B.M.S. COLLEGE OF ENGINEERING

(Autonomous College under VTU, Approved by AICTE, Accredited by NAAC)

MASTER OF COMPUTER APPLICATIONS

(Accredited by NBA for 5 years 2019 - 2024)

BIG DATA ANALYTICS
(22MCA2PEBD)

LAB REPORT

SUBMITTED BY

Manohar T H

(1BM23MC051)

UNDER THE GUIDANCE OF

S SHILPA
(Assistant Professor)

B.M.S. COLLEGE OF ENGINEERING

(Autonomous College under VTU, Approved by AICTE, Accredited by NAAC)

MASTER OF COMPUTER APPLICATIONS

(Accredited by NBA for 5 years 2019 - 2024)

LABORATORY CERTIFICATE

This is to certify that **MANOHAR T H (1BM23MC051)** has satisfactorily completed the course of practical in **“Big Data Analytics– 22MCA2PEBD”** Laboratory prescribed by **BMS College of Engineering** (Autonomous college under VTU) 2nd Semester MCA course in this college during the year 2022 - 2023.

Signature of Batch in charge

Dr.K.Vijayakumar

Signature of HOD

Dr. Ch. Ram Mohan Reddy

Examiner:

CONTENTS

1. Demonstration and installation of HADOOP cluster

Sudo apt update

Step 1: Install Java Development Kit

1. sudo apt update && sudo apt install openjdk-11-jdk
2. java -version
3. dirname $(dirname $(readlink -f $(which java)))
4. sudo adduser hadoop
5. su – hadoop
6. ssh-keygen -t rsa
7. cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
8. chmod 640 ~/.ssh/authorized_keys
9. sudo adduser hadoop sudo
10. sudo apt install openssh-server
11. ssh localhost
12. wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz
13. tar xzf hadoop-3.3.4.tar.gz
14. mv hadoop-3.3.4 hadoop

nano ~/.bashrc

```
export            JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64            export
HADOOP_HOME=/home/hadoop/hadoop                                           export
HADOOP_INSTALL=$HADOOP_HOME                                               export
HADOOP_MAPRED_HOME=$HADOOP_HOME                                           export
HADOOP_COMMON_HOME=$HADOOP_HOME                                           export
HADOOP_HDFS_HOME=$HADOOP_HOME                                             export
HADOOP_YARN_HOME=$HADOOP_HOME                                             export
HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native export
PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin                             export
HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

source ~/.bashrc nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

Step 2: Configuring Hadoop

```
mkdir -p ~/hadoopdata/hdfs/{namenode,datanode}
```

1. nano $HADOOP_HOME/etc/hadoop/core-site.xml

```
configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

2. nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.name.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>
  </property>
</configuration>
```

3. nano $HADOOP_HOME/etc/hadoop/mapred-site.xml

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>

</configuration>
```

4. nano $HADOOP_HOME/etc/hadoop/yarn-site.xml

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
    </property>
</configuration>
```

Step 3: Start Hadoop Cluster

1.hdfs namenode -format start-all.sh

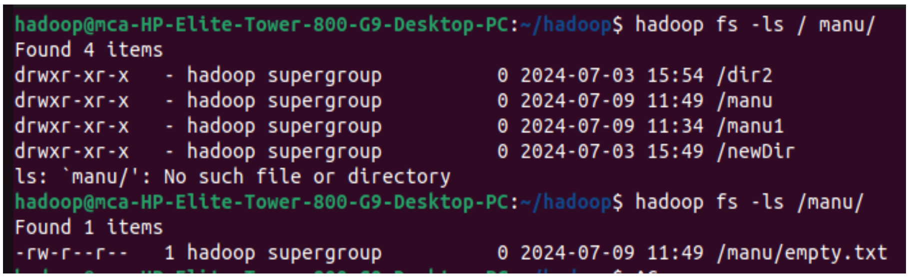
http://localhost:9870

http://localhost:8088

1 Execution of HDFS Commands for interaction with Hadoop Environment

1. Create a directory

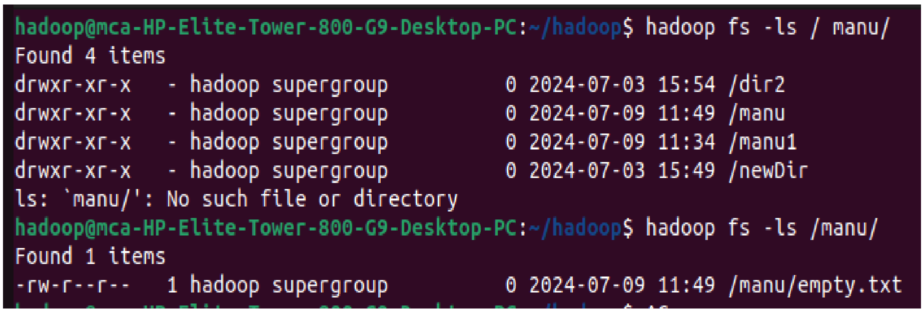
hadoop fs -mkdir /manu



```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/hadoop$ hadoop fs -ls / manu/
Found 4 items
drwxr-xr-x   - hadoop supergroup          0 2024-07-03 15:54 /dir2
drwxr-xr-x   - hadoop supergroup          0 2024-07-09 11:49 /manu
drwxr-xr-x   - hadoop supergroup          0 2024-07-09 11:34 /manu1
drwxr-xr-x   - hadoop supergroup          0 2024-07-03 15:49 /newDir
ls: `manu/': No such file or directory
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/hadoop$ hadoop fs -ls /manu/
Found 1 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 11:49 /manu/empty.txt
```

2. Create an empty file

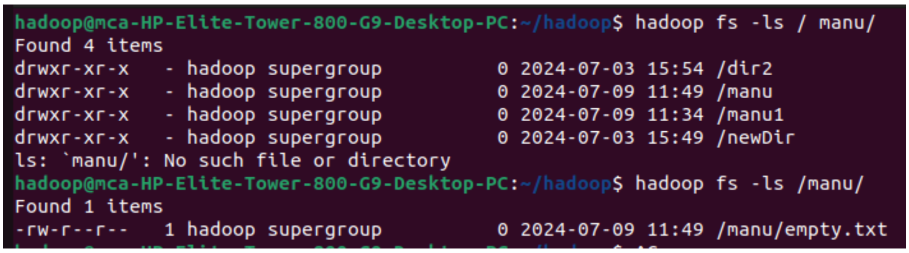
$~ hadoop fs -touchz /manu/empty.txt



```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/hadoop$ hadoop fs -ls / manu/
Found 4 items
drwxr-xr-x   - hadoop supergroup          0 2024-07-03 15:54 /dir2
drwxr-xr-x   - hadoop supergroup          0 2024-07-09 11:49 /manu
drwxr-xr-x   - hadoop supergroup          0 2024-07-09 11:34 /manu1
drwxr-xr-x   - hadoop supergroup          0 2024-07-03 15:49 /newDir
ls: `manu/': No such file or directory
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/hadoop$ hadoop fs -ls /manu/
Found 1 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 11:49 /manu/empty.txt
```

3.List all the files in a directory, recursively displays entries in all subdirectories of a path

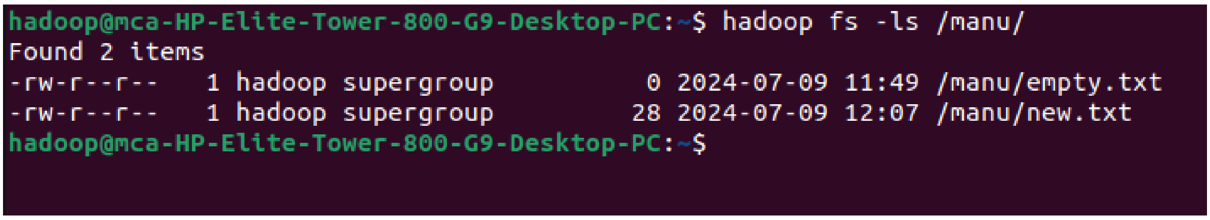
hadoop fs -ls /manu/
hadoop fs -ls /



```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/hadoop$ hadoop fs -ls / manu/
Found 4 items
drwxr-xr-x   - hadoop supergroup          0 2024-07-03 15:54 /dir2
drwxr-xr-x   - hadoop supergroup          0 2024-07-09 11:49 /manu
drwxr-xr-x   - hadoop supergroup          0 2024-07-09 11:34 /manu1
drwxr-xr-x   - hadoop supergroup          0 2024-07-03 15:49 /newDir
ls: `manu/': No such file or directory
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/hadoop$ hadoop fs -ls /manu/
Found 1 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 11:49 /manu/empty.txt
```

4.Copy files/folders from local file system to hdfs store

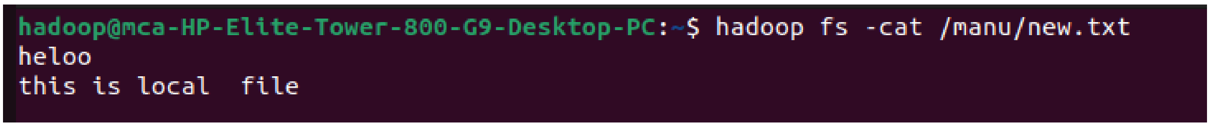
hadoop fs -put /home/hadoop/new.txt /manu/
hadoop fs -ls /manu/



```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /manu/
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 11:49 /manu/empty.txt
-rw-r--r--   1 hadoop supergroup         28 2024-07-09 12:07 /manu/new.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

5.Print the file contents

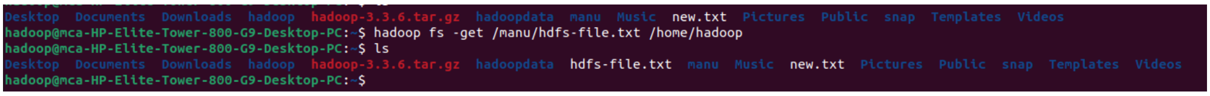
hadoop fs -cat /manu/new.txt



```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /manu/new.txt
heloo
this is local  file
```

6. To copy files/folders from hdfs store to local file system

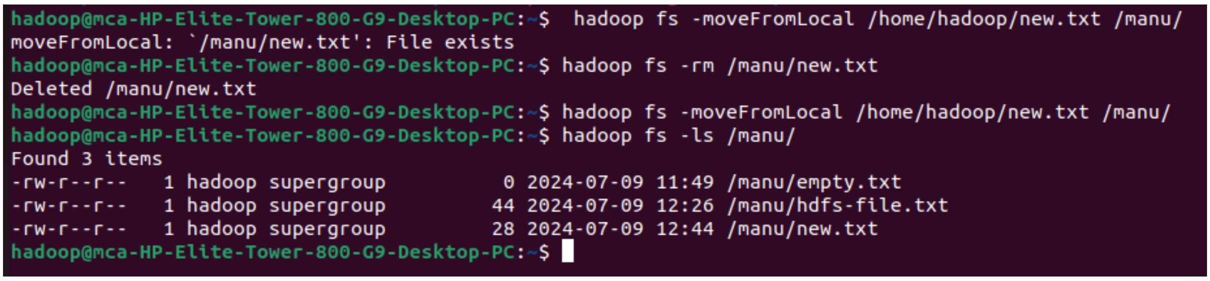
hadoop fs -get /manu/hdfs-file.txt /home/hadoop



```
Desktop  Documents  Downloads  hadoop  hadoop-3.3.6.tar.gz  hadoopdata  manu  Music  new.txt  Pictures  Public  snap  Templates  Videos
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -get /manu/hdfs-file.txt /home/hadoop
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ ls
Desktop  Documents  Downloads  hadoop  hadoop-3.3.6.tar.gz  hadoopdata  hdfs-file.txt  manu  Music  new.txt  Pictures  Public  snap  Templates  Videos
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

7. Move file from local to hdfs

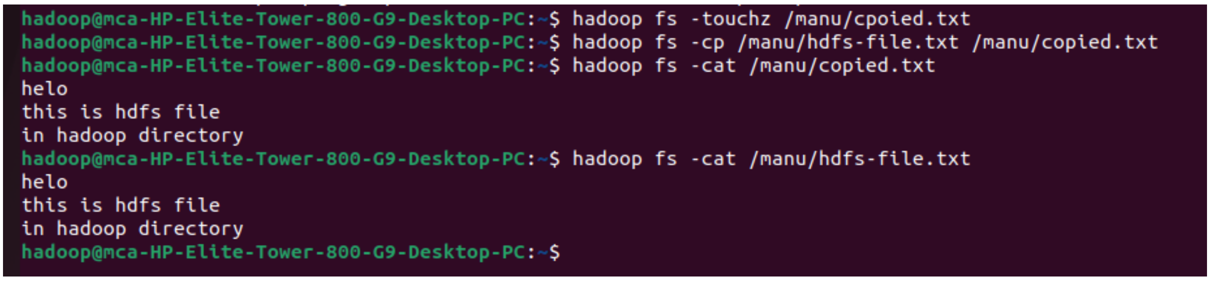
hadoop fs -moveFromLocal /home/hadoop/new.txt /manu/



```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$  hadoop fs -moveFromLocal /home/hadoop/new.txt /manu/
moveFromLocal: `/manu/new.txt': File exists
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -rm /manu/new.txt
Deleted /manu/new.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -moveFromLocal /home/hadoop/new.txt /manu/
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /manu/
Found 3 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 11:49 /manu/empty.txt
-rw-r--r--   1 hadoop supergroup         44 2024-07-09 12:26 /manu/hdfs-file.txt
-rw-r--r--   1 hadoop supergroup         28 2024-07-09 12:44 /manu/new.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

8. Copy files within hdfs

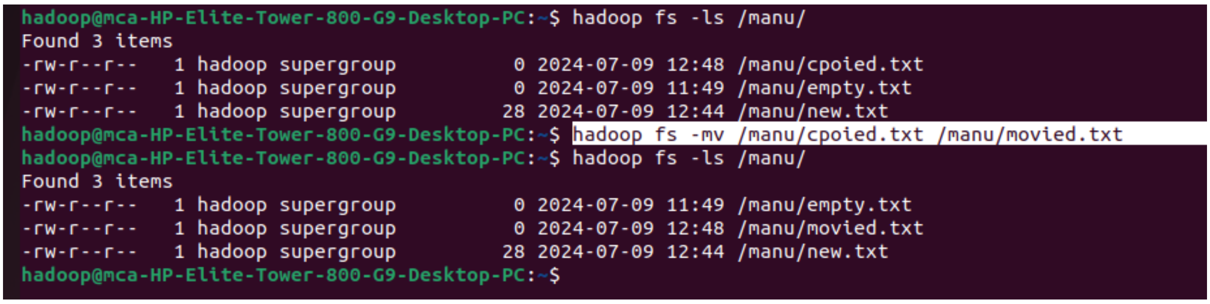
hadoop fs -moveFromLocal /home/hadoop/new.txt /manu/



```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -touchz /manu/cpoied.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cp /manu/hdfs-file.txt /manu/copied.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /manu/copied.txt
helo
this is hdfs file
in hadoop directory
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /manu/hdfs-file.txt
helo
this is hdfs file
in hadoop directory
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

9. Move/rename files within hdfs

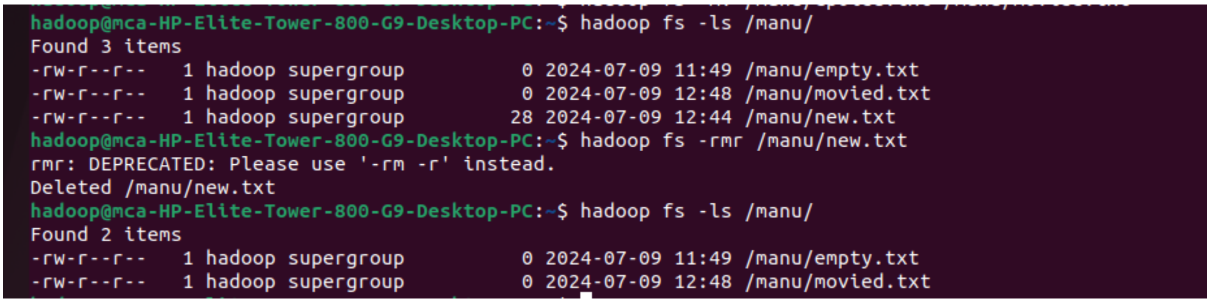
hadoop fs -mv /manu/cpoied.txt /manu/movied.txt



