# UNIVERSITY COLLEGE OF ENGINEERING NAGERCOIL

**(ANNA UNIVERSITY CONSTITUENT COLLEGE)**

**KONAM, NAGERCOIL – 629 004**


## RECORD NOTE BOOK

## CCS334-BIG DATA ANALYTICS

**REGISTER NO:**

# UNIVERSITY COLLEGE OF ENGINEERING NAGERCOIL

**(ANNA UNIVERSITY CONSTITUENT COLLEGE)**

**KONAM, NAGERCOIL – 629 004**


**Register No:**

*Certified that, this is the Bonafide record of work done by* ***Mr./Ms.*** *..... of VI Semester in Computer Science and Engineering of this college, in the* ***CCS334-BIG DATA ANALYTICS*** *during academic year 2023-2024 in partial fulfillment of the requirements of the B.E Degree course of the Anna University Chennai.*

**Staff-in-charge** **Head of the Department**

This record is submitted for the University Practical Examination held on .....

**Internal Examiner** **External Examiner**

# INDEX

| EX NO:1<br><br>01/02/2024 | DOWNLOADING AND INSTALLING HADOOP UNDERSTANDING DIFFERENT HADOOP MODES, STARTUP SCRIPTS, CONFIGURATION FILES |
|---|---|

## AIM:

To Downloading and installing Hadoop; Understanding different Hadoop modes. Startup scripts, Configuration files.

## PREREQUISITES TO INSTALL HADOOP ON WINDOWS:

VIRTUAL BOX (For Linux): it is used for installing the operating system on it.

OPERATING SYSTEM: You can install Hadoop on Windows or Linux based operating systems. Ubuntu and CentOS are very commonly used.

JAVA: You need to install the Java 8 package on your system.

HADOOP: You require Hadoop latest version

1.Install Java

Java JDK Link to download here

Extract and install Java in C:\Java

Open cmd and type -> javac –version

2. Download Hadoop

Hadoop Link to download: https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz

Extract to C:\Hadoop

3. Set the path JAVA_HOME Environment variable

4. Set the path HADOOP_HOME Environment variable

5. Configure the required files in the C:\hadoop\etc as follows:

# For core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>

```
# For hdfs-site.xml or https-site.xml
<configuration>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property><property>
  <name>dfs.namenode.name.dir</name>
  <value>C:\hadoop\data\namenode</value>
</property><property>
  <name>dfs.datanode.data.dir</name>
  <value>C:\hadoop\data\datanode</value>
</property>
</configuration>

# For mapred-site.xml
<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
</configuration>

# For yarn-site.xml
<configuration>
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property><property>
  <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
```

</property>

</configuration>

6.Create the following folders:

Create folder “data” under “C:\Hadoop”

Create folder “datanode” under “C:\Hadoop-3.3.0\data”

Create folder “namenode” under “C:\Hadoop-3.3.0\data”

7.Edit file hadoop-env.cmd:

C:/Hadoop-3.3.0/etc/hadoop/hadoop-env.cmd by closing the command line “JAVA HOME=%JAVA HOME%” instead of set ;

```
@rem The java implementation to use.  Required.
set JAVA_HOME=C:\progra~1\Java\jdk-1.8
```

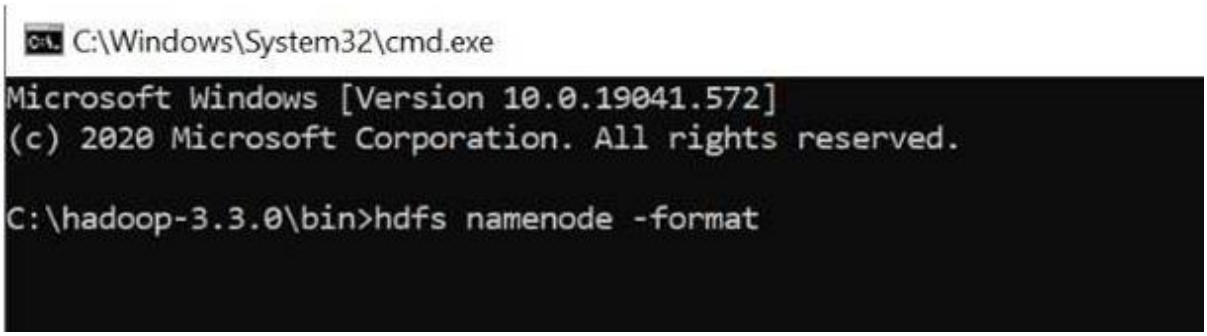
8. Hadoop Configurations:

Download https://github.com/s911415/apache-hadoop-3.1.0-winutil

Copy folder bin and replace existing bin folder in C:\Hadoop-3.3.0\bin.

9.Format the NameNode:

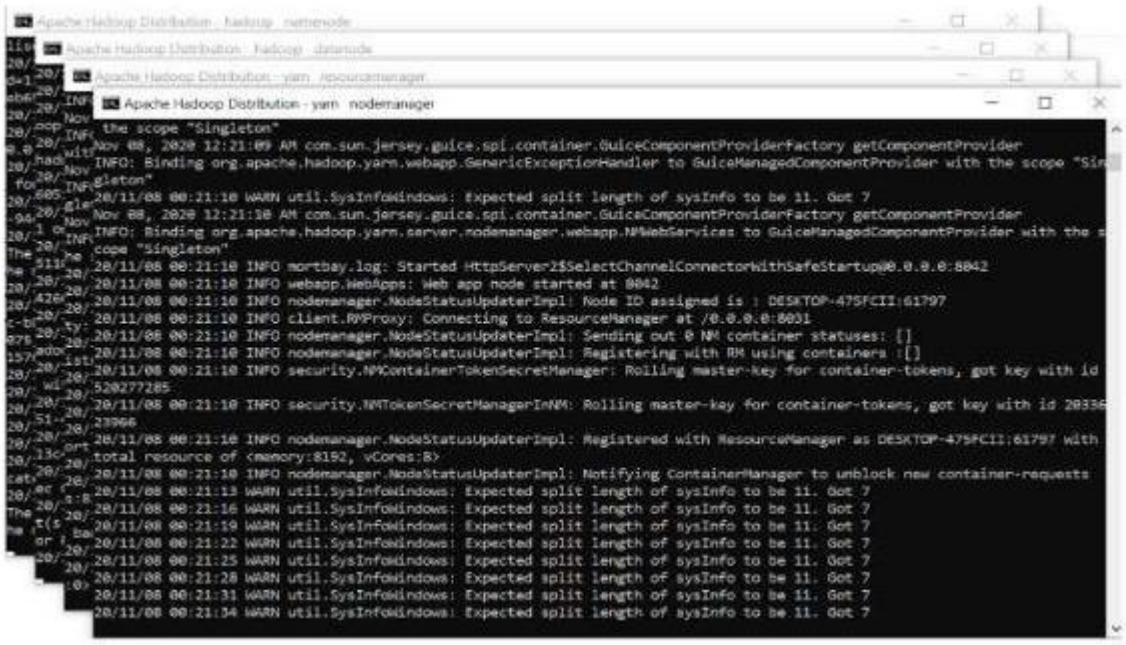
Open cmd and type command “hdfs namenode –format”




10. Start Hadoop:

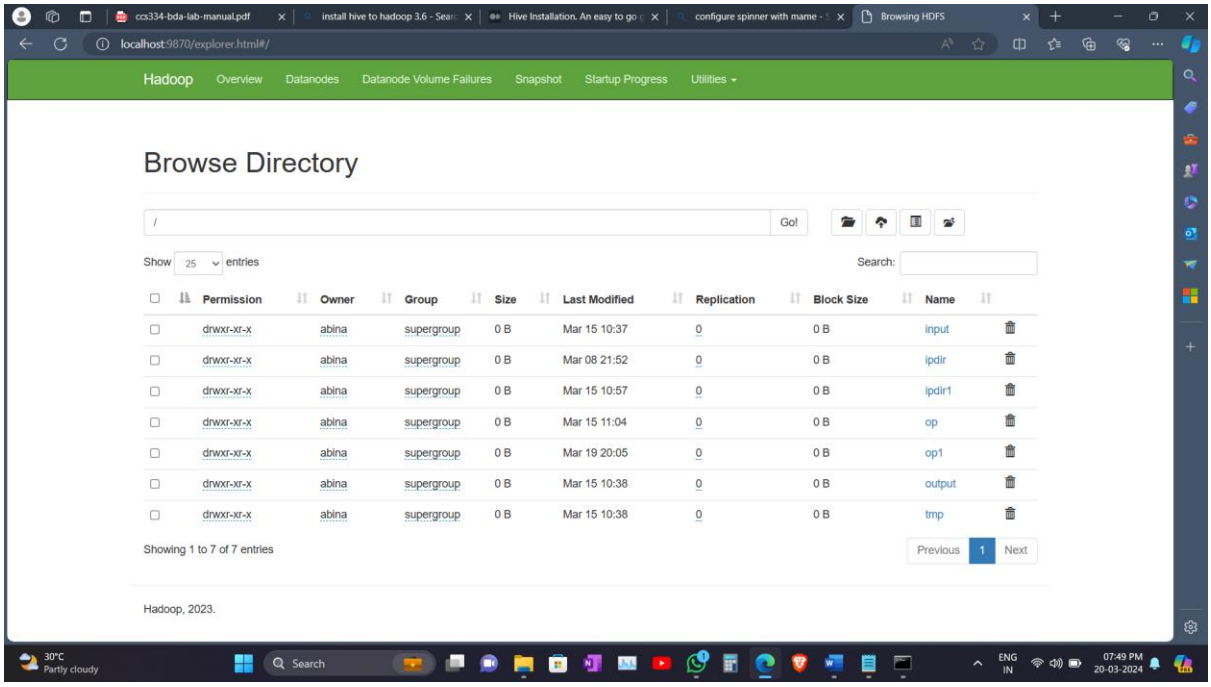
Open cmd and type command “start-all.cmd”

Then verify the following windows appear without any warnings and errors.

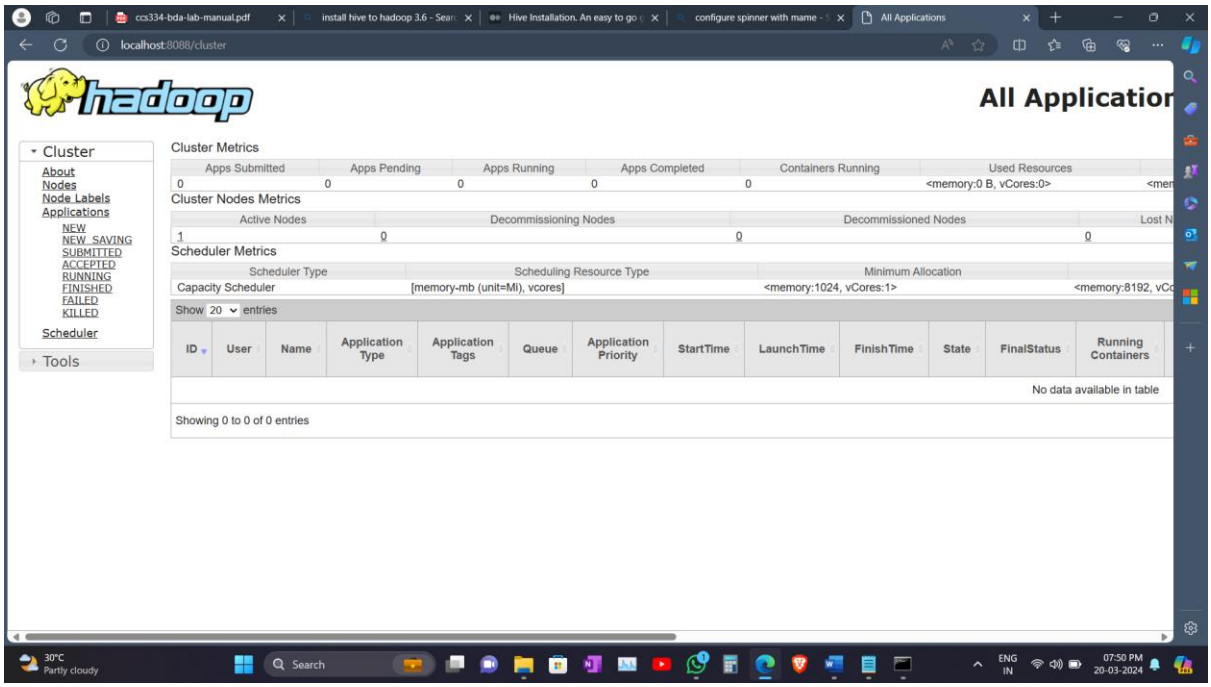



11.Open localhosts:

Localhost:9870



Localhost:8088




## Result:

To Downloading and installing Hadoop; Understanding different Hadoop modes. Startup scripts, Configuration files

| EX NO:2<br><br>08/02/2024 | HADOOP IMPLEMENTATION OF FILE MANAGEMENT TASKS SUCH AS ADDING FILES AND DIRECTORIES, RETRIEVING FILES AND DELETING FILES |
|---|---|

## AIM:

To implement the following file management tasks in Hadoop:

