VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Big Data Analytics (23CS6PEBDA)

Submitted by

Jaydev P (1BM22CS118)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Feb-2025 to July-2025

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**Big Data Analytics (22CS6PEBDA**)" carried out by **Jaydev P(1BM22CS118),** who is a bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024. The Lab report has been approved as it satisfies the academic requirements in respect of a **Big Data Analytics-(23CS6PCBDA**) work prescribed for the said degree.

Sneha P
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Kavitha Sooda
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index

GitHub link: https://github.com/jaydevpolur/6C_BDA_Lab.git

LAB 1 - MongoDB- CRUD Operations Demonstration (Practice and Self Study)

OUTPUT:

```
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student>mongosh "mongodb+srv://cluster0.qh8blz4.mongodb.net/" --apiVersion 1 --username likhithcs22
Enter password: **********
Current Mongosh Log ID: 67c6c754899c67e814fa4213
Connecting to:          mongodb+srv://<credentials>@cluster0.qh8blz4.mongodb.net/?appName=mongosh+2.4.0
Using MongoDB:          8.0.5 (API Version 1)
Using Mongosh:          2.4.0

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

Atlas atlas-2vljb9-shard-0 [primary] test> show dbs
e-commerce  108.00 KiB
myDB         40.00 KiB
admin       232.00 KiB
local        15.70 GiB
Atlas atlas-2vljb9-shard-0 [primary] test> use myDB
switched to db myDB
Atlas atlas-2vljb9-shard-0 [primary] myDB> db
myDB
Atlas atlas-2vljb9-shard-0 [primary] myDB> db.createCollection("Student");
{ ok: 1 }
Atlas atlas-2vljb9-shard-0 [primary] myDB> db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});
...
... db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});
...
... db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});
...
... db.Student.insert({RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"});
...
... db.Student.insert({RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c898899c67e814fa4218') }
}
Atlas atlas-2vljb9-shard-0 [primary] myDB> db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c8a3899c67e814fa4219') }
}
Atlas atlas-2vljb9-shard-0 [primary] myDB> db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c8f7899c67e814fa421a') }
}
Atlas atlas-2vljb9-shard-0 [primary] myDB> db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c8fb899c67e814fa421b') }
```

```
C:\Users\likhi>mongosh "mongodb+srv://cluster0.qh8blz4.mongodb.net/" --apiVersion 1 --username likhithcs22
Enter password: **********
Current Mongosh Log ID: 6833148466c722794490defd
Connecting to:          mongodb+srv://<credentials>@cluster0.qh8blz4.mongodb.net/?appName=mongosh+2.2.9
Using MongoDB:          8.0.9 (API Version 1)
Using Mongosh:          2.2.9
mongosh 2.5.1 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Atlas atlas-2vljb9-shard-0 [primary] test> show dbs
e-commerce  108.00 KiB
myDB         72.00 KiB
admin       312.00 KiB
local        64.34 GiB
Atlas atlas-2vljb9-shard-0 [primary] test> use myDB
switched to db myDB
Atlas atlas-2vljb9-shard-0 [primary] myDB> db
myDB
Atlas atlas-2vljb9-shard-0 [primary] myDB> show collections
Student
```

LAB 2:MongoDB

OUTPUT:

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
}
myDB> db.Customers.aggregate ( {$match:{Type:"Z"}},
... {$group : { _id : "$cust_id",
... TotAccBal :{$sum:"$Balance"} } }, {$match:{TotAccBal:{$gt:1200}}});
[
  { _id: 3, TotAccBal: 2300 },
  { _id: 4, TotAccBal: 2300 },
  { _id: 2, TotAccBal: 11200 }
]
myDB> db.Customers.aggregate (
... {$group : { _id : "$cust_id",
... minAccBal :{$min:"$Balance"}, maxAccBal :{$max:"$Balance"} }});
[
  { _id: 3, minAccBal: 2300, maxAccBal: 2300 },
  { _id: 1, minAccBal: 200, maxAccBal: 200 },
  { _id: 4, minAccBal: 2300, maxAccBal: 2300 },
  { _id: 5, minAccBal: 2300, maxAccBal: 2300 },
  { _id: 2, minAccBal: 200, maxAccBal: 11200 }
]
myDB> exit
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoexport --host localhost --db myDB --collection CUstomers --type=csv --out /home/bmscecse/o.txt --fields "Balance","Type"
2025-03-11T15:21:30.413+0530    connected to: mongodb://localhost/
2025-03-11T15:21:30.413+0530    exported 0 records
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoexport --host localhost --db myDB --collection Customers --type=csv --out /home/bmscecse/o.txt --fields "Balance","Type"
2025-03-11T15:21:47.812+0530    connected to: mongodb://localhost/
2025-03-11T15:21:47.814+0530    exported 6 records
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongosh
Current Mongosh Log ID: 67d007c15778809737567a2a
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.7
Using MongoDB:          7.0.16
Using Mongosh:          2.3.7
mongosh 2.4.2 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-03-11T14:05:19.345+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
   2025-03-11T14:05:22.471+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> use myDB
switched to db myDB
myDB> db.Customers.drop()
true
myDB> exit
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoimport --db myDB --collection newCust --type=csv --headerline --file /home/bmscecse/o/txt
2025-03-11T15:24:08.278+0530    Failed: open /home/bmscecse/o/txt: no such file or directory
2025-03-11T15:24:08.278+0530    0 document(s) imported successfully. 0 document(s) failed to import.
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoimport --db myDB --collection Customers --type=csv --headerline --file /home/bmscecse/o.txt
2025-03-11T15:24:56.973+0530    connected to: mongodb://localhost/
2025-03-11T15:24:57.328+0530    6 document(s) imported successfully. 0 document(s) failed to import.
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongosh
Current Mongosh Log ID: 67d00878022dc653c5567a2a
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.7
Using MongoDB:          7.0.16
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
2025-03-11T15:24:08.278+0530    0 document(s) imported successfully. 0 document(s) failed to import.
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoimport --db myDB --collection Customers --type=csv --headerline --file /home/bmscecse/o.txt
2025-03-11T15:24:56.973+0530    connected to: mongodb://localhost/
2025-03-11T15:24:57.328+0530    6 document(s) imported successfully. 0 document(s) failed to import.
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongosh
Current Mongosh Log ID: 67d00878022dc653c5567a2a
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.7
Using MongoDB:          7.0.16
Using Mongosh:          2.3.7
mongosh 2.4.2 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-03-11T14:05:19.345+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
   2025-03-11T14:05:22.471+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> use myDB
switched to db myDB
myDB> db.Customers.find()
[
  {
    _id: ObjectId('67d008706d2ac454920b0abf'),
    Balance: 11200,
    Type: 'Z'
  },
  {
    _id: ObjectId('67d008706d2ac454920b0ac0'),
    Balance: 200,
    Type: 'S'
  },
  {
    _id: ObjectId('67d008706d2ac454920b0ac1'),
    Balance: 2300,
    Type: 'Z'
  },
  {
    _id: ObjectId('67d008706d2ac454920b0ac2'),
    Balance: 2300,
    Type: 'Z'
  },
  {
    _id: ObjectId('67d008706d2ac454920b0ac3'),
    Balance: 200,
    Type: 'S'
  },
  {
    _id: ObjectId('67d008706d2ac454920b0ac4'),
    Balance: 2300,
    Type: 'S'
  }
]
myDB>
```

LAB 3:CASSANDRA

OUTPUT:

