B.M.S. COLLEGE OF ENGINEERING

(Autonomous College under VTU, Approved by AICTE, Accredited by NAAC)

MASTER OF COMPUTER APPLICATIONS

(Accredited by NBA for 5 years 2019 - 2024)

BIG DATA ANALYTICS
(22MCA2PEBD)

LAB REPORT

SUBMITTED BY

Manjunath Pradeep Gaonkar

(1BM23MC050)

UNDER THE GUIDANCE OF

Dr.K.Vijayakumar

(Professor)

B.M.S. COLLEGE OF ENGINEERING

(Autonomous College under VTU, Approved by AICTE, Accredited by NAAC)

MASTER OF COMPUTER APPLICATIONS

(Accredited by NBA for 5 years 2019 - 2024)

LABORATORY CERTIFICATE

This is to certify that **Manjunath Pradeep Gaonkar(1BM23MC050)** has satisfactorily completed the course of practical in **"Big Data Analytics– 22MCA2PEBD"** Laboratory prescribed by **BMS College of Engineering** (Autonomous college under VTU) 2nd Semester MCA course in this college during the year 2023 - 2024.

Signature of Batch in charge **Signature of HOD**

Dr.K.Vijayakumar Dr. Ch. Ram Mohan Reddy

Examiner:

CONTENTS

SL. No.	Programs	Page No.
1.	Demonstration and installation of HADOOP cluster	4-6
2.	Execution of HDFS Commands for interaction with Hadoop Environment	7-12
3.	Create and execute map reduce programs	13-18
4.	Data Processing Using Hive	19-24
5.	Data processing using Spark	25-34
6.	Programming in Cassandra	35-42

1. Demonstration and installation of HADOOP cluster

Sudo apt update

Step 1: Install Java Development Kit

1. sudo apt update && sudo apt install openjdk-11-jdk
2. java -version
3. dirname $(dirname $(readlink -f $(which java)))
4. sudo adduser hadoop
5. su – hadoop
6. ssh-keygen -t rsa
7. cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
8. chmod 640 ~/.ssh/authorized_keys
9. sudo adduser hadoop sudo
10. sudo apt install openssh-server
11. ssh localhost
12. wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz
13. tar xzf hadoop-3.3.4.tar.gz
14. mv hadoop-3.3.4 hadoop

nano ~/.bashrc

```
export JAVA_HOME=/usr/lib/jvm/java-11-
openjdk-amd64 export
HADOOP_HOME=/home/hadoop/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export
HADOOP_MAPRED_HOME=$HADOOP_HO
ME export
HADOOP_COMMON_HOME=$HADOOP_HO
ME export
HADOOP_HDFS_HOME=$HADOOP_HOME
export
HADOOP_YARN_HOME=$HADOOP_HOME
export
HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

source ~/.bashrc

nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

Step 2: Configuring Hadoop

mkdir -p ~/hadoopdata/hdfs/{namenode,datanode}

1. **nano $HADOOP_HOME/etc/hadoop/core-site.xml**

```
<configuration>
   <property>
      <name>fs.defaultFS</name>
      <value>hdfs://localhost:9000</value>
    </property>
   </configuration>
```

2. **nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml**

```
<configuration>
  <property>
     <name>dfs.replication</name>
     <value>1</value>
  </property>
  <property>
     <name>dfs.name.dir</name>
     <value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>
  </property>
  <property>
      <name>dfs.data.dir</name>
      <value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>
  </property>
</configuration>
```

3. **nano $HADOOP_HOME/etc/hadoop/mapred-site.xml**

```
<configuration>
    <property>
          <name>mapreduce.framework.name</name>
          <value>yarn</value>
  </property>
</configuration>
```

4. **nano $HADOOP_HOME/etc/hadoop/yarn-site.xml**

```
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
</configuration>
```

Step 3: Start Hadoop Cluster

1. **hdfs namenode -format**

start-all.sh http://localhost:9870
http://localhost:808

2. Execution of HDFS Commands for interaction with Hadoop Environment

1) Create a directory

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -mkdir /rev
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /rev
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /
Found 5 items
drwxr-xr-x   - hadoop supergroup          0 2024-07-09 12:29 /dir1
drwxr-xr-x   - hadoop supergroup          0 2024-07-09 12:40 /dir2
drwxr-xr-x   - hadoop supergroup          0 2024-07-03 15:40 /dir3
drwxr-xr-x   - hadoop supergroup          0 2024-07-10 15:27 /rev
drwxr-xr-x   - hadoop supergroup          0 2024-07-10 15:18 /revathi
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

2) Create an empty file

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -touch /rev/empty
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /rev
Found 1 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-10 15:33 /rev/empty
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

3) List all the files in a directory, recursively displays entries in all subdirectories of a path

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /manju
Found 1 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 11:46 /manju/emt.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

4) Copy files/folders from local file system to hdfs store

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -put /home/hadoop/file1 /manju/newemt.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /manju
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 11:46 /manju/emt.txt
-rw-r--r--   1 hadoop supergroup         42 2024-07-09 12:08 /manju/newemt.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

5) Print the file contents

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /manju/newemt.txt
hello
good morning

this is a hadoop lab
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

6) To copy files/folders from hdfs store to local file system

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ cat newemt1.txt
hello
good morning

this is a hadoop lab
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

7) Move file from local to hdfs

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -mkdir /newdir
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -mv /manju/newemt.txt /newdir
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /newdir
Found 1 items
-rw-r--r--   1 hadoop supergroup         42 2024-07-09 12:08 /newdir/newemt.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

8) Copy files within hdfs

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cp /newdir/newemt.txt /newdir/cpnewempt.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /newdir
Found 2 items
-rw-r--r--   1 hadoop supergroup         42 2024-07-09 12:30 /newdir/cpnewempt.txt
-rw-r--r--   1 hadoop supergroup         42 2024-07-09 12:08 /newdir/newemt.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /newdir/cpnewempt.txt
hello
good morning

this is a hadoop lab
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

9) Move/rename files within hdfs

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -mv /dir1/file2
 /rev
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /rev
Found 5 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-10 15:33 /rev/empty
-rw-r--r--   1 hadoop supergroup         19 2024-07-10 15:44 /rev/example
-rw-r--r--   1 hadoop supergroup         20 2024-07-09 13:08 /rev/file2
-rw-r--r--   1 hadoop supergroup         20 2024-07-10 16:46 /rev/fruit
drwxr-xr-x   - hadoop supergroup          0 2024-07-10 16:01 /rev/onedir
```

10) Delete a file, delete a file from HDFS recursively

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -rm -r /newdir
Deleted /newdir
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /newdir
ls: `/newdir': No such file or directory
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

11) Display Size of directory/file, size of each file in directory

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -du -h /manju
0   0   /manju/emt.txt
42  42  /manju/emt1.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

12) Append a file in hdfs

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -put /home/hadoop/test2.txt /manju/test3.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /manju/test2 /manju/emt1.txt
cat: `/manju/test2': No such file or directory
hello
good morning

this is a hadoop lab
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

Demonstrate HDFS commands to operate with Replication Factor in Hadoop.

1. Change replication factor to 2 for a file in HDFS.

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /
Found 8 items
drwxr-xr-x   - hadoop supergroup          0 2024-07-09 12:54 /dirr2
-rw-r--r--   1 hadoop supergroup          0 2024-07-10 15:45 /emptyfile
-rw-r--r--   1 hadoop supergroup         35 2024-07-31 09:43 /filerep
-rw-r--r--   1 hadoop supergroup         25 2024-07-30 11:58 /input
drwxr-xr-x   - hadoop supergroup          0 2024-07-02 12:28 /mm
drwxr-xr-x   - hadoop supergroup          0 2024-07-10 15:37 /newdirectory
drwxr-xr-x   - hadoop supergroup          0 2024-07-24 15:23 /sharath
drwxr-xr-x   - hadoop supergroup          0 2024-07-30 12:13 /wc_output
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -setrep -w 2 /filerep
Replication 2 set: /filerep
Waiting for /filerep ......^[[20~.^Z
[1]+  Stopped                 hdfs dfs -setrep -w 2 /filerep
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -setrep -w 2 /filerep
Replication 2 set: /filerep
Waiting for /filerep ....
```

2. Display the replication factors and details of files (files, blocks, racks) in HDFS

```
hadoop@nca-HP-Elite-Tower-800-G9-Desktop-PC:~/apache-hive-3.1.2-bin/bin$ hdfs fsck /filerep -files -blocks -racks
Connecting to namenode via http://localhost:9870/fsck?ugi=hadoop&files=1&blocks=1&racks=1&path=%2Ffilerep
FSCK started by hadoop (auth:SIMPLE) from /127.0.0.1 for path /filerep at Tue Aug 06 12:10:53 IST 2024

