ACADEMIC YEAR 2021-2022

BIGDATA LABORATORY

Report on,

Learning Activity II-Programming Assignment

Submitted by,

Chinmai Srivastava (1NT18IS046)

submitted to,

Ms. 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 CONTENTS

✞ What is Hadoop ??

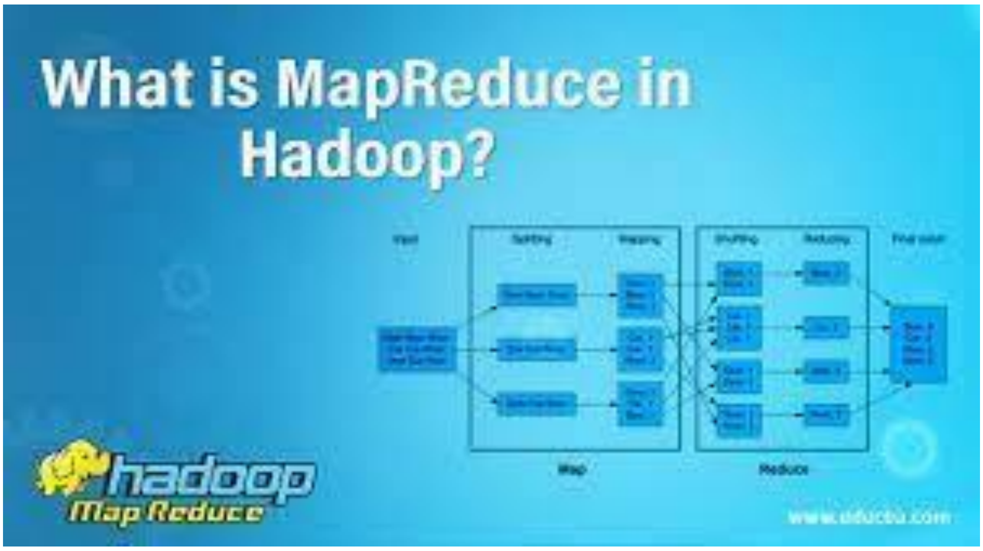
Hadoop is an open source framework from Apache and is used to store process and analyze data which are very huge in volume. Hadoop is written in Java and is not OLAP (online analytical processing).

It is used for batch/offline processing.It is being used by Facebook, Yahoo, Google, Twitter, LinkedIn and many more. Moreover it can be scaled up just by adding nodes in the cluster.

✞ Modules of Hadoop

1. **HDFS:** Hadoop Distributed File System. Google published its paper GFS and on the basis of that HDFS was developed. It states that the files will be broken into blocks and stored in nodes over the distributed architecture.
2. **Yarn:** Yet another Resource Negotiator is used for job scheduling and manage the cluster.
3. **Map Reduce:** This is a framework which helps Java programs to do the parallel computation on data using key value pair. The Map task takes input data and converts it into a data set which can be computed in Key value pair. The output of Map task is consumed by reduce task and then the out of reducer gives the desired result.
4. **Hadoop Common:** These Java libraries are used to start Hadoop and are used by other Hadoop modules.

✞ What is MapReduce ??



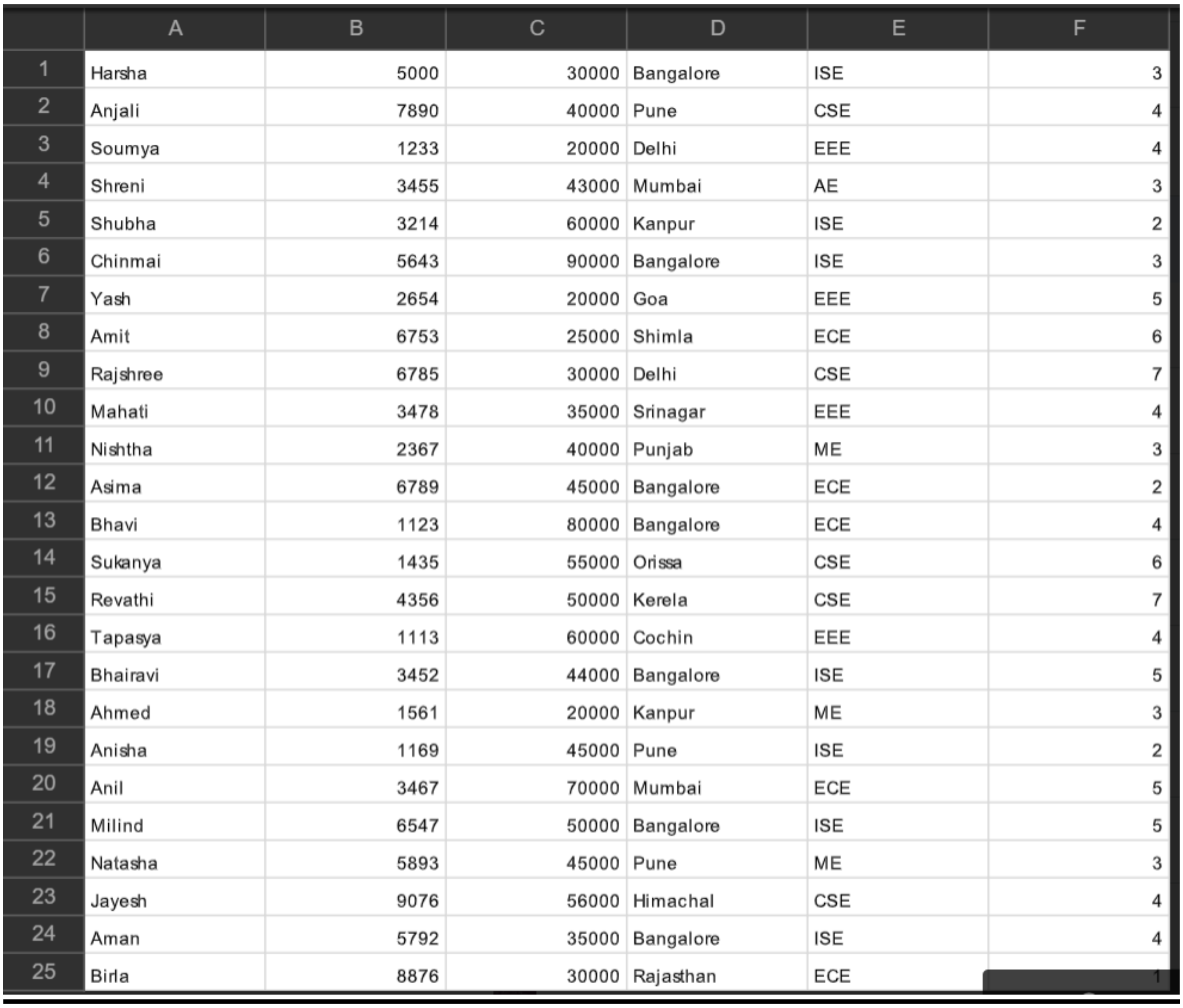
- **MapReduce** is a programming paradigm that enables massive scalability across hundreds or thousands of servers in a Hadoop cluster. As the processing component, MapReduce is the heart of **Apache Hadoop.** The term "MapReduce" refers to two separate and distinct tasks that Hadoop programs perform. 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 implies, the reduce job is always performed after the map job.

- **MapReduce programming offers several benefits to help you gain valuable insights from your big data:**

- **Scalability**. Businesses can process petabytes of data stored in the Hadoop Distributed File System (HDFS).
- **Flexibility**. Hadoop enables easier access to multiple sources of data and multiple types of data.
- **Speed**. With parallel processing and minimal data movement, Hadoop offers fast processing of massive amounts of data.
- **Simple**. Developers can write code in a choice of languages, including Java, C++ and Python.

Dataset



	A	B	C	D	E	F
1	Harsha	5000	30000	Bangalore	ISE	3
2	Anjali	7890	40000	Pune	CSE	4
3	Soumya	1233	20000	Delhi	EEE	4
4	Shreni	3455	43000	Mumbai	AE	3
5	Shubha	3214	60000	Kanpur	ISE	2
6	Chinmai	5643	90000	Bangalore	ISE	3
7	Yash	2654	20000	Goa	EEE	5
8	Amit	6753	25000	Shimla	ECE	6
9	Rajshree	6785	30000	Delhi	CSE	7
10	Mahati	3478	35000	Srinagar	EEE	4
11	Nishtha	2367	40000	Punjab	ME	3
12	Asima	6789	45000	Bangalore	ECE	2
13	Bhavi	1123	80000	Bangalore	ECE	4
14	Sukanya	1435	55000	Orissa	CSE	6
15	Revathi	4356	50000	Kerela	CSE	7
16	Tapasya	1113	60000	Cochin	EEE	4
17	Bhairavi	3452	44000	Bangalore	ISE	5
18	Ahmed	1561	20000	Kanpur	ME	3
19	Anisha	1169	45000	Pune	ISE	2
20	Anil	3467	70000	Mumbai	ECE	5
21	Milind	6547	50000	Bangalore	ISE	5
22	Natasha	5893	45000	Pune	ME	3
23	Jayesh	9076	56000	Himachal	CSE	4
24	Aman	5792	35000	Bangalore	ISE	4
25	Birla	8876	30000	Rajasthan	ECE	1

