

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT on

## Big Data Analytics (23CS6PCBDA)

*Submitted by:*

**Mohith Jain (1BM22CS162)**

**Under the Guidance of  
Amruta B  
Assistant Professor, BMSCE**

*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**

**March 2025 - June 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**” carried out by **Mohith Jain (1BM22CS162)**, who is 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 **Big Data Analytics – (23CS6PCBDA)** work prescribed for the said degree.

**Amruta B**  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

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

## Table Of Contents

<b>Sl.no</b>	<b>Program details</b>	<b>Pg no</b>
<b>1</b>	<b>MongoDB- CRUD Operations Demonstration</b>	<b>1</b>
<b>2</b>	<b>Student database and import-export files</b>	<b>2</b>
<b>3</b>	<b>Working with Cassandra in ubuntu terminal</b>	<b>3</b>
<b>4</b>	<b>Perform the following DB operations using Cassandra.</b>	<b>5</b>
<b>5</b>	<b>Execution of HDFS Commands for interaction with Hadoop Environment. (Minimum 10 commands to be executed)</b>	<b>7</b>
<b>6</b>	<b>Implement Wordcount program on Hadoop framework</b>	<b>8</b>
<b>7</b>	<b>Create a MapReduce program to find average temperature for each year from NCDC data set. b) find the mean max temperature for every month.</b>	<b>9</b>
<b>8</b>	<b>Write a Scala program to print numbers from 1 to 100 using for loop.</b>	<b>11</b>
<b>9</b>	<b>Using RDD and FlatMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark.</b>	<b>12</b>

## LAB 1

### MongoDb CRUD operations

```
Atlas atlas-z7tbbe-shard-0 [primary] test> use myDB
switched to db myDB
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db
myDB
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.createCollection("Student")
{ ok: 1 }
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.createCollection("Faculty")
{ ok: 1 }
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Faculty.drop()
true
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.insert({_id:1,StudName:"Jonathan",Grade:"VI",Hobbies:"Browsing"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{ acknowledged: true, insertedIds: { '0': 1 } }
Atlas atlas-z7tbbe-shard-0 [primary] myDB> var mystudent = [{_id:4,StudName:"Saurav",Grade:"V",Hobbies:"Dance"},{_id:5,StudName:"Kumar",Grade:"VII",Hobbies:"Singing"}]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.insert(mystudent)
{ acknowledged: true, insertedIds: { '0': 4, '1': 5 } }
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.update([{$set:{StudName:"Gaurav"}}],{$set:{StudName:"Gaurav"}});
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find()
[
  { _id: 1, StudName: 'Gaurav', Grade: 'VI', Hobbies: 'Browsing' },
  { _id: 4, StudName: 'Saurav', Grade: 'V', Hobbies: 'Dance' },
  { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' }
]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find({StudName:"Kumar"})
[ { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' } ]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find([{$eq:{Grade:"VII"}}]);
[ { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' } ]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find([{$eq:{Grade:"VII"}}]);
[ { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' } ]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.count();
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
3
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find().sort({StudName:1});
[
  { _id: 1, StudName: 'Gaurav', Grade: 'VI', Hobbies: 'Browsing' },
  { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' },
  { _id: 4, StudName: 'Saurav', Grade: 'V', Hobbies: 'Dance' }
]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find().sort({Grade:1});
[
  { _id: 4, StudName: 'Saurav', Grade: 'V', Hobbies: 'Dance' },
  { _id: 1, StudName: 'Gaurav', Grade: 'VI', Hobbies: 'Browsing' },
  { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' }
]
```

```
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find()
[
  { _id: 1, StudName: 'Gaurav', Grade: 'VI', Hobbies: 'Browsing' },
  { _id: 4, StudName: 'Saurav', Grade: 'V', Hobbies: 'Dance' },
  { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' }
]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find({StudName:"Kumar"})
[ { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' } ]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find([{$eq:{Grade:"VII"}}]);
[ { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' } ]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find([{$eq:{Grade:"VII"}}]);
[ { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' } ]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.count();
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
3
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find().sort({StudName:1});
[
  { _id: 1, StudName: 'Gaurav', Grade: 'VI', Hobbies: 'Browsing' },
  { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' },
  { _id: 4, StudName: 'Saurav', Grade: 'V', Hobbies: 'Dance' }
]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find().sort({Grade:1});
[
  { _id: 4, StudName: 'Saurav', Grade: 'V', Hobbies: 'Dance' },
  { _id: 1, StudName: 'Gaurav', Grade: 'VI', Hobbies: 'Browsing' },
  { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' }
]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find().sort({Grade:-1});
[
  { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' },
  { _id: 1, StudName: 'Gaurav', Grade: 'VI', Hobbies: 'Browsing' },
  { _id: 4, StudName: 'Saurav', Grade: 'V', Hobbies: 'Dance' }
]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.insert({_id:2,StudName:"Aryan",Grade:"IV",Hobbies:"Sketching"});
{ acknowledged: true, insertedIds: { '0': 2 } }
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find()
[
  { _id: 1, StudName: 'Gaurav', Grade: 'VI', Hobbies: 'Browsing' },
  { _id: 4, StudName: 'Saurav', Grade: 'V', Hobbies: 'Dance' },
  { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' },
  { _id: 2, StudName: 'Aryan', Grade: 'IV', Hobbies: 'Sketching' }
]
Atlas atlas-z7tbbe-shard-0 [primary] myDB> db.Student.find().sort({Grade:1});
[
  { _id: 2, StudName: 'Aryan', Grade: 'IV', Hobbies: 'Sketching' },
  { _id: 4, StudName: 'Saurav', Grade: 'V', Hobbies: 'Dance' },
  { _id: 1, StudName: 'Gaurav', Grade: 'VI', Hobbies: 'Browsing' },
  { _id: 5, StudName: 'Kumar', Grade: 'VII', Hobbies: 'Singing' }
]
```

