

1.SQOOP

```
# STEP 1: START HADOOP SERVICES
sudo service hadoop-hdfs-namenode start
sudo service hadoop-hdfs-datanode start
sudo service hadoop-yarn-resourcemanager start
sudo service hadoop-yarn-nodemanager start
sudo service hadoop-mapreduce-historyserver start

# STEP 2: START MYSQL AND SETUP DATABASE
sudo service mysqld start
mysql -u root -pcloudera

CREATE DATABASE sakec;
USE sakec;
CREATE TABLE stud(id INT, name VARCHAR(20), marks INT);
INSERT INTO stud VALUES (1,"Abhinav",23);
INSERT INTO stud VALUES (2,"Aditya",24);
INSERT INTO stud VALUES (3,"Ayush",25);
SELECT * FROM stud;
EXIT;

sqoop version

# STEP 3: SQOOP IMPORT (MYSQL ➔ HDFS)
sqoop import \
--connect jdbc:mysql://localhost/sakec \
--username root \
--password cloudera \
--table stud \
--m 1 \
--target-dir /user/cloudera/student_import

# STEP 5: VERIFY IMPORTED DATA IN HDFS
hadoop fs -ls /user/cloudera/student_import
hadoop fs -cat /user/cloudera/student_import/part-m-00000
```

2.HIVE

```
# STEP 1: START HADOOP SERVICES
sudo service hadoop-hdfs-namenode start
sudo service hadoop-hdfs-datanode start
sudo service hadoop-yarn-resourcemanager start
sudo service hadoop-yarn-nodemanager start
sudo service hadoop-mapreduce-historyserver start

# STEP 2: START HIVE
hive

CREATE DATABASE company;
USE company;
CREATE TABLE employee(id INT, name STRING, sal INT);

INSERT INTO employee VALUES (1,"RAM",1200);
INSERT INTO employee VALUES (2,"aman",1300);
INSERT INTO employee VALUES (3,"ajay",1500);

SELECT * FROM employee;
SELECT COUNT(*) AS Total_Empl FROM employee;
SELECT * FROM employee WHERE sal < 1500;
SELECT AVG(sal) AS avg_sal FROM employee;
```

3.PIG

```
# Step 1: Start Cloudera services
sudo service hbase-master start
sudo service hbase-regionserver start

# Step 2: Create directory for Pig data
mkdir /home/cloudera/pigdata
cd /home/cloudera/pigdata

# Step 3: Create the sample input file
echo "1,Hardik,29" > sample.txt
echo "2,Shahid,24" >> sample.txt
echo "3,Ashraf,22" >> sample.txt

# Verify the file
cat sample.txt

# Step 4: Copy file to HDFS
hdfs dfs -mkdir /user/cloudera/pigdata
hdfs dfs -put /home/cloudera/pigdata/sample.txt /user/cloudera/pigdata/

# Verify upload
hdfs dfs -ls /user/cloudera/pigdata

# Step 5: Create a Pig script file
nano sample.pig

student = LOAD '/user/cloudera/pigdata/sample.txt' USING PigStorage(',') AS
(id:int, name:chararray, rollno:int);
DUMP student;
# Then save using Ctrl + X, Y, and Enter.

# Step 6: Run the Pig script
pig -x mapreduce /home/cloudera/pigdata/sample.pig
# Or
pig -x mapreduce sample.pig

# Step 8: Verify HDFS output (optional)
hdfs dfs -cat /user/cloudera/pigdata/sample.txt
```

3.HDFS:

```
# =====
# HDFS COMPLETE PRACTICAL SCRIPT
# =====

# [1] Start Only Required Services
sudo service hadoop-hdfs-namenode start
sudo service hadoop-hdfs-datanode start

# [2] Create Directory in HDFS
hdfs dfs -mkdir -p /user/cloudera/demo

# [3] Create Empty File in HDFS(not required)
hdfs dfs -touchz /user/cloudera/demo/sample.txt

# [4] Create File in Local System
echo "Hello Hadoop" > sample1.txt

# [5] Upload Local File to HDFS (Using put)
hdfs dfs -put -f sample1.txt /user/cloudera/demo/

# [6] View HDFS Directory and Files
hdfs dfs -ls /user/cloudera/demo/

# [7] View File Content in HDFS
hdfs dfs -cat /user/cloudera/demo/sample1.txt

# [8] Copy File within HDFS(not required)
hdfs dfs -cp /user/cloudera/demo/sample1.txt
          /user/cloudera/demo/sample_copy.txt
```

```
# [9] Move File within HDFS(not required)
hdfs dfs -mv /user/cloudera/demo/sample_copy.txt
/usr/cloudera/demo/moved_sample.txt

# [10] Copy File from HDFS to Local (Using get)
hdfs dfs -get /user/cloudera/demo/sample1.txt /home/cloudera/Desktop/

# [11] Check Disk Usage and HDFS Space
hdfs dfs -du /user/cloudera/demo/
hdfs dfs -df

# [12] Verify All Files in Demo Folder
hdfs dfs -ls /user/cloudera/demo/

# [13] (Optional) Delete File or Directory
# hdfs dfs -rm /user/cloudera/demo/sample1.txt
# hdfs dfs -rm -r /user/cloudera/demo/

# [14] (Optional) Empty Trash Permanently
# hdfs dfs -expunge

# =====
# END OF PRACTICAL SCRIPT
# =====
```

MongoDB Commands :

```
// Step 1: Start Mongo Shell
mongosh

// Step 2: Show all databases
show dbs

// Step 3: Use or create database
use learn

// Step 4: Show databases again
show dbs

// Step 5: Create collection
db.createCollection("emp")

// Step 6: Insert one document
db.emp.insertOne({empid:1, name:"Abhinav", salary:100000})

// Step 7: Insert multiple documents
db.emp.insertMany([
  {empid:2, name:"Hardik", salary:200000},
  {empid:3, name:"Ashaaf", salary:300000}
])

// Step 8: Display all documents
db.emp.find()

// Step 9: Find document by name
db.emp.find({name:"Ashaaf"})

// Step 10: Find document by empid
db.emp.find({empid:2})

// Step 11: Update one document
db.emp.updateOne(
  {name:"Hardik"},
  {$set:{salary:750000, empid:4}}
)
```

```
// Step 12: Verify update
db.emp.find({empid:4})

// Step 13: Update another document
db.emp.updateOne(
  {name:"Abhinav"},  

  {$set:{salary:750000}}
)

// Step 14: Update multiple documents
db.emp.updateMany(  

  {salary:750000},  

  {$set:{salary:700000}}
)

// Step 15: Display updated documents
db.emp.find()

// Step 16: Delete one document
db.emp.deleteOne({name:"Ashaaf"})

// Step 17: Verify deletion
db.emp.find()

// Step 18: Drop collection
db.emp.drop()

// Step 19: Drop database
db.dropDatabase()
```