NAME: Ayush S. Tiwari

ROLL NUMBER: 511

COURSE: MSc CS

SUBJECT: BUSINESS INTELLIGENCE & BIG DATA ANALYSIS

PRACTICAL: 1-9

SR No	Name	Date	Sign
1	To Install Cloudera QuickStart VM on VMware		
2	Map-Reduce Program for WordCount Problem		
3	PIG Script for Solving Counting Problems		
4	Install HBase and Use HBase Data Model to Store and Retrieve Databases		
5	Install Hive and Use Hive to Create and Store Structured Databases		
6	Construct Different Types of K-Shingles for Given Document		
7	Measuring Similarity Among Documents and Detecting Passages Which Have Been Reused		
8	Computing n-moment		
9	Alon-Matias-Szegedy Algorithm		

Practical 1: To Install Cloudera QuickStart VM on VMware

Cloudera is software that provides a platform for data analytics, data warehousing, and machine learning. Initially, Cloudera started as an open-source Apache Hadoop distribution project, commonly known as Cloudera Distribution for Hadoop or CDH. It contains Apache Hadoop and other related projects where all the components are 100% open-source under Apache License.

Cloudera provides virtual machine images of complete Apache Hadoop clusters, making it easy to get started with Cloudera CDH. This assignment covers the following topics related to Cloudera QuickStart VM Installation:

1. What is Cloudera QuickStart VM?
2. Prerequisites for Cloudera QuickStart VM Installation
3. Downloading the Cloudera QuickStart VM
4. Cloudera QuickStart VM Installation on Windows

What is Cloudera QuickStart VM?

Cloudera QuickStart VM includes everything you need for using CDH, Impala, Cloudera Search, and Cloudera Manager. The Cloudera QuickStart VM uses a package-based install that allows you to work with or without the Cloudera Manager. It provides a sample of Cloudera's platform for "Big Data."

Prerequisites for Cloudera QuickStart VM Installation

- A virtual machine such as Oracle VirtualBox or VMware
- RAM of 12+ GB, that is 4+ GB for the operating system and 8+ GB for Cloudera
- 80 GB hard disk
- Oracle VirtualBox downloaded from https://www.virtualbox.org/wiki/Downloads and installed on your system

Downloading the Cloudera QuickStart VM

- Cloudera QuickStart VMs are available as Zip archives in VirtualBox, VMware, and KVM formats. To download the VM, go to https://www.cloudera.com/downloads.html and select the appropriate version of CDH that you require.
- Click on the "GET IT NOW" button, and it will prompt you to fill in your details.

- Once the file is downloaded, go to the download folder and unzip the files. They can then be used to set up a single-node Cloudera cluster.

The two virtual images of Cloudera QuickStart VM are shown below:

Name	Date modified	Type	Size
cloudera-quickstart-vm-5.13.0-0-vmware	20-03-2023 04:56 PM	File folder	
cloudera-quickstart-vm-5.13.0-0-vmware.zip	17-03-2023 12:13 AM	ZIP File	56,22,959 KB

Name	Date modified	Type	Size
cloudera-quickstart-vm-5.13.0-0-vmware.nvram	23-10-2017 05:40 PM	VMware Virtual M...	9 KB
cloudera-quickstart-vm-5.13.0-0-vmware.vmdk	23-10-2017 05:33 PM	VMware virtual dis...	2 KB
cloudera-quickstart-vm-5.13.0-0-vmware.vmsd	23-10-2017 04:51 PM	VMware snapshot ...	0 KB
cloudera-quickstart-vm-5.13.0-0-vmware.vmx	23-10-2017 05:40 PM	VMware virtual m...	3 KB
cloudera-quickstart-vm-5.13.0-0-vmware.vmxf	23-10-2017 04:51 PM	VMware Team Me...	1 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s001.vmdk	23-10-2017 05:42 PM	VMware virtual dis...	7,01,184 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s002.vmdk	23-10-2017 05:42 PM	VMware virtual dis...	25,45,280 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s003.vmdk	23-10-2017 05:42 PM	VMware virtual dis...	17,31,136 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s004.vmdk	23-10-2017 05:42 PM	VMware virtual dis...	94,016 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s005.vmdk	23-10-2017 05:42 PM	VMware virtual dis...	6,66,240 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s006.vmdk	23-10-2017 05:42 PM	VMware virtual dis...	15,64,352 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s007.vmdk	23-10-2017 05:43 PM	VMware virtual dis...	1,56,288 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s008.vmdk	23-10-2017 05:43 PM	VMware virtual dis...	41,408 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s009.vmdk	23-10-2017 05:43 PM	VMware virtual dis...	704 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s010.vmdk	23-10-2017 05:43 PM	VMware virtual dis...	1,22,112 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s011.vmdk	23-10-2017 05:43 PM	VMware virtual dis...	40,128 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s012.vmdk	23-10-2017 05:43 PM	VMware virtual dis...	1,024 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s013.vmdk	23-10-2017 05:43 PM	VMware virtual dis...	9,55,520 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s014.vmdk	23-10-2017 05:43 PM	VMware virtual dis...	4,08,320 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s015.vmdk	23-10-2017 05:43 PM	VMware virtual dis...	960 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s016.vmdk	23-10-2017 05:43 PM	VMware virtual dis...	512 KB
cloudera-quickstart-vm-5.13.0-0-vmware-s017.vmdk	23-10-2017 05:43 PM	VMware virtual dis...	128 KB

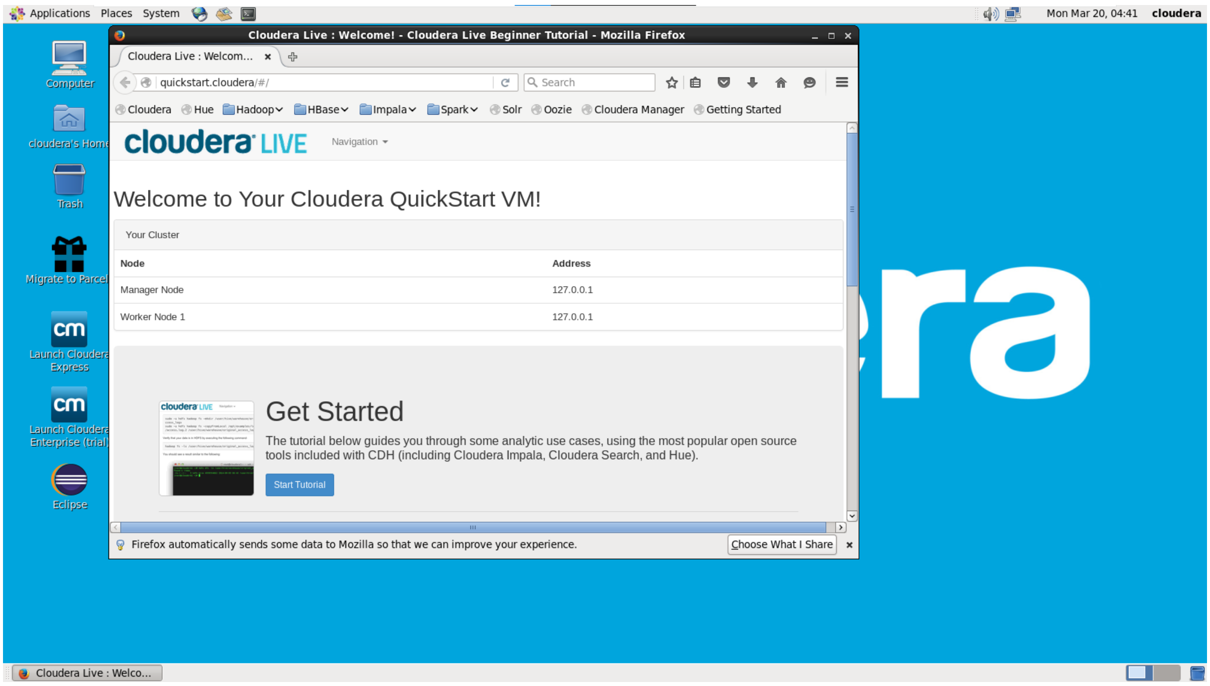
- Now that the downloading process is done with, let's move forward with this Cloudera QuickStart VM Installation guide and see the actual process.

Cloudera QuickStart VM Installation

- Before setting up the Cloudera Virtual Machine, you would need to have a virtual machine such as VMware or Oracle VirtualBox on your system.
- In this case, we are using Oracle VirtualBox to set up the Cloudera QuickStart VM.
- In order to download and install the Oracle VirtualBox on your operating system, click on the following link: Oracle VirtualBox(https://www.virtualbox.org/wiki/Downloads).

- To set up the Cloudera QuickStart VM in your Oracle VirtualBox Manager, click on file with extension as ".VMX " and then it will automatically open in VMware Workstation.
- Once complete you can see the Cloudera QuickStart VM on the left side panel.

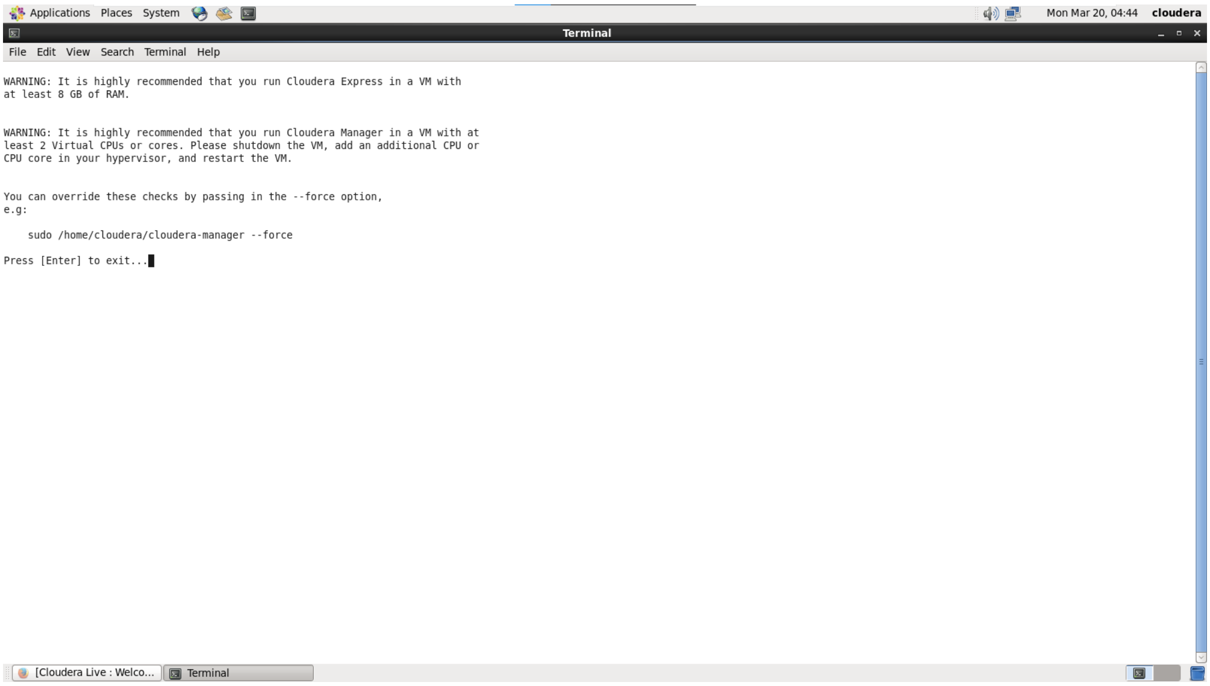
- The next step will be going ahead and starting the machine by clicking the 'Start' symbol on top.
- Once your machine comes on, it will look like this:



•Next, we have to follow a few steps to gain admin console access. You need to click on the terminal present on top of the desktop screen, and type in the following:

1. hostname # This shows the hostname which will be quickstart.cloudera
2. hdfs dfs -ls / # Checks if you have access and if your cluster is working. It displays what exists on your HDFS location by default
3. service cloudera-scm-server status # Tells what command you have to type to use cloudera express free
4. su - #Login as root
5. service cloudera-scm-server status # The password for root is cloudera

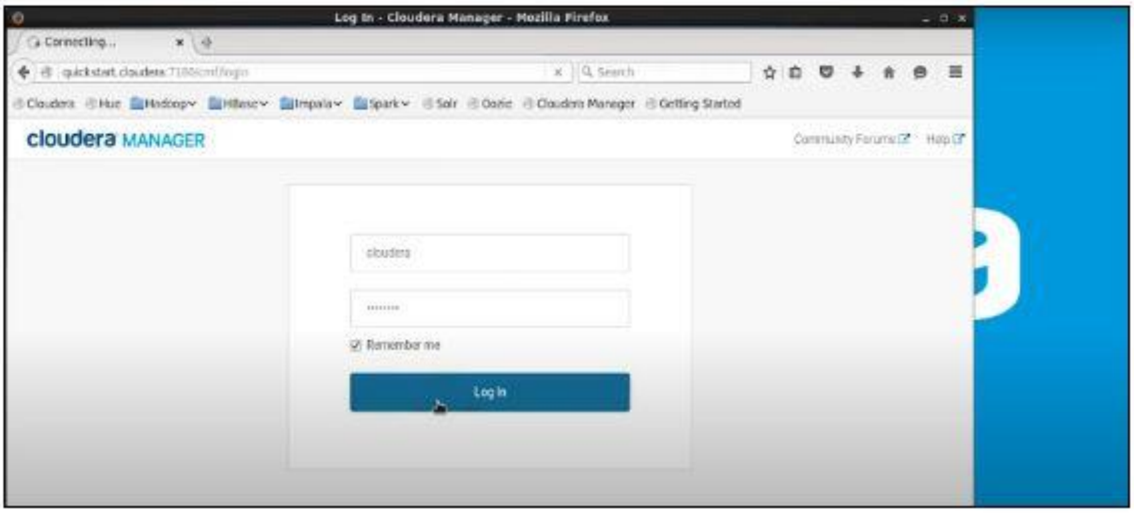
- Once you see that your HDFS access is working fine, you can close the terminal. Then, you have to click on the following icon that says 'Launch Cloudera Express'.



- You are required to copy the command, and run it on a separate terminal. Hence, open a new terminal, and use the below command to close the Cloudera based services. It will restart the services, after which you can access your admin console.

- Now that our deployment has been configured, client configurations have also been deployed. Additionally, it has restarted the Cloudera Management Service, which gives access to the Cloudera QuickStart admin console with the help of a username and password.

- Go on and open up the browser and change the port number to 7180.
- You can log in to the Cloudera Manager by providing your username and password.



- You can go ahead and restart the services now. It will ensure that the cluster becomes accessible either by Hue as a web interface or Cloudera QuickStart Terminal, where you can write your commands.

Practical 2: Map-Reduce Program for WordCount Problem

Commands:

```
[cloudera@quickstart ~]$ hdfs dfs -ls /
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -mkdir /inputdirectory
[cloudera@quickstart ~]$ hdfs dfs -ls /
[cloudera@quickstart ~]$ cat>/home/cloudera/processfile.txt
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -put
/home/cloudera/processfile.txt /inputdirectory
[cloudera@quickstart ~]$ hdfs dfs -ls /inputdirectory
[cloudera@quickstart ~]$ hadoop jar /home/cloudera/WordCount.jar WordCount
/inputdirectory/processfile.txt /out1
[cloudera@quickstart ~]$ hdfs dfs -ls /out1
[cloudera@quickstart ~]$ hdfs dfs -cat /out1/part-r-00000
```

OUTPUT

```
[cloudera@quickstart ~]$ hdfs dfs -ls /
sudo -u Found 6 items
drwxrwxrwx   - hdfs  supergroup          0 2017-10-23 10:29 /benchmarks
drwxr-xr-x   - hbase supergroup          0 2023-03-20 05:39 /hbase
drwxr-xr-x   - solr  solr                0 2017-10-23 10:32 /solr
drwxrwxrwt   - hdfs  supergroup          0 2023-03-20 04:38 /tmp
drwxr-xr-x   - hdfs  supergroup          0 2017-10-23 10:31 /user
drwxr-xr-x   - hdfs  supergroup          0 2017-10-23 10:31 /var
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -mkdir /inputdirectory
^[[A[cloudera@quickstart hdfs dfs -ls /
Found 7 items
drwxrwxrwx   - hdfs  supergroup          0 2017-10-23 10:29 /benchmarks
drwxr-xr-x   - hbase supergroup          0 2023-03-20 05:39 /hbase
drwxr-xr-x   - hdfs  supergroup          0 2023-03-20 05:52 /inputdirectory
drwxr-xr-x   - solr  solr                0 2017-10-23 10:32 /solr
drwxrwxrwt   - hdfs  supergroup          0 2023-03-20 04:38 /tmp
drwxr-xr-x   - hdfs  supergroup          0 2017-10-23 10:31 /user
drwxr-xr-x   - hdfs  supergroup          0 2017-10-23 10:31 /var
[cloudera@quickstart ~]$ cat>/home/cloudera/processfile.txt
Hii How are you Hii i am fine^C
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -put /home/cloudera/processfile.
txt /inputdirectory
^[[A^[[A[cloudera@quicksthdfs dfs -ls /^C
[cloudera@quickstart ~]$ hdfs dfs -ls /inputdirectory
Found 1 items
-rw-r--r--   1 hdfs supergroup          0 2023-03-20 05:53 /inputdirectory/proce
ssfile.txt
[cloudera@quickstart ~]$ hadoop jar /home/cloudera/WordCount.jar WordCount /inpu
tdirectory/processfile.txt /out1
23/03/20 06:00:28 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/20 06:00:28 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
23/03/20 06:00:29 INFO input.FileInputFormat: Total input paths to process : 1
23/03/20 06:00:29 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1281)
        at java.lang.Thread.join(Thread.java:1355)
```

```
        at java.lang.Thread.join(Thread.java:1355)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
23/03/20 06:00:29 INFO mapreduce.JobSubmitter: number of splits:1
23/03/20 06:00:29 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1679315869981_0001
23/03/20 06:00:30 INFO impl.YarnClientImpl: Submitted application application_1679315869981_0001
23/03/20 06:00:31 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1679315869981_0001/
23/03/20 06:00:31 INFO mapreduce.Job: Running job: job_1679315869981_0001
23/03/20 06:00:45 INFO mapreduce.Job: Job job_1679315869981_0001 running in uber mode : false
23/03/20 06:00:45 INFO mapreduce.Job:  map 0% reduce 0%
23/03/20 06:00:58 INFO mapreduce.Job:  map 100% reduce 0%
hdfs dfs -ls /out123/03/20 06:01:07 INFO mapreduce.Job:  map 100% reduce 100%
23/03/20 06:01:07 INFO mapreduce.Job: Job job_1679315869981_0001 completed succe
ssfully
23/03/20 06:01:07 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=6
                FILE: Number of bytes written=286727
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=127
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Other local map tasks=1
                Total time spent by all maps in occupied slots (ms)=9891
                Total time spent by all reduces in occupied slots (ms)=6400
                Total time spent by all map tasks (ms)=9891
                Total time spent by all reduce tasks (ms)=6400
                Total vcore-milliseconds taken by all map tasks=9891
                Total vcore-milliseconds taken by all reduce tasks=6400
                Total megabyte-milliseconds taken by all map tasks=10128384
                Total megabyte-milliseconds taken by all reduce tasks=6553600
        Map-Reduce Framework
                Map input records=0
                Map output records=0
                Map output bytes=0
                Map output materialized bytes=6
                Input split bytes=127
                Combine input records=0
```

```
                Reduce input groups=0
                Reduce shuffle bytes=6
                Reduce input records=0
                Reduce output records=0
                Spilled Records=0
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=176
                CPU time spent (ms)=1170
                Physical memory (bytes) snapshot=322920448
                Virtual memory (bytes) snapshot=3015028736
                Total committed heap usage (bytes)=195301376
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
[cloudera@quickstart ~]$ hdfs dfs -ls /out1
Found 2 items
-rw-r--r--   1 cloudera supergroup          0 2023-03-20 06:01 /out1/_SUCCESS
-rw-r--r--   1 cloudera supergroup          0 2023-03-20 06:01 /out1/part-r-0000
0
```

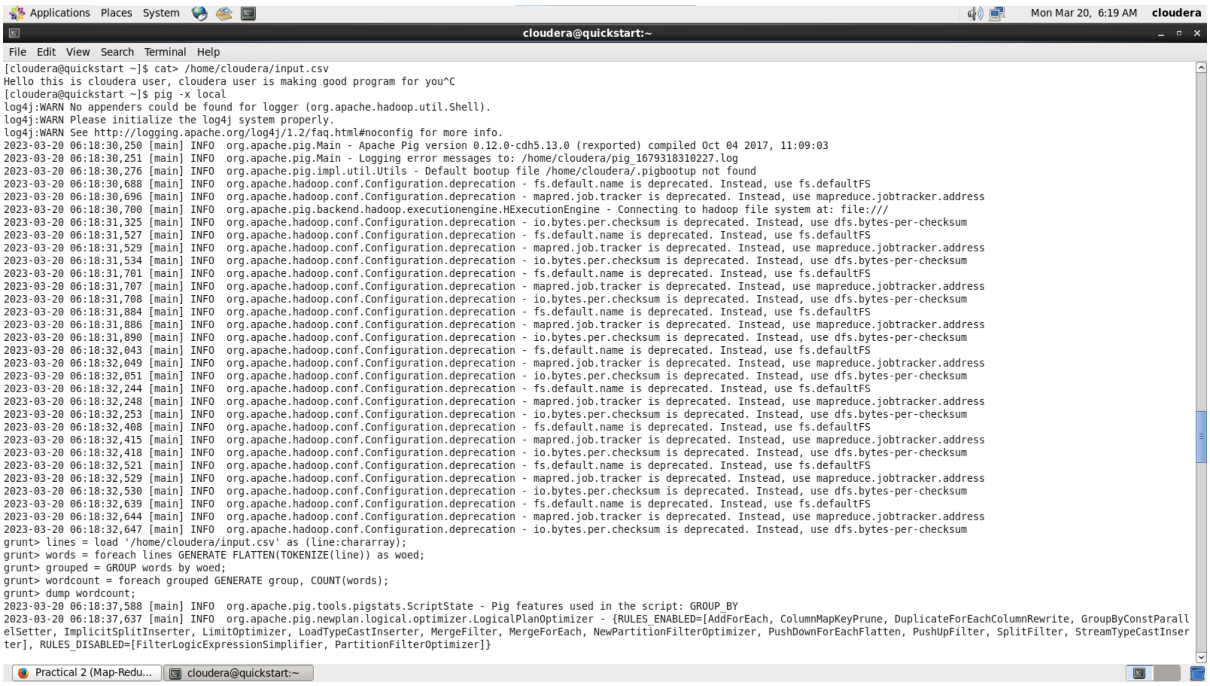
```
[cloudera@quickstart ~]$ hdfs dfs -cat /out1/part-r-00000
Hii     2
How     1
am      1
are     1
fine    1
i       1
u       1
[cloudera@quickstart ~]$
```

