



ACADEMIC YEAR 2020-2021



KNOWLEDGE • CHARACTER • UNITY

BIGDATA LABORATORY

Report on,

Learning Activity II - Programming Assignment

Submitted by,

G SUSHANTH RAJU

1NT18IS058 / VI SEM ISE-A

Submitted to,

Mrs. Disha D N

Assistant Professor,

Department of Information Science and Engineering
Nitte Meenakshi Institute of Technology Bangalore-064

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

(An autonomous institution with A+ Grade by NAAC /UGC, Affiliated to Visvesvaraya Technological University, Belgaum, Approved by UGC/AICTE/Govt. of Karnataka)
Yelahanka, Bengaluru-560064

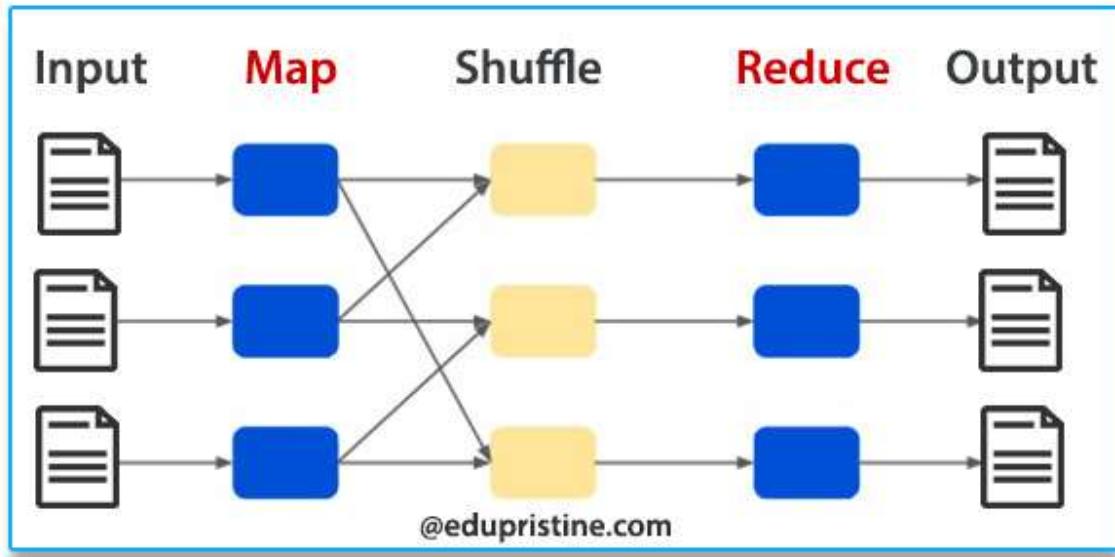
Table of Content:

- 1. A Brief About Hadoop Map Reduce and Figure**
- 2. Hadoop Problem Statements**
- 3. Dataset Description**
- 4. Source Code [GitHub link]**
- 5. Results and Snapshots of Hadoop Program**
- 6. A Brief About HIVE**
- 7. Queries of the Use-Cases**
- 8. HIVE Queries Results and Snapshots**
- 9. References**

1.HADOOP MAP-REDUCE

- Hadoop facilitates map-reduce programming which is set of interfaces.
- MapReduce is a processing technique and a program model for distributed computing based on java.
- Map-reduce contains two important interfaces:
 - 1.Mapper Interface
 - 2.Reducer Interface
- MAPPER-INTERFACE:
 - ❖ Mapping an input key-value pair into set of intermediary key-value pair. Input key-value pair, Intermediary key-value pair might be of different types.
 - ❖ Syntax:
 - void map (K1 key, V1 value, output Collector < K2, V2 >, Reporter reporter)
 - Output collector is an object which stores the output of mapper on key-value pair.
 - Reporter is an indicator , standard value repeat the program of mapping phase.
- REDUCER-INTERPHASE:
 - ❖ It is an entity which is responsible for performing the aggregate function. Reducer is a set of intermediating values which share a common key into similar set of values of perform aggregate operation on it.
 - ❖ 3 phases of reducer:
 - Shuffle
 - Sort
 - Reduce
 - ❖ Syntax:
 - Protected void reduce (Key1 key, Iterable < values > values, Reducer Context context) • It throws 2 exceptions :
 - IDEException and InterruptedException

➤ FIGURE



2. HADOOP PROBLEM STATEMENT

Exercise-I - Hadoop:

Create a dataset in excel as .csv file and it should contain the following fields with at least 20 sample datasets in it.

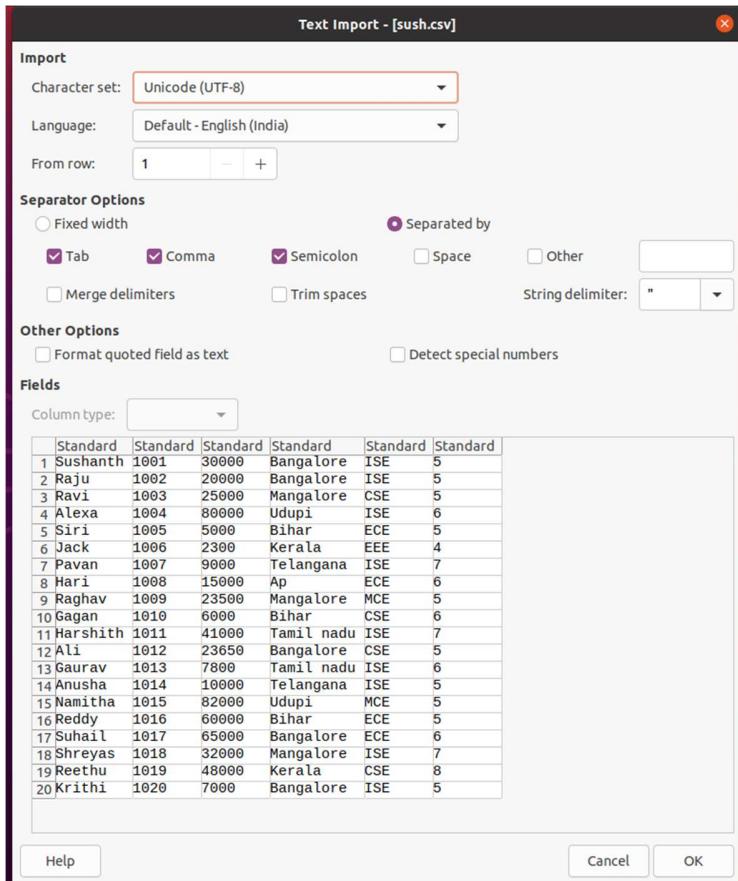
| Name | SSN | Salary | Address | Dname | Experience |
|--------|------|--------|-----------|-------|------------|
| Harsha | 5000 | 30000 | Bangalore | ISE | 5 |

Use the Hadoop MapReduce programming framework to come up with a Program which will take the data from this .csv file and computes the following.

1. Total number of employees who works in ISE Department.
2. Total number of employees with experience = 5 years.
3. Count the number of employees who lives in Bangalore

3. DATASET DESCRIPTION

Snapshot of the .csv file (sush.csv)



File Name/Type : sush/csv

File size : 620kb

Number of Rows : 20

Number of Columns : 6

4. GITHUB LINK

<https://github.com/Sushanth-raju/BigData.git>

5. RESULTS AND SNAPSHOTS OF HADOOP PROGRAM

- 1) Total number of employees who works in ISE Department.
Snapshots of the java program:

