Hindi Vidya Prachar Samiti's

RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ARTS, SCIENCE & COMMERCE

(EMPOWERED AUTONOMOUS)

Big Data Analytics

Name: Siddhi Shashikant Chavan
Roll No.: 711
Class: MSc Data Science and Artificial Intelligence Part-I

Hindi Vidya Prachar Samiti’s Ramniranjan Jhunjhunwala College of Arts, Science and Commerce

Department of Data Science and Artificial Intelligence

CERTIFICATE

This is to certify Ms. Siddhi Shashikant Chavan of Msc. Data Science and Artificial Intelligence Roll No 711 has successfully completed the practical of Big Data Analytics during the Academic Year 2023-2024.

Date :

(Prof. Mujtaba Shaikh)

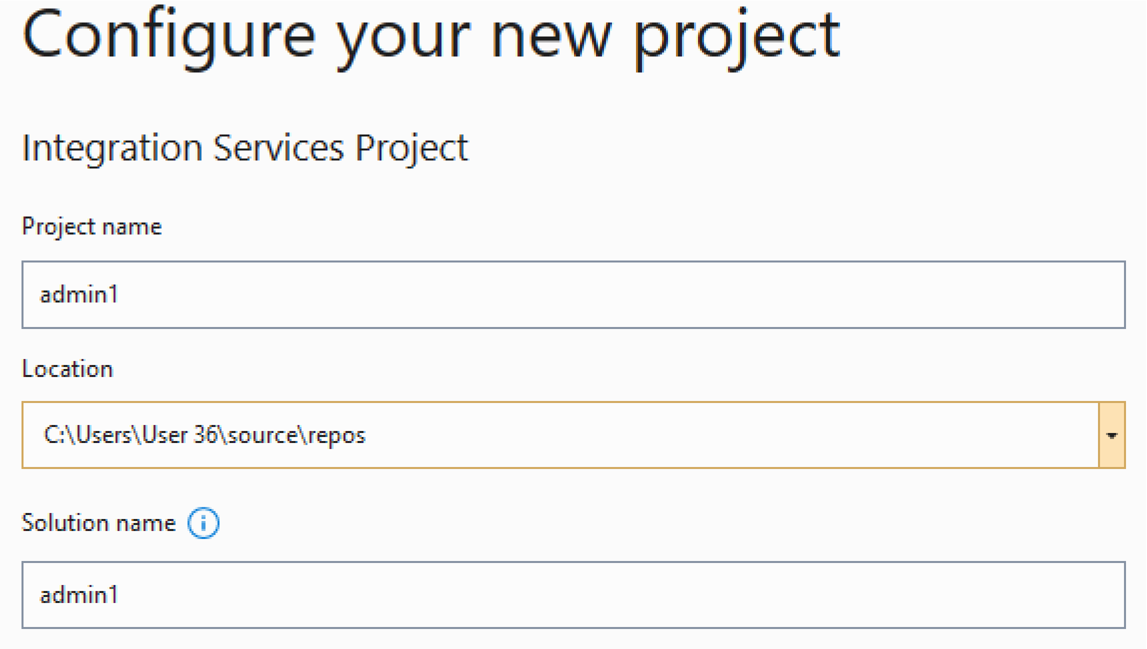
Prof-In-Charge **External Examiner**

INDEX

Practical 1

Perform ETL on student data and store transformed information in Data Warehouse.

Create dataset
Use Visual Studio 2019
Create a new project
Integration Services project
Project name



Configure your new project

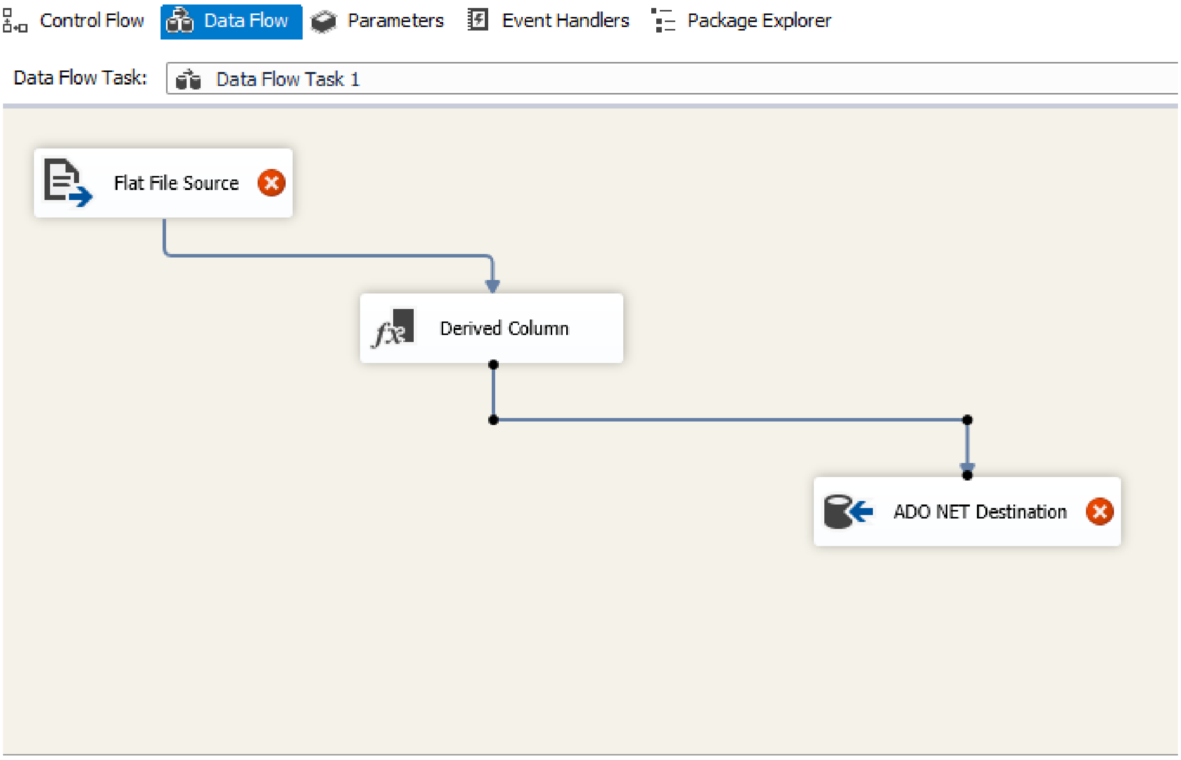
Integration Services Project

Project name: admin1

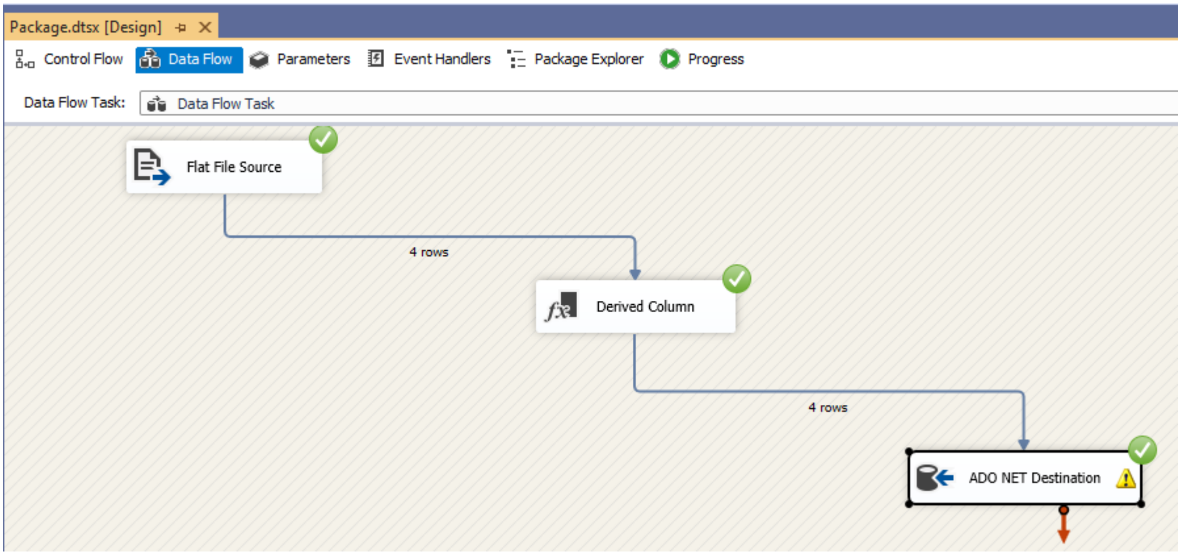
Location: C:\Users\User 36\source\repos

Solution name: admin1

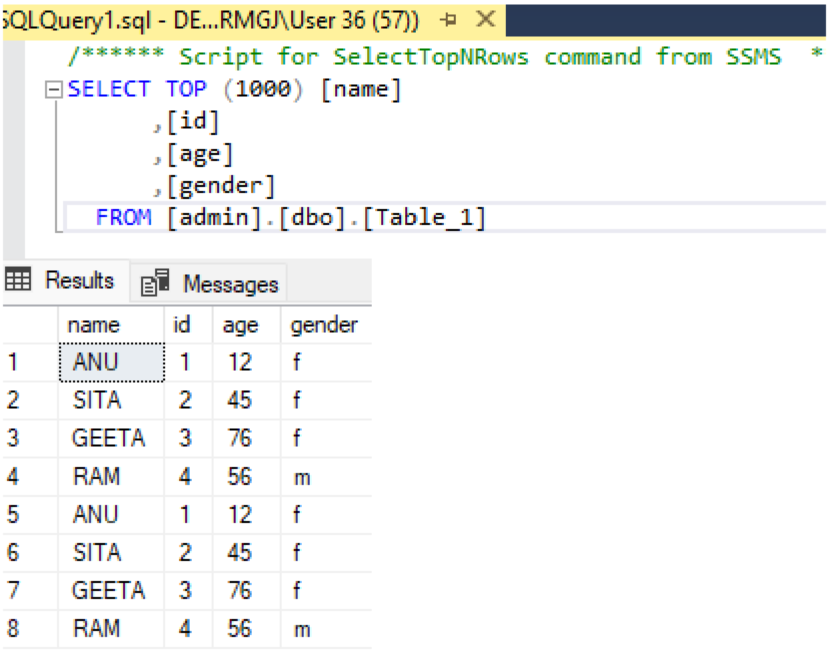
Data Flow Task
Flat File
Derived Column
ADO NET Destination



