

Exercise-I

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

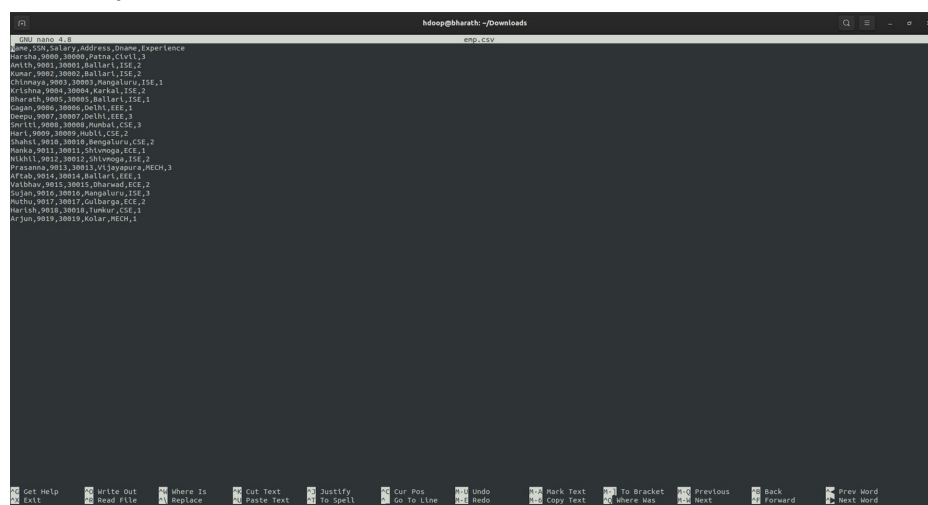
Use the Hadoop MapReduce programming framework to come up with a Program which will take the data from this .csv file and computes the following.

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

What is mapreduce?

MapReduce is a programming paradigm that enables massive scalability across hundreds or thousands of servers in a Hadoop cluster. The term "MapReduce" refers to two separate and distinct tasks that Hadoop programs perform. The first is the map job, which takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). The reduce job takes the output from a map as input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce job is always performed after the map job.

CSV file snap shot



```
data.csv
Name,SSN,Salary,Address,Dname,Experience
Harsha,5000,30000,Bangalore,ISE,5
Anith,5001,30001,Ballari,ISE,2
Kumar,5002,30002,Ballari,ISE,2
Chinmay,5003,30003,Mangalore,ISE,1
Arvindha,5004,30004,Ballari,ISE,2
Bharath,5005,30005,Ballari,ISE,1
Gagan,5006,30006,Delhi,EEE,1
Deepu,5007,30007,Delhi,ISE,3
Geri,5008,30008,Mumbai,CSE,3
Hari,5009,30009,Mumbai,CSE,2
Shakti,5010,30010,Bengaluru,CSE,2
Nisha,5011,30011,Mumbai,EEE,1
Nikhil,5012,30012,Mumbai,ISE,2
Prashant,5013,30013,Vijayapura,MECH,3
Azeab,5014,30014,Ballari,EEE,1
Nishay,5015,30015,Bharad,ISE,1
Sujan,5016,30016,Mangalore,ISE,3
Nisha,5017,30017,Gulbarga,EEE,2
Harish,5018,30018,Tumkur,CSE,1
Arjun,5019,30019,Kolar,MECH,1
```

github link: https://github.com/Bharath-k06/1nt18IS040_bharath_BigData.git

1) Total number of employees who work in ISE department

code:

```
package employee;
```

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

```
public class EmpISE {
    //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[] Ecount= data.split(",");
```

```
            if (Ecount[4].equals("ISE")) {
                output.collect(new Text("Total no.of employees working in ISE
Department : "), 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 { //{little: {1,1}}
```

```
        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(EmpISE.class);
        conf.setJobName("Total no.of employees working in ISE
Department");
        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);
    }
}

