

**B.M.S. COLLEGE OF ENGINEERING BENGALURU**

Autonomous Institute, Affiliated to VTU



Lab Record

**BIG DATA ANALYTICS**

*Submitted in partial fulfilment for the 6<sup>th</sup> Semester Laboratory*

Bachelor of Technology

in

Computer Science and Engineering

*Submitted by:*

**SAKSHI SRIVASTAVA**

1BM18CS090

Department of Computer Science and Engineering

B.M.S. College of Engineering

Bull Temple Road, Basavanagudi, Bangalore 560 019

Mar-June 2021

**B.M.S. COLLEGE OF ENGINEERING**  
**DEPARTMENT OF COMPUTER SCIENCE AND**  
**ENGINEERING**



***CERTIFICATE***

This is to certify that the Big Data Analytics(20CS6PEBDA) laboratory has been carried out by SAKSHI SRIVASTAVA (1BM18CS090) during the 6<sup>th</sup> Semester Mar-June-2021.

Signature of the Faculty In charge:

SHEETAL V A

Department of Computer Science and Engineering

B.M.S. College of Engineering, Bangalore

## TABLE OF CONTENTS

SL NO	TITLE
1	EMPLOYEE DATABASE
2	LIBRARY DATABASE
3	MONGODB SAMPLE
4	HADOOP INSTALLATION
5	HADOOP SAMPLE
6	MAPREDUCE TEMPERATURE
7	MAPREDUCE TOPN
8	MAPREDUCE JOIN
9	SCALA INSTALLATION
10	SCALA WORDCOUNT

## Employee database (CASSANDRA)

Date - 29/03/2021

Question -

Perform the following DB operations using Cassandra.

1. Create a keyspace by name Employee
2. Create a column family by name Employee-Info with attributes Emp\_Id Primary Key, Emp\_Name, Designation, Date\_of\_Joining, Salary, Dept\_Name
3. Insert the values into the table in batch
3. Update Employee name and Department of Emp-Id 121
4. Sort the details of Employee records based on salary
5. Alter the schema of the table Employee\_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.
6. Update the altered table to add project names.
7. Create a TTL of 30 seconds to display the values of Employees.

```
cqlsh:employee_info> begin batch
```

```
    ... insert into
employee_details(emp_id,emp_name,designation,doj,salary,dept_name)values(101,'Sakshi','manager','2020-09-08',35000,'testing')
```

```
    ... insert into
employee_details(emp_id,emp_name,designation,doj,salary,dept_name)values(201,'Sneha','manager','2020-08-08',85000,'development')
```

```
    ... insert into
employee_details(emp_id,emp_name,designation,doj,salary,dept_name)values(201,'Shreya','associate','2020-07-08',75000,'HR')
```

```
    ... apply batch;
```

```
cqlsh:employee_info> select *from employee_details;
```

```
emp_id | salary | dept_name | designation | doj | emp_name
```

```

-----+-----+-----+-----+-----+-----
201 | 75000 |      HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya
201 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi

```

(3 rows)

```
cqlsh:employee_info> delete from employee_details where emp_id=201;
```

```
cqlsh:employee_info> select *from employee_details;
```

```

emp_id | salary | dept_name | designation | doj                | emp_name
-----+-----+-----+-----+-----+-----
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi

```

(1 rows)

```

cqlsh:employee_info> begin batch insert into
employee_details(emp_id,emp_name,designation,doj,salary,dept_name)values(301,'Sneha','manag
er','2020-08-08',85000,'development') insert into
employee_details(emp_id,emp_name,designation,doj,salary,dept_name)values(201,'Shreya','associ
ate','2020-07-08',75000,'HR') apply batch;
```

```
cqlsh:employee_info> select *from employee_details;
```

```

emp_id | salary | dept_name | designation | doj                | emp_name
-----+-----+-----+-----+-----+-----
201 | 75000 |      HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi
301 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha

```

(3 rows)

```
cqlsh:employee_info> alter table employee_details add project text;
```

```
cqlsh:employee_info> update employee_details set project='library app' where emp_id=201 and
salary=75000;
```

```
cqlsh:employee_info> update employee_details set project='medicine app' where emp_id=301 and
salary=85000;
```

```
cqlsh:employee_info> update employee_details set project='fitness app' where emp_id=101 and salary=85000;
```

```
cqlsh:employee_info> select *from employee_details;
```

emp_id	salary	dept_name	designation	doj	emp_name	project
201	75000	HR	associate	2020-07-08 07:00:00.000000+0000	Shreya	library app
101	35000	testing	manager	2020-09-08 07:00:00.000000+0000	Sakshi	null
101	85000	null	null	null	null	fitness app
301	85000	development	manager	2020-08-08 07:00:00.000000+0000	Sneha	medicine app

(4 rows)

```
cqlsh:employee_info> update employee_details set project='fitness app' where emp_id=101 and salary=35000;
```

```
cqlsh:employee_info> select *from employee_details;
```

emp_id	salary	dept_name	designation	doj	emp_name	project
201	75000	HR	associate	2020-07-08 07:00:00.000000+0000	Shreya	library app
101	35000	testing	manager	2020-09-08 07:00:00.000000+0000	Sakshi	fitness app
101	85000	null	null	null	null	fitness app
301	85000	development	manager	2020-08-08 07:00:00.000000+0000	Sneha	medicine app

(4 rows)

```
cqlsh:employee_info> delete from employee_details where emp_id=1 and salary=85000;
```

```
cqlsh:employee_info> select *from employee_details;
```

emp_id	salary	dept_name	designation	doj	emp_name	project
201	75000	HR	associate	2020-07-08 07:00:00.000000+0000	Shreya	library app

```

101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi | fitness app
101 | 85000 | null | null | null | null | fitness app
301 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha |
medicine app

```

(4 rows)

```
cqlsh:employee_info> select *from employee_details;
```

```

emp_id | salary | dept_name | designation | doj | emp_name | project
-----+-----+-----+-----+-----+-----+-----
201 | 75000 | HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya | library app
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi | fitness app
401 | 65000 | testing | manager | 2020-05-08 07:00:00.000000+0000 | Resh | null
301 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha |
medicine app

```

(4 rows)

```
cqlsh:employee_info> insert into
employee_details(emp_id,emp_name,designation,doj,salary,dept_name)values(501,'wesh','manage
r','2020-04-08',95000,'testing') using ttl 30;
```

```
cqlsh:employee_info> select ttl(emp_name)from employee_details where emp_id=501 and
salary=95000;
```

```
ttl(emp_name)
```

```
-----
```

```
24
```

(1 rows)

```
cqlsh:employee_info> select *from employee_details;
```

```

emp_id | salary | dept_name | designation | doj | emp_name | project
-----+-----+-----+-----+-----+-----+-----
201 | 75000 | HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya | library app

```

```

101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi | fitness app
401 | 65000 | testing | manager | 2020-05-08 07:00:00.000000+0000 | Resh | null
301 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha |
medicine app

```

(4 rows)

```
cqlsh:employee_info> paging off
```

Disabled Query paging.

```
cqlsh:employee_info> select *from employee_details where emp_id in(201,101,301) order by salary;
```

```

emp_id | salary | dept_name | designation | doj | emp_name | project
-----+-----+-----+-----+-----+-----+-----
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi | fitness app
201 | 75000 | HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya | library app
301 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha |
medicine app

```



## SCREENSHOTS

The first screenshot shows a terminal window with the following SQL queries and results:

```

emp_id | salary | dept_name | designation | doj | emp_name | project
-----|-----|-----|-----|-----|-----|-----
201 | 75000 | HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya | library app
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi | fitness app
101 | 85000 | null | null | 2020-09-08 07:00:00.000000+0000 | null | null
301 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha | medicine app

(4 rows)
cqlsh:employee_info> delete from employee_details where emp_id=101 and salary=85000;
cqlsh:employee_info> select *from employee_details;

emp_id | salary | dept_name | designation | doj | emp_name | project
-----|-----|-----|-----|-----|-----|-----
201 | 75000 | HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya | library app
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi | fitness app
301 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha | medicine app

(3 rows)
cqlsh:employee_info> begin batch insert into employee_details(emp_id,emp_name,designation,doj,salary,dept_name)values(401,'Resh','manager','2020-05-08','65000','testing')
... apply batch;
cqlsh:employee_info> begin batch insert into employee_details(emp_id,emp_name,designation,doj,salary,dept_name)values(501,'wesh','manager','2020-04-08','95000','testing') using ttl 30;
...
cqlsh:employee_info> insert into employee_details(emp_id,emp_name,designation,doj,salary,dept_name)values(501,'wesh','manager','2020-04-08','95000','testing') using ttl 30;
cqlsh:employee_info> select ttl(emp_name)from employee_details where emp_id=501 and salary=95000;

ttl(emp_name)
-----
24

(0 rows)
cqlsh:employee_info> select *from employee_details;

emp_id | salary | dept_name | designation | doj | emp_name | project
-----|-----|-----|-----|-----|-----|-----
201 | 75000 | HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya | library app
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi | fitness app
401 | 65000 | testing | manager | 2020-05-08 07:00:00.000000+0000 | Resh | null
301 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha | medicine app

(4 rows)
cqlsh:employee_info> insert into employee_details(emp_id,emp_name,designation,doj,salary,dept_name)values(501,'wesh','manager','2020-04-08','95000','testing') using ttl 30;
cqlsh:employee_info> select ttl(emp_name)from employee_details where emp_id=501 and salary=95000;

ttl(emp_name)
-----
24

*** (1 rows)