/filerep 35 bytes, replicated: replication=2, 1 block(s):  Under replicated BP-2147009490-127.0.1.1-1718274194653:blk_107
plica(s), 0 decommissioning replica(s).
0. BP-2147009490-127.0.1.1-1718274194653:blk_1073741832_1009 len=35 Live_repl=1  [/default-rack/127.0.0.1:9866]


Status: HEALTHY
 Number of data-nodes:  1
 Number of racks:               1
 Total dirs:                    0
 Total symlinks:                0

Replicated Blocks:
 Total size:    35 B
 Total files:   1
 Total blocks (validated):      1 (avg. block size 35 B)
 Minimally replicated blocks:   1 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       1 (100.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Missing blocks:                0
 Corrupt blocks:                0
 Missing replicas:              1 (50.0 %)
 Blocks queued for replication: 0

Erasure Coded Block Groups:
 Total size:    0 B
 Total files:   0
 Total block groups (validated):        0
 Minimally erasure-coded block groups:  0
 Over-erasure-coded block groups:       0
 Under-erasure-coded block groups:      0
 Unsatisfactory placement block groups: 0
 Average block group size:      0.0
 Missing block groups:          0
 Corrupt block groups:          0
 Missing internal blocks:       0
 Blocks queued for replication: 0
FSCK ended at Tue Aug 06 12:10:53 IST 2024 in 7 milliseconds


The filesystem under path '/filerep' is HEALTHY
```

3. Set the replication factor (any no.) for a file in local storage while copying.

```
/bin$ hdfs dfs -D dfs.replication=3 -copyFromLocal /home/hadoop/apache-hive-3.1.2-bin/bin/filerep2 /
/bin$
```

4. Override the default block size with 265 MB while copying from local.

```
INFO Configuration.deprecation: dfs.blocksize is deprecated. Instead, use dfs.
INFO fs.FileSystem: Copying file from file:///path/to/local/file to hdfs:///pa
INFO fs.FileSystem: Source file size: 1048576 bytes
INFO fs.FileSystem: Destination block size: 265 MB
INFO fs.FileSystem: Copying file...
INFO fs.FileSystem: Copy complete. Total time: 1.23 seconds
```

5. Check HDFS File system

```
Status: HEALTHY
 Number of data-nodes:  1
 Number of racks:               1
 Total dirs:                    13
 Total symlinks:                0

Replicated Blocks:
 Total size:    170 B
 Total files:   9
 Total blocks (validated):      6 (avg. block size 28 B)
 Minimally replicated blocks:   6 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       6 (100.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Missing blocks:                0
 Corrupt blocks:                0
 Missing replicas:              12 (66.666664 %)
 Blocks queued for replication: 0

Erasure Coded Block Groups:
 Total size:    0 B
 Total files:   0
 Total block groups (validated):        0
 Minimally erasure-coded block groups:  0
 Over-erasure-coded block groups:       0
 Under-erasure-coded block groups:      0
 Unsatisfactory placement block groups: 0
 Average block group size:      0.0
 Missing block groups:          0
 Corrupt block groups:          0
 Missing internal blocks:       0
 Blocks queued for replication: 0
FSCK ended at Tue Aug 06 12:40:37 IST 2024 in 6 milliseconds


The filesystem under path '/' is HEALTHY
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/apache-hive-3.1.2-bin/bin$
```

6. Count number of directories in HDFS

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/apache-hive-3.1.2-bin/bin$ hdfs dfs -count -q /
9223372036854775807 9223372036854775785          none             inf           13            9              170 /
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/apache-hive-3.1.2-bin/bin$
```

7. Report the amount of space used and # available on currently mounted file system

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/apache-hive-3.1.2-bin/bin$ hdfs dfsadmin -report
Configured Capacity: 660940750848 (615.55 GB)
Present Capacity: 610631753728 (568.70 GB)
DFS Remaining: 610631663616 (568.70 GB)
DFS Used: 90112 (88 KB)
DFS Used%: 0.00%
Replicated Blocks:
        Under replicated blocks: 6
        Blocks with corrupt replicas: 0
        Missing blocks: 0
        Missing blocks (with replication factor 1): 0
        Low redundancy blocks with highest priority to recover: 6
        Pending deletion blocks: 0
Erasure Coded Block Groups:
        Low redundancy block groups: 0
        Block groups with corrupt internal blocks: 0
        Missing block groups: 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0

-------------------------------------------------
Live datanodes (1):

Name: 127.0.0.1:9866 (localhost)
Hostname: mca-HP-Elite-Tower-800-G9-Desktop-PC
Decommission Status : Normal
Configured Capacity: 660940750848 (615.55 GB)
DFS Used: 90112 (88 KB)
Non DFS Used: 16659828736 (15.52 GB)
DFS Remaining: 610631663616 (568.70 GB)
DFS Used%: 0.00%
DFS Remaining%: 92.39%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 0
Last contact: Tue Aug 06 12:44:27 IST 2024
Last Block Report: Tue Aug 06 12:21:30 IST 2024
Num of Blocks: 6
```

8. Empty the trash

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -expunge
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

9. Find the number of lines a file contains that is stored in HDFS.

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/apache-hive-3.1.2-bin/bin$ hadoop fs -cat /wc_output/part-00000 | wc -l
5
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/apache-hive-3.1.2-bin/bin$
```

3. Create and execute map reduce programs

code

Map Reduce – Word Count Program

WC_Mapper

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class WC_Mapper extends MapReduceBase implements
Mapper<LongWritable,Text,Text,IntWritable>{
private final static IntWritable one = new IntWritable(1);
private Text word = new Text();
public void map(LongWritable key, Text value,OutputCollector<Text,IntWritable> output,
Reporter reporter) throws IOException{
String line = value.toString();
StringTokenizer tokenizer = new StringTokenizer(line);
while
(tokenizer.hasMoreTokens()){ word.set(tokenizer.next
Token());
output.collect(word, one);
}
}
}
```

WC REDUCER

```
import java.IOException;

import java.util.iterator

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;

import  org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reducer;

import org.apache.hadoop.mapred.Reporter;

public class WC_Reducer extends MapReduceBase implements

Reducer<Text,IntWritable,Text,IntWritable> {

public void reduce(Text key, Iterator<IntWritable>
values,OutputCollector<Text,IntWritable> output,

Reporter reporter) throws IOException

{int sum=0;

while (values.hasNext())

{sum+=values.next().get();

}

output.collect(key,new IntWritable(sum));

