0001/
Kill Command = /opt/hadoop/bin/hadoop job  -kill job_1558435464009_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2019-05-21 16:28:32,285 Stage-1 map = 0%,  reduce = 0%
2019-05-21 16:29:10,079 Stage-1 map = 100%,  reduce = 0%
2019-05-21 16:29:30,261 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_1558435464009_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   HDFS Read: 1987048 HDFS Write: 106 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
100000
Time taken: 1376.77 seconds, Fetched: 1 row(s)
hive (default)>
```

5. Find weekday and corresponding count group by weekday.

Create weekday_mapper.py:

import sys

import datetime

```
for line in sys.stdin:
line = line.strip()
userid, movieid, rating, unixtime = line.split('\t')
weekday = datetime.datetime.fromtimestamp(float(unixtime)).isoweekday()
print '\t'.join([userid, movieid, rating, str(weekday)])
```

```
hive (default)> add FILE weekday_mapper.py;
```

```
hive (default)> INSERT OVERWRITE TABLE u_data_new
SELECT
TRANSFORM (userid, movieid, rating, unixtime)
USING 'python weekday_mapper.py'
AS (userid, movieid, rating, weekday)
FROM u_data;
```

```
hive (default)> SELECT weekday, COUNT(*)
```

FROM u_data_new

GROUP BY weekday;

```
hive (default)> SELECT weekday, COUNT(*)
              > FROM u_data_new
              > GROUP BY weekday;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
 execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = JigarPandya_20190521171306_a5a51582-cf62-40f1-8668-6240e32c0225
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1558435464009_0003, Tracking URL = http://10.10.25.171:8088/proxy/application_1558435464009_0003/
Kill Command = /opt/hadoop/bin/hadoop job  -kill job_1558435464009_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2019-05-21 17:13:24,219 Stage-1 map = 0%,  reduce = 0%
2019-05-21 17:13:47,027 Stage-1 map = 100%,  reduce = 0%
2019-05-21 17:14:03,144 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_1558435464009_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   HDFS Read: 1187539 HDFS Write: 227 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
1       12510
2       14054
3       14834
4       14519
5       15478
6       16702
7       11903
Time taken: 58.694 seconds, Fetched: 7 row(s)
```

http://localhost:8088/cluster/apps

application_1558435464009_0003	JigarPandya	SELECT weekday, COUNT(*) FROM u_da...weekday(Stage-1)	MAPREDUCE	default	0	Tue May 21 17:13:09 +0550 2019	Tue May 21 17:14:03 +0550 2019	FINISHED	SUCCEEDED	N/A
application_1558435464009_0002	JigarPandya	INSERT OVERWRITE TABLE u_data_new S...u_data(Stage-1)	MAPREDUCE	default	0	Tue May 21 17:08:58 +0550 2019	Tue May 21 17:10:07 +0550 2019	FINISHED	SUCCEEDED	N/A
application_1558435464009_0001	JigarPandya	select count(*) from u_data(Stage-1)	MAPREDUCE	default	0	Tue May 21 16:16:11 +0550 2019	Tue May 21 16:29:38 +0550 2019	FINISHED	SUCCEEDED	N/A