## LAB 2

### Student database

```
myDB> db.students.drop();
true
myDB> show dbs
admin      40.00 KiB
config     72.00 KiB
local      120.00 KiB
students   72.00 KiB
myDB> db.createCollection("Student");
{ ok: 1 }
myDB> show dbs
admin      40.00 KiB
config     72.00 KiB
local      120.00 KiB
myDB       8.00 KiB
students   72.00 KiB
myDB> db.Student.insert({_id:1,StudeName:"MichelleJacintha",Grade:"VII",Hobbies:"InternetSurfing",});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{ acknowledged: true, insertedIds: { '0': 1 } }
myDB> db.Student.insertOne({_id:1,StudeName:"MichelleJacintha",Grade:"VII",Hobbies:"InternetSurfing",});
MongoServerError: E11000 duplicate key error collection: myDB.Student index: _id_dup key: { _id: 1 }
myDB> db.Student.insertOne({_id:2,StudeName:"MichelleJacintha",Grade:"VII",Hobbies:"InternetSurfing",});
{ acknowledged: true, insertedId: 2 }
myDB> db.Student.find();
[
  {
    _id: 1,
    StudeName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  {
    _id: 2,
    StudeName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  }
]
```

```
]
myDB> show collections;
Student
myDB> db.Student.updateOne({_id:2,StudeName:"AryanDavid",Grade:"VIII"},{$set:{Hobbies:"Chess"}},{upsert:true});
MongoServerError: E11000 duplicate key error collection: myDB.Student index: _id_dup key: { _id: 2 }
myDB> db.Student.find({StudeName:"AryanDavid"});
myDB> db.Student.find();
[
  {
    _id: 1,
    StudeName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  {
    _id: 2,
    StudeName: 'AryanDavid',
    Grade: 'VIII',
    Hobbies: 'Skating'
  }
]
myDB> db.Student.find({StudeName:"AryanDavid"});
[
  {
    _id: 2,
    StudeName: 'AryanDavid',
    Grade: 'VIII',
    Hobbies: 'Skating'
  }
]
myDB> db.Student.updateOne({_id:2,StudeName:"AryanDavid",Grade:"VIII"},{$set:{Hobbies:"Chess"}},{upsert:true});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
myDB> db.Student.find();
[
  {
    _id: 1,
    StudeName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  { _id: 2, StudeName: 'AryanDavid', Grade: 'VIII', Hobbies: 'Chess' }
]
myDB> db.Student.find({StudeName:"AryanDavid"});
[
  { _id: 2, StudeName: 'AryanDavid', Grade: 'VIII', Hobbies: 'Chess' }
]
```

```

myDB> db.Student.find({Grade:{Seq:'VII'}}).pretty();
[
  {
    _id: 1,
    StudeName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  }
]
myDB> db.Student.find({Hobbies:{$in:['Chess','Skating']}}).pretty();
[
  { _id: 2, StudeName: 'AryanDavid', Grade: 'VIII', Hobbies: 'Chess' }
]
myDB> db.Student.find({StudeName:/^M/}).pretty();
[
  {
    _id: 1,
    StudeName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  }
]
myDB> db.Student.find({StudeName:/e/}).pretty();
[
  {
    _id: 1,
    StudeName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  }
]
myDB> db.Student.find({StudeName:/d/}).pretty();
[
  { _id: 2, StudeName: 'AryanDavid', Grade: 'VIII', Hobbies: 'Chess' }
]
myDB> db.
... db.Student.count();
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
0
myDB> db.Student.count();
2
myDB> db.Student.find().sort({StudeName:-1}).pretty();
[
  {
    _id: 1,
    StudeName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  { _id: 2, StudeName: 'AryanDavid', Grade: 'VIII', Hobbies: 'Chess' }
]

```

## LAB 4

### Working with Cassandra in ubuntu terminal

```

cqlsh> create keyspace Students_avy with replication = {'class':'SimpleStrategy'
,'replication_factor':1};
cqlsh> describe keyspaces;

```

comp	education	students	system_auth	system_views
company	employee	students_avy	system_distributed	system_virtual_schema
company	javatpoint	studentss	system_schema	
company1	student	system	system_traces	



```

cqlsh:students_avy> 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,'Krian','2012-03-12',89.9)
... insert into students_info(Roll_No,StudName,DateOfJoining,last_exam_percent) values(3,'Tarun','2012-03-12',78.9)
... insert into students_info(Roll_No,StudName,DateOfJoining,last_exam_percent) values(4,'Samrth','2012-03-12',90.9)
... insert into students_info(Roll_No,StudName,DateOfJoining,last_exam_percent) values(5,'Smitha','2012-03-12',67.9)
... insert into students_info(Roll_No,StudName,DateOfJoining,last_exam_percent) values(6,'Rohan','2012-03-12',56.9)
... apply batch;
cqlsh:students_avy> select * from students_info;

```

roll_no	dateofjoining	last_exam_percent	studname
5	2012-03-11 18:30:00.000000+0000	67.9	Smitha
1	2012-03-11 18:30:00.000000+0000	79.9	Asha
2	2012-03-11 18:30:00.000000+0000	89.9	Krian
4	2012-03-11 18:30:00.000000+0000	90.9	Samrth
6	2012-03-11 18:30:00.000000+0000	56.9	Rohan
3	2012-03-11 18:30:00.000000+0000	78.9	Tarun

(6 rows)

```

cqlsh:students_avy> 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	Krian
3	2012-03-11 18:30:00.000000+0000	78.9	Tarun

(3 rows)

```

cqlsh:students_avy> select * from students_info where StudName = 'Asha';

```

InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and query despite the performance unpredictability, use ALLOW FILTERING"

```

cqlsh:students_avy> select Roll_No, StudName from students_info LIMIT 2;

```

roll_no	studname
5	Smitha
1	Asha

(2 rows)

```

cqlsh:students_avy> create index on students_info(StudName);

```

```

cqlsh:students_avy> 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_avy> select Roll_No as "USN" from students_info;

```

USN
5
1
2
4
6
3

(6 rows)

```

cqlsh:students_avy> update students_info set StudName='David Sheen' where Roll_No = 2;

```

```

cqlsh:students_avy> update students_info set Roll_No = 6 where Roll_No = 3;

```

InvalidRequest: Error from server: code=2200 [Invalid query] message="PRIMARY KEY part roll\_no found in SET part"