```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /manu/
Found 3 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 12:48 /manu/cpoied.txt
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 11:49 /manu/empty.txt
-rw-r--r--   1 hadoop supergroup         28 2024-07-09 12:44 /manu/new.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -mv /manu/cpoied.txt /manu/movied.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /manu/
Found 3 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 11:49 /manu/empty.txt
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 12:48 /manu/movied.txt
-rw-r--r--   1 hadoop supergroup         28 2024-07-09 12:44 /manu/new.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

10. Delete a file, delete a file from HDFS recursively

hadoop fs -rmr /manu/new.txt

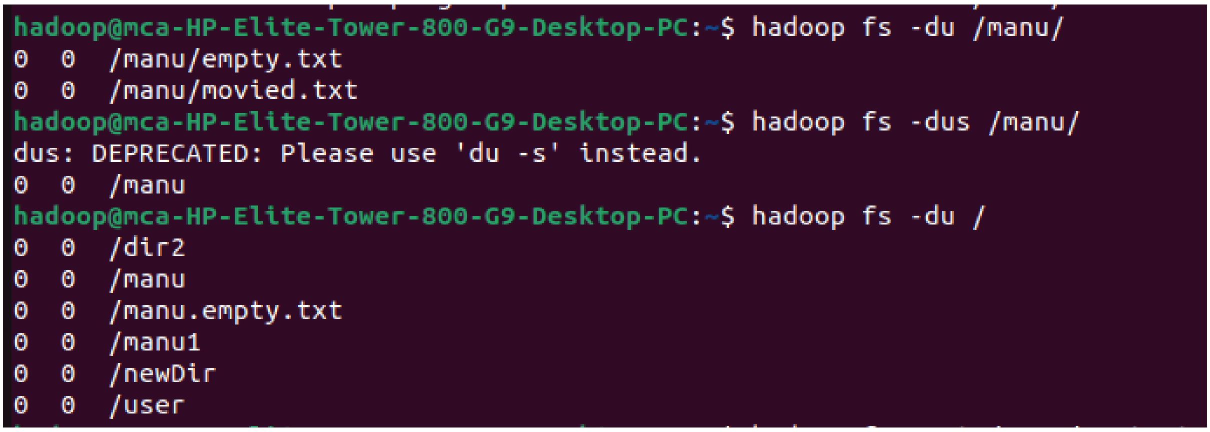


```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /manu/
Found 3 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 11:49 /manu/empty.txt
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 12:48 /manu/movied.txt
-rw-r--r--   1 hadoop supergroup         28 2024-07-09 12:44 /manu/new.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -rmr /manu/new.txt
rmr: DEPRECATED: Please use '-rm -r' instead.
Deleted /manu/new.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /manu/
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 11:49 /manu/empty.txt
-rw-r--r--   1 hadoop supergroup          0 2024-07-09 12:48 /manu/movied.txt
```

11. Display Size of directory/file, size of each file in directory

It will give the size of each file in directory.
hadoop fs -du /

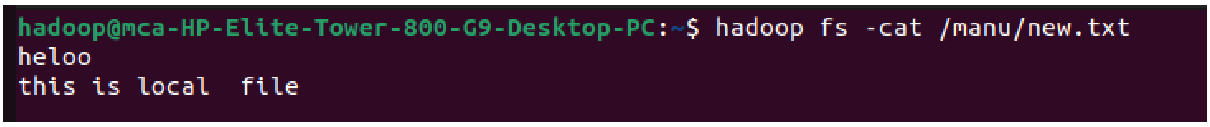
This command will give the total size of directory/file.
hadoop fs -dus /manu/



```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -du /manu/
0  0  /manu/empty.txt
0  0  /manu/movied.txt
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -dus /manu/
dus: DEPRECATED: Please use 'du -s' instead.
0  0  /manu
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -du /
0  0  /dir2
0  0  /manu
0  0  /manu.empty.txt
0  0  /manu1
0  0  /newDir
0  0  /user
```

12. Append a file in hdfs

$~ hadoop fs -appendToFile /manu/new.txt home//hadoop/data.txt



```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /manu/new.txt
heloo
this is local  file
```

3. Create and execute map reduce programs

WC_Mapper.java

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class WC_Mapper extends MapReduceBase implements
Mapper<LongWritable,Text,Text,IntWritable>{
private final static IntWritable one = new IntWritable(1);
private Text word = new Text();
public void map(LongWritable key, Text value,OutputCollector<Text,IntWritable>
output,
Reporter reporter) throws IOException{
String line = value.toString();
StringTokenizer tokenizer = new StringTokenizer(line);
while (tokenizer.hasMoreTokens()){
word.set(tokenizer.nextToken());
output.collect(word, one);
}
}}
```

WC_Reducer.java

```
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
public class WC_Reducer extends MapReduceBase implements
Reducer<Text,IntWritable,Text,IntWritable> {
public void reduce(Text key, Iterator<IntWritable>
values,OutputCollector<Text,IntWritable> output,
Reporter reporter) throws IOException {
int sum=0;
while (values.hasNext()) {
```

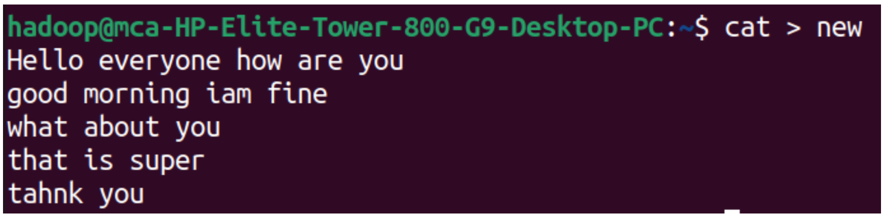
```
sum+=values.next().get();
}
output.collect(key,new IntWritable(sum));
}
}
```

WC_Runner.java

```
import java.io.IOException;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.mapred.TextInputFormat;
import org.apache.hadoop.mapred.TextOutputFormat;
public class WC_Runner {
public static void main(String[] args) throws IOException{
JobConf conf = new JobConf(WC_Runner.class);
conf.setJobName("WordCount");
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
conf.setMapperClass(WC_Mapper.class);
conf.setCombinerClass(WC_Reducer.class);
conf.setReducerClass(WC_Reducer.class);
conf.setInputFormat(TextInputFormat.class);
conf.setOutputFormat(TextOutputFormat.class);
FileInputFormat.setInputPaths(conf,new Path(args[0]));
FileOutputFormat.setOutputPath(conf,new Path(args[1]));
JobClient.runJob(conf);
}
}
```

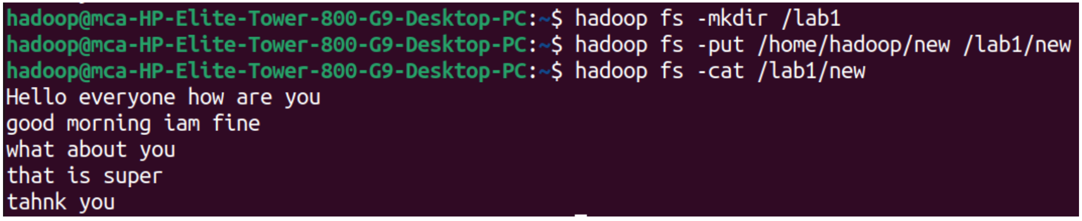
Output:

cat > new



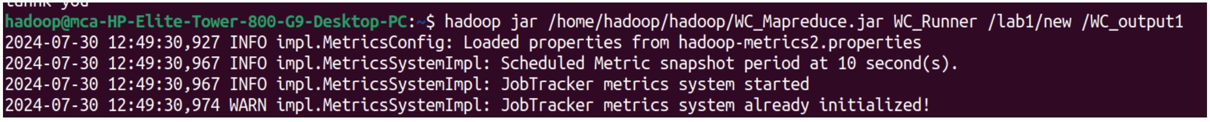
```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ cat > new
Hello everyone how are you
good morning iam fine
what about you
that is super
tahnk you
```

hadoop fs -mkdir /lab1
hadoop fs -put /home/hadoop/new /lab1/new
hadoop fs -cat /lab1/new

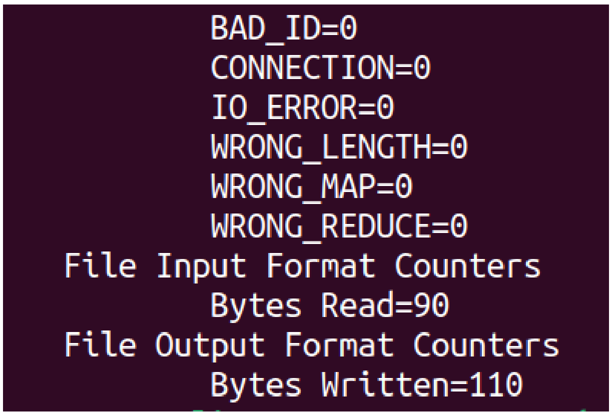


```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -mkdir /lab1
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -put /home/hadoop/new /lab1/new
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /lab1/new
Hello everyone how are you
good morning iam fine
what about you
that is super
tahnk you
```

hadoop jar /home/hadoop/hadoop/WC_Mapreduce.jar WC_Runner /lab1/new /WC_output1

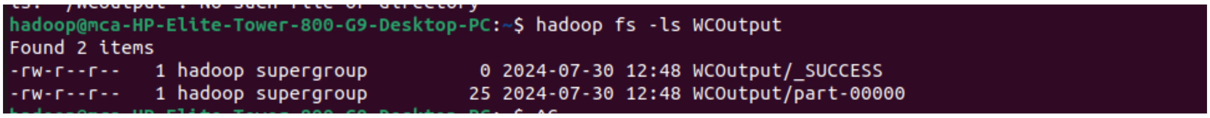


```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop jar /home/hadoop/hadoop/WC_Mapreduce.jar WC_Runner /lab1/new /WC_output1
2024-07-30 12:49:30,927 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2024-07-30 12:49:30,967 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2024-07-30 12:49:30,967 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2024-07-30 12:49:30,974 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
```



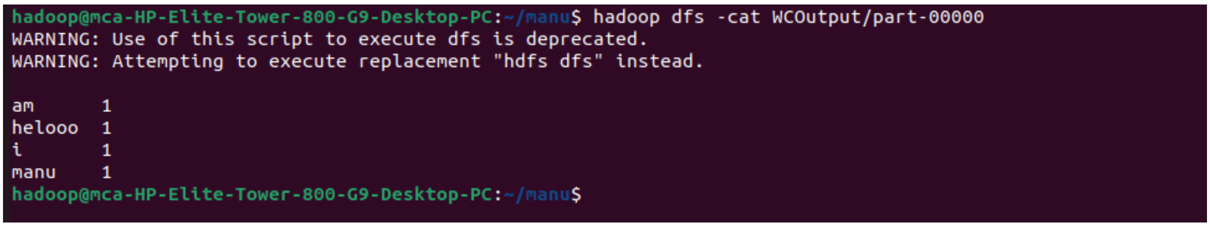
```
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=90
        File Output Format Counters
                Bytes Written=110
```

hadoop fs -ls WCOutput



```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls WCOutput
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2024-07-30 12:48 WCOutput/_SUCCESS
-rw-r--r--   1 hadoop supergroup         25 2024-07-30 12:48 WCOutput/part-00000
```

hadoop fs -cat /WCOutput1/part-00000



```
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/manu$ hadoop dfs -cat WCOutput/part-00000
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

am      1
helooo  1
i       1
manu    1
hadoop@mca-HP-Elite-Tower-800-G9-Desktop-PC:~/manu$
```

4. Data Processing Using Hive

Hive installation

Apache hive installation--before installation install jdk 1.8 in ubuntu by typing following command in the terminal

sudo apt-get install openjdk-8-jdk

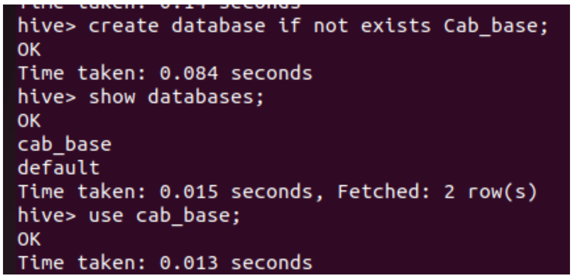
make sure that jdk 8 is default jdk by typing following command

update-alternatives–config java

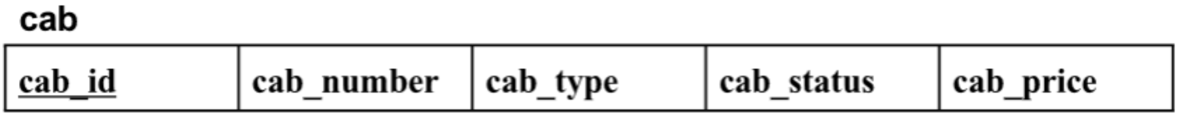
If you get error type following command one by one sudo chmod-R 777 /etc/alternatives/ sudo chmod-R 777 /var/lib/dpkg/alternatives/-- go to bashrc and change the export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64 to export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

Database name: Cabbase

1. Cabbase



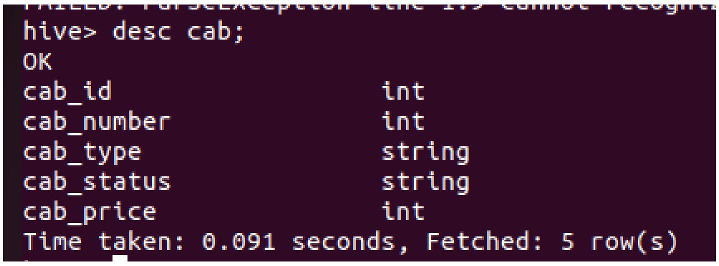
```
hive> create database if not exists Cab_base;
OK
Time taken: 0.084 seconds
hive> show databases;
OK
cab_base
default
Time taken: 0.015 seconds, Fetched: 2 row(s)
hive> use cab_base;
OK
Time taken: 0.013 seconds
```



cab

cab_id	**cab_number**	**cab_type**	**cab_status**	**cab_price**

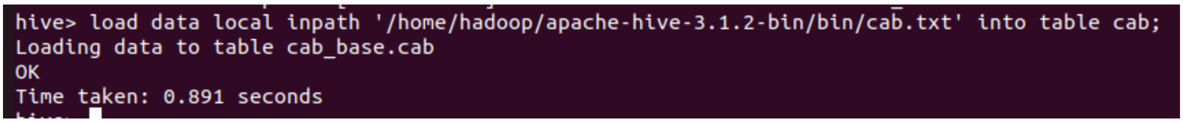
```
hive> create table if not exists cab(
    > cab_id int,
    > cab_number int,
    > cab_type string,
    > cab_status string,
    > cab_price int)
    > row format delimited
    > fields terminated by '\t'
    > lines terminated by '\n';
```