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - Employee1/src/Employee1/Employee1.java - Eclipse IDE
- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project management.
- Left Sidebar:** Activities view and Package Explorer showing a project named "Employee1" containing "Employee1" and "sush1".
- Central Area:** Employee1.java code editor with the following content:

```
1 package Employee1;
2 import java.io.IOException;
3 import java.util.*;
4 import org.apache.hadoop.fs.Path;
5 import org.apache.hadoop.conf.*;
6 import org.apache.hadoop.mapreduce.*;
7 import org.apache.hadoop.mapred.*;
8 import org.apache.hadoop.util.*;
9
10 public class Employee1 {
11     //MAPPER CODE
12
13     public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
14         private final static IntWritable one = new IntWritable(1);
15         private Text word = new Text();
16
17         public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
18             String data = value.toString();
19             String[] Ecount= data.split(",");
20
21             if (Ecount[4].equals("ISE")) {
22                 output.collect(new Text("Total no.of employees working in ISE Department : "), one);
23             }
24         }
25     }
26
27     //REDUCER CODE
28     public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {
29         public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
30             int val = 0 ;
31             while(values.hasNext()) {
32                 val += values.next().get();
33             }
34             output.collect(key, new IntWritable(val));
35         }
36     }
37     //DRIVER CODE
38     public static void main(String[] args) throws Exception {
39         JobConf conf = new JobConf(Employee1.class);
40         conf.setJobName("Total no.of employees working in ISE Department");
41         conf.setOutputKeyClass(Text.class);
42         conf.setOutputValueClass(IntWritable.class);
43         conf.setMapperClass(Map.class);
44         conf.setCombinerClass(Reduce.class);
45         conf.setReducerClass(Reduce.class);
46         conf.setInputFormat(TextInputFormat.class);
47         conf.setOutputFormat(TextOutputFormat.class);
48         FileInputFormat.setInputPaths(conf, new Path(args[0]));
49         FileOutputFormat.setOutputPath(conf, new Path(args[1]));
50         JobClient.runJob(conf);
51     }
52 }
53 }
```

Bottom Status Bar: Description, Resource, Path, Location, Type, Writable, Smart Insert, 23:61:771, 140M of 256M

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - Employee1/src/Employee1/Employee1.java - Eclipse IDE
- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project management.
- Left Sidebar:** Activities view and Package Explorer showing a project named "Employee1" containing "Employee1" and "sush1".
- Central Area:** Employee1.java code editor with the following content:

```
1 package Employee1;
2 import java.io.IOException;
3 import java.util.*;
4 import org.apache.hadoop.fs.Path;
5 import org.apache.hadoop.conf.*;
6 import org.apache.hadoop.mapreduce.*;
7 import org.apache.hadoop.mapred.*;
8 import org.apache.hadoop.util.*;
9
10 public class Employee1 {
11     private final static IntWritable one = new IntWritable(1);
12
13     public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
14         String data = value.toString();
15         String[] Ecount= data.split(",");
16
17         if (Ecount[4].equals("ISE")) {
18             output.collect(new Text("Total no.of employees working in ISE Department : "), one);
19         }
20     }
21
22     //REDUCER CODE
23     public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {
24         public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
25             int val = 0 ;
26             while(values.hasNext()) {
27                 val += values.next().get();
28             }
29             output.collect(key, new IntWritable(val));
30         }
31     }
32
33     //DRIVER CODE
34     public static void main(String[] args) throws Exception {
35         JobConf conf = new JobConf(Employee1.class);
36         conf.setJobName("Total no.of employees working in ISE Department");
37         conf.setOutputKeyClass(Text.class);
38         conf.setOutputValueClass(IntWritable.class);
39         conf.setMapperClass(Map.class);
40         conf.setCombinerClass(Reduce.class);
41         conf.setReducerClass(Reduce.class);
42         conf.setInputFormat(TextInputFormat.class);
43         conf.setOutputFormat(TextOutputFormat.class);
44         FileInputFormat.setInputPaths(conf, new Path(args[0]));
45         FileOutputFormat.setOutputPath(conf, new Path(args[1]));
46         JobClient.runJob(conf);
47     }
48 }
49 }
```

Bottom Status Bar: Description, Resource, Path, Location, Type, Writable, Smart Insert, 23:61:771, 140M of 256M

MapReducing through jar file:

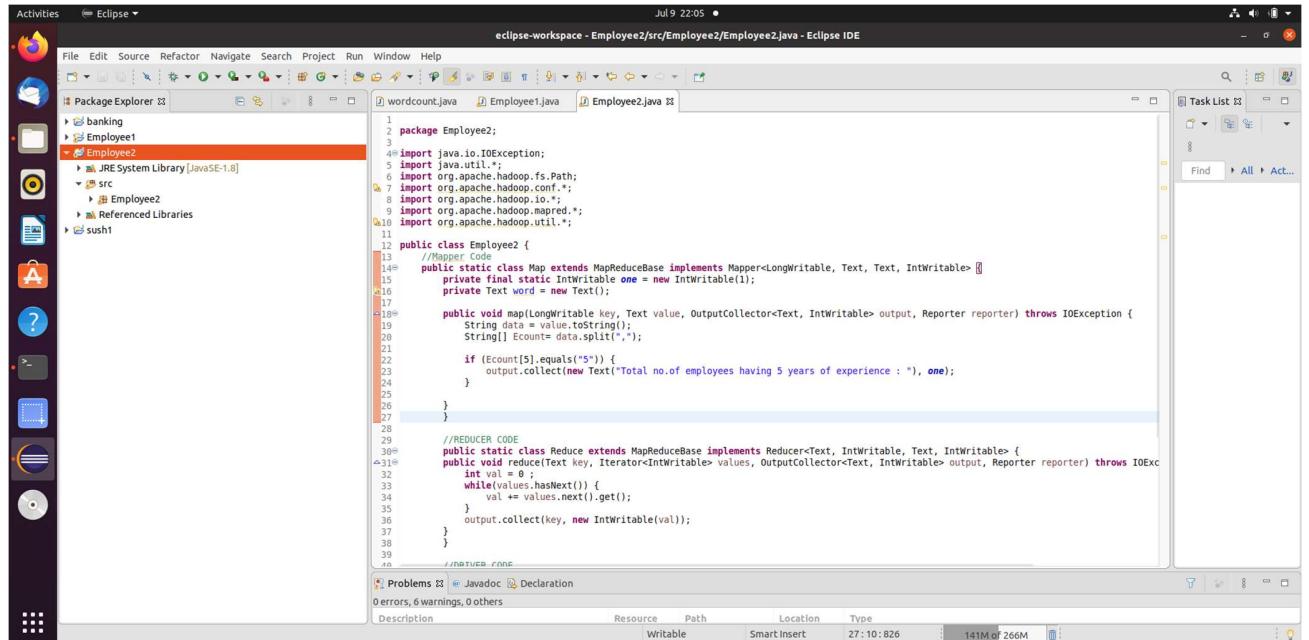
```
Activities Terminal Jul 9 22:00 hdoop@sushanth: ~/Desktop
hdoop@sushanth:~/Desktop$ hadoop fs -copyFromLocal sush.csv
2021-07-09 21:54:54,003 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
hdoop@sushanth:~/Desktop$ hadoop fs -ls
Found 11 items
-rw-r--r-- 1 hdoop supergroup 0 2021-05-08 17:16 empty.txt
-rw-r--r-- 1 hdoop supergroup 0 2021-05-05 10:37 file1.txt
-rw-r--r-- 1 hdoop supergroup 0 2021-05-05 10:37 file2.txt
-rw-r--r-- 1 hdoop supergroup 4 2021-05-08 17:35 new.txt
-rw-r--r-- 1 hdoop supergroup 0 2021-05-08 17:35 new1.txt
drwxr-xr-x 1 hdoop supergroup 0 2021-05-18 20:39 output
drwxr-xr-x 1 hdoop supergroup 0 2021-05-18 20:39 output
drwxr-xr-x 1 hdoop supergroup 0 2021-05-08 17:41 raju
drwxr-xr-x 1 hdoop supergroup 0 2021-05-08 16:57 sushanth
drwxr-xr-x 1 hdoop supergroup 69 2021-05-18 21:11 wc.txt
-rw-r--r-- 1 hdoop supergroup 0 2021-05-18 20:57 wordcount.txt
hdoop@sushanth:~/Desktop$ hadoop jar Employee.jar sush.csv Employee1.txt
JAR does not exist or is not a normal file: /home/hdoop/Desktop/Employee.jar
hdoop@sushanth:~/Desktop$ hadoop jar Employee.jar sush.csv Employee1.txt
JAR does not exist or is not a normal file: /home/hdoop/Desktop/Employee.jar
hdoop@sushanth:~/Desktop$ hadoop jar Employee.jar sush.csv Employee1.txt
Exception in thread "main" java.lang.ClassNotFoundException: sush.csv
        at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:418)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
        at java.lang.Class.forName(Class.java:348)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:316)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:236)
hdoop@sushanth:~/Desktop$ hadoop jar Employee.jar Employee1 sush.csv Employee1.txt
2021-07-09 21:58:37,586 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2021-07-09 21:58:38,111 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2021-07-09 21:58:38,544 INFO mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-07-09 21:58:38,544 INFO mapreduce.JobResourceUploader: Erasing any existing token for path: /tmp/hadoop-yarn/staging/hdoop/.staging/job_1625847246254_0001
2021-07-09 21:58:38,843 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-07-09 21:58:39,062 INFO mapred.FileInputFormat: Total input files to process : 1
2021-07-09 21:58:39,144 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-07-09 21:58:39,201 INFO mapreduce.JobSubmitter: number of splits:2
2021-07-09 21:58:39,666 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-07-09 21:58:39,821 INFO mapreduce.JobSubmitter: Submitting token for job: job_1625847246254_0001
2021-07-09 21:58:39,821 INFO mapreduce.JobSubmitter: Excluding failed tokens:
2021-07-09 21:58:40,154 INFO config.Configuration: resource-types.xml not found
2021-07-09 21:58:40,154 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-07-09 21:58:40,700 INFO impl.YarnClientImpl: Submitted application application_1625847246254_0001
2021-07-09 21:58:40,767 INFO mapreduce.Job: The url to track the job: http://sushanth:8088/proxy/application_1625847246254_0001/
2021-07-09 21:58:52,974 INFO mapreduce.Job: Job job_1625847246254_0001 running in uber mode : false
2021-07-09 21:58:52,975 INFO mapreduce.Job: Map 0% Reduce 0%
2021-07-09 21:59:00,086 INFO mapreduce.Job: Map 100% Reduce 0%
```

