

Name: Ankur Kothawade

Roll No:2193046

Enroll No: MITU19BTCS0151

Class: CSE LY IS 1

Step 1: Start all your Hadoop Daemon

```
start-dfs.sh          # this will start namenode, datanode and secondary namenode
start-yarn.sh         # this will start node manager and resource manager
jps                  # To check running daemons
```

Step 2: Launch hive from terminal

```
hive
```

```
dikshant@dikshant:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/dikshant/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:/home/dikshant/hadoop/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 = f451277a-8531-45b6-aa5e-7d04a12e8d49

Logging initialized using configuration in jar:file:/home/dikshant/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true
Hive Session ID = d5f43ce9-9b53-48f5-bace-b25b7835afcd
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> █
```

In hive with DML statements, we **can add data to the Hive table in 2 different ways.**

- Using INSERT Command
- Load Data Statement

1. Using INSERT Command

Syntax:

```
INSERT INTO TABLE <table_name> VALUES (<add values as per column entity>);
```

```
hive> CREATE TABLE IF NOT EXISTS student(  
  > Student_Name STRING,  
  > Student_Rollno INT,  
  > Student_Marks FLOAT)  
  > ROW FORMAT DELIMITED  
  > FIELDS TERMINATED BY ',';  
OK  
Time taken: 2.086 seconds  
hive> show tables in default;  
OK  
student  
Time taken: 0.268 seconds, Fetched: 1 row(s)  
hive> █
```

INSERT Query:

```
INSERT INTO TABLE student VALUES ('Dikshant',1,'95'),('Akshat', 2 , '96'),('Dhruv
```

```
hive> INSERT INTO TABLE student VALUES ('Dikshant',1,'95'),('Akshat', 2 , '96'),  
('Dhruv',3,'90');  
Query ID = dikshant_20201106121659_f5dfa694-f552-4b7a-a64b-4f3804213ab8  
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:
```

We can check the data of the *student* table with the help of the below command.

```
SELECT * FROM student;
```

```
hive> SELECT * FROM student;
OK
Dikshant      1      95.0
Akshat 2      96.0
Dhruv 3      90.0
Time taken: 0.162 seconds, Fetched: 3 row(s)
hive> █
```

2. Load Data Statement

Hive provides us the functionality to load pre-created table entities either from our local file system or from HDFS. The *LOAD DATA* statement is used to load data into the hive table.

Syntax:

```
LOAD DATA [LOCAL] INPATH '<The table data location>' [OVERWRITE] INTO TABLE <table>
```

```
cd /home/dikshant/Documents // To change the directory
touch data.csv // use to create data.csv file
nano data.csv // nano is a linux command line editor to edit file
cat data.csv // cat is used to see content of file
```

```
dikshant@dikshant:~$ cd /home/dikshant/Documents/
dikshant@dikshant:~/Documents$ touch data.csv
dikshant@dikshant:~/Documents$ nano data.csv
dikshant@dikshant:~/Documents$ cat data.csv
Ganesh,4,85
Chandan,5,65
Bhavani,6,87
dikshant@dikshant:~/Documents$ █
```

LOAD DATA to the student hive table with the help of the below command.

```
LOAD DATA LOCAL INPATH '/home/dikshant/Documents/data.csv' INTO TABLE student;
```

```
hive> LOAD DATA LOCAL INPATH '/home/dikshant/Documents/data.csv' INTO TABLE student;
Loading data to table default.student
OK
Time taken: 2.617 seconds
hive> |
```

```
SELECT * FROM student;
```

```
hive> SELECT * FROM student;
OK
Dikshant      1      95.0
Akshat 2      96.0
Dhruv 3       90.0
Ganesh 4      85.0
Chandan 5     65.0
Bhavani 6     87.0
Time taken: 1.24 seconds, Fetched: 6 row(s)
hive> |
```