Programming Exercise

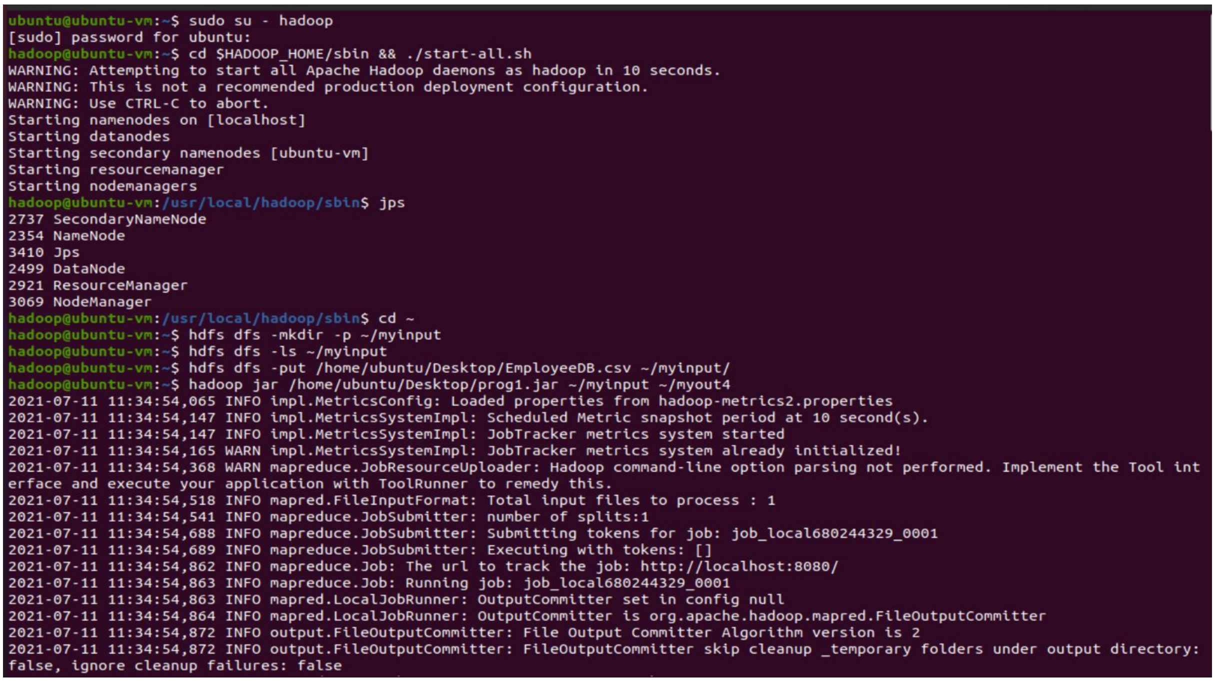
Exercise-I

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

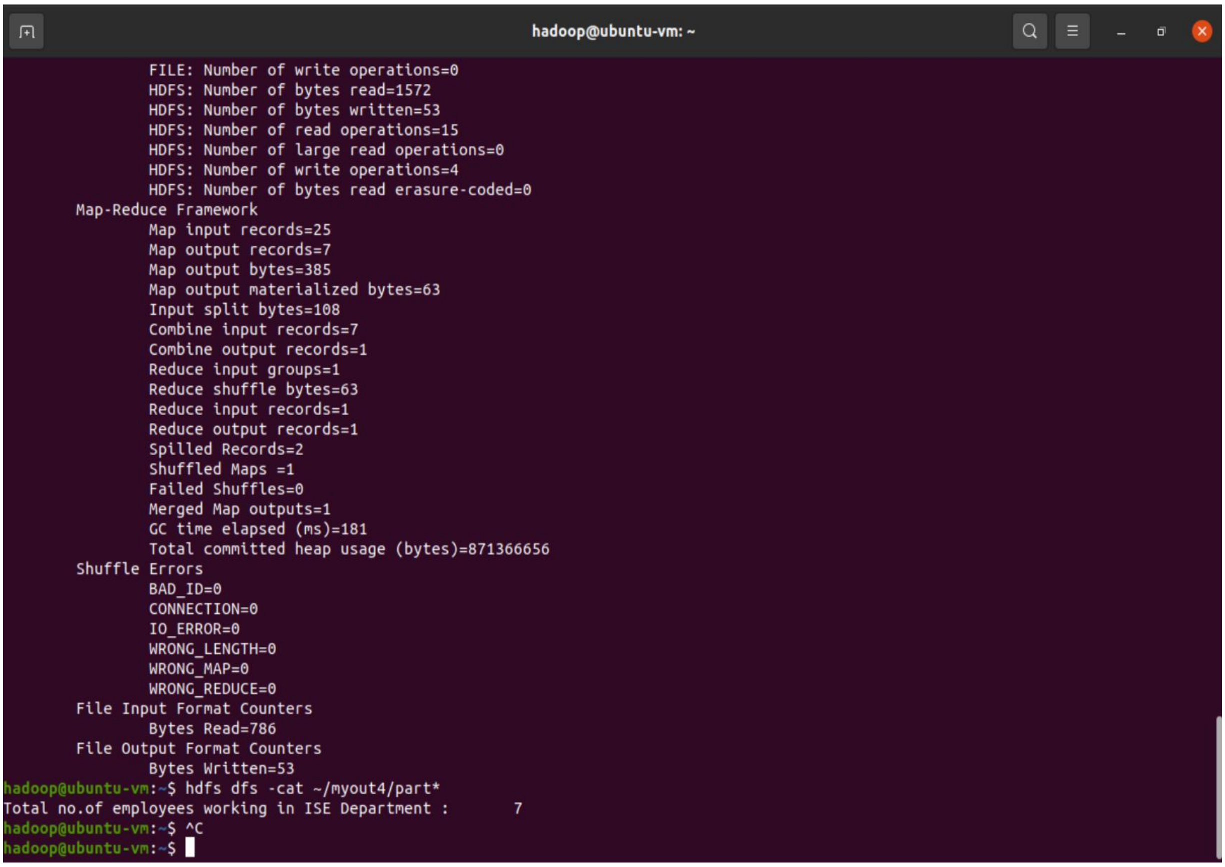
Name	SSN	Salary	Address	Dname	Experience
Harsha	5000	30000	Bangalore	ISE	5

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

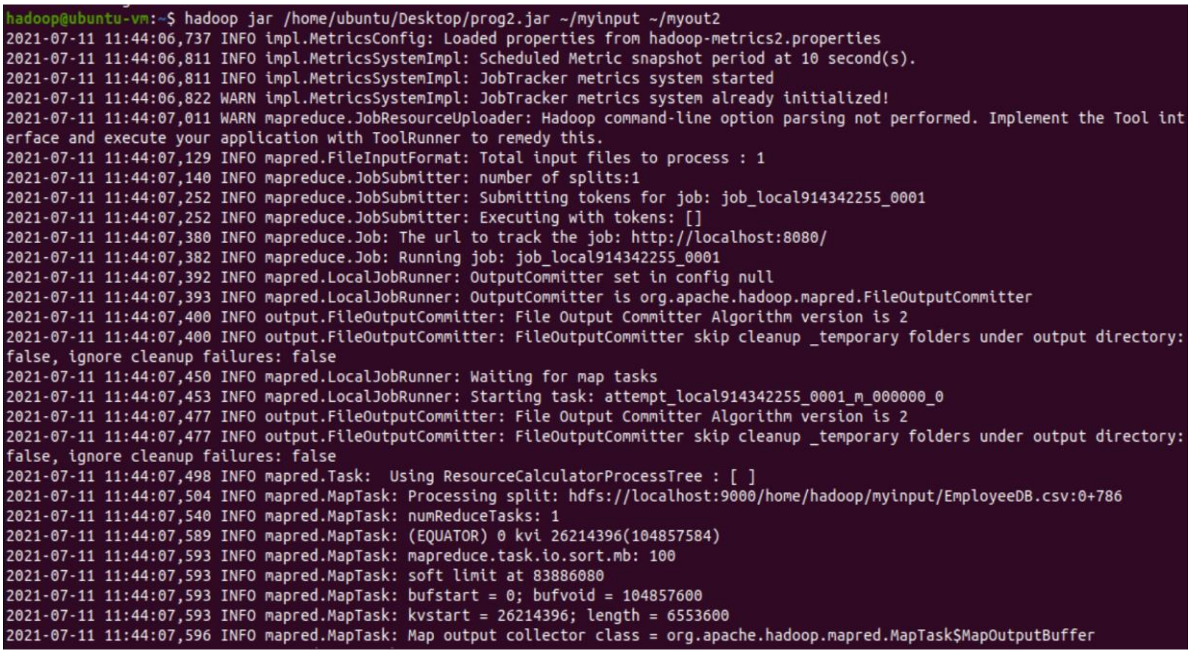


```
ubuntu@ubuntu-vm:~$ sudo su - hadoop
[sudo] password for ubuntu:
hadoop@ubuntu-vm:~$ cd $HADOOP_HOME/sbin && ./start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu-vm]
Starting resourcemanager
Starting nodemanagers
hadoop@ubuntu-vm:/usr/local/hadoop/sbin$ jps
2737 SecondaryNameNode
2354 NameNode
3410 Jps
2499 DataNode
2921 ResourceManager
3069 NodeManager
hadoop@ubuntu-vm:/usr/local/hadoop/sbin$ cd ~
hadoop@ubuntu-vm:~$ hdfs dfs -mkdir -p ~/myinput
hadoop@ubuntu-vm:~$ hdfs dfs -ls ~/myinput
hadoop@ubuntu-vm:~$ hdfs dfs -put /home/ubuntu/Desktop/EmployeeDB.csv ~/myinput/
hadoop@ubuntu-vm:~$ hadoop jar /home/ubuntu/Desktop/prog1.jar ~/myinput ~/myout4
2021-07-11 11:34:54,065 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2021-07-11 11:34:54,147 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2021-07-11 11:34:54,147 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2021-07-11 11:34:54,165 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2021-07-11 11:34:54,368 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool int
erface and execute your application with ToolRunner to remedy this.
2021-07-11 11:34:54,518 INFO mapred.FileInputFormat: Total input files to process : 1
2021-07-11 11:34:54,541 INFO mapreduce.JobSubmitter: number of splits:1
2021-07-11 11:34:54,688 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local680244329_0001
2021-07-11 11:34:54,689 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-07-11 11:34:54,862 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2021-07-11 11:34:54,863 INFO mapreduce.Job: Running job: job_local680244329_0001
2021-07-11 11:34:54,863 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2021-07-11 11:34:54,864 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapred.FileOutputCommitter
2021-07-11 11:34:54,872 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2021-07-11 11:34:54,872 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:
false, ignore cleanup failures: false
```

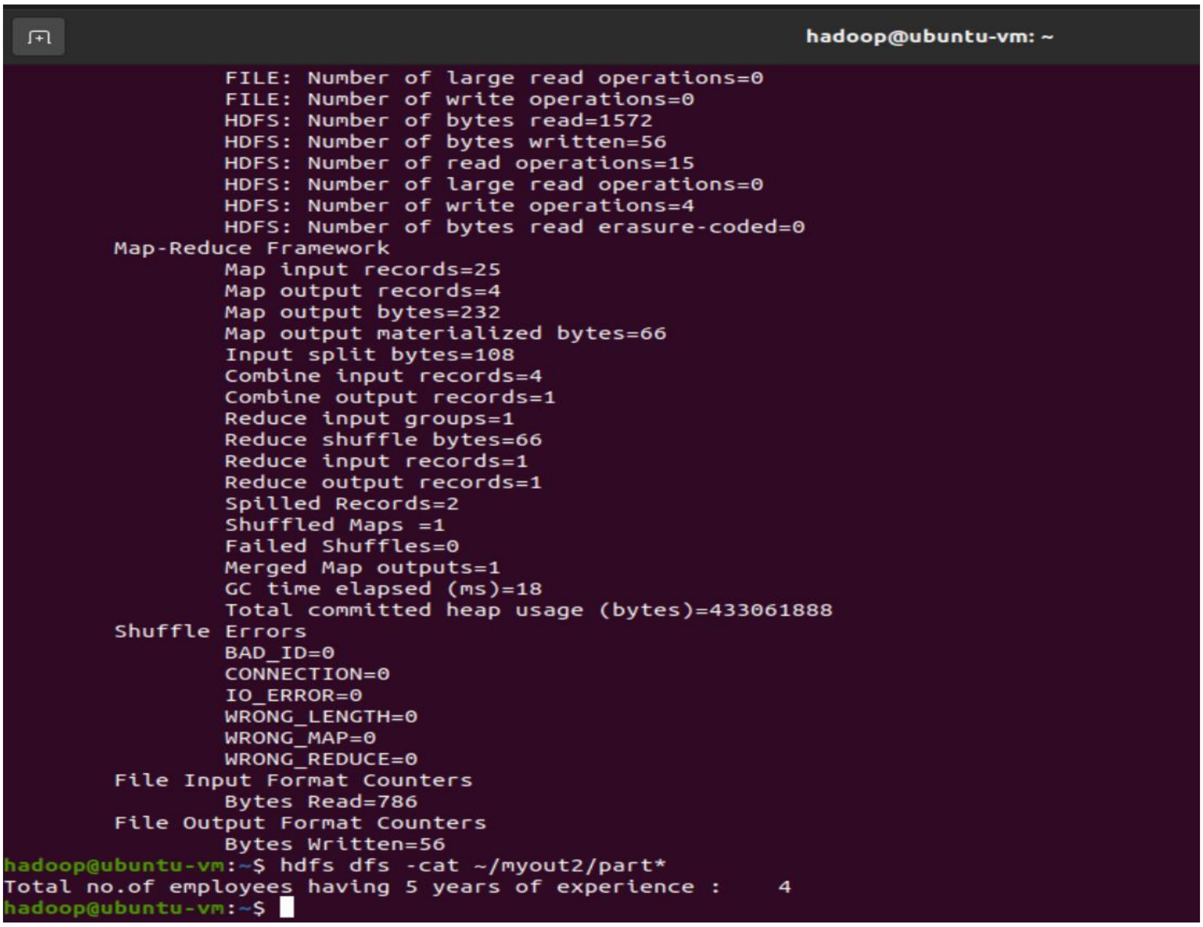


```
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1572
                HDFS: Number of bytes written=53
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=25
                Map output records=7
                Map output bytes=385
                Map output materialized bytes=63
                Input split bytes=108
                Combine input records=7
                Combine output records=1
                Reduce input groups=1
                Reduce shuffle bytes=63
                Reduce input records=1
                Reduce output records=1
                Spilled Records=2
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=181
                Total committed heap usage (bytes)=871366656
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=786
        File Output Format Counters
                Bytes Written=53
hadoop@ubuntu-vm:~$ hdfs dfs -cat ~/myout4/part*
Total no.of employees working in ISE Department :       7
hadoop@ubuntu-vm:~$ ^C
hadoop@ubuntu-vm:~$ 
```