```
cqlsh> CREATE KEYSPACE Employee WITH replication = {'class':'SimpleStrategy', 'replication_factor':1};
cqlsh> USE Employee;
cqlsh:employee> CREATE TABLE Employee_Info (
            ...     Emp_Id int PRIMARY KEY,
            ...     Emp_Name text,
            ...     Designation text,
            ...     Date_of_Joining date,
            ...     Salary decimal,
            ...     Dept_Name text
            ... );
cqlsh:employee> BEGIN BATCH
            ... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
            ... VALUES (121, 'John Doe', 'Manager', '2015-06-20', 75000, 'HR');
            ... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
            ... VALUES (122, 'Jane Smith', 'Engineer', '2017-08-15', 60000, 'IT');
            ... APPLY BATCH;
cqlsh:employee> SELECT * FROM Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name   | salary
--------+-----------------+-----------+-------------+------------+--------
    122 |      2017-08-15 |        IT |    Engineer | Jane Smith |  60000
    121 |      2015-06-20 |        HR |     Manager |   John Doe |  75000

(2 rows)
cqlsh:employee> UPDATE Employee_Info SET Emp_Name = 'John Wick', Dept_Name = 'Security' WHERE Emp_Id = 121;
cqlsh:employee> SELECT * FROM Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name   | salary
--------+-----------------+-----------+-------------+------------+--------
    122 |      2017-08-15 |        IT |    Engineer | Jane Smith |  60000
    121 |      2015-06-20 |  Security |     Manager |  John Wick |  75000
```

```
cqlsh> CREATE KEYSPACE IF NOT EXISTS Employee
   ... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> USE Employee;
cqlsh:employee> CREATE TABLE IF NOT EXISTS Employee_Info (
            ...   Emp_Id INT PRIMARY KEY,
            ...   Emp_Name TEXT,
            ...   Designation TEXT,
            ...   Date_of_Joining DATE,
            ...   Salary DOUBLE,
            ...   Dept_Name TEXT
            ... );
cqlsh:employee> BEGIN BATCH
            ... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
            ... VALUES (121, 'John Doe', 'Manager', '2018-01-01', 90000, 'HR');
            ...
            ... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
            ... VALUES (122, 'Alice Smith', 'Developer', '2019-05-21', 75000, 'IT');
            ...
            ... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
            ... VALUES (123, 'Rahul Roy', 'Analyst', '2020-07-15', 65000, 'IT');
            ... APPLY BATCH;
cqlsh:employee> UPDATE Employee_Info
            ... SET Emp_Name = 'John Smith', Dept_Name = 'Finance'
            ... WHERE Emp_Id = 121;
cqlsh:employee> select * from Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name    | salary
--------+-----------------+-----------+-------------+-------------+--------
    123 |      2020-07-15 |        IT |     Analyst |   Rahul Roy |  65000
    122 |      2019-05-21 |        IT |   Developer | Alice Smith |  75000
    121 |      2018-01-01 |   Finance |     Manager |  John Smith |  90000

(3 rows)
```

```
(3 rows)
cqlsh:employee> CREATE TABLE IF NOT EXISTS Employee_By_Dept (
            ...   Dept_Name TEXT,
            ...   Salary DOUBLE,
            ...   Emp_Id INT,
            ...   Emp_Name TEXT,
            ...   Designation TEXT,
            ...   Date_of_Joining DATE,
            ...   PRIMARY KEY (Dept_Name, Salary, Emp_Id)
            ... ) WITH CLUSTERING ORDER BY (Salary DESC, Emp_Id ASC);
cqlsh:employee> BEGIN BATCH
            ... INSERT INTO Employee_By_Dept (Dept_Name, Salary, Emp_Id, Emp_Name, Designation, Date_of_Joining)
            ... VALUES ('HR', 90000, 121, 'John Smith', 'Manager', '2018-01-01');
            ...
            ... INSERT INTO Employee_By_Dept (Dept_Name, Salary, Emp_Id, Emp_Name, Designation, Date_of_Joining)
            ... VALUES ('IT', 75000, 122, 'Alice Smith', 'Developer', '2019-05-21');
            ...
            ... INSERT INTO Employee_By_Dept (Dept_Name, Salary, Emp_Id, Emp_Name, Designation, Date_of_Joining)
            ... VALUES ('IT', 65000, 123, 'Rahul Roy', 'Analyst', '2020-07-15');
            ... APPLY BATCH;
cqlsh:employee> SELECT * FROM Employee_By_Dept WHERE Dept_Name = 'IT';

 dept_name | salary | emp_id | date_of_joining | designation | emp_name
-----------+--------+--------+-----------------+-------------+-------------
        IT |  75000 |    122 |      2019-05-21 |   Developer | Alice Smith
        IT |  65000 |    123 |      2020-07-15 |     Analyst |   Rahul Roy