1. Adding files and directories

2. Retrieving files

3. Deleting Files

## PROCEDURE:

**1.Create a directory in HDFS at given path:**

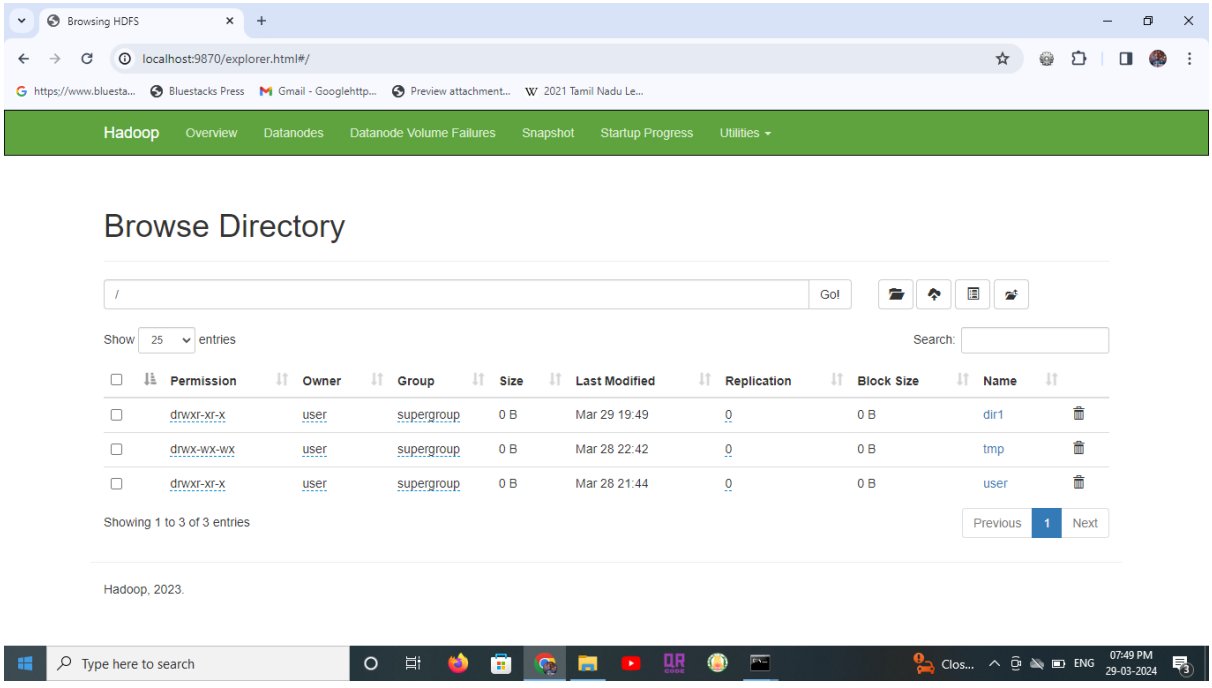
Usage:

hadoop fs -mkdir <paths> Example:

hadoop fs -mkdir /user/saurzcode/dir1 /user/saurzcode/dir2

```
C:\hadoop\sbin>hadoop fs -mkdir /dir1

C:\hadoop\sbin>
```



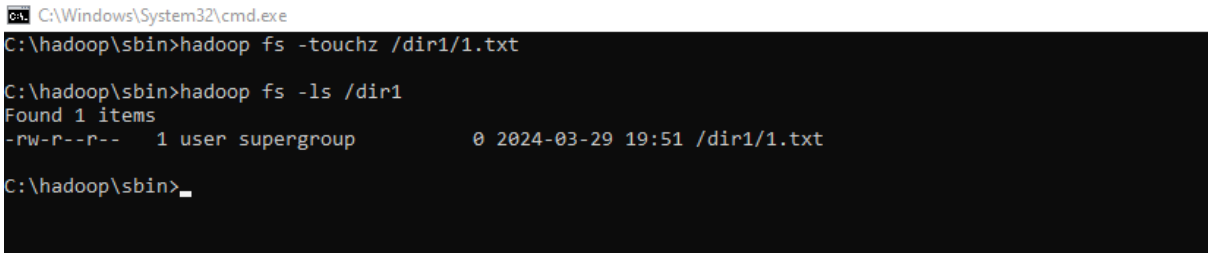
**2.List the contents of a directory:**

Usage:

hadoop fs -ls <args>

Example:

hadoop fs -ls /user/saurzcode



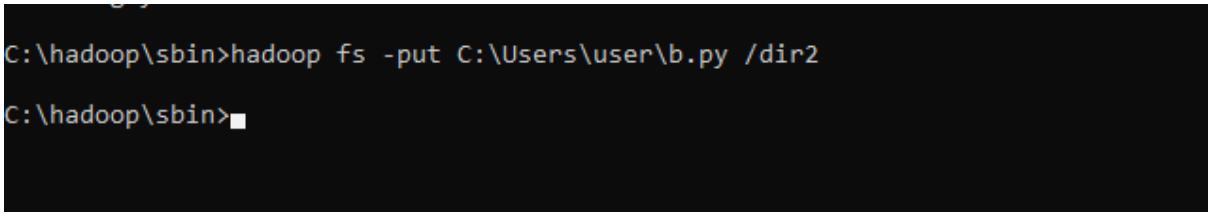

**3.Upload and download a file in HDFS:**

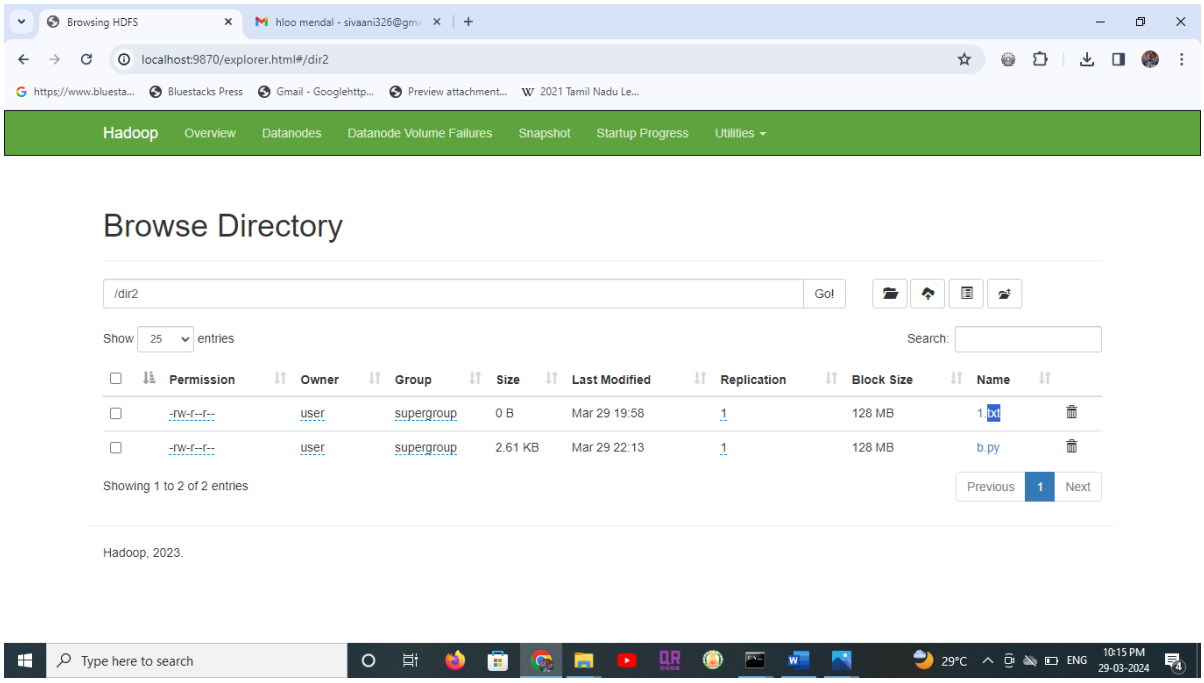
Upload: hadoop fs -put:

Copy single src file, or multiple src files from local file system to the Hadoop data file system Usage:

hadoop fs -put <localsrc>...<HDFS_dest_Path>Example:

hadoop fs -put /home/saurzcode/Samplefile.txt /user/saurzcode/dir3/



**Adding files to a directory:**

hadoop fs -touchz:

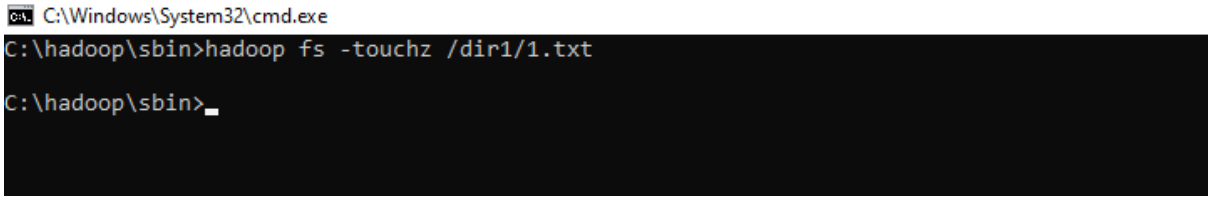
Adds files to a directory.

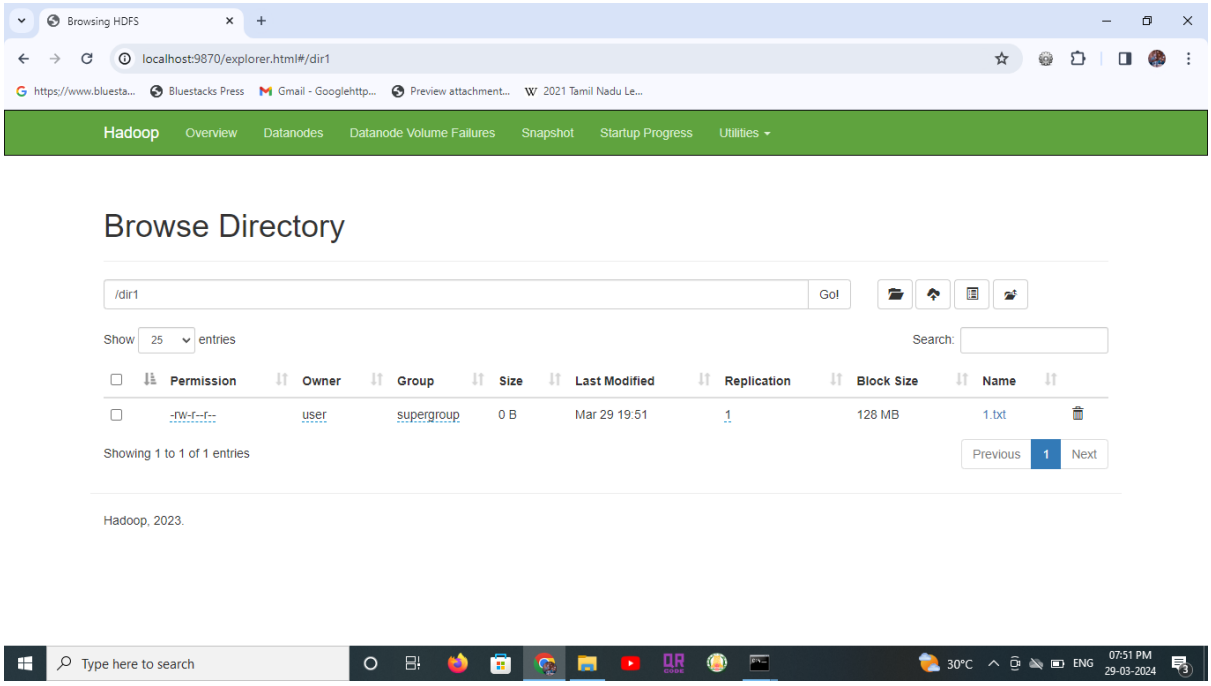
Usage:

hadoop fs -touchz /directory_name/file

Example:

hadoop fs -touchz /dir1/1.txt



**4. Remove a file or directory in HDFS.**

Remove files specified as argument. Deletes directory only when it is empty Usage:

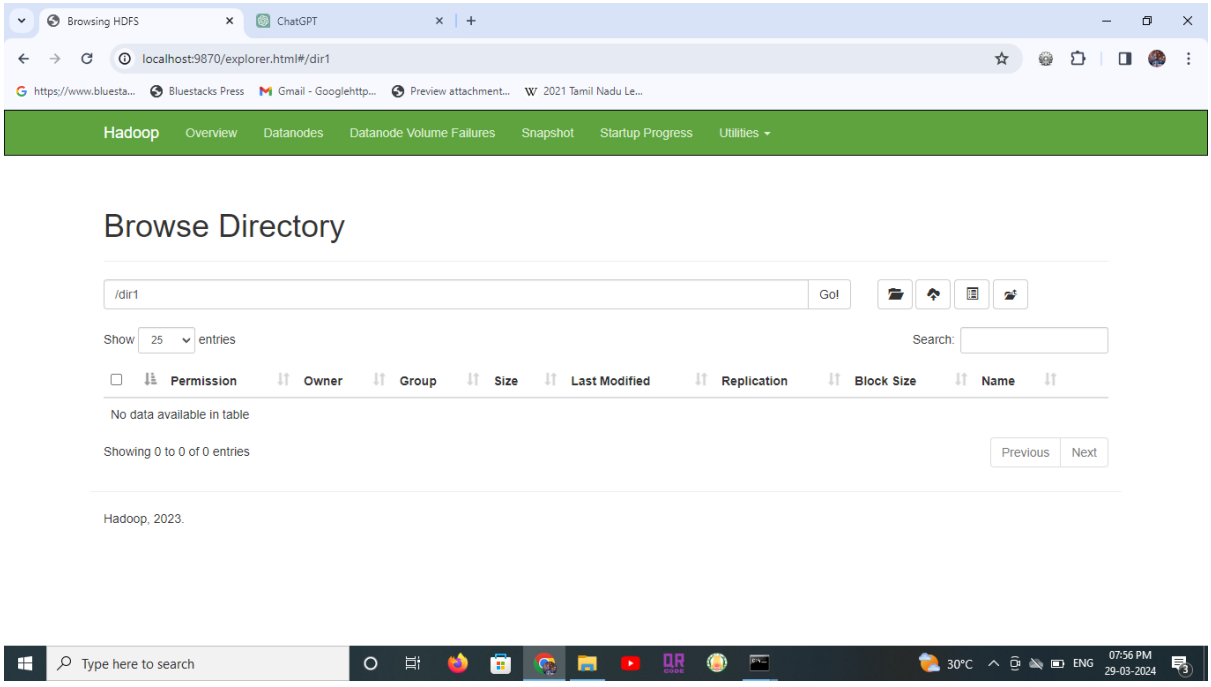
hadoop fs -rm <arg> Example:

hadoop fs -rm/user/saurzcode/dir1/abc.txt

Recursive version of delete.