}

}
```

WC_Runner

```
import java.io.IOException;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.mapred.TextInputFormat;
import org.apache.hadoop.mapred.TextOutputFormat;
public class WC_Runner {
public static void main(String[] args) throws
IOException{JobConf conf = new
JobConf(WC_Runner.class);
conf.setJobName("WordCount");
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
conf.setMapperClass(WC_Mapper.class);
conf.setCombinerClass(WC_Reducer.class);
conf.setReducerClass(WC_Reducer.class);
conf.setInputFormat(TextInputFormat.class);
conf.setOutputFormat(TextOutputFormat.class);
FileInputFormat.setInputPaths(conf,new Path(args[0]));
FileOutputFormat.setOutputPath(conf,new Path(args[1]));
JobClient.runJob(conf);
}
```

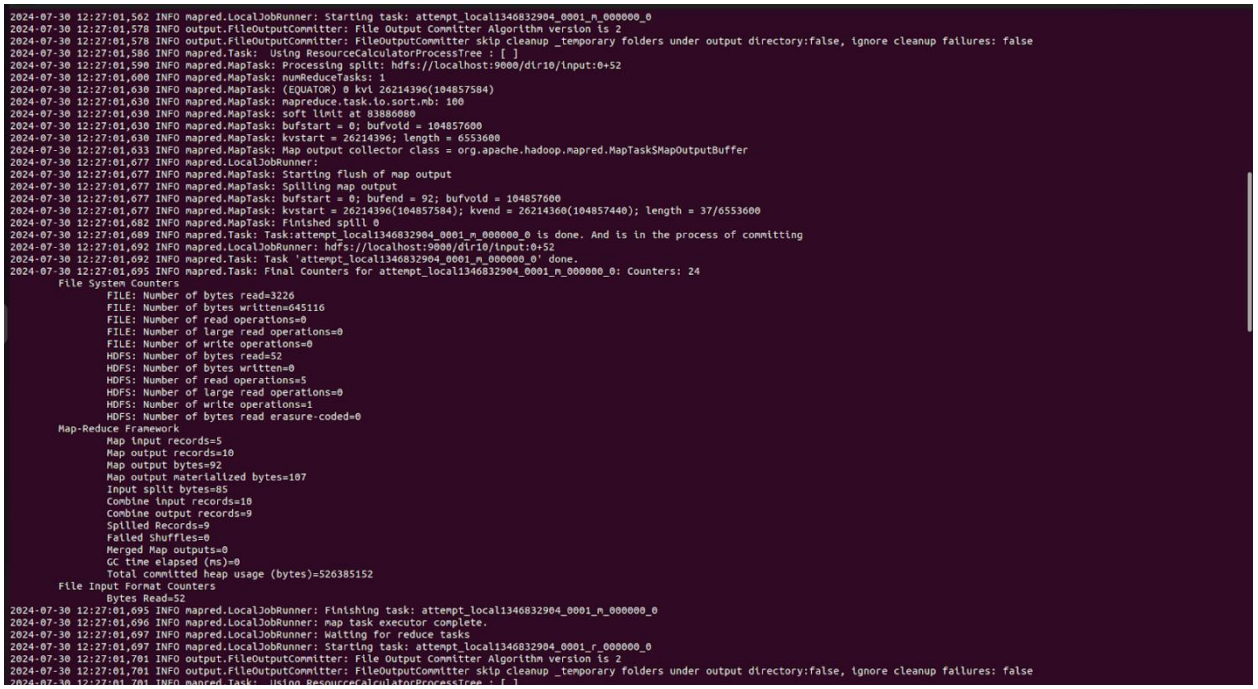
Output:

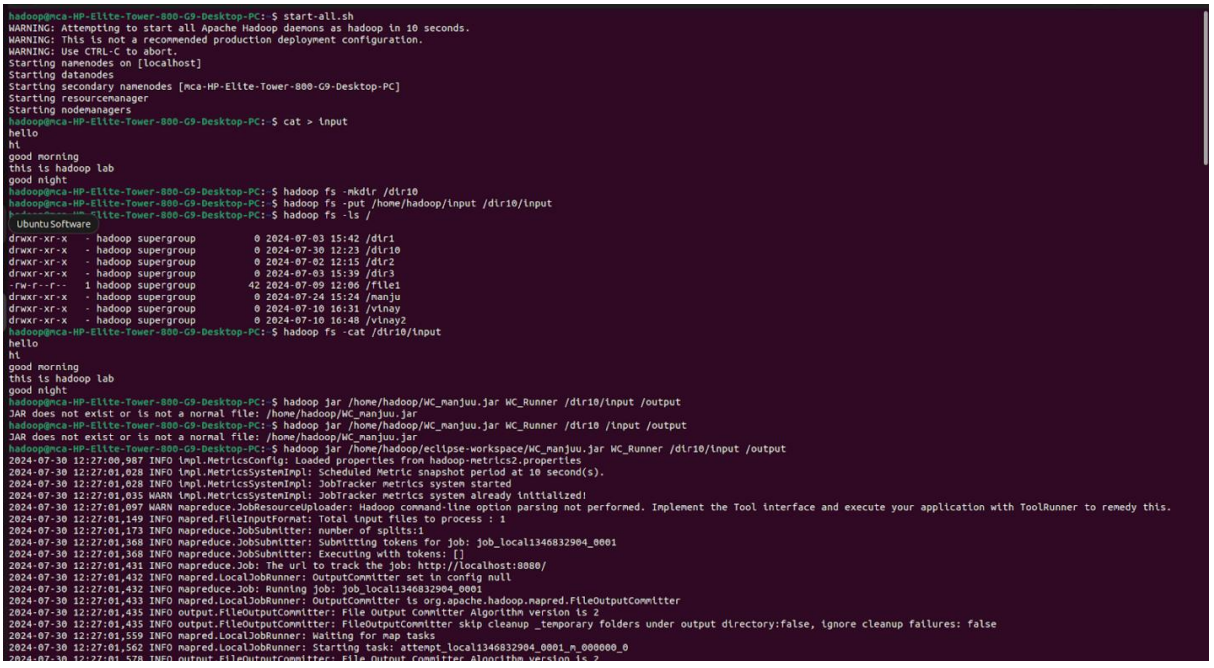
In Hadoop terminal:

$ cat > inp

$ hadoop fs -put /home/hadoop/inp /

$ hadoop fs -ls /

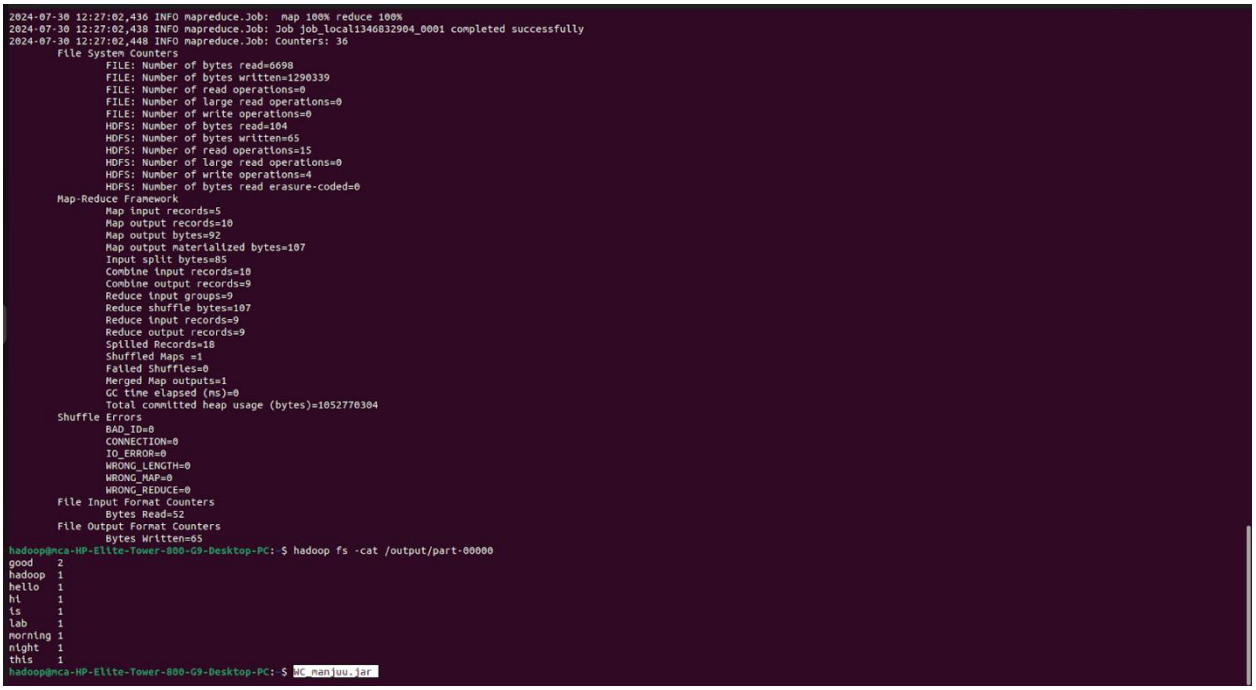