Output:

```
Activities Terminal Jul 9 22:00 hdoop@sushanth: ~/Desktop
hdoop@sushanth:~/Desktop$ 
Total vcore-milliseconds taken by all map tasks=8853
Total vcore-milliseconds taken by all reduce tasks=3433
Total megabyte-milliseconds taken by all map tasks=9665472
Total megabyte-milliseconds taken by all reduce tasks=3515392
Map-Reduce Framework
Map input records=20
Map output records=3
Map output bytes=95
Map output materialized bytes=126
Input split bytes=186
Combine input records=9
Combine output records=2
Reduce input groups=1
Reduce shuffle bytes=126
Reduce input records=2
Reduce output records=1
Spilled Records=1
Shuffled Maps =2
Failed Shuffles=0
Merged Map outputs=2
GC time elapsed (ms)=196
CPU time spent (ms)=650
Physical memory (bytes) snapshot=706595848
Virtual memory (bytes) snapshot=7605793056
Total committed heap usage (bytes)=597164032
Peak Map Physical memory (bytes)=297357312
Peak Map Virtual memory (bytes)=2533761024
Peak Reduce Physical memory (bytes)=192524288
Peak Reduce Virtual memory (bytes)=2538782720
Shuffle Errors=0
  IO=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=548
  File Output Format Counters
  Bytes Written=53
hdoop@sushanth:~/Desktop$ hadoop fs -ls Employee1.txt
Found 2 items
-rw-r--r-- 1 hdoop supergroup 0 2021-07-09 21:59 Employee1.txt/_SUCCESS
-rw-r--r-- 1 hdoop supergroup 53 2021-07-09 21:59 Employee1.txt/part-00000
hdoop@sushanth:~/Desktop$ hadoop fs -cat Employee1.txt/part-00000
2021-07-09 21:59:59,722 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Total no.of employees working in ISE Department : 9
hdoop@sushanth:~/Desktop$
```

2) Total number of employees with experience = 5 years.

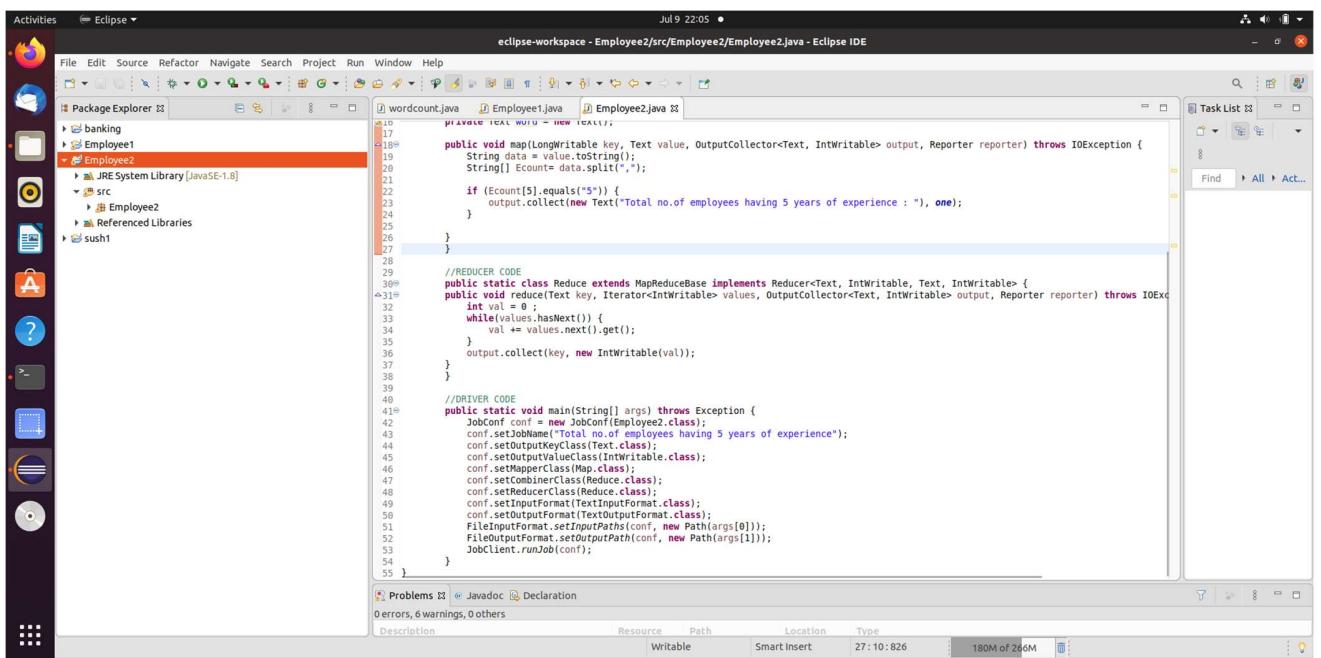
Snapshots of the java program:



The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Activities - Eclipse - eclipse-workspace - Employee2/src/Employee2/Employee2.java - Eclipse IDE
- File Menu:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard toolbar icons.
- Package Explorer:** Shows a project named "Employee2" containing "Employee1" and "Employee2". "Employee2" is selected.
- Code Editor:** Displays the `Employee2.java` file content. The code implements a MapReduce job to count employees with 5 years of experience.
- Task List:** Shows a list of tasks and their status.
- Bottom Status Bar:** Description, Resource, Path, Location, Type, Writable, Smart Insert, Date/Time, and Memory usage (141M of 266M).

```
1 package Employee2;
2
3 import java.io.IOException;
4 import java.util.*;
5 import org.apache.hadoop.fs.Path;
6 import org.apache.hadoop.conf.*;
7 import org.apache.hadoop.io.*;
8 import org.apache.hadoop.mapred.*;
9 import org.apache.hadoop.util.*;
10
11
12 public class Employee2 {
13     //Mapper Code
14     public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
15         private final static IntWritable one = new IntWritable(1);
16         private Text word = new Text();
17
18         public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
19             String data = value.toString();
20             String[] Ecount= data.split(",");
21
22             if (Ecount[5].equals("5")) {
23                 output.collect(new Text("Total no.of employees having 5 years of experience : "), one);
24             }
25         }
26     }
27
28
29     //REDUCER CODE
30     public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {
31         public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
32             int val = 0;
33             while(values.hasNext()) {
34                 val += values.next().get();
35             }
36             output.collect(key, new IntWritable(val));
37         }
38     }
39
40     //DRIVER CODE
41     public static void main(String[] args) throws Exception {
42         JobConf conf = new JobConf(Employee2.class);
43         conf.setJobName("Total no.of employees having 5 years of experience");
44         conf.setOutputKeyClass(Text.class);
45         conf.setOutputValueClass(IntWritable.class);
46         conf.setMapperClass(Map.class);
47         conf.setCombinerClass(Reduce.class);
48         conf.setReducerClass(Reduce.class);
49         conf.setInputFormat(TextInputFormat.class);
50         conf.setOutputFormat(TextOutputFormat.class);
51         FileInputFormat.setInputPaths(conf, new Path(args[0]));
52         FileOutputFormat.setOutputPath(conf, new Path(args[1]));
53         JobClient.runJob(conf);
54     }
55 }
```