Flat File source
-Connection manager-New-Browser-Preview Derived Column
-Derived Column Name-Derived Column-expression
Microsoft SQL Server Management Studio
Create Database
Create Table
ADO NET Destination
-Connection Manager-New



Select Top 1000 Rows



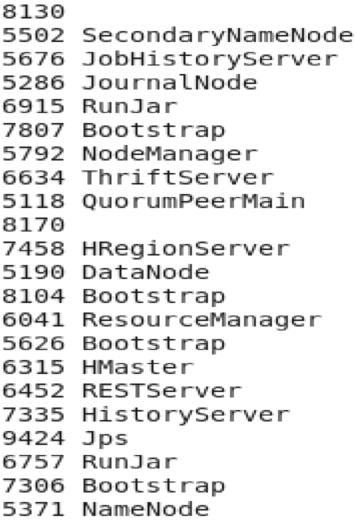
```
/****** Script for SelectTopNRows command from SSMS  *
SELECT TOP (1000) [name]
      ,[id]
      ,[age]
      ,[gender]
  FROM [admin].[dbo].[Table_1]
```

	name	id	age	gender
1	ANU	1	12	f
2	SITA	2	45	f
3	GEETA	3	76	f
4	RAM	4	56	m
5	ANU	1	12	f
6	SITA	2	45	f
7	GEETA	3	76	f
8	RAM	4	56	m

Practical 2

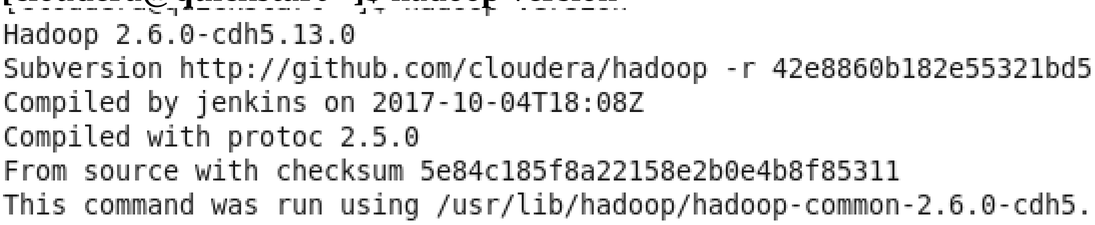
Perform basic commands on Hadoop distributed file system.

[cloudera@quickstart ~]$ sudo su
[root@quickstart cloudera]# jps



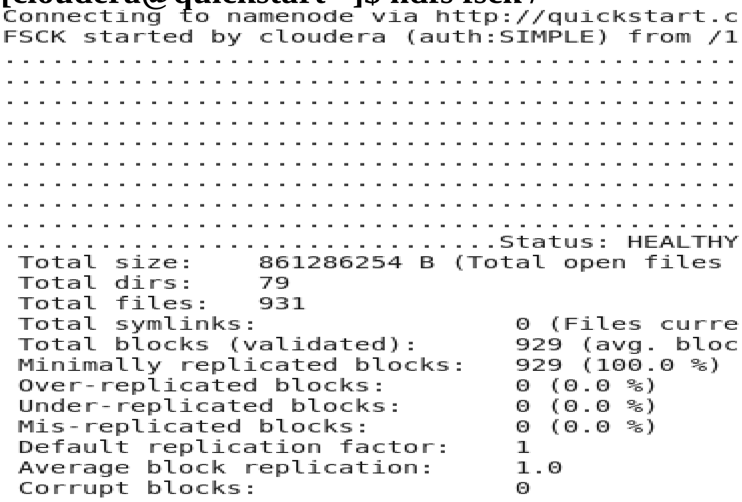
```
8130
5502 SecondaryNameNode
5676 JobHistoryServer
5286 JournalNode
6915 RunJar
7807 Bootstrap
5792 NodeManager
6634 ThriftServer
5118 QuorumPeerMain
8170
7458 HRegionServer
5190 DataNode
8104 Bootstrap
6041 ResourceManager
5626 Bootstrap
6315 HMaster
6452 RESTServer
7335 HistoryServer
9424 Jps
6757 RunJar
7306 Bootstrap
5371 NameNode
```

[root@quickstart cloudera]# su cloudera
[cloudera@quickstart ~]$ hadoop version



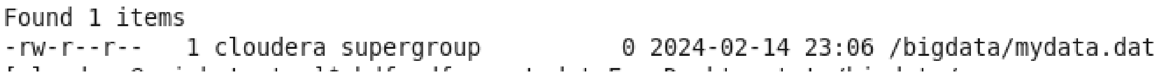
```
Hadoop 2.6.0-cdh5.13.0
Subversion http://github.com/cloudera/hadoop -r 42e8860b182e55321bd5
Compiled by jenkins on 2017-10-04T18:08Z
Compiled with protoc 2.5.0
From source with checksum 5e84c185f8a22158e2b0e4b8f85311
This command was run using /usr/lib/hadoop/hadoop-common-2.6.0-cdh5.
```

[cloudera@quickstart ~]$ hdfs fsck /



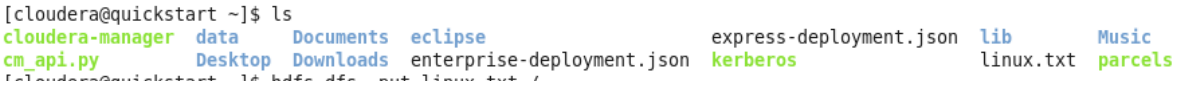
```
Connecting to namenode via http://quickstart.c
FSCK started by cloudera (auth:SIMPLE) from /1
..............................................
..............................................
..............................................
..............................................
..............................................
..............................................
..............................................
..............................................
..............................................
...............................Status: HEALTHY
 Total size:     861286254 B (Total open files
 Total dirs:     79
 Total files:    931
 Total symlinks:                0 (Files curre
 Total blocks (validated):      929 (avg. bloc
 Minimally replicated blocks:   929 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Corrupt blocks:                0
```

[cloudera@quickstart ~]$ hdfs dfs -mkdir /bigdata
[cloudera@quickstart ~]$ hdfs dfs -touchz /bigdata/mydata.dat
[cloudera@quickstart ~]$ hdfs dfs -ls /bigdata/



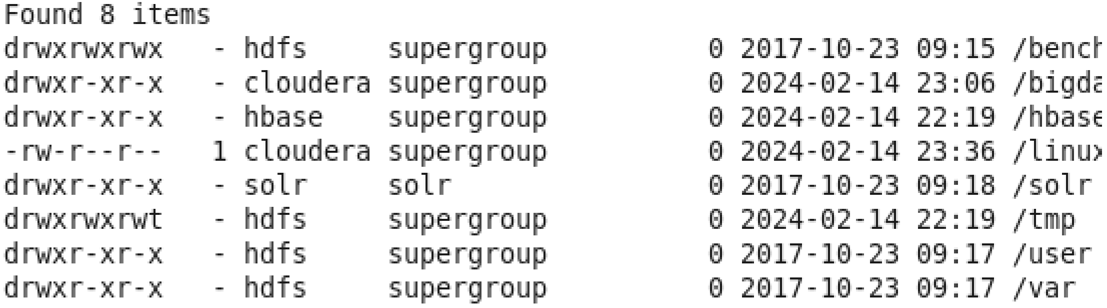
```
Found 1 items
-rw-r--r--   1 cloudera supergroup          0 2024-02-14 23:06 /bigdata/mydata.dat
```