2. Total number of employees with experience=5 years

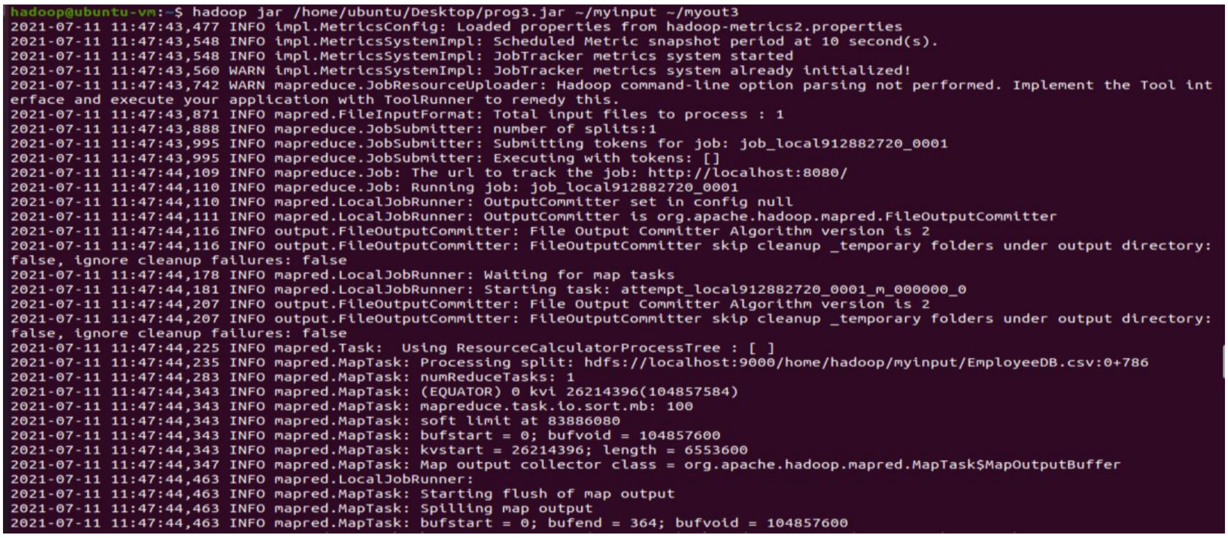


```
hadoop@ubuntu-vm:~$ hadoop jar /home/ubuntu/Desktop/prog2.jar ~/myinput ~/myout2
2021-07-11 11:44:06,737 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2021-07-11 11:44:06,811 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2021-07-11 11:44:06,811 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2021-07-11 11:44:06,822 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2021-07-11 11:44:07,011 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool int
erface and execute your application with ToolRunner to remedy this.
2021-07-11 11:44:07,129 INFO mapred.FileInputFormat: Total input files to process : 1
2021-07-11 11:44:07,140 INFO mapreduce.JobSubmitter: number of splits:1
2021-07-11 11:44:07,252 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local914342255_0001
2021-07-11 11:44:07,252 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-07-11 11:44:07,380 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2021-07-11 11:44:07,382 INFO mapreduce.Job: Running job: job_local914342255_0001
2021-07-11 11:44:07,392 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2021-07-11 11:44:07,393 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapred.FileOutputCommitter
2021-07-11 11:44:07,400 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2021-07-11 11:44:07,400 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:
false, ignore cleanup failures: false
2021-07-11 11:44:07,450 INFO mapred.LocalJobRunner: Waiting for map tasks
2021-07-11 11:44:07,453 INFO mapred.LocalJobRunner: Starting task: attempt_local914342255_0001_m_000000_0
2021-07-11 11:44:07,477 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2021-07-11 11:44:07,477 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:
false, ignore cleanup failures: false
2021-07-11 11:44:07,498 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2021-07-11 11:44:07,504 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/home/hadoop/myinput/EmployeeDB.csv:0+786
2021-07-11 11:44:07,540 INFO mapred.MapTask: numReduceTasks: 1
2021-07-11 11:44:07,589 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2021-07-11 11:44:07,593 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2021-07-11 11:44:07,593 INFO mapred.MapTask: soft limit at 83886080
2021-07-11 11:44:07,593 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2021-07-11 11:44:07,593 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2021-07-11 11:44:07,596 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
```

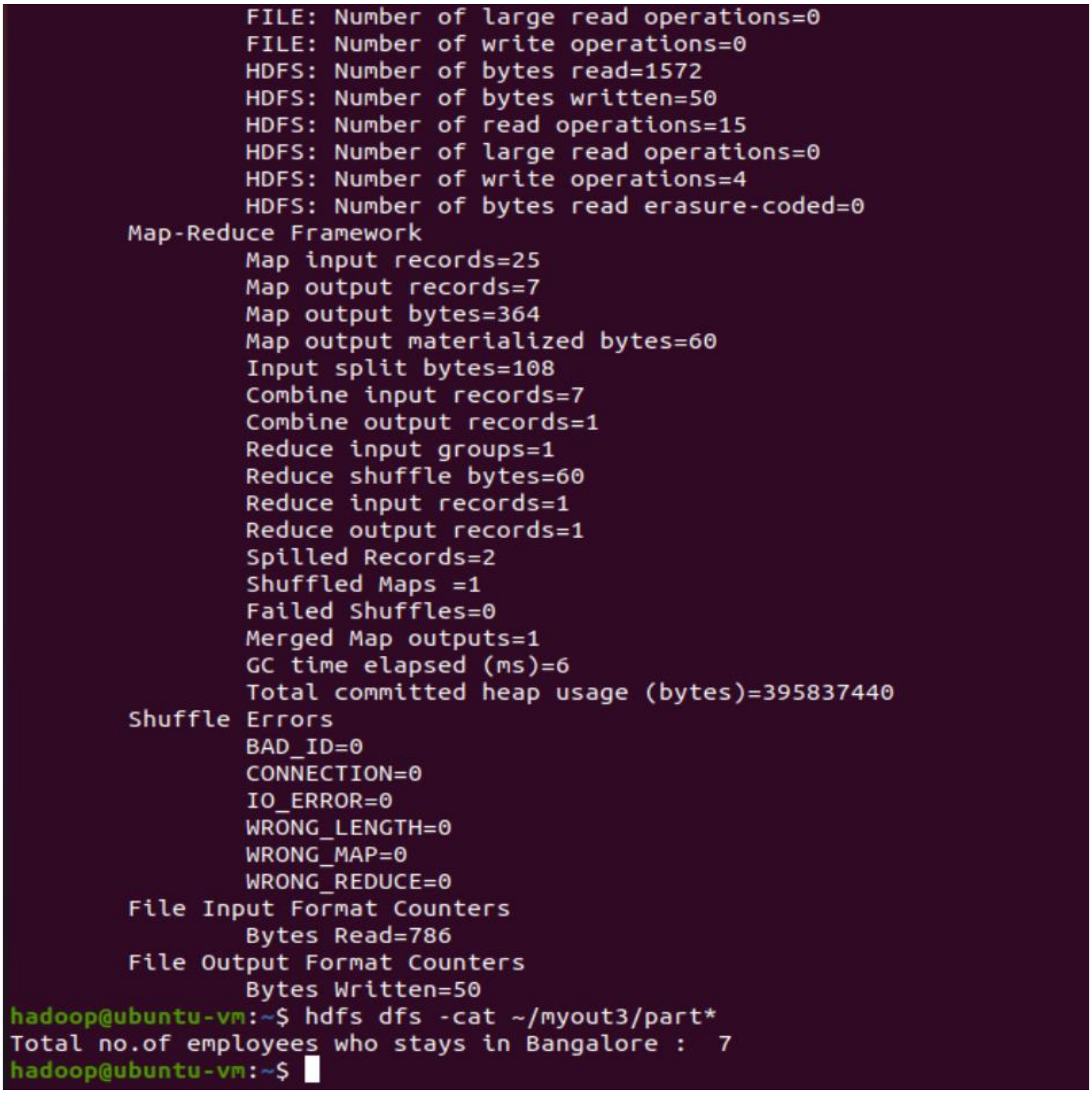


```
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1572
                HDFS: Number of bytes written=56
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=25
                Map output records=4
                Map output bytes=232
                Map output materialized bytes=66
                Input split bytes=108
                Combine input records=4
                Combine output records=1
                Reduce input groups=1
                Reduce shuffle bytes=66
                Reduce input records=1
                Reduce output records=1
                Spilled Records=2
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=18
                Total committed heap usage (bytes)=433061888
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=786
        File Output Format Counters
                Bytes Written=56
hadoop@ubuntu-vm:~$ hdfs dfs -cat ~/myout2/part*
Total no.of employees having 5 years of experience :     4
hadoop@ubuntu-vm:~$ 
```

3. Count the number of employees who lives in Bangalore



```
hadoop@ubuntu-vm:~$ hadoop jar /home/ubuntu/Desktop/prog3.jar ~/myinput ~/myout3
2021-07-11 11:47:43,477 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2021-07-11 11:47:43,548 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2021-07-11 11:47:43,548 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2021-07-11 11:47:43,560 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2021-07-11 11:47:43,742 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool int
erface and execute your application with ToolRunner to remedy this.
2021-07-11 11:47:43,871 INFO mapred.FileInputFormat: Total input files to process : 1
2021-07-11 11:47:43,888 INFO mapreduce.JobSubmitter: number of splits:1
2021-07-11 11:47:43,995 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local912882720_0001
2021-07-11 11:47:43,995 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-07-11 11:47:44,109 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2021-07-11 11:47:44,110 INFO mapreduce.Job: Running job: job_local912882720_0001
2021-07-11 11:47:44,110 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2021-07-11 11:47:44,111 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapred.FileOutputCommitter
2021-07-11 11:47:44,116 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2021-07-11 11:47:44,116 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:
false, ignore cleanup failures: false
2021-07-11 11:47:44,178 INFO mapred.LocalJobRunner: Waiting for map tasks
2021-07-11 11:47:44,181 INFO mapred.LocalJobRunner: Starting task: attempt_local912882720_0001_m_000000_0
2021-07-11 11:47:44,207 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2021-07-11 11:47:44,207 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:
false, ignore cleanup failures: false
2021-07-11 11:47:44,225 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2021-07-11 11:47:44,235 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/home/hadoop/myinput/EmployeeDB.csv:0+786
2021-07-11 11:47:44,283 INFO mapred.MapTask: numReduceTasks: 1
2021-07-11 11:47:44,343 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2021-07-11 11:47:44,343 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2021-07-11 11:47:44,343 INFO mapred.MapTask: soft limit at 83886080
2021-07-11 11:47:44,343 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2021-07-11 11:47:44,343 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2021-07-11 11:47:44,347 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2021-07-11 11:47:44,463 INFO mapred.LocalJobRunner:
2021-07-11 11:47:44,463 INFO mapred.MapTask: Starting flush of map output
2021-07-11 11:47:44,463 INFO mapred.MapTask: Spilling map output
2021-07-11 11:47:44,463 INFO mapred.MapTask: bufstart = 0; bufend = 364; bufvoid = 104857600
```



```
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1572
                HDFS: Number of bytes written=50
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=25
                Map output records=7
                Map output bytes=364
                Map output materialized bytes=60
                Input split bytes=108
                Combine input records=7
                Combine output records=1
                Reduce input groups=1
                Reduce shuffle bytes=60
                Reduce input records=1
                Reduce output records=1
                Spilled Records=2
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=6
                Total committed heap usage (bytes)=395837440
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=786
        File Output Format Counters
                Bytes Written=50
hadoop@ubuntu-vm:~$ hdfs dfs -cat ~/myout3/part*
Total no.of employees who stays in Bangalore :  7
hadoop@ubuntu-vm:~$
```

✞ What is Hive ??

Hive is a **data warehouse infrastructure tool** to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.

Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive. It is used by different companies. For example, Amazon uses it in Amazon Elastic MapReduce.