```

The second screenshot shows a terminal window with the following SQL queries and results:

```

emp_id | salary | dept_name | designation | doj | emp_name
-----|-----|-----|-----|-----|-----
201 | 75000 | HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya
101 | 35000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi

(3 rows)
cqlsh:employee_info> update employee_details set emp_id=301 where emp_name='Sneha';
cqlsh:employee_info> delete from employee
... -de
cqlsh:employee_info> delete from employee_details where emp_name='Sneha';
cqlsh:employee_info> delete from employee_details where emp_id=201;
cqlsh:employee_info> delete from employee_details where emp_id=201;
cqlsh:employee_info> select *from employee_details;

emp_id | salary | dept_name | designation | doj | emp_name
-----|-----|-----|-----|-----|-----
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi

(1 rows)
cqlsh:employee_info> begin batch insert into employee_details(emp_id,emp_name,designation,doj,salary,dept_name)values(301,'Sneha','manager','2020-08-08','85000','development') insert into employee_details(
emp_id,emp_name,designation,doj,salary,dept_name)values(201,'Shreya','associate','2020-07-08','75000','HR') apply batch;
cqlsh:employee_info> select *from employee_details;

emp_id | salary | dept_name | designation | doj | emp_name
-----|-----|-----|-----|-----|-----
201 | 75000 | HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi
301 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha

(3 rows)
cqlsh:employee_info> alter table employee_details add project text;
cqlsh:employee_info> update employee_details set project='library app' where emp_id=201 and salary=75000;
cqlsh:employee_info> update employee_details set project='medicine app' where emp_id=301 and salary=85000;
cqlsh:employee_info> update employee_details set project='fitness app' where emp_id=101 and salary=85000;
cqlsh:employee_info> select *from employee_details;

emp_id | salary | dept_name | designation | doj | emp_name | project
-----|-----|-----|-----|-----|-----|-----
201 | 75000 | HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya | library app
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi | null
101 | 85000 | null | null | 2020-09-08 07:00:00.000000+0000 | null | null
301 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha | medicine app

(4 rows)

```

```
Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Player
Activities
Terminal
May 23 10:49 PM
sakshi@ubuntu: ~

cqlsh:employee_info> insert into employee_details(emp_id,emp_name,designation,dof,salary,dept_name)values(501,'wesh','manager','2020-04-08',95000,'testing') using ttl 30;
cqlsh:employee_info> select ttl(emp_name)from employee_details where emp_id=501 and salary=95000;

ttl(emp_name)
-----
(0 rows)

cqlsh:employee_info> select *from employee_details;

emp_id | salary | dept_name | designation | dof | emp_name | project
-----|-----|-----|-----|-----|-----|-----
201 | 75000 | HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya | library app
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi | fitness app
401 | 45000 | testing | manager | 2020-05-08 07:00:00.000000+0000 | Resh | null
301 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha | medicine app

(4 rows)

cqlsh:employee_info> insert into employee_details(emp_id,emp_name,designation,dof,salary,dept_name)values(501,'wesh','manager','2020-04-08',95000,'testing') using ttl 30;
cqlsh:employee_info> select ttl(emp_name)from employee_details where emp_id=501 and salary=95000;

ttl(emp_name)
-----
24

(1 rows)

cqlsh:employee_info> select *from employee_details;

emp_id | salary | dept_name | designation | dof | emp_name | project
-----|-----|-----|-----|-----|-----|-----
201 | 75000 | HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya | library app
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi | fitness app
401 | 45000 | testing | manager | 2020-05-08 07:00:00.000000+0000 | Resh | null
301 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha | medicine app

(4 rows)

cqlsh:employee_info> paging off
Disabled Query paging.
cqlsh:employee_info> select *from employee_details where emp_id in(201,101,301) order by salary;

emp_id | salary | dept_name | designation | dof | emp_name | project
-----|-----|-----|-----|-----|-----|-----
101 | 35000 | testing | manager | 2020-09-08 07:00:00.000000+0000 | Sakshi | fitness app
201 | 75000 | HR | associate | 2020-07-08 07:00:00.000000+0000 | Shreya | library app
301 | 85000 | development | manager | 2020-08-08 07:00:00.000000+0000 | Sneha | medicine app

... (3 rows)
```

# LIBRARY DATABASE (CASSANDRA)

Date - 29/03/2021

Question -

Perform the following DB operations using Cassandra.

1. Create a keyspace by name Library
2. Create a column family by name Library-Info with attributes  
Stud\_Id Primary Key,  
Counter\_value of type Counter,  
Stud\_Name, Book-Name, Book-Id, Date\_of\_issue
3. Insert the values into the table in batch
3. Display the details of the table created and increase the value of the counter
4. Write a query to show that a student with id 112 has taken a book "BDA" 2 times.
5. Export the created column to a csv file
6. Import a given csv dataset from local file system into Cassandra column family

```
cqlsh> use library_info;
```

```
cqlsh:library_info> create table library_details(stud_id int,counter_value  
... counter,stud_name text,book_name text,date_of_issue timestamp,book_id  
... int,primary key(stud_id,stud_name,book_name,date_of_issue,book_id));
```

```
cqlsh:library_info> update library_details set counter_value=counter_value+1  
... where stud_id=111 and stud_name='sam' and book_name='ML' and  
... date_of_issue='2020-11-09' and book_id=200;
```

```
cqlsh:library_info> select *from library_details;
```

stud_id	stud_name	book_name	date_of_issue	book_id	counter_value
-----+-----+-----+-----+-----+-----					

111 | sam | ML | 2020-11-09 08:00:00.000000+0000 | 200 | 1

(1 rows)

```
cqlsh:library_info> update library_details set counter_value=counter_value+1 where stud_id=112
and stud_name='sakshi' and book_name='BDA' and date_of_issue='2020-01-01' and book_id=300;
```

```
cqlsh:library_info> update library_details set counter_value=counter_value+1 where stud_id=115
and stud_name='aditya' and book_name='OOMD' and date_of_issue='2020-06-01' and
book_id=400;
```

```
cqlsh:library_info> select *from library_details;
```

stud_id	stud_name	book_name	date_of_issue	book_id	counter_value
111	sam	ML	2020-11-09 08:00:00.000000+0000	200	1
112	sakshi	BDA	2020-01-01 08:00:00.000000+0000	300	1
115	aditya	OOMD	2020-06-01 07:00:00.000000+0000	400	1

```
qlsh:library_info> copy
library_details(stud_id,stud_name,book_name,book_id,date_of_issue,counter_value) to
'C:\Desktop\sample.csv';
```

Using 1 child processes

Starting copy of library\_info.library\_details with columns [stud\_id, stud\_name, book\_name, book\_id, date\_of\_issue, counter\_value].

cqlshlib.copyutil.ExportProcess.write\_rows\_to\_csv(): writing row

cqlshlib.copyutil.ExportProcess.write\_rows\_to\_csv(): writing row

cqlshlib.copyutil.ExportProcess.write\_rows\_to\_csv(): writing row/s

Processed: 3 rows; Rate: 37 rows/s; Avg. rate: 6 rows/s

3 rows exported to 1 files in 0.500 seconds.

```
cqlsh:library_info> truncate library_details;
```

```
cqlsh:library_info> copy
library_details(stud_id,stud_name,book_name,book_id,date_of_issue,counter_value) from
'C:\Desktop\sample.csv';
```

Using 1 child processes

Starting copy of library\_info.library\_details with columns [stud\_id, stud\_name, book\_name, book\_id, date\_of\_issue, counter\_value].