[cloudera@quickstart ~]$ touch linux.txt



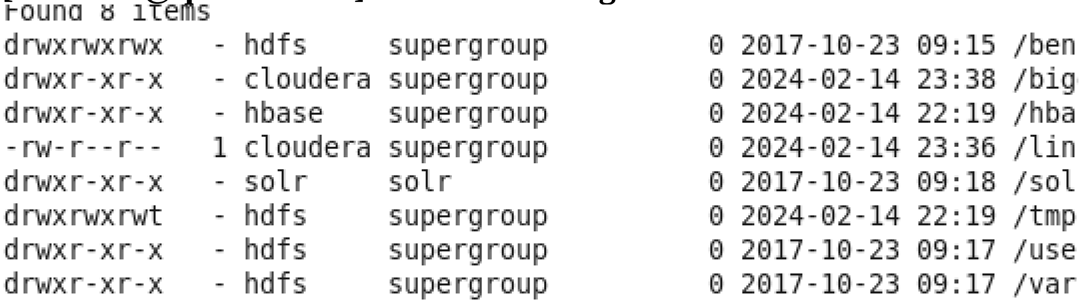
```
[cloudera@quickstart ~]$ ls
cloudera-manager  data     Documents  eclipse                    express-deployment.json  lib        Music
cm_api.py         Desktop  Downloads  enterprise-deployment.json  kerberos                 linux.txt  parcels
```

[cloudera@quickstart ~]$ hdfs dfs -put linux.txt /
[cloudera@quickstart ~]$ hdfs dfs -ls /



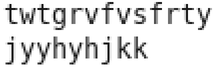
```
Found 8 items
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /bench
drwxr-xr-x   - cloudera supergroup          0 2024-02-14 23:06 /bigda
drwxr-xr-x   - hbase    supergroup          0 2024-02-14 22:19 /hbase
-rw-r--r--   1 cloudera supergroup          0 2024-02-14 23:36 /linux
drwxr-xr-x   - solr     solr                0 2017-10-23 09:18 /solr
drwxrwxrwt   - hdfs     supergroup          0 2024-02-14 22:19 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
```

[cloudera@quickstart ~]$ hdfs dfs -put linux.txt /bigdata
[cloudera@quickstart ~]$ hdfs dfs -ls /bigdata



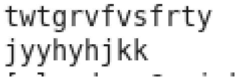
```
Found 8 items
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /ben
drwxr-xr-x   - cloudera supergroup          0 2024-02-14 23:38 /big
drwxr-xr-x   - hbase    supergroup          0 2024-02-14 22:19 /hba
-rw-r--r--   1 cloudera supergroup          0 2024-02-14 23:36 /lin
drwxr-xr-x   - solr     solr                0 2017-10-23 09:18 /sol
drwxrwxrwt   - hdfs     supergroup          0 2024-02-14 22:19 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /use
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
```

[cloudera@quickstart ~]$ hdfs dfs -appendToFile - /bigdata/data.bat



```
twtgrvfvsfrty
jyyhyhjkk
```

Type Control D
[cloudera@quickstart ~]$ hdfs dfs -cat /bigdata/data.bat

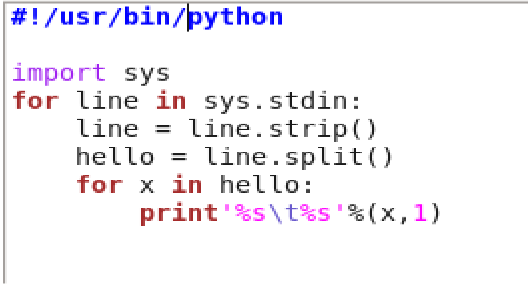


```
twtgrvfvsfrty
jyyhyhjkk
```

Practical 3

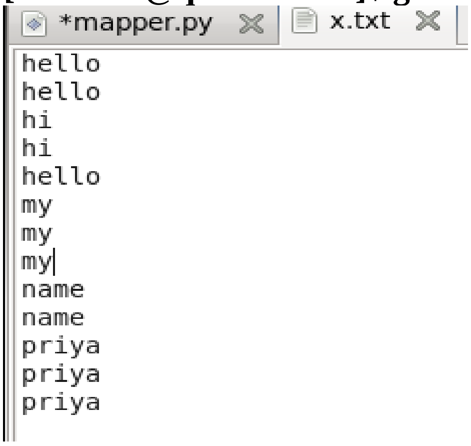
Implementation of MapReduce program to find word counts.

[cloudera@quickstart ~]$ gedit mapper.py



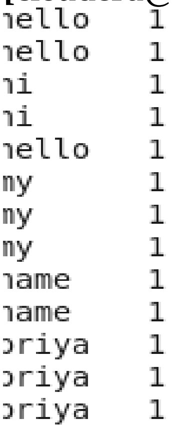
```
#!/usr/bin/python

import sys
for line in sys.stdin:
    line = line.strip()
    hello = line.split()
    for x in hello:
        print'%s\t%s'%(x,1)
```

[cloudera@quickstart ~]$ gedit x.txt



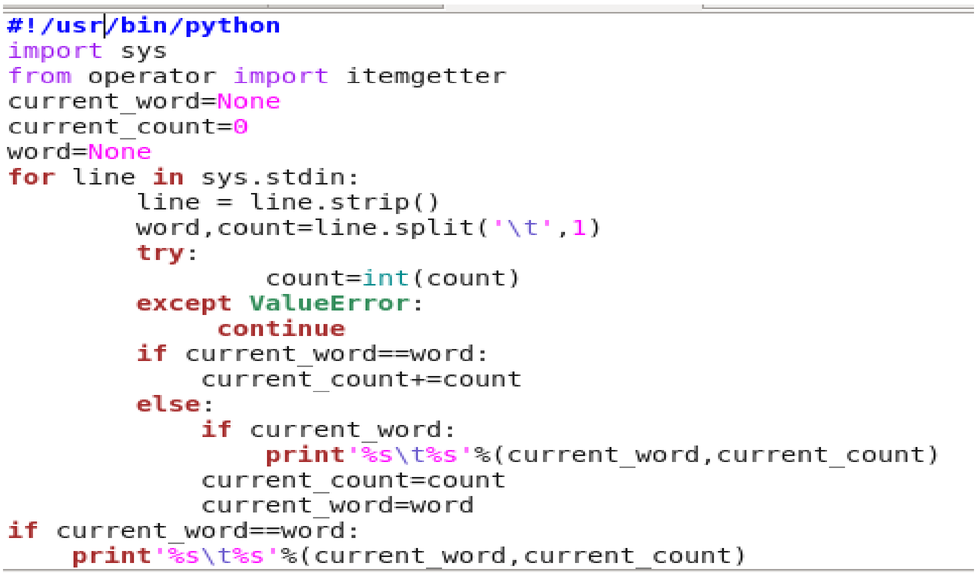
```
hello
hello
hi
hi
hello
my
my
my
name
name
priya
priya
priya
```

[cloudera@quickstart ~]$ cat x.txt | python mapper.py



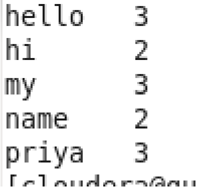
```
hello   1
hello   1
hi      1
hi      1
hello   1
my      1
my      1
my      1
name    1
name    1
priya   1
priya   1
priya   1
```

[cloudera@quickstart ~]$ gedit reducer.py



```
#!/usr/bin/python
import sys
from operator import itemgetter
current_word=None
current_count=0
word=None
for line in sys.stdin:
        line = line.strip()
        word,count=line.split('\t',1)
        try:
                count=int(count)
        except ValueError:
             continue
        if current_word==word:
            current_count+=count
        else:
            if current_word:
                print'%s\t%s'%(current_word,current_count)
            current_count=count
            current_word=word
if current_word==word:
    print'%s\t%s'%(current_word,current_count)
```

cloudera@quickstart ~]$ cat x.txt | python mapper.py| sort | python reducer.py

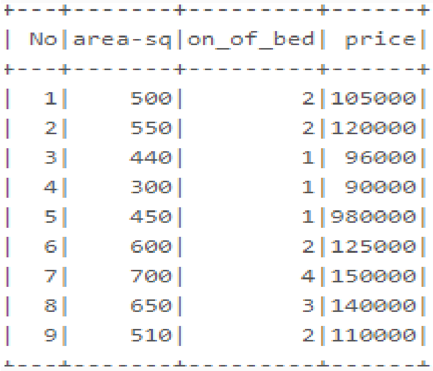


```
hello   3
hi      2
my      3
name    2
priya   3
[cloudera@qu
```

Practical 4

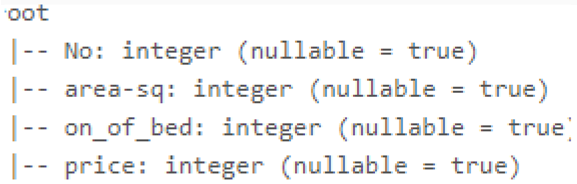
Perform linear regression on the given dataset using PySpark on Databricks.

```
from pyspark.sql import SparkSession
ss=SparkSession.builder.getOrCreate()
df=ss.read.csv('/FileStore/tables/housesales.csv',inferSchema=True,header=True)
df.show()
```