✞ Hive is not

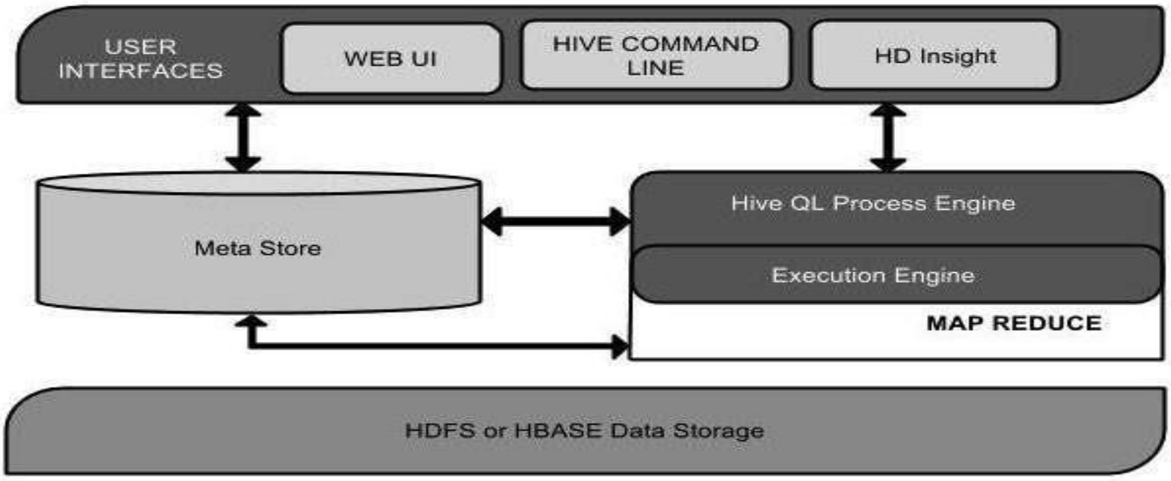
- A relational database.
- A design for OnLine Transaction Processing (OLTP)
- A language for real-time queries and row-level updates

✞ 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

The following component diagram depicts the architecture of Hive:



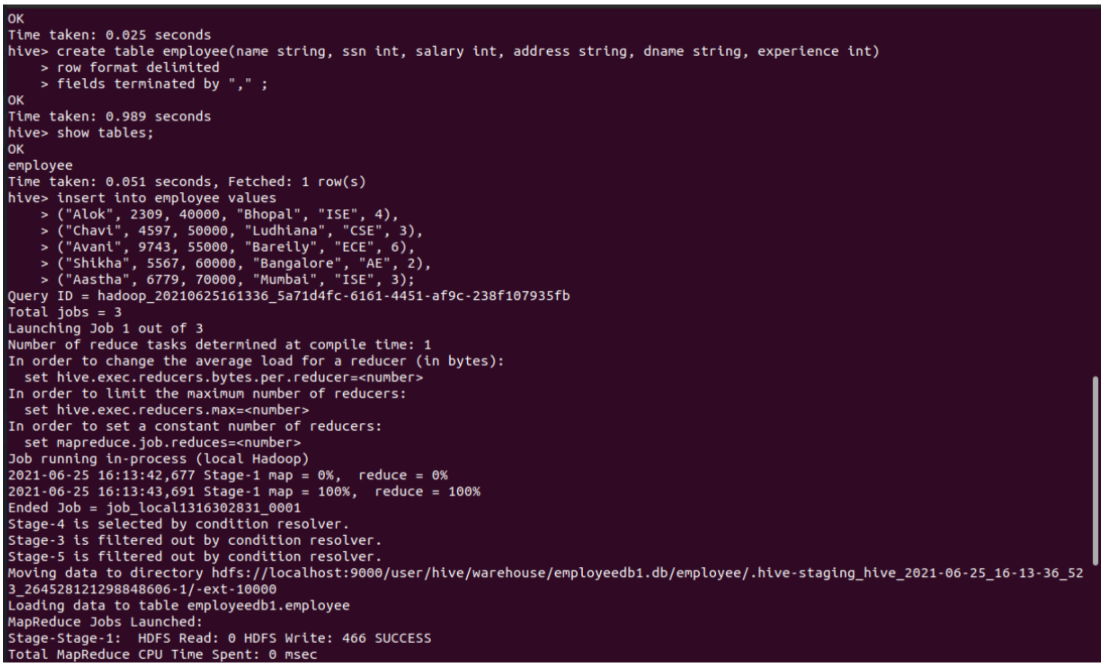
Exercise-II

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,

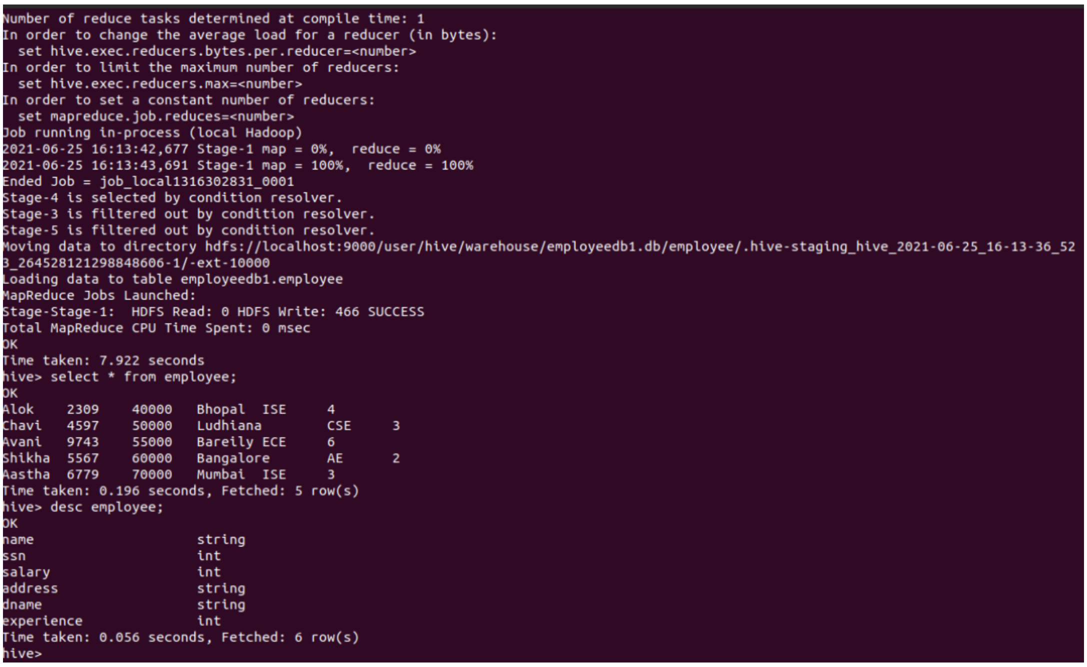
Name	SSN	Salary	Address	Dname	Experience
Harsha	5000	30000	Bangalore	ISE	5

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

1. Insert 5 records using INSERT command.

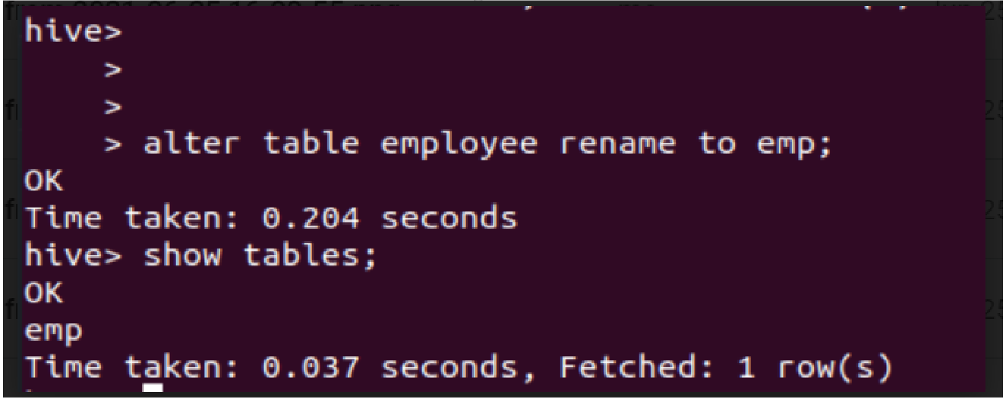


```
OK
Time taken: 0.025 seconds
hive> create table employee(name string, ssn int, salary int, address string, dname string, experience int)
    > row format delimited
    > fields terminated by "," ;
OK
Time taken: 0.989 seconds
hive> show tables;
OK
employee
Time taken: 0.051 seconds, Fetched: 1 row(s)
hive> insert into employee values
    > ("Alok", 2309, 40000, "Bhopal", "ISE", 4),
    > ("Chavi", 4597, 50000, "Ludhiana", "CSE", 3),
    > ("Avani", 9743, 55000, "Bareily", "ECE", 6),
    > ("Shikha", 5567, 60000, "Bangalore", "AE", 2),
    > ("Aastha", 6779, 70000, "Mumbai", "ISE", 3);
Query ID = hadoop_20210625161336_5a71d4fc-6161-4451-af9c-238f107935fb
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>
Job running in-process (local Hadoop)
2021-06-25 16:13:42,677 Stage-1 map = 0%,  reduce = 0%
2021-06-25 16:13:43,691 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1316302831_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://localhost:9000/user/hive/warehouse/employeedb1.db/employee/.hive-staging_hive_2021-06-25_16-13-36_52
3_264528121298848606-1/-ext-10000
Loading data to table employeedb1.employee
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 466 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
```



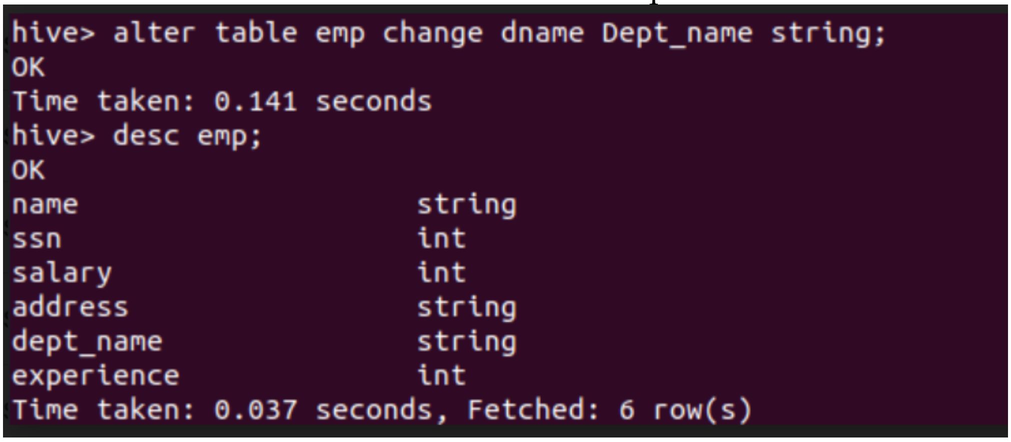
```
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2021-06-25 16:13:42,677 Stage-1 map = 0%,  reduce = 0%
2021-06-25 16:13:43,691 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1316302831_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://localhost:9000/user/hive/warehouse/employeedb1.db/employee/.hive-staging_hive_2021-06-25_16-13-36_52
3_264528121298848606-1/-ext-10000
Loading data to table employeedb1.employee
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 466 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 7.922 seconds
hive> select * from employee;
OK
Alok    2309    40000   Bhopal  ISE     4
Chavi   4597    50000   Ludhiana        CSE     3
Avani   9743    55000   Bareily ECE     6
Shikha  5567    60000   Bangalore       AE      2
Aastha  6779    70000   Mumbai  ISE     3
Time taken: 0.196 seconds, Fetched: 5 row(s)
hive> desc employee;
OK
name                    string
ssn                     int
salary                  int
address                 string
dname                   string
experience              int
Time taken: 0.056 seconds, Fetched: 6 row(s)
hive>
```