```
hive> desc cab;
OK
cab_id                  int
cab_number              int
cab_type                string
cab_status              string
cab_price               int
Time taken: 0.091 seconds, Fetched: 5 row(s)
```

Inserting Data Into Table :

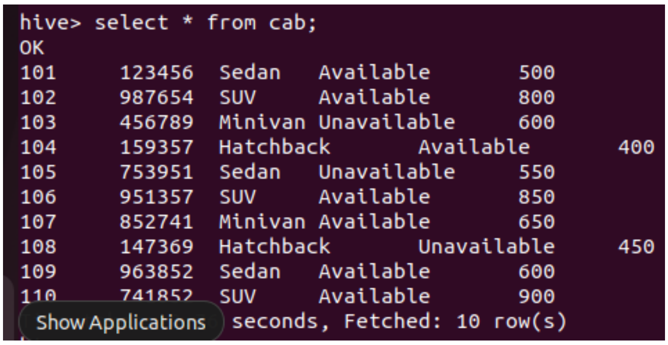
hive> load data local inpath '/home/hadoop/apache-hive-3.1.2-bin/bin/cab.txt' into table cab;



```
hive> load data local inpath '/home/hadoop/apache-hive-3.1.2-bin/bin/cab.txt' into table cab;
Loading data to table cab_base.cab
OK
Time taken: 0.891 seconds
```

1. Display the contents

hive> select * from cab;

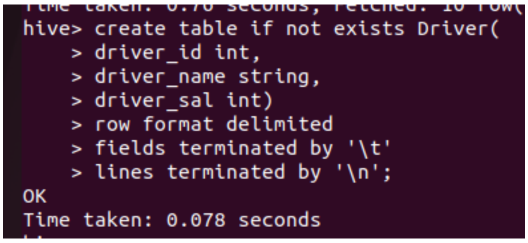


```
hive> select * from cab;
OK
101     123456  Sedan   Available       500
102     987654  SUV     Available       800
103     456789  Minivan Unavailable     600
104     159357  Hatchback       Available       400
105     753951  Sedan   Unavailable     550
106     951357  SUV     Available       850
107     852741  Minivan Available       650
108     147369  Hatchback       Unavailable     450
109     963852  Sedan   Available       600
110     741852  SUV     Available       900
seconds, Fetched: 10 row(s)
```

Driver

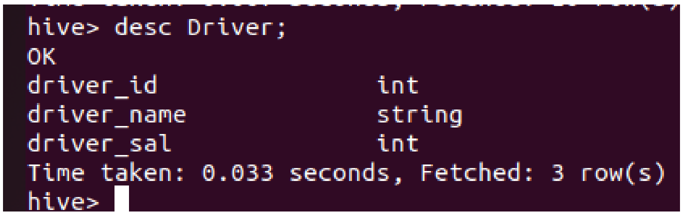
driver_id	**driver_name**	**driver_sal**

hive> create table if not exists Driver(
 > driver_id int,
 > driver_name string,
 > driver_sal int)
 > row format delimited
 > fields terminated by '\t'
 > lines terminated by '\n';



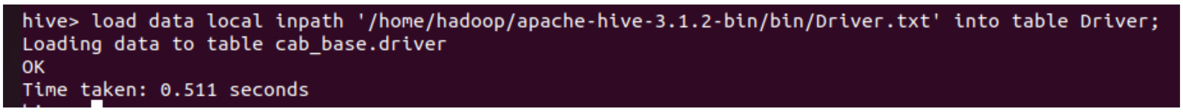
```
hive> create table if not exists Driver(
    > driver_id int,
    > driver_name string,
    > driver_sal int)
    > row format delimited
    > fields terminated by '\t'
    > lines terminated by '\n';
OK
Time taken: 0.078 seconds
```

hive> desc Driver;



```
hive> desc Driver;
OK
driver_id               int
driver_name             string
driver_sal              int
Time taken: 0.033 seconds, Fetched: 3 row(s)
hive>
```

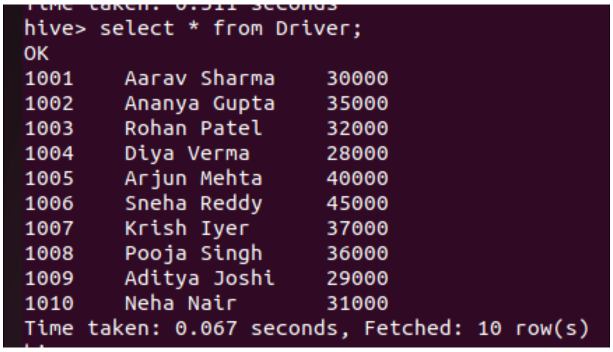
hive> load data local inpath '/home/hadoop/apache-hive-3.1.2-bin/bin/Driver.txt' into table Driver;



```
hive> load data local inpath '/home/hadoop/apache-hive-3.1.2-bin/bin/Driver.txt' into table Driver;
Loading data to table cab_base.driver
OK
Time taken: 0.511 seconds
```

1. Display the contents

hive> select * from Driver;

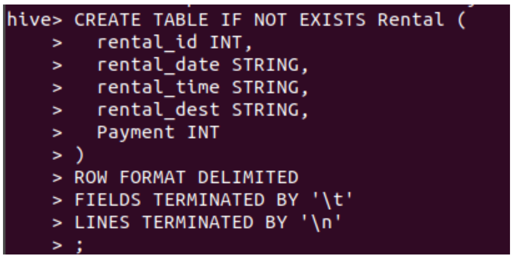


```
hive> select * from Driver;
OK
1001    Aarav Sharma    30000
1002    Ananya Gupta    35000
1003    Rohan Patel     32000
1004    Diya Verma      28000
1005    Arjun Mehta     40000
1006    Sneha Reddy     45000
1007    Krish Iyer      37000
1008    Pooja Singh     36000
1009    Aditya Joshi    29000
1010    Neha Nair       31000
Time taken: 0.067 seconds, Fetched: 10 row(s)
```

Rental

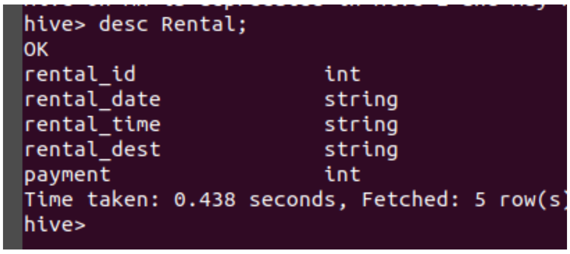
rental_id	**rental_date**	**rental_time**	**rental_dest**	**Payment**

hive> CREATE TABLE IF NOT EXISTS Rental (
 > rental_id INT,
 > rental_date STRING,
 > rental_time STRING,
 > rental_dest STRING,
 > Payment INT)
 > ROW FORMAT DELIMITED
 > FIELDS TERMINATED BY '\t'
 > LINES TERMINATED BY '\n';



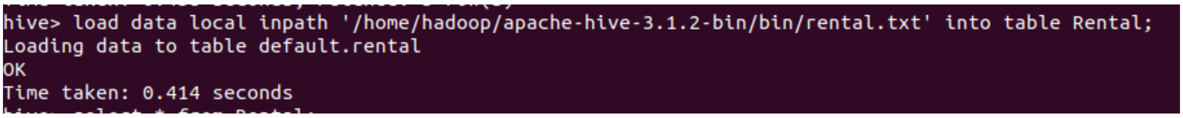
```
hive> CREATE TABLE IF NOT EXISTS Rental (
    >   rental_id INT,
    >   rental_date STRING,
    >   rental_time STRING,
    >   rental_dest STRING,
    >   Payment INT
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > LINES TERMINATED BY '\n'
    > ;
```

hive> desc Rental;



```
hive> desc Rental;
OK
rental_id               int
rental_date             string
rental_time             string
rental_dest             string
payment                 int
Time taken: 0.438 seconds, Fetched: 5 row(s)
hive>
```

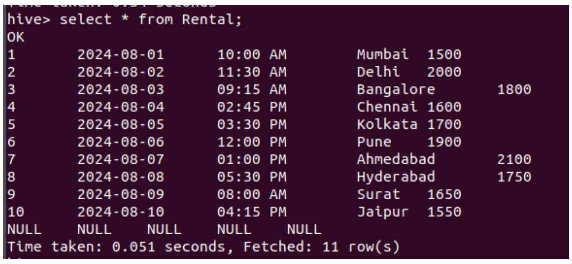
hive> load data local inpath '/home/hadoop/apache-hive-3.1.2-bin/bin/rental.txt' into table Rental;



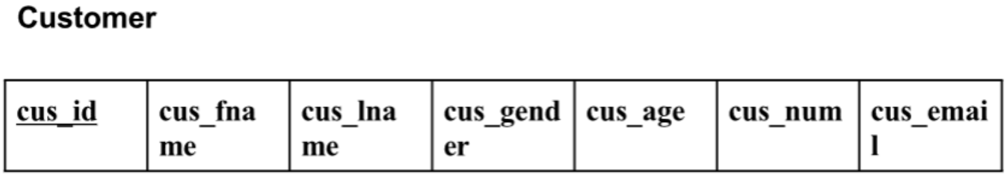
```
hive> load data local inpath '/home/hadoop/apache-hive-3.1.2-bin/bin/rental.txt' into table Rental;
Loading data to table default.rental
OK
Time taken: 0.414 seconds
```

1. Display the contents

hive> select * from Rental;



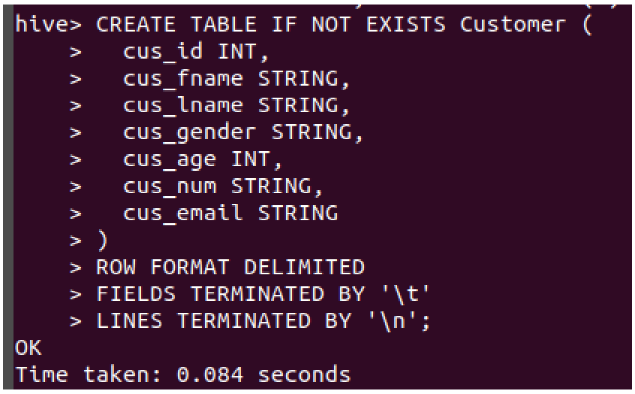
```
hive> select * from Rental;
OK
1       2024-08-01      10:00 AM        Mumbai  1500
2       2024-08-02      11:30 AM        Delhi   2000
3       2024-08-03      09:15 AM        Bangalore       1800
4       2024-08-04      02:45 PM        Chennai 1600
5       2024-08-05      03:30 PM        Kolkata 1700
6       2024-08-06      12:00 PM        Pune    1900
7       2024-08-07      01:00 PM        Ahmedabad       2100
8       2024-08-08      05:30 PM        Hyderabad       1750
9       2024-08-09      08:00 AM        Surat   1650
10      2024-08-10      04:15 PM        Jaipur  1550
NULL    NULL    NULL    NULL    NULL
Time taken: 0.051 seconds, Fetched: 11 row(s)
```



Customer

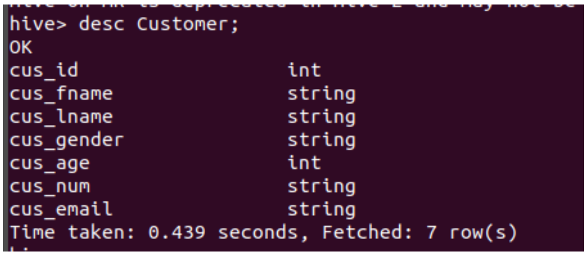
<u>cus_id</u>	cus_fname	cus_lname	cus_gender	cus_age	cus_num	cus_email

```
hive> CREATE TABLE IF NOT EXISTS Customer (
    >   cus_id INT,
    >   cus_fname STRING,
    >   cus_lname STRING,
    >   cus_gender STRING,
    >   cus_age INT,
    >   cus_num STRING,
    >   cus_email STRING)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > LINES TERMINATED BY '\n';
```



```
hive> CREATE TABLE IF NOT EXISTS Customer (
    >   cus_id INT,
    >   cus_fname STRING,
    >   cus_lname STRING,
    >   cus_gender STRING,
    >   cus_age INT,
    >   cus_num STRING,
    >   cus_email STRING
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > LINES TERMINATED BY '\n';
OK
Time taken: 0.084 seconds
```

hive> desc Customer;

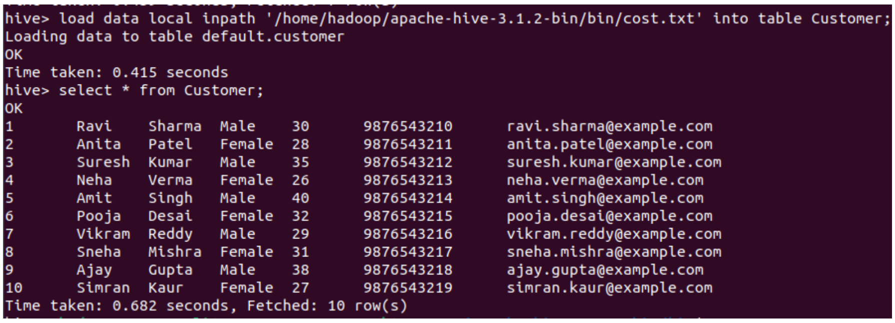


