NITTE EDUCATION TRUST | NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

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

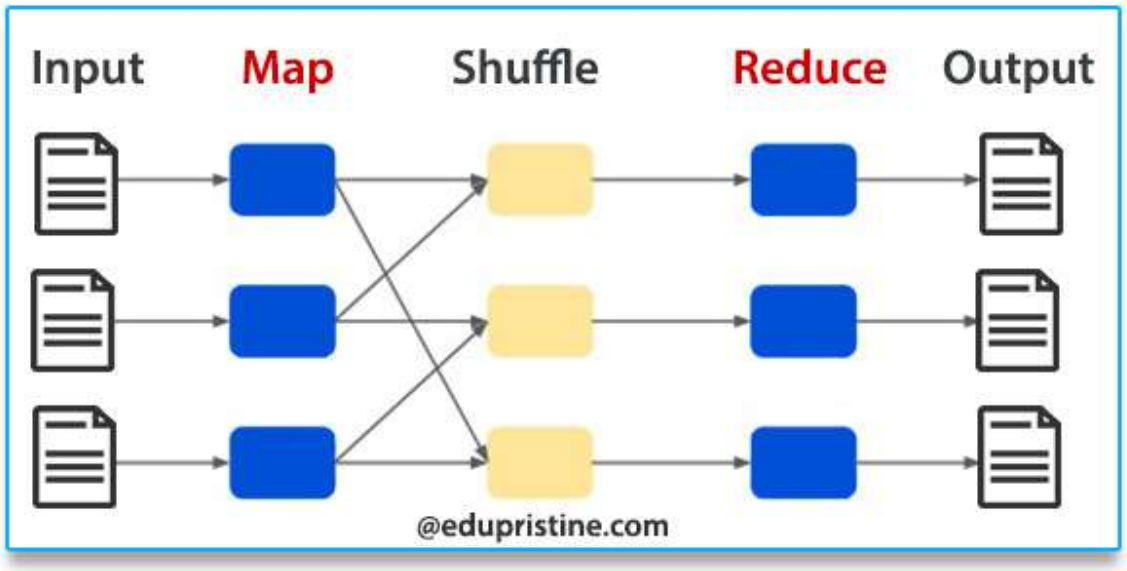
Table of Content:

1. 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 intermediatory key-value pair. Input key-value pair, Intermediatory 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 :
    - IDException 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.

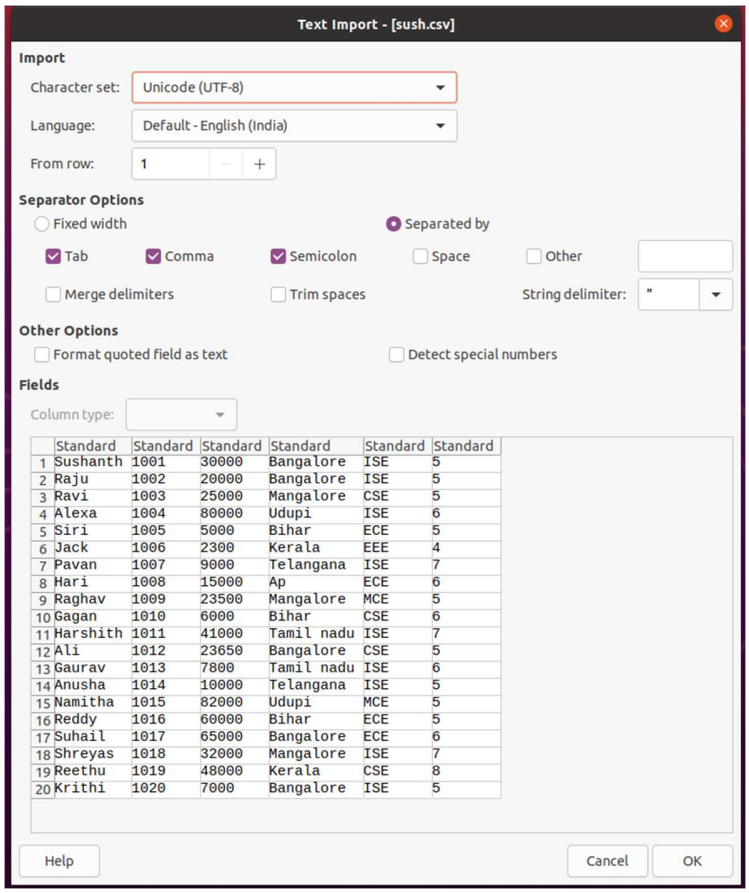
Name	SSN	Salary	Address	Dname	Experience
Harsha	5000	30000	Bangalore	ISE	5

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)