```

cqlsh:students_avy> select * from students_info;

```

roll_no	dateofjoining	last_exam_percent	studname
5	2012-03-11 18:30:00.000000+0000	67.9	Smitha
1	2012-03-11 18:30:00.000000+0000	79.9	Asha
2	2012-03-11 18:30:00.000000+0000	89.9	David Sheen
4	2012-03-11 18:30:00.000000+0000	90.9	Samrth
6	2012-03-11 18:30:00.000000+0000	56.9	Rohan
3	2012-03-11 18:30:00.000000+0000	78.9	Tarun

(6 rows)

```

cqlsh:students_avy> delete last_exam_percent from students_info where Roll_No = 2;

```

## LAB 5

### Performing DB operation using cassandra

```
Apr 8 14:43
bmscscse@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC: ~
bmscscse@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> DROP KEYSPACE employee;
... DROP KEYSPACE employee;
SyntaxException: line 2:0 mismatched input 'DROP' expecting EOF (DROP KEYSPACE employee[DROP]...)
cqlsh> DROP KEYSPACE employee;
cqlsh> DESCRIBE KEYSPACES;

employee1      students      system_distributed  system_views
student_data   system        system_schema       system_virtual_schema
student_new     system_auth   system_traces

cqlsh> CREATE KEYSPACE Employee
... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> DESCRIBE KEYSPACES;

employee      student_new  system_auth  system_traces
employee1     students    system_distributed  system_views
student_data  system      system_schema  system_virtual_schema

cqlsh> USE employee;
cqlsh:employee> CREATE TABLE Employee_Info (
...     Emp_Id int PRIMARY KEY,
...     Emp_Name text,
...     Designation text,
...     Date_of_Joining date,
...     Salary int,
...     Dept_Name text
... );
cqlsh:employee> BEGIN BATCH
... INSERT INTO Employee_Info(Emp_id,Emp_Name,Designation,Date_of_Joining,Salary,Dept_Name)VALUES (101,'John Doe','Developer','2020-01-15',60000,'IT');
... INSERT INTO Employee_Info(Emp_id,Emp_Name,Designation,Date_of_Joining,Salary,Dept_Name)VALUES (121,'Jane Smith','Manager','2019-03-10',80000,'HR');
... INSERT INTO Employee_Info(Emp_id,Emp_Name,Designation,Date_of_Joining,Salary,Dept_Name)VALUES (131,'Mike Johnson','Analyst','2021-06-20',55000,'Finance');
... APPLY BATCH;
cqlsh:employee> SELECT * FROM Employee_Info;\
Invalid syntax at char 29
SELECT * FROM Employee_Info;\
^
cqlsh:employee> SELECT * FROM Employee_Info;

emp_id | date_of_joining | dept_name | designation | emp_name | salary
-----+-----+-----+-----+-----+-----
121 | 2019-03-10 | HR | Manager | Jane Smith | 80000
131 | 2021-06-20 | Finance | Analyst | Mike Johnson | 55000
101 | 2020-01-15 | IT | Developer | John Doe | 60000
```



