

**Aim:** Implement various operations on Hive (Create, Insert, Update)

**Objective:** Study, Understand and Implement various operations on data using Hive.

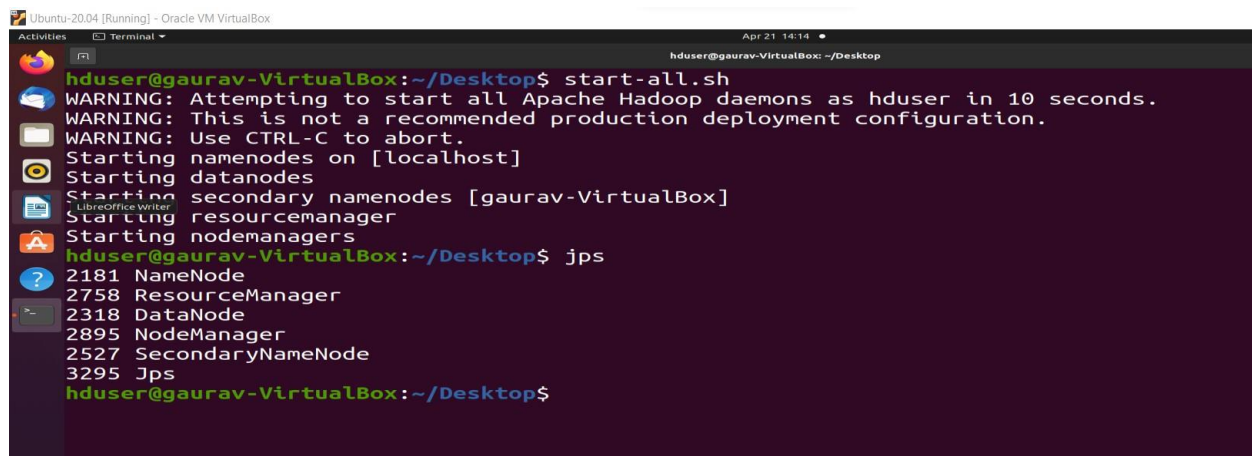
**Theory:**

**Hive:**

Apache Hive is a data warehouse and an ETL tool which provides an SQL-like interface between the user and the Hadoop distributed file system (HDFS) which integrates Hadoop. It is built on top of Hadoop. It is a software project that provides data query and analysis. It facilitates reading, writing and handling wide datasets that stored in distributed storage and queried by Structure Query Language (SQL) syntax. It is not built for Online Transactional Processing (OLTP) workloads. It is frequently used for data warehousing tasks like data encapsulation, Ad-hoc Queries, and analysis of huge datasets. It is designed to enhance scalability, extensibility, performance, fault-tolerance and loose-coupling with its input formats.

**Implementation:**

1. At the most we need to start the cluster using the start-all.sh script further check all daemons are running using the jps command



```
hduser@gaaurav-VirtualBox:~/Desktop$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [gaaurav-VirtualBox]
Starting resourcemanager
Starting nodemanagers
hduser@gaaurav-VirtualBox:~/Desktop$ jps
2181 NameNode
2758 ResourceManager
2318 DataNode
2895 NodeManager
2527 SecondaryNameNode
3295 Jps
hduser@gaaurav-VirtualBox:~/Desktop$
```

2. Once we have the cluster running we also need to download the tar.gz file which we will be using to install the hive in our system. To install the hive we will be just extracting the data in the file tar -xzf <tar file name>

```
hduser@gaurav-VirtualBox:~/hive$ ls
apache-hive-3.1.2-bin.tar.gz
hduser@gaurav-VirtualBox:~/hive$ tar -xvzf apache-hive-3.1.2-bin.tar.gz
apache-hive-3.1.2-bin/LICENSE
apache-hive-3.1.2-bin/NOTICE
apache-hive-3.1.2-bin/RELEASE_NOTES.txt
apache-hive-3.1.2-bin/binary-package-licenses/asm-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/com.google.protobuf-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/com.ibm.icu.icu4j-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/com.sun.jersey-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/com.thoughtworks.paranamer-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/javax.transaction.transaction-api-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/javolution-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/jline-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/NOTICE
apache-hive-3.1.2-bin/binary-package-licenses/org.abego.treelayout.core-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/organtlr-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/organtlr/antlr4-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/organtlr.stringtemplate-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/org.codehaus.janino-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/org.jamon.jamon-runtime-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/org.jruby-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/org.mozilla.rhino-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/org.slf4j-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/sqlite-LICENSE
apache-hive-3.1.2-bin/examples/files/2000-sqls-data.csv
```