The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Activities - Eclipse - eclipse-workspace - Employee2/src/Employee2/Employee2.java - Eclipse IDE
- File Menu:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard toolbar icons.
- Package Explorer:** Shows a project named "Employee2" containing "Employee1" and "Employee2". "Employee2" is selected.
- Code Editor:** Displays the `Employee2.java` file content. The code is identical to the previous screenshot but includes explanatory comments.
- Task List:** Shows a list of tasks and their status.
- Bottom Status Bar:** Description, Resource, Path, Location, Type, Writable, Smart Insert, Date/Time, and Memory usage (180M of 266M).

```
1 package Employee2;
2
3 import java.io.IOException;
4 import java.util.*;
5 import org.apache.hadoop.fs.Path;
6 import org.apache.hadoop.conf.*;
7 import org.apache.hadoop.io.*;
8 import org.apache.hadoop.mapred.*;
9 import org.apache.hadoop.util.*;
10
11
12 public class Employee2 {
13     //Mapper Code
14     public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
15         private final static IntWritable one = new IntWritable(1);
16         private Text word = new Text();
17
18         public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
19             String data = value.toString();
20             String[] Ecount= data.split(",");
21
22             if (Ecount[5].equals("5")) {
23                 output.collect(new Text("Total no.of employees having 5 years of experience : "), one);
24             }
25         }
26     }
27
28
29     //REDUCER CODE
30     public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {
31         public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
32             int val = 0;
33             while(values.hasNext()) {
34                 val += values.next().get();
35             }
36             output.collect(key, new IntWritable(val));
37         }
38     }
39
40     //DRIVER CODE
41     public static void main(String[] args) throws Exception {
42         JobConf conf = new JobConf(Employee2.class);
43         conf.setJobName("Total no.of employees having 5 years of experience");
44         conf.setOutputKeyClass(Text.class);
45         conf.setOutputValueClass(IntWritable.class);
46         conf.setMapperClass(Map.class);
47         conf.setCombinerClass(Reduce.class);
48         conf.setReducerClass(Reduce.class);
49         conf.setInputFormat(TextInputFormat.class);
50         conf.setOutputFormat(TextOutputFormat.class);
51         FileInputFormat.setInputPaths(conf, new Path(args[0]));
52         FileOutputFormat.setOutputPath(conf, new Path(args[1]));
53         JobClient.runJob(conf);
54     }
55 }
```

MapReducing through jar file:

```
Activities Terminal Jul 9 22:09 hdoop@sushanth: ~/Desktop
hdoop@sushanth:~/Desktop$ hadoop jar Employee2.jar Employee2 Employee2.txt Employee2.csv
2021-07-09 22:08:12,861 INFO mapreduce.JobResourceUploader:uploadArchives() - /tmp/hadoop-yarn/staging/job_1625847246254_0002
2021-07-09 22:08:13,949 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8082
2021-07-09 22:08:13,151 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-07-09 22:08:13,111 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-07-09 22:08:13,936 INFO mapred.FileInputFormat: Total input files to process : 1
2021-07-09 22:08:13,987 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-07-09 22:08:14,001 INFO mapred.FileInputFormat: Number of splits=1
2021-07-09 22:08:14,144 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-07-09 22:08:14,168 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1625847246254_0002
2021-07-09 22:08:14,168 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-07-09 22:08:14,399 INFO resource.ResourceUtils: failed to find resource-types.xml'.
2021-07-09 22:08:14,485 INFO org.apache.hadoop.mapred.YarnClientImpl: Untracked application application_1625847246254_0002
2021-07-09 22:08:14,500 INFO mapred.YarnClient: Job successfully submitted! To track the job: http://sushanth:8088/proxy/application_1625847246254_0002/
2021-07-09 22:08:14,537 INFO mapred.YarnClient: Running job: job_1625847246254_0002
2021-07-09 22:08:23,767 INFO mapreduce.Job: Job: job_1625847246254_0002 running in uber mode : false
2021-07-09 22:08:23,768 INFO mapreduce.Job: map 0% reduce 0%
2021-07-09 22:08:29,904 INFO mapreduce.Job: map 100% reduce 0%
2021-07-09 22:08:35,954 INFO mapreduce.Job: map 100% reduce 100%
2021-07-09 22:08:37,987 INFO mapreduce.Job: Job job_1625847246254_0002 completed successfully
2021-07-09 22:08:38,120 INFO mapreduce.Job: Counters: 55
File System Counters
  FILE: Number of bytes read=126
  FILE: Number of bytes written=677810
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=1134
  HDFS: Number of bytes written=57
  HDFS: Number of read operations=11
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2
  HDFS: Number of bytes read erasure-coded=0
Job Counters
  Killed map tasks=1
  Launched map tasks=2
  Launched reduce tasks=1
  Data locality requests=2
  Total time spent by all mappers in occupied slots (ms)=6913
  Total time spent by all reducers in occupied slots (ms)=4092
  Total time spent by all map tasks (ms)=6913
  Total time spent by all reduce tasks (ms)=4092
  Total vcore-milliseconds taken by all map tasks=6913
  Total vcore-milliseconds taken by all reduce tasks=4092
  Total megabyte-milliseconds taken by all map tasks=7078912
  Total megabyte-milliseconds taken by all reduce tasks=7078912
  Total minute-milliseconds taken by all reduce tasks=196788
```

Output:

```
Activities Terminal Jul 9 22:09 hdoop@sushanth: ~/Desktop
hdoop@sushanth:~/Desktop$ hadoop fs -ls Employee2.txt
Total vcore-milliseconds taken by all map tasks=6913
Total vcore-milliseconds taken by all reduce tasks=4092
Total megabyte-milliseconds taken by all map tasks=7078912
Total megabyte-milliseconds taken by all reduce tasks=4190208
Map-Reduce Framework
  Map input records=20
  Map output records=10
  Map input bytes=580
  Map output materialized bytes=132
  Input split bytes=186
  Combine input records=10
  Combine output records=2
  Reduce input groups=1
  Reduce shuffle bytes=132
  Reduce input records=2
  Reduce output records=1
  Spilled Records=4
  Shuffled Maps =2
  Failed Shuffles=0
  Merged Map outputs=2
  GC time elapsed (ms)=185
  CPU time spent (ms)=1840
  Physical memory (bytes) snapshot=72117888
  Virtual memory (bytes) snapshots=7609725216
  Total committed heap usage (bytes)=555220992
  Peak Map Physical memory (bytes)=295896320
  Peak Map Virtual memory (bytes)=2533576704
  Peak Reduce Physical memory (bytes)=189702144
  Peak Reduce Virtual memory (bytes)=2543026176
  Shuffle Errors:
    BLOB_ERRORS=0
    CONNECTION_ERRORS=0
    IO_ERRORS=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
  File Input Format Counters
    Bytes Read=948
  File Output Format Counters
    Bytes Written=57
hdoop@sushanth:~/Desktop$ hadoop fs -ls Employee2.txt
Found 2 items
-rw-r--r-- 1 hdoop supergroup 0 2021-07-09 22:08 Employee2.txt/_SUCCESS
-rw-r--r-- 1 hdoop supergroup 57 2021-07-09 22:08 Employee2.txt/part-00000
hdoop@sushanth:~/Desktop$ hadoop fs -cat Employee2.txt/part-00000
2021-07-09 22:09:27,728 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Total no.of employees having 5 years of experience : 18
hdoop@sushanth:~/Desktop$
```

3) Count the number of employees who lives in Bangalore.