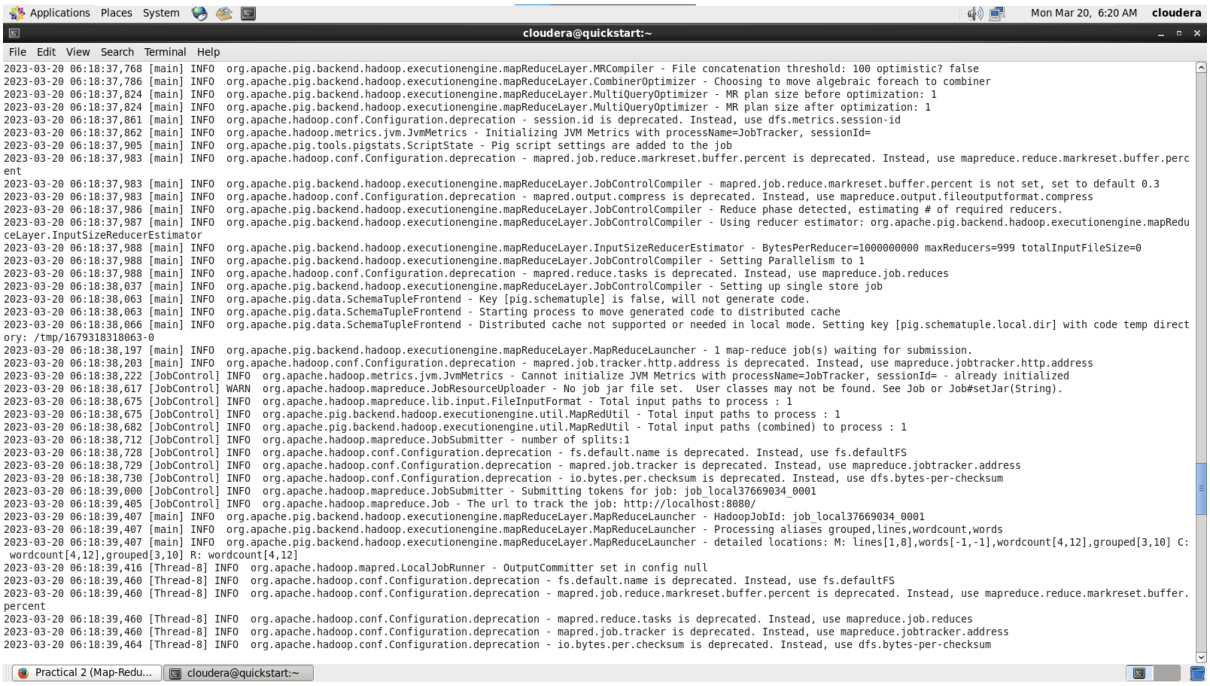
Practical 3: PIG Script for Solving Counting Problems

Commands:

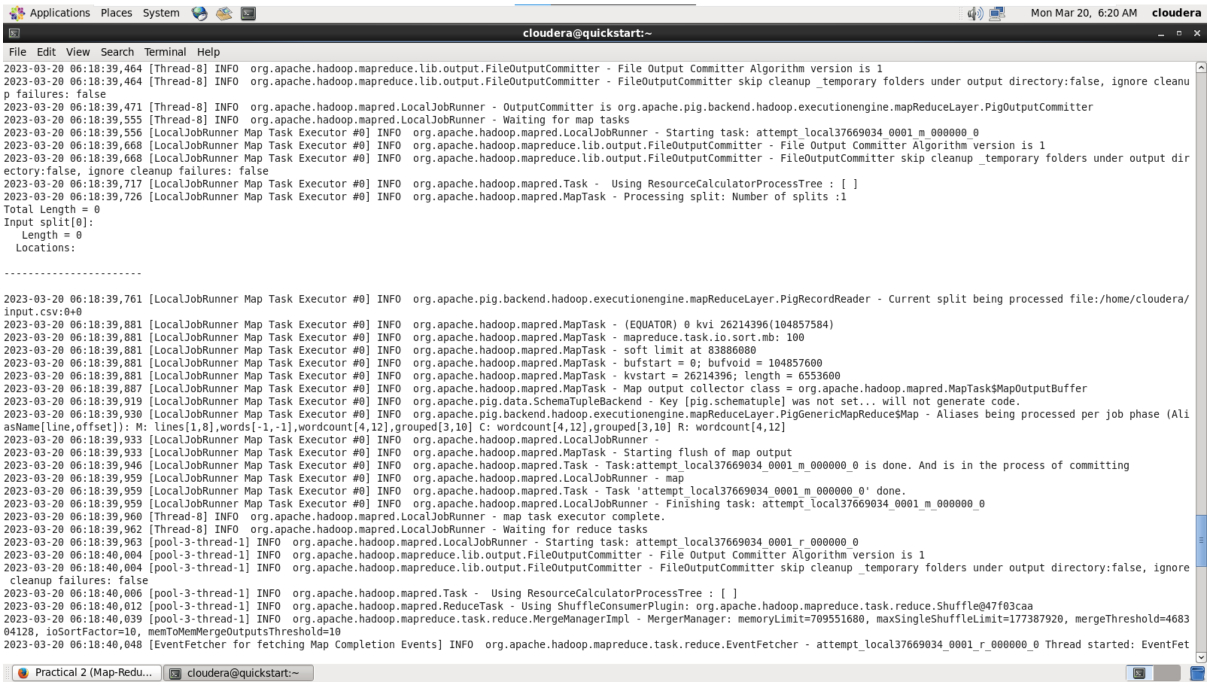
```
cat> /home/cloudera/input.csv
cat  /home/cloudera/input.csv
pig -x local
lines = load '/home/cloudera/input.csv' as (line:chararray);
words = foreach lines GENERATE FLATTEN(TOKENIZE(line)) as woed;
grouped = GROUP words by woed;
wordcount = foreach grouped GENERATE group, COUNT(words);
dump wordcount;
```