3. Now folder will be visible for Apache-hive in which we have the bin where the executable for the hive is present which will be using to get the hive shell

```
hduser@gaurav-VirtualBox:~/hive$ cd apache-hive-3.1.2-bin/
hduser@gaurav-VirtualBox:~/hive/apache-hive-3.1.2-bin$ ls
bin          conf          hcatalog      lib          NOTICE      scripts
binary-package-licenses  examples      jdbc          LICENSE      RELEASE_NOTES.txt
hduser@gaurav-VirtualBox:~/hive/apache-hive-3.1.2-bin$ cd bin
hduser@gaurav-VirtualBox:~/hive/apache-hive-3.1.2-bin/bin$ ls
beeline      ext          hive-config.sh  hplsql      metastore_db  schematool
derby.log    hive         hiveserver2     init-hive-dfs.sh  metatool
hduser@gaurav-VirtualBox:~/hive/apache-hive-3.1.2-bin/bin$ ./hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hduser/hive/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org.slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.2.1/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org.slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 92ea44fa-7aa4-4ea4-bd07-350865090fbc

Logging initialized using configuration in jar:file:/home/hduser/hive/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.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 (i.e. spark, tez) or using Hive 1.X releases.
hive>
>
>
>
```

4. Once we get the hive shell ready we will be using the HQL command first we will be creating our database using the use command which will switch us to the new database and if not existed it will create and then switch.

**\$create database <database\_name>**

**\$use <database\_name>**

```
Ubuntu-20.04 [Running] - Oracle VM VirtualBox
Activities Terminal Apr 21 14:33
hduser@gaaurav-VirtualBox: ~/hive/apache-hive-3.1.2-bin/bin

hive> show databases;
OK
default
Time taken: 1.105 seconds, Fetched: 1 row(s)
hive> create database test;
OK
Time taken: 0.328 seconds
hive> show databases;
OK
default
test
Time taken: 0.132 seconds, Fetched: 2 row(s)
hive> use test;
OK
Time taken: 0.094 seconds
hive> CREATE TABLE STUDENT
> (
>   STD_ID INT,
>   STD_NAME STRING,
>   AGE INT,
>   ADDRESS STRING
> )
> ;
OK
Time taken: 1.226 seconds
hive>
> ;
hive>
> ;
hive> show tables;
OK
student
Time taken: 0.071 seconds, Fetched: 1 row(s)
hive> █
```

5. We can ask to show the tables using the `show tables` command further we can also create the table using the `create table <table_name> (<column_name> <type of value>, ...)` query.
6. We can now insert necessary values in the table using the `insert values` command.

```
hives/apache-hive-3.1.2-bin/bin
hive> INSERT INTO TABLE STUDENT VALUES
> (101,'Gaurav',20,'Nashik'),
> (102,'Nehal',21,'Vashim'),
> (103,'Chinmay',20,'Katole'),
> (104,'Avishkar',21,'Mumbai');
Query ID = hduser_20220421143707_42ba4098-3338-40e9-a3c4-63538abf22ad
Total Jobs = 3
Launching Job 1 out of 3
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_1650530620022_0001, Tracking URL = http://gaurav-VirtualBox:8088/proxy/application_1650530620022_0001/
Kill Command = /usr/local/hadoop-3.2.1/bin/mapred job -kill job_1650530620022_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-04-21 14:37:27,988 Stage-1 map = 0%, reduce = 0%
2022-04-21 14:37:36,629 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.35 sec
2022-04-21 14:37:44,129 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 7.02 sec
MapReduce Total cumulative CPU time: 7 seconds 20 msec
Ended Job = job_1650530620022_0001
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/test.db/student/.hive-staging_hive_2022-04-21_14-37-07_363_6023316509553931186-1/-ext-10000
Loading data to table test.student
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 7.02 sec HDFS Read: 18209 HDFS Write: 448 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 20 msec
OK
Time taken: 39.113 seconds
hive>
```

```
hives/apache-hive-3.1.2-bin/bin
hive> SELECT * FROM STUDENT;
OK
101      Gaurav      20      Nashik
102      Nehal       21      Vashim
103      Chinmay     20      Katole
104      Avishkar    21      Mumbai
Time taken: 0.428 seconds, Fetched: 4 row(s)
hive>
```

**Conclusion/ outcome:** Thus we have successfully performed and understood the create, insert, . operations using the HQL in Apache hive