Processed: 3 rows; Rate: 3 rows/s; Avg. rate: 5 rows/s

3 rows imported from 1 files in 0.592 seconds (0 skipped).

cqlsh:library\_info> select \*from library\_details;

stud_id	stud_name	book_name	date_of_issue	book_id	counter_value
111	sam	ML	2020-11-09 08:00:00.000000+0000	200	1
112	sakshi	BDA	2020-08-01 07:00:00.000000+0000	300	2
115	aditya	OOMD	2020-06-01 07:00:00.000000+0000	400	1

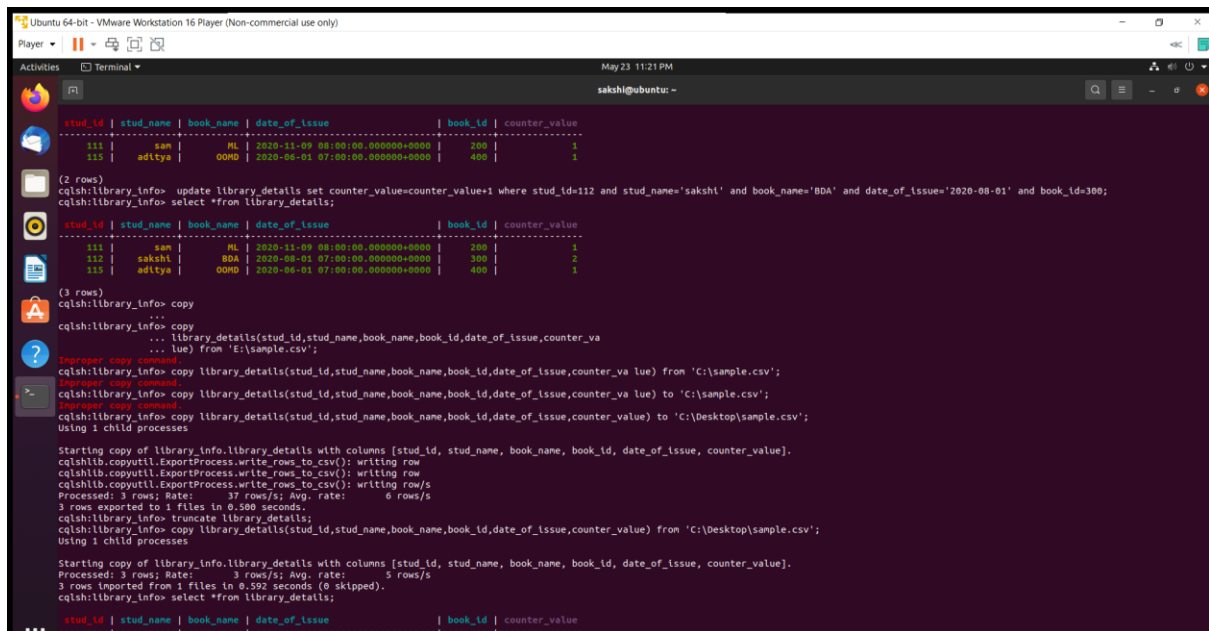
SCREENSHOTS:

```
cqlsh> use library_info;
cqlsh:library_info> create table library_details(stud_id int,counter_value
... counter,stud_name text,book_name text,date_of_issue timestamp,book_id
... int,primary key(stud_id,stud_name,book_name,date_of_issue,book_id));
cqlsh:library_info> update library_details set counter_value=counter_value+1
... where stud_id=111 and stud_name='sam' and book_name='ML' and
... date_of_issue='2020-11-09' and book_id=200;
cqlsh:library_info> select *from library_details;

stud_id | stud_name | book_name | date_of_issue | book_id | counter_value
-----+-----+-----+-----+-----+-----
111 | sam | ML | 2020-11-09 08:00:00.000000+0000 | 200 | 1
(1 rows)
cqlsh:library_info> update library_details set counter_value=counter_value+1
... where stud_id=112 and stud_name='shaan' and book_name='BDA' and
... date_of_issue='2020-01-01' and book_id=300;
SyntaxException: line 1:1 no viable alternative at input 'select *...from library_details where stud_id=112[select]...'
cqlsh:library_info> update library_details set counter_value=counter_value+1 where stud_id=112 and stud_name='sakshi' and book_name='BDA' and date_of_issue='2020-01-01' and book_id=300;
cqlsh:library_info> update library_details set counter_value=counter_value+1 where stud_id=115 and stud_name='aditya' and book_name='OOMD' and date_of_issue='2020-06-01' and book_id=400;
cqlsh:library_info> select *from library_details;

stud_id | stud_name | book_name | date_of_issue | book_id | counter_value
-----+-----+-----+-----+-----+-----
111 | sam | ML | 2020-11-09 08:00:00.000000+0000 | 200 | 1
112 | sakshi | BDA | 2020-01-01 08:00:00.000000+0000 | 300 | 1
115 | aditya | OOMD | 2020-06-01 07:00:00.000000+0000 | 400 | 1
(3 rows)
cqlsh:library_info> update library_details set counter_value=counter_value+1 where stud_id=112 and stud_name='sakshi' and book_name='BDA' and date_of_issue='2020-08-01' and book_id=300;
cqlsh:library_info> select *from library_details;

stud_id | stud_name | book_name | date_of_issue | book_id | counter_value
-----+-----+-----+-----+-----+-----
111 | sam | ML | 2020-11-09 08:00:00.000000+0000 | 200 | 1
112 | sakshi | BDA | 2020-01-01 08:00:00.000000+0000 | 300 | 1
112 | sakshi | BDA | 2020-08-01 07:00:00.000000+0000 | 300 | 1
115 | aditya | OOMD | 2020-06-01 07:00:00.000000+0000 | 400 | 1
(4 rows)
cqlsh:library_info> select *from library_details where stud_id=112
... select *from library_details where stud_id=112;
SyntaxException: line 2:18 no viable alternative at input 'select *...from library_details where stud_id=112[select]...'
cqlsh:library_info> delete *from library_details where stud_id=112 and stud_name='sakshi';
SyntaxException: line 1:1 unexpected token '*' expecting ';' or EOF (delete [;])
cqlsh:library_info> delete *from library_details where stud_id=112 and stud_name='sakshi';
SyntaxException: line 1:1 unexpected token '*' expecting ';' or EOF (delete [;])
```



# MONGODB SAMPLE

Date - 05/04/2021

Question -

Perform the following DB operations using MongoDB.

1. Create a database “Student” with the following attributes Rollno, Age, ContactNo, Email-Id.
2. Insert appropriate values
3. Write a query to update Email-Id of a student with rollno 10.
4. Replace the student name from “ABC” to “FEM” of rollno 11.
5. Export the created table into local file system
6. Drop the table
7. Import a given csv dataset from the local file system into mongodb collection.

```
> use students
```

```
switched to db students
```

```
> db.createCollection("stud_details")
```

```
{ "ok" : 1 }
```

```
>
```

```
db.stud_details.insert({'name':'Sakshi','rollno':1,'age':19,'contactno':'8670794779','email':'sakshi@bmsce.ac.in'})
```

```
WriteResult({ "nInserted" : 1 })
```

```
>
```

```
db.stud_details.insert({'name':'Sneha','rollno':2,'age':20,'contactno':'8670794789','email':'sneha@bmsce.ac.in'})
```

```
WriteResult({ "nInserted" : 1 })
```

```
>
```

```
db.stud_details.insert({'name':'Shruti','rollno':3,'age':21,'contactno':'8630394789','email':'shruti@bmsce.ac.in'})
```

```
WriteResult({ "nInserted" : 1 })
```

```

> db.stud_details.find({})

{ "_id" : ObjectId("60aaabf1b1aea56bb97beef8"), "name" : "Sakshi", "rollno" : 1, "age" : 19,
"contactno" : "8670794779", "email" : "sakshi@bmsce.ac.in" }

{ "_id" : ObjectId("60aaac16b1aea56bb97beef9"), "name" : "Sneha", "rollno" : 2, "age" : 20,
"contactno" : "8670794789", "email" : "sneha@bmsce.ac.in" }

{ "_id" : ObjectId("60aaac41b1aea56bb97beefa"), "name" : "Shruti", "rollno" : 3, "age" : 21,
"contactno" : "8630394789", "email" : "shruti@bmsce.ac.in" }

> db.student_details.update({'rollno':3},{ $set: {'email': 'update@lab.com'}})

WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })

> db.stud_details.find({'rollno':3})

{ "_id" : ObjectId("60aaac41b1aea56bb97beefa"), "name" : "Shruti", "rollno" : 3, "age" : 21,
"contactno" : "8630394789", "email" : "shruti@bmsce.ac.in" }


mongoexport --db students --collection stud_details --out C:\Desktop\sample.json

2021-05-22T10:43:30.687+0530 connected to: mongodb://localhost/ 2021-05-
22T10:43:31.026+0530 exported 3 records

mongoimport --db students --collection stud_details --type=json --file= C:\Desktop\sample.json

2021-05-22T10:46:49.898+0530 connected to: mongodb://localhost/ 2021-05-
22T10:46:50.044+0530 3 document(s) imported successfully. 0 document(s) failed to import.

db.stud_details.find({})

{ "_id" : ObjectId("60aaabf1b1aea56bb97beef8"), "name" : "Sakshi", "rollno" : 1, "age" : 19,
"contactno" : "8670794779", "email" : "sakshi@bmsce.ac.in" }

{ "_id" : ObjectId("60aaac16b1aea56bb97beef9"), "name" : "Sneha", "rollno" : 2, "age" : 20,
"contactno" : "8670794789", "email" : "sneha@bmsce.ac.in" }

{ "_id" : ObjectId("60aaac41b1aea56bb97beefa"), "name" : "Shruti", "rollno" : 3, "age" : 21,
"contactno" : "8630394789", "email" : "shruti@bmsce.ac.in" }

> db.student_details.remove({age:{$gt:20}})

WriteResult({ "nRemoved" : 0 })

> db.stud_details.find({})

{ "_id" : ObjectId("60aaabf1b1aea56bb97beef8"), "name" : "Sakshi", "rollno" : 1, "age" : 19,
"contactno" : "8670794779", "email" : "sakshi@bmsce.ac.in" }

{ "_id" : ObjectId("60aaac16b1aea56bb97beef9"), "name" : "Sneha", "rollno" : 2, "age" : 20,
"contactno" : "8670794789", "email" : "sneha@bmsce.ac.in" }

{ "_id" : ObjectId("60aaac41b1aea56bb97beefa"), "name" : "Shruti", "rollno" : 3, "age" : 21,
"contactno" : "8630394789", "email" : "shruti@bmsce.ac.in" }