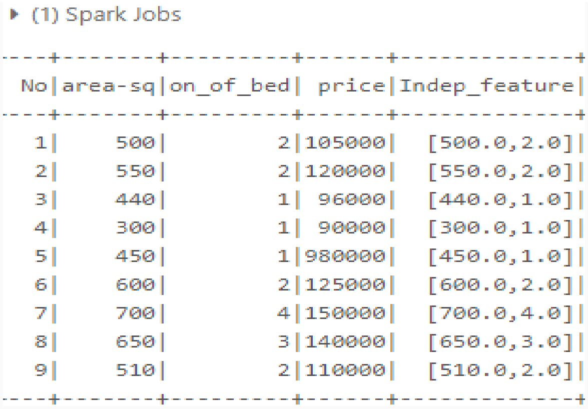
```
+---+-------+---------+------+
| No|area-sq|on_of_bed| price|
+---+-------+---------+------+
|  1|    500|        2|105000|
|  2|    550|        2|120000|
|  3|    440|        1| 96000|
|  4|    300|        1| 90000|
|  5|    450|        1|980000|
|  6|    600|        2|125000|
|  7|    700|        4|150000|
|  8|    650|        3|140000|
|  9|    510|        2|110000|
+---+-------+---------+------+
```

```
df.printSchema()
```



```
oot
 |-- No: integer (nullable = true)
 |-- area-sq: integer (nullable = true)
 |-- on_of_bed: integer (nullable = true
 |-- price: integer (nullable = true)
```

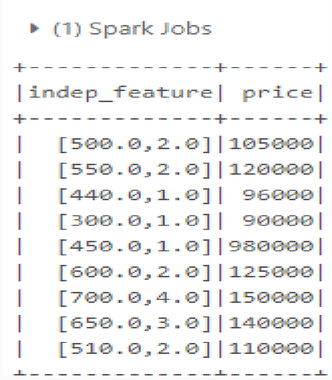
```
from pyspark.ml.feature import VectorAssembler
#fa =feature assemble
fa=VectorAssembler(inputCols=["area-sq","on_of_bed"],outputCol="Indep_feature")
output=fa.transform(df)
output.show()
```



```
▸ (1) Spark Jobs

----+-------+---------+------+-------------+
 No|area-sq|on_of_bed| price|Indep_feature|
----+-------+---------+------+-------------+
   1|    500|        2|105000|  [500.0,2.0]|
   2|    550|        2|120000|  [550.0,2.0]|
   3|    440|        1| 96000|  [440.0,1.0]|
   4|    300|        1| 90000|  [300.0,1.0]|
   5|    450|        1|980000|  [450.0,1.0]|
   6|    600|        2|125000|  [600.0,2.0]|
   7|    700|        4|150000|  [700.0,4.0]|
   8|    650|        3|140000|  [650.0,3.0]|
   9|    510|        2|110000|  [510.0,2.0]|
----+-------+---------+------+-------------+
```

find_data=output.select("indep_feature","price")

find_data.show()



```
▸ (1) Spark Jobs

+-------------+------+
|indep_feature| price|
+-------------+------+
|  [500.0,2.0]|105000|
|  [550.0,2.0]|120000|
|  [440.0,1.0]| 96000|
|  [300.0,1.0]| 90000|
|  [450.0,1.0]|980000|
|  [600.0,2.0]|125000|
|  [700.0,4.0]|150000|
|  [650.0,3.0]|140000|
|  [510.0,2.0]|110000|
+-------------+------+
```

from pyspark.ml.regression import LinearRegression

train_data,test_data=find_data.randomSplit([0.75,0.25])

reg=LinearRegression(featuresCol="indep_feature",labelCol="price")

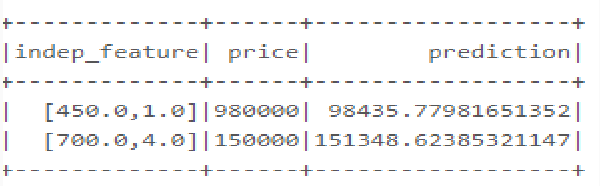
reg=reg.fit(train_data)

reg.coefficients

reg.intercept

pred_result=reg.evaluate(test_data)

pred_result.predictions.show()



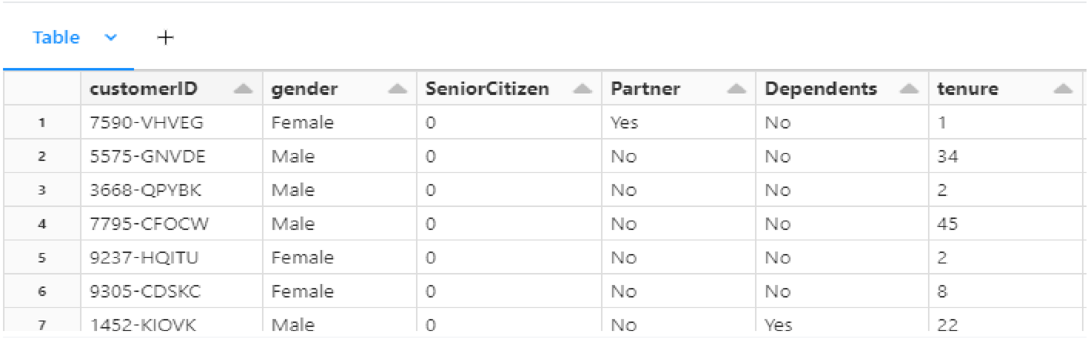
```
+-------------+------+------------------+
|indep_feature| price|        prediction|
+-------------+------+------------------+
|  [450.0,1.0]|980000| 98435.77981651352|
|  [700.0,4.0]|150000|151348.62385321147|
+-------------+------+------------------+
```

pred_result.meanAbsoluteError,pred_result.meanSquaredError

Practical 5

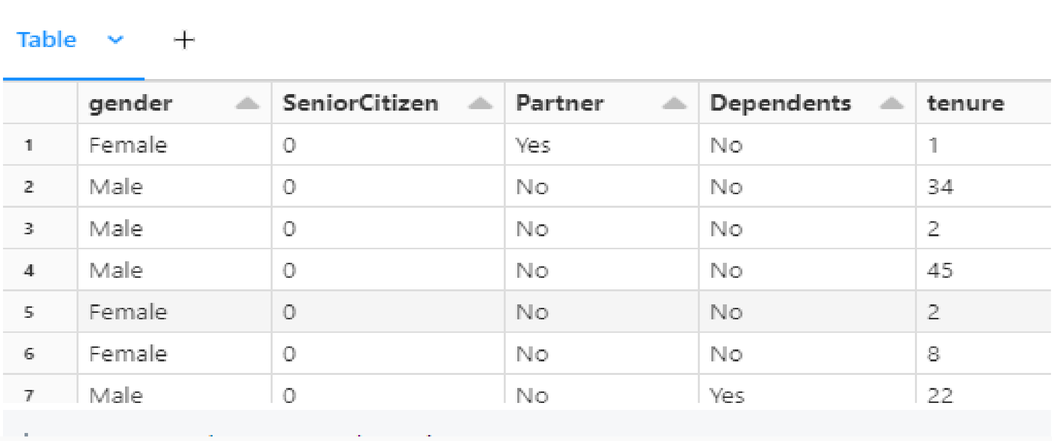
Perform customer Churn analysis on the given dataset using ML Algorithm in PySpark.

```
from pyspark.sql import SparkSession
ss=SparkSession.builder.getOrCreate()
ss
df=ss.read.csv(
"/FileStore/tables/WA_Fn_UseC__Telco_Customer_Churn_csv___WA_Fn_UseC_
_Telco_Customer_Churn-1.csv",inferSchema=True,header=True)
df.display()
```



	customerID	gender	SeniorCitizen	Partner	Dependents	tenure
1	7590-VHVEG	Female	0	Yes	No	1
2	5575-GNVDE	Male	0	No	No	34
3	3668-QPYBK	Male	0	No	No	2
4	7795-CFOCW	Male	0	No	No	45
5	9237-HQITU	Female	0	No	No	2
6	9305-CDSKC	Female	0	No	No	8
7	1452-KIOVK	Male	0	No	Yes	22

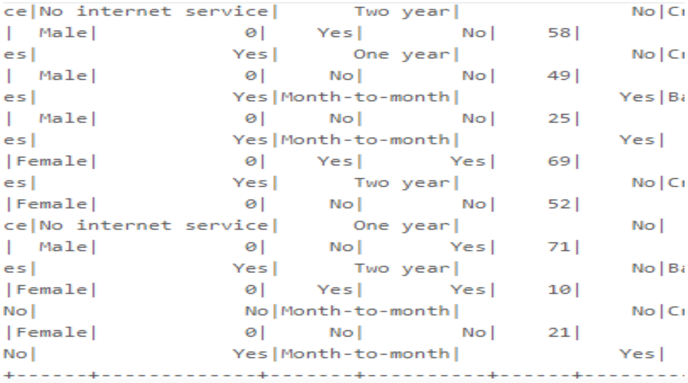
```
df.printSchema()
df=df.drop('customerID')
df.display()
```



	gender	SeniorCitizen	Partner	Dependents	tenure
1	Female	0	Yes	No	1
2	Male	0	No	No	34
3	Male	0	No	No	2
4	Male	0	No	No	45
5	Female	0	No	No	2
6	Female	0	No	No	8
7	Male	0	No	Yes	22