2. Demonstrate the Alter command for the following cases,
   a. Rename the table name to “Emp”.



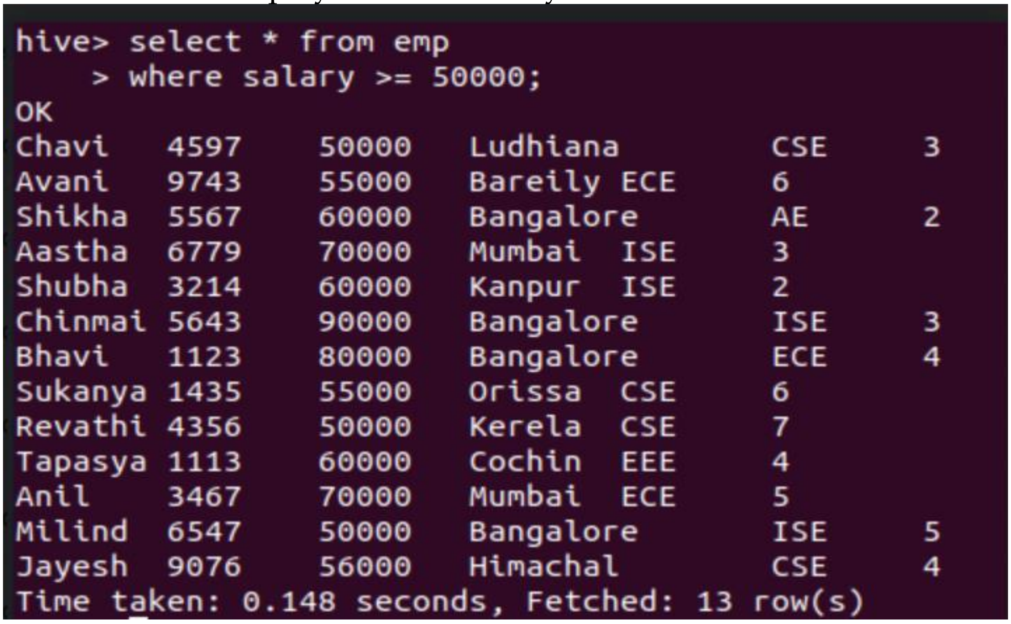
```
hive>
    >
    >
    > alter table employee rename to emp;
OK
Time taken: 0.204 seconds
hive> show tables;
OK
emp
Time taken: 0.037 seconds, Fetched: 1 row(s)
```

   b. Rename the column name “Dname” to “Dept_name”.



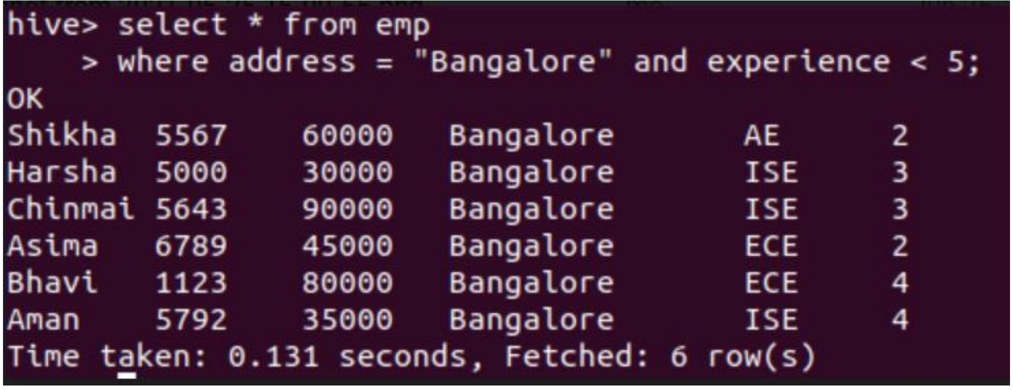
```
hive> alter table emp change dname Dept_name string;
OK
Time taken: 0.141 seconds
hive> desc emp;
OK
name                    string
ssn                     int
salary                  int
address                 string
dept_name               string
experience              int
Time taken: 0.037 seconds, Fetched: 6 row(s)
```

3. Retrieve all the employees who’s salary is not less than 50000.



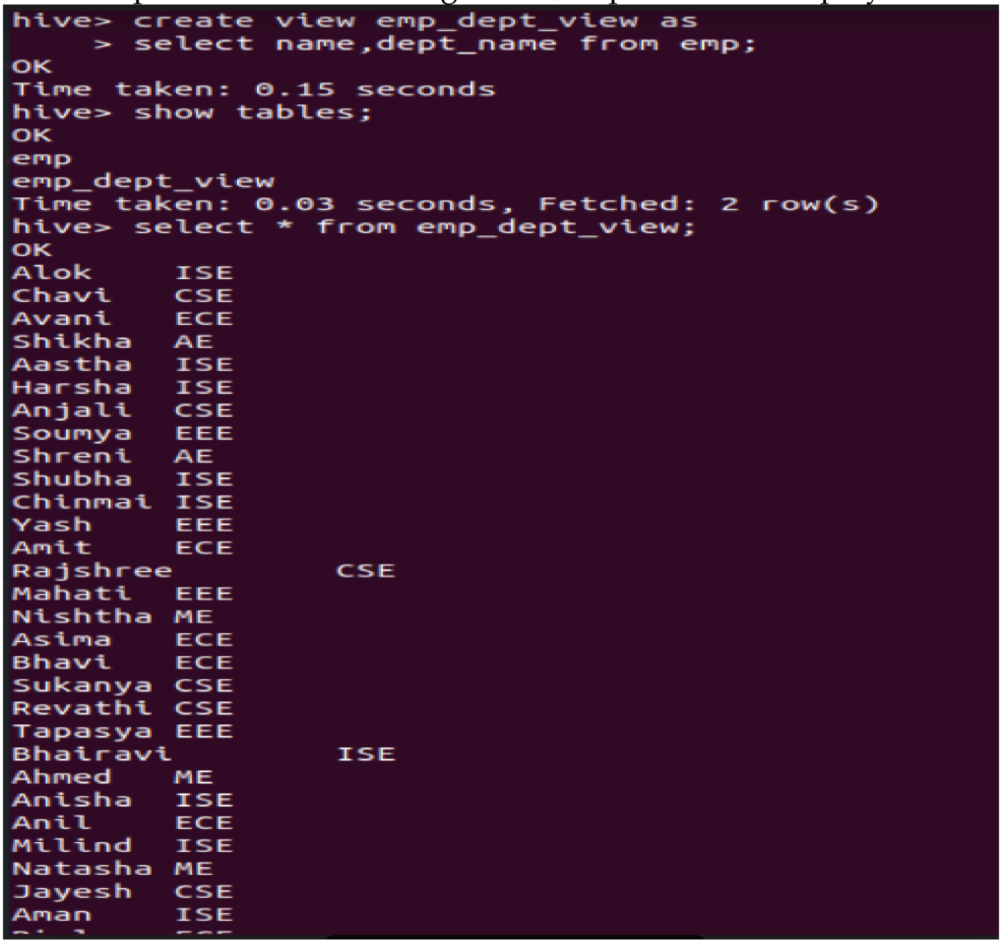
```
hive> select * from emp
    > where salary >= 50000;
OK
Chavi   4597    50000   Ludhiana        CSE     3
Avani   9743    55000   Bareily ECE     6
Shikha  5567    60000   Bangalore       AE      2
Aastha  6779    70000   Mumbai  ISE     3
Shubha  3214    60000   Kanpur  ISE     2
Chinmai 5643    90000   Bangalore       ISE     3
Bhavi   1123    80000   Bangalore       ECE     4
Sukanya 1435    55000   Orissa  CSE     6
Revathi 4356    50000   Kerela  CSE     7
Tapasya 1113    60000   Cochin  EEE     4
Anil    3467    70000   Mumbai  ECE     5
Milind  6547    50000   Bangalore       ISE     5
Jayesh  9076    56000   Himachal        CSE     4
Time taken: 0.148 seconds, Fetched: 13 row(s)
```

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



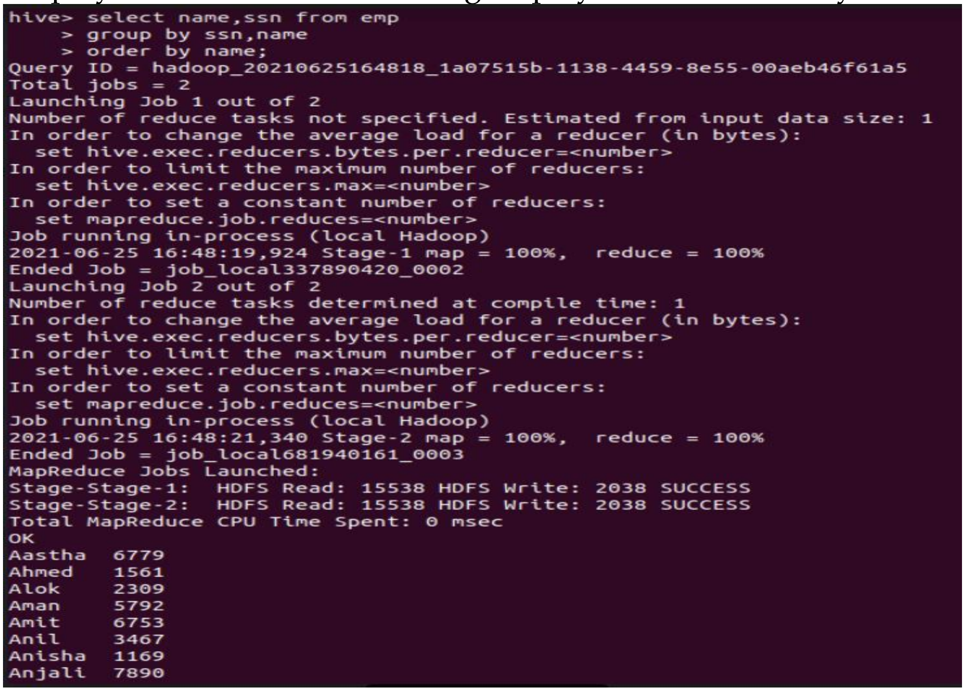
```
hive> select * from emp
    > where address = "Bangalore" and experience < 5;
OK
Shikha  5567    60000   Bangalore       AE      2
Harsha  5000    30000   Bangalore       ISE     3
Chinmai 5643    90000   Bangalore       ISE     3
Asima   6789    45000   Bangalore       ECE     2
Bhavi   1123    80000   Bangalore       ECE     4
Aman    5792    35000   Bangalore       ISE     4
Time taken: 0.131 seconds, Fetched: 6 row(s)
```