```
hive> desc Customer;
OK
cus_id                  int
cus_fname               string
cus_lname               string
cus_gender              string
cus_age                 int
cus_num                 string
cus_email               string
Time taken: 0.439 seconds, Fetched: 7 row(s)
```

hive> load data local inpath '/home/hadoop/apache-hive-3.1.2-bin/bin/cost.txt' into table Customer;

1. Display the contents

hive> select * from Customer;



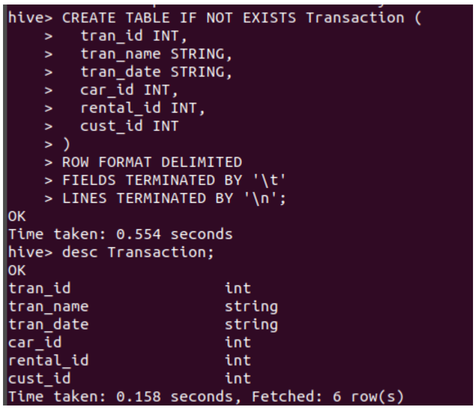
```
hive> load data local inpath '/home/hadoop/apache-hive-3.1.2-bin/bin/cost.txt' into table Customer;
Loading data to table default.customer
OK
Time taken: 0.415 seconds
hive> select * from Customer;
OK
1       Ravi    Sharma  Male    30      9876543210      ravi.sharma@example.com
2       Anita   Patel   Female  28      9876543211      anita.patel@example.com
3       Suresh  Kumar   Male    35      9876543212      suresh.kumar@example.com
4       Neha    Verma   Female  26      9876543213      neha.verma@example.com
5       Amit    Singh   Male    40      9876543214      amit.singh@example.com
6       Pooja   Desai   Female  32      9876543215      pooja.desai@example.com
7       Vikram  Reddy   Male    29      9876543216      vikram.reddy@example.com
8       Sneha   Mishra  Female  31      9876543217      sneha.mishra@example.com
9       Ajay    Gupta   Male    38      9876543218      ajay.gupta@example.com
10      Simran  Kaur    Female  27      9876543219      simran.kaur@example.com
Time taken: 0.682 seconds, Fetched: 10 row(s)
```

Transaction

<u>tran_id</u>	tran_name	tran_date	car_id	rental_id	cust_id

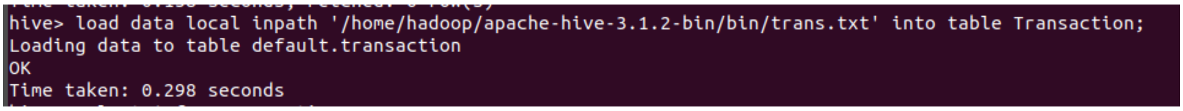
```
hive> CREATE TABLE IF NOT EXISTS Transaction (
    >   tran_id INT,
    >   tran_name STRING,
    >   tran_date STRING,
    >   car_id INT,
    >   rental_id INT,
    >   cust_id INT
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > LINES TERMINATED BY '\n';
```

```
hive> desc Transaction;
```



```
hive> CREATE TABLE IF NOT EXISTS Transaction (
    >   tran_id INT,
    >   tran_name STRING,
    >   tran_date STRING,
    >   car_id INT,
    >   rental_id INT,
    >   cust_id INT
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > LINES TERMINATED BY '\n';
OK
Time taken: 0.554 seconds
hive> desc Transaction;
OK
tran_id                 int
tran_name               string
tran_date               string
car_id                  int
rental_id               int
cust_id                 int
Time taken: 0.158 seconds, Fetched: 6 row(s)
```

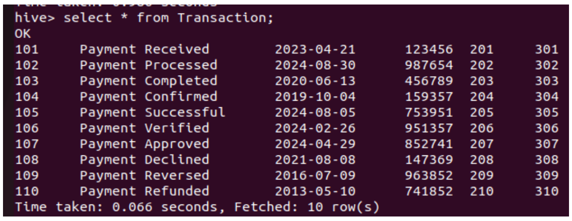
```
hive> load data local inpath '/home/hadoop/apache-hive-3.1.2-bin/bin/trans.txt' into table Transaction;
```



```
hive> load data local inpath '/home/hadoop/apache-hive-3.1.2-bin/bin/trans.txt' into table Transaction;
Loading data to table default.transaction
OK
Time taken: 0.298 seconds
```

1. Display the contents

hive> select * from Transaction;



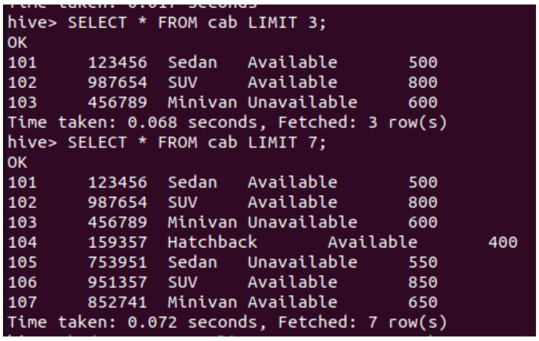
```
hive> select * from Transaction;
OK
101     Payment Received        2023-04-21      123456  201     301
102     Payment Processed       2024-08-30      987654  202     302
103     Payment Completed       2020-06-13      456789  203     303
104     Payment Confirmed       2019-10-04      159357  204     304
105     Payment Successful      2024-08-05      753951  205     305
106     Payment Verified        2024-02-26      951357  206     306
107     Payment Approved        2024-04-29      852741  207     307
108     Payment Declined        2021-08-08      147369  208     308
109     Payment Reversed        2016-07-09      963852  209     309
110     Payment Refunded        2013-05-10      741852  210     310
Time taken: 0.066 seconds, Fetched: 10 row(s)
```

2. Limit the display to three rows or n no. of rows.

cab

hive> SELECT * FROM cab LIMIT 3;

hive> SELECT * FROM cab LIMIT 7;

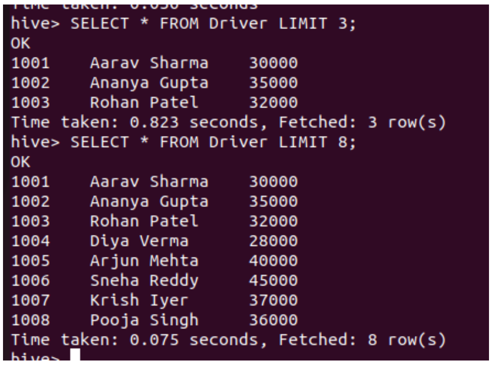


```
hive> SELECT * FROM cab LIMIT 3;
OK
101     123456  Sedan   Available       500
102     987654  SUV     Available       800
103     456789  Minivan Unavailable     600
Time taken: 0.068 seconds, Fetched: 3 row(s)
hive> SELECT * FROM cab LIMIT 7;
OK
101     123456  Sedan   Available       500
102     987654  SUV     Available       800
103     456789  Minivan Unavailable     600
104     159357  Hatchback       Available       400
105     753951  Sedan   Unavailable     550
106     951357  SUV     Available       850
107     852741  Minivan Available       650
Time taken: 0.072 seconds, Fetched: 7 row(s)
```

Driver

hive> SELECT * FROM Driver LIMIT 3;

hive> SELECT * FROM Driver LIMIT 8;

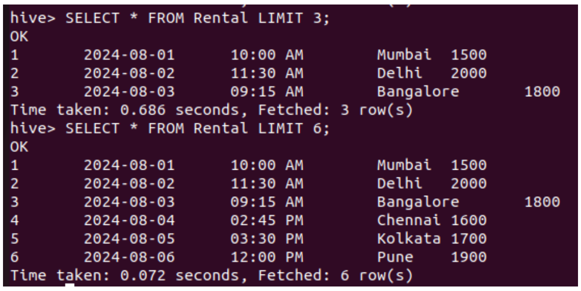


```
hive> SELECT * FROM Driver LIMIT 3;
OK
1001    Aarav Sharma    30000
1002    Ananya Gupta    35000
1003    Rohan Patel     32000
Time taken: 0.823 seconds, Fetched: 3 row(s)
hive> SELECT * FROM Driver LIMIT 8;
OK
1001    Aarav Sharma    30000
1002    Ananya Gupta    35000
1003    Rohan Patel     32000
1004    Diya Verma      28000
1005    Arjun Mehta     40000
1006    Sneha Reddy     45000
1007    Krish Iyer      37000
1008    Pooja Singh     36000
Time taken: 0.075 seconds, Fetched: 8 row(s)
```

Rental

hive> SELECT * FROM Rental LIMIT 3;

hive> SELECT * FROM Rental LIMIT 6;

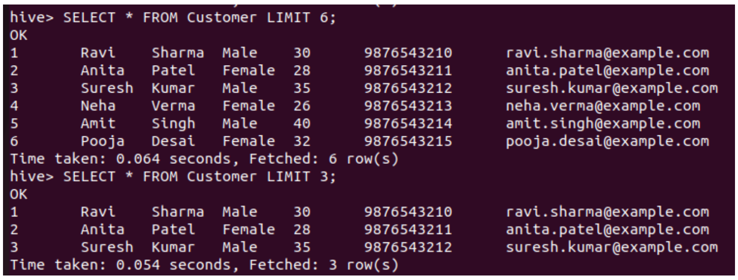


```
hive> SELECT * FROM Rental LIMIT 3;
OK
1       2024-08-01      10:00 AM        Mumbai  1500
2       2024-08-02      11:30 AM        Delhi   2000
3       2024-08-03      09:15 AM        Bangalore       1800
Time taken: 0.686 seconds, Fetched: 3 row(s)
hive> SELECT * FROM Rental LIMIT 6;
OK
1       2024-08-01      10:00 AM        Mumbai  1500
2       2024-08-02      11:30 AM        Delhi   2000
3       2024-08-03      09:15 AM        Bangalore       1800
4       2024-08-04      02:45 PM        Chennai 1600
5       2024-08-05      03:30 PM        Kolkata 1700
6       2024-08-06      12:00 PM        Pune    1900
Time taken: 0.072 seconds, Fetched: 6 row(s)
```

Customer

hive> SELECT * FROM Customer LIMIT 6;

hive> SELECT * FROM Customer LIMIT 3;

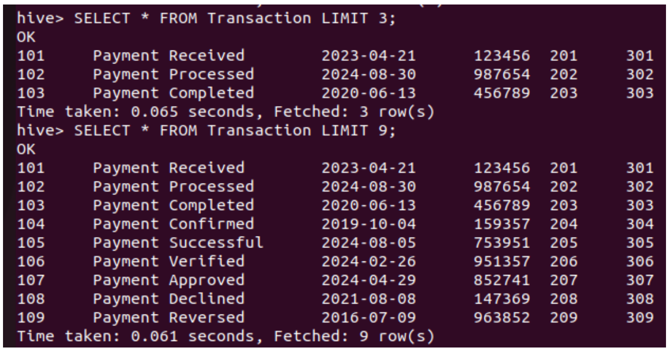


```
hive> SELECT * FROM Customer LIMIT 6;
OK
1       Ravi    Sharma  Male    30      9876543210      ravi.sharma@example.com
2       Anita   Patel   Female  28      9876543211      anita.patel@example.com
3       Suresh  Kumar   Male    35      9876543212      suresh.kumar@example.com
4       Neha    Verma   Female  26      9876543213      neha.verma@example.com
5       Amit    Singh   Male    40      9876543214      amit.singh@example.com
6       Pooja   Desai   Female  32      9876543215      pooja.desai@example.com
Time taken: 0.064 seconds, Fetched: 6 row(s)
hive> SELECT * FROM Customer LIMIT 3;
OK
1       Ravi    Sharma  Male    30      9876543210      ravi.sharma@example.com
2       Anita   Patel   Female  28      9876543211      anita.patel@example.com
3       Suresh  Kumar   Male    35      9876543212      suresh.kumar@example.com
Time taken: 0.054 seconds, Fetched: 3 row(s)
```

Transaction

hive> SELECT * FROM Transaction LIMIT 3;

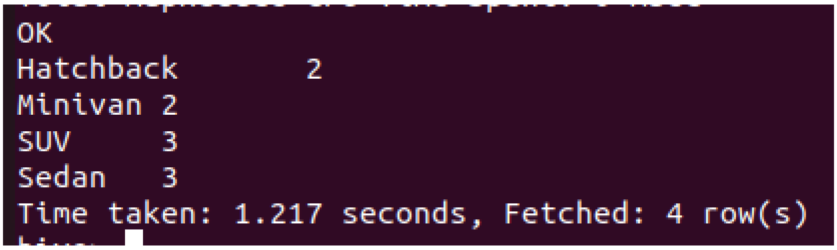
hive> SELECT * FROM Transaction LIMIT 9;



```
hive> SELECT * FROM Transaction LIMIT 3;
OK
101     Payment Received        2023-04-21      123456  201     301
102     Payment Processed       2024-08-30      987654  202     302
103     Payment Completed       2020-06-13      456789  203     303
Time taken: 0.065 seconds, Fetched: 3 row(s)
hive> SELECT * FROM Transaction LIMIT 9;
OK
101     Payment Received        2023-04-21      123456  201     301
102     Payment Processed       2024-08-30      987654  202     302
103     Payment Completed       2020-06-13      456789  203     303
104     Payment Confirmed       2019-10-04      159357  204     304
105     Payment Successful      2024-08-05      753951  205     305
106     Payment Verified        2024-02-26      951357  206     306
107     Payment Approved        2024-04-29      852741  207     307
108     Payment Declined        2021-08-08      147369  208     308
109     Payment Reversed        2016-07-09      963852  209     309
Time taken: 0.061 seconds, Fetched: 9 row(s)
```

3. Display the count of each car type.

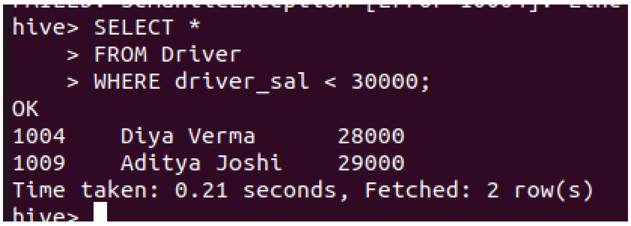
```
hive> SELECT cab_type, COUNT(*) AS count
    > FROM cab
    > GROUP BY cab_type;
```



```
OK
Hatchback       2
Minivan 2
SUV     3
Sedan   3
Time taken: 1.217 seconds, Fetched: 4 row(s)
```

4. Display the driver details whose salary is less than 30000.

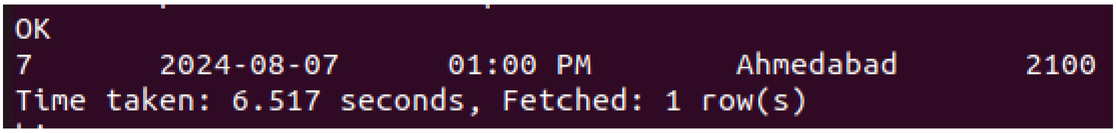
```
hive> SELECT *
    > FROM Driver
    > WHERE driver_sal < 30000;
```



```
hive> SELECT *
    > FROM Driver
    > WHERE driver_sal < 30000;
OK
1004    Diya Verma      28000
1009    Aditya Joshi    29000
Time taken: 0.21 seconds, Fetched: 2 row(s)
hive>
```

5. Display the rental details where the payment is maximum.

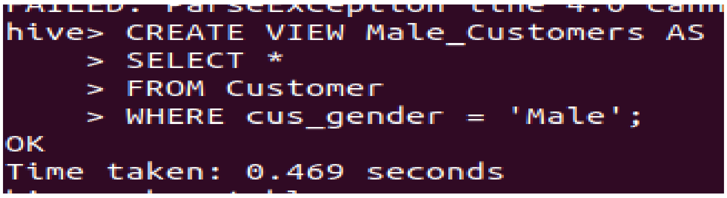
```
hive> SELECT *
    > FROM Rental
    > WHERE Payment =
(SELECT MAX(Payment) FROM
Rental);
```



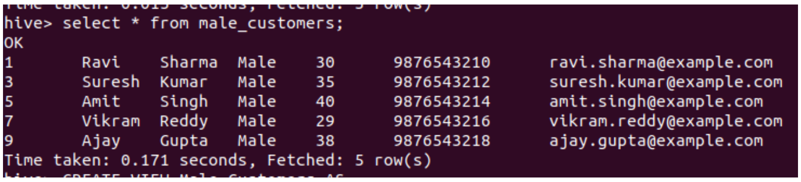
```
OK
7       2024-08-07      01:00 PM        Ahmedabad       2100
Time taken: 6.517 seconds, Fetched: 1 row(s)
```

6. Create a partition for customers based on male and female.