Usage:

hadoop fs -rm <arg> Example: hadoop fs -rmr/user/saurzcode/


**RESULT:**

Thus, the Hadoop Implementation of file management tasks, such as Adding files and directories, retrieving files and Deleting files is executed successfully.

| EX NO:3<br><br>04/03/2024 | IMPLEMENTATION OF MATRIX MULTIPLICATION USING MAP REDUCE |
|---|---|

**AIM:**

To implement matrix multiplication using mapreduce in Hadoop.

**PROGRAM:**

**Map.java**

```
package matrix;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
//import org.apache.hadoop.mapreduce.Mapper;
import java.io.IOException;
public class Map extends org.apache.hadoop.mapreduce.Mapper<LongWritable, Text, Text, Text>
{
    @Override
public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException {
        Configuration conf = context.getConfiguration();
        int m = Integer.parseInt(conf.get("m"));
        int p = Integer.parseInt(conf.get("p"));
String line = value.toString();
// (M, i, j, Mij);
String[] indicesAndValue = line.split(",");
Text outputKey = new Text();
Text outputValue = new Text();
if (indicesAndValue[0].equals("M")) {
for (int k = 0; k < p; k++) {
outputKey.set(indicesAndValue[1] + "," + k);
```

```
// outputKey.set(i,k);
outputValue.set(indicesAndValue[0] + "," + indicesAndValue[2]
+ "," + indicesAndValue[3]);
// outputValue.set(M,j,Mij);
context.write(outputKey, outputValue);
}
} else {
        // (N, j, k, Njk);
        for (int i = 0; i < m; i++) {
        outputKey.set(i + "," + indicesAndValue[2]); outputValue.set("N," +
indicesAndValue[1] + ","
        + indicesAndValue[3]); context.write(outputKey, outputValue);
        }
        }
        }
}
```

**MatrixMultiply.java**

```
package matrix;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
public class MatrixMultiply {
public static void main(String[] args) throws Exception { if (args.length != 2) {
System.err.println("Usage: MatrixMultiply <in_dir> <out_dir>");
System.exit(2);
}
```

```
Configuration conf = new Configuration();
conf.set("m", "1000");
conf.set("n", "100");
conf.set("p", "1000");
@SuppressWarnings("deprecation")
Job job = new Job(conf, "MatrixMultiply");
job.setJarByClass(MatrixMultiply.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(Text.class);
job.setMapperClass(Map.class);
job.setReducerClass(Reduce.class);
job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
job.waitForCompletion(true);
}
}
```

**Reduce.java**

```
package matrix;
import org.apache.hadoop.io.Text;
// import org.apache.hadoop.mapreduce.Reducer;
import java.io.IOException;
import java.util.HashMap;
public class Reduce
extends org.apache.hadoop.mapreduce.Reducer<Text, Text, Text, Text> { @Override
public void reduce(Text key, Iterable<Text> values, Context context)
throws IOException, InterruptedException {
String[] value;
```

```
//key=(i,k),

//Values = [(M/N,j,V/W),..]

HashMap<Integer, Float> hashA = new HashMap<Integer, Float>(); HashMap<Integer,
Float> hashB = new HashMap<Integer, Float>(); for (Text val : values) {

value = val.toString().split(",");

if (value[0].equals("M")) {

hashA.put(Integer.parseInt(value[1]), Float.parseFloat(value[2])); } else {

hashB.put(Integer.parseInt(value[1]), Float.parseFloat(value[2]));

}

}

int n = Integer.parseInt(context.getConfiguration().get("n"));

float result = 0.0f;

float m_ij;

float n_jk;

for (int j = 0; j < n; j++) {

m_ij = hashA.containsKey(j) ? hashA.get(j) : 0.0f; n_jk = hashB.containsKey(j) ?
hashB.get(j) : 0.0f; result += m_ij * n_jk;

}

if (result != 0.0f) {

context.write(null,

new Text(key.toString() + "," + Float.toString(result)));

}

}

}
```

**PROCEDURE:**

Step 1:create a directory in hadoop

Input the two text files with the matrix files int into the direcrory in Hadoop.

Using

Hadoop fs -put <file location>/<input directory>.

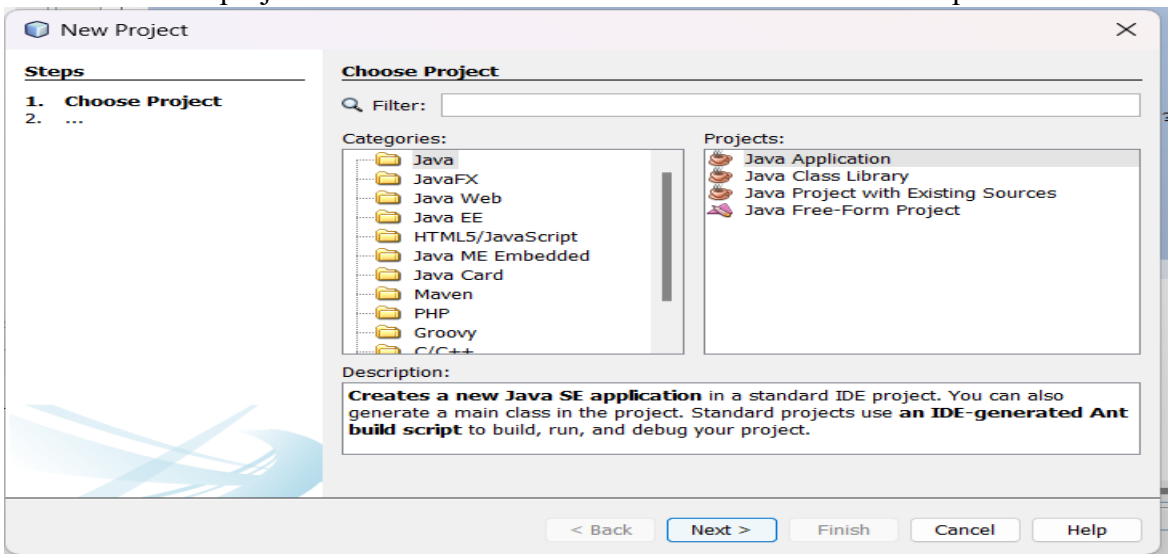
| Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|---|---|
| -rw-r--r-- | abina | supergroup | 34 B | Apr 05 21:16 | 1 | 128 MB | 1.txt |
| -rw-r--r-- | abina | supergroup | 34 B | Apr 05 21:16 | 1 | 128 MB | 2.txt |

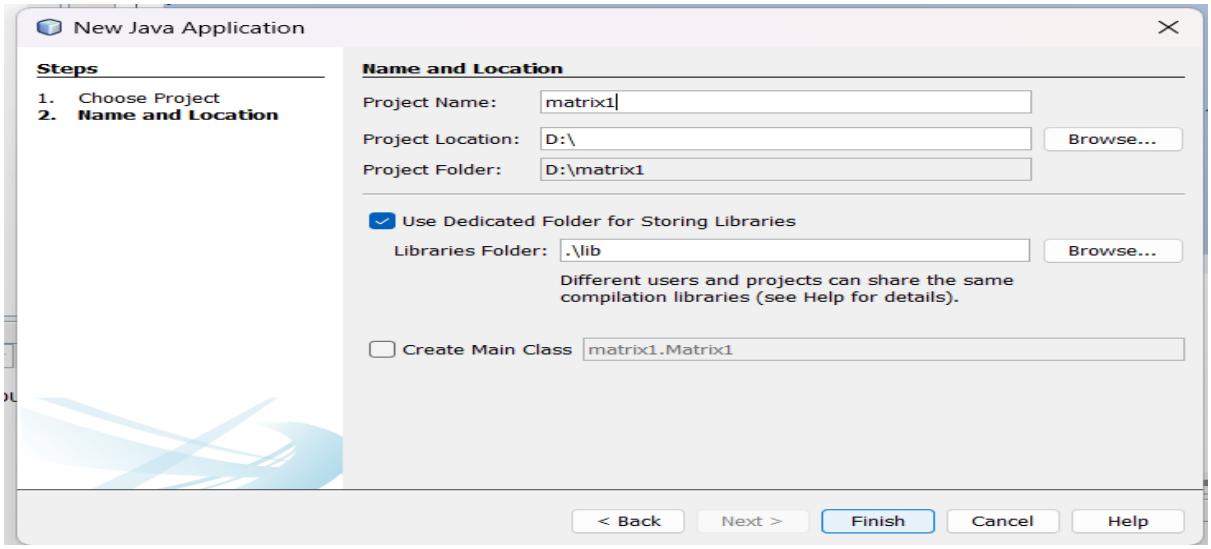
```
C:\hadoop\sbin>hadoop dfs -cat /ipdir/1.txt
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
M,0,0,1
M,0,0,2
M,0,0,3
M,0,0,4
C:\hadoop\sbin>hadoop dfs -cat /ipdir/2.txt
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
N,0,1,2
N,0,1,3
N,0,1,4
N,0,1,5
C:\hadoop\sbin>
```

Step 2:Open the netbeans application an open a new project.

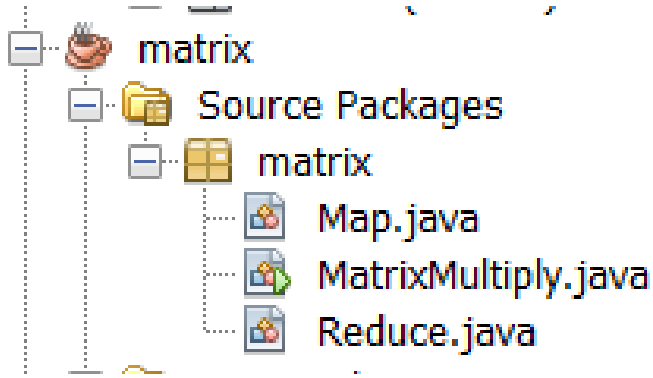
1.Choose categories as java and projects as java application.

2.Set the project name as matrix and untick the create main class option.

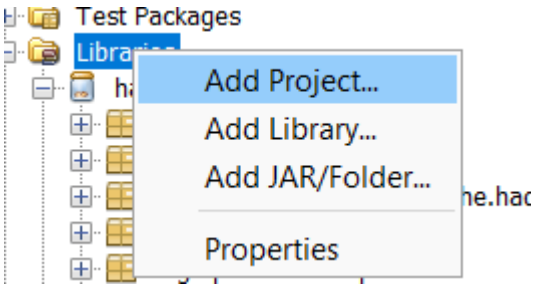


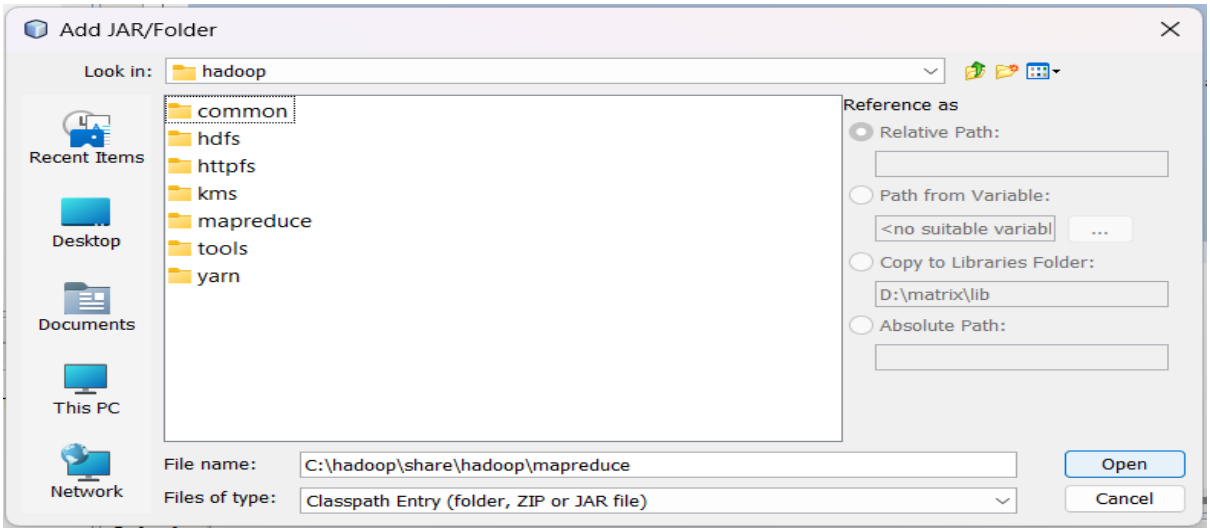
3.Add a new package a matrix and add the above three programs as three different class files



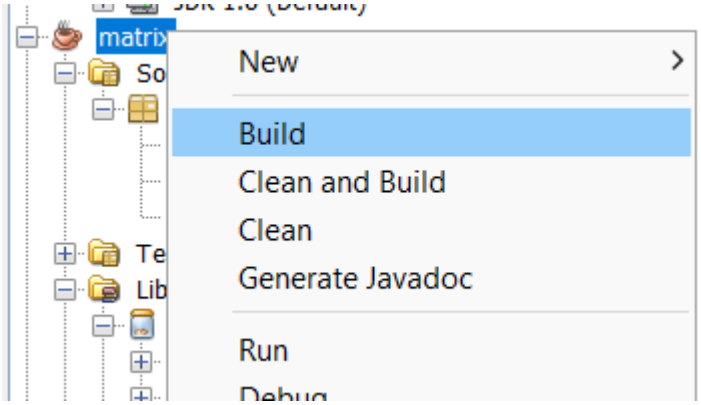

4.Add libraries to the package by right clicking in the libraries tab an select add JAR/FOLDER.