	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

File Name/Type : sush/csv

File size : 620kb

Number of Rows : 20

Number of Columns : 6

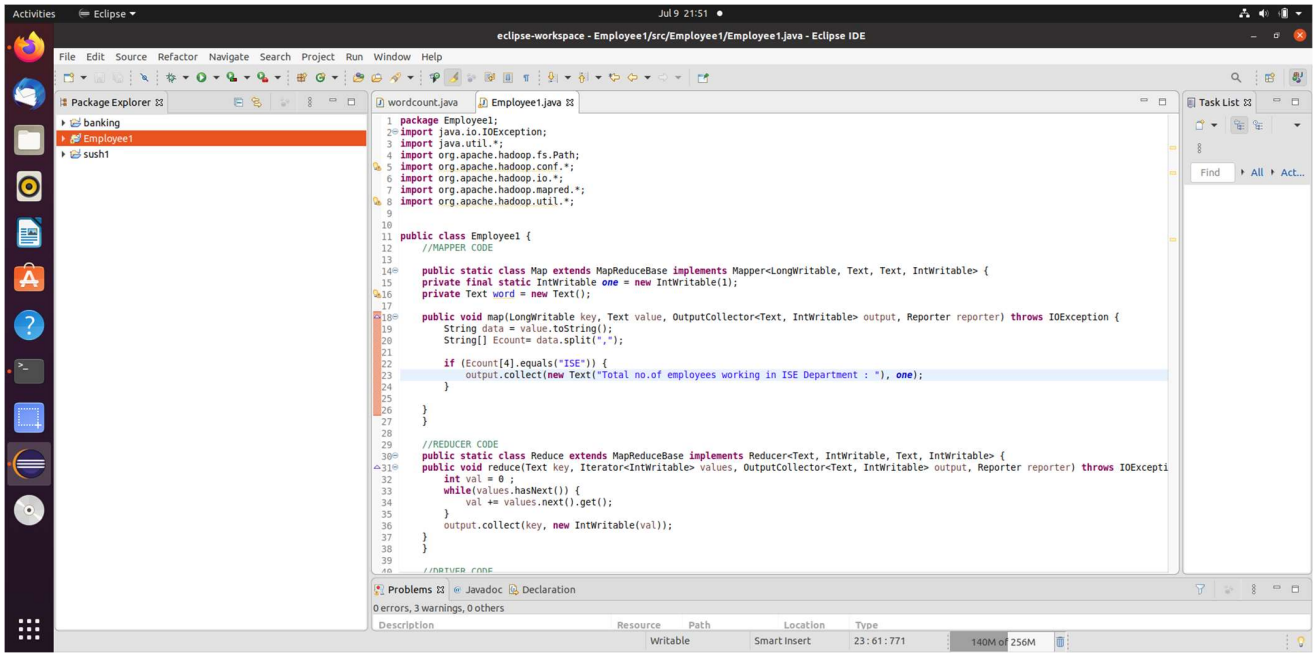
4. GITHUB LINK

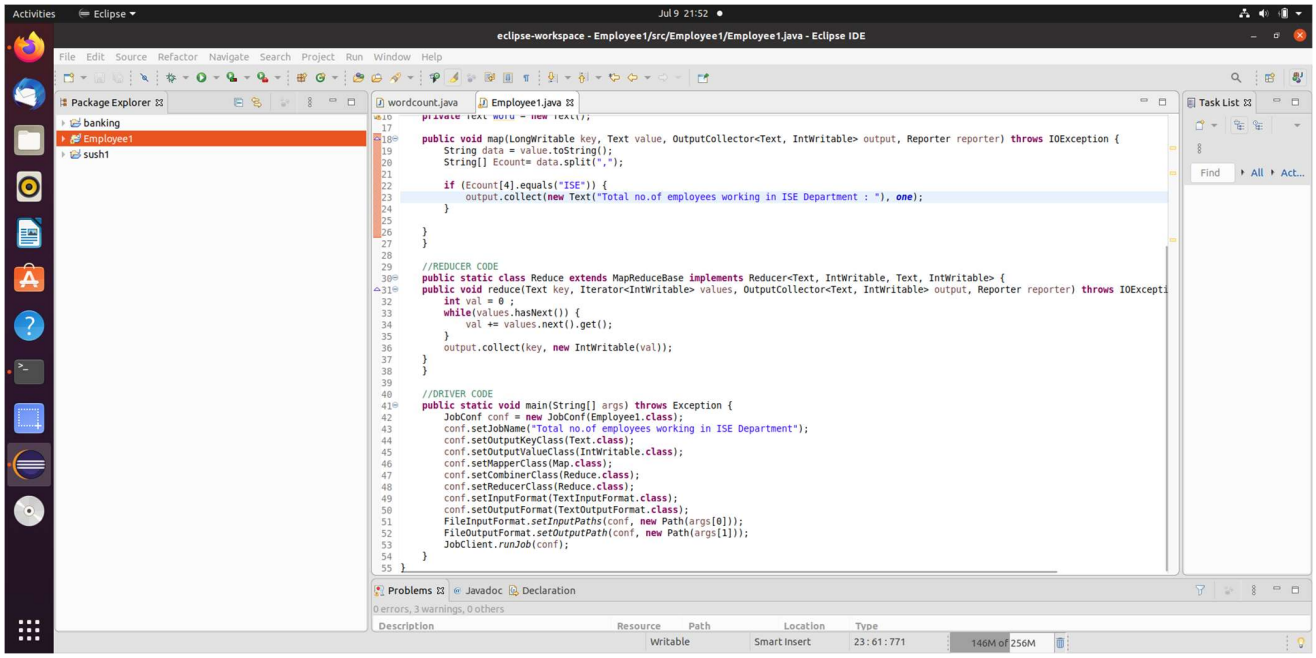
https://github.com/Sushanth-raju/BigData.git

5. RESULTS AND SNAPSHOTS OF HADOOP PROGRAM

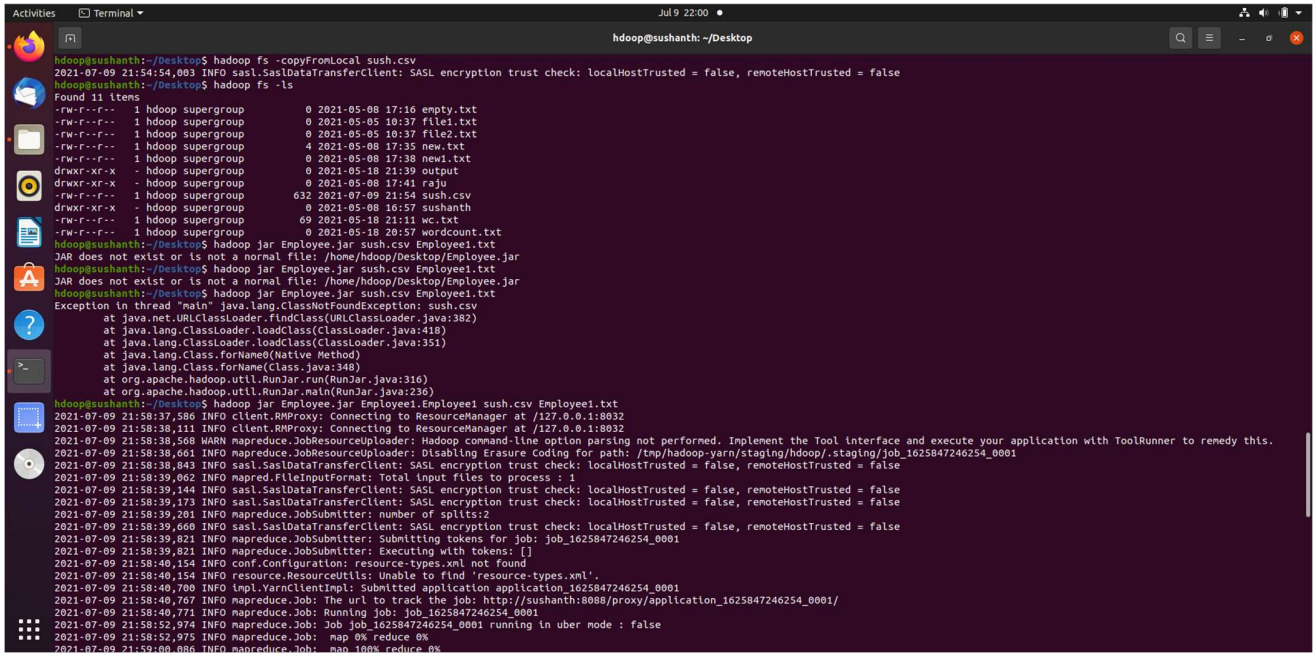
1) Total number of employees who works in ISE Department.