Snapshots of the java program:

The screenshot shows the Eclipse IDE interface with the Employee3.java file open in the editor. The code implements a MapReduce job to count employees living in Bangalore. The Mapper part reads input lines and emits a key-value pair where the value is a string of comma-separated employee names. The Reducer part counts the occurrences of the 'Bangalore' key and outputs the total count. The Driver part sets up the JobConf and runs the job.

```
private static final IntWritable ONE = new IntWritable(1);
private static final Text TOTAL_TEXT = new Text("Total no.of employees who lives in Bangalore : ");

public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
    String data = value.toString();
    String[] dataCount = data.split(",");
    if (dataCount[3].equals("Bangalore")) {
        output.collect(TOTAL_TEXT, ONE);
    }
}

//REDUCER CODE
public static class Reduce 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 val = 0;
        while(values.hasNext()) {
            val += values.next().get();
        }
        output.collect(key, new IntWritable(val));
    }
}

//DRIVER CODE
public static void main(String[] args) throws Exception {
    JobConf conf = new JobConf(Employee3.class);
    conf.setJobName("Total no.of employees who lives in Bangalore");
    conf.setOutputKeyClass(Text.class);
    conf.setOutputValueClass(IntWritable.class);
    conf.setMapperClass(Mapper.class);
    conf.setCombinerClass(Reduce.class);
    conf.setReducerClass(Reduce.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);
}
```

This screenshot shows the same Eclipse IDE interface, but the code has been refactored. The imports for java.io, java.util, org.apache.hadoop, org.apache.hadoop.conf, org.apache.hadoop.mapred, and org.apache.hadoop.util have been moved to the top of the file. The code structure remains the same as in the previous screenshot.

```
package Employee3;
import java.io.IOException;
import java.util.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.util.*;

public class Employee3 {
    //Mapper Code
    public static class Map 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 data = value.toString();
            String[] dataCount = data.split(",");
            if (dataCount[3].equals("Bangalore")) {
                output.collect(TOTAL_TEXT, ONE);
            }
        }
    }

    //REDUCER CODE
    public static class Reduce 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 val = 0;
            while(values.hasNext()) {
                val += values.next().get();
            }
            output.collect(key, new IntWritable(val));
        }
    }

    //DRIVER CODE
    public static void main(String[] args) throws Exception {
        JobConf conf = new JobConf(Employee3.class);
        conf.setJobName("Total no.of employees who lives in Bangalore");
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);
        conf.setMapperClass(Map.class);
        conf.setCombinerClass(Reduce.class);
        conf.setReducerClass(Reduce.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);
    }
}
```

MapReducing through jar file:

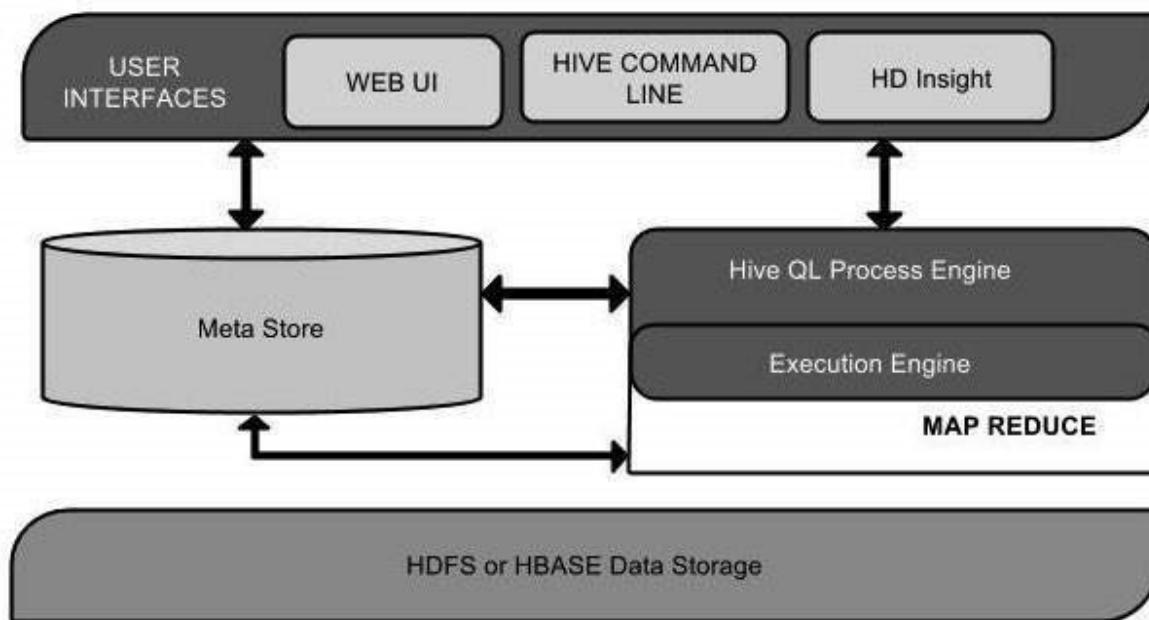
```
hadoop@ushanth:~/Desktop$ hadoop jar Employee3.jar Employee3 sush.csv Employee3.txt
2021-07-09 22:13:36.833 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2021-07-09 22:13:37.080 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2021-07-09 22:13:37.288 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1625847246254_0003
2021-07-09 22:13:37.432 INFO mapreduce.JobResourceUploader: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-07-09 22:13:37.433 INFO mapred.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-07-09 22:13:37.634 INFO mapred.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-07-09 22:13:37.733 INFO mapreduce.JobSubmitter: number of splits=2
2021-07-09 22:13:37.897 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1625847246254_0003
2021-07-09 22:13:37.897 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-07-09 22:13:38.108 INFO resource.ResourceUtil: Unable to find resource-types.xml'.
2021-07-09 22:13:38.192 INFO ampl.YarnClientImpl: The URL to track this job: http://ushanth:8088/proxy/application_1625847246254_0003/
2021-07-09 22:13:38.243 INFO mapreduce.Job: Job job_1625847246254_0003 running in uber mode : false
2021-07-09 22:13:44.375 INFO mapreduce.Job: map 0% reduce 0%
2021-07-09 22:13:56.524 INFO mapreduce.Job: map 100% reduce 100%
2021-07-09 22:13:57.551 INFO mapreduce.Job: Job job_1625847246254_0003 completed successfully
2021-07-09 22:13:57.694 INFO mapreduce.Job: Counters: 54
  File System Counters
    FILE: Number of bytes read=114
    FILE: Number of bytes written=677768
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=134
    HDFS: Number of bytes written=50
    HDFS: Number of read operations=11
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
    HDFS: Number of bytes read erasure-coded=0
  Job Counters
    Launched map tasks=2
    Launched reduce tasks=1
    Data-local map tasks=2
    Total time spent by all map tasks in occupied slots (ms)=7029
    Total time spent by all reduces in occupied slots (ms)=4093
    Total time spent by all map tasks (ms)=7029
    Total time spent by all reduce tasks (ms)=4093
    Total vcore-milliseconds taken by all map tasks=7029
    Total vcore-milliseconds taken by all reduce tasks=4093
    Total megabyte-milliseconds taken by all map tasks=7197696
    Total megabyte-milliseconds taken by all reduce tasks=4191232
Man-Reduce Framework
```