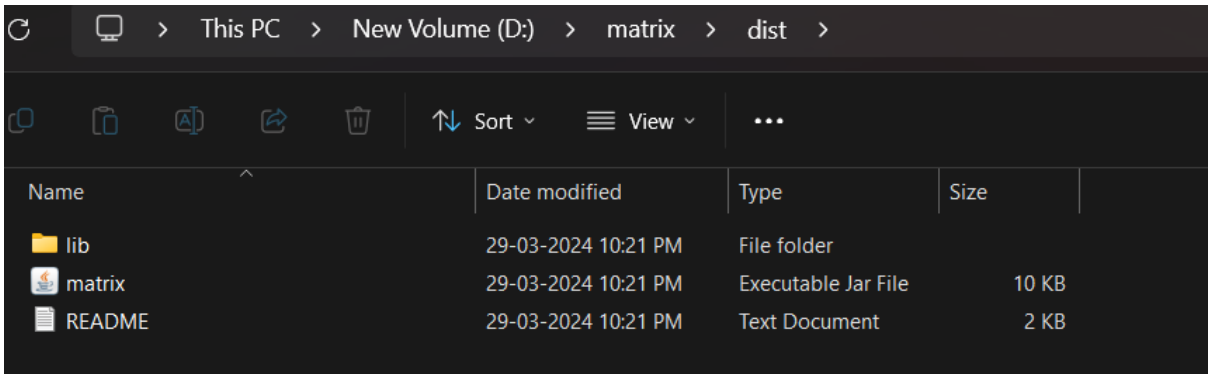

5.Add the folders like mapreduce and common.



6.Right click on the matrix project and clean and build the project.




7.The jar file will be added to the following location.




Step 3 : Input the following jar file into the same directory where we inserted the matrix files.

Using command hadoop jar <jar file name.jar jar name.package name.main class name > / input directory /output directory

```
D:\matrix\dist>hadoop jar matrix.jar matrix.MatrixMultiply /ipdir /op
```

Step 4:Get the output matrix

Using

hadoop dfs -cat /<output directory>/*

```
C:\hadoop\sbin>hadoop dfs -cat /op/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
0,1,8.0
```

**RESULT:**

Thus the implementation of matrix multiplication using map reduce was executed successfully.

| EX NO:4<br><br>07/03/2024 | RUN A BASIC WORD COUNT MAP REDUCE PROGRAM TO UNDERSTAND MAP REDUCE PARADIGM |
|---|---|

## AIM:

To run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.

## PREREQUISITES:

**VIRTUAL BOX (For Linux):** it is used for installing the operating system on it.

**OPERATING SYSTEM:** You can install Hadoop on Windows or Linux based operating systems. Ubuntu and CentOS are very commonly used.

**JAVA:** You need to install the Java 8 package on your system.

**HADOOP:** You require Hadoop latest version

## PROCEDURE:

Step 1: Install Hadoop with java and required configuration files.

Step 2: Create a directory in Hadoop.

```
C:\hadoop\sbin>hadoop fs -mkdir /ipdir1
```

Step 3: Upload a word file into the directory.

```
C:\hadoop\sbin>hadoop fs -mkdir /ipdir1

C:\hadoop\sbin>hadoop fs -put D:\word.txt /ipdir1
```

Step 4: View the file using cat option.

```
C:\hadoop\sbin>hadoop dfs -cat /ipdir1/word.txt
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
apple is a red color fruit. it is sweet in taste.
```

Step 5:Go to file location C:\hadoop\share\hadoop\mapreduce and open cmd.And type the following cmd.

```
C:\hadoop\share\hadoop\mapreduce>hadoop jar hadoop-mapreduce-examples-3.3.6.jar
An example program must be given as the first argument.
Valid program names are:
  aggregatewordcount: An Aggregate based map/reduce program that counts the words in the input files.
  aggregatewordhist: An Aggregate based map/reduce program that computes the histogram of the words in the input files.
  bbp: A map/reduce program that uses Bailey-Borwein-Plouffe to compute exact digits of Pi.
  dbcount: An example job that count the pageview counts from a database.
  distbbp: A map/reduce program that uses a BBP-type formula to compute exact bits of Pi.
  grep: A map/reduce program that counts the matches of a regex in the input.
  join: A job that effects a join over sorted, equally partitioned datasets
  multifilewc: A job that counts words from several files.
  pentomino: A map/reduce tile laying program to find solutions to pentomino problems.
  pi: A map/reduce program that estimates Pi using a quasi-Monte Carlo method.
  randomtextwriter: A map/reduce program that writes 10GB of random textual data per node.
  randomwriter: A map/reduce program that writes 10GB of random data per node.
  secondarysort: An example defining a secondary sort to the reduce.
  sort: A map/reduce program that sorts the data written by the random writer.
  sudoku: A sudoku solver.
  teragen: Generate data for the terasort
  terasort: Run the terasort
  teravalidate: Checking results of terasort
  wordcount: A map/reduce program that counts the words in the input files.
  wordmean: A map/reduce program that counts the average length of the words in the input files.
  wordmedian: A map/reduce program that counts the median length of the words in the input files.
  wordstandarddeviation: A map/reduce program that counts the standard deviation of the length of the words in the input files.

C:\hadoop\share\hadoop\mapreduce>hadoop jar hadoop-mapreduce-examples-3.3.6.jar wordcount /ipdir1
Usage: wordcount <in> [<in>...] <out>

C:\hadoop\share\hadoop\mapreduce>hadoop jar hadoop-mapreduce-examples-3.3.6.jar wordcount /ipdir1 /op1
24/04/07 22:45:20 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
24/04/07 22:45:20 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
```

Step 6: View the output file using cat option.

```
C:\hadoop\share\hadoop\mapreduce>hadoop dfs -cat /op1/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
a       1
apple   1
color   1
fruit.  1
in      1
is      2
it      1
red     1
sweet   1
taste.  1
```

## RESULT:

Thus running a basic Word Count Map Reduce program to understand Map Reduce Paradigm was executed successfully.

| EX.NO:5<br><br>14/03/2024 | INSTALLATION OF HIVE ALONG WITH PRACTICE EXAMPLES |
|---|---|

**AIM:**

To install HIVE along with practice examples.

**PREREQUISITES:**

1. Java Development Kit (JDK8) installed and the JAVA_HOME environment variable set.

2. Hadoop (version 2.9.2) installed and configured on your Windows system.

3. Apache Derby installed and DERBY_HOME environment variable set.

**PROCEDURE:**

**1.Download HIVE:**

Visit the Apache Hive website and download the stable version (version 3.1.2) of Hive.
Apache Hive website: **https://hive.apache.org/**

**2. Extract the Downloaded Hive Archive to a Directory on Your Windows Machine:**
C:\hive.

**3.Configure Hive:**

1. Download a Hive configuration file (hive-site.xml) in the conf folder of the extracted hive.

**Download link:** https://drive.google.com/file/d/1tsBbHdvM1fFktmn9O0-u0pbG1vWWFoyE/view

2. Copy the library files in the derby lib folder and paste them into the hive lib folder.
3. Replace the bin folder in Hive with the fixed bin folder from this website:
GitHub - HadiFadl/Hive-cmd: All cmd files needed to run Hive on windows (taken from https://svn.apache.org/repos/asf/hive/trunk/bin/)

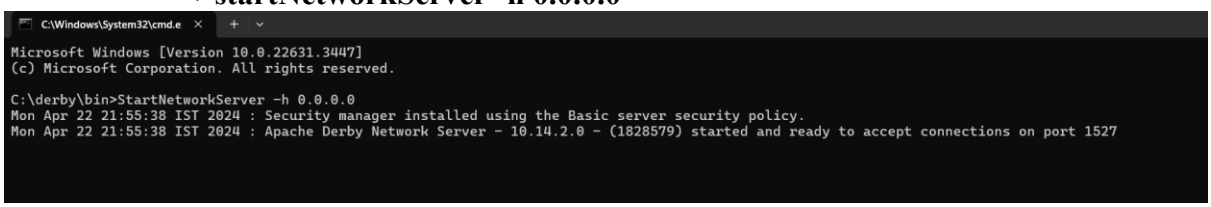
**4.Environment Variable Setup:**

1. Add the HIVE binary directory (C:\hive\bin) to your PATH environment variable.
2. Set the HIVE_HOME, HIVE_LIB and HIVE_BIN environment variables.
3. Add the Path variable HADOOP_USER_CLASSPATH_FIRST = true in both User and System variables.

**5.Start Hive:**

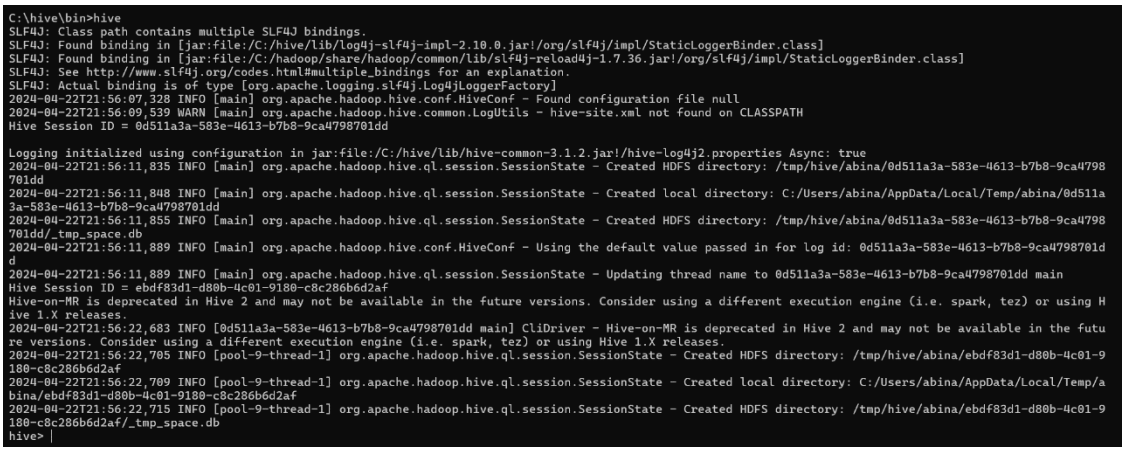
1. Run Hadoop in the command prompt.
2. Open a new command prompt and navigate to the Derby installation directory i.e., C:\derby\bin and execute the command:

**>startNetworkServer -h 0.0.0.0**



3. Open a new command prompt and navigate to the Hive installation directory i.e., C:\hive\bin and execute the command:

**>hive**




**EXAMPLES:**

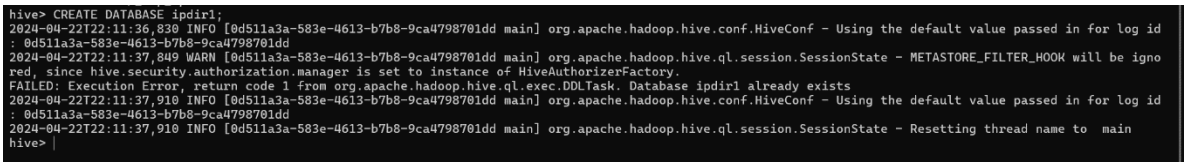
**Create a Database:**

To create a new database HIVE, use the following syntax:

CREATE DATABASE database_name;

**Example:**

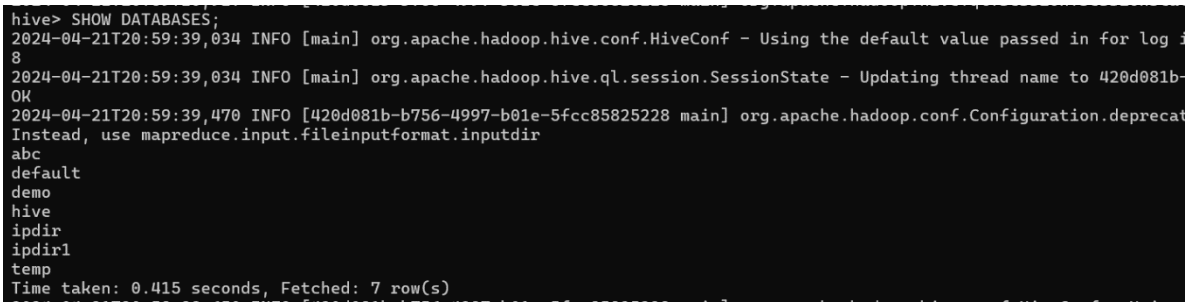
CREATE DATABASE ipdir;




**Show Database:**

To display a list of available databases in HIVE, use the following syntax:

SHOW DATABASES;



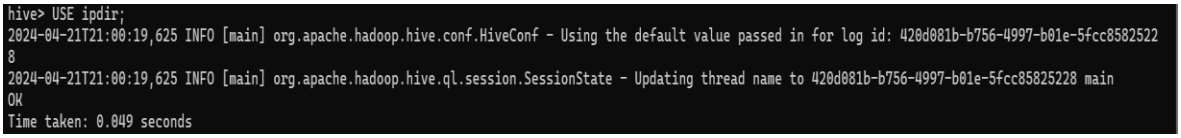

**Use a Database:**

To use a specific database in HIVE, use the following syntax:

USE database_name;

**Example:**

USE ipdir;



**Create a Table:**

To create a table in HIVE, use the following syntax:

CREATE TABLE table_name(column1 datatype, column 2 datatype, …);

**Example:**

CREATE TABLE student(id int, name string, age int);

```
hive> CREATE TABLE student(id int, name string, age int);
2024-04-21T21:01:15,742 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: 420d081b-b756-4997-b01e-5fcc8582522
8
2024-04-21T21:01:15,742 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to 420d081b-b756-4997-b01e-5fcc85825228 main
OK
Time taken: 1.16 seconds
```

**Insert data into a Table:**

To insert data into a table in HIVE, use the following syntax:

INSERT INTO table_name (column1, column2, ..) VALUES(value1, value2, ..);

**Example:**

INSERT INTO student(id, name, age) VALUES(1, ‘John Doe’, 25);

```
hive> INSERT INTO student(id, name, age) VALUES(001, 'manu ', 25);
2024-04-21T21:02:59,930 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: 420d081b-b756-4997-b01e-5fcc8582522
8
2024-04-21T21:02:59,930 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to 420d081b-b756-4997-b01e-5fcc85825228 main
2024-04-21T21:03:03,085 INFO [420d081b-b756-4997-b01e-5fcc85825228 main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: h
dfs://localhost:9000/user/hive/warehouse/ipdir.db/student/.hive-staging_hive_2024-04-21_21-02-59_960_7916773066749099885-1
2024-04-21T21:03:03,314 INFO [420d081b-b756-4997-b01e-5fcc85825228 main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: f
ile:/C:/Users/abina/AppData/Local/Temp/abina/420d081b-b756-4997-b01e-5fcc85825228/hive_2024-04-21_21-03-03_159_2527877755128061028-1/-mr-10000/.hive-staging
_hive_2024-04-21_21-03-03_159_2527877755128061028-1
Query ID = abina_20240421210259_0ba4ef63-20b1-4d34-b6e4-e3ba5f60acc6
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:
```

**Select Data from a Table:**

To select data from a table in HIVE, use the following syntax:

SELECT * FROM table_name;

**Example:**

SELECT * FROM student;

```
hive> SELECT * FROM student;
2024-04-21T21:04:01,400 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: 420d081b-b756-4997-b01e-5fcc8582522
8
2024-04-21T21:04:01,400 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to 420d081b-b756-4997-b01e-5fcc85825228 main
2024-04-21T21:04:01,629 INFO [420d081b-b756-4997-b01e-5fcc85825228 main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: f
ile:/C:/Users/abina/AppData/Local/Temp/abina/420d081b-b756-4997-b01e-5fcc85825228/hive_2024-04-21_21-04-01_425_7343547800977414264-1/-mr-10001/.hive-staging
_hive_2024-04-21_21-04-01_425_7343547800977414264-1
OK
1       manu    25
Time taken: 0.263 seconds, Fetched: 1 row(s)
2024-04-21T21:04:01,711 INFO [420d081b-b756-4997-b01e-5fcc85825228 main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id
: 420d081b-b756-4997-b01e-5fcc85825228
2024-04-21T21:04:01,711 INFO [420d081b-b756-4997-b01e-5fcc85825228 main] org.apache.hadoop.hive.ql.session.SessionState - Resetting thread name to  main
```

**RESULT:**

Thus, the installation of HIVE along with practice examples was executed successfully.