```



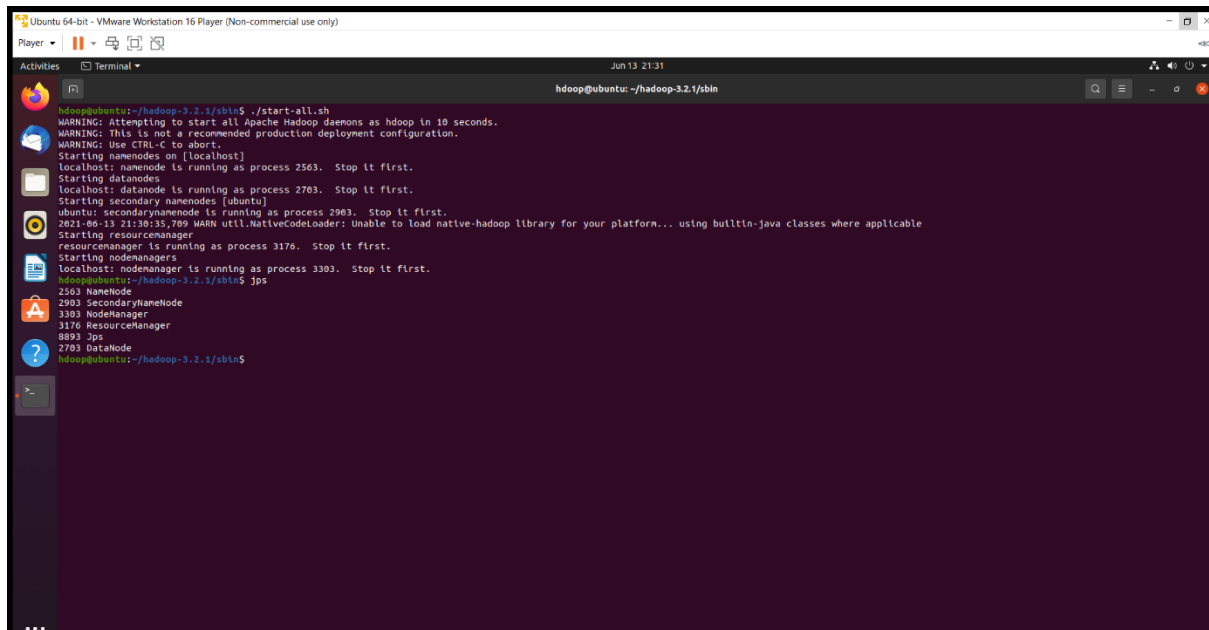
## SCREENSHOTS:

```
C:\Program Files\MongoDB\Server\4.4\bin\mongo.exe
---
> use student
switched to db student
> db.createCollection("details")
{
  "ok" : 0,
  "errmsg" : "db already exists with different case already have: [Student] trying to create [student]",
  "code" : 13297,
  "codeName" : "DatabaseDifferCase"
}
> db.createCollection("detail")
{
  "ok" : 0,
  "errmsg" : "db already exists with different case already have: [Student] trying to create [student]",
  "code" : 13297,
  "codeName" : "DatabaseDifferCase"
}
> db.createCollection("details")
{
  "ok" : 0,
  "errmsg" : "db already exists with different case already have: [Student] trying to create [student]",
  "code" : 13297,
  "codeName" : "DatabaseDifferCase"
}
> clear
uncaught exception: ReferenceError: clear is not defined :
@shell:1:1
> exot
uncaught exception: ReferenceError: exot is not defined :
@shell:1:1
> use students
switched to db students
> createcollection("stud_details")
uncaught exception: ReferenceError: createcollection is not defined :
@shell:1:1
> db.createCollection("stud_details")
{ "ok" : 1 }
> db.stud_details.insert({'name':'Sakshi','rollno':1,'age':19,'contactno':'8670794779','email':'sakshi@bmsce.ac.in'})
WriteResult({ "nInserted" : 1 })
> db.stud_details.insert({'name':'Sneha','rollno':2,'age':20,'contactno':'8670794789','email':'sneha@bmsce.ac.in'})
WriteResult({ "nInserted" : 1 })
> db.stud_details.insert({'name':'Shruti','rollno':3,'age':21,'contactno':'8630394789','email':'shruti@bmsce.ac.in'})
WriteResult({ "nInserted" : 1 })
> db.stud_details.find()
{ "_id" : ObjectId("60aaabf1b1aea56bb97beef8"), "name" : "Sakshi", "rollno" : 1, "age" : 19, "contactno" : "8670794779", "email" : "sakshi@bmsce.ac.in" }
{ "_id" : ObjectId("60aaac16b1aea56bb97beef9"), "name" : "Sneha", "rollno" : 2, "age" : 20, "contactno" : "8670794789", "email" : "sneha@bmsce.ac.in" }
{ "_id" : ObjectId("60aaac41b1aea56bb97beefa"), "name" : "Shruti", "rollno" : 3, "age" : 21, "contactno" : "8630394789", "email" : "shruti@bmsce.ac.in" }
> db.stud_details.update({'rollno':3},{set:{'email':'update@lab.com'}})
WriteResult({ "nInserted" : 0, "nModified" : 1 })

C:\Program Files\MongoDB\Server\4.4\bin\mongo.exe
> db.stud_details.find()
> db.stud_details.find()
{ "_id" : ObjectId("60aaabf1b1aea56bb97beef8"), "name" : "Sakshi", "rollno" : 1, "age" : 19, "contactno" : "8670794779", "email" : "sakshi@bmsce.ac.in" }
{ "_id" : ObjectId("60aaac16b1aea56bb97beef9"), "name" : "Sneha", "rollno" : 2, "age" : 20, "contactno" : "8670794789", "email" : "sneha@bmsce.ac.in" }
{ "_id" : ObjectId("60aaac41b1aea56bb97beefa"), "name" : "Shruti", "rollno" : 3, "age" : 21, "contactno" : "8630394789", "email" : "shruti@bmsce.ac.in" }
> db.stud_details.remove({'age':{'$gt':20}})
WriteResult({ "nRemoved" : 0 })
> db.stud_details.find()
{ "_id" : ObjectId("60aaabf1b1aea56bb97beef8"), "name" : "Sakshi", "rollno" : 1, "age" : 19, "contactno" : "8670794779", "email" : "sakshi@bmsce.ac.in" }
{ "_id" : ObjectId("60aaac16b1aea56bb97beef9"), "name" : "Sneha", "rollno" : 2, "age" : 20, "contactno" : "8670794789", "email" : "sneha@bmsce.ac.in" }
{ "_id" : ObjectId("60aaac41b1aea56bb97beefa"), "name" : "Shruti", "rollno" : 3, "age" : 21, "contactno" : "8630394789", "email" : "shruti@bmsce.ac.in" }
```

# SCREENSHOT OF HADOOP INSTALLATION

Date - 12/04/2021



The screenshot shows a terminal window within a VMware Workstation 16 Player. The terminal is running the command `./start-all.sh` in the `/hadoop-3.2.1/sbin` directory. The output displays various warnings and status messages for starting Hadoop daemons. It indicates that the NameNode is running as process 2503, the DataNode as process 2703, and the Secondary NameNode as process 2903. It also shows the Resource Manager running as process 3176 and the Node Manager as process 3303. The terminal output is as follows:

```
hadoop@ubuntu:~/hadoop-3.2.1/sbin$ ./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 2503. Stop it first.
Starting datanodes
localhost: datanode is running as process 2703. Stop it first.
Starting secondary namenodes [ubuntu]
ubuntu: secondarynamenode is running as process 2903. Stop it first.
2021-06-13 21:38:35,709 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting resourceManager
resourceManager is running as process 3176. Stop it first.
Starting nodeManagers
localhost: nodemanager is running as process 3303. Stop it first.
hadoop@ubuntu:~/hadoop-3.2.1/sbin$ jps
2503 NameNode
2903 SecondaryNameNode
3303 NodeManager
3176 ResourceManager
8893 Jps
2703 DataNode
hadoop@ubuntu:~/hadoop-3.2.1/sbin$
```