```

Output:

```

hadoop@bharath:~$ hadoop fs -ls
Found 5 items
-rw-r--r--  1 hadoop supergroup      680 2021-07-09 23:28 emp.csv
drwxr-xr-x  - hadoop supergroup       0 2021-07-10 00:40 output
drwxr-xr-x  - hadoop supergroup       0 2021-07-10 00:54 output2
drwxr-xr-x  - hadoop supergroup       0 2021-07-10 01:29 output3
drwxr-xr-x  - hadoop supergroup       0 2021-07-03 22:20 test
hadoop@bharath:~$ hadoop fs -ls output
Found 2 items
-rw-r--r--  1 hadoop supergroup       0 2021-07-10 00:40 output/_SUCCESS
-rw-r--r--  1 hadoop supergroup     53 2021-07-10 00:40 output/part-00000
hadoop@bharath:~$ hadoop fs -cat output/part-00000
2021-07-10 14:34:00,615 INFO sasl.SaslDataTransferClient: SASL encryption trust
check: localhostTrusted = false, remoteHostTrusted = false
Total no.of employees working in ISE Department :      7
hadoop@bharath:~$

```

Count the number of employees who lives in Bangalore/

package employee;

```

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

```

```

public class ExpAdd {

```

```

//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[] Ecount= data.split(",");

        if (Ecount[3].equals("Bengaluru")) {
            output.collect(new Text("Total no.of employees working
in Bengaluru : "), 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 { //{little: {1,1}}
        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(EmpExp.class);
    conf.setJobName("Total no.of employees working in
Bengaluru");

    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);
}
}

```

OutPut:

```
hadoop@bharath:~$ hadoop fs -cat output3/part-00000
2021-07-10 14:39:36,012 INFO sasl.SaslDataTransferClient: SASL encryption trust
check: localhostTrusted = false, remoteHostTrusted = false
Total no.of employees working in Bengaluru : 1
hadoop@bharath:~$
```

Total number of employees with experience=2 years

package employee;

import java.io.IOException;

import java.util.*;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.conf.*;

import org.apache.hadoop.io.*;

import org.apache.hadoop.mapred.*;

import org.apache.hadoop.util.*;

public class EmpExp {

 //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[] Ecount= data.split(",");

 if (Ecount[5].equals("2")) {

 output.collect(new Text("Total no.of employees having 5 years of
experience : "), 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 { //{little: {1,1}}

 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(EmpExp.class);
    conf.setJobName("Total no.of employees having 5 years of experience");
    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);
}
}

```

Output3:

```

hadoop@bharath:~$ hadoop fs -cat output2/part-00000
2021-07-10 14:35:53,022 INFO sasl.SaslDataTransferClient: SASL encryption trust
check: localhostTrusted = false, remoteHostTrusted = false
Total no.of employees having 2 years of experience :    8

```

Exercise-II

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.

```

hive> insert into employee values('Bharath',5001,55000,'Banglore','ISE',4),
> ('Nlkhll',5005,58000,'Lonvala','MECH',7),
> ('Harish',5004,48000,'Noida','CSE',6),
> ('Chinnaya',5003,32000,'Delhi','Aero',7),
> ('Anith',5002,90000,'Mumbai','CSE',10);
Query ID = hdoop_20210710082414_e992bd0e-c61c-4c18-9263-2af75b151733
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_1625885523926_0001, Tracking URL = http://bharath:8088/proxy/application_1625885523926_0001/
Kill Command = /home/hadoop/hadoop-3.2.1/bin/mapred job -kill job_1625885523926_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-07-10 08:24:26,473 Stage-1 map = 0%, reduce = 0%
2021-07-10 08:24:31,677 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.13 sec
2021-07-10 08:24:36,843 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.04 sec
MapReduce Total cumulative CPU time: 5 seconds 40 msec
Ended Job = job_1625885523926_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/employee.db/employee/.hive-staging_hive_2021-07-10_08-24-14_621_332290471210862080-1/-ext-10000
Loading data to table employee.db.employee
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.04 sec HDFS Read: 20970 HDFS Write: 673 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 40 msec
OK
Time taken: 25.59 seconds
hive> select * from employee;
OK
Harsha 5000 30000 Banglore ISE 5
Bharath 5001 55000 Banglore ISE 4
Nlkhll 5005 58000 Lonvala MECH 7
Harish 5004 48000 Noida CSE 6
Chinnaya 5003 32000 Delhi Aero 7
Anith 5002 90000 Mumbai CSE 10
Time taken: 0.154 seconds, Fetched: 6 row(s)
hive>

```

- 2) Demonstrate the Alter command for the following cases,
 Rename the table name to “Emp”.
 Rename the column name “Dname” to “Dept_name”.

OutPut:

```

hive> show tables;
OK
employee
Time taken: 0.035 seconds, Fetched: 1 row(s)
hive> alter table employee rename to Emp;
OK
Time taken: 0.369 seconds
hive> show tables;
OK
emp
Time taken: 0.026 seconds, Fetched: 1 row(s)
hive> alter table emp change dname dept_name string;
OK
Time taken: 0.284 seconds
hive> desc emp;
OK
name                string
ssn                  int
salary              int
address              string
dept_name            string
experience            int
Time taken: 0.048 seconds, Fetched: 6 row(s)
hive>

```

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

```
hive> select * from emp where salary>=50000;
OK
Bharath 5001      55000    Bangalore      ISE      4
Nikhil  5005      58000    Lonvala MECH   7
Amith   5002      90000    Mumbai CSE    10
Time taken: 0.249 seconds, Fetched: 3 row(s)
hive> □
```

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

```
hive> select * from emp where address= 'Bangalore' and experience <5;
OK
Bharath 5001      55000    Bangalore      ISE      4
Time taken: 0.179 seconds, Fetched: 1 row(s)
hive> □
```

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

```
hive> create view details as (select name,dept_name from emp);
OK
Time taken: 0.323 seconds
hive> select * from details;
OK
Harsha    ISE
Bharath   ISE
Nikhil    MECH
Harish    CSE
Chinmaya      Area
Amith      CSE
Time taken: 0.179 seconds, Fetched: 6 row(s)
hive> □
```

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


```
Activities Terminal Jul 10 08:44
hadoop@bharath: ~/apache-hive-3.12-bin/conf

hive> select name,ssn from emp group by ssn order by name;
FAILED: SemanticException [Error 10025]: Line 1:7 Expression not in GROUP BY key 'name'
hive> select name,ssn from emp group by ssn,name order by name;
Query ID = hdoop_20210710084359_fcd97d5d-1df9-4a2e-bd69-22d74e8c0c13
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_1625885523926_0002, Tracking URL = http://bharath:8088/proxy/application_1625885523926_0002/
Kill Command = /home/hadoop/hadoop-3.2.1/bin/mapred job -kill job_1625885523926_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-07-10 08:44:02,074 Stage-1 map = 0%, reduce = 0%
2021-07-10 08:44:08,035 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.05 sec
2021-07-10 08:44:13,182 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.11 sec
MapReduce Total cumulative CPU time: 4 seconds 110 msec
Ended Job = job_1625885523926_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_1625885523926_0003, Tracking URL = http://bharath:8088/proxy/application_1625885523926_0003/
Kill Command = /home/hadoop/hadoop-3.2.1/bin/mapred job -kill job_1625885523926_0003
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2021-07-10 08:44:20,081 Stage-2 map = 0%, reduce = 0%
2021-07-10 08:44:30,205 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.51 sec
2021-07-10 08:44:36,376 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 3.57 sec
MapReduce Total cumulative CPU time: 3 seconds 570 msec
Ended Job = job_1625885523926_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.11 sec HDFS Read: 12071 HDFS Write: 260 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 3.57 sec HDFS Read: 7670 HDFS Write: 233 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 680 msec
OK
name      ssn
-----
Anith      5002
Bharath    5001
Chimaya    5003
Harish     5004
Harsha     5000
Nikhil     5005
Time taken: 44.485 seconds, Fetched: 6 row(s)
hive>
```

7.)Retrieve Maximum salary, minimum salary and Average salary of the employees .