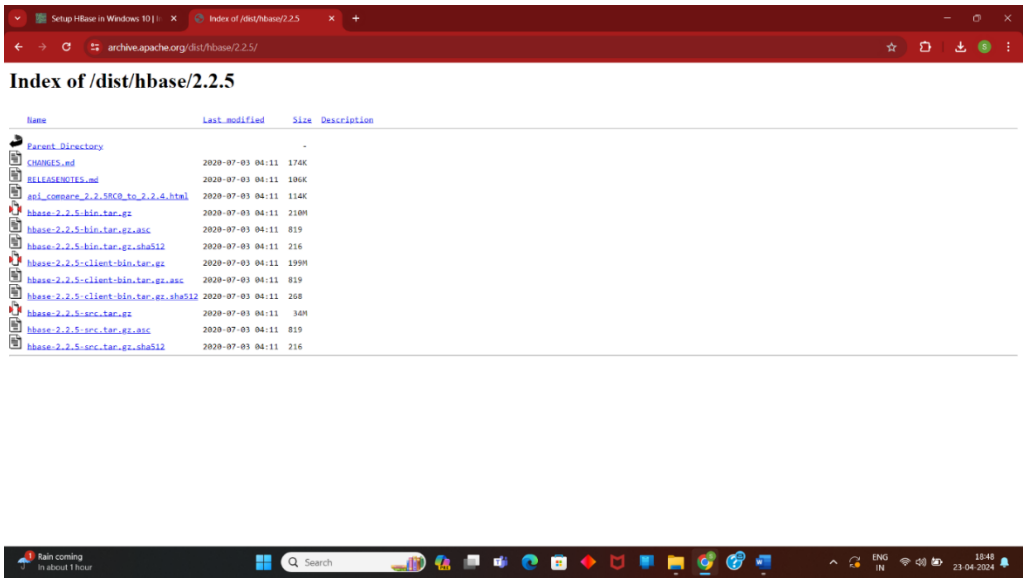
| EX.NO:6a<br><br>26/03/2024 | INSTALLATION OF HBASE ALONG WITH PRACTICE EXAMPLES |
|---|---|

## AIM:

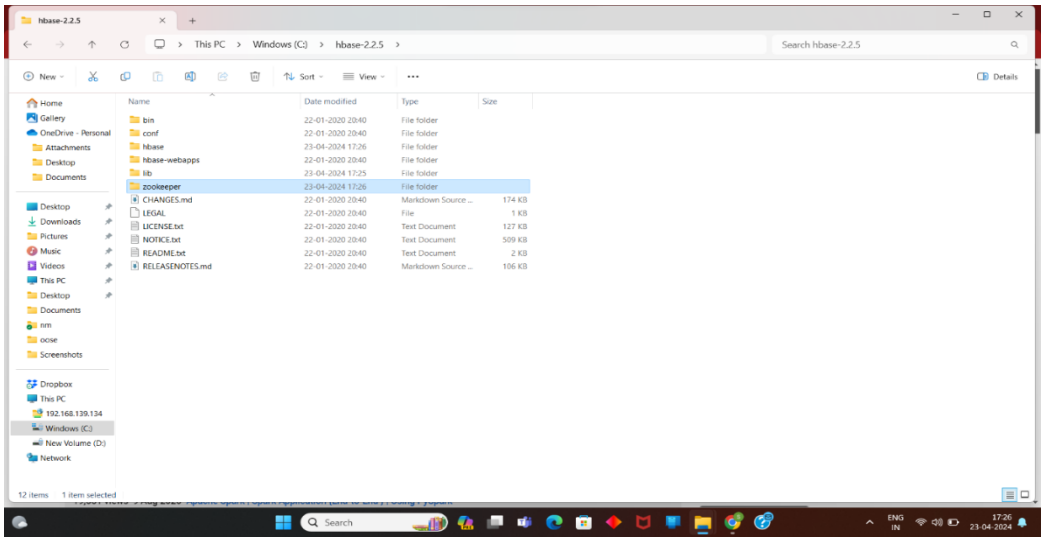
To install hbase along with practice examples.

## PROCEDURE:

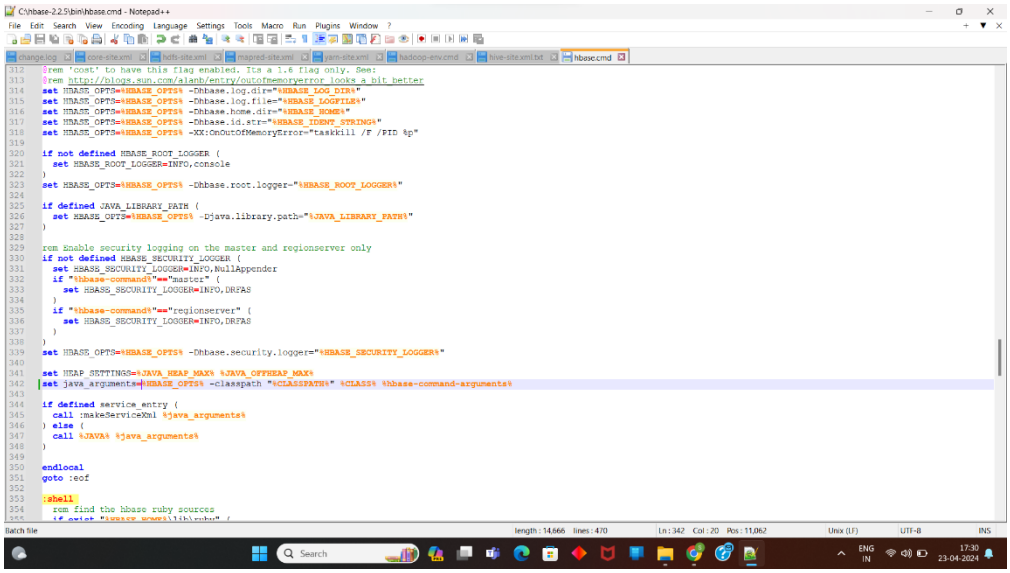
**Step 1:**Download the hbase 2.2.5 bin file from apache website and unzip the file and extract the file.




**Step 2:**Create a folders as shown below inside root folder for HBase data and zookeeper.



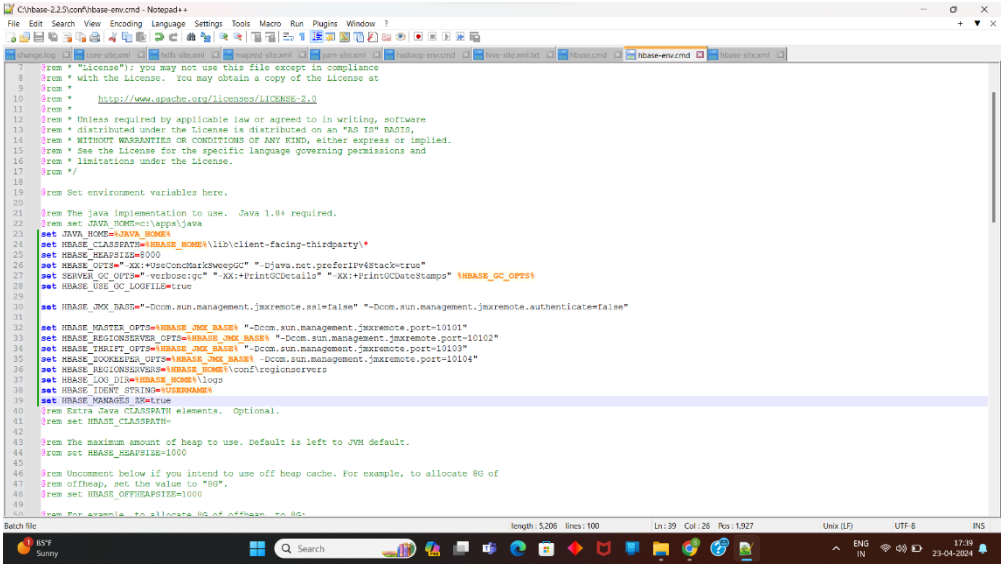
**Step 3:**Open C:/ hbase-2.2.5/bin/hbase.cmd in notepad++. Search for below given lines and remove **%HEAP_SETTINGS%** from that line



**Step 4:**Open C: /hbase-2.2.5/conf/hbase-env.cmd n notepad++. Add the below lines to the file after the comment session

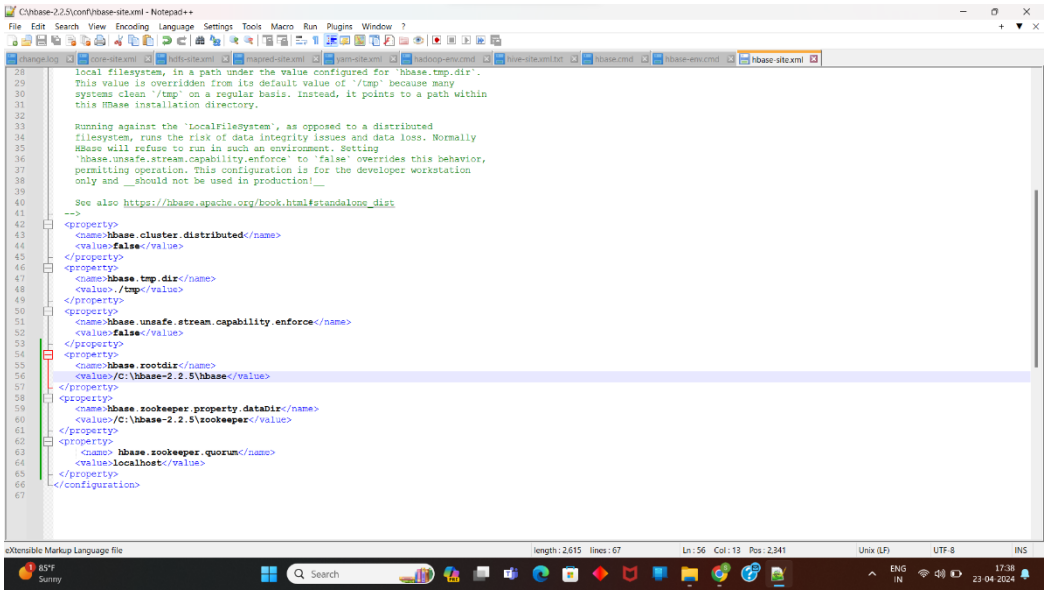
- set JAVA_HOME=%JAVA_HOME%
- set HBASE_CLASSPATH=%HBASE_HOME%\lib\client-facing-thirdparty*
- set HBASE_HEAPSIZE=8000
- set HBASE_OPTS="-XX:+UseConcMarkSweepGC" "-Djava.net.preferIPv4Stack=true"
- set SERVER_GC_OPTS="-verbose:gc" "-XX:+PrintGCDetails" "-XX:+PrintGCDateStamps" %HBASE_GC_OPTS%
- set HBASE_USE_GC_LOGFILE=true
- set HBASE_JMX_BASE="-Dcom.sun.management.jmxremote.ssl=false" "-Dcom.sun.management.jmxremote.authenticate=false"
- set HBASE_MASTER_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10101"
- set HBASE_REGIONSERVER_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10102"
- set HBASE_THRIFT_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10103"
- set HBASE_ZOOKEEPER_OPTS=%HBASE_JMX_BASE% -Dcom.sun.management.jmxremote.port=10104"
- set HBASE_REGIONSERVERS=%HBASE_HOME%\conf\regionservers

- set HBASE_LOG_DIR=%HBASE_HOME%\logs
- set HBASE_IDENT_STRING=%USERNAME%
- set HBASE_MANAGES_ZK=true