Snapshots of the java program:

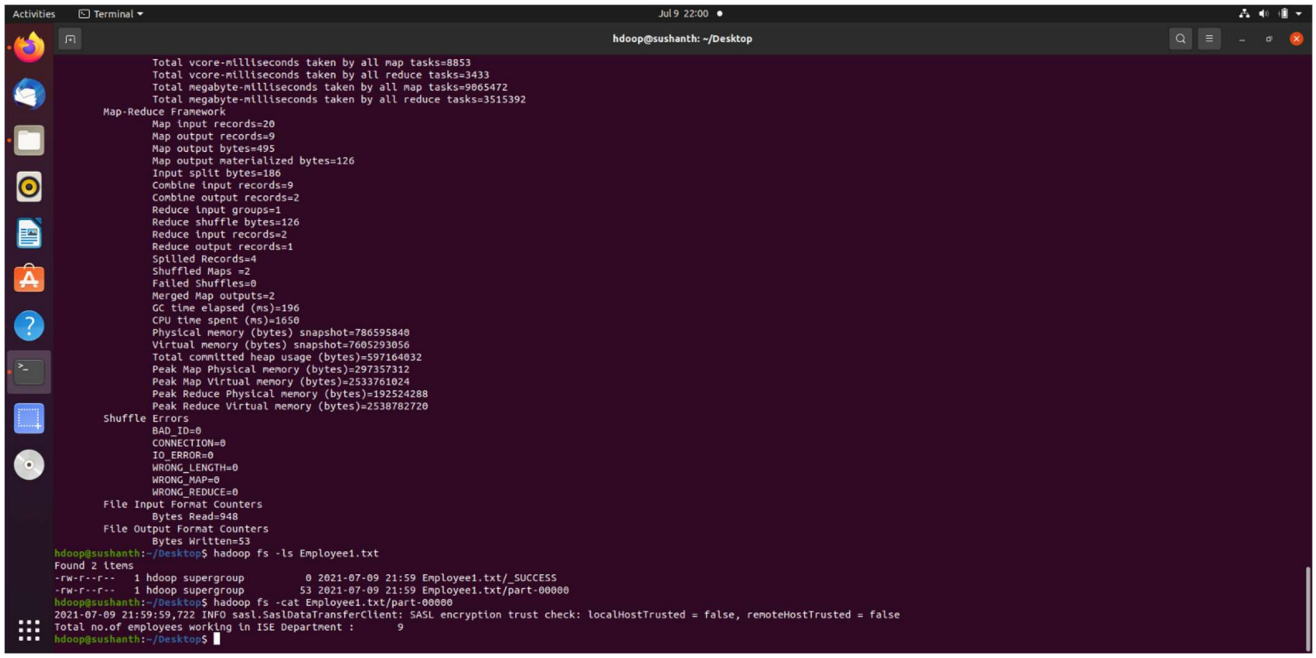




MapReducing through jar file:

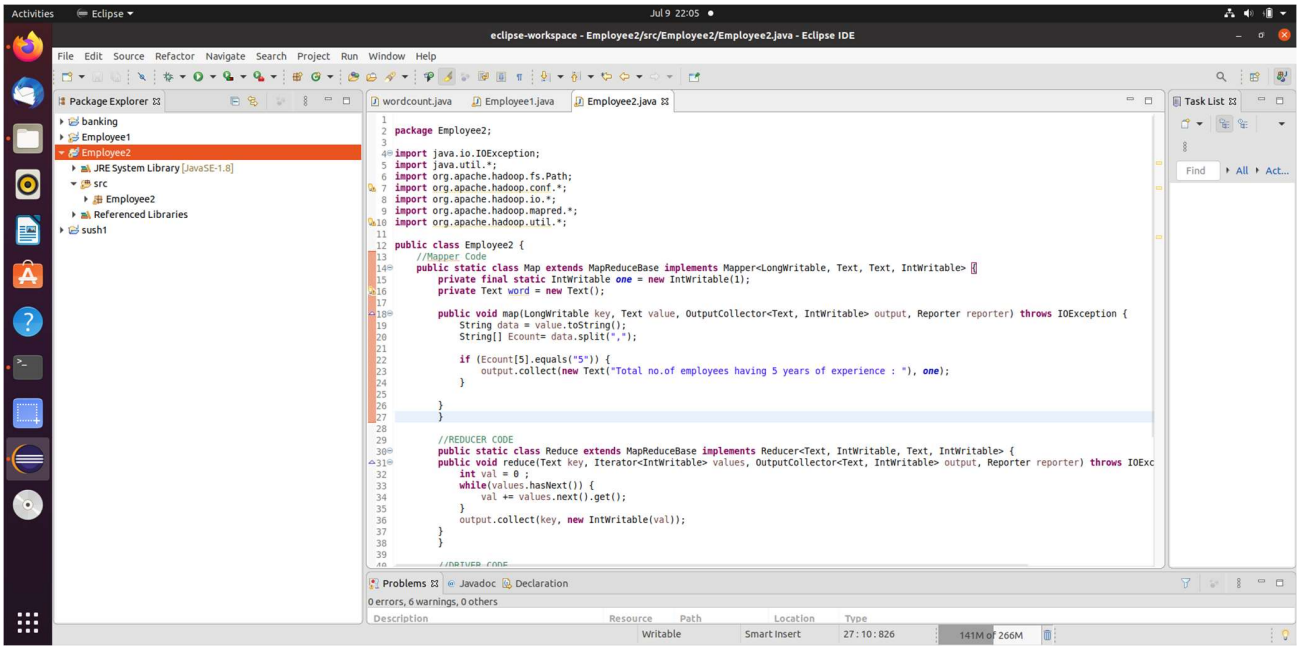


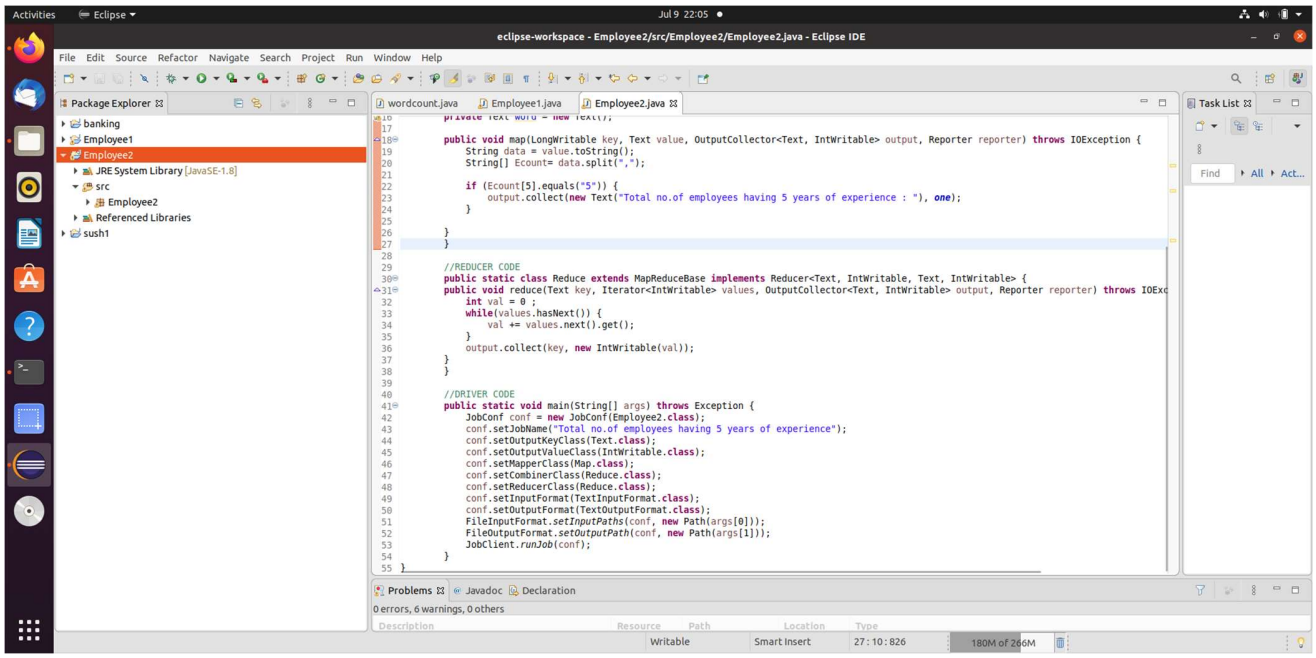
Output:



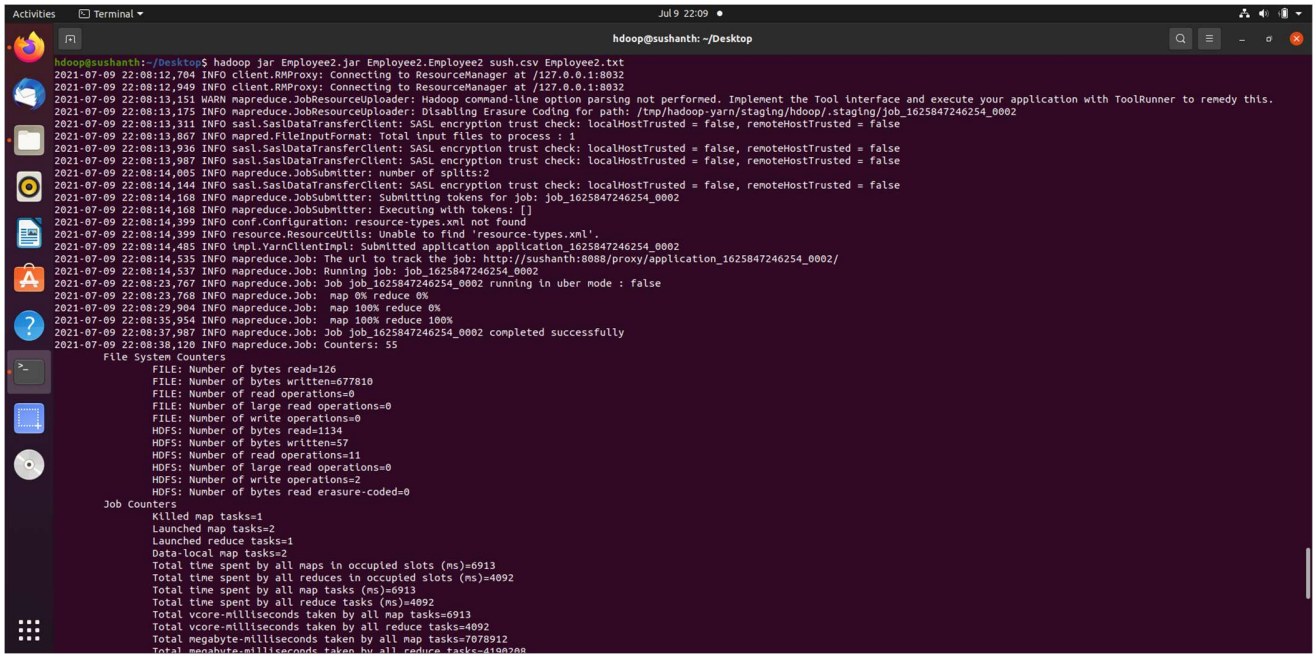
2) Total number of employees with experience = 5 years.