5. Create separate view containing Name, Dept_name of employees

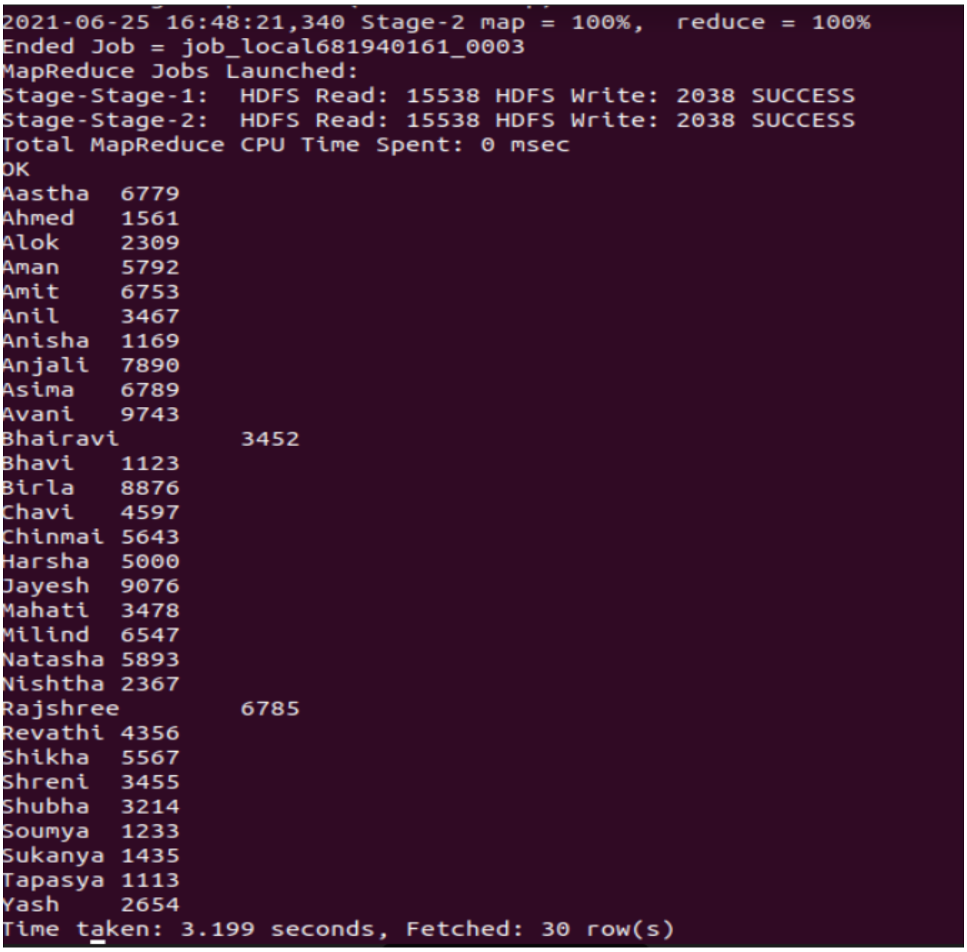


```
hive> create view emp_dept_view as
    > select name,dept_name from emp;
OK
Time taken: 0.15 seconds
hive> show tables;
OK
emp
emp_dept_view
Time taken: 0.03 seconds, Fetched: 2 row(s)
hive> select * from emp_dept_view;
OK
Alok    ISE
Chavi   CSE
Avani   ECE
Shikha  AE
Aastha  ISE
Harsha  ISE
Anjali  CSE
Soumya  EEE
Shreni  AE
Shubha  ISE
Chinmai ISE
Yash    EEE
Amit    ECE
Rajshree        CSE
Mahati  EEE
Nishtha ME
Asima   ECE
Bhavi   ECE
Sukanya CSE
Revathi CSE
Tapasya EEE
Bhairavi        ISE
Ahmed   ME
Anisha  ISE
Anil    ECE
Milind  ISE
Natasha ME
Jayesh  CSE
Aman    ISE
```

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

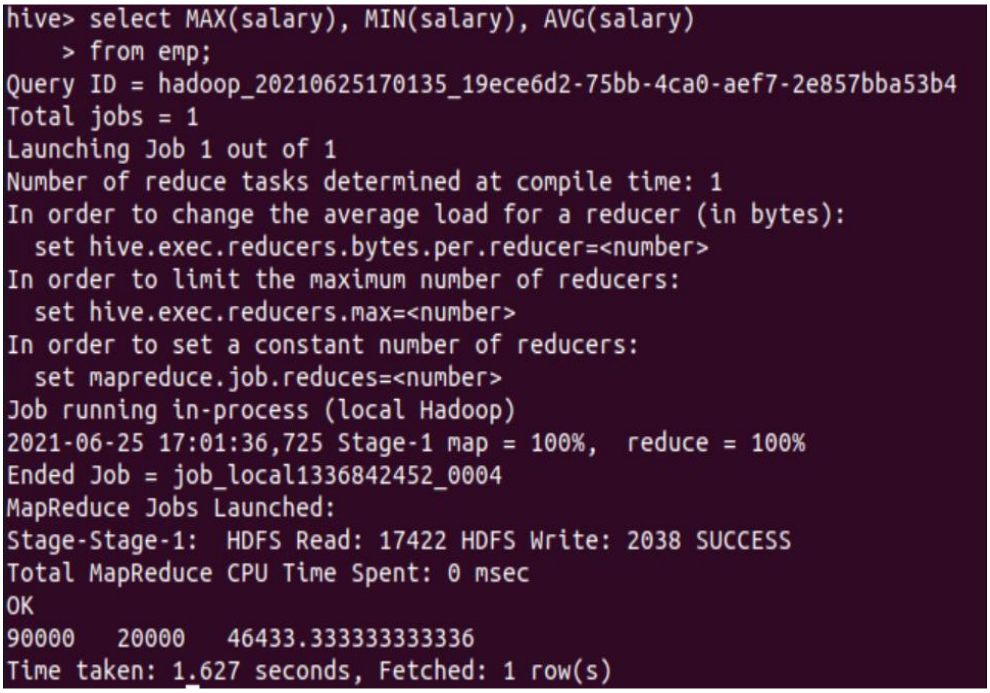


```
hive> select name,ssn from emp
    > group by ssn,name
    > order by name;
Query ID = hadoop_20210625164818_1a07515b-1138-4459-8e55-00aeb46f61a5
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for 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>
Job running in-process (local Hadoop)
2021-06-25 16:48:19,924 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local337890420_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>
Job running in-process (local Hadoop)
2021-06-25 16:48:21,340 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_local681940161_0003
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 15538 HDFS Write: 2038 SUCCESS
Stage-Stage-2:  HDFS Read: 15538 HDFS Write: 2038 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Aastha  6779
Ahmed   1561
Alok    2309
Aman    5792
Amit    6753
Anil    3467
Anisha  1169
Anjali  7890
```



```
2021-06-25 16:48:21,340 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_local681940161_0003
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 15538 HDFS Write: 2038 SUCCESS
Stage-Stage-2:  HDFS Read: 15538 HDFS Write: 2038 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Aastha  6779
Ahmed   1561
Alok    2309
Aman    5792
Amit    6753
Anil    3467
Anisha  1169
Anjali  7890
Asima   6789
Avani   9743
Bhairavi        3452
Bhavi   1123
Birla   8876
Chavi   4597
Chinmai 5643
Harsha  5000
Jayesh  9076
Mahati  3478
Milind  6547
Natasha 5893
Nishtha 2367
Rajshree        6785
Revathi 4356
Shikha  5567
Shreni  3455
Shubha  3214
Soumya  1233
Sukanya 1435
Tapasya 1113
Yash    2654
Time taken: 3.199 seconds, Fetched: 30 row(s)
```

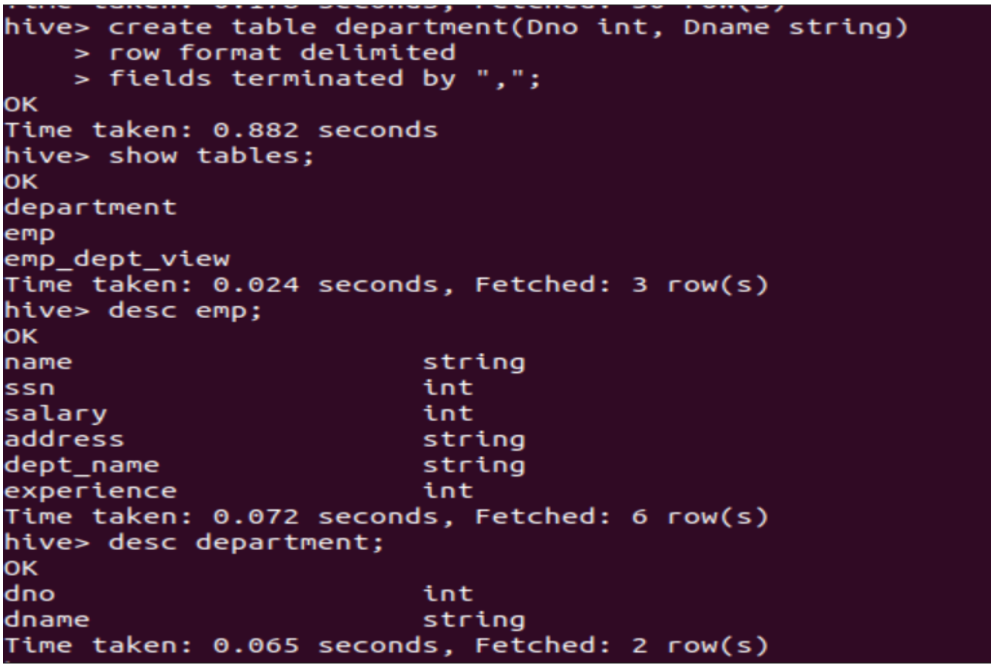
7. Retrieve Maximum salary, minimum salary and Average salary of the employees



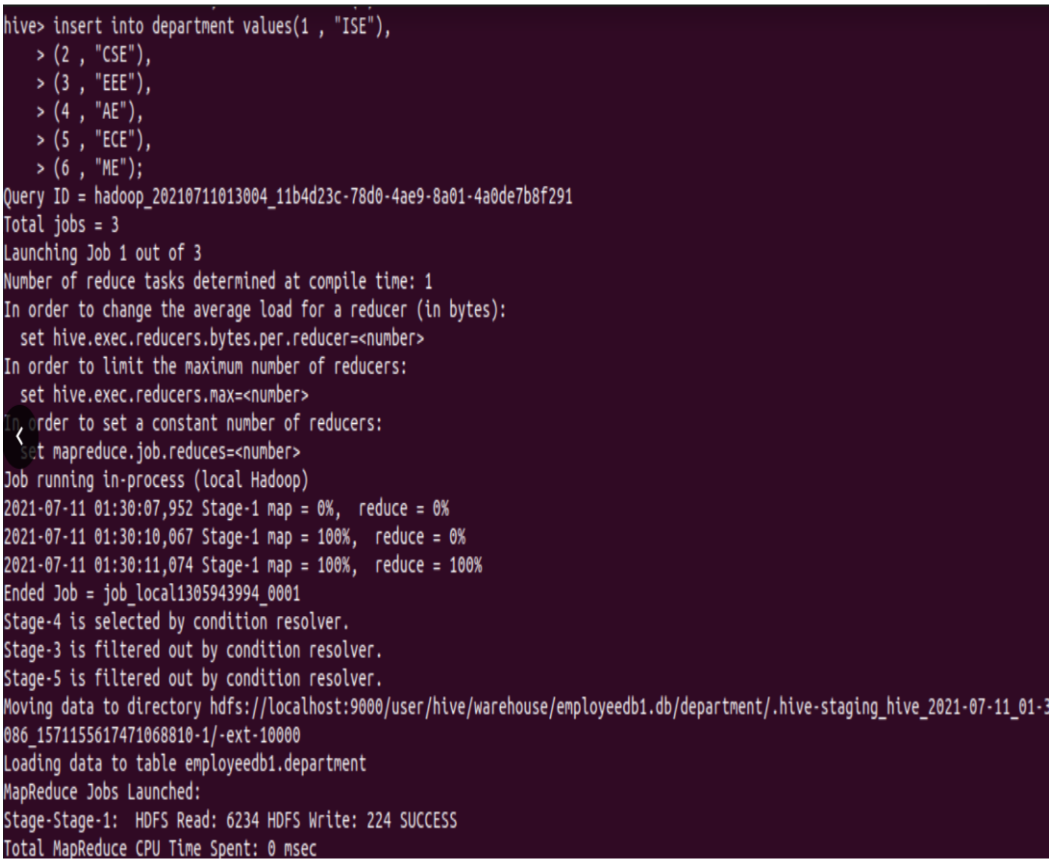
```
hive> select MAX(salary), MIN(salary), AVG(salary)
    > from emp;
Query ID = hadoop_20210625170135_19ece6d2-75bb-4ca0-aef7-2e857bba53b4
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set 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)
2021-06-25 17:01:36,725 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1336842452_0004
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 17422 HDFS Write: 2038 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
90000   20000   46433.333333333336
Time taken: 1.627 seconds, Fetched: 1 row(s)
```