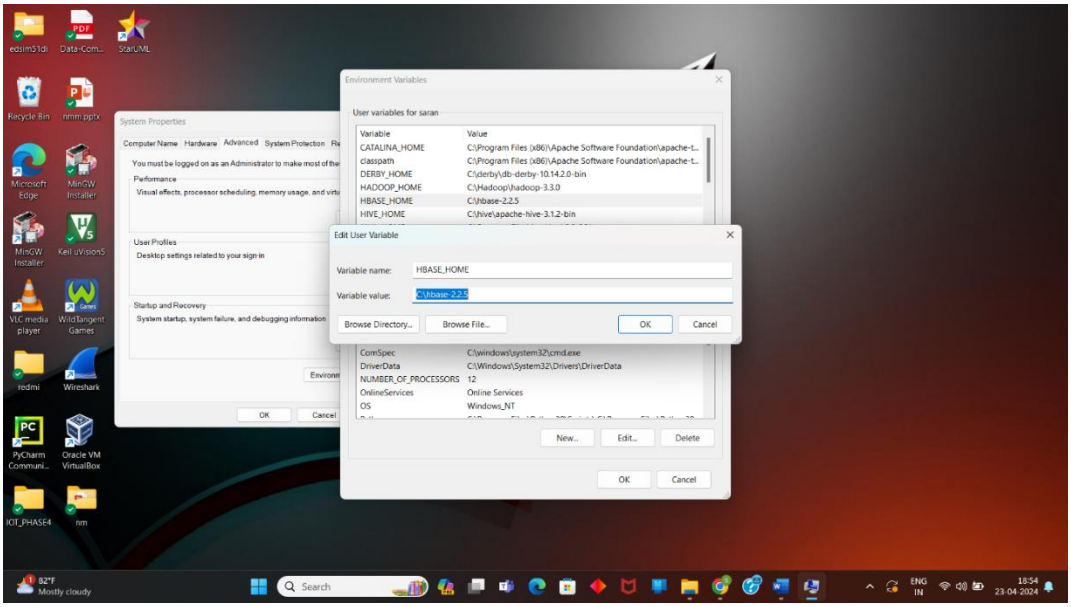


**Step 5:**Open C:/hbase-2.2.5/conf/hbase-site.xml notepad++. Add the below lines inside <configuration> tag.

```
<property>
   <name>hbase.rootdir</name>
   <value>C:/hbase-2.2.5/hbase</value>
 </property>
 <property>
   <name>hbase.zookeeper.property.dataDir</name>
   <value>C:/hbase-2.2.5/zookeeper</value>
 </property>
 <property>
    <name> hbase.zookeeper.quorum</name>
   <value>localhost</value>
 </property>
```




**Step 6:**Setup the Environment variable for **HBASE_HOME**




**Step 7:**Open terminal and direct to the hbase bin path.

- Type “start-hbase.cmd” to start the hbase
- Type “hbase version” to verify the hbase server is running

```
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd..

C:\Windows>cd C:\hbase-2.2.5\bin

C:\hbase-2.2.5\bin>start-hbase.cmd
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/hbase-2.2.5/lib/client-facing-thirdparty/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/C:/Hadoop/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.slf4j.impl.Log4jLoggerFactory]

C:\hbase-2.2.5\bin>
```

**Step 8:**Type “hbase shell” to open the hbase shell**.**

Create a table by using “create”command

```
        at org.jruby.runtime.callsite.CachingCallSite.cacheAndCall(CachingCallSite.java:338)
        at org.jruby.runtime.callsite.CachingCallSite.call(CachingCallSite.java:163)
        at org.jruby.ir.interpreter.InterpreterEngine.processCall(InterpreterEngine.java:314
        at org.jruby.ir.interpreter.StartupInterpreterEngine.interpret(StartupInterpreterEng
        at org.jruby.ir.interpreter.InterpreterEngine.interpret(InterpreterEngine.java:77)
        at org.jruby.internal.runtime.methods.MixedModeIRMethod.INTERPRET_METHOD(MixedModeIR
        at org.jruby.internal.runtime.methods.MixedModeIRMethod.call(MixedModeIRMethod.java:
        at org.jruby.internal.runtime.methods.DynamicMethod.call(DynamicMethod.java:192)
        at org.jruby.runtime.callsite.CachingCallSite.cacheAndCall(CachingCallSite.java:318)
        at org.jruby.runtime.callsite.CachingCallSite.call(CachingCallSite.java:131)
        at C_3a_.Users.as5272.Documents.hbase_minus_2_dot_2_dot_5.bin.hirb.invokeOther172:pr
2.2.5\bin\hirb.rb:190)
        at C_3a_.Users.as5272.Documents.hbase_minus_2_dot_2_dot_5.bin.hirb.RUBY$script(C:\Us
b:190)
        at java.lang.invoke.MethodHandle.invokeWithArguments(MethodHandle.java:627)
        at org.jruby.ir.Compiler$1.load(Compiler.java:95)
        at org.jruby.Ruby.runScript(Ruby.java:828)
        at org.jruby.Ruby.runNormally(Ruby.java:747)
        at org.jruby.Ruby.runNormally(Ruby.java:765)
        at org.jruby.Ruby.runFromMain(Ruby.java:578)
        at org.jruby.Main.doRunFromMain(Main.java:417)
        at org.jruby.Main.internalRun(Main.java:305)
        at org.jruby.Main.run(Main.java:232)
        at org.jruby.Main.main(Main.java:204)

Took 0.0130 seconds
'stty' is not recognized as an internal or external command,
operable program or batch file.
hbase(main):001:0> create 'demo1','check'
Created table demo1
Took 4.3780 seconds
=> Hbase::Table - demo1
hbase(main):002:0>
```

## RESULT:

Thus the installation of hbase along with practice examples was executed successfully.

| EX.NO:6b<br><br>26/03/2024 | INSTALLATION OF THRIFT ALONG WITH PRACTICE EXAMPLES |
|---|---|

## AIM:

To install hbase along with practice examples.

## PROCEDURE:

**Step 1: Install dependencies**

- To use Java you will need to install Apache Ant
  - sudo apt-get install ant
- Installing required tools and libraries
  - sudo apt-get install libboost-dev libboost-test-dev libboost-program-options-dev libboost-filesystem-dev libboost-thread-dev libevent-dev automake libtool flex bison pkgconfig g++ libssl-dev
- You can check for specific requirements for each language you wish to use here: http://thrift.apache.org/docs/install/

**Step 2: Download Thrift**

- Download Thrift: http://thrift.apache.org/download
- Copy the downloaded file into the desired directory and untar the file

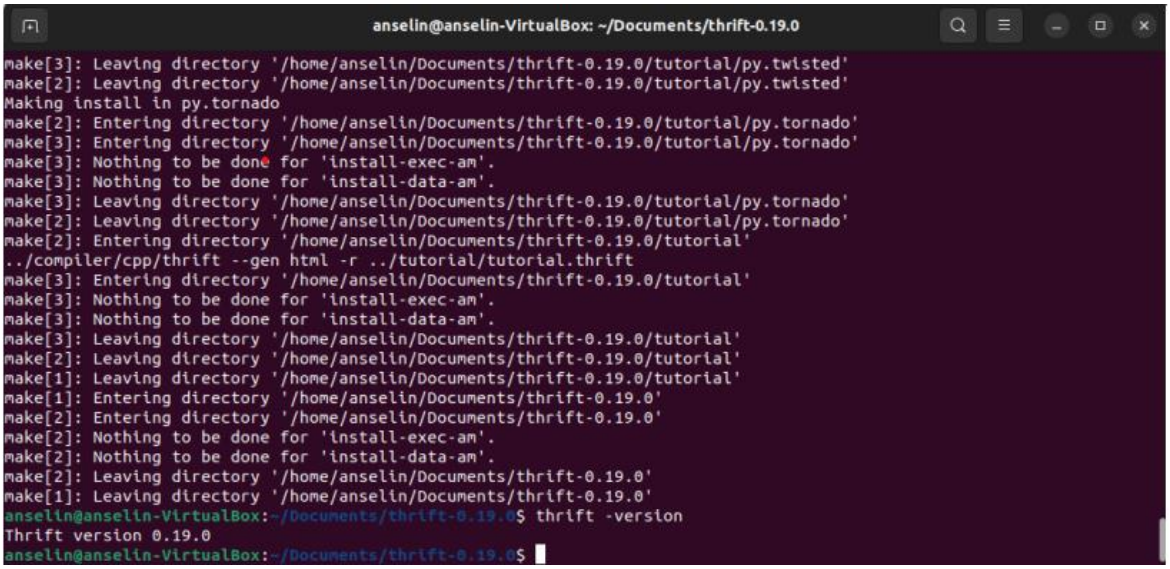
  **"tar -xvf thrift-0.9.3.tar.gz"**

**Step 3: Configure Thrift**

- For an Ubuntu linux distribution you just need to go to the thrift directory and type
  - ./bootstrap.sh
  - ./configure
- At the end of the output you should be able to see a list of all the libraries that are currently built in your system and ready to use with your desired programming languages. If a component is missing you should download the missing language and repeat the above step.
- Here http://thrift.apache.org/docs/install/debian/ you can find all the packages you might need to support your desired language in case some of them are missing.

**Step 4: Build and Run Thrift**

- On the same directory run make to build Thrift
  - sudo make
- (Optional) Run the test suite if you want
  - sudo make check
- And finally you are ready to install Thrift by running
  - sudo make install
- To verify that you have succesfully installed Thrift just type "thrift –version"




**RESULT:**

Thus the installation of hbase along with practice examples was executed successfully.

| EX NO:7<br><br>28/03/2024 | PRACTICE IMPORTING AND EXPORTING DATA FROM VARIOUS DATABASES |
|---|---|

## AIM:

Practice importing and exporting data from various databases.

## PREREQUISITES:

JAVA: You need to install the Java 8 package on your system.

HADOOP: You require Hadoop latest version.

Mysql.

Hive.

## PROCEDURE:

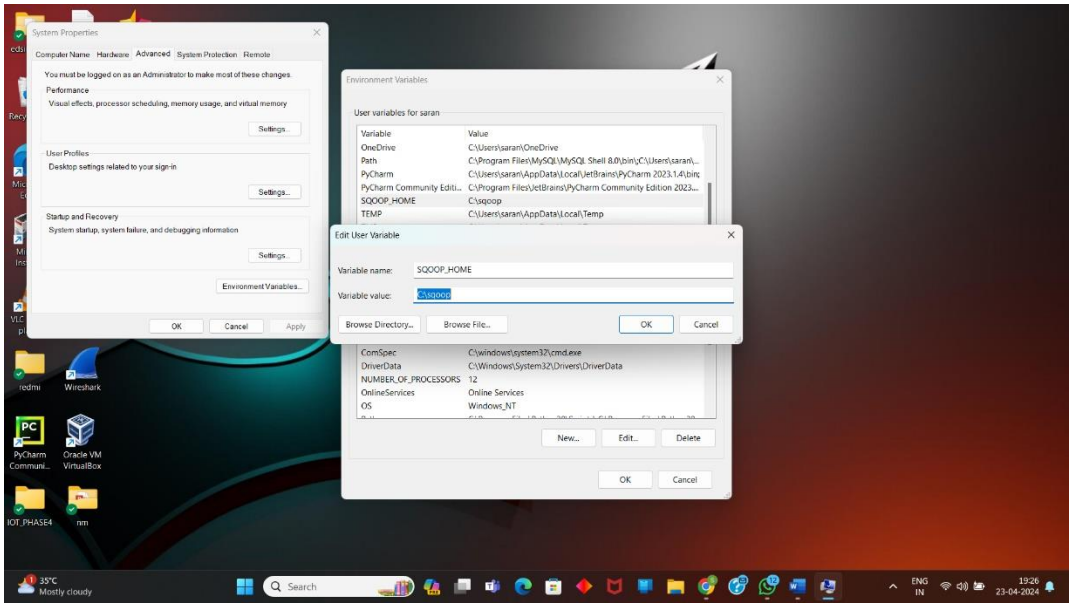
Step 1: Start hadoop

Step 2: Download sqoop

Download sqoop-1.4.6.bin.tar.gz file from apache website. And extract the files.

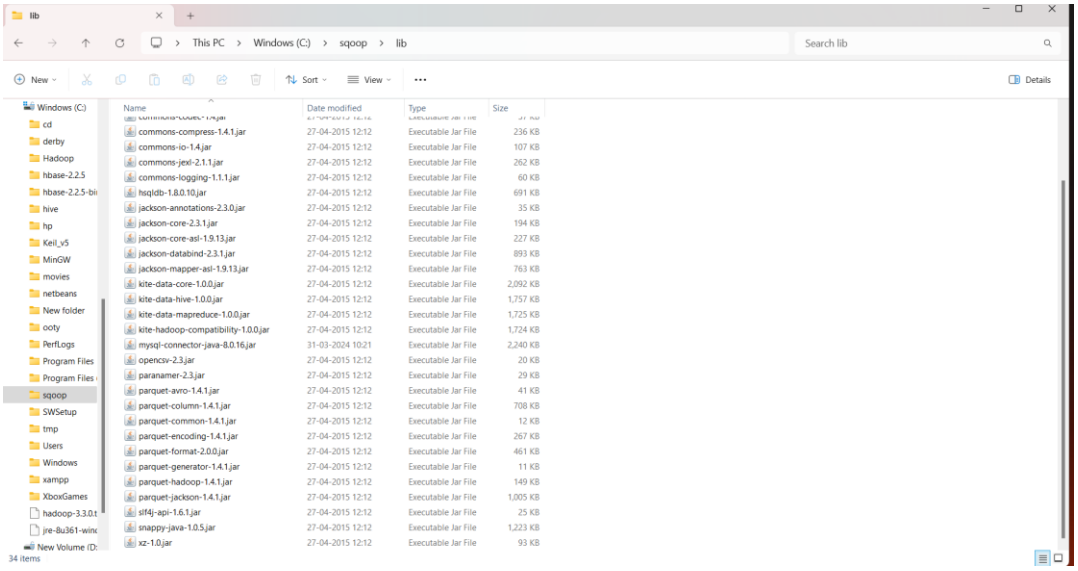

Step 3: Configure user variables

**Variable name: SQOOP_HOME**



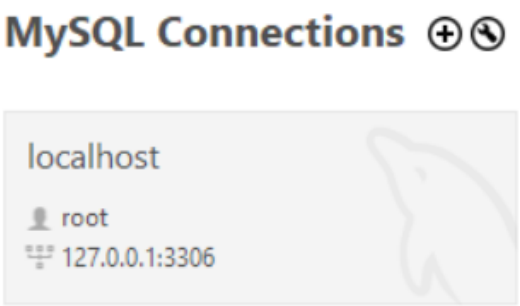

Step 4: **download mysql-connector**

- download mysql-connector-java-5.1.42.zip
- extract the file and copy the file and paste it in the path “C:\sqoop\lib”