(2 rows)
cqlsh:employee> ALTER TABLE Employee_Info ADD Projects SET<TEXT>;
cqlsh:employee> UPDATE Employee_Info SET Projects = {'ERP System', 'HR Portal'} WHERE Emp_Id = 121;
cqlsh:employee> INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
            ... VALUES (124, 'Sneha Kapoor', 'Tester', '2023-03-10', 55000, 'QA') USING TTL 15;
cqlsh:employee> select * from Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name    | projects                    | salary
--------+-----------------+-----------+-------------+-------------+-----------------------------+--------
    123 |      2020-07-15 |        IT |     Analyst |   Rahul Roy |                        null |  65000
    122 |      2019-05-21 |        IT |   Developer | Alice Smith |                        null |  75000
    121 |      2018-01-01 |   Finance |     Manager |  John Smith | {'ERP System', 'HR Portal'} |  90000

(3 rows)
```

LAB 04:CASSANDRA

OUTPUT:

```
cqlsh:employee> CREATE TABLE Employee_By_Dept (
           ...     Dept_Name text,
           ...     Salary decimal,
           ...     Emp_Id int,
           ...     Emp_Name text,
           ...     Designation text,
           ...     Date_of_Joining date,
           ...     PRIMARY KEY (Dept_Name, Salary)
           ... ) WITH CLUSTERING ORDER BY (Salary DESC);
cqlsh:employee>
cqlsh:employee>
cqlsh:employee>
cqlsh:employee> INSERT INTO Employee_By_Dept (Dept_Name, Salary, Emp_Id, Emp_Name, Designation, Date_of_Joining)
           ... VALUES ('IT', 60000, 122, 'Jane Smith', 'Engineer', '2017-08-15');
cqlsh:employee> INSERT INTO Employee_By_Dept (Dept_Name, Salary, Emp_Id, Emp_Name, Designation, Date_of_Joining)
           ... VALUES ('Security', 75000, 122, 'John Wick', 'Manager', '2015-06-20');
cqlsh:employee>
cqlsh:employee> INSERT INTO Employee_By_Dept (Dept_Name, Salary, Emp_Id, Emp_Name, Designation, Date_of_Joining)
           ... VALUES ('IT', 80000, 123, 'Alice', 'Senior Engineer', '2015-04-10');
cqlsh:employee> SELECT * FROM Employee_By_Dept WHERE Dept_Name = 'IT';

 dept_name | salary | date_of_joining | designation     | emp_id | emp_name
-----------+--------+-----------------+-----------------+--------+------------
        IT |  80000 |      2015-04-10 | Senior Engineer |    123 |      Alice
        IT |  60000 |      2017-08-15 |        Engineer |    122 | Jane Smith

(2 rows)
cqlsh:employee> ALTER TABLE Employee_Info ADD Projects list<text>;
cqlsh:employee> UPDATE Employee_Info SET Projects = ['Website Revamp', 'Cloud Migration'] WHERE Emp_Id = 121;
cqlsh:employee> INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
           ... VALUES (123, 'Temp User', 'Intern', '2024-01-01', 30000, 'Temp') USING TTL 15;
cqlsh:employee> SELECT * FROM Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name   | projects                                | salary
--------+-----------------+-----------+-------------+------------+-----------------------------------------+--------
    123 |      2024-01-01 |      Temp |      Intern |  Temp User |                                    null |  30000
    122 |      2017-08-15 |        IT |    Engineer | Jane Smith |                                    null |  60000
    121 |      2015-06-20 |  Security |     Manager |  John Wick | ['Website Revamp', 'Cloud Migration'] |  75000
```

```
cqlsh:employee> CREATE KEYSPACE IF NOT EXISTS Library
           ... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh:employee> USE Library;
cqlsh:library> CREATE TABLE IF NOT EXISTS Library_Info (
           ...   Stud_Id INT PRIMARY KEY,
           ...   Stud_Name TEXT,
           ...   Book_Name TEXT,
           ...   Book_Id TEXT,
           ...   Date_of_issue DATE
           ... );
cqlsh:library> CREATE TABLE IF NOT EXISTS Book_Counter (
           ...   Stud_Id INT,
           ...   Book_Name TEXT,
           ...   Counter_value COUNTER,
           ...   PRIMARY KEY ((Stud_Id), Book_Name)
           ... );
cqlsh:library> BEGIN BATCH
           ... INSERT INTO Library_Info (Stud_Id, Stud_Name, Book_Name, Book_Id, Date_of_issue)
           ... VALUES (112, 'Anjali Rao', 'BDA', 'B101', '2024-10-01');
           ...
           ... INSERT INTO Library_Info (Stud_Id, Stud_Name, Book_Name, Book_Id, Date_of_issue)
           ... VALUES (113, 'Karthik N', 'AI', 'B102', '2024-11-11');
           ... APPLY BATCH;
cqlsh:library> UPDATE Book_Counter SET Counter_value = Counter_value + 1 WHERE Stud_Id = 112 AND Book_Name = 'BDA';
cqlsh:library> UPDATE Book_Counter SET Counter_value = Counter_value + 1 WHERE Stud_Id = 112 AND Book_Name = 'BDA';
cqlsh:library> SELECT * FROM Book_Counter WHERE Stud_Id = 112 AND Book_Name = 'BDA';

 stud_id | book_name | counter_value
---------+-----------+---------------
     112 |       BDA |             4

(1 rows)
```

```
cqlsh:students> DESCRIBE TABLE Students_Info;

CREATE TABLE students.students_info (
    roll_no int PRIMARY KEY,
    dateofjoining timestamp,
    last_exam_percent double,
    studname text
) WITH additional_write_policy = '99p'
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND cdc = false
    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 memtable = 'default'
    AND crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND extensions = {}
    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> BEGIN BATCH
            ... INSERT INTO Students_Info (Roll_No, StudName, DateOfJoining, Last_Exam_Percent)
            ... VALUES (1, 'Asha', '2012-03-12', 79.9);
            ... INSERT INTO Students_Info (Roll_No, StudName, DateOfJoining, Last_Exam_Percent)
            ... VALUES (2, 'Kiran', '2012-03-12', 89.9);
            ... INSERT INTO Students_Info (Roll_No, StudName, DateOfJoining, Last_Exam_Percent)
            ... VALUES (3, 'Shanthi', '2012-03-12', 90.9);
            ... INSERT INTO Students_Info (Roll_No, StudName, DateOfJoining, Last_Exam_Percent)
            ... VALUES (4, 'Smith', '2012-03-12', 67.9);
            ... INSERT INTO Students_Info (Roll_No, StudName, DateOfJoining, Last_Exam_Percent)
            ... VALUES (5, 'Rohan', '2012-03-12', 56.9);
            ... APPLY BATCH;
cqlsh:students> SELECT * FROM Students_Info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       5 | 2012-03-11 18:30:00.000000+0000 |              56.9 |    Rohan
       1 | 2012-03-11 18:30:00.000000+0000 |              79.9 |     Asha
       2 | 2012-03-11 18:30:00.000000+0000 |              89.9 |    Kiran
       4 | 2012-03-11 18:30:00.000000+0000 |              67.9 |    Smith
       3 | 2012-03-11 18:30:00.000000+0000 |              90.9 |  Shanthi