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

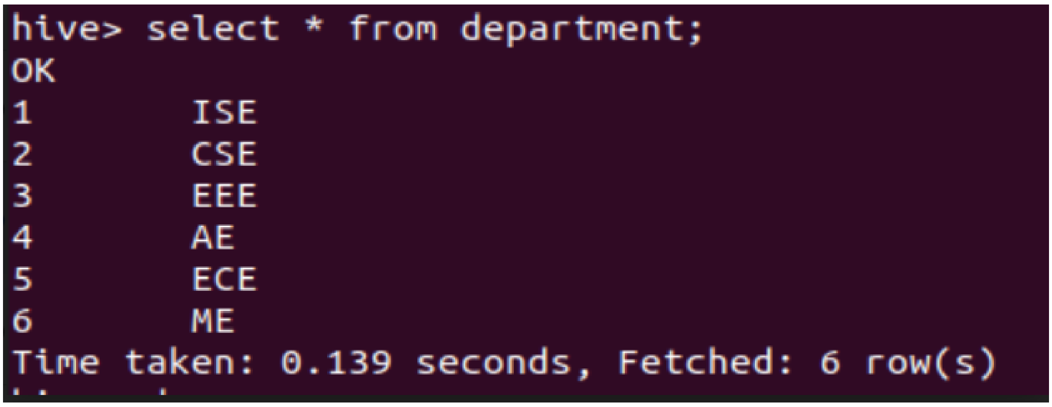
Dno	Dname
6	ISE



```
hive> create table department(Dno int, Dname string)
    > row format delimited
    > fields terminated by ",";
OK
Time taken: 0.882 seconds
hive> show tables;
OK
department
emp
emp_dept_view
Time taken: 0.024 seconds, Fetched: 3 row(s)
hive> desc emp;
OK
name                    string
ssn                     int
salary                  int
address                 string
dept_name               string
experience              int
Time taken: 0.072 seconds, Fetched: 6 row(s)
hive> desc department;
OK
dno                     int
dname                   string
Time taken: 0.065 seconds, Fetched: 2 row(s)
```

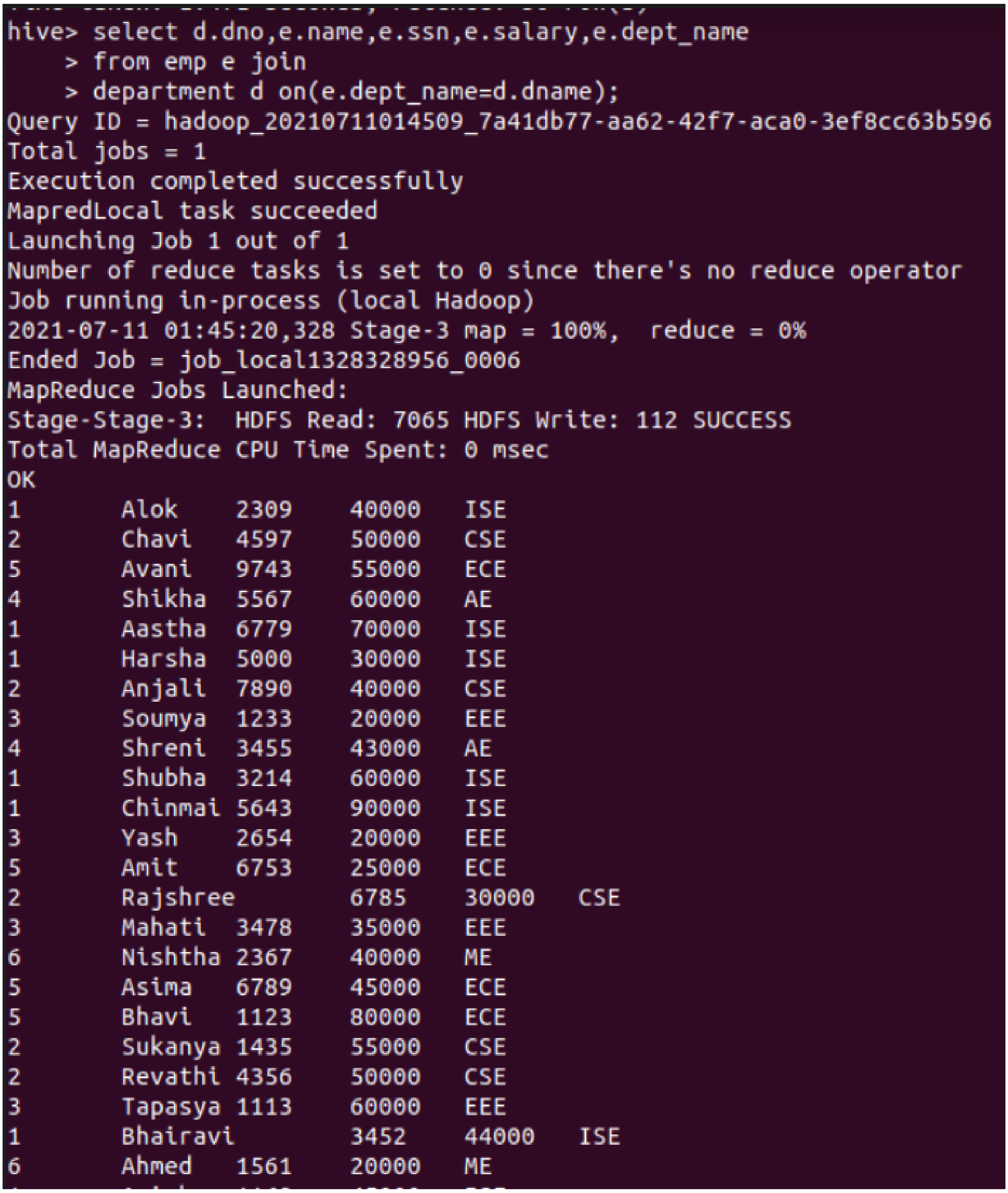


```
hive> insert into department values(1 , "ISE"),
    > (2 , "CSE"),
    > (3 , "EEE"),
    > (4 , "AE"),
    > (5 , "ECE"),
    > (6 , "ME");
Query ID = hadoop_20210711013004_11b4d23c-78d0-4ae9-8a01-4a0de7b8f291
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 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>
Job running in-process (local Hadoop)
2021-07-11 01:30:07,952 Stage-1 map = 0%,  reduce = 0%
2021-07-11 01:30:10,067 Stage-1 map = 100%,  reduce = 0%
2021-07-11 01:30:11,074 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1305943994_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://localhost:9000/user/hive/warehouse/employeedb1.db/department/.hive-staging_hive_2021-07-11_01-3
086_1571155617471068810-1/-ext-10000
Loading data to table employeedb1.department
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 6234 HDFS Write: 224 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
```



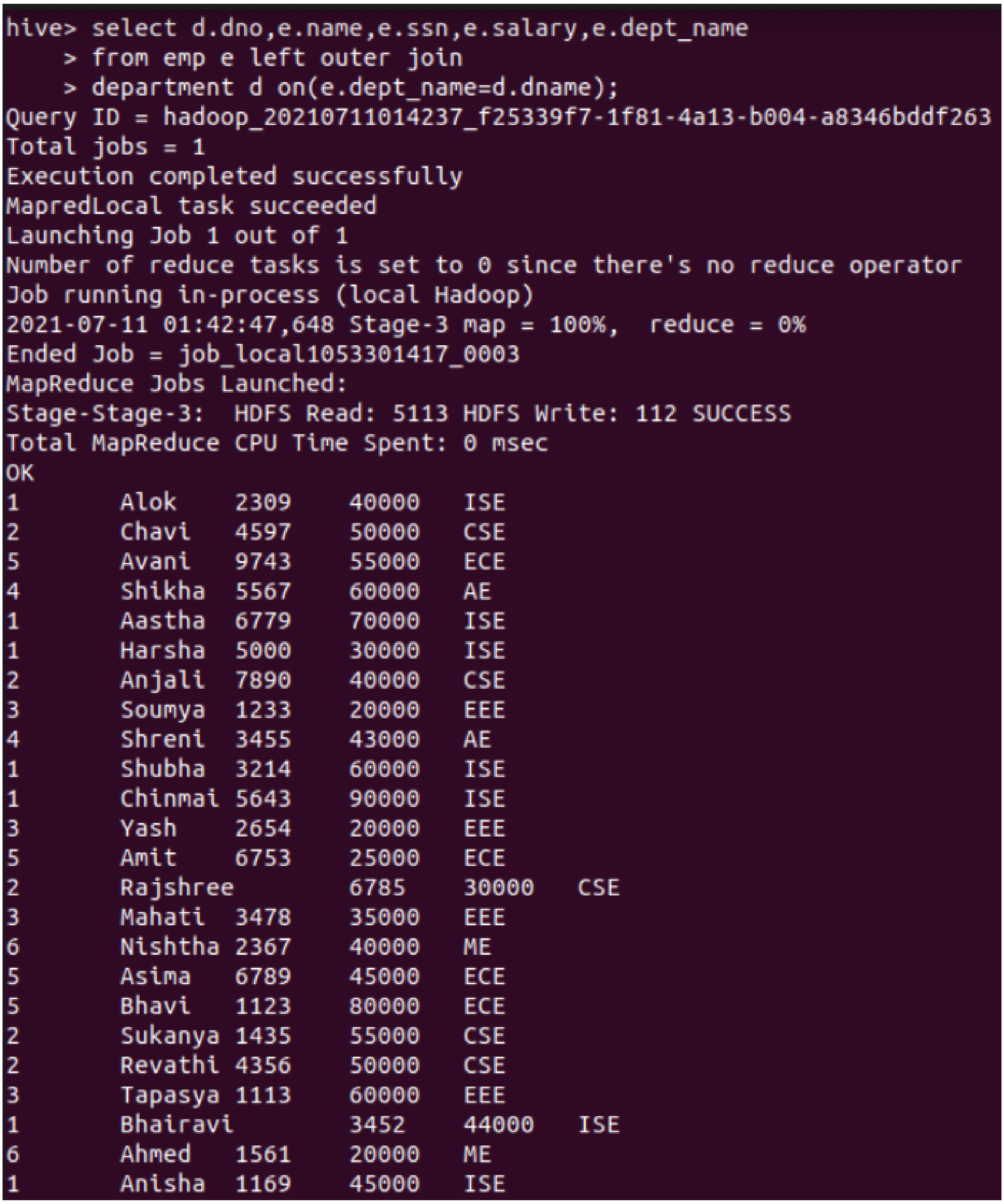
```
hive> select * from department;
OK
1       ISE
2       CSE
3       EEE
4       AE
5       ECE
6       ME
Time taken: 0.139 seconds, Fetched: 6 row(s)
```