```

Apr 8 14:45
bmscece@bmscece-HP-Elite-Tower-800-G9-Desktop-PC: ~

icted by an EQ or an IN."
cqlsh:employee> SELECT * FROM Employee_Info WHERE Emp_Id IN (101, 121, 131) ORDER BY Salary ASC;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Order by is currently only supported on the clustered columns of the PRIMARY KEY, got salary"
cqlsh:employee> SELECT * FROM Employee_Info WHERE Emp_Id = 101 ORDER BY Salary ASC;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Order by is currently only supported on the clustered columns of the PRIMARY KEY, got salary"
cqlsh:employee> ALTER TABLE Employee_Info ADD Projects set<text>;
cqlsh:employee> UPDATE Employee_Info SET Projects = {'Project A','Project B'} WHERE Emp_Id = 101;
cqlsh:employee> UPDATE Employee_Info SET Projects = {'Project X','Project Y'} WHERE Emp_Id = 121;
cqlsh:employee> UPDATE Employee_Info SET Projects = {'Project Z'} WHERE Emp_Id = 131;
cqlsh:employee> SELECT * from Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----+-----+-----+-----+-----+-----+-----
 121    | 2019-03-10     | IT        | Manager     | Jane Doe | {'Project X', 'Project Y'} | 80000
 131    | 2021-06-20     | Finance   | Analyst     | Mike Johnson | {'Project Z'} | 55000
 101    | 2020-01-15     | IT        | Developer   | John Doe | {'Project A', 'Project B'} | 60000
(3 rows)
cqlsh:employee> INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name, Projects)
... VALUES (101, 'John Doe', 'Developer', '2020-01-15', 60000, 'IT', {'Project A', 'Project B'})
... USING TTL 15;
cqlsh:employee>
cqlsh:employee> INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name, Projects) VALUES (
101, 'John Doe', 'Developer', '2020-01-15', 60000, 'IT', {'Project A', 'Project B'}) USING TTL 15;
cqlsh:employee> SELECT * from Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----+-----+-----+-----+-----+-----+-----
 121    | 2019-03-10     | IT        | Manager     | Jane Doe | {'Project X', 'Project Y'} | 80000
 131    | 2021-06-20     | Finance   | Analyst     | Mike Johnson | {'Project Z'} | 55000
 101    | 2020-01-15     | IT        | Developer   | John Doe | {'Project A', 'Project B'} | 60000
(3 rows)
cqlsh:employee> INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name, Projects) VALUES (
141, 'John Doe', 'Developer', '2020-01-15', 60000, 'IT', {'Project A', 'Project B'}) USING TTL 15;
cqlsh:employee> SELECT * from Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----+-----+-----+-----+-----+-----+-----
 121    | 2019-03-10     | IT        | Manager     | Jane Doe | {'Project X', 'Project Y'} | 80000
 141    | 2020-01-15     | IT        | Developer   | John Doe | {'Project A', 'Project B'} | 60000
 131    | 2021-06-20     | Finance   | Analyst     | Mike Johnson | {'Project Z'} | 55000
(3 rows)
cqlsh:employee> SELECT * from Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----+-----+-----+-----+-----+-----+-----
 121    | 2019-03-10     | IT        | Manager     | Jane Doe | {'Project X', 'Project Y'} | 80000
 131    | 2021-06-20     | Finance   | Analyst     | Mike Johnson | {'Project Z'} | 55000
(2 rows)
cqlsh:employee> 

```

## LAB 6

### Hadoop basic command execution in Ubuntu

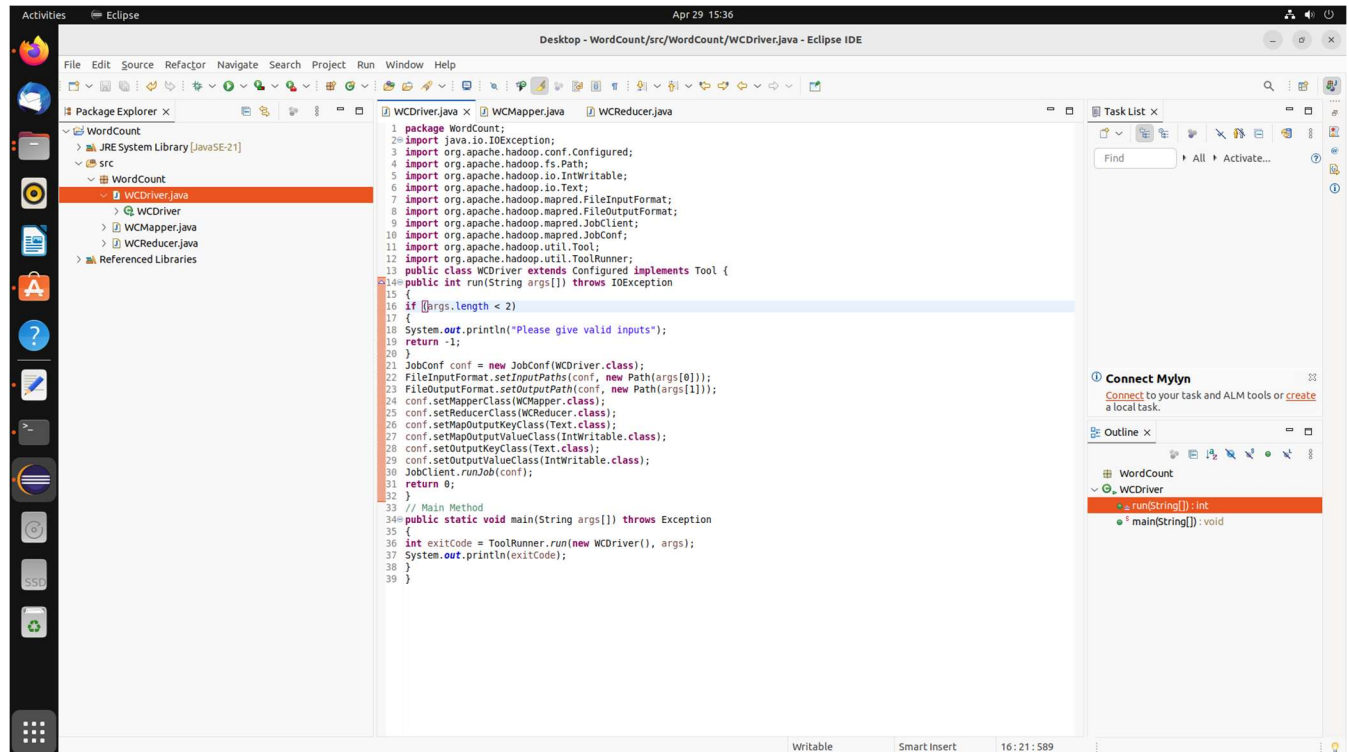
```
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
localhost: namenode is running as process 6418. Stop it first and ensure /tmp/hadoop-hadoop-namenode.pid file is empty before retry.
Starting datanodes
localhost: datanode is running as process 6591. Stop it first and ensure /tmp/hadoop-hadoop-datanode.pid file is empty before retry.
Starting secondary namenodes [bmscscse-HP-Elite-Tower-800-G9-Desktop-PC]
bmscscse-HP-Elite-Tower-800-G9-Desktop-PC: secondarynamenode is running as process 6870. Stop it first and ensure /tmp/hadoop-hadoop-secondarynamenode.
file is empty before retry.
Starting resourcemanager
resourcemanager is running as process 7157. Stop it first and ensure /tmp/hadoop-hadoop-resourcemanager.pid file is empty before retry.
Starting nodemanagers
localhost: nodemanager is running as process 7319. Stop it first and ensure /tmp/hadoop-hadoop-nodemanager.pid file is empty before retry.
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ jps
6418 NameNode
7157 ResourceManager
6870 SecondaryNameNode
7319 NodeManager
12717 Jps
6591 DataNode
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /
ls: Found 1 items
drwxr-xr-x 1 hadoop supergroup 0 2025-04-15 14:32 /rgs
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ nano /home/hadoop/Desktop/file1.txt
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -copyFromLocal /home/hadoop/Desktop/file1.txt /rgs/test.txt
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /rgs
ls: Found 1 items
-rw-r--r-- 1 hadoop supergroup 89 2025-04-15 15:03 /rgs/test.txt
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /rgs
ls: Found 1 items
-rw-r--r-- 1 hadoop supergroup 89 2025-04-15 15:03 /rgs/test.txt
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop jar /home/hadoop/Desktop/WordCount.jar wordcount.WordCount /rgs/test.txt /output
JAR does not exist or is not a normal file: /home/hadoop/Desktop/WordCount.jar
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /Hadoop
ls: '/Hadoop': No such file or directory
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ ls
ls: command not found
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ ls
'ACFrGcUjh3SUnwpbNA9r5ZGDw7_C_49d0lfoe1dplqetFZ8Ghfx-ugoCx6lMnJLRY-IUXyqoEhxLEtB1L483dCRqjzgfQvp5XHT-eAlnKKSXNbdZsV2Xfer7ow5S5wb8eY_3vAbH2TL9KwBokfZ2e
Sd9boAfH5Vv2vjw==.pdf'
Desktop
Documents
Downloads
eclipse-workspace
hadoop
hadoop-3.3.6.tar.gz
hadoopdata
hs_err_pid5585.log
hs_err_pid8027.log
hs_err_pid8572.log
Music
Pictures
```