```
hive> select min(salary) ,max(salary) , avg(salary) from emp;
Query ID = hdoop_20210710084744_2bfe5db8-43eb-4ab9-bbb0-b3a8683ff005
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_1625885523926_0005, Tracking URL = http://bharath:8088/proxy/application_1625885523926_0005/
Kill Command = /home/hadoop/hadoop-3.2.1/bin/mapred job -kill job_1625885523926_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-07-10 08:47:51,887 Stage-1 map = 0%, reduce = 0%
2021-07-10 08:47:57,040 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.06 sec
2021-07-10 08:48:02,191 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.19 sec
MapReduce Total cumulative CPU time: 5 seconds 190 msec
Ended Job = job_1625885523926_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.19 sec HDFS Read: 17509 HDFS Write: 130 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 190 msec
OK
30000      90000      52166.666666666664
Time taken: 20.211 seconds, Fetched: 1 row(s)
hive>
```

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

```

hive> create table department (dno int , dname string);
OK
Time taken: 0.156 seconds
hive> show tables;
OK
department
details
emp
Time taken: 0.165 seconds, Fetched: 3 row(s)
hive> desc department;
OK
dno              int
dname            string
Time taken: 0.045 seconds, Fetched: 2 row(s)
hive> insert into table department values(6,'ISE'),(5,'CSE'),(7,'Arco');
Query ID = hdoop_20210710085042_f1ef743d-9380-40e9-8a5a-d451a503a09b
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.reducers=<number>
Starting Job = job_1625885523926_0006, Tracking URL = http://bharath:8088/proxy/application_1625885523926_0006/
Kill Command = /home/hadoop/hadoop-3.2.1/bin/mapred job -kill job_1625885523926_0006
Hadoop job information for Stage 1: number of mappers: 1; number of reducers: 1
2021-07-10 08:50:49,935 Stage-1 map = 0%, reduce = 0%
2021-07-10 08:50:56,123 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.06 sec
2021-07-10 08:51:07,290 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.16 sec
MapReduce Total cumulative CPU time: 5 seconds 160 msec
Ended Job = job_1625885523926_0006
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://127.0.0.1:5000/user/hive/warehouse/employee.db/department/.hive-staging_hive_2021-07-10_08-50-42_052_5980947521624981520-1/-ext-10000
Loading data to table employee.db.department
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.16 sec HDFS Read: 15656 HDFS Write: 298 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 160 msec
OK
Time taken: 21.762 seconds
hive>

```

```

hive> select * from emp left outer join department where dname=dept_name;
Query ID = hdoop_20210710085608_fe6e9564-95f0-4b8c-bc30-f74f054bfe6c
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]
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]

2021-07-10 08:56:18 End of local task; Time Taken: 1.473 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_1625885523926_0007, Tracking URL = http://bharath:8088/proxy/application_1625885523926_0007/
Kill Command = /home/hadoop/hadoop-3.2.1/bin/mapred job -kill job_1625885523926_0007
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2021-07-10 08:56:29,397 Stage-3 map = 0%, reduce = 0%
2021-07-10 08:56:35,585 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 3.43 sec
MapReduce Total cumulative CPU time: 3 seconds 430 msec
Ended Job = job_1625885523926_0007
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1 Cumulative CPU: 3.43 sec HDFS Read: 16551 HDFS Write: 339 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 430 msec
OK
Harsha 5000 30000 Bangalore ISE 5 6 ISE
Bharath 5001 55000 Bangalore ISE 4 6 ISE
Harish 5004 48000 Noida CSE 6 5 CSE
Chinmaya 5003 32000 Delhi Arco 7 7 Arco
Amlth 5002 90000 Mumbai CSE 10 5 CSE
Time taken: 30.021 seconds, Fetched: 5 row(s)
hive>

```