```
hive> CREATE VIEW
Male_Customers AS
    > SELECT *
    > FROM Customer
    > WHERE cus_gender = 'Male';
```

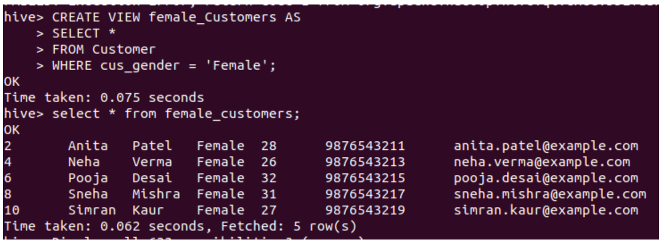


```
hive> CREATE VIEW Male_Customers AS
    > SELECT *
    > FROM Customer
    > WHERE cus_gender = 'Male';
OK
Time taken: 0.469 seconds
```



```
hive> select * from male_customers;
OK
1       Ravi    Sharma  Male    30      9876543210      ravi.sharma@example.com
3       Suresh  Kumar   Male    35      9876543212      suresh.kumar@example.com
5       Amit    Singh   Male    40      9876543214      amit.singh@example.com
7       Vikram  Reddy   Male    29      9876543216      vikram.reddy@example.com
9       Ajay    Gupta   Male    38      9876543218      ajay.gupta@example.com
Time taken: 0.171 seconds, Fetched: 5 row(s)
```

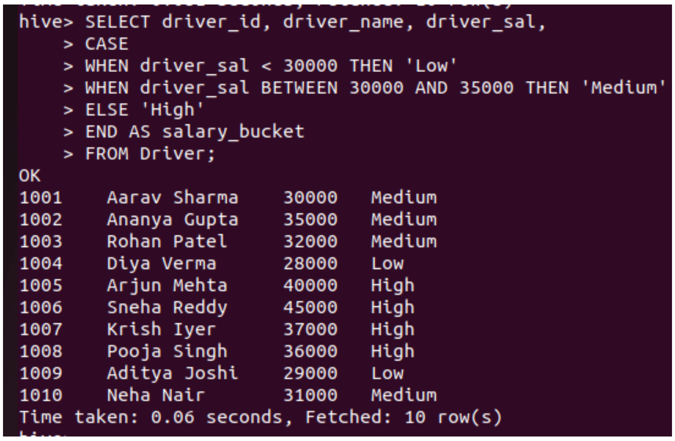
```
hive> CREATE VIEW female_Customers AS
    > SELECT *
    > FROM Customer
    >
WHERE cus_gender = 'Female';
```



```
hive> CREATE VIEW female_Customers AS
    > SELECT *
    > FROM Customer
    > WHERE cus_gender = 'Female';
OK
Time taken: 0.075 seconds
hive> select * from female_customers;
OK
2       Anita   Patel   Female  28      9876543211      anita.patel@example.com
4       Neha    Verma   Female  26      9876543213      neha.verma@example.com
6       Pooja   Desai   Female  32      9876543215      pooja.desai@example.com
8       Sneha   Mishra  Female  31      9876543217      sneha.mishra@example.com
10      Simran  Kaur    Female  27      9876543219      simran.kaur@example.com
Time taken: 0.062 seconds, Fetched: 5 row(s)
```

7. Create three buckets on drivers based on salary.

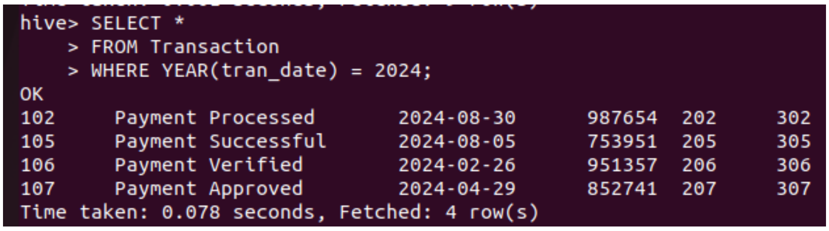
```
hive> SELECT driver_id, driver_name, driver_sal,
    > CASE
    > WHEN driver_sal < 30000 THEN 'Low'
    > WHEN driver_sal BETWEEN 30000 AND 35000 THEN 'Medium'
    > ELSE 'High'
    > END AS salary_bucket
    > FROM Driver;
```



```
hive> SELECT driver_id, driver_name, driver_sal,
    > CASE
    > WHEN driver_sal < 30000 THEN 'Low'
    > WHEN driver_sal BETWEEN 30000 AND 35000 THEN 'Medium'
    > ELSE 'High'
    > END AS salary_bucket
    > FROM Driver;
OK
1001    Aarav Sharma    30000   Medium
1002    Ananya Gupta    35000   Medium
1003    Rohan Patel     32000   Medium
1004    Diya Verma      28000   Low
1005    Arjun Mehta     40000   High
1006    Sneha Reddy     45000   High
1007    Krish Iyer      37000   High
1008    Pooja Singh     36000   High
1009    Aditya Joshi    29000   Low
1010    Neha Nair       31000   Medium
Time taken: 0.06 seconds, Fetched: 10 row(s)
```

8. Display the transaction details of transactions that happened during the year 2024.

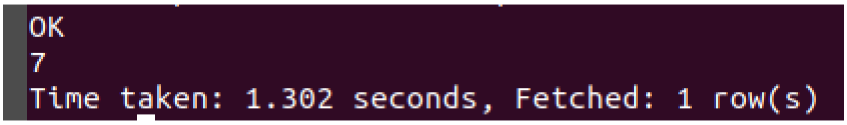
```
hive> SELECT *
    > FROM Transaction
    > WHERE YEAR(tran_date) = 2024;
```



```
hive> SELECT *
    > FROM Transaction
    > WHERE YEAR(tran_date) = 2024;
OK
102     Payment Processed       2024-08-30      987654  202     302
105     Payment Successful      2024-08-05      753951  205     305
106     Payment Verified        2024-02-26      951357  206     306
107     Payment Approved        2024-04-29      852741  207     307
Time taken: 0.078 seconds, Fetched: 4 row(s)
```

9. Display the count of no. of cabs that are available (status='A')

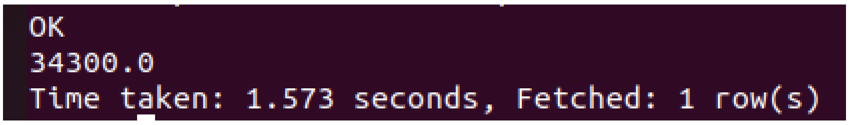
```
hive> SELECT COUNT(*) AS available_cabs
    > FROM cab
    > WHERE cab_status = 'Available';
```



```
OK
7
Time taken: 1.302 seconds, Fetched: 1 row(s)
```

10. Display the average salary of all the drivers.

hive> SELECT AVG(driver_sal) AS average_salary
    > FROM Driver;

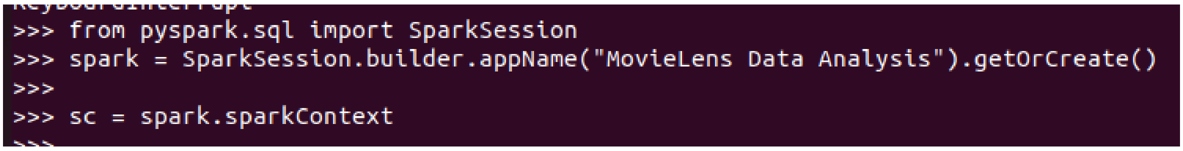


```
OK
34300.0
Time taken: 1.573 seconds, Fetched: 1 row(s)
```

5. Data processing using Spark

MovieLens datasets were collected by the GroupLens Research Project at the University of minnesota. It represents users' reviews of movies.
This data set consists of:
*** 100,000 ratings (1-5) from 943 users on 1682 movies.**
*** Each user has rated at least 20 movies.**
*** Simple demographic info for the users (age, gender, occupation, zip)**
u.data -- The full u data set, 100000 ratings by 943 users on 1682 items.
Each user has rated at least 20 movies. Users and items are numbered consecutively from 1. The data is randomly ordered. This is a tab separated list of
user id item id rating timestamp.
The time stamps are unix seconds since 1/1/1970 UTC
u.user -- Demographic information about the users; this is a pipe separated list of
user id | age | gender | occupation | zip code
The user ids are the ones used in the u.data data set.

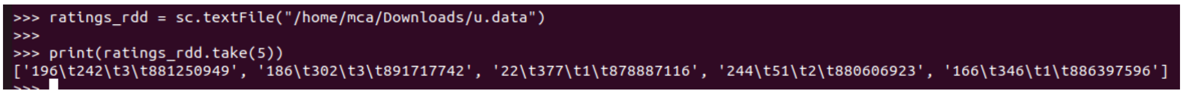
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("MovieLens Data Analysis").getOrCreate()
sc = spark.sparkContext
```



```
KeyboardInterrupt
>>> from pyspark.sql import SparkSession
>>> spark = SparkSession.builder.appName("MovieLens Data Analysis").getOrCreate()
>>>
>>> sc = spark.sparkContext
>>>
```

1.load the u.data into the new RDD.

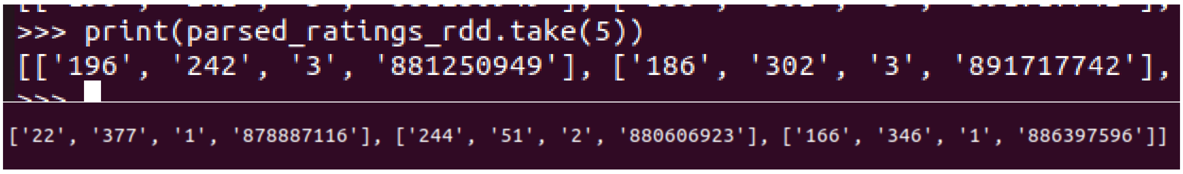
```
#Load u.data into an RDD
>> ratings_rdd = sc.textFile("/home/mca/Downloads/u.data")

#  Display the first few records
>>> print(ratings_rdd.take(5))
```



```
>>> ratings_rdd = sc.textFile("/home/mca/Downloads/u.data")
>>>
>>> print(ratings_rdd.take(5))
['196\t242\t3\t881250949', '186\t302\t3\t891717742', '22\t377\t1\t878887116', '244\t51\t2\t880606923', '166\t346\t1\t886397596']
>>>
```

# Step 4: Parse the RDD into a structured format
parsed_ratings_rdd = ratings_rdd.map(lambda line:line.split('\t'))

>>> print(parsed_ratings_rdd.take(5))



```
>>> print(parsed_ratings_rdd.take(5))
[['196', '242', '3', '881250949'], ['186', '302', '3', '891717742'],
>>>
['22', '377', '1', '878887116'], ['244', '51', '2', '880606923'], ['166', '346', '1', '886397596']]
```

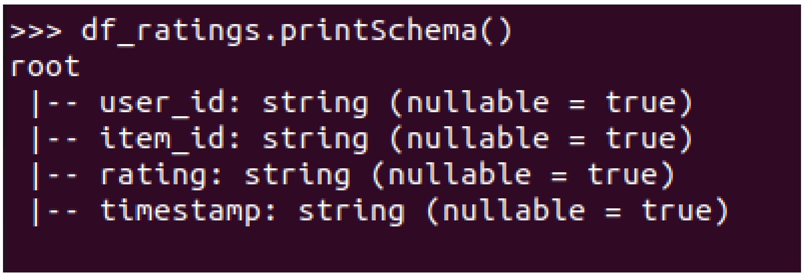
2.Change the RDD to a Dataframe.

# Convert RDD to DataFrame with appropriate column names
>>> df_ratings = parsed_ratings_rdd.toDF(["user_id", "item_id", "rating", "timestamp"])

ratings_df = spark.createDataFrame(parsed_ratings_rdd, schema=rating_schema)

3.Return the schema of this DataFrame.

>>> df_ratings.printSchema()



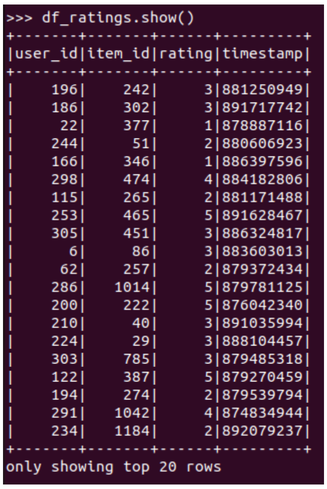
```
>>> df_ratings.printSchema()
root
 |-- user_id: string (nullable = true)
 |-- item_id: string (nullable = true)
 |-- rating: string (nullable = true)
 |-- timestamp: string (nullable = true)
```

4.Register the DataFrame as a temp u_data table.

>>> df_ratings.createOrReplaceTempView("u_data")

5 Display the contents of newly created u_data table

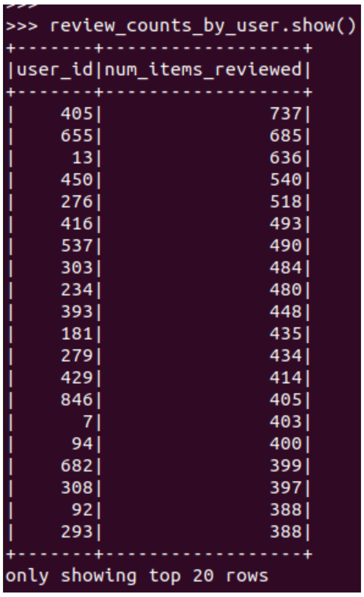
```
>>> df_ratings.show()
```



```
>>> df_ratings.show()
+-------+-------+------+---------+
|user_id|item_id|rating|timestamp|
+-------+-------+------+---------+
|    196|    242|     3|881250949|
|    186|    302|     3|891717742|
|     22|    377|     1|878887116|
|    244|     51|     2|880606923|
|    166|    346|     1|886397596|
|    298|    474|     4|884182806|
|    115|    265|     2|881171488|
|    253|    465|     5|891628467|
|    305|    451|     3|886324817|
|      6|     86|     3|883603013|
|     62|    257|     2|879372434|
|    286|   1014|     5|879781125|
|    200|    222|     5|876042340|
|    210|     40|     3|891035994|
|    224|     29|     3|888104457|
|    303|    785|     3|879485318|
|    122|    387|     5|879270459|
|    194|    274|     2|879539794|
|    291|   1042|     4|874834944|
|    234|   1184|     2|892079237|
+-------+-------+------+---------+
only showing top 20 rows
```

6.Show the numbers of items reviewed by each user in the newly created u_data table.