(5 rows)
```

```
cqlsh> CREATE KEYSPACE Students WITH REPLICATION =
   ... {'class': 'SimpleStrategy', 'replication_factor': '1'};
cqlsh>
cqlsh> USE Students;
cqlsh:students> DESCRIBE KEYSPACES;

companies  library      products    system              system_traces
company    pro          productss   system_auth         system_views
employe    prod         productsss  system_distributed  system_virtual_schema
employee   productname  students    system_schema

cqlsh:students> CREATE TABLE Students_Info (
            ...     Roll_No int PRIMARY KEY,
            ...     StudName text,
            ...     DateOfJoining timestamp,
            ...     last_exam_Percent double
            ... );
cqlsh:students> SELECT * FROM systen.schema_keyspaces;
InvalidRequest: Error from server: code=2200 [Invalid query] message="table schema_keyspaces does not exist"
cqlsh:students>
cqlsh:students> SELECT * FROM system_schema.keyspaces;

 keyspace_name      | durable_writes | replication
--------------------+----------------+-------------------------------------------------------------------------------
          companies |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
        systen_auth |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
       systen_schema |           True |                                 {'class': 'org.apache.cassandra.locator.LocalStrategy'}
            library |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
           products |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
 systen_distributed |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
             system |           True |                                 {'class': 'org.apache.cassandra.locator.LocalStrategy'}
         productsss |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
               prod |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
                pro |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
      systen_traces |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '2'}
           students |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
            company |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
           employee |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
        productname |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
            employe |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
           productss |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}

(17 rows)
cqlsh:students> DESCRIBE TABLES;

students_info
```

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

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2012-03-11 18:30:00.000000+0000 |              79.9 |     Asha
       2 | 2012-03-11 18:30:00.000000+0000 |              89.9 |    Kiran
       3 | 2012-03-11 18:30:00.000000+0000 |              90.9 |  Shanthi

(3 rows)
cqlsh:students> CREATE INDEX ON Students_Info (StudName);
cqlsh:students> SELECT * FROM Students_Info WHERE StudName = 'Asha';

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2012-03-11 18:30:00.000000+0000 |              79.9 |     Asha

(1 rows)
cqlsh:students> SELECT Roll_No, StudName FROM Students_Info LIMIT 2;

 roll_no | studname
---------+----------
       5 |    Rohan
       1 |     Asha

(2 rows)
cqlsh:students> SELECT Roll_No AS USN FROM Students_Info;

 usn
-----
   5
   1
   2
   4
   3

(5 rows)
cqlsh:students> UPDATE Students_Info
            ... SET StudName = 'David Sheen'
            ... WHERE Roll_No = 2;
cqlsh:students> UPDATE Students_Info SET Roll_No = 6 WHERE Roll_No = 3; -- ❌ ERROR!
InvalidRequest: Error from server: code=2200 [Invalid query] message="PRIMARY KEY part roll_no found in SET part"
```

LAB 05: HDFS

OUTPUT:

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd ./Desktop/
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]
Starting resourcemanager
Starting nodemanagers
```

```
command [genericOptions] [commandOptions]

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -mkdir /Lab6
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Hadoop
ls: `/Hadoop': No such file or directory
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Lab6
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ touch test.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ nano text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -put ./text.txt /Lab6/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Lab6
Found 1 items
-rw-r--r--   1 hadoop supergroup         89 2025-04-15 14:23 /Lab6/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -cat /Lab6/text.txt
hi how are you
how is your job
how is your family
how is your brother
how is your sister
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Lab6
Found 1 items
-rw-r--r--   1 hadoop supergroup         89 2025-04-15 14:23 /Lab6/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ nano test.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -put ./test.txt /Lab6/test.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Lab6
Found 2 items
-rw-r--r--   1 hadoop supergroup         34 2025-04-15 14:26 /Lab6/test.txt
-rw-r--r--   1 hadoop supergroup         89 2025-04-15 14:23 /Lab6/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -getmerge /Lab6 /text.txt /Lab6 /test.txt ../Downloads/Merged.txt
getmerge: `/text.txt': No such file or directory
getmerge: `/test.txt': No such file or directory
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -getmerge /Lab6/text.txt /Lab6/test.txt ../Downloads/Merged.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -getfacl /Lab6
# file: /Lab6
# owner: hadoop
# group: supergroup
user::rwx
group::r-x
other::r-x
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -copyToLocal /Lab6/text.text ../Documents
copyToLocal: `/Lab6/text.text': No such file or directory
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -copyToLocal /Lab6/text.txt ../Documents
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -copyToLocal /Lab6/test.txt ../Documents
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -cat /Lab6/text.txt
hi how are you
how is your job
how is your family
how is your brother
how is your sister
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -mv /Lab6 /test_Lab6
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -ls /test_Lab6
Found 2 items
-rw-r--r--   1 hadoop supergroup         34 2025-04-15 14:26 /test_Lab6/test.txt
-rw-r--r--   1 hadoop supergroup         89 2025-04-15 14:23 /test_Lab6/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -cp /test_Lab6/ /Lab6
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -ls /Lab6
Found 2 items
-rw-r--r--   1 hadoop supergroup         34 2025-04-15 14:31 /Lab6/test.txt
-rw-r--r--   1 hadoop supergroup         89 2025-04-15 14:31 /Lab6/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -ls /test_Lab6
Found 2 items
-rw-r--r--   1 hadoop supergroup         34 2025-04-15 14:26 /test_Lab6/test.txt
-rw-r--r--   1 hadoop supergroup         89 2025-04-15 14:23 /test_Lab6/text.txt
```

LAB 06:WORDCOUNT PROBLEM(HADOOP)

CODE:

#driver.java