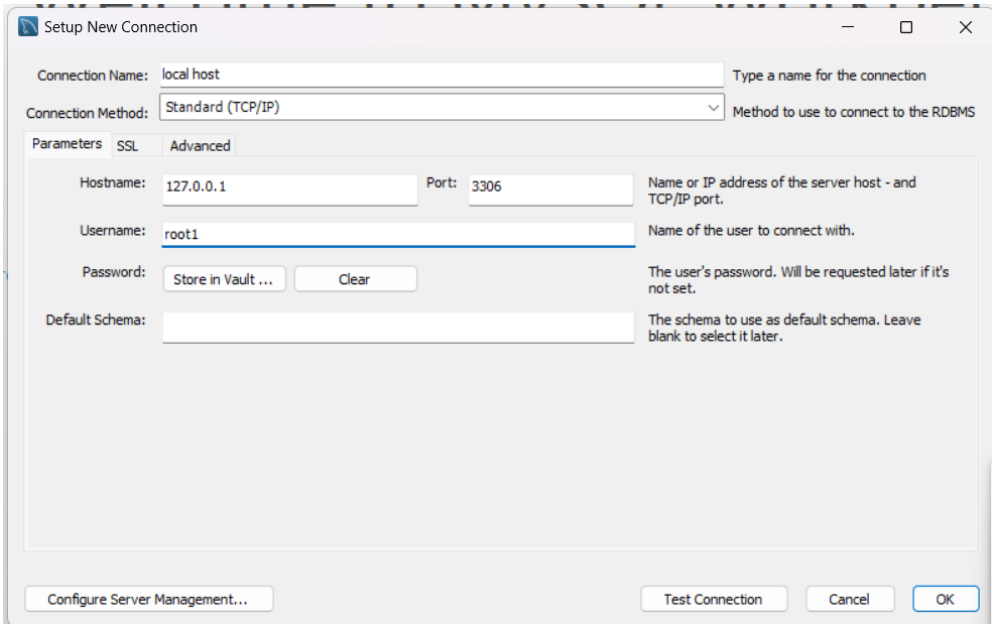



Step 5: Download mysql work bench.

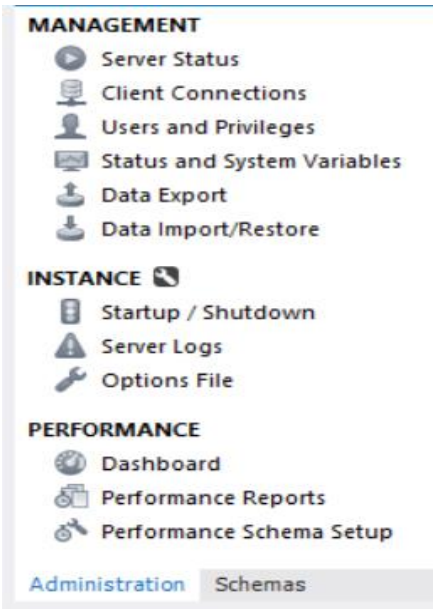
- To create a new connection click on to the plus icon.



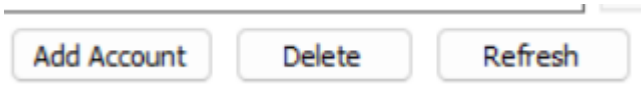
- Configure as the following to set up the connection.



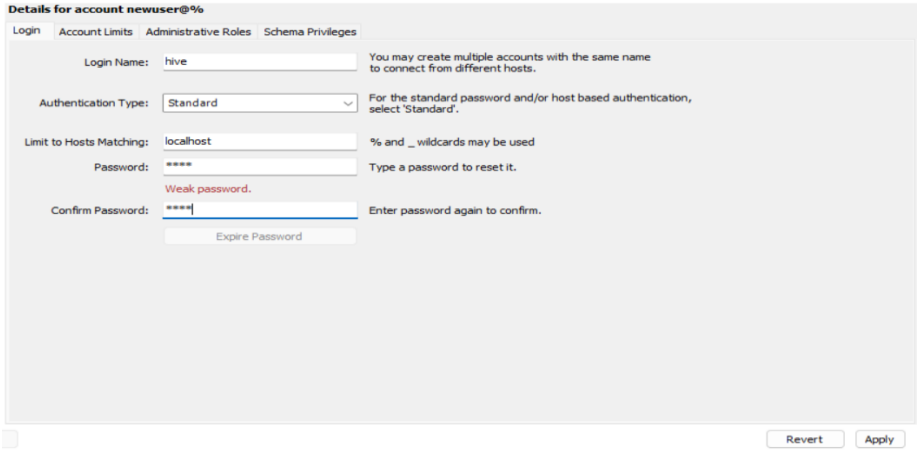

- Click the created localhost and in the left side go to administration setting and go to users and privileges



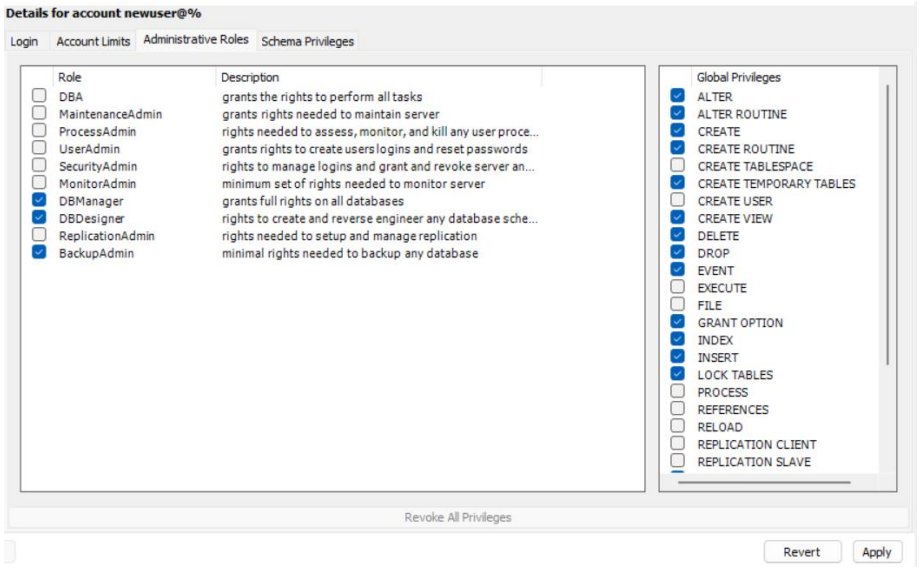

- Go to add account.



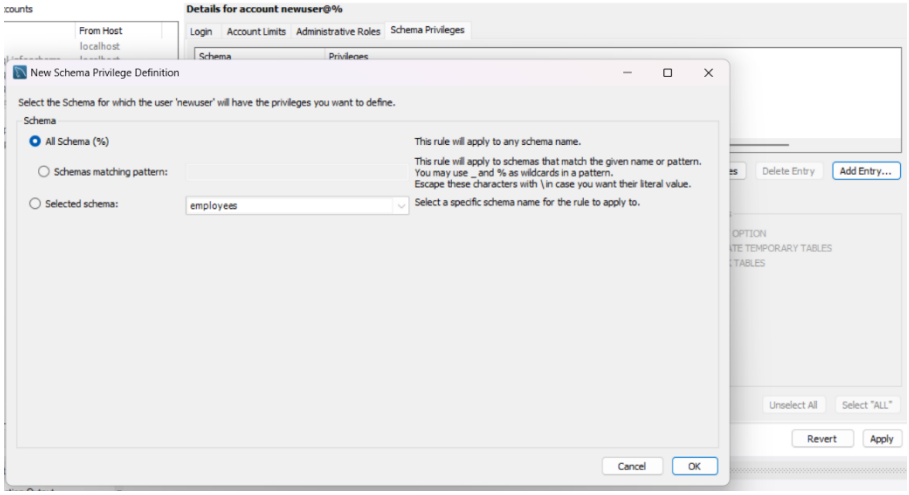
- Configure the following.



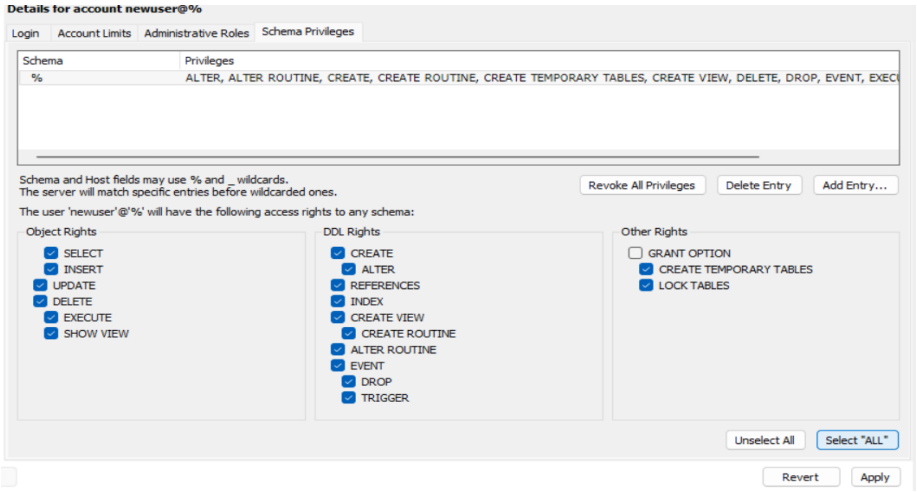

- In administrative roles select the following DB manager ,DB designer and Backup Admin.




- In the schema privileges select the employee data set an apply changes.



- In addition, select all and apply changes

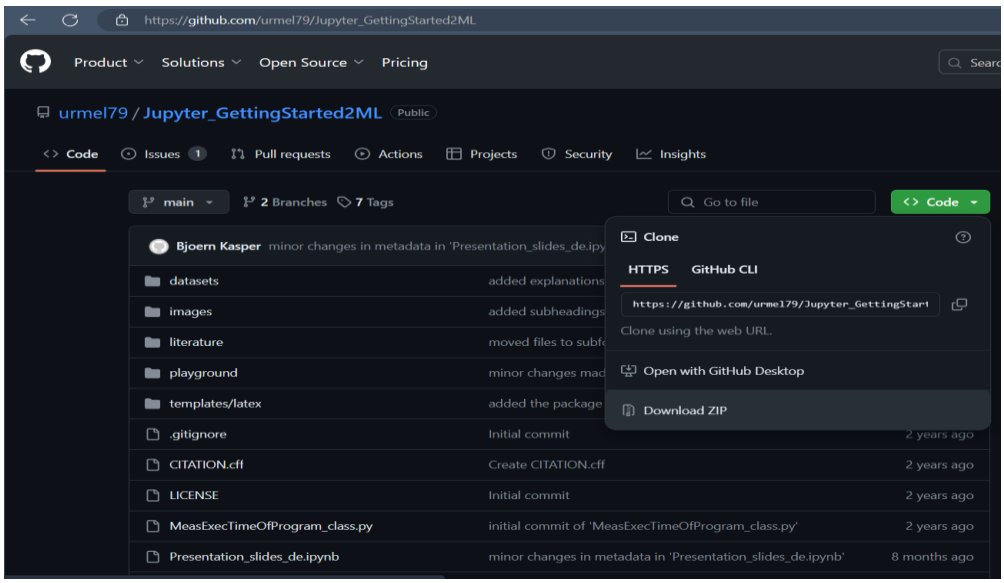



Step 6: Repeat step 5 for installation of Sqoop.

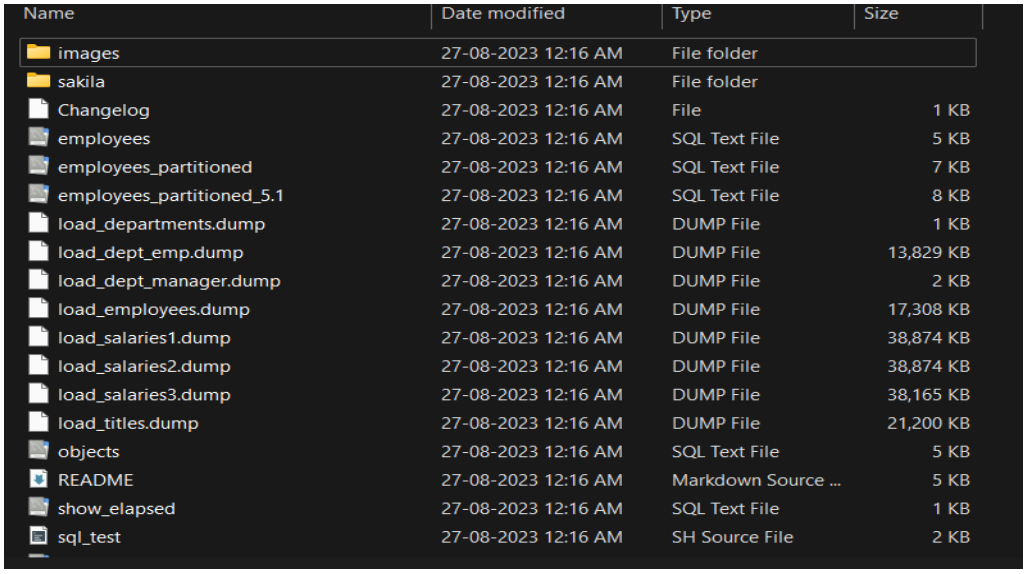
Step 7: To add the data set into the mysql database.

Do the following:

- Go to the following github link : [GitHub - urmel79/Jupyter_GettingStarted2ML: This Getting Started Tutorial systematically demonstrates the typical ML work process step-by-step using the powerful and performant Support Vector Classifier (SVC) and the beginner-friendly Iris Dataset. Furthermore, the selection of the correct SVC kernel and its parameters are described and their effects on the classification result are shown.]
- Download the data set.