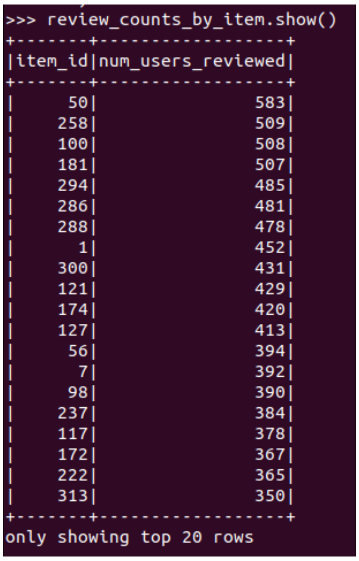
```
>>> review_counts_by_user = spark.sql("""
...         SELECT user_id, COUNT(item_id) AS num_items_reviewed
...         FROM u_data
...         GROUP BY user_id
...         ORDER BY num_items_reviewed DESC  """)

>>> review_counts_by_user.show()
```



```
>>> review_counts_by_user.show()
+-------+------------------+
|user_id|num_items_reviewed|
+-------+------------------+
|    405|               737|
|    655|               685|
|     13|               636|
|    450|               540|
|    276|               518|
|    416|               493|
|    537|               490|
|    303|               484|
|    234|               480|
|    393|               448|
|    181|               435|
|    279|               434|
|    429|               414|
|    846|               405|
|      7|               403|
|     94|               400|
|    682|               399|
|    308|               397|
|     92|               388|
|    293|               388|
+-------+------------------+
only showing top 20 rows
```

7.Show the numbers of users reviewed each item in the newly created u_data table

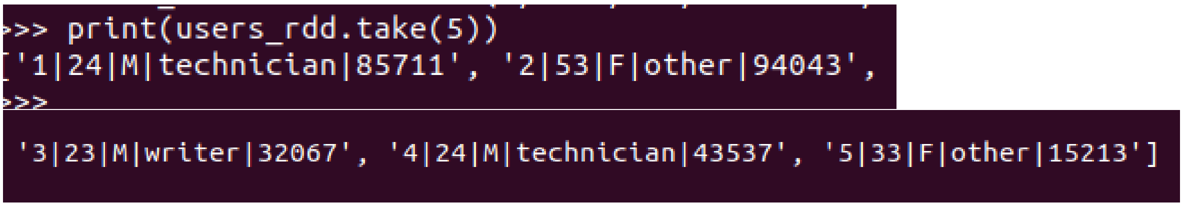
```
>>> review_counts_by_item = spark.sql("""
...        SELECT item_id, COUNT(user_id) AS num_users_reviewed
...        FROM u_data
...        GROUP BY item_id
...        ORDER BY num_users_reviewed DESC  """)

>>> review_counts_by_item.show()
```



```
>>> review_counts_by_item.show()
+-------+------------------+
|item_id|num_users_reviewed|
+-------+------------------+
|     50|               583|
|    258|               509|
|    100|               508|
|    181|               507|
|    294|               485|
|    286|               481|
|    288|               478|
|      1|               452|
|    300|               431|
|    121|               429|
|    174|               420|
|    127|               413|
|     56|               394|
|      7|               392|
|     98|               390|
|    237|               384|
|    117|               378|
|    172|               367|
|    222|               365|
|    313|               350|
+-------+------------------+
only showing top 20 rows
```

8.Load the u.user into a new RDD.

```
>>> users_rdd = sc.textFile("/home/mca/Downloads/u.user")

>>> print(users_rdd.take(5))
```

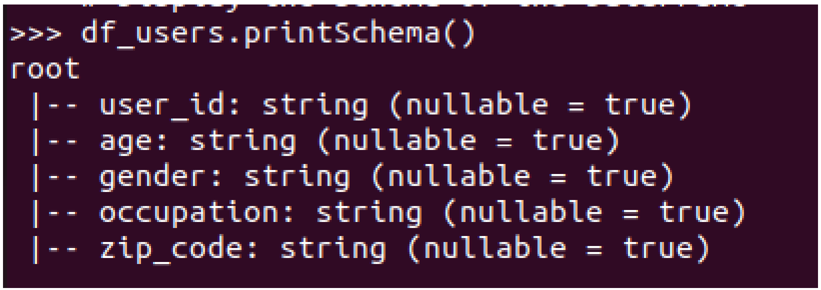


```
>> print(users_rdd.take(5))
'1|24|M|technician|85711', '2|53|F|other|94043',
>>
'3|23|M|writer|32067', '4|24|M|technician|43537', '5|33|F|other|15213']
```

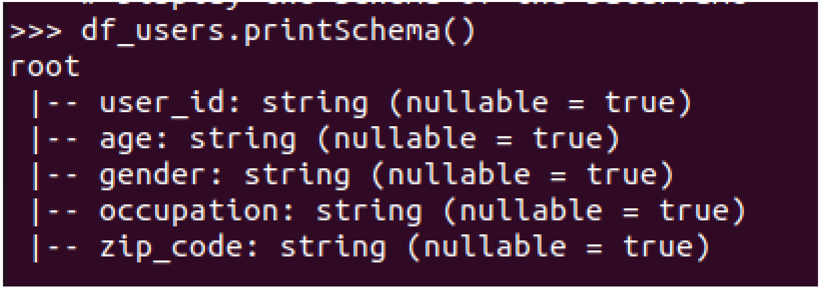
9.Change the RDD to a Dataframe.

```
>>> parsed_users_rdd = users_rdd.map(lambda line: line.split('|'))

>>> df_users = parsed_users_rdd.toDF(["user_id", "age", "gender", "occupation",
"zip_code"])
```

>>> df_users.printSchema()



```
>>> df_users.printSchema()
root
 |-- user_id: string (nullable = true)
 |-- age: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- occupation: string (nullable = true)
 |-- zip_code: string (nullable = true)
```



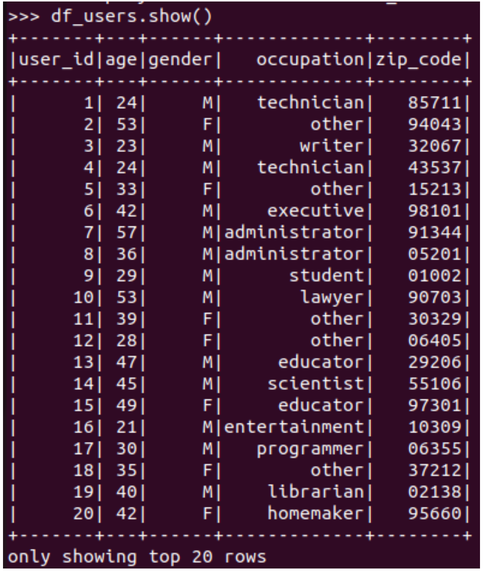
```
>>> df_users.printSchema()
root
 |-- user_id: string (nullable = true)
 |-- age: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- occupation: string (nullable = true)
 |-- zip_code: string (nullable = true)
```

10 Register the DataFrame as a temp u_user table.

>>> df_users.createOrReplaceTempView("u_user")

11.Display the contents of newly created user table

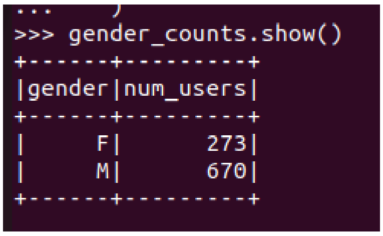
>>> # Display the contents of the u_user table
>>> df_users.show()



```
>>> df_users.show()
+-------+---+------+-------------+--------+
|user_id|age|gender|   occupation|zip_code|
+-------+---+------+-------------+--------+
|      1| 24|     M|   technician|   85711|
|      2| 53|     F|        other|   94043|
|      3| 23|     M|       writer|   32067|
|      4| 24|     M|   technician|   43537|
|      5| 33|     F|        other|   15213|
|      6| 42|     M|    executive|   98101|
|      7| 57|     M|administrator|   91344|
|      8| 36|     M|administrator|   05201|
|      9| 29|     M|      student|   01002|
|     10| 53|     M|       lawyer|   90703|
|     11| 39|     F|        other|   30329|
|     12| 28|     F|        other|   06405|
|     13| 47|     M|     educator|   29206|
|     14| 45|     M|    scientist|   55106|
|     15| 49|     F|     educator|   97301|
|     16| 21|     M|entertainment|   10309|
|     17| 30|     M|   programmer|   06355|
|     18| 35|     F|        other|   37212|
|     19| 40|     M|    librarian|   02138|
|     20| 42|     F|    homemaker|   95660|
+-------+---+------+-------------+--------+
only showing top 20 rows
```

12.Count the number of user in the u_user table gender wise

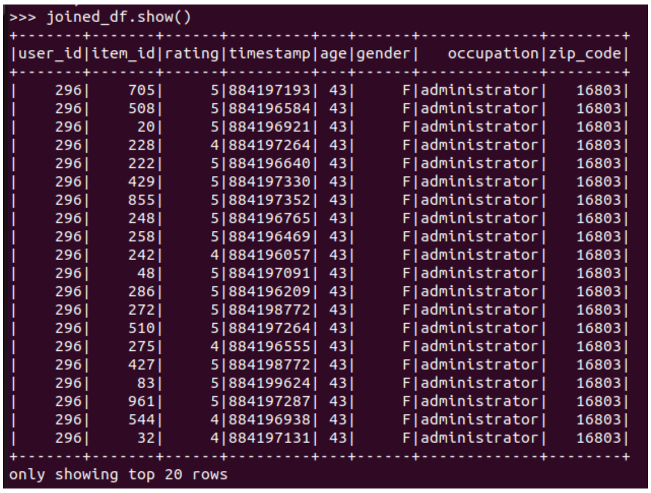
```
>>> gender_counts = spark.sql("""
...         SELECT gender, COUNT(user_id) AS num_users
...         FROM u_user
...         GROUP BY gender
... """)

>>> gender_counts.show()
```



```
...     )
>>> gender_counts.show()
+------+---------+
|gender|num_users|
+------+---------+
|     F|      273|
|     M|      670|
+------+---------+
```

13. Join u_data table and u_user tables based on userid

```
>>> df_ratings.createOrReplaceTempView("u_data")
>>> df_users.createOrReplaceTempView("u_user")

>>> joined_df = spark.sql("""
...         SELECT d.user_id, d.item_id, d.rating, d.timestamp, u.age, u.gender, u.occupation,
u.zip_code
...         FROM u_data d
...         JOIN u_user u
...         ON d.user_id = u.user_id
... """)
>>> joined_df.show()
```



```
>>> joined_df.show()
+-------+-------+------+---------+---+------+-------------+--------+
|user_id|item_id|rating|timestamp|age|gender|   occupation|zip_code|
+-------+-------+------+---------+---+------+-------------+--------+
|    296|    705|     5|884197193| 43|     F|administrator|   16803|
|    296|    508|     5|884196584| 43|     F|administrator|   16803|
|    296|     20|     5|884196921| 43|     F|administrator|   16803|
|    296|    228|     4|884197264| 43|     F|administrator|   16803|
|    296|    222|     5|884196640| 43|     F|administrator|   16803|
|    296|    429|     5|884197330| 43|     F|administrator|   16803|
|    296|    855|     5|884197352| 43|     F|administrator|   16803|
|    296|    248|     5|884196765| 43|     F|administrator|   16803|
|    296|    258|     5|884196469| 43|     F|administrator|   16803|
|    296|    242|     4|884196057| 43|     F|administrator|   16803|
|    296|     48|     5|884197091| 43|     F|administrator|   16803|
|    296|    286|     5|884196209| 43|     F|administrator|   16803|
|    296|    272|     5|884198772| 43|     F|administrator|   16803|
|    296|    510|     5|884197264| 43|     F|administrator|   16803|
|    296|    275|     4|884196555| 43|     F|administrator|   16803|
|    296|    427|     5|884198772| 43|     F|administrator|   16803|
|    296|     83|     5|884199624| 43|     F|administrator|   16803|
|    296|    961|     5|884197287| 43|     F|administrator|   16803|
|    296|    544|     4|884196938| 43|     F|administrator|   16803|
|    296|     32|     4|884197131| 43|     F|administrator|   16803|
+-------+-------+------+---------+---+------+-------------+--------+
only showing top 20 rows
```

6. Programming in Cassandra

Create KeySpace “Students”
Describe the existing Keyspaces
Display for More details on existing keyspaces
Use the keyspace “Students”
Student details:
Create table (column family) by name Student_Info(RollNo int, StudName text,
DateOfJoining
timestamp, PrevSemPercentage double), RollNo is primary key
Book_Borrowed
Create table (column family) by name Library_Book (CountValue counter, BookName
varchar,
RollNo int, StudName varchar), PRIMARY KEY is (book_name,stud_name));
Lookup the names of all tables in the current keyspace
Describe the table information
CRUD
Insert at least 5 rows for Student_Info
View data from the table “Students_Info”
View data from the table “Students_Info” where RollNo column either has a value 1 or 2 or 3
To execute a non primary key
Create an INDEX on the Column StudName
Execute the query based on the INDEXED Column:
Display students details for a specific student name.
Specify the number of rows to display in the output
Alias for Column:
Display RollNo as “USN”
UPDATEthe student name with last name for a specific RollNo
Change the RollNo to 10 for a existing RollNo with value 1.
DELETE PrevSemPercent for student with RollNo=2;
Delete a Row FROM student_info WHERE RollNo is 3;
Set Collection
Acolumn of type set consists of unordered unique values. However, when the column is
queried,
it returns the values in sorted order. For example, for text values, it sorts in alphabetical order.
Alter the StudentsInfo table to add hobbies as a set of text
Page|43
Department Of Computer Applications, BMSCE
22MCA2PEBD: Big Data Analytics
List Collection
When the order of elements matters, one should go for a list collection.
Alter the StudentsInfo table to add language as a list of text
Update the values for hobbies column (Music Cricket) and language column (Kannada,
Hindi,

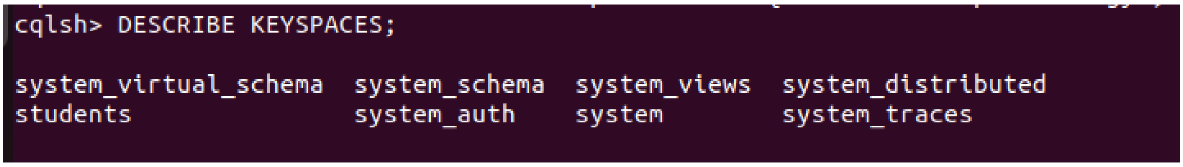
English) for RollNo with value 10 and display the student-info
Remove Hindi from the language list for RollNo 10 and display the student info
USING ACOUNTER
Acounter is a special column that is changed in increments. For example, we may need a counter column to count the number of times a particular book is issued from the library by the
student.
Load data into the counter column
Increase the counter column, CountValue by 1 in the table Library_Book for the student named
as "Ram" and book names as "Big data Analytics

1.Create KeySpace "Students"

>> CREATE KEYSPACE Students WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};

2.Describe the existing Keyspaces

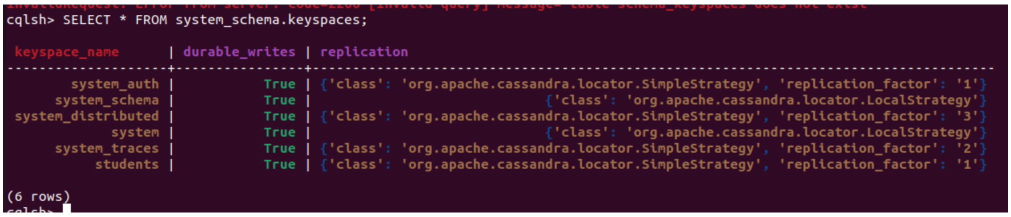
>> DESCRIBE KEYSPACES;



```
cqlsh> DESCRIBE KEYSPACES;

system_virtual_schema  system_schema  system_views  system_distributed
students               system_auth    system        system_traces
```