Snapshots of the java program:

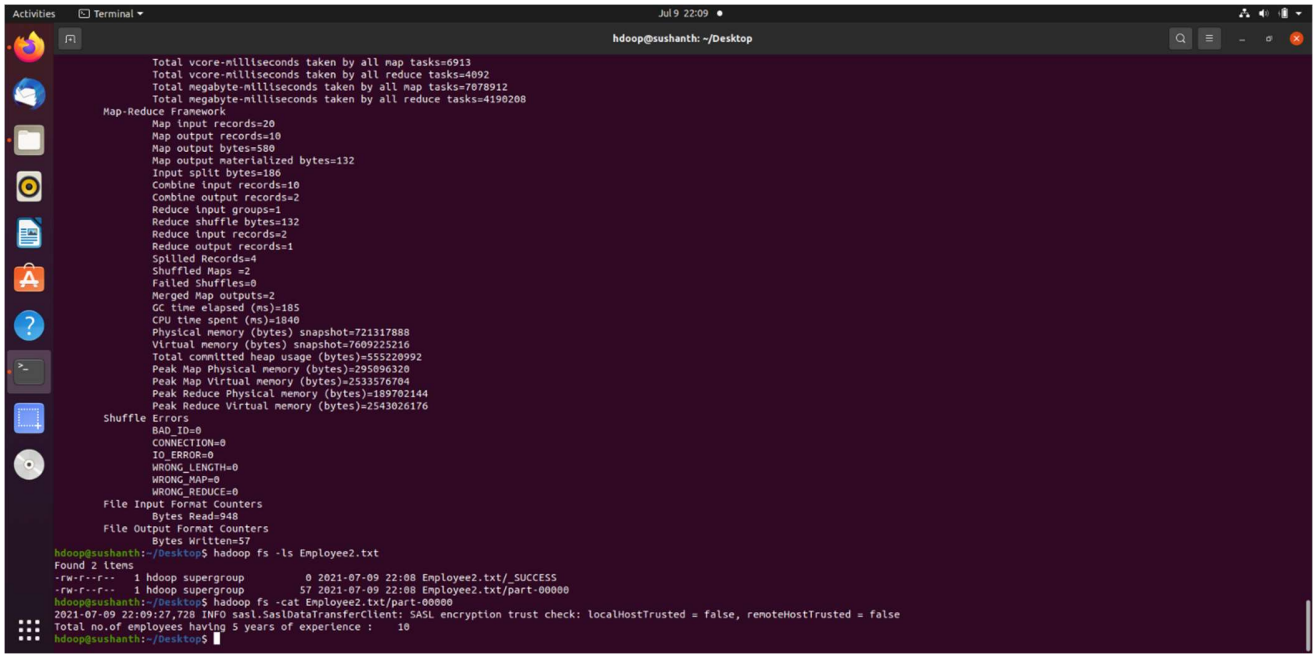




MapReducing through jar file:

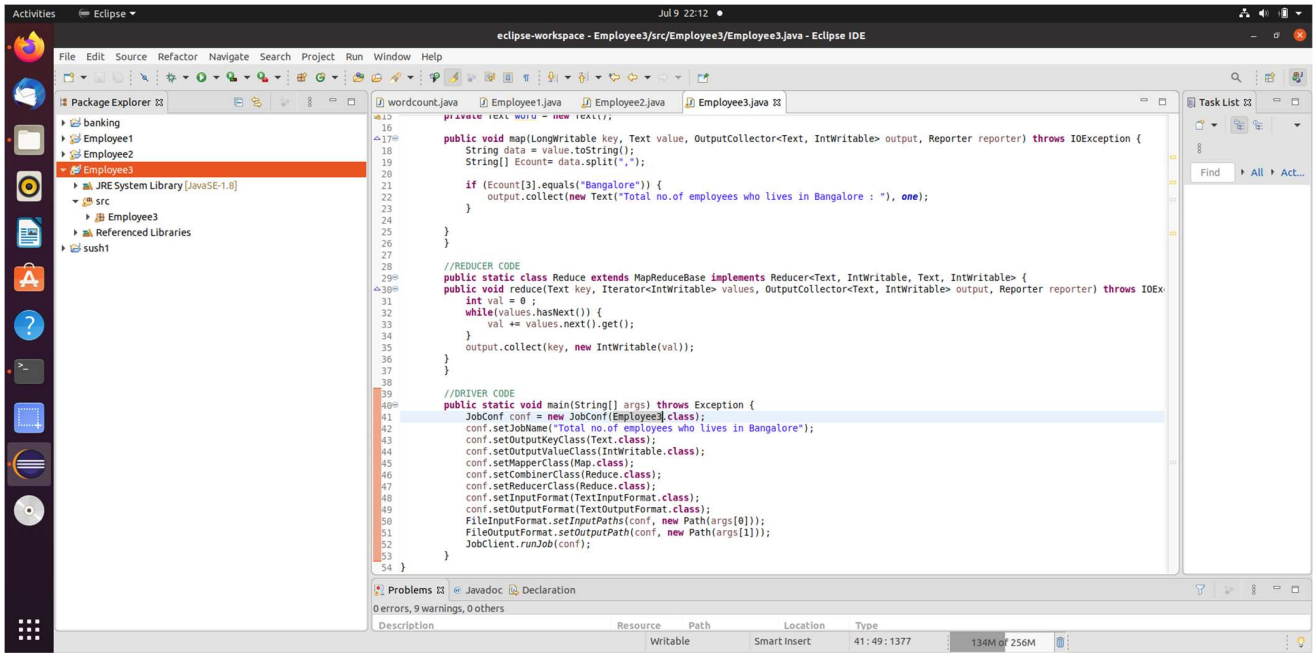


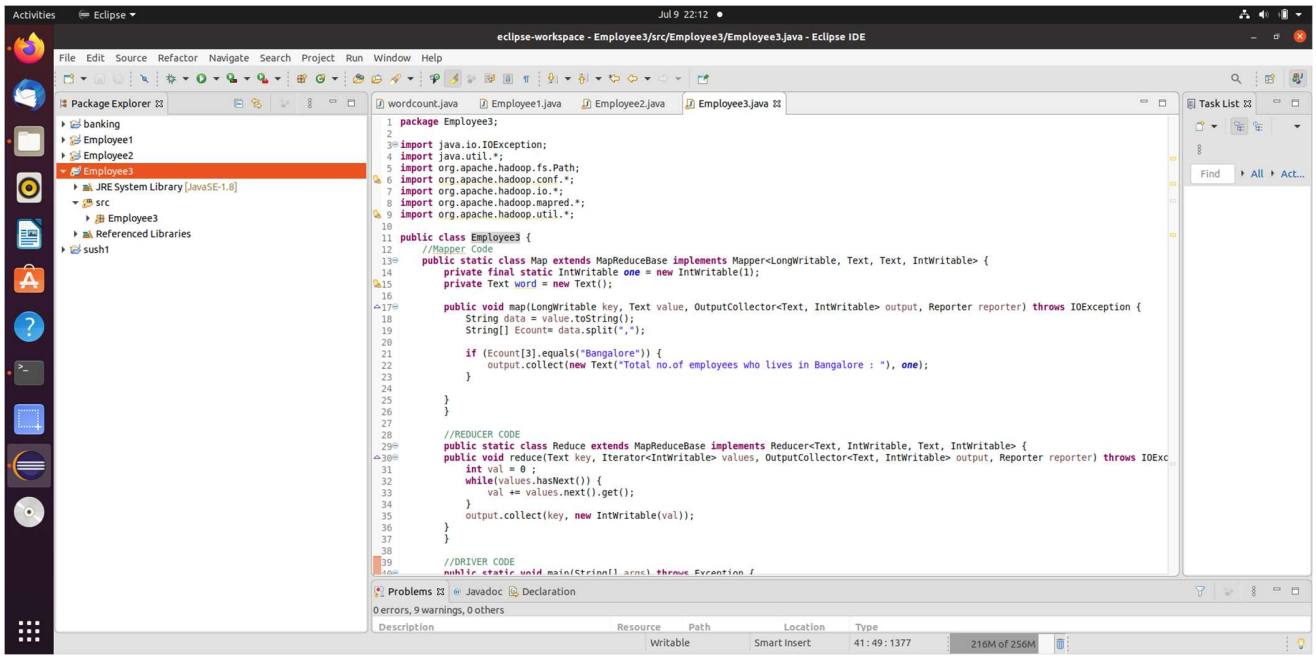
Output:



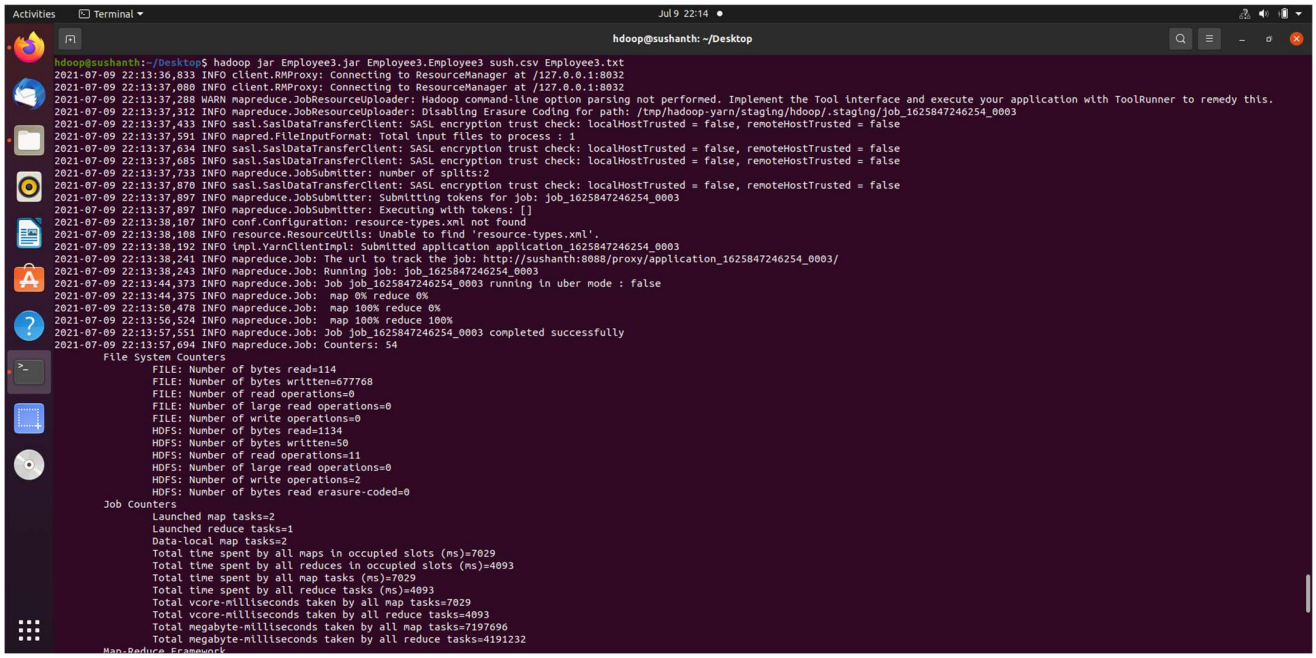
3) Count the number of employees who lives in Bangalore.