Output:

```
hadoop@ushanth:~/Desktop$ hadoop jar Employee3.jar Employee3 sush.csv Employee3.txt
Total vcore-milliseconds taken by all map tasks=7029
Total vcore-milliseconds taken by all reduce tasks=4093
Total megabyte-milliseconds taken by all map tasks=7197696
Total megabyte-milliseconds taken by all reduce tasks=4191232
Map-Reduce Framework
  Map input records=20
  Map output records=5
  Map output bytes=260
  Map output materialized bytes=120
  Input split bytes=180
  Combine output records=5
  Combine output bytes=120
  Reduce input groups=2
  Reduce input records=120
  Reduce shuffle bytes=120
  Reduce input records=2
  Reduce output records=1
  Spilled Records=4
  Shuffled Maps =2
  Failed Shuffles=0
  Merged Map outputs=2
  GC time elapsed (ms)=161
  CPU time spent (ms)=170
  Physical memory (bytes) snapshot=778539008
  Virtual memory (bytes) snapshot=7611756544
  Total committed heap usage (bytes)=603455488
  Peak Map Physical memory (bytes)=296480768
  Peak Map Virtual memory (bytes)=2535866368
  Peak Reduce Physical memory (bytes)=189841408
  Peak Reduce Virtual memory (bytes)=2542501888
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_TYPE=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=948
File Output Format Counters
  Bytes Written=50
hadoop@ushanth:~/Desktop$ hadoop fs -ls Employee3.txt
Found 2 items
-rw-r--r-- 1 hadoop supergroup 0 2021-07-09 22:13 Employee3.txt/_SUCCESS
-rw-r--r-- 1 hadoop supergroup 50 2021-07-09 22:13 Employee3.txt/part-00000
hadoop@ushanth:~/Desktop$ hadoop fs -cat Employee3.txt/part-00000
2021-07-09 22:14:30,815 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Total no.of employees who lives in Bangalore : 5
hadoop@ushanth:~/Desktop$
```

6. A BRIEF ABOUT HIVE

- Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop
- Hive is not
 - A relational database
 - A design for OnLine Transaction Processing (OLTP)
 - A language for real-time queries and row-level updates
- Features of Hive:
 - It stores schema in a database and processed data into HDFS.
 - It is designed for OLAP.
 - It provides SQL type language for querying called HiveQL or HQL.
 - It is familiar, fast, scalable, and extensible.
- Architecture of HIVE:



➤ User Interface

- Hive is a data warehouse infrastructure software that can create interaction between user and HDFS. The userinterfaces that Hive supports are Hive Web UI, Hive command line, and Hive HD Insight

➤ Meta Store

- Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table, their data types, and HDFS mapping.

➤ HiveQL Process Engine

- HiveQL is similar to SQL for querying on schema info on the Metastore. It is one of the replacements of traditional approach for MapReduce program. Instead of writing MapReduce program in Java, we can write a query for MapReduce job and process it.

➤ Execution Engine

- The conjunction part of HiveQL process Engine and MapReduce is Hive Execution Engine. Execution engine processes the query and generates results as same as MapReduce results. It uses the flavor of MapReduce.

➤ HDFS or HBASE

- Hadoop distributed file system or HBASE are the data storage techniques to store data into file system.

7. QUERIES OF THE USE-CASES

Exercise-II – HIVE:

Use the above dataset in .csv file and create a database called as EmployeeDB. Create a table under the database called as Employee using HIVEQL. The table fields are same, that is,

| Name | SSN | Salary | Address | Dname | Experience |
|--------|------|--------|-----------|-------|------------|
| Harsha | 5000 | 30000 | Bangalore | ISE | 5 |

Use the HiveQL language to perform the following Query based Map-reduce operations,

1. Insert 5 records using INSERT command.
2. Demonstrate the Alter command for the following cases,
 - a. Rename the table name to “Emp”.
 - b. Rename the column name “Dname” to “Dept_name”.
3. Retrieve all the employees who’s salary is not less than 50000.
4. Extract all employees who live in Bangalore but having less than 5 years of experience.
5. Create separate view containing Name, Dept_name of employees
6. Display Name and SSN and use group by SSN and order by Name
7. Retrieve Maximum salary, minimum salary and Average salary of the employees.
8. Create Another table called Department with the following fields (Dname = Dept_name and perform the following joins (outer, left outer, right outer) over Dname

| Dno | Dname |
|-----|-------|
| 6 | ISE |

8. HIVE QUERIES RESULTS AND SNAPSHOTS

- 1) Dataset Snapshot for the Query which is imported (LA2.csv):

Text Import - [sush.csv]

Import

Character set: Unicode (UTF-8)

Language: Default - English (India)

From row: 1

Separator Options

Fixed width Separated by

Tab Comma Semicolon Space Other String delimiter: "

Merge delimiters Trim spaces

Other Options

Format quoted field as text Detect special numbers

Fields

Column type:

| | Standard | Standard | Standard | Standard | Standard | Standard |
|----|----------|----------|----------|------------|----------|----------|
| 1 | Sushanth | 1001 | 30000 | Bangalore | ISE | 5 |
| 2 | Raju | 1002 | 20000 | Bangalore | ISE | 5 |
| 3 | Ravi | 1003 | 25000 | Mangalore | CSE | 5 |
| 4 | Alexa | 1004 | 80000 | Udupi | ISE | 6 |
| 5 | Siri | 1005 | 5000 | Bihar | ECE | 5 |
| 6 | Jack | 1006 | 2300 | Kerala | EEE | 4 |
| 7 | Pavan | 1007 | 9000 | Telangana | ISE | 7 |
| 8 | Hari | 1008 | 15000 | Ap | ECE | 6 |
| 9 | Raghav | 1009 | 23500 | Mangalore | MCE | 5 |
| 10 | Gagan | 1010 | 6000 | Bihar | CSE | 6 |
| 11 | Harshith | 1011 | 41000 | Tamil nadu | ISE | 7 |
| 12 | Ali | 1012 | 23650 | Bangalore | CSE | 5 |
| 13 | Gaurav | 1013 | 7800 | Tamil nadu | ISE | 6 |
| 14 | Anusha | 1014 | 10000 | Telangana | ISE | 5 |
| 15 | Namitha | 1015 | 82000 | Udupi | MCE | 5 |
| 16 | Reddy | 1016 | 60000 | Bihar | ECE | 5 |
| 17 | Suhail | 1017 | 65000 | Bangalore | ECE | 6 |
| 18 | Shreyas | 1018 | 32000 | Mangalore | ISE | 7 |
| 19 | Reethu | 1019 | 48000 | Kerala | CSE | 8 |
| 20 | Krithi | 1020 | 7000 | Bangalore | ISE | 5 |

Help Cancel OK

2) Creating database EmployeeDB and Creating table employee :

```
hadoop@sushanth:~/apache-hive-3.1.2-bin/conf$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop-3.2.1/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 2df1eb88-fd38-4e55-9acd-6573b37c4ceb

Logging initialized using configuration in jar:file:/home/hadoop/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true
Hive Session ID = 16796663-a12a-4b92-b427-aedcb4a0f1a
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> show databases;
OK
customer
default
sushanth
Time taken: 0.855 seconds, Fetched: 3 row(s)
hive> create database employeeDB;
OK
Time taken: 0.228 seconds
hive> use employeeDB;
OK
Time taken: 0.031 seconds
hive> create table employee(name string,ssn int, salary float, address string, dname string, experience int) row format delimited fields terminated by ",";
OK
Time taken: 2.196 seconds
hive> desc Employee;
OK
name          string
ssn           int
salary         float
address        string
dname          string
experience     int
Time taken: 1.766 seconds, Fetched: 6 row(s)
hive> ■
```

3) Importing csv file values into table employee.

```
hive> load data local inpath '/home/hadoop/Documents/sush.csv' into table employee;
Loading data to table employeeDB.employee
OK
Time taken: 1.213 seconds
hive> select* from employee;
OK
Sushanth  1001  30000.0 Bangalore    ISE   5
Raju      1002  20000.0 Bangalore    ISE   5
Ravi      1003  25000.0 Mangalore    CSE   5
Alexa     1004  80000.0 Udupi       ISE   6
Siri      1005  5000.0 Bihar        ECE   5
Jack      1006  2300.0 Kerala       EEE   4
Pavan     1007  9000.0 Telangana    ISE   7
Harl      1008  15000.0 Ap          ECE   6
Raghav    1009  23500.0 Mangalore    MCE   5
Gagan     1010  6000.0 Bihar        CSE   6
Harshith  1011  41000.0 Tamil nadu  ISE   7
Ali       1012  23650.0 Bangalore    CSE   5
Gaurav    1013  7880.0 Tamil nadu   ISE   6
Anusha    1014  10000.0 Telangana   ISE   5
Namitha   1015  82000.0 Udupi       MCE   5
Reddy     1016  60000.0 Bihar        ECE   5
Suhail    1017  65000.0 Bangalore    ECE   6
Shreyas   1018  32000.0 Mangalore    ISE   7
Reethu    1019  48000.0 Kerala       CSE   8
Krithi   1020  7000.0 Bangalore    ISE   5
Time taken: 2.967 seconds, Fetched: 20 row(s)
hive> ■
```