```
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -mkdir /abc
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /Hadoop
ls: '/Hadoop': No such file or directory
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ ls
ls: command not found
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ ls
'ACFrGcUjh3SUnwpbNA9r5ZGDw7_C_49d0lfoe1dplqetFZ8Ghfx-ugoCx6lMnJLRY-IUXyqoEhxLEtB1L483dCRqjzgfQvp5XHT-eAlnKKSXNbdZsV2Xfer7ow5S5wb8eY_3vAbH2TL9KwBokfZ2e
Sd9boAfH5Vv2vjw==.pdf'
Desktop
Documents
Downloads
eclipse-workspace
hadoop
hadoop-3.3.6.tar.gz
hadoopdata
hs_err_pid5585.log
hs_err_pid8027.log
hs_err_pid8572.log
Music
Pictures
Public
snap
Templates
Videos
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /hadoop
ls: '/hadoop': No such file or directory
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -mkdir /hadoop
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /hadoop
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -put /home/hadoop/Desktop/Welcome.txt /abc/WC.txt
put: '/home/hadoop/Desktop/Welcome.txt': No such file or directory
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ nano /home/hadoop/Desktop/welcome.txt
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -put /home/hadoop/Desktop/Welcome.txt /abc/WC.txt
put: '/home/hadoop/Desktop/Welcome.txt': No such file or directory
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -put /home/hadoop/Desktop/welcome.txt /abc/WC.txt
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -cat /abc/WC.txt
hii welcome to hadoop tutorial
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -put /home/hadoop/Desktop/welcome.txt /abc/WC.txtHadoop
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ WC2.txtHdfs dfs -cat /abc/WC2.txt
WC2.txtHdfs: command not found
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ WC.txtHdfs dfs -cat /abc/WC2.txt
WC.txtHdfs: command not found
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ get [-crc]
Command 'get' not found, but there are 18 similar ones.
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -put /home/hadoop/Desktop/Welcome.txt /abc/WC.txtHadoop
put: '/abc/WC.txtHadoop': File exists
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ WC2.txtHdfs dfs -cat /abc/WC2.txt
WC2.txtHdfs: command not found
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /abc/WC2.txt
cat: '/abc/WC2.txt': No such file or directory
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /abc/WC.txt
hii welcome to hadoop tutorial
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -copyToLocal /abc/WC.txt /home/hadoop/Desktop
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -cat /abc/WC.txt
hii welcome to hadoop tutorial
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