a) JOIN



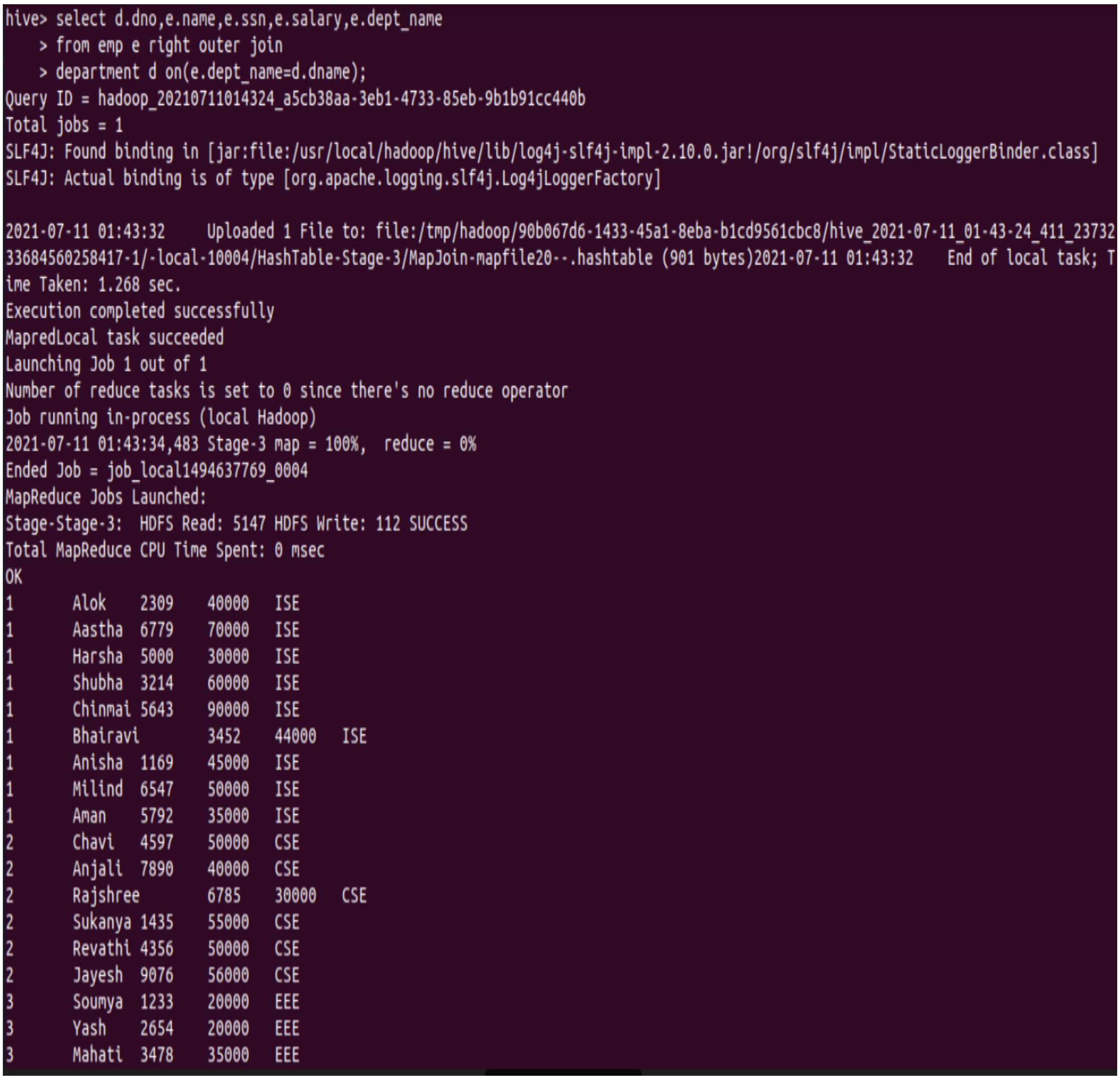
```
hive> select d.dno,e.name,e.ssn,e.salary,e.dept_name
    > from emp e join
    > department d on(e.dept_name=d.dname);
Query ID = hadoop_20210711014509_7a41db77-aa62-42f7-aca0-3ef8cc63b596
Total jobs = 1
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2021-07-11 01:45:20,328 Stage-3 map = 100%,  reduce = 0%
Ended Job = job_local1328328956_0006
MapReduce Jobs Launched:
Stage-Stage-3:  HDFS Read: 7065 HDFS Write: 112 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
1       Alok    2309    40000   ISE
2       Chavi   4597    50000   CSE
5       Avani   9743    55000   ECE
4       Shikha  5567    60000   AE
1       Aastha  6779    70000   ISE
1       Harsha  5000    30000   ISE
2       Anjali  7890    40000   CSE
3       Soumya  1233    20000   EEE
4       Shreni  3455    43000   AE
1       Shubha  3214    60000   ISE
1       Chinmai 5643    90000   ISE
3       Yash    2654    20000   EEE
5       Amit    6753    25000   ECE
2       Rajshree        6785    30000   CSE
3       Mahati  3478    35000   EEE
6       Nishtha 2367    40000   ME
5       Asima   6789    45000   ECE
5       Bhavi   1123    80000   ECE
2       Sukanya 1435    55000   CSE
2       Revathi 4356    50000   CSE
3       Tapasya 1113    60000   EEE
1       Bhairavi        3452    44000   ISE
6       Ahmed   1561    20000   ME
```

b) LEFT OUTER JOIN



```
hive> select d.dno,e.name,e.ssn,e.salary,e.dept_name
    > from emp e left outer join
    > department d on(e.dept_name=d.dname);
Query ID = hadoop_20210711014237_f25339f7-1f81-4a13-b004-a8346bddf263
Total jobs = 1
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
Job running in-process (local Hadoop)
2021-07-11 01:42:47,648 Stage-3 map = 100%,  reduce = 0%
Ended Job = job_local1053301417_0003
MapReduce Jobs Launched:
Stage-Stage-3:  HDFS Read: 5113 HDFS Write: 112 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
1       Alok    2309    40000   ISE
2       Chavi   4597    50000   CSE
5       Avani   9743    55000   ECE
4       Shikha  5567    60000   AE
1       Aastha  6779    70000   ISE
1       Harsha  5000    30000   ISE
2       Anjali  7890    40000   CSE
3       Soumya  1233    20000   EEE
4       Shreni  3455    43000   AE
1       Shubha  3214    60000   ISE
1       Chinmai 5643    90000   ISE
3       Yash    2654    20000   EEE
5       Amit    6753    25000   ECE
2       Rajshree        6785    30000   CSE
3       Mahati  3478    35000   EEE
6       Nishtha 2367    40000   ME
5       Asima   6789    45000   ECE
5       Bhavi   1123    80000   ECE
2       Sukanya 1435    55000   CSE
2       Revathi 4356    50000   CSE
3       Tapasya 1113    60000   EEE
1       Bhairavi        3452    44000   ISE
6       Ahmed   1561    20000   ME
1       Anisha  1169    45000   ISE
```

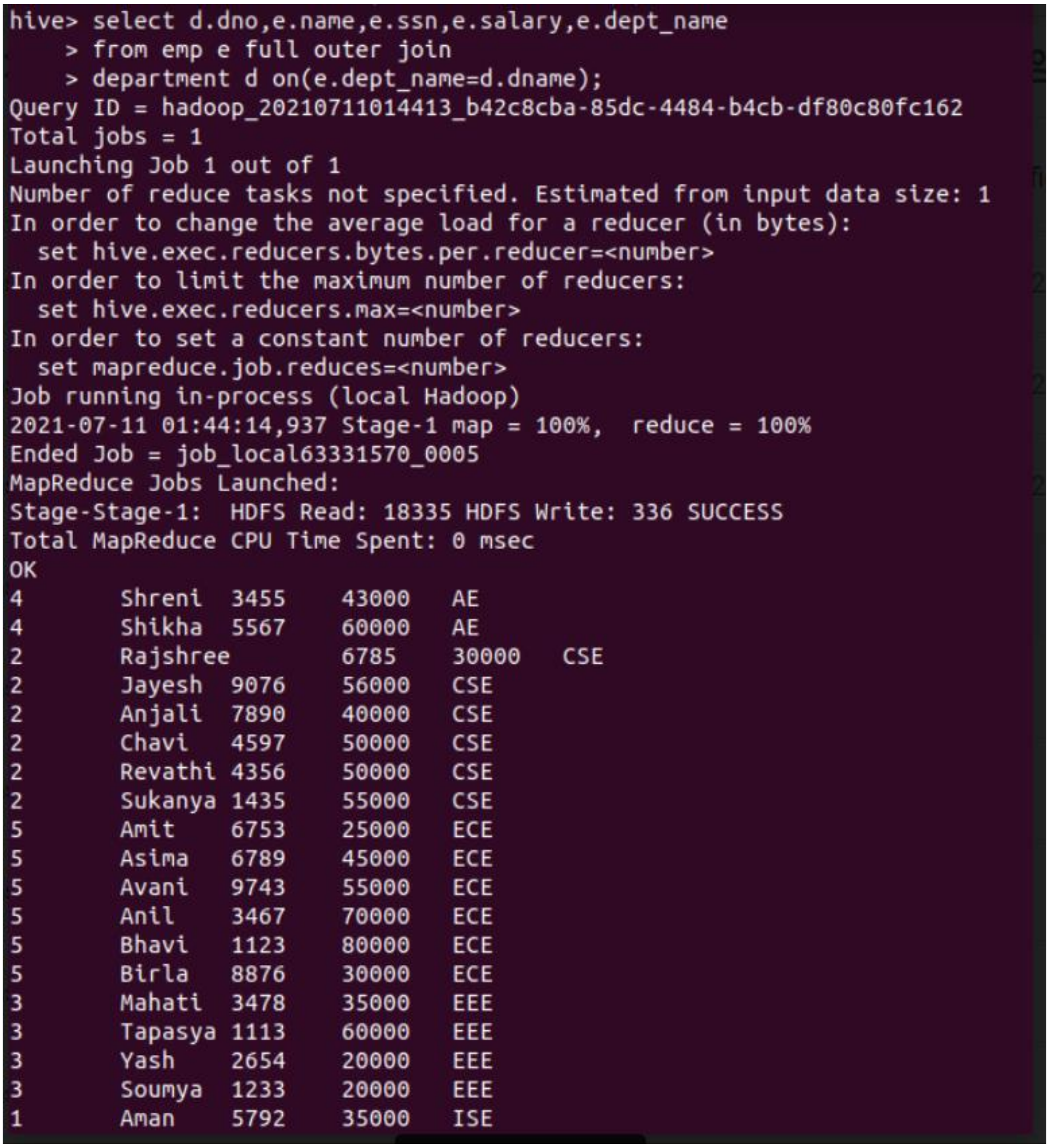
c) RIGHT OUTER JOIN



```
hive> select d.dno,e.name,e.ssn,e.salary,e.dept_name
    > from emp e right outer join
    > department d on(e.dept_name=d.dname);
Query ID = hadoop_20210711014324_a5cb38aa-3eb1-4733-85eb-9b1b91cc440b
Total jobs = 1
SLF4J: Found binding in [jar:file:/usr/local/hadoop/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

2021-07-11 01:43:32     Uploaded 1 File to: file:/tmp/hadoop/90b067d6-1433-45a1-8eba-b1cd9561cbc8/hive_2021-07-11_01-43-24_411_23732
33684560258417-1/-local-10004/HashTable-Stage-3/MapJoin-mapfile20--.hashtable (901 bytes)2021-07-11 01:43:32    End of local task; T
ime Taken: 1.268 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
Job running in-process (local Hadoop)
2021-07-11 01:43:34,483 Stage-3 map = 100%,  reduce = 0%
Ended Job = job_local1494637769_0004
MapReduce Jobs Launched:
Stage-Stage-3:  HDFS Read: 5147 HDFS Write: 112 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
1       Alok    2309    40000   ISE
1       Aastha  6779    70000   ISE
1       Harsha  5000    30000   ISE
1       Shubha  3214    60000   ISE
1       Chinmai 5643    90000   ISE
1       Bhairavi        3452    44000   ISE
1       Anisha  1169    45000   ISE
1       Milind  6547    50000   ISE
1       Aman    5792    35000   ISE
2       Chavi   4597    50000   CSE
2       Anjali  7890    40000   CSE
2       Rajshree        6785    30000   CSE
2       Sukanya 1435    55000   CSE
2       Revathi 4356    50000   CSE
2       Jayesh  9076    56000   CSE
3       Soumya  1233    20000   EEE
3       Yash    2654    20000   EEE
3       Mahati  3478    35000   EEE
```

d) FULL OUTER JOIN



```
hive> select d.dno,e.name,e.ssn,e.salary,e.dept_name
    > from emp e full outer join
    > department d on(e.dept_name=d.dname);
Query ID = hadoop_20210711014413_b42c8cba-85dc-4484-b4cb-df80c80fc162
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>
Job running in-process (local Hadoop)
2021-07-11 01:44:14,937 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local63331570_0005
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 18335 HDFS Write: 336 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
4       Shreni  3455    43000   AE
4       Shikha  5567    60000   AE
2       Rajshree        6785    30000   CSE
2       Jayesh  9076    56000   CSE
2       Anjali  7890    40000   CSE
2       Chavi   4597    50000   CSE
2       Revathi 4356    50000   CSE
2       Sukanya 1435    55000   CSE
5       Amit    6753    25000   ECE
5       Asima   6789    45000   ECE
5       Avani   9743    55000   ECE
5       Anil    3467    70000   ECE
5       Bhavi   1123    80000   ECE
5       Birla   8876    30000   ECE
3       Mahati  3478    35000   EEE
3       Tapasya 1113    60000   EEE
3       Yash    2654    20000   EEE
3       Soumya  1233    20000   EEE
1       Aman    5792    35000   ISE
```

Github links for the source code :

https://github.com/1nt18is046/BIGDATA

References:

Video References:

1. https://youtu.be/K0aDh_sfVrc
2. https://youtu.be/U3fkWvaqgl8
3. https://youtu.be/SAX8b3AN3Uc

Information resources:

1. https://www.google.co.in/
2. https://en.wikipedia.org
3. https://www.tutorialspoint.com/
4. https://hadoop.apache.org/
5. https://www.geeksforgeeks.org/