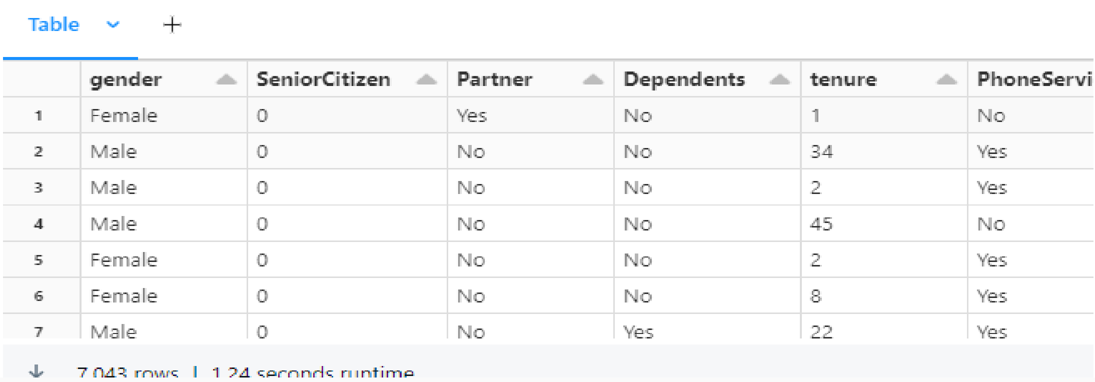
```
from pyspark.ml.feature import StringIndexer
gen=StringIndexer(inputCol='gender',outputCol='new_gender')
```

```
df.show()
```



```
ce|No internet service|      Two year|             No|C
|  Male|            0|    Yes|       No|    58|
es|                Yes|      One year|             No|C
|  Male|            0|     No|       No|    49|
es|                Yes|Month-to-month|            Yes|B
|  Male|            0|     No|       No|    25|
es|                Yes|Month-to-month|            Yes|
|Female|            0|    Yes|      Yes|    69|
es|                Yes|      Two year|             No|C
|Female|            0|     No|       No|    52|
ce|No internet service|      One year|             No|
|  Male|            0|     No|      Yes|    71|
es|                Yes|      Two year|             No|B
|Female|            0|    Yes|      Yes|    10|
No|                 No|Month-to-month|             No|C
|Female|            0|     No|       No|    21|
No|                Yes|Month-to-month|            Yes|
+------+-------------+-------+---------+------+---------
```

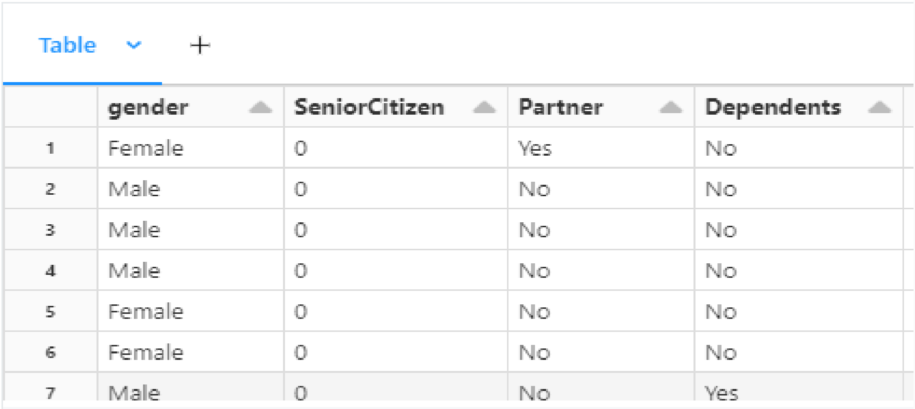
```
df.display()
```



	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneServi
1	Female	0	Yes	No	1	No
2	Male	0	No	No	34	Yes
3	Male	0	No	No	2	Yes
4	Male	0	No	No	45	No
5	Female	0	No	No	2	Yes
6	Female	0	No	No	8	Yes
7	Male	0	No	Yes	22	Yes

7,043 rows | 1.24 seconds runtime

```
from pyspark.ml.feature import VectorAssembler
trans_col=trans.fit(df).transform(df)
trans_col.display()
```

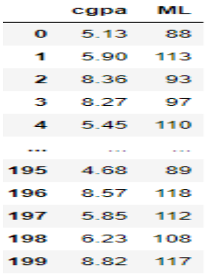


	gender	SeniorCitizen	Partner	Dependents
1	Female	0	Yes	No
2	Male	0	No	No
3	Male	0	No	No
4	Male	0	No	No
5	Female	0	No	No
6	Female	0	No	No
7	Male	0	No	Yes

Practical 6

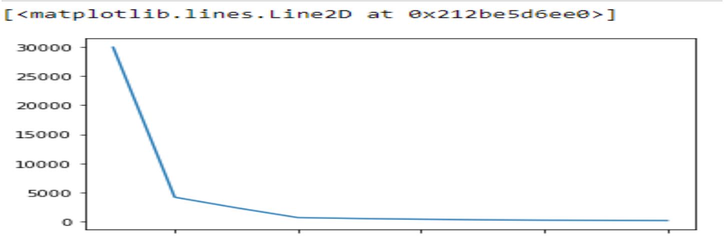
Implementation of Cluster analysis using K-Means Clustering Algorithm & also find the optimal number of cluster using elbow method.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
df=pd.read_csv("student.csv")
Df
```



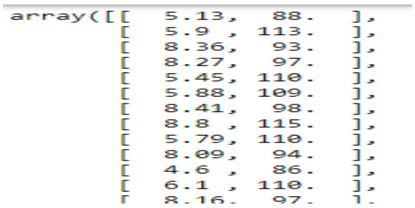
	cgpa	ML
0	5.13	88
1	5.90	113
2	8.36	93
3	8.27	97
4	5.45	110
...	...	...
195	4.68	89
196	8.57	118
197	5.85	112
198	6.23	108
199	8.82	117

```
plt.scatter(df[cgpa],df[ML] ,color="skyblue")
wss=[]
for i in range(1,11):
    km=KMeans(n_clusters=i)
    km.fit(df)
    wss.append(km.inertia_)
plt.plot(range(1,11),wss)
```



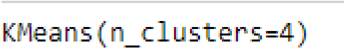
```
[<matplotlib.lines.Line2D at 0x212be5d6ee0>]
```

```
x=df.iloc[:,:].values
X
```



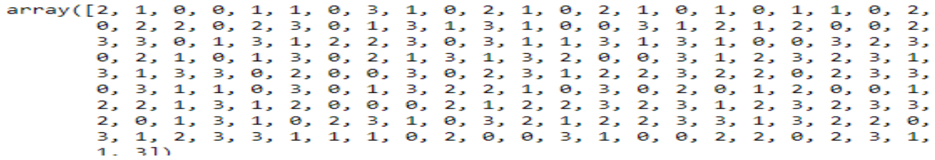
```
array([[   5.13,   88.   ],
       [   5.9 ,  113.   ],
       [   8.36,   93.   ],
       [   8.27,   97.   ],
       [   5.45,  110.   ],
       [   5.88,  109.   ],
       [   8.41,   98.   ],
       [   8.8 ,  115.   ],
       [   5.79,  110.   ],
       [   8.09,   94.   ],
       [   4.6 ,   86.   ],
       [   6.1 ,  110.   ],
       [   8.16,   97.   ],
```

```
km=KMeans(n_clusters=4)
km.fit(x)
```



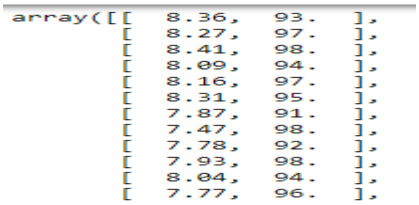
```
KMeans(n_clusters=4)
```

```
y_pred=km.predict(x)
y_pred
```



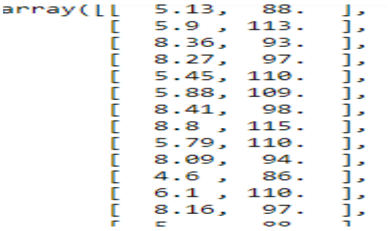
```
array([2, 1, 0, 0, 1, 1, 0, 3, 1, 0, 2, 1, 0, 2, 1, 0, 1, 0, 1, 1, 0, 2,
       0, 2, 2, 0, 2, 3, 0, 1, 3, 1, 3, 1, 0, 0, 3, 1, 2, 1, 2, 0, 0, 2,
       3, 3, 0, 1, 3, 1, 2, 2, 3, 0, 3, 1, 1, 3, 1, 3, 1, 0, 0, 3, 2, 3,
       0, 2, 1, 0, 1, 3, 0, 2, 1, 3, 1, 3, 2, 0, 0, 3, 1, 2, 3, 2, 3, 1,
       3, 1, 3, 3, 0, 2, 0, 0, 3, 0, 2, 3, 1, 2, 2, 3, 2, 2, 0, 2, 3, 3,
       0, 3, 1, 1, 0, 3, 0, 1, 3, 2, 2, 1, 0, 3, 0, 2, 0, 1, 2, 0, 0, 1,
       2, 2, 1, 3, 1, 2, 0, 0, 0, 2, 1, 2, 2, 3, 2, 3, 1, 2, 3, 2, 3, 3,
       2, 0, 1, 3, 1, 0, 2, 3, 1, 0, 3, 2, 1, 2, 2, 3, 3, 1, 3, 2, 2, 0,
       3, 1, 2, 3, 3, 1, 1, 1, 0, 2, 0, 0, 3, 1, 0, 0, 2, 2, 0, 2, 3, 1,
       1, 3])
```