## LAB 7

### Map reduce program for word count using eclipse

#### Driver code

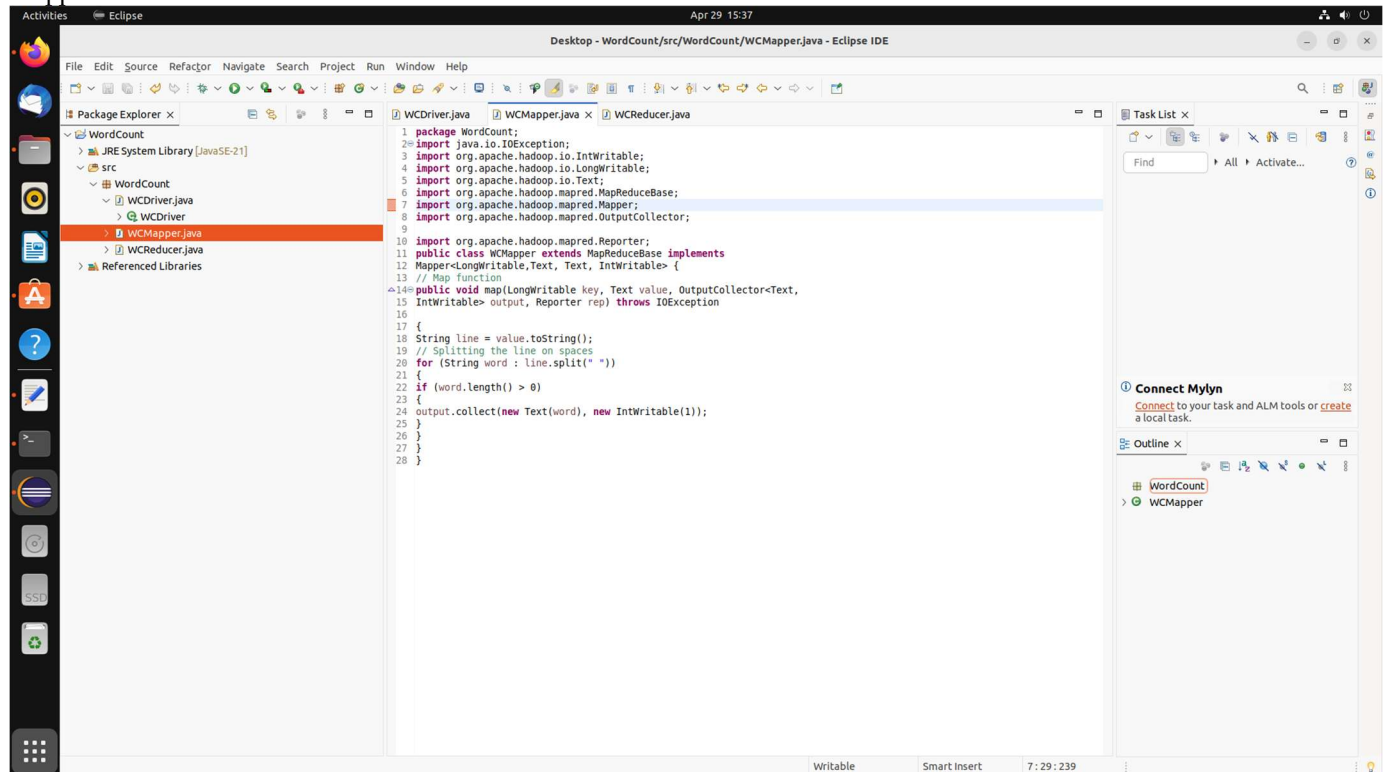


The screenshot shows the Eclipse IDE with the file 'WCDriver.java' open. The code is as follows:

```
1 package WordCount;
2 import java.io.IOException;
3 import org.apache.hadoop.conf.Configuration;
4 import org.apache.hadoop.fs.Path;
5 import org.apache.hadoop.io.IntWritable;
6 import org.apache.hadoop.io.Text;
7 import org.apache.hadoop.mapred.FileInputFormat;
8 import org.apache.hadoop.mapred.FileOutputFormat;
9 import org.apache.hadoop.mapred.JobClient;
10 import org.apache.hadoop.mapred.JobConf;
11 import org.apache.hadoop.util.Tool;
12 import org.apache.hadoop.util.ToolRunner;
13 public class WCDriver extends Configuration Tool {
14     public int run(String args[]) throws IOException
15     {
16         if (args.length < 2)
17         {
18             System.out.println("Please give valid inputs");
19             return -1;
20         }
21         JobConf conf = new JobConf(WCDriver.class);
22         FileInputFormat.setInputPaths(conf, new Path(args[0]));
23         FileOutputFormat.setOutputPath(conf, new Path(args[1]));
24         conf.setMapperClass(WCMapper.class);
25         conf.setReducerClass(WCReducer.class);
26         conf.setMapOutputKeyClass(Text.class);
27         conf.setMapOutputValueClass(IntWritable.class);
28         conf.setOutputKeyClass(Text.class);
29         conf.setOutputValueClass(IntWritable.class);
30         JobClient.runJob(conf);
31         return 0;
32     }
33     // Main Method
34     public static void main(String args[]) throws Exception
35     {
36         int exitCode = ToolRunner.run(new WCDriver(), args);
37         System.out.println(exitCode);
38     }
39 }
```

The Package Explorer on the left shows the project structure: WordCount > src > WordCount > WCDriver.java. The Outline on the right shows the class structure: WordCount > WCDriver > run(String[]): int > main(String[]): void.

#### Mapper code



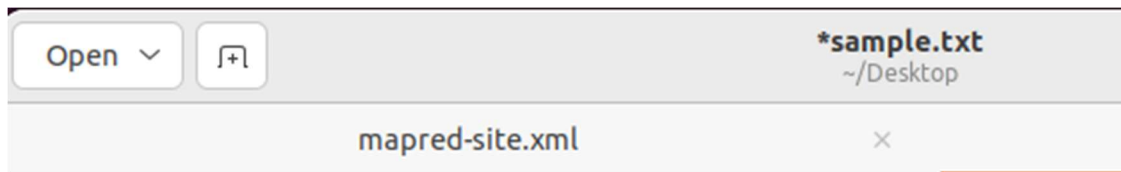
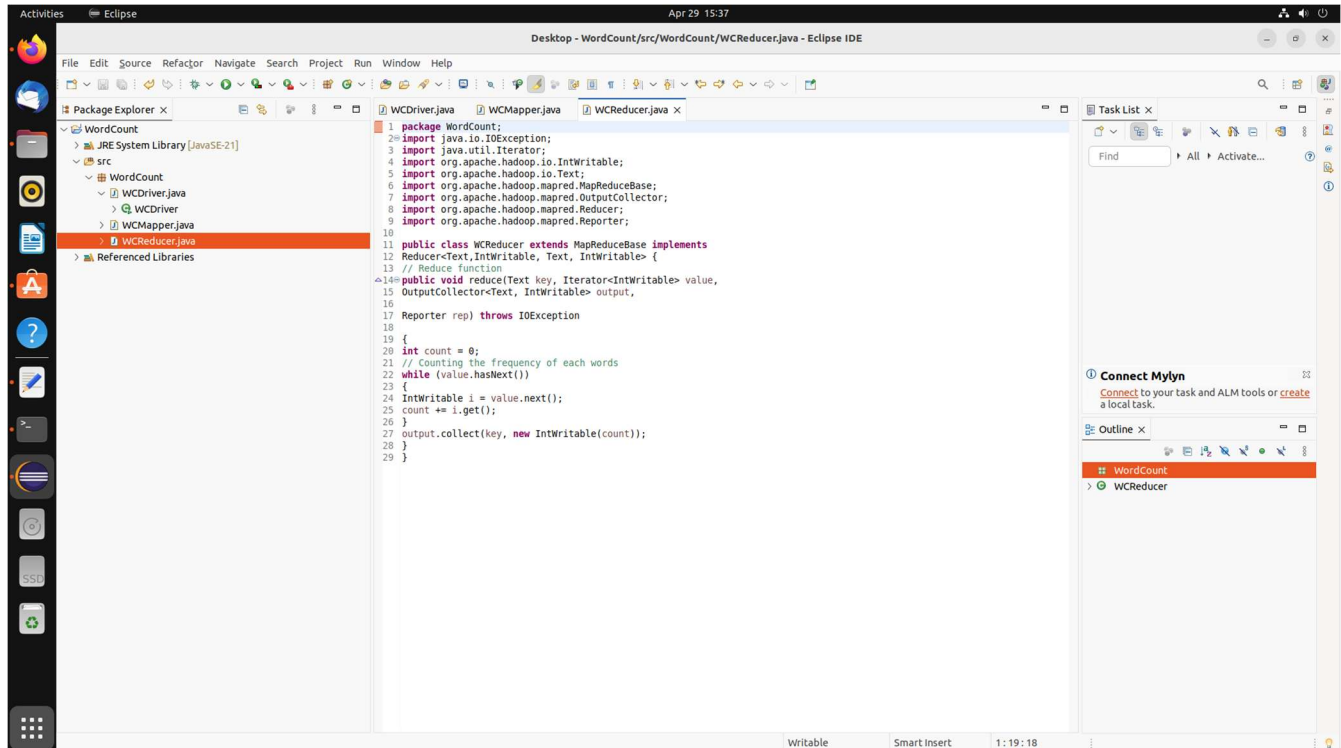
The screenshot shows the Eclipse IDE with the file 'WCMapper.java' open. The code is as follows:

```
1 package WordCount;
2 import java.io.IOException;
3 import org.apache.hadoop.io.IntWritable;
4 import org.apache.hadoop.io.LongWritable;
5 import org.apache.hadoop.io.Text;
6 import org.apache.hadoop.mapred.MapReduceBase;
7 import org.apache.hadoop.mapred.Mapper;
8 import org.apache.hadoop.mapred.OutputCollector;
9
10 import org.apache.hadoop.mapred.Reporter;
11 public class WCMapper extends MapReduceBase implements
12     Mapper<LongWritable, Text, Text, IntWritable> {
13     // Map function
14     public void map(LongWritable key, Text value, OutputCollector<Text,
15         IntWritable> output, Reporter rep) throws IOException
16     {
17         String line = value.toString();
18         // Splitting the line on spaces
19         for (String word : line.split(" "))
20         {
21             if (word.length() > 0)
22             {
23                 output.collect(new Text(word), new IntWritable(1));
24             }
25         }
26     }
27 }
28 }
```

The Package Explorer on the left shows the project structure: WordCount > src > WordCount > WCDriver.java > WCMapper.java. The Outline on the right shows the class structure: WordCount > WCMapper.