OUTPUT





```
2023-03-20 06:18:37,768 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenation threshold: 100 optimistic? false
2023-03-20 06:18:37,786 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.CombinerOptimizer - Choosing to move algebraic foreach to combiner
2023-03-20 06:18:37,824 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size before optimization: 1
2023-03-20 06:18:37,824 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size after optimization: 1
2023-03-20 06:18:37,861 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - session.id is deprecated. Instead, use dfs.metrics.session-id
2023-03-20 06:18:37,862 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Initializing JVM Metrics with processName=JobTracker, sessionId=
2023-03-20 06:18:37,905 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig script settings are added to the job
2023-03-20 06:18:37,983 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.reduce.markreset.buffer.percent is deprecated. Instead, use mapreduce.reduce.markreset.buffer.perc
ent
2023-03-20 06:18:37,983 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - mapred.job.reduce.markreset.buffer.percent is not set, set to default 0.3
2023-03-20 06:18:37,983 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.output.compress is deprecated. Instead, use mapreduce.output.fileoutputformat.compress
2023-03-20 06:18:37,986 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Reduce phase detected, estimating # of required reducers.
2023-03-20 06:18:37,987 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Using reducer estimator: org.apache.pig.backend.hadoop.executionengine.mapRedu
ceLayer.InputSizeReducerEstimator
2023-03-20 06:18:37,988 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.InputSizeReducerEstimator - BytesPerReducer=1000000000 maxReducers=999 totalInputFileSize=0
2023-03-20 06:18:37,988 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Setting Parallelism to 1
2023-03-20 06:18:37,988 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
2023-03-20 06:18:38,037 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Setting up single store job
2023-03-20 06:18:38,063 [main] INFO  org.apache.pig.data.SchemaTupleFrontend - Key [pig.schematuple] is false, will not generate code.
2023-03-20 06:18:38,063 [main] INFO  org.apache.pig.data.SchemaTupleFrontend - Starting process to move generated code to distributed cache
2023-03-20 06:18:38,066 [main] INFO  org.apache.pig.data.SchemaTupleFrontend - Distributed cache not supported or needed in local mode. Setting key [pig.schematuple.local.dir] with code temp direct
ory: /tmp/1679318318063-0
2023-03-20 06:18:38,197 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 1 map-reduce job(s) waiting for submission.
2023-03-20 06:18:38,203 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker.http.address is deprecated. Instead, use mapreduce.jobtracker.http.address
2023-03-20 06:18:38,222 [JobControl] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2023-03-20 06:18:38,617 [JobControl] WARN  org.apache.hadoop.mapreduce.JobResourceUploader - No job jar file set.  User classes may not be found. See Job or Job#setJar(String).
2023-03-20 06:18:38,675 [JobControl] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2023-03-20 06:18:38,675 [JobControl] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2023-03-20 06:18:38,682 [JobControl] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths (combined) to process : 1
2023-03-20 06:18:38,712 [JobControl] INFO  org.apache.hadoop.mapreduce.JobSubmitter - number of splits:1
2023-03-20 06:18:38,728 [JobControl] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2023-03-20 06:18:38,729 [JobControl] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2023-03-20 06:18:38,730 [JobControl] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2023-03-20 06:18:39,000 [JobControl] INFO  org.apache.hadoop.mapreduce.JobSubmitter - Submitting tokens for job: job_local37669034_0001
2023-03-20 06:18:39,405 [JobControl] INFO  org.apache.hadoop.mapreduce.Job - The url to track the job: http://localhost:8080/
2023-03-20 06:18:39,407 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - HadoopJobId: job_local37669034_0001
2023-03-20 06:18:39,407 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Processing aliases grouped,lines,wordcount,words
2023-03-20 06:18:39,407 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - detailed locations: M: lines[1,8],words[-1,-1],wordcount[4,12],grouped[3,10] C:
 wordcount[4,12],grouped[3,10] R: wordcount[4,12]
2023-03-20 06:18:39,416 [Thread-8] INFO  org.apache.hadoop.mapred.LocalJobRunner - OutputCommitter set in config null
2023-03-20 06:18:39,460 [Thread-8] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2023-03-20 06:18:39,460 [Thread-8] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.reduce.markreset.buffer.percent is deprecated. Instead, use mapreduce.reduce.markreset.buffer.
percent
2023-03-20 06:18:39,460 [Thread-8] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
2023-03-20 06:18:39,460 [Thread-8] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2023-03-20 06:18:39,464 [Thread-8] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
```



```
2023-03-20 06:18:39,464 [Thread-8] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - File Output Committer Algorithm version is 1
2023-03-20 06:18:39,464 [Thread-8] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanu
p failures: false
2023-03-20 06:18:39,471 [Thread-8] INFO  org.apache.hadoop.mapred.LocalJobRunner - OutputCommitter is org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.PigOutputCommitter
2023-03-20 06:18:39,555 [Thread-8] INFO  org.apache.hadoop.mapred.LocalJobRunner - Waiting for map tasks
2023-03-20 06:18:39,556 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.LocalJobRunner - Starting task: attempt_local37669034_0001_m_000000_0
2023-03-20 06:18:39,668 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - File Output Committer Algorithm version is 1
2023-03-20 06:18:39,668 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - FileOutputCommitter skip cleanup _temporary folders under output dir
ectory:false, ignore cleanup failures: false
2023-03-20 06:18:39,717 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.Task -  Using ResourceCalculatorProcessTree : [ ]
2023-03-20 06:18:39,726 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.MapTask - Processing split: Number of splits :1
Total Length = 0
Input split[0]:
   Length = 0
  Locations:

-----------------------

2023-03-20 06:18:39,761 [LocalJobRunner Map Task Executor #0] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.PigRecordReader - Current split being processed file:/home/cloudera/
input.csv:0+0
2023-03-20 06:18:39,881 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.MapTask - (EQUATOR) 0 kvi 26214396(104857584)
2023-03-20 06:18:39,881 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.MapTask - mapreduce.task.io.sort.mb: 100
2023-03-20 06:18:39,881 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.MapTask - soft limit at 83886080
2023-03-20 06:18:39,881 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.MapTask - bufstart = 0; bufvoid = 104857600
2023-03-20 06:18:39,881 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.MapTask - kvstart = 26214396; length = 6553600
2023-03-20 06:18:39,887 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.MapTask - Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2023-03-20 06:18:39,919 [LocalJobRunner Map Task Executor #0] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2023-03-20 06:18:39,930 [LocalJobRunner Map Task Executor #0] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.PigGenericMapReduce$Map - Aliases being processed per job phase (Ali
asName[line,offset]): M: lines[1,8],words[-1,-1],wordcount[4,12],grouped[3,10] C: wordcount[4,12],grouped[3,10] R: wordcount[4,12]
2023-03-20 06:18:39,933 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.LocalJobRunner -
2023-03-20 06:18:39,933 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.MapTask - Starting flush of map output
2023-03-20 06:18:39,946 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.Task - Task:attempt_local37669034_0001_m_000000_0 is done. And is in the process of committing
2023-03-20 06:18:39,959 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.LocalJobRunner - map
2023-03-20 06:18:39,959 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.Task - Task 'attempt_local37669034_0001_m_000000_0' done.
2023-03-20 06:18:39,959 [LocalJobRunner Map Task Executor #0] INFO  org.apache.hadoop.mapred.LocalJobRunner - Finishing task: attempt_local37669034_0001_m_000000_0
2023-03-20 06:18:39,960 [Thread-8] INFO  org.apache.hadoop.mapred.LocalJobRunner - map task executor complete.
2023-03-20 06:18:39,962 [Thread-8] INFO  org.apache.hadoop.mapred.LocalJobRunner - Waiting for reduce tasks
2023-03-20 06:18:39,963 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.LocalJobRunner - Starting task: attempt_local37669034_0001_r_000000_0
2023-03-20 06:18:40,004 [pool-3-thread-1] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - File Output Committer Algorithm version is 1
2023-03-20 06:18:40,004 [pool-3-thread-1] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore
 cleanup failures: false
2023-03-20 06:18:40,006 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.Task -  Using ResourceCalculatorProcessTree : [ ]
2023-03-20 06:18:40,012 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.ReduceTask - Using ShuffleConsumerPlugin: org.apache.hadoop.mapreduce.task.reduce.Shuffle@47f03caa
2023-03-20 06:18:40,039 [pool-3-thread-1] INFO  org.apache.hadoop.mapreduce.task.reduce.MergeManagerImpl - MergerManager: memoryLimit=709551680, maxSingleShuffleLimit=177387920, mergeThreshold=4683
04128, ioSortFactor=10, memToMemMergeOutputsThreshold=10
2023-03-20 06:18:40,048 [EventFetcher for fetching Map Completion Events] INFO  org.apache.hadoop.mapreduce.task.reduce.EventFetcher - attempt_local37669034_0001_r_000000_0 Thread started: EventFet
```

```
Applications  Places  System                                    Mon Mar 20, 6:20 AM  cloudera
cloudera@quickstart:~
File  Edit  View  Search  Terminal  Help
cher for fetching Map Completion Events
2023-03-20 06:18:40,102 [localfetcher#1] INFO  org.apache.hadoop.mapreduce.task.reduce.LocalFetcher - localfetcher#1 about to shuffle output of map attempt_local37669034_0001_m_000000_0 decomp: 2 len: 6 to MEMORY
2023-03-20 06:18:40,109 [localfetcher#1] INFO  org.apache.hadoop.mapreduce.task.reduce.InMemoryMapOutput - Read 2 bytes from map-output for attempt_local37669034_0001_m_000000_0
2023-03-20 06:18:40,112 [localfetcher#1] INFO  org.apache.hadoop.mapreduce.task.reduce.MergeManagerImpl - closeInMemoryFile -> map-output of size: 2, inMemoryMapOutputs.size() -> 1, commitMemory -> 0, usedMemory ->2
2023-03-20 06:18:40,113 [Readahead Thread #0] WARN  org.apache.hadoop.io.ReadaheadPool - Failed readahead on ifile
EBADF: Bad file descriptor
        at org.apache.hadoop.io.nativeio.NativeIO$POSIX.posix_fadvise(Native Method)
        at org.apache.hadoop.io.nativeio.NativeIO$POSIX.posixFadviseIfPossible(NativeIO.java:267)
        at org.apache.hadoop.io.nativeio.NativeIO$POSIX$CacheManipulator.posixFadviseIfPossible(NativeIO.java:146)
        at org.apache.hadoop.io.ReadaheadPool$ReadaheadRequestImpl.run(ReadaheadPool.java:206)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
        at java.lang.Thread.run(Thread.java:745)
2023-03-20 06:18:40,116 [EventFetcher for fetching Map Completion Events] INFO  org.apache.hadoop.mapreduce.task.reduce.EventFetcher - EventFetcher is interrupted.. Returning
2023-03-20 06:18:40,117 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.LocalJobRunner - 1 / 1 copied.
2023-03-20 06:18:40,117 [pool-3-thread-1] INFO  org.apache.hadoop.mapreduce.task.reduce.MergeManagerImpl - finalMerge called with 1 in-memory map-outputs and 0 on-disk map-outputs
2023-03-20 06:18:40,125 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.Merger - Merging 1 sorted segments
2023-03-20 06:18:40,125 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.Merger - Down to the last merge-pass, with 0 segments left of total size: 0 bytes
2023-03-20 06:18:40,126 [pool-3-thread-1] INFO  org.apache.hadoop.mapreduce.task.reduce.MergeManagerImpl - Merged 1 segments, 2 bytes to disk to satisfy reduce memory limit
2023-03-20 06:18:40,126 [pool-3-thread-1] INFO  org.apache.hadoop.mapreduce.task.reduce.MergeManagerImpl - Merging 1 files, 6 bytes from disk
2023-03-20 06:18:40,127 [pool-3-thread-1] INFO  org.apache.hadoop.mapreduce.task.reduce.MergeManagerImpl - Merging 0 segments, 0 bytes from memory into reduce
2023-03-20 06:18:40,127 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.Merger - Merging 1 sorted segments
2023-03-20 06:18:40,127 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.Merger - Down to the last merge-pass, with 0 segments left of total size: 0 bytes
2023-03-20 06:18:40,128 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.LocalJobRunner - 1 / 1 copied.
2023-03-20 06:18:40,141 [pool-3-thread-1] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - File Output Committer Algorithm version is 1
2023-03-20 06:18:40,141 [pool-3-thread-1] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2023-03-20 06:18:40,144 [pool-3-thread-1] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.skip.on is deprecated. Instead, use mapreduce.job.skiprecords
2023-03-20 06:18:40,171 [pool-3-thread-1] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2023-03-20 06:18:40,181 [pool-3-thread-1] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.PigMapReduce$Reduce - Aliases being processed per job phase (AliasName[line,offset]): M: lines[1,8],words[-1,-1],wordcount[4,12],grouped[3,10] C: wordcount[4,12],grouped[3,10] R: wordcount[4,12]
2023-03-20 06:18:40,182 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.Task - Task:attempt_local37669034_0001_r_000000_0 is done. And is in the process of committing
2023-03-20 06:18:40,184 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.LocalJobRunner - 1 / 1 copied.
2023-03-20 06:18:40,184 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.Task - Task attempt_local37669034_0001_r_000000_0 is allowed to commit now
2023-03-20 06:18:40,190 [pool-3-thread-1] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - Saved output of task 'attempt_local37669034_0001_r_000000_0' to file:/tmp/temp-893915745/tmp-1577467803/_temporary/0/task_local37669034_0001_r_000000
2023-03-20 06:18:40,191 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.LocalJobRunner - reduce > reduce
2023-03-20 06:18:40,191 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.Task - Task 'attempt_local37669034_0001_r_000000_0' done.
2023-03-20 06:18:40,191 [pool-3-thread-1] INFO  org.apache.hadoop.mapred.LocalJobRunner - Finishing task: attempt_local37669034_0001_r_000000_0
2023-03-20 06:18:40,191 [Thread-8] INFO  org.apache.hadoop.mapred.LocalJobRunner - reduce task executor complete.
2023-03-20 06:18:45,421 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
2023-03-20 06:18:51,425 [main] WARN  org.apache.pig.tools.pigstats.PigStatsUtil - Failed to get RunningJob for job job_local37669034_0001
2023-03-20 06:18:51,435 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2023-03-20 06:18:51,436 [main] INFO  org.apache.pig.tools.pigstats.SimplePigStats - Detected Local mode. Stats reported below may be incomplete
Practical 2 (Map-Redu...   cloudera@quickstart:~
```

```
2023-03-20 06:18:51,446 [main] INFO  org.apache.pig.tools.pigstats.SimplePigStats - Script Statistics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.13.0 0.12.0-cdh5.13.0        cloudera        2023-03-20 06:18:37     2023-03-20 06:18:51     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Alias   Feature Outputs
job_local37669034_0001  grouped,lines,wordcount,words   GROUP_BY,COMBINER       file:/tmp/temp-893915745/tmp-1577467803,

Input(s):
Successfully read records from: "/home/cloudera/input.csv"

Output(s):
Successfully stored records in: "file:/tmp/temp-893915745/tmp-1577467803"

Job DAG:
job_local37669034_0001


2023-03-20 06:18:57,463 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2023-03-20 06:18:57,469 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2023-03-20 06:18:57,469 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2023-03-20 06:18:57,478 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2023-03-20 06:18:57,478 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2023-03-20 06:18:57,494 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2023-03-20 06:18:57,494 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
grunt>
Practical 2 (Map-Redu...   cloudera@quickstart:~
```