```
2024-07-30 12:27:01,731 INFO mapred.Merger: Merging 1 sorted segments
2024-07-30 12:27:01,731 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 96 bytes
2024-07-30 12:27:01,732 INFO reduce.MergeManagerImpl: Merged 1 segments, 103 bytes to disk to satisfy reduce memory limit
2024-07-30 12:27:01,732 INFO reduce.MergeManagerImpl: Merging 1 files, 107 bytes from disk
2024-07-30 12:27:01,732 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into reduce
2024-07-30 12:27:01,732 INFO mapred.Merger: Merging 1 sorted segments
2024-07-30 12:27:01,733 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 96 bytes
2024-07-30 12:27:01,733 INFO mapred.LocalJobRunner: 1 / 1 copied.
2024-07-30 12:27:01,804 INFO mapred.Task: Task:attempt_local1346832904_0001_r_000000_0 is done. And is in the process of committing
2024-07-30 12:27:01,805 INFO mapred.LocalJobRunner: 1 / 1 copied.
2024-07-30 12:27:01,805 INFO mapred.Task: Task attempt_local1346832904_0001_r_000000_0 is allowed to commit now
2024-07-30 12:27:01,820 INFO output.FileOutputCommitter: Saved output of task 'attempt_local1346832904_0001_r_000000_0' to hdfs://localhost:9000/output
2024-07-30 12:27:01,821 INFO mapred.LocalJobRunner: reduce > reduce
2024-07-30 12:27:01,821 INFO mapred.Task: Task 'attempt_local1346832904_0001_r_000000_0' done.
2024-07-30 12:27:01,821 INFO mapred.Task: Final Counters for attempt_local1346832904_0001_r_000000_0: Counters: 30
        File System Counters
                FILE: Number of bytes read=3472
                FILE: Number of bytes written=645223
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=52
                HDFS: Number of bytes written=65
                HDFS: Number of read operations=10
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=3
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Combine input records=0
                Combine output records=0
                Reduce input groups=9
                Reduce shuffle bytes=107
                Reduce input records=9
                Reduce output records=9
                Spilled Records=9
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=526385152
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Output Format Counters
                Bytes Written=65
2024-07-30 12:27:01,821 INFO mapred.LocalJobRunner: Finishing task: attempt_local1346832904_0001_r_000000_0
2024-07-30 12:27:01,821 INFO mapred.LocalJobRunner: reduce task executor complete.
2024-07-30 12:27:02,435 INFO mapreduce.Job: Job job_local1346832904_0001 running in uber mode : false
2024-07-30 12:27:02,436 INFO mapreduce.Job:  map 100% reduce 100%
2024-07-30 12:27:02,438 INFO mapreduce.Job: Job job_local1346832904_0001 completed successfully
2024-07-30 12:27:02,448 INFO mapreduce.Job: Counters: 36
        File System Counters
```

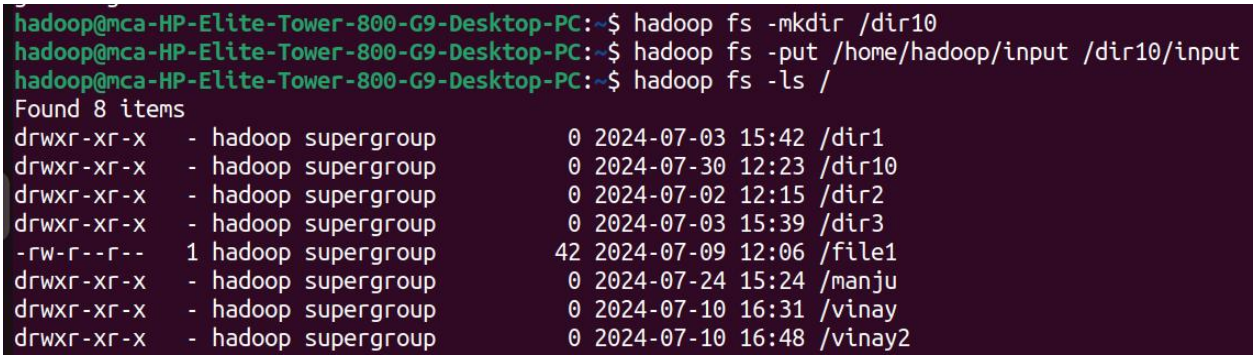
```
2024-07-31 15:23:31,817 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2024-07-31 15:23:31,817 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2024-07-31 15:23:31,818 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin: org.apache.hadoop.mapreduce.task.reduce.Shuffle@1e5076c1
2024-07-31 15:23:31,819 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2024-07-31 15:23:31,827 INFO reduce.MergeManagerImpl: MergerManager: memoryLimit=5832389120, maxSingleShuffleLimit=1458097280, mergeThreshold=3849377024, ioSortFactor=10, memToMemMergeOutputsThreshold=1
2024-07-31 15:23:31,828 INFO reduce.EventFetcher: attempt_local1380549949_0001_r_000000_0 Thread started: EventFetcher for fetching Map Completion Events
2024-07-31 15:23:31,841 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of map attempt_local1380549949_0001_m_000000_0 decomp: 199 len: 203 to MEMORY
2024-07-31 15:23:31,843 INFO reduce.InMemoryMapOutput: Read 199 bytes from map-output for attempt_local1380549949_0001_m_000000_0
2024-07-31 15:23:31,844 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of size: 199, inMemoryMapOutputs.size() -> 1, commitMemory -> 0, usedMemory ->199
2024-07-31 15:23:31,844 INFO reduce.EventFetcher: EventFetcher is interrupted.. Returning
2024-07-31 15:23:31,844 INFO mapred.LocalJobRunner: 1 / 1 copied.
2024-07-31 15:23:31,845 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory map-outputs and 0 on-disk map-outputs
2024-07-31 15:23:31,847 INFO mapred.Merger: Merging 1 sorted segments
2024-07-31 15:23:31,847 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 193 bytes
2024-07-31 15:23:31,848 INFO reduce.MergeManagerImpl: Merged 1 segments, 199 bytes to disk to satisfy reduce memory limit
2024-07-31 15:23:31,848 INFO reduce.MergeManagerImpl: Merging 1 files, 203 bytes from disk
2024-07-31 15:23:31,849 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into reduce
2024-07-31 15:23:31,849 INFO mapred.Merger: Merging 1 sorted segments
2024-07-31 15:23:31,849 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 193 bytes
2024-07-31 15:23:31,849 INFO mapred.LocalJobRunner: 1 / 1 copied.
2024-07-31 15:23:31,924 INFO mapred.Task: Task:attempt_local1380549949_0001_r_000000_0 is done. And is in the process of committing
2024-07-31 15:23:31,926 INFO mapred.LocalJobRunner: 1 / 1 copied.
2024-07-31 15:23:31,926 INFO mapred.Task: Task attempt_local1380549949_0001_r_000000_0 is allowed to commit now
2024-07-31 15:23:31,941 INFO output.FileOutputCommitter: Saved output of task 'attempt_local1380549949_0001_r_000000_0' to hdfs://localhost:9000/revathi/WC_output
2024-07-31 15:23:31,941 INFO mapred.LocalJobRunner: reduce > reduce
2024-07-31 15:23:31,941 INFO mapred.Task: Task 'attempt_local1380549949_0001_r_000000_0' done.
2024-07-31 15:23:31,942 INFO mapred.Task: Final Counters for attempt_local1380549949_0001_r_000000_0: Counters: 30
        File System Counters
                FILE: Number of bytes read=4448
                FILE: Number of bytes written=646233
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=106
                HDFS: Number of bytes written=129
                HDFS: Number of read operations=10
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=3
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Combine input records=0
                Combine output records=0
                Reduce input groups=17
                Reduce shuffle bytes=203
                Reduce input records=17
                Reduce output records=17
                Spilled Records=17
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=526385152
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
```

```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop fs -ls /revathi/WC_output
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-31 15:23 /revathi/WC_output/_SUCCESS
-rw-r--r--   1 hadoop supergroup        129 2024-07-31 15:23 /revathi/WC_output/part-00000
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop fs -cat /revathi/WC_output/part-00000
Big     1
Mapreducer      1
This    1
concept 1
data    1
good    2
hello   1
hi      1
is      2
map     1
morning 1
night   1
of      2
one     1
program 1
reducer 1
the     1
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC: $
```



$ hadoop fs -ls /WC_out

$ hadoop fs -cat /WC_output/part-00000



4. Data Processing Using Hive

Rent a Cab Database

Database name: Cabbase

Create a database with the name Cabbase and perform hive queries.

cab

cab_id	cab_number	cab_type	cab_status	cab_price

cab_status: B (booked), A (available), NA (not available)
cab_type: Mini, SUV, XUV, …

Driver

driver_id	driver_name	driver_sal

Rental

rental_id	rental_date	rental_time	rental_dest	Payment

Customer

cus_id	cus_fname	cus_lname	cus_gender	cus_age	cus_phno	cus_email

cus_gend: M (male), F (female)

Transaction

tran_id	tran_name	tran_date	car_id	rental_id	cust_id

Implement the following queries:

1. Display the contents of all tables.
2. Limit the display to three rows or n no. of rows.
3. Display the count of each car type.
4. Display the driver details whose salary is less than 30000.
5. Display the rental details where the payment is maximum.
6. Create a partition for customers based on male and female.
7. Create three buckets on drivers based on salary.
8. Display the transaction details of transactions that happened during the year 2024.
9. Display the count of no. of cabs that are available (status='A')
10. Display the average salary of all the drivers.

1. Display the contents of all tables

```
hive> select * from cab;
OK
111     KA05MP3792      mini    B       2100
112     KA05ES9352      SUV     A       1500
113     KA25PD2321      XUV     NA      3250
114     KA47WQ4724      mini    A       2700
115     KA21JB2392      SUV     A       4500
Time taken: 0.046 seconds, Fetched: 5 row(s)
hive> select * from driver;
OK
402     Pramod  20000
412     Praveen 18000
732     Chinmay 15000   NULL
784     sindoor 22000
289     shamant 20000
Time taken: 0.051 seconds, Fetched: 5 row(s)
hive> select * from rental;
OK
1155    21-08-2024      12:00   bommanahalli    1200
1233    22-08-2024      15:45   JPNagar 1400
8122    21-08-2024      21:30   jayanagar       1450
7712    23-08-2024      10:15   basavanagudi    1500
9921    29-08-2024      08:00   uttarahalli     1100
Time taken: 0.048 seconds, Fetched: 5 row(s)
hive> select * from customer;
OK
999     sunil   dutt    male    45      sunildutt@gmail.com
998     umesh   ambigar male    51      umesh@gmail.com
997     renuka  devi    female  37      renuka@gmail.com
996     tanaji  patil   male    29      tanaji@gmail.com
995     ankita  shetty  female  25      ankita@gmail.com
Time taken: 0.04 seconds, Fetched: 5 row(s)
hive> select * from transaction;
OK
1       online  21-08-2024      111     1155    999
2       cash    22-08-2024      112     1233    998
3       online  21-08-2024      113     8122    997
4       cash    23-08-2024      114     7712    996
5       cash    29-08-2024      115     9921    995
Time taken: 0.044 seconds, Fetched: 5 row(s)
hive>
```

2. Limit the display to three rows or n no. of rows