```
import java.io.IOException;
import org.apache.hadoop.conf.Configured;
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.util.Tool;
import org.apache.hadoop.util.ToolRunner;
public class WCDriver extends Configured implements Tool {
public int run(String args[]) throws IOException
{
if (args.length < 2)
{
System.out.println("Please give valid inputs");
return -1;
}
JobConf conf = new JobConf(WCDriver.class);
FileInputFormat.setInputPaths(conf, new Path(args[0]));
FileOutputFormat.setOutputPath(conf, new Path(args[1]));
conf.setMapperClass(WCMapper.class);
```

```
conf.setReducerClass(WCReducer.class);
conf.setMapOutputKeyClass(Text.class);
conf.setMapOutputValueClass(IntWritable.class);
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
JobClient.runJob(conf);
return 0;
}
public static void main(String args[]) throws Exception
{
int exitCode = ToolRunner.run(new WCDriver(), args);
System.out.println(exitCode);
}
}
```

#mapper.java

```
import java.io.IOException;
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 WCMapper extends MapReduceBase implements Mapper<LongWritable,Text, Text,
IntWritable> {
public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output,
Reporter rep)
```

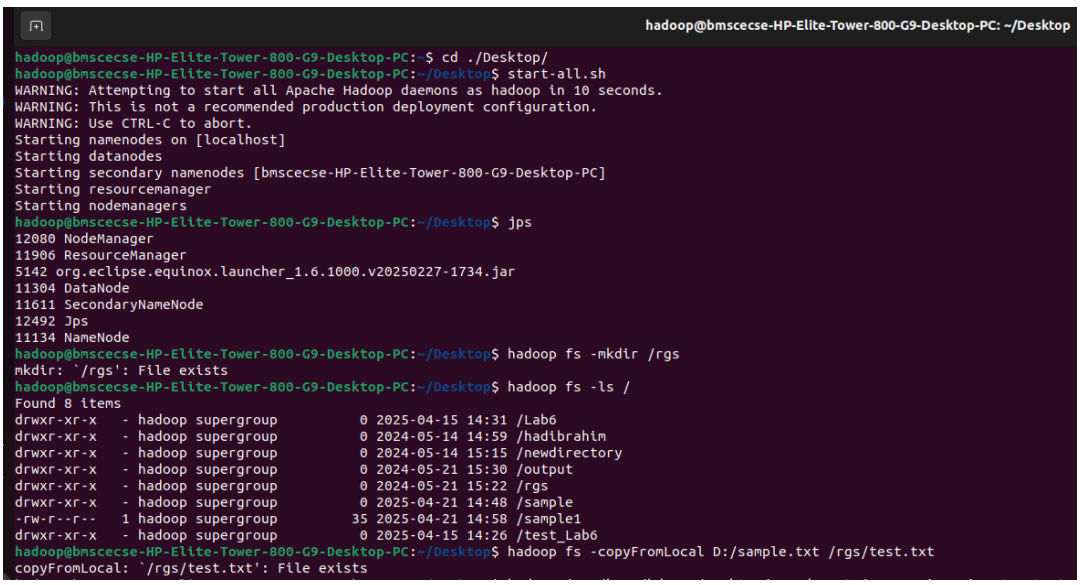
```
throws IOException
{
String line = value.toString();
for (String word : line.split(" "))
{
if (word.length() > 0)
{
output.collect(new Text(word), new IntWritable(1));
} } } }
```

#reducer.java

```
// Importing libraries
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 WCReducer extends MapReduceBase implements Reducer<Text,IntWritable, Text,
IntWritable> {
// Reduce function
public void reduce(Text key, Iterator<IntWritable> value,
OutputCollector<Text, IntWritable> output,
Reporter rep) throws IOException
{
int count = 0;
```

```
// Counting the frequency of each words
while (value.hasNext())
{
IntWritable i = value.next();
count += i.get();
}
output.collect(key, new IntWritable(count));
```

OUTPUT:



```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC: ~/Desktop
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd ./Desktop/
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]
Starting resourcemanager
Starting nodemanagers
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ jps
12080 NodeManager
11906 ResourceManager
5142 org.eclipse.equinox.launcher_1.6.1000.v20250227-1734.jar
11304 DataNode
11611 SecondaryNameNode
12492 Jps
11134 NameNode
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -mkdir /rgs
mkdir: `/rgs': File exists
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /
Found 8 items
drwxr-xr-x   - hadoop supergroup          0 2025-04-15 14:31 /Lab6
drwxr-xr-x   - hadoop supergroup          0 2024-05-14 14:59 /hadibrahim
drwxr-xr-x   - hadoop supergroup          0 2024-05-14 15:15 /newdirectory
drwxr-xr-x   - hadoop supergroup          0 2024-05-21 15:30 /output
drwxr-xr-x   - hadoop supergroup          0 2024-05-21 15:22 /rgs
drwxr-xr-x   - hadoop supergroup          0 2025-04-21 14:48 /sample
-rw-r--r--   1 hadoop supergroup         35 2025-04-21 14:58 /sample1
drwxr-xr-x   - hadoop supergroup          0 2025-04-15 14:26 /test_Lab6
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -copyFromLocal D:/sample.txt /rgs/test.txt
copyFromLocal: `/rgs/test.txt': File exists
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop jar /home/hadoop/eclipse-workspace/WordCount.jar WCDriver input output
2025-05-06 14:45:31,601 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-06 14:45:31,636 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-06 14:45:31,636 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-05-06 14:45:31,695 WARN mapreduce.JobResourceUploader: No job jar file set.  User classes may not be found. See Job or Job#setJar(String).
2025-05-06 14:45:31,738 INFO input.FileInputFormat: Total input files to process : 1
2025-05-06 14:45:31,765 INFO mapreduce.JobSubmitter: number of splits:1
2025-05-06 14:45:31,825 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1385456850_0001
2025-05-06 14:45:31,825 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-05-06 14:45:31,887 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-05-06 14:45:31,887 INFO mapreduce.Job: Running job: job_local1385456850_0001
2025-05-06 14:45:31,888 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-05-06 14:45:31,892 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2025-05-06 14:45:31,893 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-06 14:45:31,893 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-06 14:45:31,893 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-05-06 14:45:31,925 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-05-06 14:45:31,925 INFO mapred.LocalJobRunner: Starting task: attempt_local1385456850_0001_m_000000_0
2025-05-06 14:45:31,935 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2025-05-06 14:45:31,935 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-06 14:45:31,935 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-06 14:45:31,943 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2025-05-06 14:45:31,945 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/user/hadoop/input/sample.txt:0+90
2025-05-06 14:45:31,978 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2025-05-06 14:45:31,978 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2025-05-06 14:45:31,978 INFO mapred.MapTask: soft limit at 83886080
2025-05-06 14:45:31,979 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2025-05-06 14:45:31,979 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2025-05-06 14:45:31,980 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2025-05-06 14:45:32,049 INFO mapred.LocalJobRunner:
2025-05-06 14:45:32,050 INFO mapred.MapTask: Starting flush of map output
2025-05-06 14:45:32,050 INFO mapred.MapTask: Spilling map output
2025-05-06 14:45:32,050 INFO mapred.MapTask: bufstart = 0; bufend = 174; bufvoid = 104857600
2025-05-06 14:45:32,050 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26214316(104857264); length = 81/6553600
2025-05-06 14:45:32,052 INFO mapred.MapTask: Finished spill 0
2025-05-06 14:45:32,057 INFO mapred.Task: Task:attempt_local1385456850_0001_m_000000_0 is done. And is in the process of committing
2025-05-06 14:45:32,059 INFO mapred.LocalJobRunner: map
2025-05-06 14:45:32,059 INFO mapred.Task: Task 'attempt_local1385456850_0001_m_000000_0' done.
2025-05-06 14:45:32,061 INFO mapred.Task: Final Counters for attempt_local1385456850_0001_m_000000_0: Counters: 23
        File System Counters
                FILE: Number of bytes read=201
                FILE: Number of bytes written=641533
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=90
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=5
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=1
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=6
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /output/
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2024-05-21 15:30 /output/_SUCCESS
-rw-r--r--   1 hadoop supergroup         69 2024-05-21 15:30 /output/part-00000
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -cat /output/part-00000
are     1
brother 1
family  1
hi      1
how     5
is      4
job     1
sister  1
you     1
your    4
```

LAB 07:WEATHER DATA(HADOOP)

CODE:

#AvgDriver.java

```
package temp;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class AverageDriver {
public static void main(String[] args) throws Exception {
if (args.length != 2) {
System.err.println("Please Enter the input and output parameters");
System.exit(-1);
}
Job job = new Job();
job.setJarByClass(AverageDriver.class);
job.setJobName("Max temperature");
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
job.setMapperClass(AverageMapper.class);
job.setReducerClass(AverageReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
System.exit(job.waitForCompletion(true) ? 0 : 1);
```

```
}
}
```

#AvgMapper.java

```
package temp;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class AverageMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
public static final int MISSING = 9999;
public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
int temperature;
String line = value.toString();
String year = line.substring(15, 19);
if (line.charAt(87) == '+') {
  temperature = Integer.parseInt(line.substring(88, 92));
} else {
temperature = Integer.parseInt(line.substring(87, 92));
}
String quality = line.substring(92, 93);
if (temperature != 9999 && quality.matches("[01459]"))
context.write(new Text(year), new IntWritable(temperature));
}
}
```

#AvgReducer.java

```
package temp;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class AverageReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable,
Text, IntWritable>.Context context) throws IOException, InterruptedException {
int max_temp = 0;
int count = 0;
for (IntWritable value : values) {
max_temp += value.get();
count++;
}
context.write(key, new IntWritable(max_temp / count));
}}
```

#MeanMaxDriver.java

```
package meanmax;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class MeanMaxDriver {
public static void main(String[] args) throws Exception {
if (args.length != 2) {
```

```
System.err.println("Please Enter the input and output parameters");
System.exit(-1);
}
Job job = new Job();
job.setJarByClass(MeanMaxDriver.class);
job.setJobName("Max temperature");
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
job.setMapperClass(MeanMaxMapper.class);
job.setReducerClass(MeanMaxReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}
```

#MeanMaxMapper.java

```
package meanmax;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class MeanMaxMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
public static final int MISSING = 9999;
public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
int temperature;
```

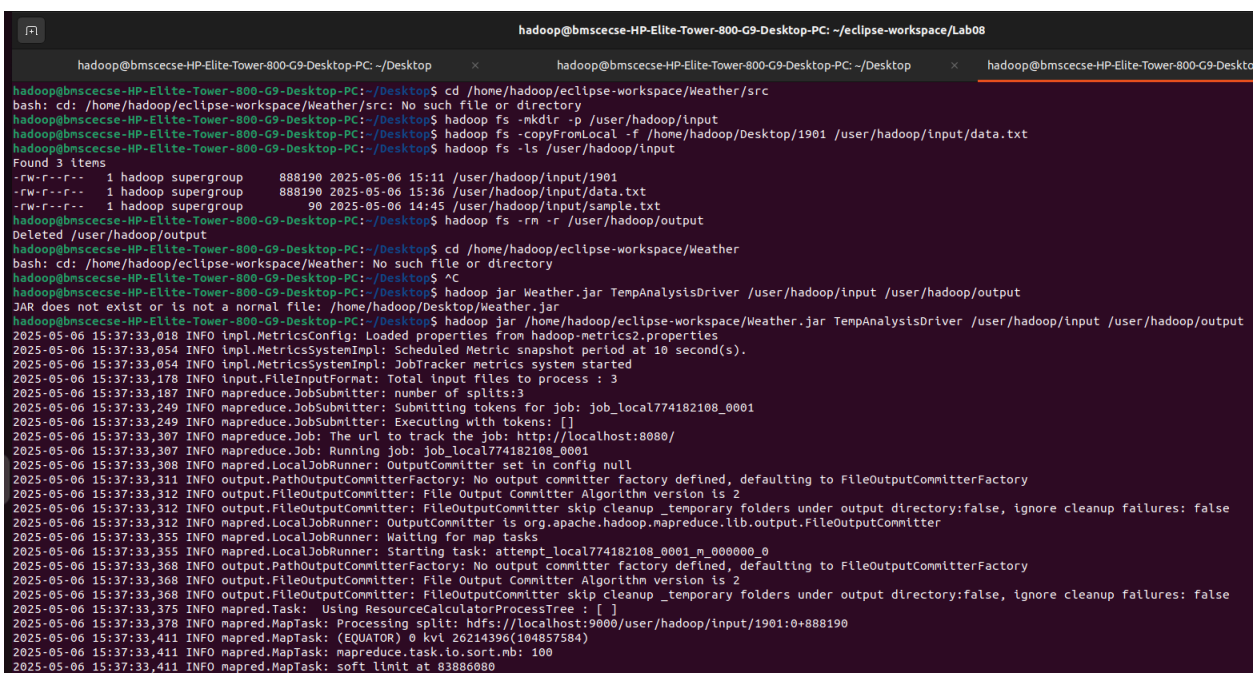
```
String line = value.toString();
String month = line.substring(19, 21);
if (line.charAt(87) == '+') {
temperature = Integer.parseInt(line.substring(88, 92));
} else {
temperature = Integer.parseInt(line.substring(87, 92));
  }
String quality = line.substring(92, 93);
if (temperature != 9999 && quality.matches("[01459]"))
context.write(new Text(month), new IntWritable(temperature));
}
}
```

#MeanMaxReducer.java