## HADOOP SAMPLE

Date - 19/04/2021

Execution of HDFS Commands for interaction with Hadoop Environment. (Minimum 10 commands to be executed)

```
c:\hadoop_new\sbin>hdfs dfs -mkdir /temp
```

```
c:\hadoop_new\sbin>hdfs dfs -copyFromLocal E:\Desktop\sample.txt \temp
```

```
c:\hadoop_new\sbin>hdfs dfs -ls \temp
```

Found 1 items

```
-rw-r--r--  1 Admin supergroup      11 2021-06-11 21:12 /temp/sample.txt
```

```
c:\hadoop_new\sbin>hdfs dfs -cat \temp\sample.txt
```

hello world

```
c:\hadoop_new\sbin>hdfs dfs -get \temp\sample.txt E:\Desktop\temp
```

```
c:\hadoop_new\sbin>hdfs dfs -put E:\Desktop\temp \temp
```

```
c:\hadoop_new\sbin>hdfs dfs -ls \temp
```

Found 2 items

```
-rw-r--r--  1 Admin supergroup      11 2021-06-11 21:12 /temp/sample.txt
drwxr-xr-x  - Admin supergroup      0 2021-06-11 21:15 /temp/temp
```

```
c:\hadoop_new\sbin>hdfs dfs -mv \lab1 \temp
```

```
c:\hadoop_new\sbin>hdfs dfs -ls \temp
```

Found 3 items

```
drwxr-xr-x  - Admin supergroup      0 2021-04-19 15:07 /temp/lab1
-rw-r--r--  1 Admin supergroup      11 2021-06-11 21:12 /temp/sample.txt
drwxr-xr-x  - Admin supergroup      0 2021-06-11 21:15 /temp/temp
```

```
c:\hadoop_new\sbin>hdfs dfs -rm /temp/sample.txt
```

Deleted /temp/sample.txt

```
c:\hadoop_new\sbin>hdfs dfs -ls \temp
```

Found 2 items

```
drwxr-xr-x - Admin supergroup      0 2021-04-19 15:07 /temp/lab1
drwxr-xr-x - Admin supergroup      0 2021-06-11 21:15 /temp/temp
```

```
c:\hadoop_new\sbin>hdfs dfs -copyFromLocal E:\Desktop\sample.txt \temp
```

```
c:\hadoop_new\sbin>hdfs dfs -ls \temp
```

Found 3 items

```
drwxr-xr-x - Admin supergroup      0 2021-04-19 15:07 /temp/lab1
-rw-r--r--  1 Admin supergroup     11 2021-06-11 21:17 /temp/sample.txt
drwxr-xr-x - Admin supergroup      0 2021-06-11 21:15 /temp/temp
```

```
c:\hadoop_new\sbin>hdfs dfs -copyToLocal \temp\sample.txt E:\Desktop\sample.txt
```

SCREENSHOTS:

```
c:\hadoop_new\sbin>hdfs dfs -mkdir /temp
c:\hadoop_new\sbin>hdfs dfs -copyFromLocal E:\Desktop\sample.txt \temp
c:\hadoop_new\sbin>hdfs dfs -ls \temp
Found 1 items
-rw-r--r--  1 Admin supergroup     11 2021-06-11 21:12 /temp/sample.txt
c:\hadoop_new\sbin>hdfs dfs -cat \temp\sample.txt
hello world
c:\hadoop_new\sbin>hdfs dfs -get \temp\sample.txt E:\Desktop\temp
c:\hadoop_new\sbin>hdfs dfs -put E:\Desktop\temp \temp
c:\hadoop_new\sbin>hdfs dfs -ls \temp
Found 2 items
-rw-r--r--  1 Admin supergroup     11 2021-06-11 21:12 /temp/sample.txt
drwxr-xr-x - Admin supergroup      0 2021-06-11 21:15 /temp/temp
```

# MAPREDUCE TEMPERATURE

Date - 10/05/2021

For the given file, Create a Map Reduce program to

a) Find the average temperature for each year from the NCDC data set.

```
// AverageDriver.java
```

```
package temperature;
```

```
import org.apache.hadoop.io.*;
```

```
import org.apache.hadoop.fs.*;
```

```
import org.apache.hadoop.mapreduce.*;
```

```
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);

    }
}

```

//AverageMapper.java

```
package temperature;
```

```
import org.apache.hadoop.io.*;
```

```
import org.apache.hadoop.mapreduce.*;
```

```
import java.io.IOException;
```

```
public class AverageMapper extends Mapper <LongWritable, Text, Text, IntWritable>
```

```
{
```

```
    public static final int MISSING = 9999;
```

```
public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException
```

```
{
    String line = value.toString();
    String year = line.substring(15,19);
    int temperature;
    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 != MISSING && quality.matches("[01459]"))
        context.write(new Text(year),new IntWritable(temperature));
    }
}
```

```
//AverageReducer.java
```

```
package temperature;
```

```
import org.apache.hadoop.io.IntWritable;
```

```
import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapreduce.*;
```

```
import java.io.IOException;
```

```

public class AverageReducer extends Reducer <Text, IntWritable,Text, IntWritable>
{
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException,InterruptedException
    {
        int max_temp = 0;

        int count = 0;

        for (IntWritable value : values)
        {
            max_temp += value.get();

            count+=1;

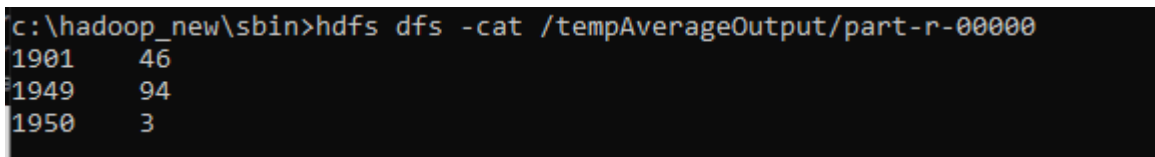
        }

        context.write(key, new IntWritable(max_temp/count));

    }
}

```

SCREENSHOT –



```

c:\hadoop_new\sbin>hdfs dfs -cat /tempAverageOutput/part-r-00000
1901    46
1949    94
1950     3

```

b) Find the mean max temperature for every month.

```
//TempDriver.java
```

```
package temperatureMax;
```



```
import org.apache.hadoop.io.*;

import org.apache.hadoop.fs.*;

import org.apache.hadoop.mapreduce.*;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


public class TempDriver
{

    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(TempDriver.class);

        job.setJobName("Max temperature");

        FileInputFormat.addInputPath(job,new Path(args[0]));

        FileOutputFormat.setOutputPath(job,new Path (args[1]));


        job.setMapperClass(TempMapper.class);

        job.setReducerClass(TempReducer.class);
```

```

        job.setOutputKeyClass(Text.class);

        job.setOutputValueClass(IntWritable.class);

        System.exit(job.waitForCompletion(true)?0:1);

    }

}

```

//TempMapper.java

```
package temperatureMax;
```

```
import org.apache.hadoop.io.*;
```

```
import org.apache.hadoop.mapreduce.*;
```

```
import java.io.IOException;
```

```
public class TempMapper extends Mapper <LongWritable, Text, Text, IntWritable>
```

```
{
```

```
    public static final int MISSING = 9999;
```

```
    public void map(LongWritable key, Text value, Context context) throws IOException,
        InterruptedException
```

```
{
```

```
        String line = value.toString();
```

```
        String month = line.substring(19,21);
```

```
        int temperature;
```

```
        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 != MISSING && quality.matches("[01459]"))

        context.write(new Text(month),new IntWritable(temperature));

    }
}

```

//TempReducer.java

```
package temperatureMax;
```

```
import org.apache.hadoop.io.*;
```

```
import org.apache.hadoop.mapreduce.*;
```

```
import java.io.IOException;
```

```
public class TempMapper extends Mapper <LongWritable, Text, Text, IntWritable>
```

```
{
```

```
    public static final int MISSING = 9999;
```

```
    public void map(LongWritable key, Text value, Context context) throws IOException,
        InterruptedException
```

```
{
```

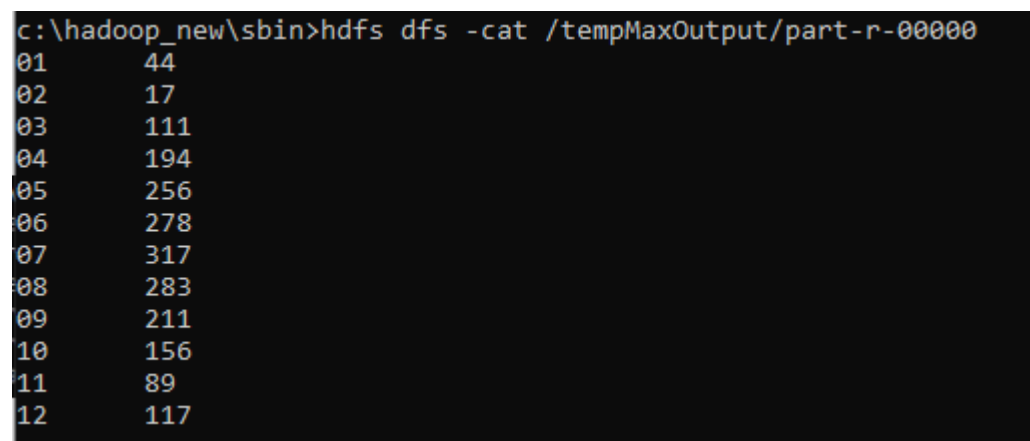
```
    String line = value.toString();
```

```
    String month = line.substring(19,21);
```

```
    int temperature;
```

```
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 != MISSING && quality.matches("[01459]"))  
  
    context.write(new Text(month),new IntWritable(temperature));  
  
}  
  
}
```

SCREENSHOT -



A screenshot of a terminal window with a black background and yellow text. The command `c:\hadoop_new\sbin>hdfs dfs -cat /tempMaxOutput/part-r-00000` has been executed. The output displays a list of 12 lines, each containing a line number followed by a temperature value.