3.DESCRIBE KEYSPACE Students;

cqlsh> SELECT * FROM **students.**schema_keyspaces;



```
cqlsh> SELECT * FROM system_schema.keyspaces;

 keyspace_name      | durable_writes | replication
--------------------+----------------+-------------------------------------------------------------------------------------
        system_auth |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
      system_schema |           True |                             {'class': 'org.apache.cassandra.locator.LocalStrategy'}
 system_distributed |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
             system |           True |                             {'class': 'org.apache.cassandra.locator.LocalStrategy'}
      system_traces |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '2'}
           students |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}

(6 rows)
```

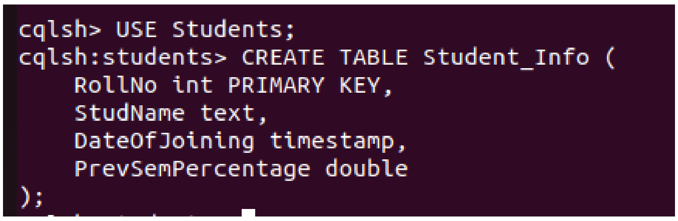
4.Use the keyspace "Students"

cqlsh:students>USE Students;

5.Student details:
Create table (column family) by name Student_Info(RollNo int, StudName text,

DateOfJoining timestamp, PrevSemPercentage double), RollNo is primary key

```
cqlsh:students> CREATE TABLE Student_Info (
        RollNo int PRIMARY KEY,
        StudName text,
        DateOfJoining timestamp,
        PrevSemPercentage double
);
```

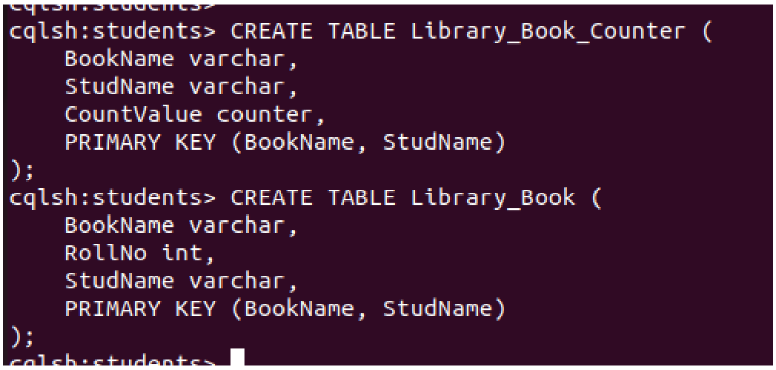


```
cqlsh> USE Students;
cqlsh:students> CREATE TABLE Student_Info (
    RollNo int PRIMARY KEY,
    StudName text,
    DateOfJoining timestamp,
    PrevSemPercentage double
);
```

6.Book_Borrowed
Create table (column family) by name Library_Book (CountValue counter, BookName varchar, RollNo int, StudName varchar), PRIMARY KEY is (book_name,stud_name));

```
cqlsh:students> CREATE TABLE Library_Book_Counter (
        BookName varchar,
        StudName varchar,
        CountValue counter,
        PRIMARY KEY (BookName, StudName)
);
```

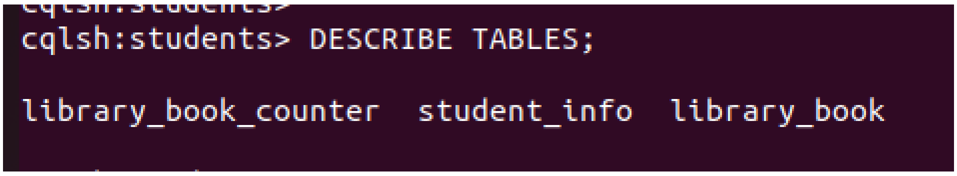
```
cqlsh:students> CREATE TABLE Library_Book (
        BookName varchar,
        RollNo int,
        StudName varchar,
        PRIMARY KEY (BookName, StudName)
);
```



```
cqlsh:students> CREATE TABLE Library_Book_Counter (
    BookName varchar,
    StudName varchar,
    CountValue counter,
    PRIMARY KEY (BookName, StudName)
);
cqlsh:students> CREATE TABLE Library_Book (
    BookName varchar,
    RollNo int,
    StudName varchar,
    PRIMARY KEY (BookName, StudName)
);
cqlsh:students>
```

7.Lookup the names of all tables in the current keyspace

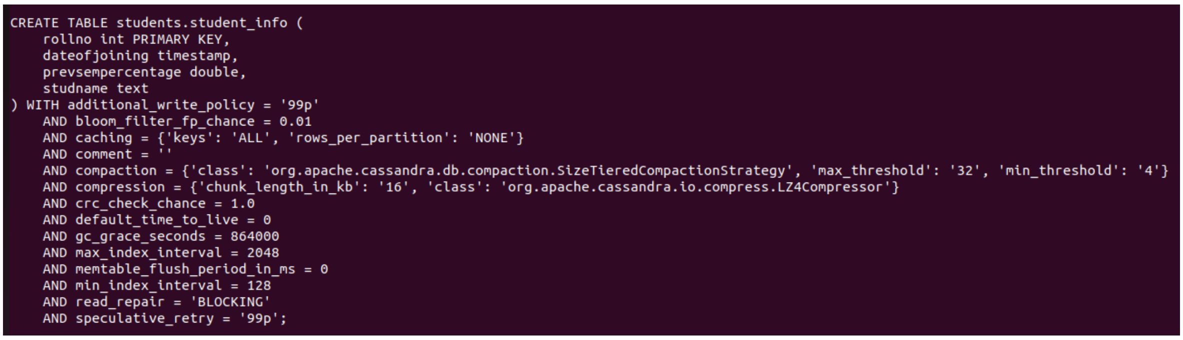
cqlsh:students> DESCRIBE TABLES;



```
cqlsh:students> DESCRIBE TABLES;

library_book_counter  student_info  library_book
```

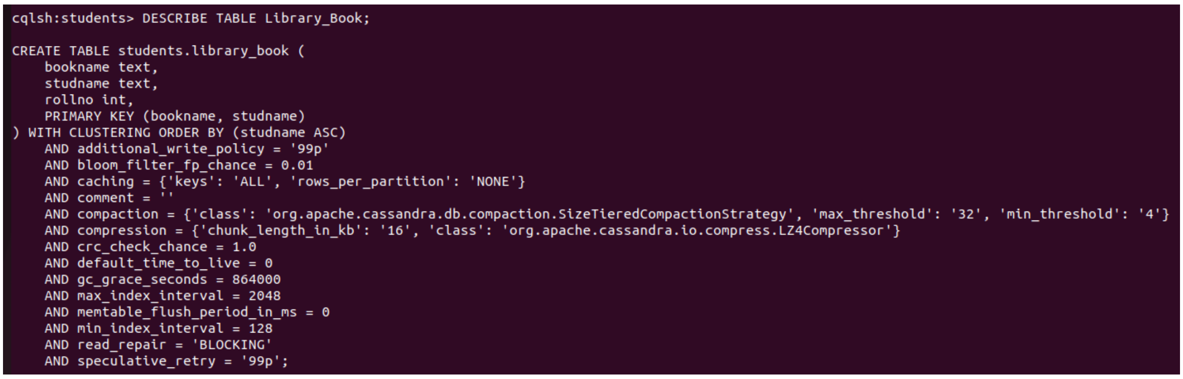
8.Describe the table information

cqlsh:students> DESCRIBE TABLE Student_Info;



```
CREATE TABLE students.student_info (
    rollno int PRIMARY KEY,
    dateofjoining timestamp,
    prevsempercentage double,
    studname text
) WITH additional_write_policy = '99p'
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair = 'BLOCKING'
    AND speculative_retry = '99p';
```

cqlsh:students> DESCRIBE TABLE Library_Book;



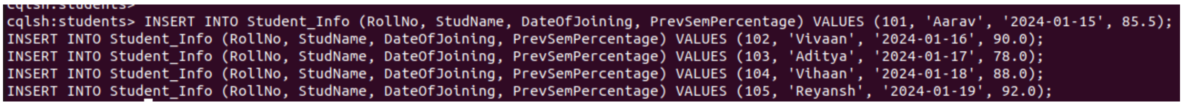
```
cqlsh:students> DESCRIBE TABLE Library_Book;

CREATE TABLE students.library_book (
    bookname text,
    studname text,
    rollno int,
    PRIMARY KEY (bookname, studname)
) WITH CLUSTERING ORDER BY (studname ASC)
    AND additional_write_policy = '99p'
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair = 'BLOCKING'
    AND speculative_retry = '99p';
```

9.CRUD

Insert at least 5 rows for Student_Info

cqlsh:students>

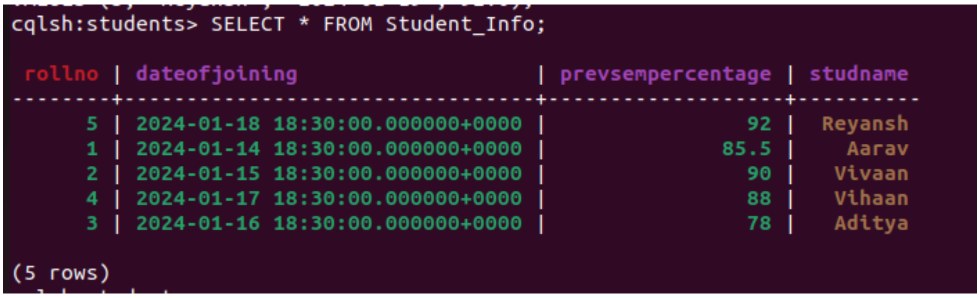
INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (101, 'Aarav', '2024-01-15', 85.5);
INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (102, 'Vivaan', '2024-01-16', 90.0);
INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (103, 'Aditya', '2024-01-17', 78.0);
INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (104, 'Vihaan', '2024-01-18', 88.0);
INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (105, 'Reyansh', '2024-01-19', 92.0);



```
cqlsh:students>
cqlsh:students> INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (101, 'Aarav', '2024-01-15', 85.5);
INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (102, 'Vivaan', '2024-01-16', 90.0);
INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (103, 'Aditya', '2024-01-17', 78.0);
INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (104, 'Vihaan', '2024-01-18', 88.0);
INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage) VALUES (105, 'Reyansh', '2024-01-19', 92.0);
```

10.View data from the table “Students_Info”

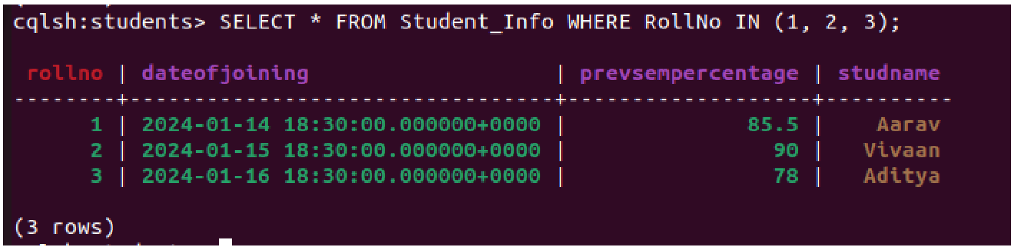
cqlsh:students> SELECT * FROM Student_Info;



```
cqlsh:students> SELECT * FROM Student_Info;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+----------
      5 | 2024-01-18 18:30:00.000000+0000 |                92 |  Reyansh
      1 | 2024-01-14 18:30:00.000000+0000 |              85.5 |    Aarav
      2 | 2024-01-15 18:30:00.000000+0000 |                90 |   Vivaan
      4 | 2024-01-17 18:30:00.000000+0000 |                88 |   Vihaan
      3 | 2024-01-16 18:30:00.000000+0000 |                78 |   Aditya

(5 rows)
```

11.View data from the table “Students_Info” where RollNo column either has a value 1 or 2 or 3

cqlsh:students> SELECT * FROM Student_Info WHERE RollNo IN (1, 2, 3);



```
cqlsh:students> SELECT * FROM Student_Info WHERE RollNo IN (1, 2, 3);

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+----------
      1 | 2024-01-14 18:30:00.000000+0000 |              85.5 |    Aarav
      2 | 2024-01-15 18:30:00.000000+0000 |                90 |   Vivaan
      3 | 2024-01-16 18:30:00.000000+0000 |                78 |   Aditya

(3 rows)
```

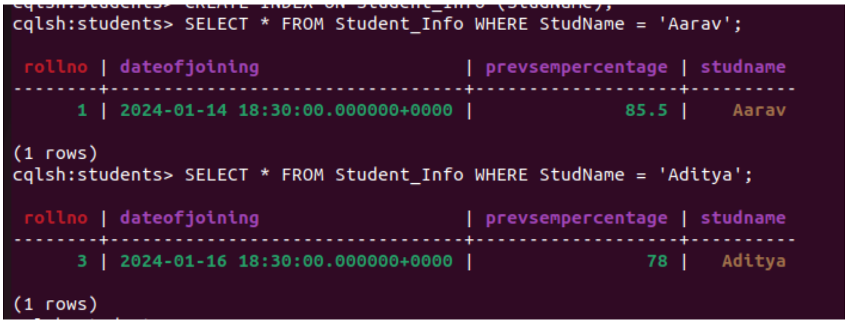
12. To execute a non primary key
Create an INDEX on the Column StudName

cqlsh:students> CREATE INDEX ON Student_Info (StudName);

13.Execute the query based on the INDEXED Column:
Display students details for a specific student name.

cqlsh:students> SELECT * FROM Student_Info WHERE StudName = 'Aarav';

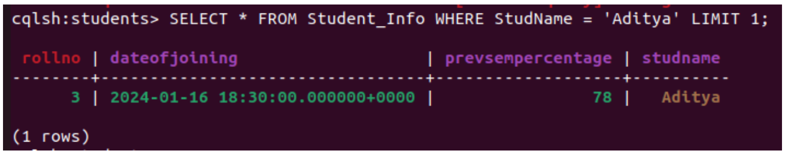
cqlsh:students> SELECT * FROM Student_Info WHERE StudName = 'Aditya';



```
cqlsh:students> SELECT * FROM Student_Info WHERE StudName = 'Aarav';

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+----------
      1 | 2024-01-14 18:30:00.000000+0000 |              85.5 |    Aarav

(1 rows)
cqlsh:students> SELECT * FROM Student_Info WHERE StudName = 'Aditya';

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+----------
      3 | 2024-01-16 18:30:00.000000+0000 |                78 |   Aditya

(1 rows)
```