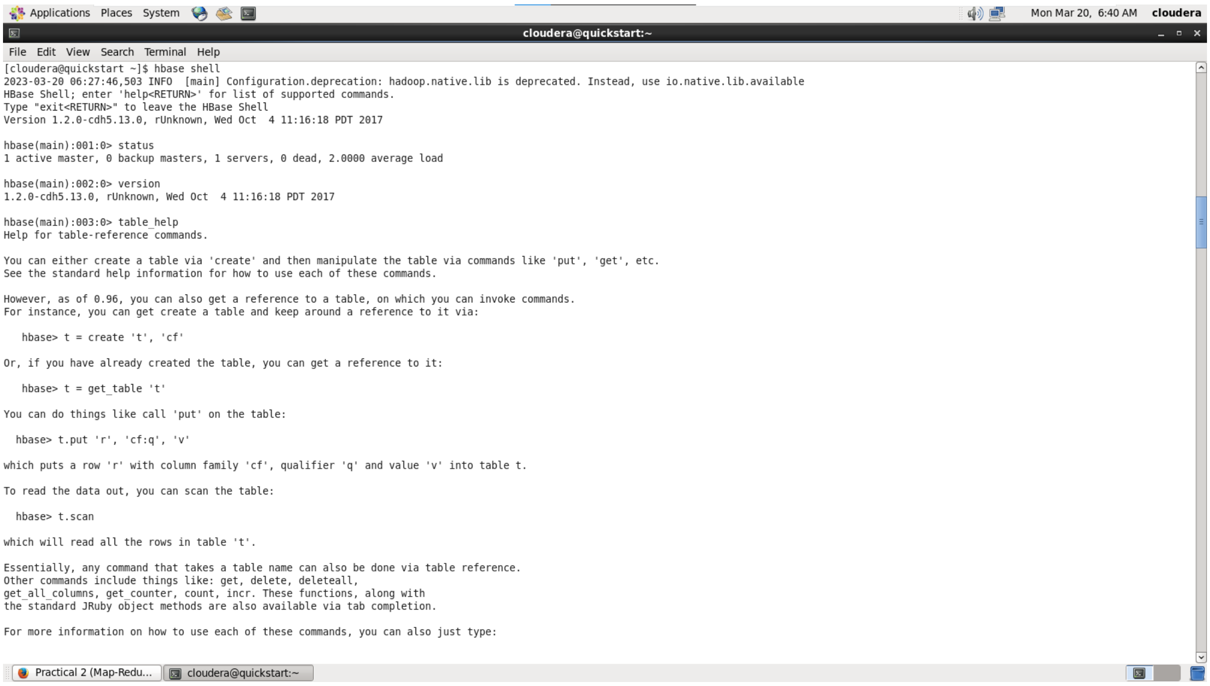
```
(.,1)
(is,2)
(for,1)
(you,1)
(good,1)
(this,1)
(cloudera,2)
(Hello,1)
(user,2)
(making,1)
(program,1)
grunt>
```

Practical 4: Install HBase and Use HBase Data Model to Store and Retrieve Databases

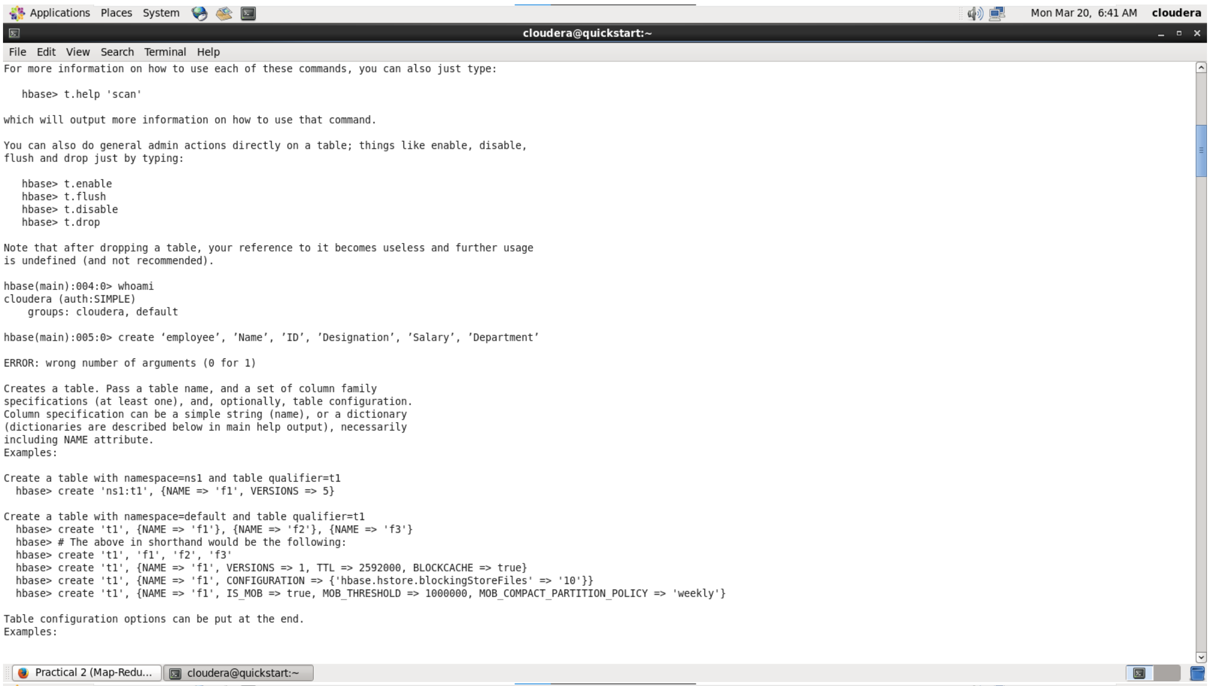
Commands:

```
hbase shell
status
version,
table_help
whoami
create ‘employee’, ’Name’, ’ID’, ’Designation’, ’Salary’, ’Department’
list
disable ‘employee’ (or is_disable ‘employee’)
scan ‘employee’
disable_all ‘e.*’
enable‘employee’ (or scan ‘is_enabled'employee')

//create new table
create‘student’, ‘name’, ‘age’, ‘course’
put ‘student’, ‘sharath’, ‘name:fullname’, ‘sharathkumar’
put ‘student’, ‘sharath’, ‘age:presentage’, ‘24’
put ‘student’, ‘sharath’, ‘course:pursuing’, ‘Hadoop’
put ‘student’, ‘shashank’, ‘name:fullname’, ‘shashank R
put ‘student’, ‘shashank’, ‘age:presentage’, ‘23’
put ‘student’, ‘shashank’, ‘course:pursuing’, ‘Java’

//Get Information
get ‘student’, ‘shashank’
get ‘student’, ‘sharath’
get ‘student’, ‘sharath’, ‘course’
get ‘student’, ‘shashank’, ‘course’
get ‘student’, ‘sharath’, ‘name’


scan ‘student’
Count ‘student’

//Alter
alter ‘student’, NAME=>’name’, VERSIONS=>5
put ‘student’, ‘shashank’, ‘name:fullname’, ‘shashank Rao’
scan ‘student’

//Delete
delete ‘student’, ‘shashank’, ‘name:fullname’
```