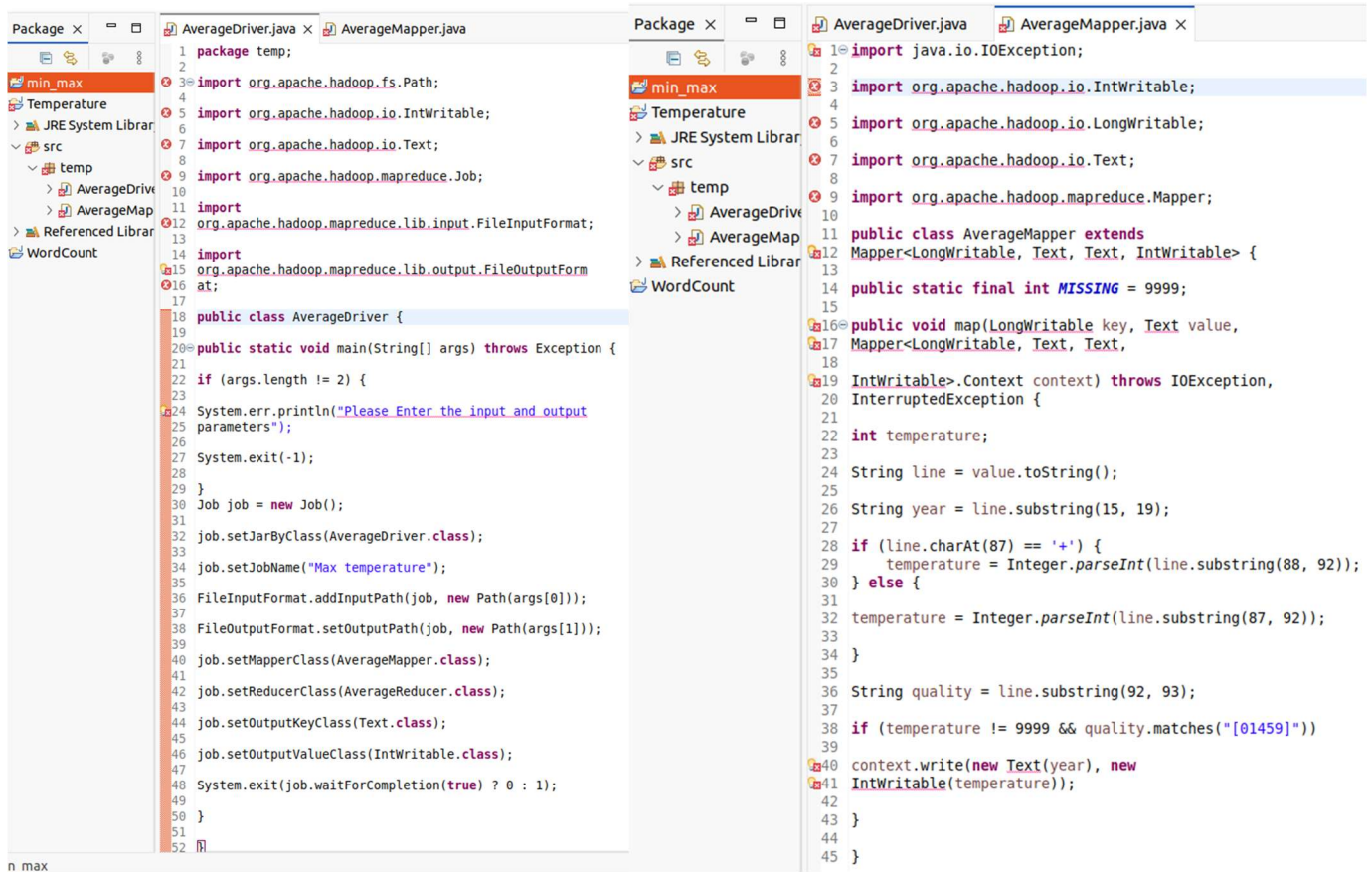
## Reducer code



```
hadoop@bnsccese-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -copyFromLocal -f /home/hadoop/Desktop/file1.txt /rgs/test.txt
hadoop@bnsccese-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop jar /home/hadoop/Desktop/WordCount.jar wordcount.WordCount /rgs/test.txt /output
JAR does not exist or is not a normal file: /home/hadoop/Desktop/WordCount.jar
hadoop@bnsccese-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop jar /home/hadoop/Desktop/Word_Count.jar wordcount.WordCount /rgs/test.txt /output
Exception in thread "main" java.lang.ClassNotFoundException: wordcount.WordCount
    at java.base/java.net.URLClassLoader.findClass(URLClassLoader.java:476)
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:594)
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:527)
    at java.base/java.lang.Class.forName0(Native Method)
    at java.base/java.lang.Class.forName(Class.java:398)
    at org.apache.hadoop.util.RunJar.run(RunJar.java:321)
    at org.apache.hadoop.util.RunJar.main(RunJar.java:241)
hadoop@bnsccese-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
hadoop@bnsccese-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 15:21 /output/_SUCCESS
-rw-r--r-- 1 hadoop supergroup 69 2024-05-21 15:21 /output/part-00000
```