Line Number	Temperature
01	44
02	17
03	111
04	194
05	256
06	278
07	317
08	283
09	211
10	156
11	89
12	117

# MAPREDUCE TOPN

Date - 03/05/2021

For a given Text file, create a Map Reduce program to sort the content in an alphabetic order listing only top 'n' maximum occurrence of words.

```
// TopN.java
```

```
package sortWords;
```

```
import org.apache.hadoop.conf.Configuration;
```

```
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.Mapper;
```

```
import org.apache.hadoop.mapreduce.Reducer;
```

```
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
```

```
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

```
import org.apache.hadoop.util.GenericOptionsParser;
```

```
import utils.MiscUtils;
```

```
import java.io.IOException;
```

```
import java.util.*;
```

```
public class TopN {
```

```

public static void main(String[] args) throws Exception {

    Configuration conf = new Configuration();

    String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();

    if (otherArgs.length != 2) {

        System.err.println("Usage: TopN <in> <out>");

        System.exit(2);

    }

    Job job = Job.getInstance(conf);

    job.setJobName("Top N");

    job.setJarByClass(TopN.class);

    job.setMapperClass(TopNMapper.class);

    //job.setCombinerClass(TopNReducer.class);

    job.setReducerClass(TopNReducer.class);

    job.setOutputKeyClass(Text.class);

    job.setOutputValueClass(IntWritable.class);

    FileInputFormat.addInputPath(job, new Path(otherArgs[0]));

    FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));

    System.exit(job.waitForCompletion(true) ? 0 : 1);

}

/**

    * The mapper reads one line at the time, splits it into an array of single words and emits
    every

    * word to the reducers with the value of 1.

```

```

*/

public static class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {

    private final static IntWritable one = new IntWritable(1);

    private Text word = new Text();

    private String tokens = "[_!$#<>\\^=\\[\\]\\*^\\\\\\,;\\.\\|-:()?!\"'"]";

    @Override

    public void map(Object key, Text value, Context context) throws IOException,
InterruptedException {

        String cleanLine = value.toString().toLowerCase().replaceAll(tokens, " ");

        StringTokenizer itr = new StringTokenizer(cleanLine);

        while (itr.hasMoreTokens()) {

            word.set(itr.nextToken().trim());

            context.write(word, one);

        }

    }

}

/**

 * The reducer retrieves every word and puts it into a Map: if the word already exists in the

 * map, increments its value, otherwise sets it to 1.

 */

public static class TopNReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

```

```
private Map<Text, IntWritable> countMap = new HashMap<>();
```

```
@Override
```

```
public void reduce(Text key, Iterable<IntWritable> values, Context context) throws  
IOException, InterruptedException {
```

```
    // computes the number of occurrences of a single word
```

```
    int sum = 0;
```

```
    for (IntWritable val : values) {
```

```
        sum += val.get();
```

```
    }
```

```
    // puts the number of occurrences of this word into the map.
```

```
    // We need to create another Text object because the Text instance
```

```
    // we receive is the same for all the words
```

```
    countMap.put(new Text(key), new IntWritable(sum));
```

```
}
```

```
@Override
```

```
protected void cleanup(Context context) throws IOException, InterruptedException {
```

```
    Map<Text, IntWritable> sortedMap = MiscUtils.sortByValues(countMap);
```

```
    int counter = 0;
```

```
    for (Text key : sortedMap.keySet()) {
```



```

        if (counter++ == 3) {

            break;

        }

        context.write(key, sortedMap.get(key));

    }

}

}

```

```

/**

```

\* The combiner retrieves every word and puts it into a Map: if the word already exists in the

\* map, increments its value, otherwise sets it to 1.

```

*/

```

```

public static class TopNCombiner extends Reducer<Text, IntWritable, Text, IntWritable>
{

```

```

    @Override

```

```

    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException, InterruptedException {

```

```

        // computes the number of occurrences of a single word

```

```

        int sum = 0;

```

```

        for (IntWritable val : values) {

```

```

            sum += val.get();

```

```

        }

```

```
        context.write(key, new IntWritable(sum));
    }
}
}
```

```
// MiscUtils.java
```

```
package utils;
```

```
import java.util.*;
```

```
public class MiscUtils {
```

```
    /**
```

```
     * sorts the map by values. Taken from:
```

```
     * http://javarevisited.blogspot.it/2012/12/how-to-sort-hashmap-java-by-key-and-value.html
```

```
     */
```

```
    public static <K extends Comparable, V extends Comparable> Map<K, V>
    sortByValues(Map<K, V> map) {
```

```
        List<Map.Entry<K, V>> entries = new LinkedList<Map.Entry<K,
        V>>(map.entrySet());
```

```
        Collections.sort(entries, new Comparator<Map.Entry<K, V>>() {
```

```
            @Override
```

```

        public int compare(Map.Entry<K, V> o1, Map.Entry<K, V> o2) {

            return o2.getValue().compareTo(o1.getValue());

        }

    });

    //LinkedHashMap will keep the keys in the order they are inserted

    //which is currently sorted on natural ordering

    Map<K, V> sortedMap = new LinkedHashMap<K, V>();

    for (Map.Entry<K, V> entry : entries) {

        sortedMap.put(entry.getKey(), entry.getValue());

    }

    return sortedMap;

}

}

```

SCREENSHOTS:

```
hadoop@ubuntu: ~/hadoop-3.2.1/sbin
bash: export: 'HADOOP_OPTS=Djava.library.path=/home/hadoop/hadoop-3.2.1/lib/nativ': not a valid identifier
hadoop@ubuntu:~/hadoop-3.2.1/sbin$ ./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 2563. Stop it first.
Starting datanodes
localhost: datanode is running as process 2703. Stop it first.
Starting secondary namenodes [ubuntu]
ubuntu: secondarynamenode is running as process 2903. Stop it first.
2021-06-13 19:31:44,466 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting resource manager
resource manager is running as process 3176. Stop it first.
localhost: nodemanager is running as process 3303. Stop it first.
hadoop@ubuntu:~/hadoop-3.2.1/sbin$ jps
2563 NameNode
6900 Jps
2903 SecondaryNameNode
3303 NodeManager
3176 ResourceManager
2703 DataNode
hadoop@ubuntu:~/hadoop-3.2.1/sbin$ hdfs fs -ls /
2021-06-13 19:32:16,116 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 4 items
-rw-r--r-- 1 hadoop supergroup 18 2021-06-13 13:20 /dest.txt
drwxr-xr-x 1 hadoop supergroup 0 2021-06-13 18:43 /output_tmp
drwxr-xr-x 1 hadoop supergroup 0 2021-06-13 18:38 /tmp
drwx----- 1 hadoop supergroup 0 2021-06-13 18:37 /tmp
hadoop@ubuntu:~/hadoop-3.2.1/sbin$ hadoop jar /home/hadoop/Desktop/TopM.jar /tmp/test.txt /output_6
2021-06-13 19:35:25,168 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2021-06-13 19:35:26,789 INFO client.RMProxy: Connecting to Resource Manager at /127.0.0.1:8032
2021-06-13 19:35:27,831 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1623633797667_0003
2021-06-13 19:35:28,104 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhost/trusted = false, remotenottrusted = false
2021-06-13 19:35:28,440 INFO mapreduce.JobSubmitter: Cleaning up the staging area /tmp/hadoop-yarn/staging/hadoop/.staging/job_1623633797667_0003
Exception in thread "main" org.apache.hadoop.mapreduce.lib.input.InvalidInputException: Input path does not exist: hdfs://localhost:9000/tmp/test.txt
at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.singleThreadedListStatus(FileInputFormat.java:332)
at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.listStatus(FileInputFormat.java:274)
at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.getSplits(FileInputFormat.java:396)
at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.getSplits(FileInputFormat.java:310)
at org.apache.hadoop.mapreduce.JobSubmitter.writeNewSplits(JobSubmitter.java:310)
at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitter.java:200)
at org.apache.hadoop.mapreduce.Job$1.run(Job.java:1570)
at org.apache.hadoop.mapreduce.Job$1.run(Job.java:1567)
at java.security.AccessController.doPrivileged(Native Method)