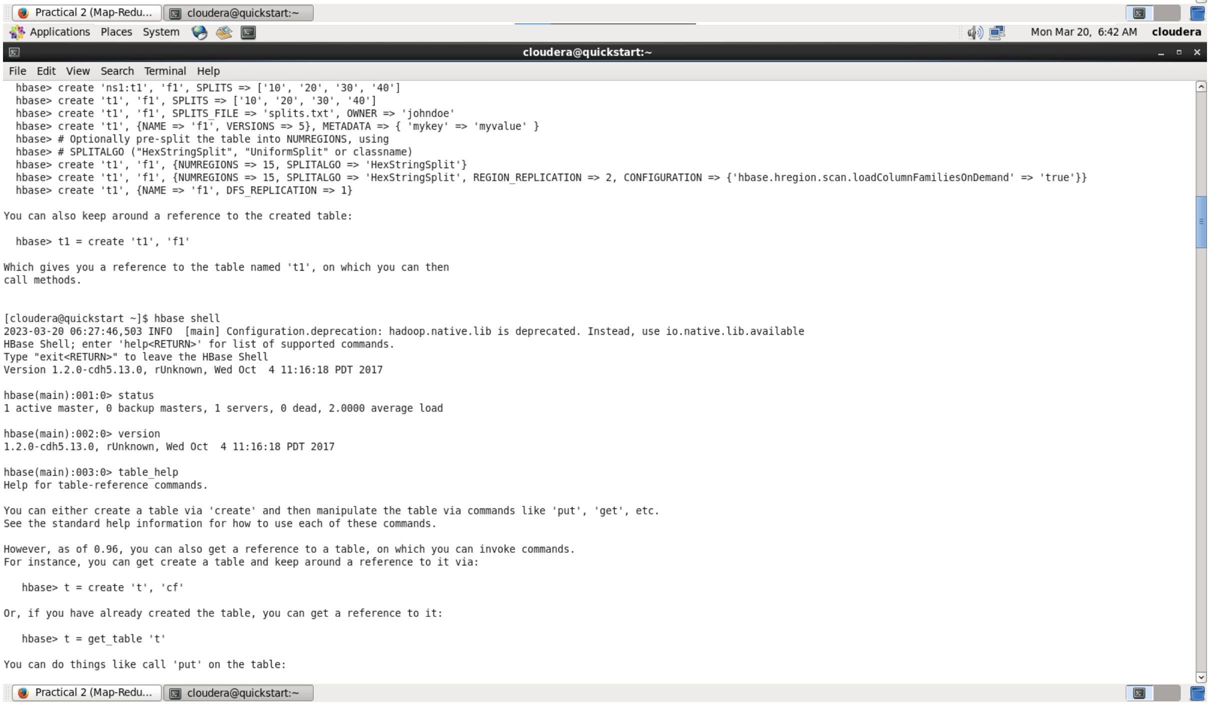
OUTPUT



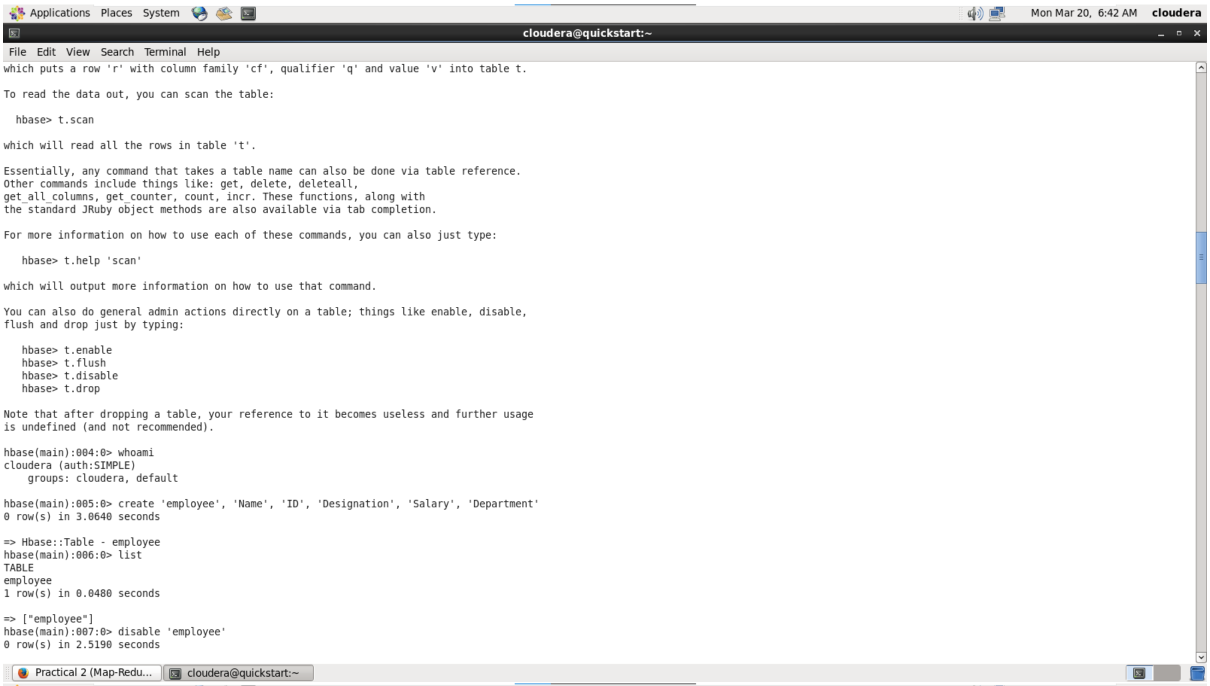
Applications Places System
Mon Mar 20, 6:40 AM cloudera
cloudera@quickstart:~
File Edit View Search Terminal Help
```
[cloudera@quickstart ~]$ hbase shell
2023-03-20 06:27:46,503 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.13.0, rUnknown, Wed Oct  4 11:16:18 PDT 2017

hbase(main):001:0> status
1 active master, 0 backup masters, 1 servers, 0 dead, 2.0000 average load

hbase(main):002:0> version
1.2.0-cdh5.13.0, rUnknown, Wed Oct  4 11:16:18 PDT 2017

hbase(main):003:0> table_help
Help for table-reference commands.

You can either create a table via 'create' and then manipulate the table via commands like 'put', 'get', etc.
See the standard help information for how to use each of these commands.

However, as of 0.96, you can also get a reference to a table, on which you can invoke commands.
For instance, you can get create a table and keep around a reference to it via:

   hbase> t = create 't', 'cf'

Or, if you have already created the table, you can get a reference to it:

   hbase> t = get_table 't'

You can do things like call 'put' on the table:

  hbase> t.put 'r', 'cf:q', 'v'

which puts a row 'r' with column family 'cf', qualifier 'q' and value 'v' into table t.

To read the data out, you can scan the table:

  hbase> t.scan

which will read all the rows in table 't'.

Essentially, any command that takes a table name can also be done via table reference.
Other commands include things like: get, delete, deleteall,
get_all_columns, get_counter, count, incr. These functions, along with
the standard JRuby object methods are also available via tab completion.

For more information on how to use each of these commands, you can also just type:
```
Practical 2 (Map-Redu...
cloudera@quickstart:~



```
For more information on how to use each of these commands, you can also just type:

   hbase> t.help 'scan'

which will output more information on how to use that command.

You can also do general admin actions directly on a table; things like enable, disable,
flush and drop just by typing:

   hbase> t.enable
   hbase> t.flush
   hbase> t.disable
   hbase> t.drop

Note that after dropping a table, your reference to it becomes useless and further usage
is undefined (and not recommended).

hbase(main):004:0> whoami
cloudera (auth:SIMPLE)
    groups: cloudera, default

hbase(main):005:0> create ‘employee’, ’Name’, ’ID’, ’Designation’, ’Salary’, ’Department’

ERROR: wrong number of arguments (0 for 1)

Creates a table. Pass a table name, and a set of column family
specifications (at least one), and, optionally, table configuration.
Column specification can be a simple string (name), or a dictionary
(dictionaries are described below in main help output), necessarily
including NAME attribute.
Examples:

Create a table with namespace=ns1 and table qualifier=t1
  hbase> create 'ns1:t1', {NAME => 'f1', VERSIONS => 5}

Create a table with namespace=default and table qualifier=t1
  hbase> create 't1', {NAME => 'f1'}, {NAME => 'f2'}, {NAME => 'f3'}
  hbase> # The above in shorthand would be the following:
  hbase> create 't1', 'f1', 'f2', 'f3'
  hbase> create 't1', {NAME => 'f1', VERSIONS => 1, TTL => 2592000, BLOCKCACHE => true}
  hbase> create 't1', {NAME => 'f1', CONFIGURATION => {'hbase.hstore.blockingStoreFiles' => '10'}}
  hbase> create 't1', {NAME => 'f1', IS_MOB => true, MOB_THRESHOLD => 1000000, MOB_COMPACT_PARTITION_POLICY => 'weekly'}

Table configuration options can be put at the end.
Examples:
```



```
  hbase> create 'ns1:t1', 'f1', SPLITS => ['10', '20', '30', '40']
  hbase> create 't1', 'f1', SPLITS => ['10', '20', '30', '40']
  hbase> create 't1', 'f1', SPLITS_FILE => 'splits.txt', OWNER => 'johndoe'
  hbase> create 't1', {NAME => 'f1', VERSIONS => 5}, METADATA => { 'mykey' => 'myvalue' }
  hbase> # Optionally pre-split the table into NUMREGIONS, using
  hbase> # SPLITALGO ("HexStringSplit", "UniformSplit" or classname)
  hbase> create 't1', 'f1', {NUMREGIONS => 15, SPLITALGO => 'HexStringSplit'}
  hbase> create 't1', 'f1', {NUMREGIONS => 15, SPLITALGO => 'HexStringSplit', REGION_REPLICATION => 2, CONFIGURATION => {'hbase.hregion.scan.loadColumnFamiliesOnDemand' => 'true'}}
  hbase> create 't1', {NAME => 'f1', DFS_REPLICATION => 1}

You can also keep around a reference to the created table:

  hbase> t1 = create 't1', 'f1'

Which gives you a reference to the table named 't1', on which you can then
call methods.


[cloudera@quickstart ~]$ hbase shell
2023-03-20 06:27:46,503 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.13.0, rUnknown, Wed Oct  4 11:16:18 PDT 2017

hbase(main):001:0> status
1 active master, 0 backup masters, 1 servers, 0 dead, 2.0000 average load

hbase(main):002:0> version
1.2.0-cdh5.13.0, rUnknown, Wed Oct  4 11:16:18 PDT 2017

hbase(main):003:0> table_help
Help for table-reference commands.

You can either create a table via 'create' and then manipulate the table via commands like 'put', 'get', etc.
See the standard help information for how to use each of these commands.

However, as of 0.96, you can also get a reference to a table, on which you can invoke commands.
For instance, you can get create a table and keep around a reference to it via:

   hbase> t = create 't', 'cf'

Or, if you have already created the table, you can get a reference to it:

   hbase> t = get_table 't'

You can do things like call 'put' on the table:
```



```
which puts a row 'r' with column family 'cf', qualifier 'q' and value 'v' into table t.

To read the data out, you can scan the table:

  hbase> t.scan

which will read all the rows in table 't'.

Essentially, any command that takes a table name can also be done via table reference.
Other commands include things like: get, delete, deleteall,
get_all_columns, get_counter, count, incr. These functions, along with
the standard JRuby object methods are also available via tab completion.

For more information on how to use each of these commands, you can also just type:

   hbase> t.help 'scan'

which will output more information on how to use that command.

You can also do general admin actions directly on a table; things like enable, disable,
flush and drop just by typing:

   hbase> t.enable
   hbase> t.flush
   hbase> t.disable
   hbase> t.drop

Note that after dropping a table, your reference to it becomes useless and further usage
is undefined (and not recommended).

hbase(main):004:0> whoami
cloudera (auth:SIMPLE)
    groups: cloudera, default

hbase(main):005:0> create 'employee', 'Name', 'ID', 'Designation', 'Salary', 'Department'
0 row(s) in 3.0640 seconds

=> Hbase::Table - employee
hbase(main):006:0> list
TABLE
employee
1 row(s) in 0.0480 seconds

=> ["employee"]
hbase(main):007:0> disable 'employee'
0 row(s) in 2.5190 seconds
```

```
Scan a table; pass table name and optionally a dictionary of scanner
specifications.  Scanner specifications may include one or more of:
TIMERANGE, FILTER, LIMIT, STARTROW, STOPROW, ROWPREFIXFILTER, TIMESTAMP,
MAXLENGTH or COLUMNS, CACHE or RAW, VERSIONS, ALL_METRICS or METRICS

If no columns are specified, all columns will be scanned.
To scan all members of a column family, leave the qualifier empty as in
'col_family'.

The filter can be specified in two ways:
1. Using a filterString - more information on this is available in the
Filter Language document attached to the HBASE-4176 JIRA
2. Using the entire package name of the filter.

If you wish to see metrics regarding the execution of the scan, the
ALL_METRICS boolean should be set to true. Alternatively, if you would
prefer to see only a subset of the metrics, the METRICS array can be
defined to include the names of only the metrics you care about.

Some examples:

  hbase> scan 'hbase:meta'
  hbase> scan 'hbase:meta', {COLUMNS => 'info:regioninfo'}
  hbase> scan 'ns1:t1', {COLUMNS => ['c1', 'c2'], LIMIT => 10, STARTROW => 'xyz'}
  hbase> scan 't1', {COLUMNS => ['c1', 'c2'], LIMIT => 10, STARTROW => 'xyz'}
  hbase> scan 't1', {COLUMNS => 'c1', TIMERANGE => [1303668804, 1303668904]}
  hbase> scan 't1', {REVERSED => true}
  hbase> scan 't1', {ALL_METRICS => true}
  hbase> scan 't1', {METRICS => ['RPC_RETRIES', 'ROWS_FILTERED']}
  hbase> scan 't1', {ROWPREFIXFILTER => 'row2', FILTER => "
    (QualifierFilter (>=, 'binary:xyz')) AND (TimestampsFilter ( 123, 456))"}
  hbase> scan 't1', {FILTER =>
    org.apache.hadoop.hbase.filter.ColumnPaginationFilter.new(1, 0)}
  hbase> scan 't1', {CONSISTENCY => 'TIMELINE'}
For setting the Operation Attributes
  hbase> scan 't1', { COLUMNS => ['c1', 'c2'], ATTRIBUTES => {'mykey' => 'myvalue'}}
  hbase> scan 't1', { COLUMNS => ['c1', 'c2'], AUTHORIZATIONS => ['PRIVATE','SECRET']}
For experts, there is an additional option -- CACHE_BLOCKS -- which
switches block caching for the scanner on (true) or off (false).  By
default it is enabled.  Examples:

  hbase> scan 't1', {COLUMNS => ['c1', 'c2'], CACHE_BLOCKS => false}

Also for experts, there is an advanced option -- RAW -- which instructs the
scanner to return all cells (including delete markers and uncollected deleted
```

```
Disabled by default.  Example:

  hbase> scan 't1', {RAW => true, VERSIONS => 10}

Besides the default 'toStringBinary' format, 'scan' supports custom formatting
by column.  A user can define a FORMATTER by adding it to the column name in
the scan specification.  The FORMATTER can be stipulated:

 1. either as a org.apache.hadoop.hbase.util.Bytes method name (e.g, toInt, toString)
 2. or as a custom class followed by method name: e.g. 'c(MyFormatterClass).format'.

Example formatting cf:qualifier1 and cf:qualifier2 both as Integers:
  hbase> scan 't1', {COLUMNS => ['cf:qualifier1:toInt',
    'cf:qualifier2:c(org.apache.hadoop.hbase.util.Bytes).toInt'] }

Note that you can specify a FORMATTER by column only (cf:qualifier).  You cannot
specify a FORMATTER for all columns of a column family.

Scan can also be used directly from a table, by first getting a reference to a
table, like such:

  hbase> t = get_table 't'
  hbase> t.scan

Note in the above situation, you can still provide all the filtering, columns,
options, etc as described above.



hbase(main):009:0> enable 'employee'
0 row(s) in 1.3080 seconds

hbase(main):010:0> create 'student', 'name', 'age', 'course'
0 row(s) in 1.2460 seconds

=> Hbase::Table - student
hbase(main):011:0> put 'student', 'sharath', 'name:fullname', 'sharathkumar'
0 row(s) in 0.2000 seconds

hbase(main):012:0> put 'student', 'sharath', 'age:presentage', '24'
0 row(s) in 0.0090 seconds

hbase(main):013:0> put 'student', 'sharath', 'course:pursuing', 'Hadoop'
0 row(s) in 0.0060 seconds

hbase(main):014:0> put 'student', 'shashank', 'name:fullname', 'shashank R'
```

```
hbase(main):013:0> put 'student', 'sharath', 'course:pursuing', 'Hadoop'
0 row(s) in 0.0060 seconds

hbase(main):014:0> put 'student', 'shashank', 'name:fullname', 'shashank R'
0 row(s) in 0.0050 seconds

hbase(main):015:0> put 'student', 'shashank', 'age:presentage', '23'
0 row(s) in 0.0050 seconds

hbase(main):016:0> put 'student', 'shashank', 'course:pursuing', 'Java'
0 row(s) in 0.0050 seconds

hbase(main):017:0> get 'student', 'shashank'
COLUMN                                      CELL
 age:presentage                             timestamp=1679319172900, value=23
 course:pursuing                            timestamp=1679319186342, value=Java
 name:fullname                              timestamp=1679319163299, value=shashank R
3 row(s) in 0.0240 seconds

hbase(main):018:0> get 'student', 'sharath'
COLUMN                                      CELL
 age:presentage                             timestamp=1679319092488, value=24
 course:pursuing                            timestamp=1679319110610, value=Hadoop
 name:fullname                              timestamp=1679319025975, value=sharathkumar
3 row(s) in 0.0080 seconds

hbase(main):019:0> get 'student', 'sharath',  'course'
COLUMN                                      CELL
 course:pursuing                            timestamp=1679319110610, value=Hadoop
1 row(s) in 0.0160 seconds

hbase(main):020:0> get 'student', 'shashank', 'course'
COLUMN                                      CELL
 course:pursuing                            timestamp=1679319186342, value=Java
1 row(s) in 0.0090 seconds

hbase(main):021:0> get 'student', 'sharath',  'name'
COLUMN                                      CELL
 name:fullname                              timestamp=1679319025975, value=sharathkumar
1 row(s) in 0.0050 seconds

hbase(main):022:0> scan 'student'
ROW                                         COLUMN+CELL
 sharath                                    column=age:presentage, timestamp=1679319092488, value=24
 sharath                                    column=course:pursuing, timestamp=1679319110610, value=Hadoop
 sharath                                    column=name:fullname, timestamp=1679319025975, value=sharathkumar
```

```
[cloudera@quickstart ~]$ hbase shell
2023-03-20 06:36:12,086 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.13.0, rUnknown, Wed Oct  4 11:16:18 PDT 2017

hbase(main):001:0> alter 'student', Name=>'name', VERSIONS=>5
NameError: uninitialized constant Name

hbase(main):002:0> put 'student', 'shashank', 'name:fullname', 'shashank Rao'
0 row(s) in 0.7650 seconds

hbase(main):003:0> scan 'student'
ROW                                         COLUMN+CELL
 sharath                                    column=age:presentage, timestamp=1679319092488, value=24
 sharath                                    column=course:pursuing, timestamp=1679319110610, value=Hadoop
 sharath                                    column=name:fullname, timestamp=1679319025975, value=sharathkumar
 shashank                                   column=age:presentage, timestamp=1679319172900, value=23
 shashank                                   column=course:pursuing, timestamp=1679319186342, value=Java
 shashank                                   column=name:fullname, timestamp=1679319443018, value=shashank Rao
2 row(s) in 0.0930 seconds

hbase(main):004:0> delete 'student' ,'shashank', 'name:fullname'
0 row(s) in 0.0990 seconds

hbase(main):005:0>
```