- Extract the zip file.
- Copy all the db master contents.



| Name | Date modified | Type | Size |
|---|---|---|---|
| images | 27-08-2023 12:16 AM | File folder | |
| sakila | 27-08-2023 12:16 AM | File folder | |
| Changelog | 27-08-2023 12:16 AM | File | 1 KB |
| employees | 27-08-2023 12:16 AM | SQL Text File | 5 KB |
| employees_partitioned | 27-08-2023 12:16 AM | SQL Text File | 7 KB |
| employees_partitioned_5.1 | 27-08-2023 12:16 AM | SQL Text File | 8 KB |
| load_departments.dump | 27-08-2023 12:16 AM | DUMP File | 1 KB |
| load_dept_emp.dump | 27-08-2023 12:16 AM | DUMP File | 13,829 KB |
| load_dept_manager.dump | 27-08-2023 12:16 AM | DUMP File | 2 KB |
| load_employees.dump | 27-08-2023 12:16 AM | DUMP File | 17,308 KB |
| load_salaries1.dump | 27-08-2023 12:16 AM | DUMP File | 38,874 KB |
| load_salaries2.dump | 27-08-2023 12:16 AM | DUMP File | 38,874 KB |
| load_salaries3.dump | 27-08-2023 12:16 AM | DUMP File | 38,165 KB |
| load_titles.dump | 27-08-2023 12:16 AM | DUMP File | 21,200 KB |
| objects | 27-08-2023 12:16 AM | SQL Text File | 5 KB |
| README | 27-08-2023 12:16 AM | Markdown Source ... | 5 KB |
| show_elapsed | 27-08-2023 12:16 AM | SQL Text File | 1 KB |
| sql_test | 27-08-2023 12:16 AM | SH Source File | 2 KB |

- Go to the following folder and past the contents C:\Program Files\MySQL\MySQL Server 8.0\bin.
- After the above task open cmd in the same folder and type the below cmd.

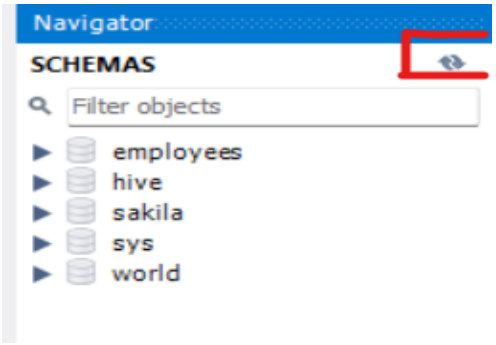
```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -t -p
Enter password: ****
```

```
mysql> source employees.sql
Query OK, 8 rows affected (0.13 sec)
```

- After this wait for the configuration message.

```
+---------------------+
| data_load_time_diff |
+---------------------+
| 00:01:07            |
+---------------------+
1 row in set (0.01 sec)
```

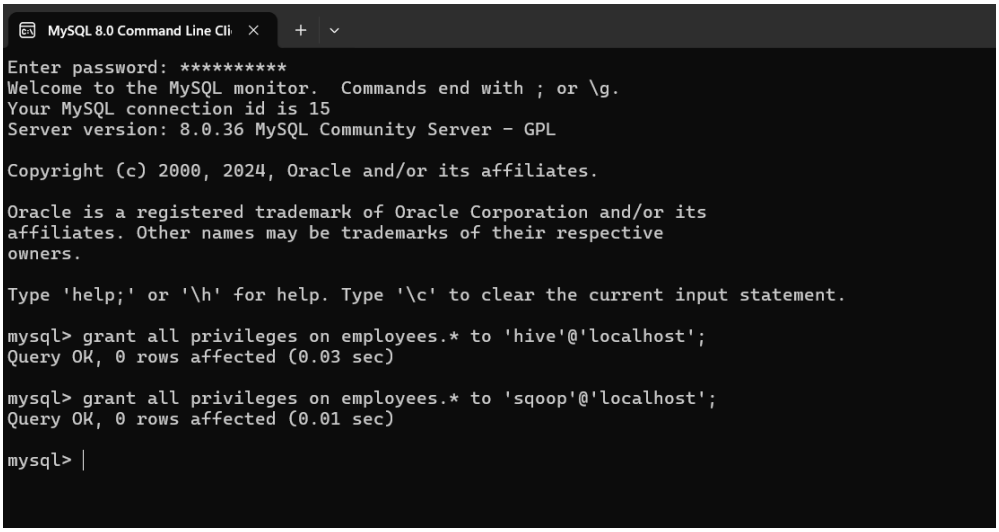
- Then refresh the mysql connector page



- It will import the data set into the mysql.

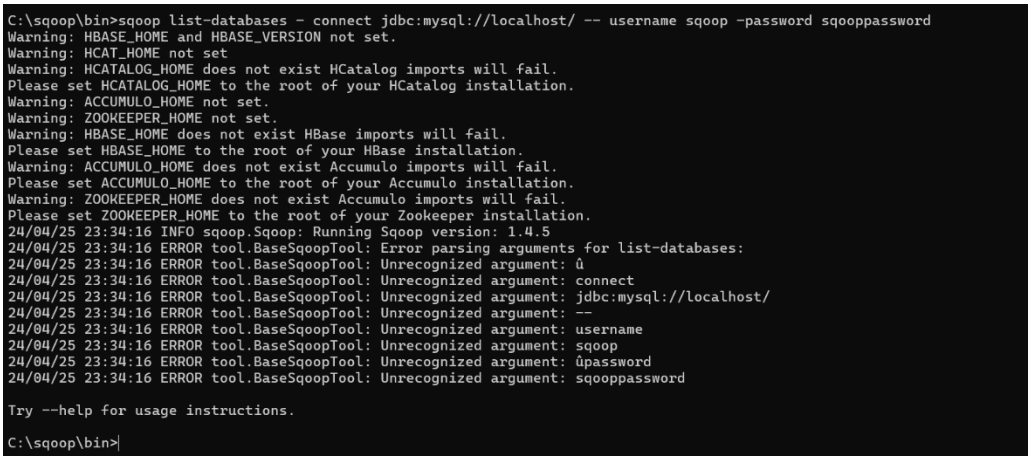
Step 8: Open **mysql command line** and grant all privileges to user hive and user sqoop.

Use the following cmds.




Step 9: run sqoop

- Open sqoop"c:\sqoop\bin\" in terminal
- Check the database "sqoop list-databases – connect jdbc:mysql://localhost/ -- username sqoop –password sqooppassword".




Step 10: **import data**

Import data from mysql server into warehouse by using "sqoop import –connect jdbc:mysql://localhost/employees –table departments –username hive –password hivepassword –hive-import –fields-terminated-by ","

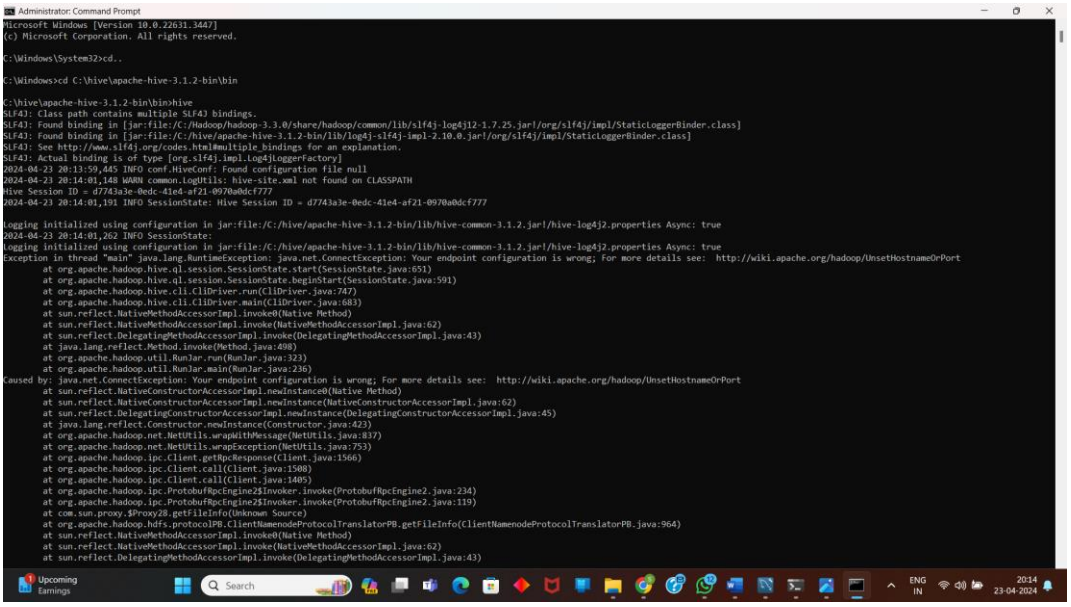
```
C:\sqoop\bin>sqoop import -connect jdbc:mysql://localhost/employees -table departments -username hive -password hivepassword -hive-import -fields-terminated
-by “,”
Warning: HBASE_HOME and HBASE_VERSION not set.
Warning: HCAT_HOME not set
Warning: HCATALOG_HOME does not exist HCatalog imports will fail.
Please set HCATALOG_HOME to the root of your HCatalog installation.
Warning: ACCUMULO_HOME not set.
Warning: ZOOKEEPER_HOME not set.
Warning: HBASE_HOME does not exist HBase imports will fail.
Please set HBASE_HOME to the root of your HBase installation.
Warning: ACCUMULO_HOME does not exist Accumulo imports will fail.
Please set ACCUMULO_HOME to the root of your Accumulo installation.
Warning: ZOOKEEPER_HOME does not exist Accumulo imports will fail.
Please set ZOOKEEPER_HOME to the root of your Zookeeper installation.
24/04/25 23:35:47 INFO sqoop.Sqoop: Running Sqoop version: 1.4.5
24/04/25 23:35:47 ERROR tool.BaseSqoopTool: Error parsing arguments for import:
24/04/25 23:35:47 ERROR tool.BaseSqoopTool: Unrecognized argument: ûconnect
24/04/25 23:35:47 ERROR tool.BaseSqoopTool: Unrecognized argument: jdbc:mysql://localhost/employees
24/04/25 23:35:47 ERROR tool.BaseSqoopTool: Unrecognized argument: ûtable
24/04/25 23:35:47 ERROR tool.BaseSqoopTool: Unrecognized argument: departments
24/04/25 23:35:47 ERROR tool.BaseSqoopTool: Unrecognized argument: ûusername
24/04/25 23:35:47 ERROR tool.BaseSqoopTool: Unrecognized argument: hive
24/04/25 23:35:47 ERROR tool.BaseSqoopTool: Unrecognized argument: ûpassword
24/04/25 23:35:47 ERROR tool.BaseSqoopTool: Unrecognized argument: hivepassword
24/04/25 23:35:47 ERROR tool.BaseSqoopTool: Unrecognized argument: ûhive-import
24/04/25 23:35:47 ERROR tool.BaseSqoopTool: Unrecognized argument: ûfields-terminated-by
24/04/25 23:35:47 ERROR tool.BaseSqoopTool: Unrecognized argument: ô
24/04/25 23:35:47 ERROR tool.BaseSqoopTool: Unrecognized argument: ö

Try --help for usage instructions.

C:\sqoop\bin>
```

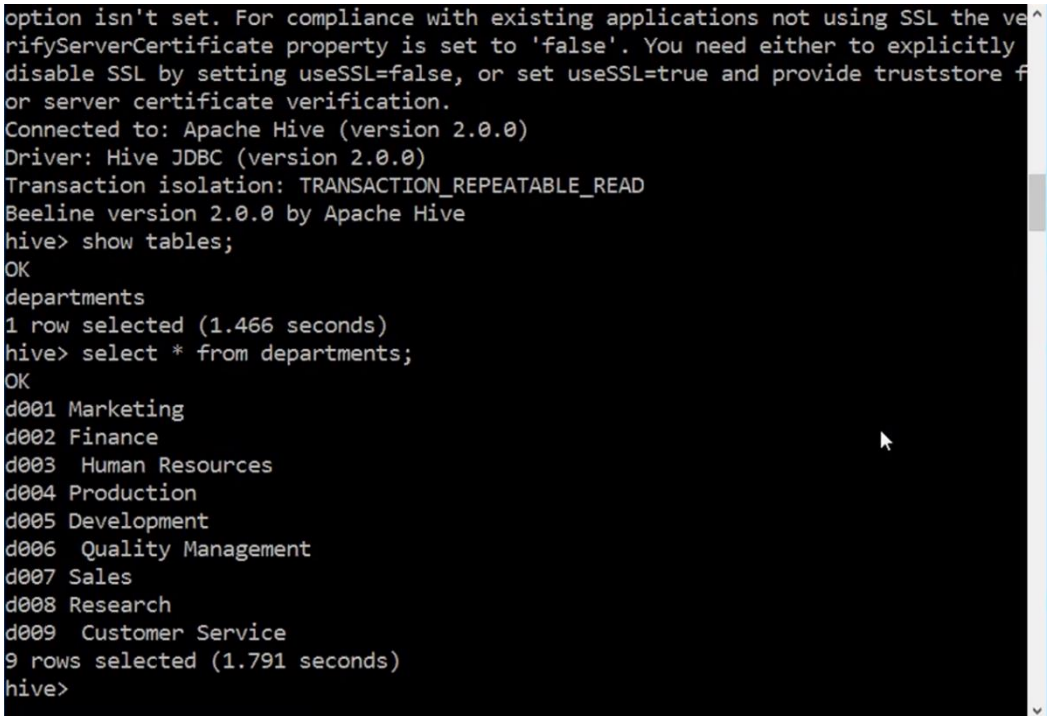
Step 11: Run hive.

- Open terminal and direct to the path “c:/hive/bin” and run hive by using “hive” command.




Step 12: export data

- Check data in hive by using “**show table;**” command
- Check data in department table by using “**select * from departments:**”




## RESULT:

Thus practice importing and exporting data from various databases was executed successfully.