```
package meanmax;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class MeanMaxReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable,
Text, IntWritable>.Context context) throws IOException, InterruptedException {
int max_temp = 0;
int total_temp = 0;
int count = 0;
int days = 0;
for (IntWritable value : values) {
int temp = value.get();
```

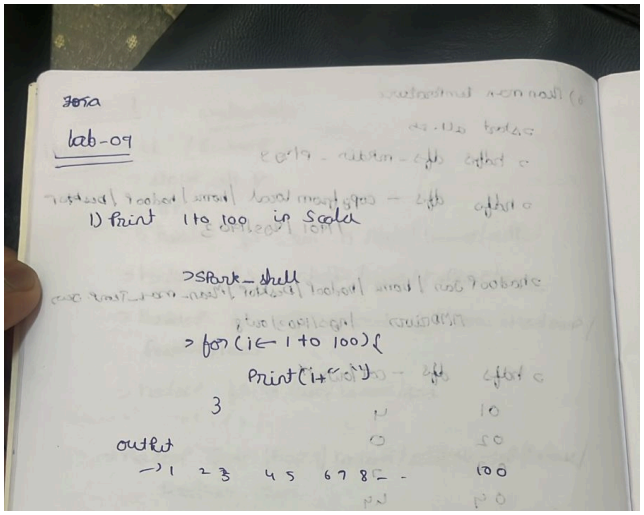
```
if (temp > max_temp)
max_temp = temp;
count++;
if (count == 3) {
total_temp += max_temp;
max_temp = 0;
count = 0;
days++;
}
}
context.write(key, new IntWritable(total_temp / days));
}
}
```

OUTPUT:

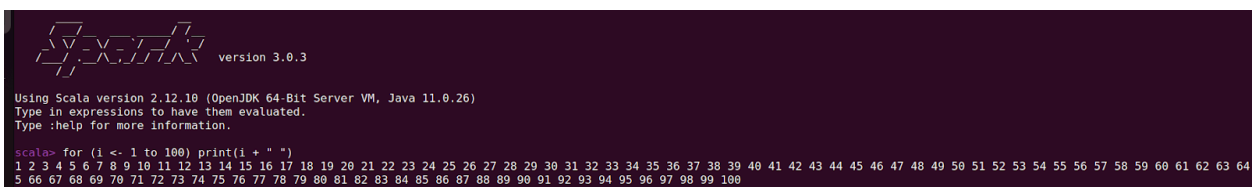


```
                Merged Map outputs=0
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=1052770304
        File Input Format Counters
                Bytes Read=1776380
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /user/hadoop/output
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ cd ~/eclipse-workspace/Weather/src
hadoop com.sun.tools.javac.Main *.java
jar cf Weather.jar *.class
mv Weather.jar ..
bash: cd: /home/hadoop/eclipse-workspace/Weather/src: No such file or directory
error: file not found: *.java
Usage: javac <options> <source files>
use --help for a list of possible options
*.class : no such file or directory
mv: cannot stat 'Weather.jar': No such file or directory
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ cd ~/eclipse-workspace/Lab08/src
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/eclipse-workspace/Lab08/src$ hadoop com.sun.tools.javac.Main *.java
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/eclipse-workspace/Lab08/src$ jar cf Weather.jar *.class
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/eclipse-workspace/Lab08/src$ mv Weather.jar ..
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/eclipse-workspace/Lab08/src$ hadoop fs -mkdir -p /user/hadoop/aaa
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/eclipse-workspace/Lab08/src$ hadoop fs -copyFromLocal -f /home/hadoop/Desktop/1901 /user/hadoop/aaa/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/eclipse-workspace/Lab08/src$ hadoop fs -rm -r /user/hadoop/aaa/output
Deleted /user/hadoop/aaa/output
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/eclipse-workspace/Lab08/src$ cd ~/eclipse-workspace/Lab08
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/eclipse-workspace/Lab08$ hadoop jar Weather.jar TempAnalysisDriver /user/hadoop/aaa/text.txt /user/hadoop/aaa/output
2025-05-06 15:42:33,864 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-06 15:42:33,903 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-06 15:42:33,903 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-05-06 15:42:34,007 INFO input.FileInputFormat: Total input files to process : 1
2025-05-06 15:42:34,031 INFO mapreduce.JobSubmitter: number of splits:1
2025-05-06 15:42:34,097 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1383725974_0001
2025-05-06 15:42:34,098 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-05-06 15:42:34,152 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-05-06 15:42:34,153 INFO mapreduce.Job: Running job: job_local1383725974_0001
2025-05-06 15:42:34,153 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-05-06 15:42:34,158 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2025-05-06 15:42:34,158 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-06 15:42:34,158 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-06 15:42:34,159 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-05-06 15:42:34,206 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-05-06 15:42:34,207 INFO mapred.LocalJobRunner: Starting task: attempt_local1383725974_0001_m_000000_0
2025-05-06 15:42:34,220 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2025-05-06 15:42:34,220 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-06 15:42:34,220 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-06 15:42:34,227 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
```

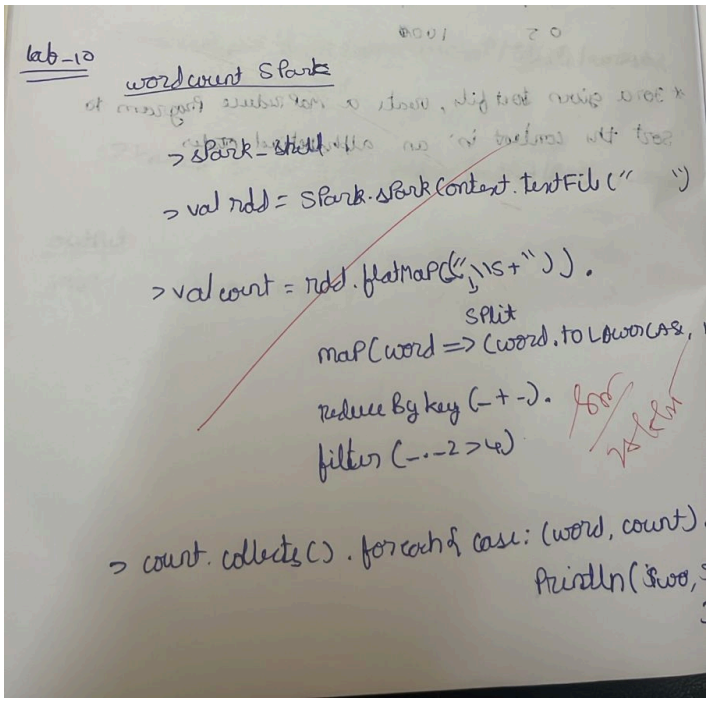
LAB 08:SCALA(PRINTING THE NUMBER)



OUTPUT:



LAB 09:SCALA SPARK(RDD AND FLATMAP)



OUTPUT:

```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ spark-shell
25/05/20 11:28:13 WARN Utils: Your hostname, bmscecse-HP-Elite-Tower-600-G9-Desktop-PC resolves to a loopback address: 127.0.1.1; using 10.124.3.80 instead (on interface eno1)
25/05/20 11:28:13 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/opt/spark/jars/spark-unsafe_2.12-3.0.3.jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
25/05/20 11:28:13 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://10.124.3.80:4040
Spark context available as 'sc' (master = local[*], app id = local-1747720695950).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.0.3
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.26)
Type in expressions to have them evaluated.
Type :help for more information.

scala> for (i <- 1 to 100) print(i + " ")
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 6
5 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
scala> val rdd = spark.sparkContext.textFile("file:/home/bmscecse/Desktop/scala")
rdd: org.apache.spark.rdd.RDD[String] = file:/home/bmscecse/Desktop/scala MapPartitionsRDD[1] at textFile at <console>:23

scala> val counts = rdd.flatMap(_.split("\\s+")).map(word => (word.toLowerCase, 1)).reduceByKey(_ + _).filter(_._2 > 4)
counts: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[5] at filter at <console>:25

scala> counts.collect().foreach{ case (word, count) => println(s"$word $count") }
spark 6
```

```
scala> val rdd = spark.sparkContext.textFile("file:/home/bmscecse/Desktop/scala")
rdd: org.apache.spark.rdd.RDD[String] = file:/home/bmscecse/Desktop/scala MapPartitionsRDD[1] at textFile at <console>:23

scala> val counts = rdd.flatMap(_.split("\\s+")).map(word => (word.toLowerCase, 1)).reduceByKey(_ + _).filter(_._2 > 4)
counts: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[5] at filter at <console>:25

scala> counts.collect().foreach{ case (word, count) => println(s"$word $count") }
spark 6

scala>
```

LAB 10:

Write a simple streaming program in Spark to receive text data streams on a particular port, perform basic text cleaning (like white space removal, stop words removal, lemmatization, etc.), and print the cleaned text on the screen. (Open Ended Question).

```
# Install NLTK and download required data (run once)

!pip install nltk


import nltk

nltk.download('punkt')

nltk.download('stopwords')

nltk.download('wordnet')


from pyspark.sql import SparkSession

from pyspark.sql.functions import col, lower, regexp_replace, split, explode, udf

from pyspark.sql.types import ArrayType, StringType

from pyspark.ml.feature import StopWordsRemover

from nltk.stem import WordNetLemmatizer


# Initialize SparkSession

spark = SparkSession.builder.appName("TextProcessing").getOrCreate()


# Define your input lines
```

```
lines = [
    "Hello, I hate you.",
    "I hate that I love you.",
    "Don't want to, but I can't put",
    "nobody else above you."
]

# Create DataFrame from lines
df = spark.createDataFrame(lines, "string").toDF("value")

# Step 1: Lowercase and remove punctuation
df_clean = df.select(regexp_replace(lower(col("value")), "[^a-zA-Z\\s]", "").alias("cleaned"))

# Step 2: Tokenize the cleaned text
df_tokens = df_clean.select(split(col("cleaned"), "\\s+").alias("tokens"))

# Step 3: Remove stop words
remover = StopWordsRemover(inputCol="tokens", outputCol="filtered")
df_filtered = remover.transform(df_tokens)

# Step 4: Lemmatization using NLTK WordNetLemmatizer with UDF
lemmatizer = WordNetLemmatizer()
```

```
def lemmatize_words(words):
    return [lemmatizer.lemmatize(word) for word in words]

lemmatize_udf = udf(lemmatize_words, ArrayType(StringType()))

df_lemmatized = df_filtered.withColumn("lemmatized", lemmatize_udf(col("filtered")))

# Step 5: Explode the lemmatized words and show results
df_lemmatized.select(explode(col("lemmatized")).alias("word")).show(truncate=False)
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.2.0)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.5.0)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.11/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
+------+
|word  |
+------+
|hello |
|hate  |
|hate  |
|love  |
|dont  |
|want  |
|cant  |
|put   |
|nobody|
|else  |
+------+
```