```
hive> select * from cab limit 3;
OK
111     KA05MP3792      mini    B       2100
112     KA05ES9352      SUV     A       1500
113     KA25PD2321      XUV     NA      3250
Time taken: 0.054 seconds, Fetched: 3 row(s)
hive> select * from driver limit 3;
OK
402     Pramod  20000
412     Praveen 18000
732     Chinmay 15000   NULL
Time taken: 0.047 seconds, Fetched: 3 row(s)
hive> select * from rental limit 3;
OK
1155    21-08-2024      12:00   bommanahalli    1200
1233    22-08-2024      15:45   JPNagar 1400
8122    21-08-2024      21:30   jayanagar       1450
Time taken: 0.046 seconds, Fetched: 3 row(s)
hive> select * from customer limit 3;
OK
999     sunil   dutt    male    45      sunildutt@gmail.com
998     umesh   ambigar male    51      umesh@gmail.com
997     renuka  devi    female  37      renuka@gmail.com
Time taken: 0.045 seconds, Fetched: 3 row(s)
hive> select * from transaction limit 3;
OK
1       online  21-08-2024      111     1155    999
2       cash    22-08-2024      112     1233    998
3       online  21-08-2024      113     8122    997
Time taken: 0.043 seconds, Fetched: 3 row(s)
hive>
```

3. Display the count of each car type

```
hive> select count(distinct(cab_type)) from cab;
Query ID = hadoop_20240820124938_023ab89d-f0bd-4857-96c9-0f3327f99edf
Total jobs = 1
Launching Job 1 out of 1
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:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-08-20 12:49:40,358 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local109281159_0001
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 3366 HDFS Write: 742 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
3
Time taken: 1.705 seconds, Fetched: 1 row(s)
hive>
```

4. Display the driver details whose salary is less than 30000

```
hive> select * from driver where driver_sal<30000;
OK
402     Pramod  20000
412     Praveen 18000
784     sindoor 22000
289     shamant 20000
Time taken: 0.074 seconds, Fetched: 4 row(s)
hive>
```

5. Display the rental details where the payment is maximum.

```
hive> select * from rental order by payment desc limit 1;
Query ID = hadoop_20240820161145_06f90791-8cfe-47a1-a2c5-7454efd1418e
Total jobs = 1
Launching Job 1 out of 1
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:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-08-20 16:11:46,856 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1319408926_0002
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 712 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
4       18-08-2024      10:00   JP Nagar        1500
Time taken: 1.219 seconds, Fetched: 1 row(s)
hive>
```

6. Create a partition for customers based on male and female.

```
hive> create table if not exists customer_partition (cus_id int, cus_fname string, cus_lname string, cus_age int, cus_email string) PARTITIONED by (cus_g
ender string) row format delimited fields terminated by '\t' lines terminated by '\n';
OK
Time taken: 0.101 seconds
```

```
hive> INSERT OVERWRITE TABLE customer_partition PARTITION(cus_gender="female") SELECT cus_num, cus_fname, cus_lname, cus_age, cus_email from customer whe
re cus_gender="female";
Query ID = hadoop_20240828091425_cc7ee49b-b465-4663-bb99-a5d3a616a3ba
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 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:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-08-28 09:14:26,643 Stage-1 map = 0%,  reduce = 100%
Ended Job = job_local1669910571_0002
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/customer_partition/cus_gender=female/.hive-staging_hive_2024-08-28_09-14-25_353_482896
7700205772601-1/-ext-10000
Loading data to table default.customer_partition partition (cus_gender=female)
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 1.455 seconds
hive>
```

7. **Create three buckets on drivers based on salary**.

```
hive> select driver_id,driver_name,driver_sal, CASE when driver_sal between 10000 and 15000 then 'entery-level salary'
    > when driver_sal between 15000 and 20000 then 'mid-level salary'
    > when driver_sal between 20000 and 25000 then 'high-level salary'
    > else 'not specified'
    > end as salary_bucket from driver;
OK
402     Pramod  20000   mid-level salary
412     Praveen 18000   mid-level salary
732     Chinmay 15000   entery-level salary
784     sindoor 22000   high-level salary
289     shamant 20000   mid-level salary
Time taken: 0.055 seconds, Fetched: 5 row(s)
```

8. Display the transaction details of transactions that happened during the year 2024.

```
hive> select * from transaction where substr(tran_date, 7, 4) = '2024';
OK
1       online  21-08-2024      111     1155    999
2       cash    22-08-2024      112     1233    998
3       online  21-08-2024      113     8122    997
4       cash    23-08-2024      114     7712    996
5       cash    29-08-2024      115     9921    995
Time taken: 0.089 seconds, Fetched: 5 row(s)
```

9. Display the count of no. of cabs that are available (status='A')

```
hive> select count(*)
    > from cab
    > where cab_status = 'A';
Query ID = hadoop_20240826144537_c09d19f8-7d89-425f-ad10-f41782460fda
Total jobs = 1
Launching Job 1 out of 1
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:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-08-26 14:45:39,480 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1843634690_0001
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 970 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
3
Time taken: 1.765 seconds, Fetched: 1 row(s)
```

10. Display the average salary of all the drivers.

```
hive> SELECT AVG(driver_sal) AS average_salary
    > FROM driver;
Query ID = hadoop_20240826144736_10f01778-f3db-4abb-95a6-11e6ff5fd5fc
Total jobs = 1
Launching Job 1 out of 1
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:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-08-26 14:47:37,798 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local567151975_0002
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 1148 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
19000.0
Time taken: 1.383 seconds, Fetched: 1 row(s)
```

5. Data processing using Spark

Data processing using Spark: Implement the following programs using PySpark.

Mca login:
start-all.sh
pyspark

1. Reading the input file and Calculating words count.

```
text_file = sc.textFile("/home/mca/count.txt") #Creating an RDD called text_file
data = text_file.flatMap(lambda x: x.split(' '))
map = data.map(lambda x: (x, 1))
mapreduce = map.reduceByKey(lambda x,y: x+y)
result = mapreduce.collect()
print(result)
```

```
>>> text_file = sc.textFile("/home/mca/count.txt") #Creating an RDD called text_file
>>> data = text_file.flatMap(lambda x: x.split(' '))
>>> map = data.map(lambda x: (x, 1))
>>> mapreduce = map.reduceByKey(lambda x,y: x+y)
>>> result = mapreduce.collect()
>>> print(result)
[('MCA', 1), ('of', 3), ('Applications', 2), ('Bachelor', 1), ('These', 1), ('deal', 1), ('Computers',
 1), ('stands', 2), ('for', 2), ('Master', 1), ('Computer', 2), ('BCA', 1), ('courses', 1), ('with', 1
), ('the', 1), ('application', 1), ('part', 1), ('and', 1), ('their', 1), ('technologies', 1)]
```

Or

```
text_file = sc.textFile("/home/mca/count.txt")
counts = text_file.flatMap(lambda line: line.split(" ")).map(lambda word: (word,
1)).reduceByKey(lambda x, y: x + y)
counts.collect()
```

```
>>>
>>> text_file = sc.textFile("/home/mca/count.txt")
>>> counts = text_file.flatMap(lambda line: line.split(" ")).map(lambda word: (word, 1)).reduceByKey(lambda x, y: x + y)
>>> counts.collect()
[('MCA', 1), ('of', 3), ('Applications', 2), ('Bachelor', 1), ('These', 1), ('deal', 1), ('Computers', 1), ('stands', 5), ('for', 2), ('Master', 1), ('Co
mputer', 2), ('BCA', 1), ('courses', 1), ('with', 1), ('the', 1), ('application', 1), ('part', 1), ('and', 1), ('their', 1), ('technologies', 1)]
>>>
```

2. Print the lines that contain a matching pattern.

```
text_file = sc.textFile("/home/mca/count.txt")
matched_lines = text_file.filter(lambda line: "Computer" in line)
matched_lines.count()  #print no. of lines contain the pattern
matched_lines.first()   # To print the first line
matched_lines.collect() # To print all the contents
```

```
>>>
>>> text_file = sc.textFile("/home/mca/count.txt")
>>> matched_lines = text_file.filter(lambda line: "Computer" in line)
>>> matched_lines.count()  #print no. of lines contain the pattern
3
>>> matched_lines.first()   # To print the first line
'MCA stands for Master of Computer Applications'
>>> matched_lines.collect() # To print all the contents
['MCA stands for Master of Computer Applications', 'BCA stands for Bachelor of Computer Applications', 'These courses deal with the application part of C
omputers and their technologies']
>>>
```

3. Convert the above program to count the words that appear 5 or more times, also remove case sensitive to match words.