Reduce input records=20
Reduce output records=10
Spilled Records=40
Shuffled Maps=1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=371
CPU time spent (ms)=3320
Physical memory (bytes) snapshot=435990528
Virtual memory (bytes) snapshot=585911960
Total committed heap usage (bytes)=365420736
Peak Map Physical memory (bytes)=261833984
Peak Map Virtual memory (bytes)=2525442048
Peak Reduce Physical memory (bytes)=174266544
Peak Reduce Virtual memory (bytes)=2533670912

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=49
File Output Format Counters
Bytes Written=49
hadoop@ubuntu:~/hadoop-3.2.1/sbin$ hdfs fs -ls /output_6
2021-06-13 19:41:11,881 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 hadoop supergroup 0 2021-06-13 19:40 /output_6/_SUCCESS
-rw-r--r-- 1 hadoop supergroup 60 2021-06-13 19:40 /output_6/part-r-00000
hadoop@ubuntu:~/hadoop-3.2.1/sbin$ hdfs fs -cat /output/part-r-00000
2021-06-13 19:41:46,465 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
cat: /output/part-r-00000: No such file or directory
hadoop@ubuntu:~/hadoop-3.2.1/sbin$ hdfs fs -cat /output_6/part-r-00000
2021-06-13 19:42:00,353 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2021-06-13 19:42:02,331 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhost/trusted = false, remotenottrusted = false
how 5
your 4
ls 4
brother 1
are 1
ht 1
sister 1
family 1
you 1
job 1
```

## MAPREDUCE JOIN

Date - 31/05/2021

Create a Hadoop Map Reduce program to combine information from the users file along with Information from the posts file by using the concept of join and display user\_id, Reputation and Score.

// JoinDriver.java

```

import org.apache.hadoop.conf.Configured;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.*;

import org.apache.hadoop.mapred.lib.MultipleInputs;

import org.apache.hadoop.util.*;


public class JoinDriver extends Configured implements Tool {


    public static class KeyPartitioner implements Partitioner<TextPair, Text> {

        @Override

        public void configure(JobConf job) {}


        @Override

        public int getPartition(TextPair key, Text value, int numPartitions) {

            return (key.getFirst().hashCode() & Integer.MAX_VALUE) %
numPartitions;

        }

    }


    @Override

    public int run(String[] args) throws Exception {

        if (args.length != 3) {

```

```
        System.out.println("Usage: <Department Emp Strength input>  
<Department Name input> <output>");  
  
        return -1;  
  
    }
```

```
    JobConf conf = new JobConf(getConf(), getClass());  
  
    conf.setJobName("Join 'Department Emp Strength input' with 'Department  
Name input'");
```

```
    Path AInputPath = new Path(args[0]);  
  
    Path BInputPath = new Path(args[1]);  
  
    Path outputPath = new Path(args[2]);
```

```
    MultipleInputs.addInputPath(conf, AInputPath, TextInputFormat.class,  
Posts.class);
```

```
    MultipleInputs.addInputPath(conf, BInputPath, TextInputFormat.class,  
User.class);
```

```
    FileOutputFormat.setOutputPath(conf, outputPath);
```

```
    conf.setPartitionerClass(KeyPartitioner.class);
```

```
    conf.setOutputValueGroupingComparator(TextPair.FirstComparator.class);
```

```
    conf.setMapOutputKeyClass(TextPair.class);
```

```

        conf.setReducerClass(JoinReducer.class);

        conf.setOutputKeyClass(Text.class);

        JobClient.runJob(conf);

        return 0;
    }

    public static void main(String[] args) throws Exception {

        int exitCode = ToolRunner.run(new JoinDriver(), args);

        System.exit(exitCode);
    }
}

// JoinReducer.java

import java.io.IOException;

import java.util.Iterator;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.*;

public class JoinReducer extends MapReduceBase implements Reducer<TextPair, Text,
Text, Text> {

```

```

@Override

    public void reduce (TextPair key, Iterator<Text> values, OutputCollector<Text,
Text> output, Reporter reporter)

        throws IOException

    {

        Text nodeId = new Text(values.next());

        while (values.hasNext()) {

            Text node = values.next();

            Text outValue = new Text(nodeId.toString() + "\t\t" + node.toString());

            output.collect(key.getFirst(), outValue);

        }

    }
}

```

```

// User.java

import java.io.IOException;

import java.util.Iterator;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.FSDataInputStream;

import org.apache.hadoop.fs.FSDataOutputStream;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.LongWritable;

```



```

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.*;

import org.apache.hadoop.io.IntWritable;

public class User extends MapReduceBase implements Mapper<LongWritable, Text,
TextPair, Text> {

    @Override

    public void map(LongWritable key, Text value, OutputCollector<TextPair, Text>
output, Reporter reporter)

        throws IOException

    {

        String valueString = value.toString();

        String[] SingleNodeData = valueString.split("\t");

        output.collect(new TextPair(SingleNodeData[0], "1"), new
Text(SingleNodeData[1]));

    }

}

```

//Posts.java

```

import java.io.IOException;

import org.apache.hadoop.io.*;

```

```
import org.apache.hadoop.mapred.*;
```

```
public class Posts extends MapReduceBase implements Mapper<LongWritable, Text,  
TextPair, Text> {
```

```
    @Override
```

```
    public void map(LongWritable key, Text value, OutputCollector<TextPair, Text>  
output, Reporter reporter)
```

```
        throws IOException
```

```
    {
```

```
        String valueString = value.toString();
```

```
        String[] SingleNodeData = valueString.split("\t");
```

```
        output.collect(new TextPair(SingleNodeData[3], "0"), new  
Text(SingleNodeData[9]));
```

```
    }
```

```
}
```

```
// TextPair.java
```

```
import java.io.*;
```

```
import org.apache.hadoop.io.*;
```

```
public class TextPair implements WritableComparable<TextPair> {
```

```
    private Text first;
```

```
private Text second;
```

```
public TextPair() {  
    set(new Text(), new Text());  
}
```

```
public TextPair(String first, String second) {  
    set(new Text(first), new Text(second));  
}
```

```
public TextPair(Text first, Text second) {  
    set(first, second);  
}
```

```
public void set(Text first, Text second) {  
    this.first = first;  
    this.second = second;  
}
```

```
public Text getFirst() {  
    return first;  
}
```

```
public Text getSecond() {
```

```
    return second;
}
```

```
@Override

public void write(DataOutput out) throws IOException {

    first.write(out);

    second.write(out);

}
```

```
@Override

public void readFields(DataInput in) throws IOException {

    first.readFields(in);

    second.readFields(in);

}
```

```
@Override

public int hashCode() {

    return first.hashCode() * 163 + second.hashCode();

}
```

```
@Override

public boolean equals(Object o) {

    if (o instanceof TextPair) {

        TextPair tp = (TextPair) o;
```

```
        return first.equals(tp.first) && second.equals(tp.second);
    }

    return false;
}
```

@Override

```
public String toString() {
    return first + "\t" + second;
}
```

@Override

```
public int compareTo(TextPair tp) {
    int cmp = first.compareTo(tp.first);
    if (cmp != 0) {
        return cmp;
    }
    return second.compareTo(tp.second);
}
```

// ^^ TextPair

// vv TextPairComparator

```
public static class Comparator extends WritableComparator {
```

```
    private static final Text.Comparator TEXT_COMPARATOR = new Text.Comparator();
```

```

public Comparator() {

    super(TextPair.class);

}

@Override

public int compare(byte[] b1, int s1, int l1,

                    byte[] b2, int s2, int l2) {

    try {

        int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1);

        int firstL2 = WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2);

        int cmp = TEXT_COMPARATOR.compare(b1, s1, firstL1, b2, s2, firstL2);

        if (cmp != 0) {

            return cmp;

        }

        return TEXT_COMPARATOR.compare(b1, s1 + firstL1, l1 - firstL1,

                                        b2, s2 + firstL2, l2 - firstL2);

    } catch (IOException e) {

        throw new IllegalArgumentException(e);

    }

}
}

```

```

static {

    WritableComparator.define(TextPair.class, new Comparator());

}

public static class FirstComparator extends WritableComparator {

    private static final Text.Comparator TEXT_COMPARATOR = new Text.Comparator();

    public FirstComparator() {

        super(TextPair.class);

    }

    @Override

    public int compare(byte[] b1, int s1, int l1,

        byte[] b2, int s2, int l2) {

        try {

            int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1);

            int firstL2 = WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2);

            return TEXT_COMPARATOR.compare(b1, s1, firstL1, b2, s2, firstL2);

        } catch (IOException e) {

            throw new IllegalArgumentException(e);