4) Insert 5 records using INSERT command.

```
hive> insert into employee values("Ram",1021,56000.0,"Mandya","ECE",4),("shyam",1022,59000.0,"Mandya","ECE",6),("Rinku",1023,8000.0,"Ap","ISE",6),("Anu",1024,86000.0,"Mangalore","ECE",5),("Krithi",1025,45000.0,"Kerala","ECE",6);
Query ID = hdoop_20210705161900_e3f5206e-4130-4df5-b06b-bd8d7bde40ec
Total Jobs = 1
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (<n> 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>
Starting Job = job_1625480980760_0001, Tracking URL = http://sushanth:8088/proxy/application_1625480980760_0001/
Kill Command = /home/hdoop/hadoop-3.2.1/bin/mrapp job -kill job_1625480980760_0001
Hadoop version: 3.2.1
MapReduce Stage 0 map = 0%, reduce = 0%
2021-07-05 16:19:59,383 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.22 sec
2021-07-05 16:20:29,203 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.5 sec
MapReduce Total cumulative CPU time: 5 seconds 500 msec
Ended Job = job_1625480980760_0001
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://127.0.0.1:9000/user/hive/warehouse/employeedb.db/employee/.hive-staging_hive_2021-07-05_16-19-00_459_4237719562725660576-1/-ext-10000
Loading data to table employeedb.employee
Mapreduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.5 sec HDFS Read: 22681 HDFS Write: 647 SUCCESS
Total Mapreduce CPU Time Spent: 5 seconds 500 msec
OK
Time taken: 94.761 seconds
hive> select* from employee;
OK
Ram 1021 56000.0 Mandya ECE 4
shyam 1022 59000.0 Mandya ECE 6
Rinku 1023 8000.0 Ap ISE 6
Anu 1024 86000.0 Mangalore ECE 5
Krithi 1025 45000.0 Kerala ECE 6
Sushanth 1026 30000.0 Bangalore ISE 5
Raju 1002 20000.0 Bangalore ISE 5
Ravi 1003 25000.0 Mangalore CSE 5
Alexa 1004 80000.0 Udupi ISE 6
Siri 1005 5000.0 Bihar ECE 5
Jack 1006 2300.0 Kerala EEE 4
Pavan 1007 9000.0 Telangana ISE 7
Hart 1008 15000.0 Ap ECE 6
Raghav 1009 23500.0 Mangalore MCE 5
Gagan 1010 6000.0 Bihar CSE 6
Harshith 1011 41000.0 Tamil nadu ISE 7
```

5) Demonstrate the Alter command for the following cases,

- Rename the table name to “Emp”.
- Rename the column name “Dname” to “Dept_name”.

```
hive> show tables;
OK
employee
Time taken: 0.036 seconds, Fetched: 1 row(s)
hive> alter table employee rename to Emp;
OK
Time taken: 0.22 seconds
hive> show tables;
OK
emp
Time taken: 0.036 seconds, Fetched: 1 row(s)
hive> 
```

```
hive> desc emp;
OK
name          string
ssn           int
salary         float
address        string
dname          string
experience     int
Time taken: 0.075 seconds, Fetched: 6 row(s)
hive> alter table emp change dname deptname string;
OK
Time taken: 0.425 seconds
hive> desc emp;
OK
name          string
ssn           int
salary         float
address        string
deptname       string
experience     int
Time taken: 0.075 seconds, Fetched: 6 row(s)
hive> 
```

6) Retrieve all the employees who's salary is not less than 50000.

```
hive> select name,ssn,salary from emp where salary>=50000;
OK
Ram    1021   56000.0
shyam  1022   59000.0
Anu    1024   86000.0
Alexa   1004   80000.0
Namitha 1015   82000.0
Reddy   1016   60000.0
Suhail  1017   65000.0
Time taken: 1.955 seconds, Fetched: 7 row(s)
hive>
```

7) Extract all employees who live in Bangalore but having less than 5years of experience.

```
hive> select name,address,experience from emp where address="Bangalore" and experience<5;
OK
Time taken: 1.385 seconds
hive> |
```

8) Create separate view containing Name, Dept_name of employees.

```
hive> create view Emp_Details as select name,deptname from emp;
OK
Time taken: 1.517 seconds
hive> select* from Emp_Details;
OK
Ram      ECE
shyam    ECE
Rinku   ISE
Anu     ECE
Krithi   ECE
Sushanth  ISE
Raju     ISE
Ravl     CSE
Alexa    ISE
Siri     ECE
Jack     EEE
Pavan    ISE
Hari     ECE
Raghav   MCE
Gagan    CSE
Harshith  ISE
Ali      CSE
Gaurav   ISE
Anusha   ISE
Namitha  MCE
Reddy    ECE
Suhail   ECE
Shreyas  ISE
Reethu   CSE
Krithi   ISE
Time taken: 1.768 seconds, Fetched: 25 row(s)
hive> |
```

9) Display Name and SSN and use group by SSN and order by Name.

```
hive> select name,ssn from emp group by name,ssn order by name;
Query ID = hdoop_20210705163908_71ec3db0-3c84-42b1-a3b4-a5b432057ac6
Total jobs = 2
Launching Job 1 out of 2
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>
Starting Job = job_1625480980760_0002, Tracking URL = http://sushanth:8088/proxy/application_1625480980760_0002/
Kill Command = /home/hdoop/hadoop-3.2.1/bin/mapred job -kill job_1625480980760_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-07-05 16:39:45,074 Stage-1 map = 0%, reduce = 0%
2021-07-05 16:40:06,234 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.2 sec
2021-07-05 16:40:06,413 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.67 sec
MapReduce Total cumulative CPU time: 4 seconds 670 msec
Ended Job = job_1625480980760_0002
Launching Job 2 out of 2
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>
Starting Job = job_1625480980760_0003, Tracking URL = http://sushanth:8088/proxy/application_1625480980760_0003/
Kill Command = /home/hdoop/hadoop-3.2.1/bin/mapred job -kill job_1625480980760_0003
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2021-07-05 16:40:23,195 Stage-2 map = 0%, reduce = 0%
2021-07-05 16:40:29,387 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.95 sec
2021-07-05 16:40:39,671 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 4.02 sec
MapReduce Total cumulative CPU time: 4 seconds 20 msec
Ended Job = job_1625480980760_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.67 sec HDFS Read: 12724 HDFS Write: 752 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 4.02 sec HDFS Read: 8162 HDFS Write: 668 SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 690 msec
OK
Alexa 1004
All 1012
Anu 1024
Anusha 1014
Gagan 1010
Gaurav 1013
Harl 1008
Varshith 1011
```

10) Retrieve Maximum salary, minimum salary and Average salary of the employees

```
hive> select max(salary),min(salary),avg(salary) from emp;
Query ID = hdoop_20210705164286_3ca96bb6-3169-4155-aebd-d053d06c0b6c
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>
Starting Job = job_1625480980760_0004, Tracking URL = http://sushanth:8088/proxy/application_1625480980760_0004/
Kill Command = /home/hdoop/hadoop-3.2.1/bin/mapred job -kill job_1625480980760_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-07-05 16:42:17,044 Stage-1 map = 0%, reduce = 0%
2021-07-05 16:42:22,216 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.04 sec
2021-07-05 16:42:34,609 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.34 sec
MapReduce Total cumulative CPU time: 5 seconds 340 msec
Ended Job = job_1625480980760_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.34 sec HDFS Read: 18256 HDFS Write: 122 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 340 msec
OK
86000.0 2300.0 33850.0
Time taken: 31.206 seconds, Fetched: 1 row(s)
hive>
```