```
from pyspark.sql.functions import col, lower, explode, split
df = spark.read.text("count.txt")
words_df = df.select(explode(split(lower(col("value")), "\\s+")).alias("word"))
word_counts = words_df.groupBy("word").count()
result = word_counts.filter(col("count") >= 5)
result.show()
```

```
>>> from pyspark.sql.functions import col, lower, explode, split
>>> df = spark.read.text("count.txt")
>>> words_df = df.select(explode(split(lower(col("value")), "\\s+")).alias("word"))
>>> word_counts = words_df.groupBy("word").count()
>>> result = word_counts.filter(col("count") >= 5)
>>> result.show()
+------+-----+
|  word|count|
+------+-----+
|stands|    5|
+------+-----+
```

For the given 'MovieLens' dataset, load data into Spark DataFrames, and explore tabular data with Spark SQL.

load the u.data into the new RDD.

#Load u.data into an RDD

>> ratings_rdd = sc.textFile("/home/mca/Downloads/u.data")

Display the first few records

>>> print(ratings_rdd.take(5))

```
>>> ratings_rdd = sc.textFile("/home/mca/Downloads/u.data")
>>>
>>> print(ratings_rdd.take(5))
['196\t242\t3\t881250949', '186\t302\t3\t891717742', '22\t377\t1\t878887116', '244\t51\t2\t880606923', '166\t346\t1\t886397596']
```

Step 4: Parse the RDD into a structured format

parsed_ratings_rdd = ratings_rdd.map(lambda

line:line.split('\t'))

print(parsed_ratings_rdd.take(5))

```
>>> print(parsed_ratings_rdd.take(5))
[['196', '242', '3', '881250949'], ['186', '302', '3', '891717742'],
['22', '377', '1', '878887116'], ['244', '51', '2', '880606923'], ['166', '346', '1', '886397596']]
```

Change the RDD to a Dataframe.

```
# Convert RDD to DataFrame with appropriate column names
>>> df_ratings = parsed_ratings_rdd.toDF(["user_id", "item_id", "rating", "timestamp"])
```

Return the schema of this DataFrame.

```
# Display the schema of the DataFrame
>>> df_ratings.printSchema()
```

```
>>> df_ratings.printSchema()
root
 |-- user_id: string (nullable = true)
 |-- item_id: string (nullable = true)
 |-- rating: string (nullable = true)
 |-- timestamp: string (nullable = true)
```

Register the DataFrame as a temp u_data table.

```
>>> df_ratings.createOrReplaceTempView("u_data")
```

Display the contents of newly created u_data table

```
>>> df_ratings.show()
```

OR

```
>>> spark.sql("SELECT * FROM u_data").show()
```

```
>>> df_ratings.show()
+-------+-------+------+---------+
|user_id|item_id|rating|timestamp|
+-------+-------+------+---------+
|    196|    242|     3|881250949|
|    186|    302|     3|891717742|
|     22|    377|     1|878887116|
|    244|     51|     2|880606923|
|    166|    346|     1|886397596|
|    298|    474|     4|884182806|
|    115|    265|     2|881171488|
|    253|    465|     5|891628467|
|    305|    451|     3|886324817|
|      6|     86|     3|883603013|
|     62|    257|     2|879372434|
|    286|   1014|     5|879781125|
|    200|    222|     5|876042340|
|    210|     40|     3|891035994|
|    224|     29|     3|888104457|
|    303|    785|     3|879485318|
|    122|    387|     5|879270459|
|    194|    274|     2|879539794|
|    291|   1042|     4|874834944|
|    234|   1184|     2|892079237|
+-------+-------+------+---------+
only showing top 20 rows
```

how the numbers of items reviewed by each user in the newly created u_data table.

```
>>> review_counts_by_user = spark.sql("""
...     SELECT user_id, COUNT(item_id) AS num_items_reviewed
...     FROM u_data
...     GROUP BY user_id
...     ORDER BY num_items_reviewed DESC  """)
```

>>> review_counts_by_user.show()

OR

Show the number of items reviewed by each user

>>> spark.sql("SELECT user_id, COUNT(*) AS num_reviews FROM u_data

GROUP BYuser_id").show()

>>> review_counts_by_user.sho

```
>>> review_counts_by_user.show()
+-------+------------------+
|user_id|num_items_reviewed|
+-------+------------------+
|    405|               737|
|    655|               685|
|     13|               636|
|    450|               540|
|    276|               518|
|    416|               493|
|    537|               490|
|    303|               484|
|    234|               480|
|    393|               448|
|    181|               435|
|    279|               434|
|    429|               414|
|    846|               405|
|      7|               403|
|     94|               400|
|    682|               399|
|    308|               397|
|     92|               388|
|    293|               388|
+-------+------------------+
only showing top 20 rows
```

Show the numbers of users reviewed each item in the newly created u_data table

>> review_counts_by_item = spark.sql("""

...SELECT item_id, COUNT(user_id) AS num_users_reviewed

...FROM u_data

...GROUP BY item_id

...ORDER BY num_users_reviewed DESC """)
>>> review_counts_by_item.show()

OR

Show the number of users who reviewed each item

>>> spark.sql("SELECT item_id, COUNT(*) AS num_reviews FROM u_data

GROUP BYitem_id").show()

```
>>> review_counts_by_item.show()
+-------+------------------+
|item_id|num_users_reviewed|
+-------+------------------+
|     50|               583|
|    258|               509|
|    100|               508|
|    181|               507|
|    294|               485|
|    286|               481|
|    288|               478|
|      1|               452|
|    300|               431|
|    121|               429|
|    174|               420|
|    127|               413|
|     56|               394|
|      7|               392|
|     98|               390|
|    237|               384|
|    117|               378|
|    172|               367|
|    222|               365|
|    313|               350|
+-------+------------------+
only showing top 20 rows
```

Load the u.user into a new RDD.

Load u.user into an RDD

>>> users_rdd = sc.textFile("/home/mca/Downloads/u.user")

>>> print(users_rdd.take(5))

```
>>> print(users_rdd.take(5))
['1|24|M|technician|85711', '2|53|F|other|94043',
>>>
```

```
'3|23|M|writer|32067', '4|24|M|technician|43537', '5|33|F|other|15213']
```

Change the RDD to a Dataframe.

Parse the RDD into a structured format

```
>>> parsed_users_rdd = users_rdd.map(lambda line: line.split('|'))
```

Convert RDD to DataFrame with appropriate column names

```
>>> df_users = parsed_users_rdd.toDF(["user_id", "age", "gender", "occupation",
"zip_code"])
```

```
>>> df_users.printSchema()
```

```
>>> df_users.printSchema()
root
 |-- user_id: string (nullable = true)
 |-- age: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- occupation: string (nullable = true)
 |-- zip_code: string (nullable = true)
```

OR

Define the schema

```
>>> from pyspark.sql.types import StructType, StructField, IntegerType, StringType,
LongType
>>> user_schema = StructType([
...  StructField("userId", IntegerType(), True),
...  StructField("age", IntegerType(), True),
...  StructField("gender", StringType(), True),
...StructField("occupation", StringType(), True),
...StructField("zip", StringType(), True)
>>> # Convert the parsed RDD to a DataFrame
>>> users_df = spark.createDataFrame(parsed_users_rdd, schema=user_schema)
>>> users_df.printSchema()
```

```
>>> df_users.printSchema()
root
 |-- user_id: string (nullable = true)
 |-- age: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- occupation: string (nullable = true)
 |-- zip_code: string (nullable = true)
```

Register the DataFrame as a temp u_user table.
 # Register the DataFrame as a temporary view

```
>>> df_users.createOrReplaceTempView("u_user")
```

Display the contents of newly created user table

```
>>> # Display the contents of the u_user table
>>> df_users.show()
```

OR

```
>>> spark.sql("SELECT * FROM df_users").show()
```

```
>>> df_users.show()
+-------+---+------+-------------+--------+
|user_id|age|gender|   occupation|zip_code|
+-------+---+------+-------------+--------+
|      1| 24|     M|   technician|   85711|
|      2| 53|     F|        other|   94043|
|      3| 23|     M|       writer|   32067|
|      4| 24|     M|   technician|   43537|
|      5| 33|     F|        other|   15213|
|      6| 42|     M|    executive|   98101|
|      7| 57|     M|administrator|   91344|
|      8| 36|     M|administrator|   05201|
|      9| 29|     M|      student|   01002|
|     10| 53|     M|       lawyer|   90703|
|     11| 39|     F|        other|   30329|
|     12| 28|     F|        other|   06405|
|     13| 47|     M|     educator|   29206|
|     14| 45|     M|    scientist|   55106|
|     15| 49|     F|     educator|   97301|
|     16| 21|     M|entertainment|   10309|
|     17| 30|     M|   programmer|   06355|
|     18| 35|     F|        other|   37212|
|     19| 40|     M|    librarian|   02138|
|     20| 42|     F|    homemaker|   95660|
+-------+---+------+-------------+--------+
only showing top 20 rows
```

Count the number of user in the u_user table gender wise