## LAB 8

### Map reduce program for weather . Find average temperature. Find min max temperature



```
Package x Package x
AverageDriver.java x AverageMapper.java x
1 package temp;
2
3 import org.apache.hadoop.fs.Path;
4
5 import org.apache.hadoop.io.IntWritable;
6
7 import org.apache.hadoop.io.Text;
8
9 import org.apache.hadoop.mapreduce.Job;
10
11 import
12 org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
13
14 import
15 org.apache.hadoop.mapreduce.lib.output.FileOutputForm
16 at;
17
18 public class AverageDriver {
19
20 public static void main(String[] args) throws Exception {
21
22 if (args.length != 2) {
23
24 System.err.println("Please Enter the input and output
25 parameters");
26
27 System.exit(-1);
28
29 }
30 Job job = new Job();
31
32 job.setJarByClass(AverageDriver.class);
33
34 job.setJobName("Max temperature");
35
36 FileInputFormat.addInputPath(job, new Path(args[0]));
37
38 FileOutputFormat.setOutputPath(job, new Path(args[1]));
39
40 job.setMapperClass(AverageMapper.class);
41
42 job.setReducerClass(AverageReducer.class);
43
44 job.setOutputKeyClass(Text.class);
45
46 job.setOutputValueClass(IntWritable.class);
47
48 System.exit(job.waitForCompletion(true) ? 0 : 1);
49
50 }
51
52 }
1 import java.io.IOException;
2
3 import org.apache.hadoop.io.IntWritable;
4
5 import org.apache.hadoop.io.LongWritable;
6
7 import org.apache.hadoop.io.Text;
8
9 import org.apache.hadoop.mapreduce.Mapper;
10
11 public class AverageMapper extends
12 Mapper<LongWritable, Text, Text, IntWritable> {
13
14 public static final int MISSING = 9999;
15
16 public void map(LongWritable key, Text value,
17 Mapper<LongWritable, Text, Text,
18 IntWritable>.Context context) throws IOException,
19 InterruptedException {
20
21 int temperature;
22
23 String line = value.toString();
24
25 String year = line.substring(15, 19);
26
27 if (line.charAt(87) == '+') {
28 temperature = Integer.parseInt(line.substring(88, 92));
29 } else {
30 temperature = Integer.parseInt(line.substring(87, 92));
31
32 }
33
34 String quality = line.substring(92, 93);
35
36 if (temperature != 9999 && quality.matches("[01459]"))
37 context.write(new Text(year), new
38 IntWritable(temperature));
39
40 }
41
42 }
43
44 }
45 }
```

```
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ 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 [bmscscse-HP-Elite-Tower-800-G9-Desktop-PC]
Starting resourcemanager
Starting nodemanagers
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ jps
7056 DataNode
7332 SecondaryNameNode
7638 ResourceManager
8231 Jps
5883 org.eclipse.equinox.launcher_1.6.1000.v20250227-1734.jar
7804 NodeManager
6877 NameNode
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /\
> ^C
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /
Found 4 items
drwxr-xr-x - hadoop supergroup 0 2025-04-15 15:00 /FFF
drwxr-xr-x - hadoop supergroup 0 2025-04-15 15:34 /LLL
drwxr-xr-x - hadoop supergroup 0 2024-05-13 14:46 /file
drwxr-xr-x - hadoop supergroup 0 2024-05-13 15:18 /newDataFlair
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /weather
ls: '/weather': No such file or directory
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -mkdir /weather
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -copyFromLocal /home/hadoop/Desktop/1901.txt /weather/test.txt
```



```

2025-05-06 14:59:24,581 INFO mapreduce.Job: Counters: 36
  File System Counters
    FILE: Number of bytes read=153118
    FILE: Number of bytes written=1493804
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=1776380
    HDFS: Number of bytes written=8
    HDFS: Number of read operations=15
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=4
    HDFS: Number of bytes read erasure-coded=0
  Map-Reduce Framework
    Map input records=6565
    Map output records=6564
    Map output bytes=59076
    Map output materialized bytes=72210
    Input split bytes=103
    Combine input records=0
    Combine output records=0
    Reduce input groups=1
    Reduce shuffle bytes=72210
    Reduce input records=6564
    Reduce output records=1
    Spilled Records=13128
    Shuffled Maps =1
    Failed Shuffles=0
    Merged Map outputs=1
    GC time elapsed (ms)=0
    Total committed heap usage (bytes)=1266679808
  Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
  File Input Format Counters
    Bytes Read=888190
  File Output Format Counters
    Bytes Written=8

```

```

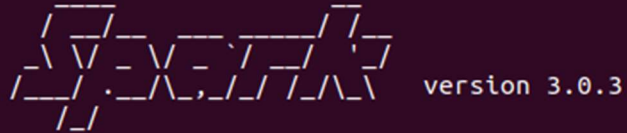
    Bytes Written=8
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /weather
Found 2 items
drwxr-xr-x   - hadoop supergroup          0 2025-05-06 14:59 /weather/output
-rw-r--r--   1 hadoop supergroup      888190 2025-05-06 14:50 /weather/test.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /weather/output
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2025-05-06 14:59 /weather/output/_SUCCESS
-rw-r--r--   1 hadoop supergroup          8 2025-05-06 14:59 /weather/output/part-r-00000
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /weather/output/part-r-00000
1901    46
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ █

```

## LAB 9

### Scala program to print number from 1 to 100 using for loop

```
Spark context available as 'sc' (master = local[*], app id = local-174771751320)
Spark session available as 'spark'.
Welcome to
```



```
Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.18)
Type in expressions to have them evaluated.
Type :help for more information.
```

```
scala> for (i <- 1 to 100){ println(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
```

## Lab 10

Using RDD and flatmap count the frequency of words appear in a file and write out list of words where count > 4

```
Open ▾ word_count.py
~/
1 from pyspark import SparkContext
2
3 sc = SparkContext("local", "SimpleWordCount")
4 text_file = sc.textFile("file1.txt")
5 counts = text_file.flatMap(lambda line: line.split()) \
6                       .map(lambda word: (word, 1)) \
7                       .reduceByKey(lambda a, b: a + b)
8 output = counts.collect()
9
10 for (word, count) in output:
11     print(f"{word}: {count}")
12
```

```
hello: 2
bmsce: 3
how: 1
are: 1
you: 1
```