```
x[y_pred==0]
```



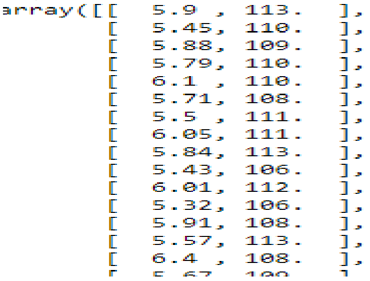
```
array([[  8.36,  93.   ],
       [  8.27,  97.   ],
       [  8.41,  98.   ],
       [  8.09,  94.   ],
       [  8.16,  97.   ],
       [  8.31,  95.   ],
       [  7.87,  91.   ],
       [  7.47,  98.   ],
       [  7.78,  92.   ],
       [  7.93,  98.   ],
       [  8.04,  94.   ],
       [  7.77,  96.   ],
```

```
X
```



```
array([[  5.13,  88.   ],
       [  5.9 , 113.   ],
       [  8.36,  93.   ],
       [  8.27,  97.   ],
       [  5.45, 110.   ],
       [  5.88, 109.   ],
       [  8.41,  98.   ],
       [  8.8 , 115.   ],
       [  5.79, 110.   ],
       [  8.09,  94.   ],
       [  4.6 ,  86.   ],
       [  6.1 , 110.   ],
       [  8.16,  97.   ],
```

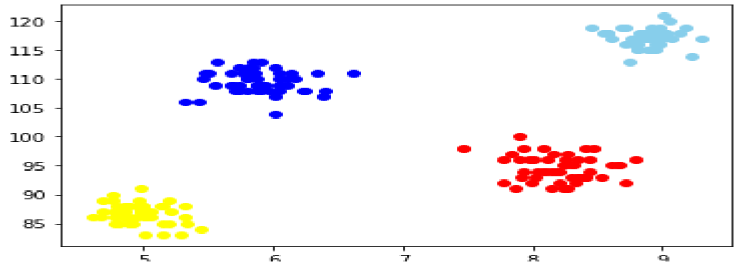
```
x[y_pred==1]
```



```
array([[  5.9 , 113.   ],
       [  5.45, 110.   ],
       [  5.88, 109.   ],
       [  5.79, 110.   ],
       [  6.1 , 110.   ],
       [  5.71, 108.   ],
       [  5.5 , 111.   ],
       [  6.05, 111.   ],
       [  5.84, 113.   ],
       [  5.43, 106.   ],
       [  6.01, 112.   ],
       [  5.32, 106.   ],
       [  5.91, 108.   ],
       [  5.57, 113.   ],
       [  6.4 , 108.   ],
```

```
plt.scatter(x[y_pred==0,0],x[y_pred==0,1] ,color="red")
plt.scatter(x[y_pred==1,0],x[y_pred==1,1] ,color="blue")
plt.scatter(x[y_pred==2,0],x[y_pred==2,1] ,color="yellow")
plt.scatter(x[y_pred==3,0],x[y_pred==3,1] ,color="skyblue")
```

```
<matplotlib.collections.PathCollection at 0x212bf
```

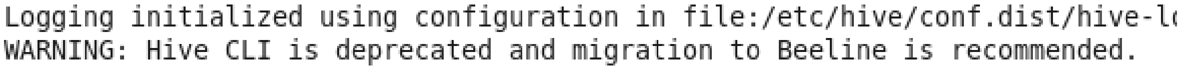


Practical 7

Demonstrate the working of internal and external table in Hive.

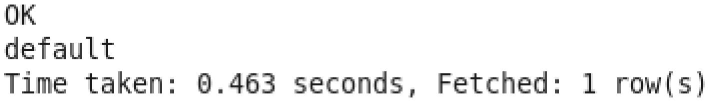
For first terminal

```
[cloudera@quickstart ~]$ hive home
```



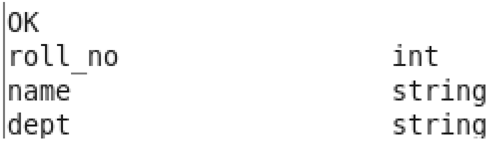
```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-l
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
```

```
hive> show databases;
```



```
OK
default
Time taken: 0.463 seconds, Fetched: 1 row(s)
```

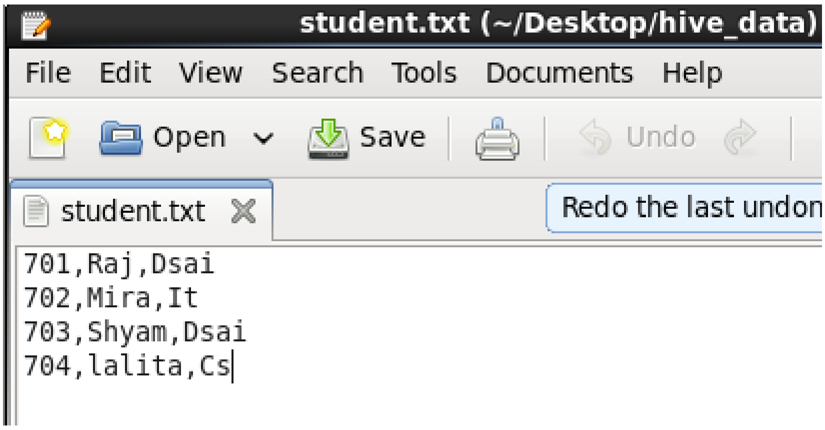
```
hive> create table student(roll_no int, name string,dept string);
hive> create database info;
hive> use info;
hive> create table student(roll_no int, name string,dept string);
hive> drop table student;
hive> use default;
hive> drop table student;
hive> use info;
hive> create table student(roll_no int, name string,dept string)
    > row format delimited
    > fields terminated by ',';
hive> describe student;
```



```
OK
roll_no                 int
name                    string
dept                    string
```

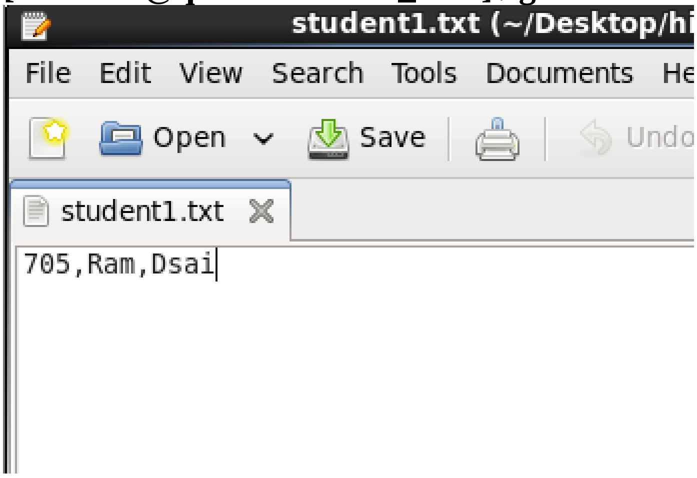
For second terminal