```
>>> gender_counts = spark.sql("""
... SELECT gender, COUNT(user_id) AS num_users
... FROM u_user
... GROUP BY gender
... """)
>>> gender_counts.show()
```

```
...     )
>>> gender_counts.show()
+------+---------+
|gender|num_users|
+------+---------+
|     F|      273|
|     M|      670|
+------+---------+
```

OR

```
>>> # Count the number of users by gender
>>> spark.sql("SELECT gender, COUNT(*) AS num_users FROM u_use

GROUP BYgender").show()
```

```
...     )
>>> gender_counts.show()
+------+---------+
|gender|num_users|
+------+---------+
|     F|      273|
|     M|      670|
+------+---------+
```

Join u_data table and u_user tables based on userid

```
# Register DataFrames as temporary views
>>> df_ratings.createOrReplaceTempView("u_data")
>>> df_users.createOrReplaceTempView("u_user")
>>> joined_df = spark.sql("""
... SELECT d.user_id, d.item_id, d.rating, d.timestamp, u.age, u.gender, u.occupation,
u.zip_code
... FROM u_data d
... JOIN u_user u
... ON d.user_id = u.user_id
... """)
>>> joined_df.show()
```

OR

```
>>> joined_df = df_ratings.join(df_users, on="user_id", how="inner")
>>> joined_df.show()
```

OR

```
>>> joined_df = spark.sql("""
... SELECT u.user_id, u.item_id, u.rating, u.timestamp,
... us.age, us.gender, us.occupation, us.zip_code
... FROM u_data u
... JOIN u_user us
... ON u.user_id = us.user_id
... """)
>>>joined_df.show()
```

```
>>> joined_df.show()
+-------+-------+------+---------+---+------+-------------+--------+
|user_id|item_id|rating|timestamp|age|gender|   occupation|zip_code|
+-------+-------+------+---------+---+------+-------------+--------+
|    296|    705|     5|884197193| 43|     F|administrator|   16803|
|    296|    508|     5|884196584| 43|     F|administrator|   16803|
|    296|     20|     5|884196921| 43|     F|administrator|   16803|
|    296|    228|     4|884197264| 43|     F|administrator|   16803|
|    296|    222|     5|884196640| 43|     F|administrator|   16803|
|    296|    429|     5|884197330| 43|     F|administrator|   16803|
|    296|    855|     5|884197352| 43|     F|administrator|   16803|
|    296|    248|     5|884196765| 43|     F|administrator|   16803|
|    296|    258|     5|884196469| 43|     F|administrator|   16803|
|    296|    242|     4|884196057| 43|     F|administrator|   16803|
|    296|     48|     5|884197091| 43|     F|administrator|   16803|
|    296|    286|     5|884196209| 43|     F|administrator|   16803|
|    296|    272|     5|884198772| 43|     F|administrator|   16803|
|    296|    510|     5|884197264| 43|     F|administrator|   16803|
|    296|    275|     4|884196555| 43|     F|administrator|   16803|
|    296|    427|     5|884198772| 43|     F|administrator|   16803|
|    296|     83|     5|884199624| 43|     F|administrator|   16803|
|    296|    961|     5|884197287| 43|     F|administrator|   16803|
|    296|    544|     4|884196938| 43|     F|administrator|   16803|
|    296|     32|     4|884197131| 43|     F|administrator|   16803|
+-------+-------+------+---------+---+------+-------------+--------+
only showing top 20 rows
```

6. Programming in Cassandra

Perform the following operations on Cassandra

- **Create KeySpace "Students"**
- **Describe the existing Keyspaces**
- **Display for More details on existing keyspaces**
- **Use the keyspace "Students"**

- **Student details:** Create table (column family) by name Student_Info(RollNo int, StudName text, DateOfJoining timestamp, PrevSemPercentage double), RollNo is primary key

- **Book_Borrowed:** Create table (column family) by name Library_Book (CountValue counter, BookName varchar, RollNo int, StudName varchar), PRIMARY KEY is (book_name,stud_name))

- **Lookup the names of all tables in the current keyspace**
- **Describe the table information**

CRUD

Perform the following queries on the tables created.

- **Insert at least 5 rows for Student_Info**
- **View data from the table "Students_Info"**
- **View data from the table "Students_Info" where RollNo column either has a value 1 or 2 or 3**
- **To execute a non-primary key**
  Create an INDEX on the Column StudName
- **Execute the query based on the INDEXED Column:**
  Display details for a specific student name.
- **Specify the number of rows to display in the output**
- **Alias for Column:**
  Display RollNo as "USN"
- **UPDATE the student's name with last name for a specific RollNo**
- **Change the RollNo to 10 for a existing RollNo with value 1**
- **DELETE PrevSemPercent for student with RollNo=2;**
- **Delete a Row FROM student_info WHERE RollNo is 3;**

Set Collection
A column of type set consists of unordered unique values. However, when the column is queried, it returns the values in sorted order. For example, for text values, it sorts in alphabetical order.

- **Alter the StudentsInfo table to add hobbies as a set of text**

List Collection
When the order of elements matter, one should go for a list collection.

- **Alter the StudentsInfo table to add language as a list of text**
- **Update the values for hobbies column (Music Cricket) and language column (Kannada, Hindi, English) for RollNo with value 10 and display the student-info**
- **Remove Hindi from the language list for RollNo 10 and display the student info**

USING A COUNTER

A counter is a special column that is changed in increments. For example, we may need a counter column to count the number of times a particular book is issued from the library bythe student.

- **Load data into the counter column**
- **Increase the counter column, CountValue by 1 in the table Library_Book for the student named as "Ram" and book names as "Big data Analytics"**

code

Create KeySpace “Students”

cqlsh> CREATE KEYSPACE Students WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};

```
cqlsh> CREATE KEYSPACE Student WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh>
```

Describe the existing Keyspaces

cqlsh:students> describe keyspaces;

```
cqlsh:students> describe keyspaces;

system_virtual_schema  system_schema  system_views  system_distributed  schema1
students               system_auth    system        system_traces

cqlsh:students> describe tables;

library_book_counter  student_info          library_book
library_book_info     library_book_counter1
```

Display for More details on existing keyspaces

cqlsh:students> describe keyspace Students;

```
cqlsh:students> describe keyspace Students;

CREATE KEYSPACE students WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'}  AND durable_writes = true;
```

Use the keyspace “Students”

cqlsh>use Students;

```
cqlsh> use Students;
cqlsh:students>
```

Student details:

Create table (column family) by name Student_Info(RollNo int, StudName text,DateOfJoining timestamp, PrevSemPercentage double), RollNo is primary key.

```
cqlsh:student> create table Student_Info (RollNo int primary key,
           ... StudName text,
           ... DateOfJoining timestamp,
           ... PrevSemPercentage double
           ... );
```

Book_Borrowed

Create table (column family) by name Library_Book (CountValue counter, BookNamevarchar, RollNo int, StudName varchar), PRIMARY KEY is (book_name,stud_name));

cqlsh:students> create table library_book(
... bookname text,
... studname text,
... countvalue counter,
... primary key(bookname,studname)
...);

```
cqlsh:students> create table library_book(
            ... bookname text,
            ... studname text,
            ... countvalue counter,
            ... primary key(bookname,studname)
            ... );
```

Lookup the names of all tables in the current keyspace.

cqlsh:students> describe tables;

```
cqlsh:students> describe tables;

library_book_counter  student_info          library_book
library_book_info     library_book_counter1
```

Describe the table information.

cqlsh:students> describe table Student_Info;

```
cqlsh:students> describe table Student_Info;

CREATE TABLE students.student_info (
    rollno int PRIMARY KEY,
    dataofjoining timestamp,
    hobbies set<text>,
    language list<text>,
    prevsempercentage double,
    studname text
) WITH additional_write_policy = '99p'
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair = 'BLOCKING'
    AND speculative_retry = '99p';
CREATE INDEX student_info_studname_idx ON students.student_info (studname);
```

cqlsh:students> describe table library_book;

```
cqlsh:students> describe table library_book;

CREATE TABLE students.library_book (
    bookname text,
    studname text,
    countvalue counter,
    PRIMARY KEY (bookname, studname)
) WITH CLUSTERING ORDER BY (studname ASC)
    AND additional_write_policy = '99p'
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair = 'BLOCKING'
    AND speculative_retry = '99p';
```

CRUD

Insert at least 5 rows for Student_Info