Practical 5: Install Hive and Use Hive to Create and Store Structured Databases

Commands:

```
cat > /home/cloudera/employee.txt
cat  /home/cloudera/employee.txt
sudo -u hdfs hadoop fs -put /home/cloudera/employee.txt /inputdirectroy
hdfs dfs -ls /inputdirectory
hadoop fs -cat  /inputdirectory/employee.txt
hive
show databases;
create database organization;
show databases;
use organization;
show tables;
hive> create table employee(
    > id int,
    > name string,
    > city string,
    > department string,
    > salary int,
    > domain string)
    > row format delimited
    > fields terminated by '~';

show tables;
select * from employee;
show tables;
load data inpath '/inputdirectory/employee.txt' overwrite into table employee;
show tables;
select * from employee;
```

OUTPUT

```
[cloudera@quickstart ~]$ cat > /home/cloudera/employee.txt
1~Sachine~Pune~Product Engineering~100000~Big Data
2~Gaurav~Banglore~Sales~90000~CRM
3~Manish~Chennai~Recruiter~125000~HR
4~Bhushan~Hyderabad~Developer~50000~BFSI^C
[cloudera@quickstart ~]$ cat  /home/cloudera/employee.txt
1~Sachine~Pune~Product Engineering~100000~Big Data
2~Gaurav~Banglore~Sales~90000~CRM
3~Manish~Chennai~Recruiter~125000~HR
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -put /home/cloudear/employee.txt /inputdirectory
put: `/home/cloudear/employee.txt': No such file or directory
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -put /home/cloudera/employee.txt /inputdirectory
[cloudera@quickstart ~]$ hdfs dfs -ls /inputdirectory
Found 2 items
-rw-r--r--   1 hdfs supergroup        122 2023-03-20 06:46 /inputdirectory/employee.txt
-rw-r--r--   1 hdfs supergroup          0 2023-03-20 05:53 /inputdirectory/processfile.txt
[cloudera@quickstart ~]$ hadoop fs -cat /inputdirectory/employee.txt
1~Sachine~Pune~Product Engineering~100000~Big Data
2~Gaurav~Banglore~Sales~90000~CRM
3~Manish~Chennai~Recruiter~125000~HR
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> show databases;
OK
default
Time taken: 0.778 seconds, Fetched: 1 row(s)
hive> create database organization;
OK
Time taken: 4.94 seconds
hive> show databases;
OK
default
organization
Time taken: 0.034 seconds, Fetched: 2 row(s)
hive> use organization;
OK
Time taken: 0.111 seconds
hive> show tables;
OK
Time taken: 0.048 seconds
hive> create table employee(
    > id int,
    > name string,
    > city string,
```

```
Time taken: 0.034 seconds, Fetched: 2 row(s)
hive> use organization;
OK
Time taken: 0.111 seconds
hive> show tables;
OK
Time taken: 0.048 seconds
hive> create table employee(
    > id int,
    > name string,
    > city string,
    > department string,
    > salary int,
    > domain string)
    > row format delimited
    > fields terminated by '~';
OK
Time taken: 0.425 seconds
hive> show tables;
OK
employee
Time taken: 0.017 seconds, Fetched: 1 row(s)
hive> select * from employee;
OK
Time taken: 0.703 seconds
hive> show tables;
OK
employee
Time taken: 0.025 seconds, Fetched: 1 row(s)
hive> load data inpath '/inputdirectory/employee.txt' overwrite into table employee;
Loading data to table organization.employee
chmod: changing permissions of 'hdfs://quickstart.cloudera:8020/user/hive/warehouse/organization.db/employee/employee.txt': Permission denied. user=cloudera is not the owner of inode=employee.txt
Table organization.employee stats: [numFiles=1, numRows=0, totalSize=122, rawDataSize=0]
OK
Time taken: 0.593 seconds
hive> show tables;
OK
employee
Time taken: 0.034 seconds, Fetched: 1 row(s)
hive> select * from employee;
OK
1       Sachine Pune    Product Engineering     100000  Big Data
2       Gaurav  Banglore        Sales   90000   CRM
3       Manish  Chennai Recruiter       125000  HR
Time taken: 0.076 seconds, Fetched: 3 row(s)
hive>
```

Practical 6: Construct Different Types of K-Shingles for Given Document

Code:

```
# Install necessary packages
install.packages("tm")
require("tm")
install.packages("devtools")

# Define function to read an integer and create shingles from a file
readinteger <- function() {
  n <- readline(prompt = "Enter value of k-1: ")  # Prompt user for input
  k <- as.integer(n)  # Convert input to integer
  u1 <- readLines("C:/Users/asif0/Documents/File1.txt")  # Read in file
  Shingle <- 0  # Initialize variable for storing shingles
  i <- 0  # Initialize loop counter
  while (i < nchar(u1) - k + 1) {  # Loop through file, creating shingles
    Shingle[i] <- substr(u1, start = i, stop = i + k)  # Extract shingle
from file
    print(Shingle[i])  # Print shingle to console
    i <- i + 1  # Increment loop counter
  }
}

# Call readinteger function if running interactively
if (interactive()) {
  readinteger()
}
```