- 11) Create Another table called Department and Inserting values.
(Dname = Deptname(employee))

```
hive> create table department(dno int,dname string)row format delimited fields terminated by " ";
OK
Time taken: 2.15 seconds
```

- 12) Performing outer join over Dname

```
hive> insert into department values(1,"ISE"),(2,"MCE"),(3,"CSE"),(4,"EEE"),(5,"ECE");
Query ID = hdoop_20210705164819_4b12000d-059b-4cc4-9d8b-a4d5fc52a0a7
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1625480980760_0005, Tracking URL = http://sushanth:8088/proxy/application_1625480980760_0005/
Kill Command = /home/hdoop/hadoop-3.2.1/bin/mrapp job -kill job_1625480980760_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-07-05 10:48:32,179 Stage-1 map = 0%,  reduce = 0%
2021-07-05 10:48:57,871 Stage-1 map = 67%,  reduce = 0%, Cumulative CPU 15.93 sec
2021-07-05 10:49:03,009 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 16.13 sec
2021-07-05 10:49:21,504 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 18.62 sec
MapReduce Total cumulative CPU time: 18 seconds 620 msec
Ended Job = job_1625480980760_0005
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://127.0.0.1:9000/user/hive/warehouse/employeedb.db/department/.hive-staging_hive_2021-07-05_10-48-19_820_8447158862064138451-1/-ext-10000
Loading data to table employeedb.department
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 18.62 sec  HDFS Read: 15804 HDFS Write: 310 SUCCESS
Total MapReduce CPU Time Spent: 18 seconds 620 msec
OK
Time taken: 71.251 seconds
hive> select* from department;
OK
1      ISE
2      MCE
3      CSE
4      EEE
5      ECE
Time taken: 0.682 seconds, Fetched: 5 row(s)
hive> ■
```

```
hive> select name,ssn,d.dname,dno from emp e full outer join department d on e.deptname=d.dname;
Query ID = hdoop_20210705165436_17bd72dc-bde1-4f07-a738-7a26ebab3e03
Total jobs = 1
Launching Job 1 out of 1
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>
Starting Job = job_1625480980760_0006, Tracking URL = http://sushanth:8088/proxy/application_1625480980760_0006/
Kill Command = /home/hdoop/hadoop-3.2.1/bin/mrapp job -kill job_1625480980760_0006
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2021-07-05 10:55:11,286 Stage-1 map = 0%,  reduce = 0%
2021-07-05 10:55:21,819 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 4.96 sec
2021-07-05 10:55:32,087 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 10.72 sec
2021-07-05 10:55:44,451 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 13.3 sec
MapReduce Total cumulative CPU time: 13 seconds 300 msec
Ended Job = job_1625480980760_0006
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2 Reduce: 1 Cumulative CPU: 13.3 sec  HDFS Read: 17669 HDFS Write: 818 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 300 msec
OK
Ravi    1003  CSE   3
Gagan   1010  CSE   3
Ali     1012  CSE   3
Reethu  1019  CSE   3
Ram     1021  ECE   5
Suhail  1017  ECE   5
Reddy   1016  ECE   5
Hari    1008  ECE   5
Siri    1005  ECE   5
Kirthi  1025  ECE   5
Anu     1024  ECE   5
Shyam   1022  ECE   5
Jack    1006  EEE   4
Rinku   1023  ISE   1
Pavan   1007  ISE   1
Krithi  1020  ISE   1
Alexa   1004  ISE   1
Shreyas 1018  ISE   1
Raju   1002  ISE   1
Sushanth 1001  ISE   1
Gaurav  1013  ISE   1
Anusha  1014  ISE   1
Harishk 1011  TCE   1
```

13) Performing on left outer join over Dname

```
hive> select name ssn,d.dname,dno from emp e left outer join department d on e.deptname=d.dname;
Query ID = hdoop_20210705165730_6fcc2b97-9ae9-4db9-a6af-17c1d775753c
Total jobs = 1
SLF4J: Found binding in [jar::file:/home/hadoop/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!org/slf4j/impl/StaticLoggerBinder.class]
2021-07-05 17:00:04      Starting to launch local task to process map join;      maximum memory = 239075328

2021-07-05 17:00:17      Uploaded 1 File to: file:/tmp/hive/java/hadoop/2df1eb88-fd30-4e55-9acd-6573b37c4ceb/hive_2021-07-05_16-57-30_543_8296294841225471837-1/-local-10004/HashTable-Stage-3/MapJoin-mapfil
e@--.hashtable (375 bytes)
2021-07-05 17:00:17      End of local task; Time Taken: 13.408 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1625480980760_0007, Tracking URL = http://sushanth:8088/proxy/application_1625480980760_0007/
Kill Command = /home/hadoop/hadoop-3.2.1/bin/mapred job -kill job_1625480980760_0007
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2021-07-05 17:00:46,088 Stage-3 map = 0%, reduce = 0%
2021-07-05 17:00:52,438 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 2.27 sec
MapReduce Total cumulative CPU time: 2 seconds 270 msec
Ended Job = job_1625480980760_0007
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1  Cumulative CPU: 2.27 sec  HDFS Read: 10091 HDFS Write: 693 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 270 msec
OK
Ram      ECE      5
shyam    ECE      5
Rinku    ISE      1
Anu      ECE      5
Krithi   ECE      5
Sushanth ISE      1
Raju     ISE      1
Ravi     CSE      3
Alexa    ISE      1
Siri     ECE      5
Jack     EEE      4
Pavan    ISE      1
Hari     ECE      5
Raghav   MCE      2
Gagan    CSE      3
Harshit  ISE      1
Ali      CSE      3
Gaurav   ISE      1
Anusha   ISE      1
Namitha  MCE      2
Reddy    ECE      5
Suhail   ECE      5
Shreyas  TCE      1
```

14) Performing on right outer join over Dname.

```
hive> select name,ssn,d.dname,dno from emp e right outer join department d on e.deptname = d.dname;
Query ID = hdoop_20210705170351_1b0d1ca5-435e-4c5c-9404-69d95c381ed5
Total jobs = 1

2021-07-05 17:03:57      Starting to launch local task to process map join;      maximum memory = 239075328
2021-07-05 17:03:59      Uploaded 1 File to: file:/tmp/hive/java/hadoop/2df1eb88-fd30-4e55-9acd-6573b37c4ceb/hive_2021-07-05_17-03-51_623_3353144385287407466-1/-local-10004/HashTable-Stage-3/MapJoin-mapfil
e@--.hashtable (794 bytes)
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1625480980760_0008, Tracking URL = http://sushanth:8088/proxy/application_1625480980760_0008/
Kill Command = /home/hadoop/hadoop-3.2.1/bin/mapred job -kill job_1625480980760_0008
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2021-07-05 17:04:17,125 Stage-3 map = 0%, reduce = 0%
2021-07-05 17:04:23,314 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 2.18 sec
MapReduce Total cumulative CPU time: 2 seconds 180 msec
Ended Job = job_1625480980760_0008
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1  Cumulative CPU: 2.18 sec  HDFS Read: 9031 HDFS Write: 818 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 180 msec
OK
Rinku    1023  ISE      1
Sushanth 1001  ISE      1
Raju     1002  ISE      1
Alexa    1004  ISE      1
Pavan    1007  ISE      1
Harshit  1011  ISE      1
Gaurav   1013  ISE      1
Anusha   1014  ISE      1
Shreyas  1018  ISE      1
Krithi   1020  ISE      1
Raghav   1009  MCE      2
Namitha  1015  MCE      2
Ravi     1003  CSE      3
Gagan    1010  CSE      3
Ali      1012  CSE      3
Reethu   1019  CSE      3
Jack     1006  EEE      4
Ram      1021  ECE      5
shyam    1022  ECE      5
Anu      1024  ECE      5
Krithi   1025  ECE      5
Siri     1005  ECE      5
Hari     1008  ECE      5
Reddy    1016  ECE      5
Suhail   1017  ECE      5
Time taken: 37.03 seconds, Fetched: 25 row(s)
```

GITHUB LINK:

<https://github.com/Sushanth-raju/BigData.git>

9. REFERENCES

https://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm

https://www.tutorialspoint.com/hive/hive_introduction.htm