Snapshots of the java program:

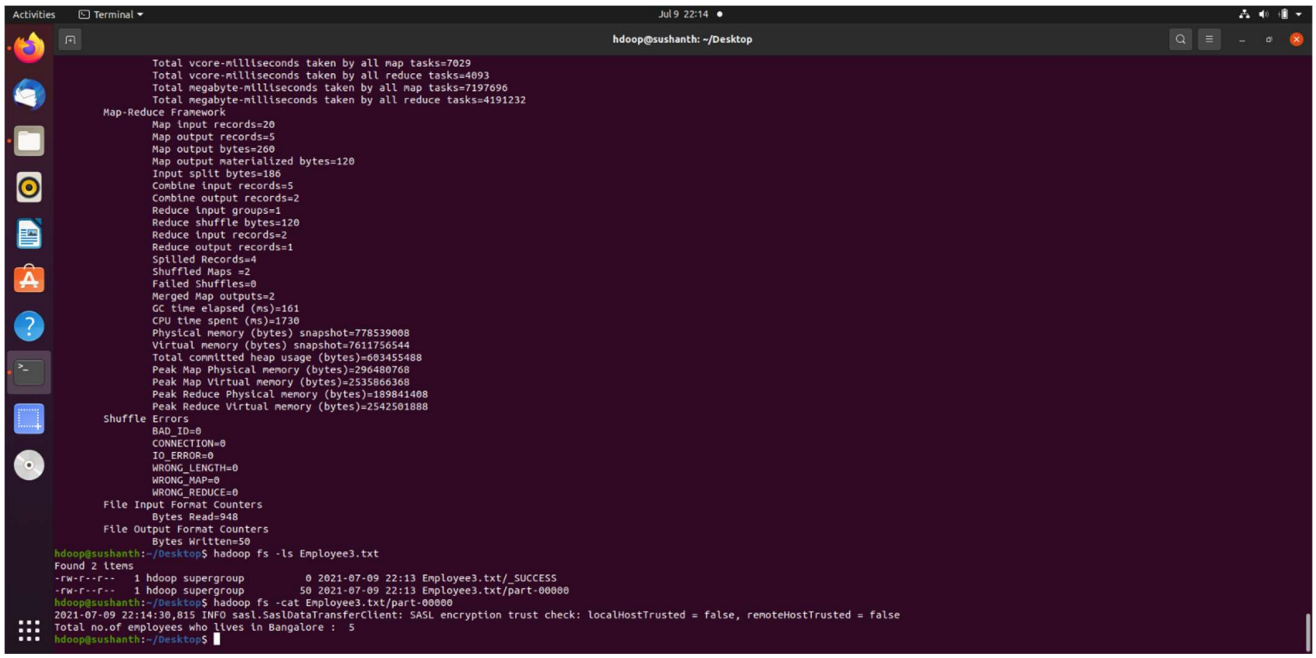




MapReducing through jar file:



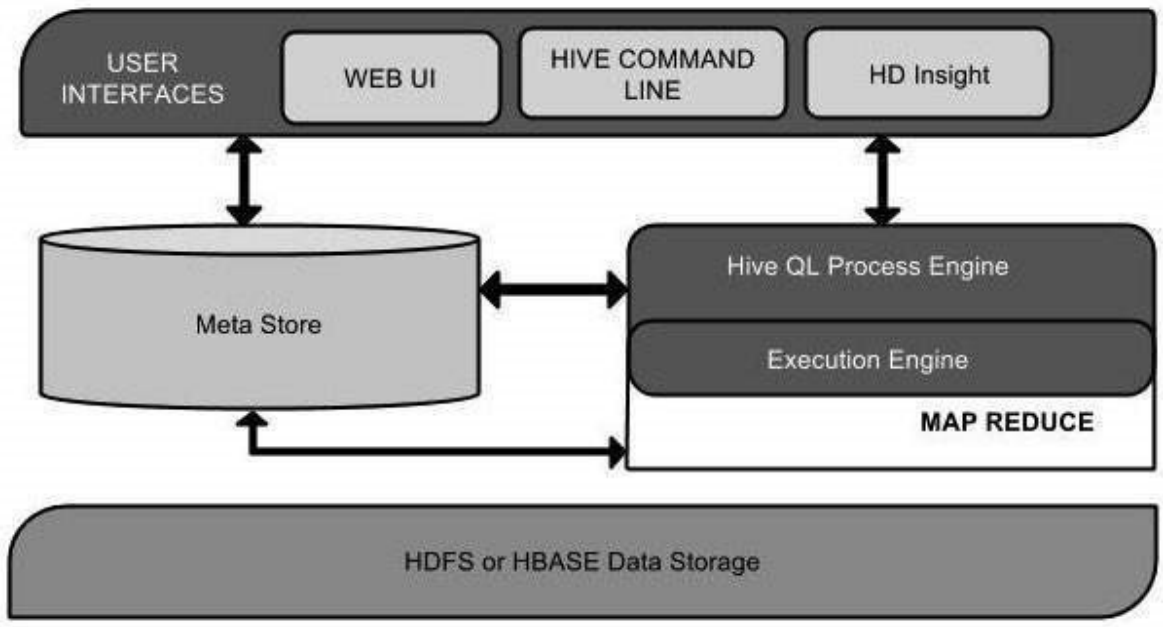
Output:



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

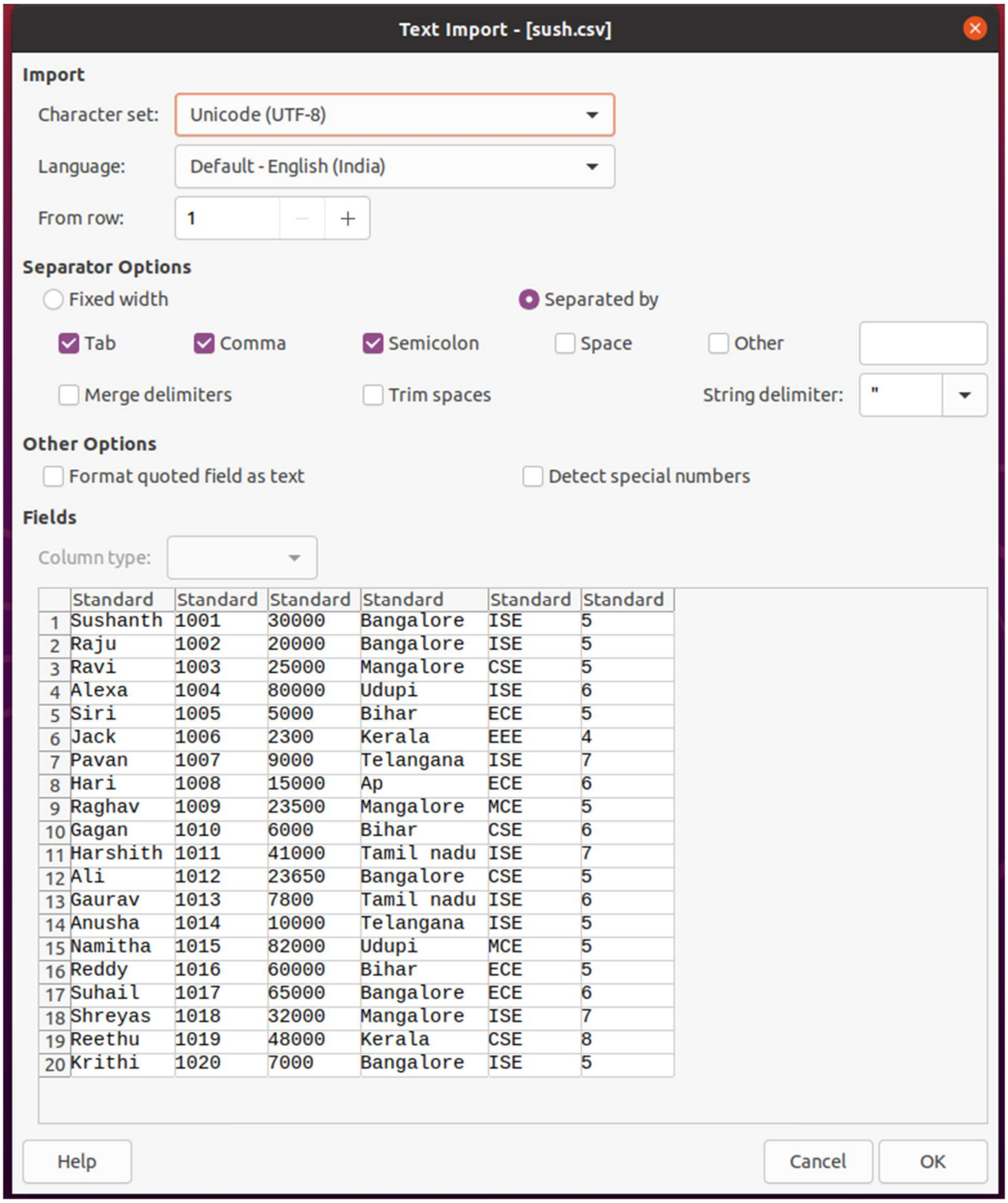
Use the HiveQL language to perform the following Query based Map-reduce operations,

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):



	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

2) Creating database EmployeeDB and Creating table employee :

```
hdoop@sushanth:~/apache-hive-3.1.2-bin/conf$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hdoop/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/hdoop/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-fd30-4e55-9acd-6573b37c4ceb

Logging initialized using configuration in jar:file:/home/hdoop/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-aedcb4a0f11a
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/hdoop/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
Hari    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    7800.0  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 = 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_0001, Tracking URL = http://sushanth:8088/proxy/application_1625480980760_0001/
Kill Command = /home/hdoop/hadoop-3.2.1/bin/mapred job  -kill job_1625480980760_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-07-05 16:19:15,512 Stage-1 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        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
Hari    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,
   a. Rename the table name to “Emp”.
   b. 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
Ravi    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 16:40:00,234 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
Ali     1012
Anu     1024
Anusha  1014
Gagan   1010
Gaurav  1013
Hari    1008
Harshith        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_20210705164206_3ca96b06-3169-4155-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/mapred job  -kill job_1625480980760_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-07-05 16:48:32,179 Stage-1 map = 0%,  reduce = 0%
2021-07-05 16:48:57,871 Stage-1 map = 67%,  reduce = 0%, Cumulative CPU 15.93 sec
2021-07-05 16:49:03,009 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 16.13 sec
2021-07-05 16:49:21,504 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_16-48-19_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/mapred job  -kill job_1625480980760_0006
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2021-07-05 16:55:11,286 Stage-1 map = 0%,  reduce = 0%
2021-07-05 16:55:21,819 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 4.96 sec
2021-07-05 16:55:32,087 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 10.72 sec
2021-07-05 16:55:44,451 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
Krithi  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
```

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/hdoop/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/hdoop/2df1eb88-fd30-4e55-9acd-6573b37c4ceb/hive_2021-07-05_16-57-30_543_8296294841225471837-1/-local-10004/HashTable-Stage-3/MapJoin-mapfil
e01--.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/hdoop/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 17:00:40,088 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
Harshith        ISE     1
Ali     CSE     3
Gaurav  ISE     1
Anusha  ISE     1
Namitha MCE     2
Reddy   ECE     5
Suhail  ECE     5
Shreyas ISE     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/hdoop/2df1eb88-fd30-4e55-9acd-6573b37c4ceb/hive_2021-07-05_17-03-51_623_3353144385287407466-1/-local-10004/HashTable-Stage-3/MapJoin-mapfil
e10--.hashtable (704 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/hdoop/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
Harshith        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.ht m

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