```
[cloudera@quickstart ~]$ dir
[cloudera@quickstart ~]$ cd Desktop
[cloudera@quickstart Desktop]$ mkdir hive_data
[cloudera@quickstart Desktop]$ cd hive_data
[cloudera@quickstart hive_data]$ dir
[cloudera@quickstart hive_data]$ touch student.txt
[cloudera@quickstart hive_data]$ dir
student.txt
[cloudera@quickstart hive_data]$ gedit student.txt
```


student.txt (~/Desktop/hive_data)
File Edit View Search Tools Documents Help
Open
Save
Undo
student.txt
Redo the last undon

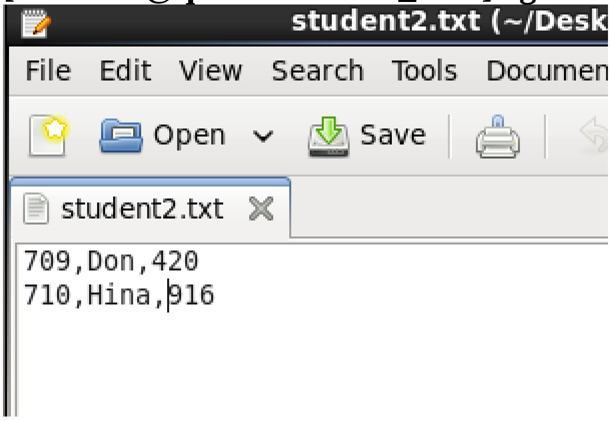
```
701,Raj,Dsai
702,Mira,It
703,Shyam,Dsai
704,lalita,Cs
```

[cloudera@quickstart hive_data]$ gedit student1.txt


student1.txt (~/Desktop/hi
File Edit View Search Tools Documents He
Open
Save
Undo
student1.txt

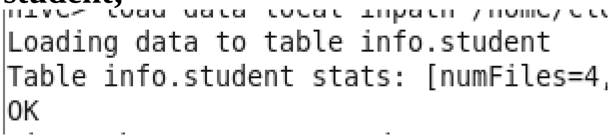
```
705,Ram,Dsai
```

[cloudera@quickstart hive_data]$ gedit student2.txt



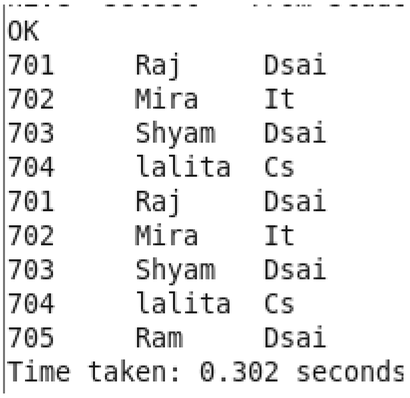
```
709,Don,420
710,Hina,916
```

[cloudera@quickstart hive_data]$ pwd
/home/cloudera/Desktop/hive_data….(path)
First terminal
hive> load data local inpath'/home/cloudera/Desktop/hive_data' into table student;



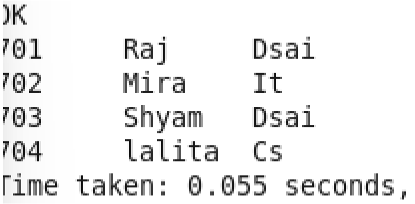
```
hive> load data local inpath'/home/cl
Loading data to table info.student
Table info.student stats: [numFiles=4,
OK
```

hive> select * from student;



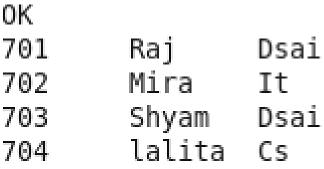
```
OK
701     Raj     Dsai
702     Mira    It
703     Shyam   Dsai
704     lalita  Cs
701     Raj     Dsai
702     Mira    It
703     Shyam   Dsai
704     lalita  Cs
705     Ram     Dsai
Time taken: 0.302 seconds
```

hive> truncate table student;
hive> select * from student;
hive> load data local inpath'/home/cloudera/Desktop/hive_data/student.txt' into table student;
hive> select * from student;



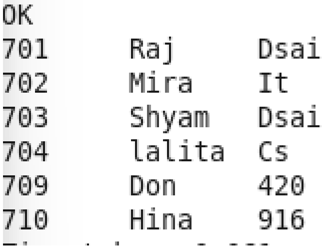
```
OK
701     Raj     Dsai
702     Mira    It
703     Shyam   Dsai
704     lalita  Cs
Time taken: 0.055 seconds,
```

load data local inpath'/home/cloudera/Desktop/hive_data/student.txt' into table student;
hive> select * from student;



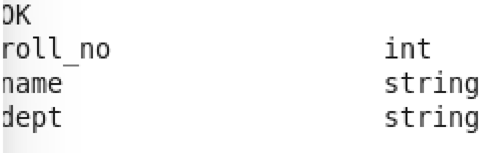
```
OK
701     Raj     Dsai
702     Mira    It
703     Shyam   Dsai
704     lalita  Cs
```

hive> load data local inpath'/home/cloudera/Desktop/hive_data/student2.txt' into table student;
hive> select * from student;



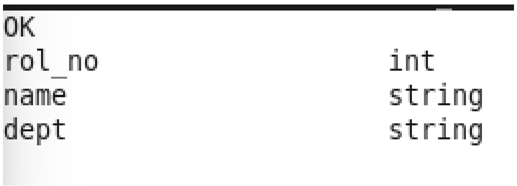
```
OK
701     Raj     Dsai
702     Mira    It
703     Shyam   Dsai
704     lalita  Cs
709     Don     420
710     Hina    916
```

hive> describe extended student;



```
OK
roll_no                 int
name                    string
dept                    string
```

hive> describe extended ext_student;

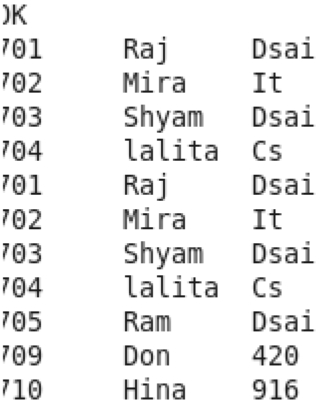


```
OK
rol_no                  int
name                    string
dept                    string
```

tableType:EXTERNAL_TABLE).......

hive> drop table ext_student;
hive> use info;

Terminal 3
[cloudera@quickstart ~]$ hdfs dfs -mkdir /hadoop_data
[cloudera@quickstart ~]$ hdfs dfs -ls /
Terminal 1
create table ext_student(roll_no int, name string,dept string)
> row format delimited
> fields terminated by ','
> location '/hadoop_data';
hive> select *from ext_student;



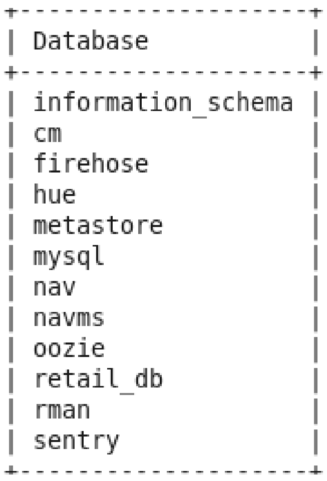
```
OK
701     Raj     Dsai
702     Mira    It
703     Shyam   Dsai
704     lalita  Cs
701     Raj     Dsai
702     Mira    It
703     Shyam   Dsai
704     lalita  Cs
705     Ram     Dsai
709     Don     420
710     Hina    916
```

Practical 8

Perform Data Ingestion using Apache Sqoop tool:

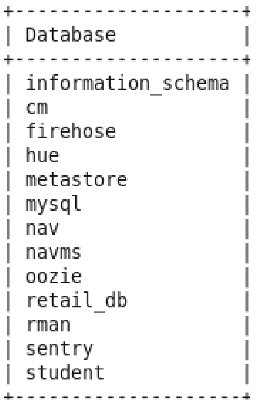
a) MySQL to HDFS (import)
b) MySQL to HIVE (import)
c) MySQL to HBASE (import)
d) HDFS to MySQL (export)
e) HIVE to MySQL (export)

```
[cloudera@quickstart ~]$ mysql -u root -pcloudera
mysql> show databases;
```



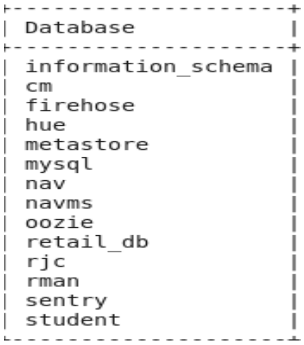
```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cm                 |
| firehose           |
| hue                |
| metastore          |
| mysql              |
| nav                |
| navms              |
| oozie              |
| retail_db          |
| rman               |
| sentry             |
+--------------------+
```

```
mysql> create database student;
```



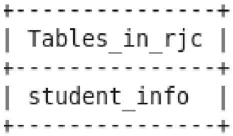
```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cm                 |
| firehose           |
| hue                |
| metastore          |
| mysql              |
| nav                |
| navms              |
| oozie              |
| retail_db          |
| rman               |
| sentry             |
| student            |
+--------------------+
```