        }

    }

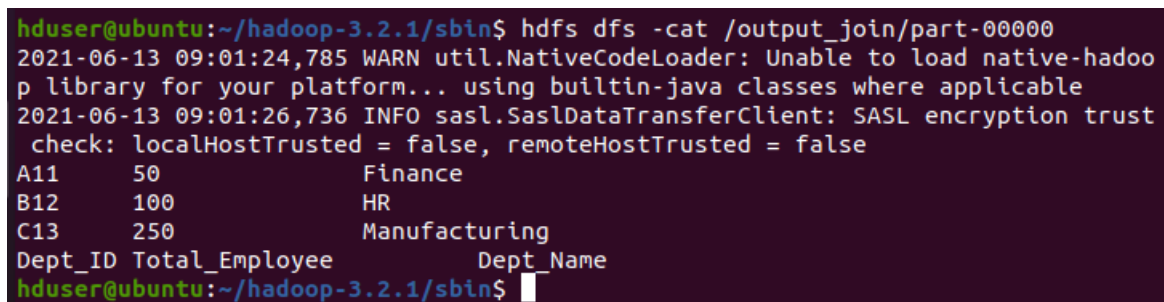
}

```

@Override

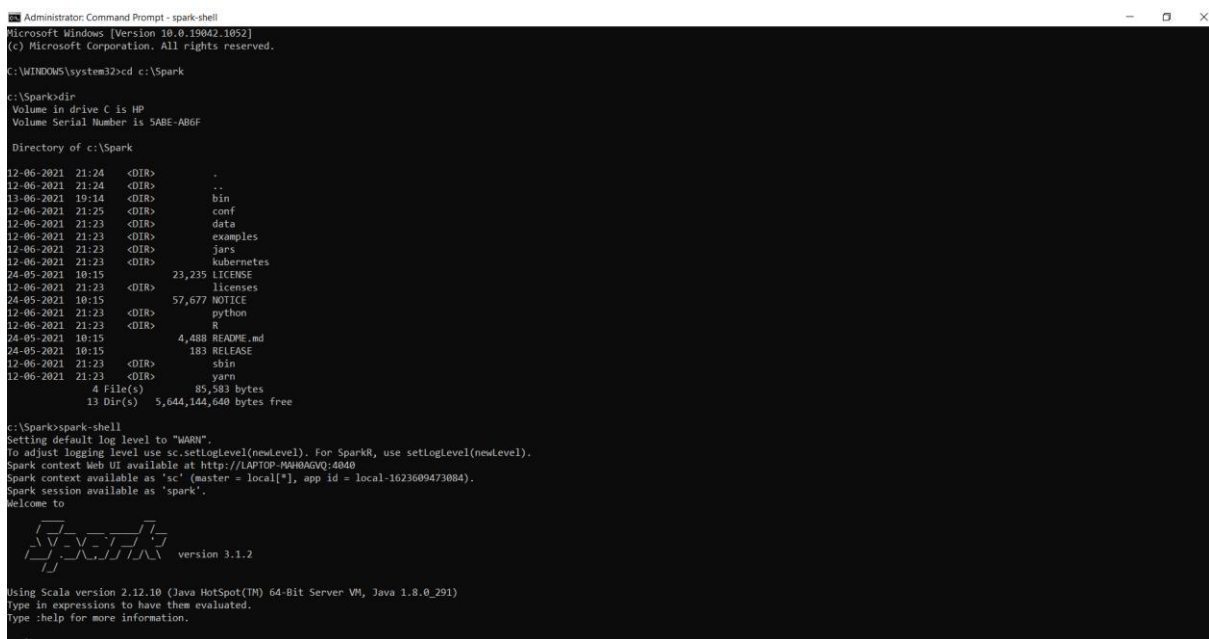
```
public int compare(WritableComparable a, WritableComparable b) {  
  
    if (a instanceof TextPair && b instanceof TextPair) {  
  
        return ((TextPair) a).first.compareTo(((TextPair) b).first);  
  
    }  
  
    return super.compare(a, b);  
  
}  
  
}  
  
}
```

SCREENSHOTS –



```
hduser@ubuntu:~/hadoop-3.2.1/sbin$ hdfs dfs -cat /output_join/part-00000  
2021-06-13 09:01:24,785 WARN util.NativeCodeLoader: Unable to load native-hadoop  
p library for your platform... using builtin-java classes where applicable  
2021-06-13 09:01:26,736 INFO sasl.SaslDataTransferClient: SASL encryption trust  
check: localhostTrusted = false, remoteHostTrusted = false  
A11      50      Finance  
B12      100     HR  
C13      250     Manufacturing  
Dept_ID Total_Employee      Dept_Name  
hduser@ubuntu:~/hadoop-3.2.1/sbin$
```

## SCALA INSTALLATION SCREENSHOT



```
Administrator: Command Prompt - spark-shell  
Microsoft Windows [Version 10.0.19042.1052]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\WINDOWS\system32>cd c:\Spark  
  
c:\Spark>dir  
Volume in drive C is HP  
Volume Serial Number is 5ABE-AB6F  
  
Directory of c:\Spark  
  
12-06-2021 21:24 <DIR> .  
12-06-2021 21:24 <DIR> ..  
13-06-2021 19:14 <DIR> bin  
12-06-2021 21:25 <DIR> conf  
12-06-2021 21:23 <DIR> data  
12-06-2021 21:23 <DIR> examples  
12-06-2021 21:23 <DIR> jars  
12-06-2021 21:23 <DIR> kubernetes  
24-05-2021 10:15 23,235 LICENSE  
12-06-2021 21:23 <DIR> licenses  
24-05-2021 10:15 57,677 NOTICE  
12-06-2021 21:23 <DIR> python  
12-06-2021 21:23 <DIR> R  
24-05-2021 10:15 4,488 README.md  
24-05-2021 10:15 183 RELEASE  
12-06-2021 21:23 <DIR> sbin  
12-06-2021 21:23 <DIR> yarn  
4 File(s) 85,582 bytes  
13 Dir(s) 5,644,144,640 bytes free  
  
c:\Spark>spark-shell  
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://LAPTOP-MAHAGVQ:4040  
Spark context available as 'sc' (master = local[*], app id = local-1623609473884).  
Spark session available as 'spark'.  
Welcome to  
  
version 3.1.2  
  
Using Scala version 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_291)  
Type in expressions to have them evaluated.  
Type :help for more information.  
  
scala>
```



# SCALA WORDCOUNT

Date - 07/06/2021

```
package count
import org.apache.spark.SparkConf
```

```
import org.apache.spark.SparkContext
```

```
import org.apache.spark.rdd.RDD.rddToPairRDDFunctions
```

```
object count {
```

```
  def
  main(args: Array[String]) = {
```

```
    //Start the Spark context
```

```
    val conf = new SparkConf()
```

```
    .setAppName("count")
```

```
    .setMaster("local")
```

```
    val sc = new SparkContext(conf)
```

```
    //Read some example file to a test RDD
```

```
val test = sc.textFile("C:\\Spark\\spark-2.4.8-bin-hadoop2.7\\bin\\testdata\\sparkdata.txt")
```

```
test.flatMap {  
  line => //for each line
```

```
  line.split(" ") //split the line in word by word.
```

```
}
```

```
  .map {  
    word => //for each word
```

```
      (word, 1) //Return a key/value tuple, with the word as key and 1 as value
```

```
}
```

```
  .reduceByKey(_ + _) //Sum all of the value same key
```

```
  .saveAsTextFile("C:\\Spark\\spark-2.4.8-bin-hadoop2.7\\bin\\testdata\\output2.txt") //Save to a  
  text file
```

```
//Stop the Spark context
```

```
sc.stop}}
```

## SCREENSHOTS:

```
(are,1)
(sister?,1)
(is,2)
(you,1)
(jib?,1)
(hi,1)
(have,1)
(how,4)
(you?,1)
(,4)
(been?,1)
(your,2)
```

```
scala> val data=sc.textFile("C:\\Spark\\spark-2.4.8-bin-hadoop2.7\\bin\\testdata\\sparkdata.txt")
data: org.apache.spark.rdd.RDD[String] = C:\Spark\spark-2.4.8-bin-hadoop2.7\bin\testdata\sparkdata.txt MapPartitionsRDD[61] at textFile at <console>:24

scala> data.collect;
res31: Array[String] = Array(hi how are you?, how is your sister?, how is your jib?, how have you been?, "", "", "", "")

scala> val splitdata = data.flatMap(line => line.split(" "));
splitdata: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[62] at flatMap at <console>:25

scala> splitdata.collect;
res32: Array[String] = Array(hi, how, are, you?, how, is, your, sister?, how, is, your, jib?, how, have, you, been?, "", "", "", "")

scala> val mapdata = splitdata.map(word => (word,1));
mapdata: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[63] at map at <console>:25

scala> mapdata.collect;
res33: Array[(String, Int)] = Array((hi,1), (how,1), (are,1), (you?,1), (how,1), (is,1), (your,1), (sister?,1), (how,1), (is,1), (your,1), (jib?,1), (how,1), (have,1), (you,1), (been?,1), ("",1), ("",1), ("",1), ("",1))

scala> val reducedata = mapdata.reduceByKey(_+_);
reducedata: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[64] at reduceByKey at <console>:25

scala> reducedata.collect;
res34: Array[(String, Int)] = Array((are,1), (is,2), (jib?,1), (have,1), (how,4), (you?,1), ("",4), (sister?,1), (you,1), (hi,1), (been?,1), (your,2))
```