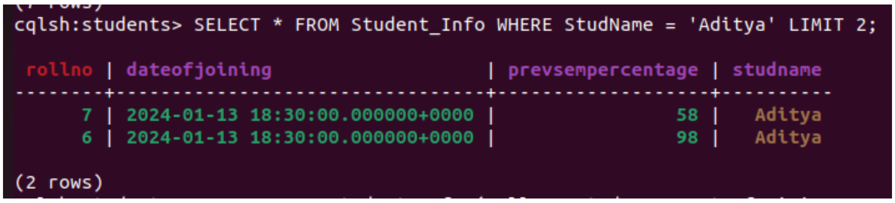
14. Specify the number of rows to display in the output

cqlsh:students> SELECT * FROM Student_Info WHERE StudName = 'Aditya' LIMIT 1;



```
cqlsh:students> SELECT * FROM Student_Info WHERE StudName = 'Aditya' LIMIT 1;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+----------
      3 | 2024-01-16 18:30:00.000000+0000 |                78 |   Aditya

(1 rows)
```

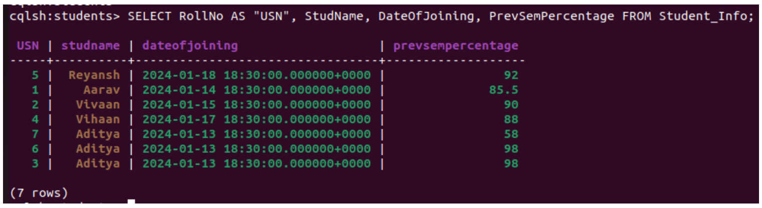
cqlsh:students> SELECT * FROM Student_Info WHERE StudName = 'Aditya' LIMIT 2;



```
cqlsh:students> SELECT * FROM Student_Info WHERE StudName = 'Aditya' LIMIT 2;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+----------
      7 | 2024-01-13 18:30:00.000000+0000 |                58 |   Aditya
      6 | 2024-01-13 18:30:00.000000+0000 |                98 |   Aditya

(2 rows)
```

15.Alias for Column:
Display RollNo as “USN”

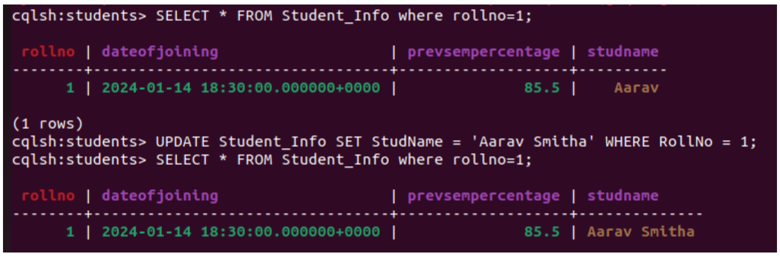
cqlsh:students> SELECT RollNo AS "USN", StudName, DateOfJoining, PrevSemPercentage FROM Student_Info;



```
cqlsh:students> SELECT RollNo AS "USN", StudName, DateOfJoining, PrevSemPercentage FROM Student_Info;

 USN | studname | dateofjoining                   | prevsempercentage
-----+----------+---------------------------------+-------------------
   5 |  Reyansh | 2024-01-18 18:30:00.000000+0000 |                92
   1 |    Aarav | 2024-01-14 18:30:00.000000+0000 |              85.5
   2 |   Vivaan | 2024-01-15 18:30:00.000000+0000 |                90
   4 |   Vihaan | 2024-01-17 18:30:00.000000+0000 |                88
   7 |   Aditya | 2024-01-13 18:30:00.000000+0000 |                58
   6 |   Aditya | 2024-01-13 18:30:00.000000+0000 |                98
   3 |   Aditya | 2024-01-13 18:30:00.000000+0000 |                98

(7 rows)
```

16.UPDATE the student name with last name for a specific RollNo

cqlsh:students> UPDATE Student_Info SET StudName = 'Aarav Smitha' WHERE RollNo = 1;

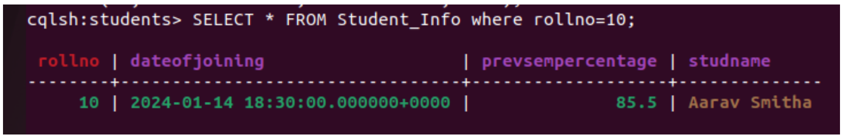


```
cqlsh:students> SELECT * FROM Student_Info where rollno=1;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+----------
      1 | 2024-01-14 18:30:00.000000+0000 |              85.5 |    Aarav

(1 rows)
cqlsh:students> UPDATE Student_Info SET StudName = 'Aarav Smitha' WHERE RollNo = 1;
cqlsh:students> SELECT * FROM Student_Info where rollno=1;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+--------------
      1 | 2024-01-14 18:30:00.000000+0000 |              85.5 | Aarav Smitha
```

Change the RollNo to 10 for a existing RollNo with value 1.

cqlsh:students> INSERT INTO Student_Info (RollNo, StudName, DateOfJoining, PrevSemPercentage)
VALUES (10, 'Aarav Smitha', '2024-01-15', 85.5);

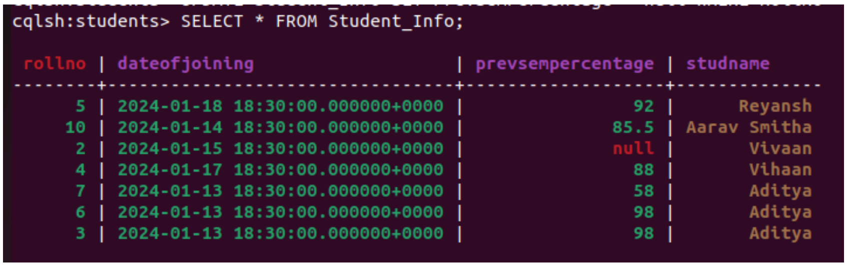
cqlsh:students> DELETE FROM Student_Info WHERE RollNo = 1;



```
cqlsh:students> SELECT * FROM Student_Info where rollno=10;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+--------------
     10 | 2024-01-14 18:30:00.000000+0000 |              85.5 | Aarav Smitha
```

17.DELETE PrevSemPercent for student with RollNo=2;

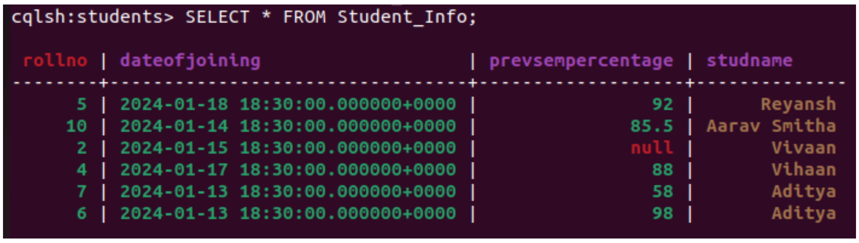
cqlsh:students> UPDATE Student_Info SET PrevSemPercentage = null WHERE RollNo = 2;



```
cqlsh:students> SELECT * FROM Student_Info;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+--------------
      5 | 2024-01-18 18:30:00.000000+0000 |                92 |      Reyansh
     10 | 2024-01-14 18:30:00.000000+0000 |              85.5 | Aarav Smitha
      2 | 2024-01-15 18:30:00.000000+0000 |              null |       Vivaan
      4 | 2024-01-17 18:30:00.000000+0000 |                88 |       Vihaan
      7 | 2024-01-13 18:30:00.000000+0000 |                58 |       Aditya
      6 | 2024-01-13 18:30:00.000000+0000 |                98 |       Aditya
      3 | 2024-01-13 18:30:00.000000+0000 |                98 |       Aditya
```

18.Delete a Row FROM student_info WHERE RollNo is 3;

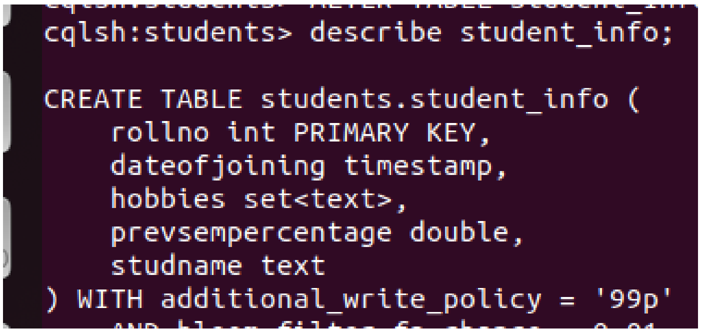
cqlsh:students> DELETE FROM Student_Info WHERE RollNo = 3;



```
cqlsh:students> SELECT * FROM Student_Info;

 rollno | dateofjoining                   | prevsempercentage | studname
--------+---------------------------------+-------------------+--------------
      5 | 2024-01-18 18:30:00.000000+0000 |                92 |      Reyansh
     10 | 2024-01-14 18:30:00.000000+0000 |              85.5 | Aarav Smitha
      2 | 2024-01-15 18:30:00.000000+0000 |              null |       Vivaan
      4 | 2024-01-17 18:30:00.000000+0000 |                88 |       Vihaan
      7 | 2024-01-13 18:30:00.000000+0000 |                58 |       Aditya
      6 | 2024-01-13 18:30:00.000000+0000 |                98 |       Aditya
```

19.Set Collection
A column of type set consists of unordered unique values. However, when the column is queried, it returns the values in sorted order. For example, for text values, it sorts in alphabetical order.

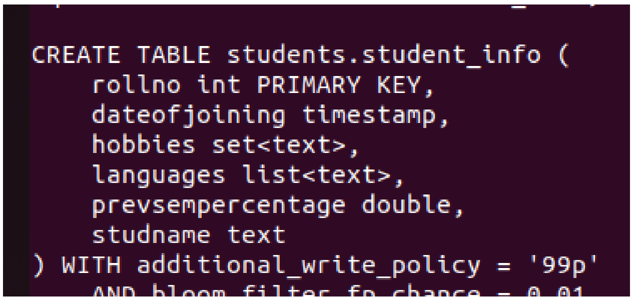
cqlsh:students> ALTER TABLE Student_Info ADD hobbies SET<text>;



```
cqlsh:students> describe student_info;

CREATE TABLE students.student_info (
    rollno int PRIMARY KEY,
    dateofjoining timestamp,
    hobbies set<text>,
    prevsempercentage double,
    studname text
) WITH additional_write_policy = '99p'
    AND bloom_filter_fp_chance = 0.01
```

20.List Collection
When the order of elements matter, one should go for a list collection.
Alter the StudentsInfo table to add language as a list of text

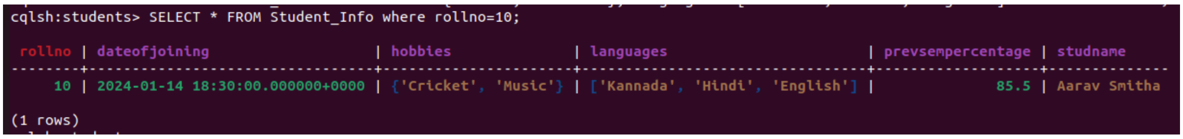
cqlsh:students> ALTER TABLE Student_Info ADD languages LIST<text>;



```
CREATE TABLE students.student_info (
    rollno int PRIMARY KEY,
    dateofjoining timestamp,
    hobbies set<text>,
    languages list<text>,
    prevsempercentage double,
    studname text
) WITH additional_write_policy = '99p'
    AND bloom_filter_fp_chance = 0.01
```

21.Update the values for hobbies column (Music Cricket) and language column (Kannada, Hindi,English) for RollNo with value 10 and display the student-info

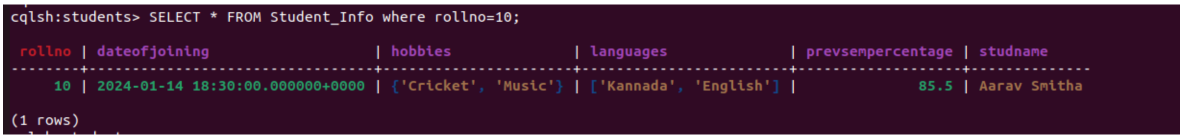
cqlsh:students> UPDATE Student_Info SET hobbies = {'Music', 'Cricket'}, languages = ['Kannada', 'Hindi', 'English'] WHERE RollNo = 10;



```
cqlsh:students> SELECT * FROM Student_Info where rollno=10;

 rollno | dateofjoining                   | hobbies              | languages                       | prevsempercentage | studname
--------+---------------------------------+----------------------+---------------------------------+-------------------+--------------
     10 | 2024-01-14 18:30:00.000000+0000 | {'Cricket', 'Music'} | ['Kannada', 'Hindi', 'English'] |              85.5 | Aarav Smitha

(1 rows)
```

22.Remove Hindi from the language list for RollNo 10 and display the student info

cqlsh:students> UPDATE Student_Info SET languages = languages - ['Hindi'] WHERE RollNo = 10;

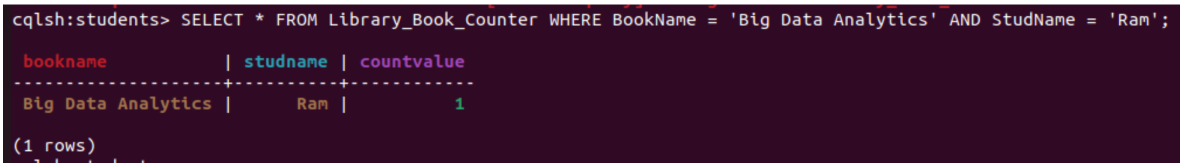


```
cqlsh:students> SELECT * FROM Student_Info where rollno=10;

 rollno | dateofjoining                   | hobbies              | languages              | prevsempercentage | studname
--------+---------------------------------+----------------------+------------------------+-------------------+--------------
     10 | 2024-01-14 18:30:00.000000+0000 | {'Cricket', 'Music'} | ['Kannada', 'English'] |              85.5 | Aarav Smitha

(1 rows)
```

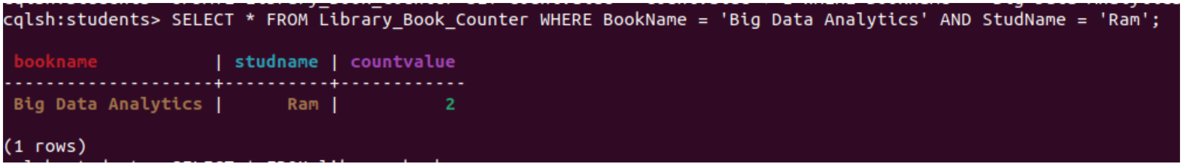
23.USING A COUNTER

A counter is a special column that is changed in increments. For example, we may need a counter column to count the number of times a particular book is issued from the library bythe student.

cqlsh:students> UPDATE Library_Book_Counter SET CountValue = CountValue + 1 WHERE BookName = 'Big Data Analytics' AND StudName = 'Ram';



```
cqlsh:students> SELECT * FROM Library_Book_Counter WHERE BookName = 'Big Data Analytics' AND StudName = 'Ram';

 bookname           | studname | countvalue
--------------------+----------+------------
 Big Data Analytics |      Ram |          1

(1 rows)
```



```
cqlsh:students> SELECT * FROM Library_Book_Counter WHERE BookName = 'Big Data Analytics' AND StudName = 'Ram';

 bookname           | studname | countvalue
--------------------+----------+------------
 Big Data Analytics |      Ram |          2

(1 rows)
```