OUTPUT

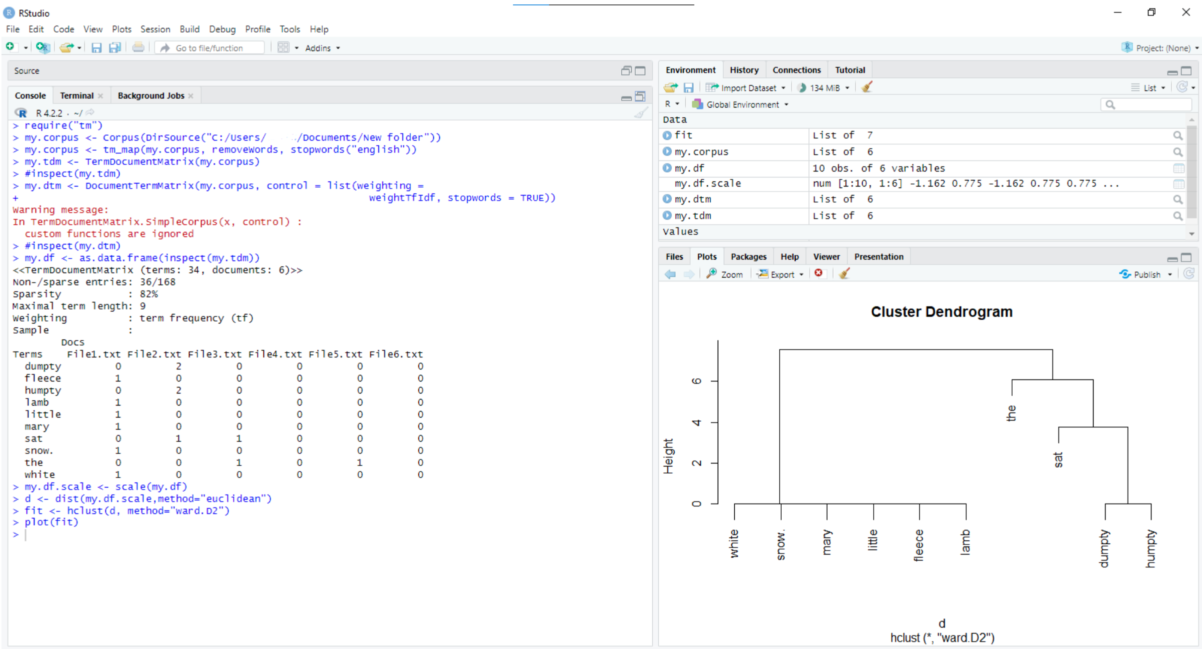
```
Enter value of k-1: 4
character(0)
[1] "This "
[1] "his i"
[1] "is is"
[1] "s is "
[1] " is i"
[1] "is is"
[1] "s is "
[1] " is a"
[1] "is a "
[1] "s a  "
[1] " a  a"
[1] "a  a "
[1] "  a t"
[1] " a te"
[1] "a tes"
[1] " test"
[1] "test "
[1] "est o"
[1] "st of"
[1] "t of "
[1] " of o"
[1] "of of"
[1] "f of "
[1] " of s"
[1] "of sh"
[1] "f shi"
[1] " shin"
[1] "shing"
[1] "hingl"
[1] "ingle"
[1] "ngle "
[1] "gle s"
[1] "le sh"
[1] "e shi"
[1] " shin"
[1] "shing"
[1] "hingl"
[1] "ingle"
[1] "ngles"
```

Practical 7: Measuring Similarity Among Documents and Detecting Passages Which Have Been Reused

Codes:

```
# Install necessary packages
install.packages("tm")
require("tm")
install.packages("ggplot2")
install.packages("textreuse")
install.packages("devtools")

# Load in corpus and preprocess text
my.corpus <- Corpus(DirSource("C:/Users/asif0/Documents/New folder"))  # Load
in corpus from directory
my.corpus <- tm_map(my.corpus, removeWords, stopwords("english"))  # Remove
stop words from corpus

# Create term-document matrix
my.tdm <- TermDocumentMatrix(my.corpus)  # Create term-document matrix from
corpus
#inspect(my.tdm)  # Inspect term-document matrix (optional)

# Create document-term matrix
my.dtm <- DocumentTermMatrix(my.corpus, control = list(weighting =
weightTfIdf, stopwords = TRUE))  # Create document-term matrix from corpus,
using TF-IDF weighting and removing stop words
#inspect(my.dtm)  # Inspect document-term matrix (optional)

# Convert document-term matrix to data frame and scale data
my.df <- as.data.frame(inspect(my.tdm))  # Convert document-term matrix to
data frame
my.df.scale <- scale(my.df)  # Scale data using z-score normalization

# Perform hierarchical clustering and plot dendrogram
d <- dist(my.df.scale, method = "euclidean")  # Calculate distance matrix
using Euclidean distance
fit <- hclust(d, method = "ward")  # Perform hierarchical clustering using
Ward's method
plot(fit)  # Plot dendrogram
```

OUTPUT:



RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console Terminal Background Jobs
R 4.2.2 · ~/
> require("tm")
> my.corpus <- Corpus(DirSource("C:/Users/ /Documents/New folder"))
> my.corpus <- tm_map(my.corpus, removeWords, stopwords("english"))
> my.tdm <- TermDocumentMatrix(my.corpus)
> #inspect(my.tdm)
> my.dtm <- DocumentTermMatrix(my.corpus, control = list(weighting =
+ weightTfIdf, stopwords = TRUE))
Warning message:
In TermDocumentMatrix.SimpleCorpus(x, control) :
custom functions are ignored
> #inspect(my.dtm)
> my.df <- as.data.frame(inspect(my.tdm))
<<TermDocumentMatrix (terms: 34, documents: 6)>>
Non-/sparse entries: 36/168
Sparsity : 82%
Maximal term length: 9
Weighting : term frequency (tf)
Sample :
Docs
Terms File1.txt File2.txt File3.txt File4.txt File5.txt File6.txt
dumpty 0 2 0 0 0 0
fleece 1 0 0 0 0 0
humpty 0 2 0 0 0 0
lamb 1 0 0 0 0 0
little 1 0 0 0 0 0
mary 1 0 0 0 0 0
sat 0 1 1 0 0 0
snow. 1 0 0 0 0 0
the 0 0 1 0 1 0
white 1 0 0 0 0 0
> my.df.scale <- scale(my.df)
> d <- dist(my.df.scale,method="euclidean")
> fit <- hclust(d, method="ward.D2")
> plot(fit)
Environment History Connections Tutorial
Import Dataset
Global Environment
Data
fit List of 7
my.corpus List of 6
my.df 10 obs. of 6 variables
my.df.scale num [1:10, 1:6] -1.162 0.775 -1.162 0.775 0.775 ...
my.dtm List of 6
my.tdm List of 6
Values
Files Plots Packages Help Viewer Presentation
Zoom Export Publish
Cluster Dendrogram
Height
0 2 4 6
white snow. mary little fleece lamb the sat dumpty humpty
d
hclust (*, "ward.D2")

Practical 8: Compute n-moment

Codes:

```
import java.io.*;
import java.util.*;

public class n_moment {
    public static void main(String args[]) {
        int n = 15; // Total number of elements in the stream
        String stream[] = {"a", "b", "c", "b", "d", "a", "c", "d", "a", "b",
"d", "c", "a", "a", "b"};
        int zero_moment = 0, first_moment = 0, second_moment = 0, count = 1,
flag = 0;
        ArrayList<Integer> arrlist = new ArrayList(); // Creating a new
ArrayList

        System.out.println("Arraylist elements are::");
        for (int i = 0; i < 15; i++) {
            System.out.println(stream[i] + " "); // Printing the elements of
the stream
        }

        Arrays.sort(stream); // Sorting the elements of the stream

        for (int i = 1; i < n; i++) {
            if (stream[i] == stream[i - 1]) { // If current element is same as
previous element
                count++; // Increment the count
            } else {
                // System.out.println("Hello"+i);
                arrlist.add(count); // Add the count to the ArrayList
                count = 1; // Reset the count
            }
        }
        arrlist.add(count); // Add the last count to the ArrayList

        zero_moment = arrlist.size(); // Zeroth moment is the size of the
ArrayList
        System.out.println("\n\n\nValue of Zeroth moment for given stream::" +
zero_moment);

        for (int i = 0; i < arrlist.size(); i++) {
            first_moment += arrlist.get(i); // Summing up all the elements in
the ArrayList
        }
        System.out.println("\n\nValue of First moment for given stream::" +
first_moment);
```

```
        for (int i = 0; i < arrlist.size(); i++) {
            int j = arrlist.get(i);
            second_moment += (j * j); // Computing the second moment by
summing up the squares of all elements in the ArrayList
        }
        System.out.println("\n\nValue of Second moment for given stream::" +
second_moment);
    }
}
```

OUTPUT

```
Arraylist elements are::a
b
c b
d
a
c d
a
b
d c
a
a b
Value of Zeroth moment for given stream::4
Value of First moment for given stream::15


Value of Second moment for given stream::59
```

Practical 9: Alon-Matias-Szegedy Algorithm

Codes:

```
import java.io.*;
import java.util.*;

class AMSA {
  public static int findCharCount(String stream, char XE, int random, int n) {
    int countoccurance = 0;
    for (int i = random; i < n; i++) {
      if (stream.charAt(i) == XE) {
        countoccurance++;
      }
    }
    return countoccurance;
  }

  public static int estimateValue(int XV1, int n) {
    int ExpValue;
    ExpValue = n * (2 * XV1 - 1);
    return ExpValue;
  }

  public static void main(String args[]) {
    int n = 15;
    String stream = "abcbdacdabdcaab";
    int random1 = 3, random2 = 8, random3 = 13;
    char XE1, XE2, XE3;
    int XV1, XV2, XV3;
    int ExpValuXE1, ExpValuXE2, ExpValuXE3;
    int apprSecondMomentValue;

    // Select three random characters from the stream
    XE1 = stream.charAt(random1 - 1);
    XE2 = stream.charAt(random2 - 1);
    XE3 = stream.charAt(random3 - 1);

    // Count the number of occurrences of each character in the stream
    XV1 = findCharCount(stream, XE1, random1 - 1, n);
    XV2 = findCharCount(stream, XE2, random2 - 1, n);
    XV3 = findCharCount(stream, XE3, random3 - 1, n);

    // Print the counts of the selected characters
    System.out.println(XE1 + "=" + XV1 + " " + XE2 + "=" + XV2 + " " + XE3 +
"=" + XV3);

    // Estimate the expected value for each selected character
    ExpValuXE1 = estimateValue(XV1, n);
```

```java
    ExpValuXE2 = estimateValue(XV2, n);
    ExpValuXE3 = estimateValue(XV3, n);

    // Print the expected values for each selected character
    System.out.println("Expected value for" + XE1 + " is::" + ExpValuXE1);
    System.out.println("Expected value for" + XE2 + " is::" + ExpValuXE2);
    System.out.println("Expected value for" + XE3 + " is::" + ExpValuXE3);

    // Compute the approximate second moment value using Alon-Matias-Szegedy
algorithm
    apprSecondMomentValue = (ExpValuXE1 + ExpValuXE2 + ExpValuXE3) / 3;
    System.out.println("approximate second moment value using alon-matis-
szegedy is::" + apprSecondMomentValue);
  }
}
```

OUTPUT

```
c=3 d=2 a=2
Expected value forc is::75
Expected value ford is::45
Expected value fora is::45
approximate second moment value using alon-matis-szegedy is::55
```