```
mysql> create database rjc;
mysql> show databases;
```



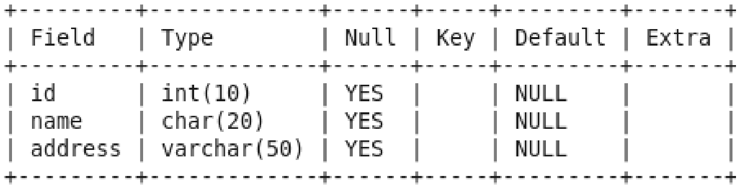
```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cm                 |
| firehose           |
| hue                |
| metastore          |
| mysql              |
| nav                |
| navms              |
| oozie              |
| retail_db          |
| rjc                |
| rman               |
| sentry             |
| student            |
+--------------------+
```

mysql> use rjc;
mysql> create table student_info(id int(10),name char(20),address varchar(50));
mysql> show tables;



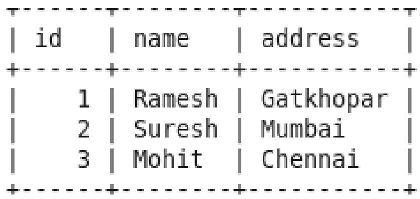
```
+---------------+
| Tables_in_rjc |
+---------------+
| student_info  |
+---------------+
```

mysql> describe student_info;



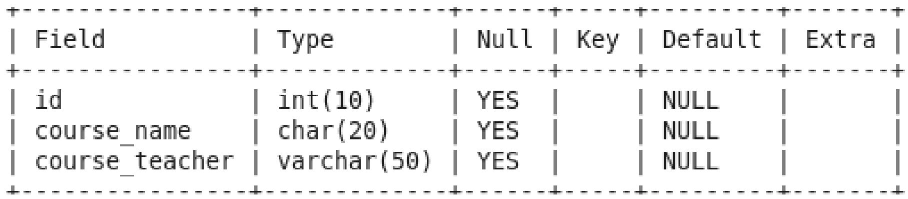
```
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| id      | int(10)     | YES  |     | NULL    |       |
| name    | char(20)    | YES  |     | NULL    |       |
| address | varchar(50) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
```

mysql> insert into student_info values(1,'Ramesh','Gatkhopar');
mysql> insert into student_info values(2,'Suresh','Mumbai');
mysql> insert into student_info values(3,'Mohit','Chennai');
mysql> select * from student_info



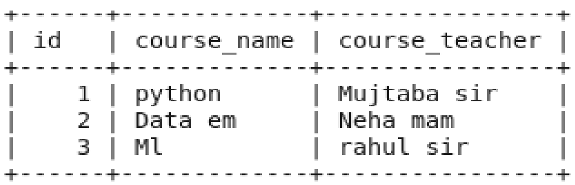
```
+------+--------+-----------+
| id   | name   | address   |
+------+--------+-----------+
|    1 | Ramesh | Gatkhopar |
|    2 | Suresh | Mumbai    |
|    3 | Mohit  | Chennai   |
+------+--------+-----------+
```

mysql> create table course_info(id int(10),course_name char(20),course_teacher varchar(50))
mysql> describe course_info;



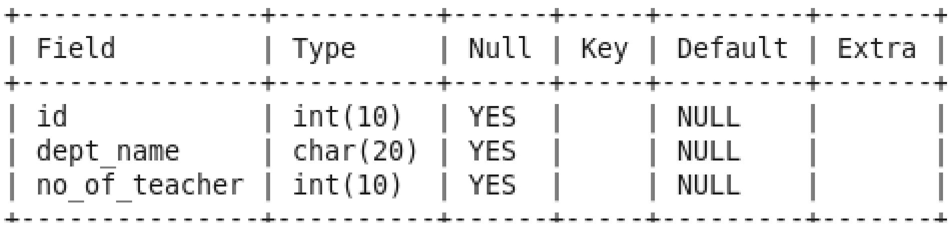
```
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| id             | int(10)     | YES  |     | NULL    |       |
| course_name    | char(20)    | YES  |     | NULL    |       |
| course_teacher | varchar(50) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
```

mysql> insert into course_info values(1,'python','Mujtaba sir');
mysql> insert into course_info values(2,'Data em','Neha mam');
mysql> insert into course_info values(3,'Ml','rahul sir');
mysql> select * from course_info;



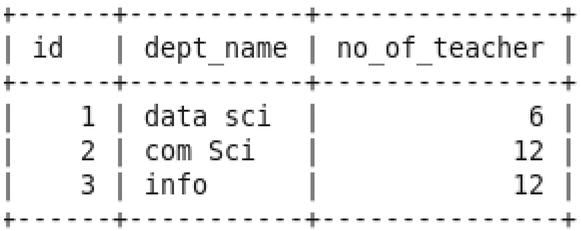
```
+------+-------------+----------------+
| id   | course_name | course_teacher |
+------+-------------+----------------+
|    1 | python      | Mujtaba sir    |
|    2 | Data em     | Neha mam       |
|    3 | Ml          | rahul sir      |
+------+-------------+----------------+
```

mysql> create table Department(id int(10),dept_name char(20),no_of_teacher int(10));
create table Department(id int(10),dept_name char(20),no_of_teacher int(10));
mysql> select * from Department;
mysql> describe Department;



```
+---------------+----------+------+-----+---------+-------+
| Field         | Type     | Null | Key | Default | Extra |
+---------------+----------+------+-----+---------+-------+
| id            | int(10)  | YES  |     | NULL    |       |
| dept_name     | char(20) | YES  |     | NULL    |       |
| no_of_teacher | int(10)  | YES  |     | NULL    |       |
+---------------+----------+------+-----+---------+-------+
```

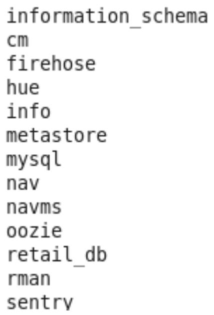
mysql> insert into Department values(1,'data sci',6);
mysql> insert into Department values(1,'data sci',6);
mysql> insert into Department values(3,'info',12);
mysql> select * from Department;



```
+------+-----------+---------------+
| id   | dept_name | no_of_teacher |
+------+-----------+---------------+
|    1 | data sci  |             6 |
|    2 | com Sci   |            12 |
|    3 | info      |            12 |
+------+-----------+---------------+
```

Terminal 2

[cloudera@quickstart ~]$ whoami
[cloudera@quickstart ~]$ hostname
[cloudera@quickstart ~]$ sqoop list-databases --connect jdbc:mysql://quickstart.cloudera:3306/ --password cloudera
--username root;



```
information_schema
cm
firehose
hue
info
metastore
mysql
nav
navms
oozie
retail_db
rman
sentry
```

From sql to hdfs(import)

```
[cloudera@quickstart ~]$ sqoop import --connect jdbc:mysql://quickstart.cloudera:3306/info --password cloudera --username root --table stu --m 1;
```

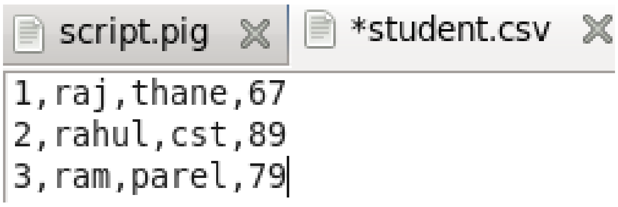
From

```
[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/student_info/part*
```

Practical 9

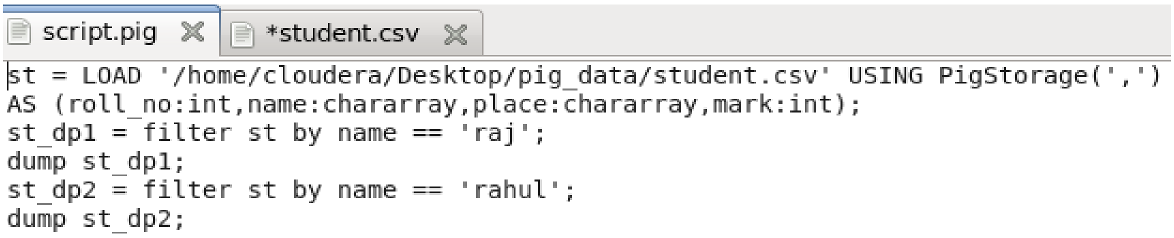
Perform data processing using pig latin.

[cloudera@quickstart pig_practical]gedit student.csv



```
1,raj,thane,67
2,rahul,cst,89
3,ram,parel,79
```

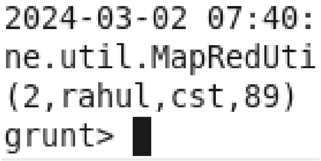
[cloudera@quickstart pig_practical]gedit script.pig



```
st = LOAD '/home/cloudera/Desktop/pig_data/student.csv' USING PigStorage(',')
AS (roll_no:int,name:chararray,place:chararray,mark:int);
st_dp1 = filter st by name == 'raj';
dump st_dp1;
st_dp2 = filter st by name == 'rahul';
dump st_dp2;
```

Write on terminal pig -x local

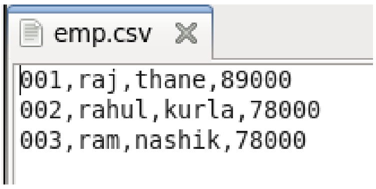
exec script.pig



```
2024-03-02 07:40:
ne.util.MapRedUti
(2,rahul,cst,89)
grunt>
```

terminal 1

[cloudera@quickstart ~]$ cd Desktop
[cloudera@quickstart Desktop]$ cd pig_practical
[cloudera@quickstart pig_practical]$ hdfs dfs -mkdir /big_practical
[cloudera@quickstart pig_practical]$ gedit emp.csv

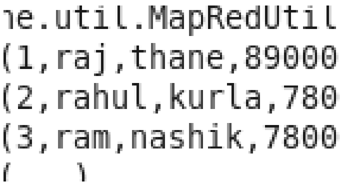


```
001,raj,thane,89000
002,rahul,kurla,78000
003,ram,nashik,78000
```

[cloudera@quickstart pig_practical]$ hdfs dfs -put emp.csv /big_practical

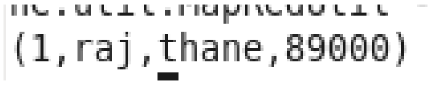
Terminal 2 pig

```
grunt> emp = LOAD '/big_practical/emp.csv' USING PigStorage(',') AS
(emp_id:int,emp_name:chararray,emp_loc:chararray,emp_salary:int);
grunt> dump emp;
```



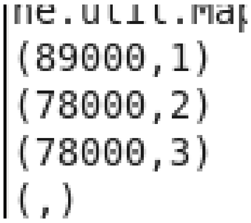
```
ne.util.MapRedUtil
(1,raj,thane,89000
(2,rahul,kurla,780
(3,ram,nashik,7800
( )
```

```
grunt> emp_dp = filter emp by emp_name == 'raj';
grunt> dump emp_dp;
```



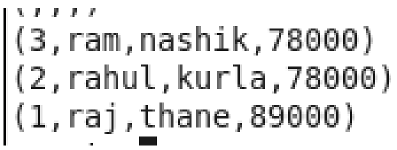
```
ne.util.MapRedUtil
(1,raj,thane,89000)
```

```
grunt> emp_dp2 = foreach emp generate emp_salary,emp_id;
grunt> dump emp_dp2;
```



```
ne.util.Map
(89000,1)
(78000,2)
(78000,3)
(,)
```

```
grunt> emp_dp3 = order emp by emp_salary ;
grunt>  dump emp_dp3 ;
```



```
(,,,)
(3,ram,nashik,78000)
(2,rahul,kurla,78000)
(1,raj,thane,89000)
```