```

hive> select * from emp right outer join department where dname=dept_name;
Query ID = hdoop_20210710085752_f094d4f6-d2cb-4fcc-bids-ajad27e427af
Total jobs = 1
SLF4J: Found binding in [jar:file:/home/hadoop/apache-hive-3.12.2-bin/lib/log4j-slf4j-impl-2.18.0.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]
2021-07-10 08:57:55 Starting to launch local task to process map join; maximum memory = 2380763282021-07-10 08:57:59 Dump the side-table for tag: 1 with group count: 3 into file: file:/tmp/hive/java/hadoop/44bc7b2c-77c2-4312-8ea5-ed640f89a58e/hive_2021-07-10_08-57-52_910_25770029c2321950760-1/-local-10004/HashTable-Stage-3/MapJoin-mapfile11--.hashtable
2021-07-10 08:57:59 End of local task; Time Taken: 1.492 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_1625885523926_0009, Tracking URL = http://bharath:8888/proxy/application_1625885523926_0009/
Kill Command = /home/hadoop/hadoop-3.2.1/bin/mapred job -kill job_1625885523926_0009
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2021-07-10 08:58:08,730 Stage-3 map = 0%, reduce = 0%
2021-07-10 08:58:11,961 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 3.09 sec
MapReduce Total cumulative CPU time: 3 seconds 90 msec
Ended Job = job_1625885523926_0009
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1 Cumulative CPU: 3.09 sec HDFS Read: 10551 HDFS Write: 339 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 90 msec
OK
harsha 5000 30000 Bangalore ISE 5 6 ISE
bharath 5001 55000 Bangalore ISE 4 0 ISE
harish 5004 48000 Noida CSE 6 5 CSE
chinmaya 5003 32000 Delhi Area 7 7 Area
amth 5002 90000 Mumbai CSE 10 5 CSE
Time taken: 24.202 seconds, Fetched: 5 row(s)
hive>

```

```

hive> select * from emp full outer join department where dname=dept_name;
Query ID = hdoop_20210710085910_8fc42735-c07b-4067-89db-e41923d65cec
Total jobs = 1
SLF4J: Found binding in [jar:file:/home/hadoop/apache-hive-3.12.2-bin/lib/log4j-slf4j-impl-2.18.0.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: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2021-07-10 08:59:17 Dump the side-table for tag: 1 with group count: 3 into file: file:/tmp/hive/java/hadoop/44bd7b2c-77c2-4312-8ea5-ed640f89a58e/hive_2021-07-10_08-59-10_843_5714866211662444303-1/-local-10004/HashTable-Stage-3/MapJoin-mapfile12--.hashtable
2021-07-10 08:59:17 End of local task; Time Taken: 1.368 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_1625885523926_0009, Tracking URL = http://bharath:8888/proxy/application_1625885523926_0009/
Kill Command = /home/hadoop/hadoop-3.2.1/bin/mapred job -kill job_1625885523926_0009
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2021-07-10 08:59:27,244 Stage-3 map = 0%, reduce = 0%
2021-07-10 08:59:32,414 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 3.02 sec
MapReduce Total cumulative CPU time: 3 seconds 20 msec
Ended Job = job_1625885523926_0009
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1 Cumulative CPU: 3.02 sec HDFS Read: 10564 HDFS Write: 339 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 20 msec
OK
harsha 5000 30000 Bangalore ISE 5 0 ISE
bharath 5001 55000 Bangalore ISE 4 6 ISE
harish 5004 48000 Noida CSE 6 5 CSE
chinmaya 5003 32000 Delhi Area 7 7 Area
amth 5002 90000 Mumbai CSE 10 5 CSE
Time taken: 23.735 seconds, Fetched: 5 row(s)
hive>

```