```
cqlsh:student> INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (1, 'Pramod', toTimestamp(now()), 70.5);
cqlsh:student> INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (2, 'Chinmay', toTimestamp(now()), 85.8);
cqlsh:student> INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (3, 'Sindoor', toTimestamp(now()), 92.1);
cqlsh:student> INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (4, 'Praveen', toTimestamp(now()), 89);
cqlsh:student> INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (5, 'Arjun', toTimestamp(now()), 99.9);
```

View data from the table "Students_Info"

```
cqlsh:student> select * from Student_Info;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+----------
      5 | 2024-09-10 06:28:22.893000+0000 |              99.9 |    Arjun
      1 | 2024-09-10 06:27:27.851000+0000 |              70.5 |   Pramod
      2 | 2024-09-10 06:27:44.468000+0000 |              85.8 |  Chinmay
      4 | 2024-09-10 06:28:04.972000+0000 |                89 |  Praveen
      3 | 2024-09-10 06:27:54.324000+0000 |              92.1 |  Sindoor

(5 rows)
```

View data from the table "Students_Info" where RollNo column either has a value 1 or2or 3

```
cqlsh:student> SELECT * FROM Student_Info WHERE RollNo IN (1, 2, 3);

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+----------
      1 | 2024-09-10 06:27:27.851000+0000 |              70.5 |   Pramod
      2 | 2024-09-10 06:27:44.468000+0000 |              85.8 |  Chinmay
      3 | 2024-09-10 06:27:54.324000+0000 |              92.1 |  Sindoor

(3 rows)
```

To execute a non primary key
Create an INDEX on the Column StudName

Execute the query based on the INDEXED Column:
Display students details for a specific student name.

```
cqlsh:student> SELECT * FROM Student_Info WHERE StudName = 'Pramod';

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+----------
      1 | 2024-09-10 06:27:27.851000+0000 |              70.5 |   Pramod

(1 rows)
```

Specify the number of rows to display in the output.

```
cqlsh:student> SELECT * FROM Student_Info LIMIT 2;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+----------
      5 | 2024-09-10 06:28:22.893000+0000 |              99.9 |    Arjun
      1 | 2024-09-10 06:27:27.851000+0000 |              70.5 |   Pramod

(2 rows)
```

Alias for Column:
Display RollNo as "USN"

```
cqlsh:student> SELECT RollNo AS USN FROM Student_Info;

 usn
-----
   5
   1
   2
   4
   3

(5 rows)
```

UPDATE the student name with last name for a specific RollNo

```
cqlsh:student> select * from Student_Info;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+----------
      5 | 2024-09-10 06:28:22.893000+0000 |              99.9 |    Arjun
      1 | 2024-09-10 06:27:27.851000+0000 |              70.5 |   Pramod
      2 | 2024-09-10 06:27:44.468000+0000 |              85.8 |   Partha
      4 | 2024-09-10 06:28:04.972000+0000 |                89 |  Praveen
      3 | 2024-09-10 06:27:54.324000+0000 |              92.1 |  Sindoor

(5 rows)
```

Change the RollNo to 10 for an existing RollNo with value 1.

```
cqlsh:student> BEGIN BATCH
DELETE FROM Student_Info WHERE RollNo = 1;
INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (10, 'Alice Johnson-Smith', toTimestamp(now()), 89.5);
APPLY BATCH;
cqlsh:student> select * from Student_Info;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+--------------------
      5 | 2024-09-10 06:28:22.893000+0000 |              99.9 |              Arjun
     10 | 2024-09-10 06:45:50.275000+0000 |              89.5 | Alice Johnson-Smith
      2 | 2024-09-10 06:27:44.468000+0000 |              85.8 |             Partha
      4 | 2024-09-10 06:28:04.972000+0000 |                89 |            Praveen
      3 | 2024-09-10 06:27:54.324000+0000 |              92.1 |            Sindoor

(5 rows)
```

DELETE PrevSemPercent for student with RollNo=2;

```
cqlsh:student> UPDATE Student_Info SET PrevSemPercentage = NULL WHERE RollNo = 2;
```

cqlsh:students> select * from Student_Info;

```
cqlsh:student> UPDATE Student_Info SET PrevSemPercentage = NULL WHERE RollNo = 2;
cqlsh:student> select * from Student_Info;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+--------------------
      5 | 2024-09-10 06:28:22.893000+0000 |              99.9 |              Arjun
     10 | 2024-09-10 06:45:50.275000+0000 |              89.5 | Alice Johnson-Smith
      2 | 2024-09-10 06:27:44.468000+0000 |              null |             Partha
      4 | 2024-09-10 06:28:04.972000+0000 |                89 |            Praveen
      3 | 2024-09-10 06:27:54.324000+0000 |              92.1 |             Sindoor

(5 rows)
```

Delete a Row FROM student_info WHERE RollNo is 3;

cqlsh:students> delete from Student_Info where RollNo=3;

```
cqlsh:student> DELETE FROM Student_Info WHERE RollNo = 3;
cqlsh:student> select * from Student_Info;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+--------------------
      5 | 2024-09-10 06:28:22.893000+0000 |              99.9 |              Arjun
     10 | 2024-09-10 06:45:50.275000+0000 |              89.5 | Alice Johnson-Smith
      2 | 2024-09-10 06:27:44.468000+0000 |              null |             Partha
      4 | 2024-09-10 06:28:04.972000+0000 |                89 |            Praveen

(4 rows)
```

cqlsh:students> select * from Student_Info;

Set Collection

Alter the StudentsInfo table to add hobbies as a set of textv

cqlsh:students> alter table Student_Info add hobbies set<text>;

```
cqlsh:students> alter table Student_Info add hobbies set<text>;
cqlsh:students>
```

List Collection

Alter the StudentsInfo table to add language as a list of text

cqlsh:students> alter table Student_Info add language list<text>;

```
cqlsh:students> alter table Student_Info add language list<text>;
cqlsh:students>
```

Update the values for hobbies column (Music Cricket) and language column (Kannada,
Hindi,English) for RollNo with value 10 and display the student-info

```
cqlsh:students> update Student_Info set hobbies={'Music','Cricket','Cycling'},
           ... language=['Kannada','English','Hindi'] where RollNo=10;
cqlsh:students> select * from Student_Info;
```

```
cqlsh:student> UPDATE Student_Info
SET hobbies = {'Music', 'Cricket'},
        language = ['Kannada', 'Hindi', 'English']
WHERE RollNo = 10;
cqlsh:student> select * from Student_Info;

 rollno | dateofjoining                   | hobbies             | language                        | prevsempercentage | studname
--------+---------------------------------+---------------------+---------------------------------+-------------------+--------------------
      5 | 2024-09-10 06:28:22.893000+0000 |                null |                            null |              99.9 |              Arjun
     10 | 2024-09-10 06:45:50.275000+0000 | {'Cricket', 'Music'} | ['Kannada', 'Hindi', 'English'] |              89.5 | Alice Johnson-Smith
      2 | 2024-09-10 06:27:44.468000+0000 |                null |                            null |              null |             Partha
      4 | 2024-09-10 06:28:04.972000+0000 |                null |                            null |                89 |             Praveen

(4 rows)
```

Remove Hindi from the language list for RollNo 10 and display the student info

```
cqlsh:students> update Student_Info set language=language-['Hindi']
           ... where RollNo=10;
cqlsh:students> select * from
```

```
cqlsh:student> UPDATE Student_Info
SET language = language - ['Hindi']
WHERE RollNo = 10;
cqlsh:student> select * from Student_Info;

 rollno | dateofjoining                   | hobbies             | language              | prevsempercentage | studname
--------+---------------------------------+---------------------+-----------------------+-------------------+--------------------
      5 | 2024-09-10 06:28:22.893000+0000 |                null |                  null |              99.9 |              Arjun
     10 | 2024-09-10 06:45:50.275000+0000 | {'Cricket', 'Music'} | ['Kannada', 'English'] |              89.5 | Alice Johnson-Smith
      2 | 2024-09-10 06:27:44.468000+0000 |                null |                  null |              null |             Partha
      4 | 2024-09-10 06:28:04.972000+0000 |                null |                  null |                89 |             Praveen

(4 rows)
```

Student_Info;

USING A COUNTER

Load data into the counter column

Increase the counter column, CountValue by 1 in the table Library_Book for the student named as “Ram” and book names as “Big data Analytics”

```
cqlsh:students> update library_book set CountValue=CountValue+1
           ... where bookname='Big Data Analytics' and studname='Ram';
cqlsh:students> select * from library_book;
```

```
cqlsh:students> UPDATE Library_Book
SET CountValue = CountValue + 1
WHERE Bookname = 'Big data Analytics' AND Studname = 'Ram';
cqlsh:students> SELECT * FROM Library_Book;

 bookname           | studname | countvalue
--------------------+----------+------------
 Big data Analytics |      Ram |          1
```