

# Big Data Management

## Programming Assignment #4

### Hive

#### Login and Setup:

Step 1: Login into Hadoop Cluster using ssh command and password



```
achyuthanagaveti — ssh anagave@hadoop-nn001.cs.okstate.edu — 141x24
Last login: Mon May 3 22:30:44 on ttys002

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) Achyuthas-MacBook-Air:~ achyuthanagaveti$ ssh anagave@hadoop-nn001.cs.okstate.edu
anagave@hadoop-nn001.cs.okstate.edu's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-70-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Last login: Mon May 3 22:30:57 2021 from 10.200.218.71
```

Step 2: Create a Metastore db as shown in the screenshot

```
anagave@hadoop-nn001:~$ mv metastore_db metastore_db_22
```

Step 3: Initiate the derby Schema as shown in the screenshot

```
anagave@hadoop-nn001:~$ schematool -initSchema -dbType derby
[SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/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-3.3.0/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]
Metastore connection URL:      jdbc:derby;;databaseName=metastore_db;create=true
Metastore Connection Driver :  org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:     APP
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.derby.sql
```

Step 4: Start Hive shell using hive command

```
anagave@hadoop-nn001:~$ hive
[SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/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-3.3.0/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 = d282388d-7971-4652-9e3c-5132d45d0082
```

```
Hive Session ID = 14569b6a-2d33-416d-8ebc-75f262e03a84
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.
```

---

## Creating and Description of Database:

**Task 1:** Create a table using HIVE The table must contain the following types of fields: string, MAP, ARRAY, STRUCT and it must be partitioned.

Step 1: I have created a Database and named it as “achyu”

```
hive> create database achyu location '/user/anagave/achyu';
OK
Time taken: 0.731 seconds
```

Step 2: To check the database

```
hive> show databases;
OK
achyu
default
Time taken: 0.281 seconds, Fetched: 2 row(s)
```

Step 3: To Use the Database

```
hive> use achyu;
OK
Time taken: 0.072 seconds
```

Step 4: Create two tables

Table 1:

Table name : MASCOT

Number of Columns : 2 Columns

Number of Rows : 6 Rows

Names of Columns : “id” which is an Integer

“mascot” which is a String, which contains mascot of each college

```
hive> create table mascot(id int, mascot string);
OK
Time taken: 0.822 seconds
```

```
hive> select * from mascot;
OK
101      Pistol Pete
102      Brutus Buckeye
103      Raider Red
104      WuShock
105      Beaver
106      Sparty
```

### Table 2:

Table name : COLLEGE

Number of Columns : 5 Columns

Number of Rows : 6 Rows

Names of Columns : 1. "id" which is an Integer

2. "name" which is a String

3. "phones" which is an Integer

4. "place" which is a Struct type contains two parameters one is city and other is state.

5. "corecourses" which is a MAP type, Which contains both string and integer

6. The table is PARTITIONED by Country name which is a String

```
[hive> create table College (id int, name string, phones array<int>, place struct<city:string,State:string>, corecourses Map<string,int>) PARTITIONED BY (Country String);
OK
```

### Table 2: College

```
[hive> select * from College;
OK
103   Ariana  [1215678903,1217504890] {"city":"Lubbock","state":"Texas"} {"OS":2,"DS":2} US
102   Lee    [1215678903,1217504890] {"city":"Columbus","state":"Ohio"} {"DS":2,"FLT":2} US
106   Tina   [1215678903,1217504890] {"city":"East Lansing","state":"Michigan"} {"COA":2,"DS":2} US
101   Kim    [2127893564,2121123546] {"city":"Stillwater","state":"Oklahoma"} {"COA":2,"OS":2} US
105   Lee    [212128978,212976542] {"city":"PASADENA","state":"California"} {"FLT":2,"DS":2} US
104   John   [212120201,212456392] {"city":"Wichita","state":"Kansas"} {"OS":2,"DS":2} US
Time taken: 0.163 seconds, Fetched: 6 row(s)
```

### Step 4: Show tables

```
[hive> show tables;
OK
college
mascot
Time taken: 0.074 seconds, Fetched: 2 row(s)
```