VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

Big Data Analytics Lab(20CS6PEBDA)

Submitted by

Supriya M Lingdal (1BM20CS415)

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

April-2022 to July-202

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 Lab(20CS6PEBDA)** carried out by **Supriya M Lingdal(1BM20CS415),** 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 academic year 2021-2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Big Data Analytics Lab(20CS6PEBDA)** work prescribed for the said degree.

Dr. Shyamala G Dr. Jyothi S Nayak

Professor and Head

Department of CSE Department of CSE

BMSCE, Bengaluru BMSCE, Bengaluru

`

Index Sheet

|  |  |  |
| --- | --- | --- |
| Sl.No. | Experiment Title | Page No. |
| 1 | MONGO DB- Employee Database |  |
| 2 | MONGO DB- Student Database |  |
| 3 | CASANDRA-Student Database |  |
| 4 | CASANDRA-Employee Database |  |
| 5 | HDFS |  |
| 6 | HADOOP -TopN |  |
| 7 | HADOOP -Wordcount |  |
| 8 | SPARK-Scala (10 programs) |  |

Course Outcome

**Ability to conduct practical experiment to solve a given problem using Unified Modeling language.**

**CO4**

**MONGO DB**

**Queries executed on Employee Database using Mongo db**

bmsce@bmsce-Precision-T1700:~$ mongo

MongoDB shell version v3.6.8

connecting to: mongodb://127.0.0.1:27017

Implicit session: session { "id" : UUID("d66acdb3-8482-417d-8b75-d65dae4b53ee") }

MongoDB server version: 3.6.8

Server has startup warnings:

2022-04-11T18:49:15.627+0530 I STORAGE [initandlisten]

2022-04-11T18:49:15.627+0530 I STORAGE [initandlisten] \*\* WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine

2022-04-11T18:49:15.627+0530 I STORAGE [initandlisten] \*\* See http://dochub.mongodb.org/core/prodnotes-filesystem

2022-04-11T18:49:18.771+0530 I CONTROL [initandlisten]

2022-04-11T18:49:18.771+0530 I CONTROL [initandlisten] \*\* WARNING: Access control is not enabled for the database.

2022-04-11T18:49:18.771+0530 I CONTROL [initandlisten] \*\* Read and write access to data and configuration is unrestricted.

2022-04-11T18:49:18.771+0530 I CONTROL [initandlisten]

**Creation of Database**

use Employee

switched to db Employee

**Creation of Table**

db.createCollection("Employee");

{ "ok" : 1 }

**Display Collections**

show collections;

Employee

**Inserting values in to table**

db.Employee.insert({Emp\_id:1,Name:"Ravi",Designation:"Manager",Experience:"3\_years"});

WriteResult({ "nInserted" : 1 })

db.Employee.insertMany([

{Emp\_id:2,Name:"Raj",Designation:"HR",Experience:"5\_years" },

{Emp\_id:6,Name:"Rekha",Designation:"CEO",Experience:"10\_years" }

]);

WriteResult({ "nInserted" : 1 })

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("62a21013cb0f317b8d1eed64"),

ObjectId("62a21013cb0f317b8d1eed65")

}

**Use of find()**

db.Employee.find();

{ "\_id" : ObjectId("62a20f4e5ac3ac8c60471ff7"), "Emp\_id" : 1, "Name" : "Ravi", "Designation" : "Manager", "Experience" : "3\_years" }

{ "\_id" : ObjectId("62a20fe6b95714dc08dab8a4"), "Emp\_id" : 2, "Name" : "Raj", "Designation" : "HR", "Experience" : "5\_years" }

{ "\_id" : ObjectId("62a20fe6b95714dc08dab8a5"), "Emp\_id" : 6, "Name" : "Rekha", "Designation" : "CEO", "Experience" : "10\_years" }

db.Employee.update({Emp\_id:2,Name:"Raj"},{$set:{Designation:"Administrator"}},{upsert:true});

{ "\_id" : ObjectId("62a210fa974e548155150b49"), "Emp\_id" : 6, "Name" : "Rekha", "Designation" : "CEO", "Experience" : "10\_years" }

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

db.Employee.find({Name:"Rekha"});

{ "\_id" : ObjectId("62a211bdaeca4b0619a5e33e"), "Emp\_id" : 6, "Name" : "Rekha", "Designation" : "CEO", "Experience" : "10\_years" }

**Use of Count()**

db.Employee.count();

3

**Use of limit()**

db.Employee.find().limit(2);

{ "\_id" : ObjectId("62a20f4e5ac3ac8c60471ff7"), "Emp\_id" : 1, "Name" : "Ravi", "Designation" : "Manager", "Experience" : "3\_years" }

{ "\_id" : ObjectId("62a20fe6b95714dc08dab8a4"), "Emp\_id" : 2, "Name" : "Raj", "Designation" : "HR", "Experience" : "5\_years" }

db.Employee.count({Designation:"HR"});

1

**CASANDRA**

**Queries Executed on Student Database using Casandra**

**Create database/keyspace Students**

cqlsh> create keyspace Students with replication={'class':'SimpleStrategy','replication\_factor':2};

cqlsh> describe keyspaces;

students system\_schema system\_auth system system\_distributed system\_traces

cqlsh> select \*from system.schema\_keyspaces;

InvalidRequest: Error from server: code=2200 [Invalid query] message="unconfigured table schema\_keyspaces"

cqlsh> use students;

cqlsh:students> create table Student\_Info(

... RollNo int PRIMARY KEY,

... StuName text,

... DateOfjoining timestamp,

... lastExamPercent double

... );

cqlsh:students> describe tables;

student\_info

cqlsh:students> describe table student\_info;

**Creating Table student\_info**

CREATE TABLE students.student\_info (

rollno int PRIMARY KEY,

dateofjoining timestamp,

lastexampercent double,

stuname text

) WITH bloom\_filter\_fp\_chance = 0.01

AND caching = {'keys': 'ALL', 'rows\_per\_partition': 'NONE'}

AND comment = ''

AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max\_threshold': '32', 'min\_threshold': '4'}

ANDcompression={'chunk\_length\_in\_kb':'64','class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

AND crc\_check\_chance = 1.0

AND dclocal\_read\_repair\_chance = 0.1

AND default\_time\_to\_live = 0

AND gc\_grace\_seconds = 864000

AND max\_index\_interval = 2048

AND memtable\_flush\_period\_in\_ms = 0

AND min\_index\_interval = 128

AND read\_repair\_chance = 0.0

AND speculative\_retry = '99PERCENTILE';

**Inserting values**

cqlsh:students> begin batch

... insert into student\_info(rollno,stuname,dateofjoining,lastexampercent) values(1,'Micheal Storm','2012-03-29',69.6)

... insert into student\_info(rollno,stuname,dateofjoining,lastexampercent) values(2,'Albert inglan','2014-07-18',70.2)

... insert into student\_info(rollno,stuname,dateofjoining,lastexampercent) values(3,'Isha Jaganwal','2013-09-12',65.3)

... insert into student\_info(rollno,stuname,dateofjoining,lastexampercent) values(4,'Eshaan Pandey','2018-02-25',80.8)

... apply batch;

**Display contents of the Table**

cqlsh:students> select \*from student\_info;

rollno | dateofjoining | lastexampercent | stuname

--------+---------------------------------+-----------------+---------------

1 | 2012-03-28 18:30:00.000000+0000 | 69.6 | Micheal Storm

2 | 2014-07-17 18:30:00.000000+0000 | 70.2 | Albert inglan

4 | 2018-02-24 18:30:00.000000+0000 | 80.8 | Eshaan Pandey

3 | 2013-09-11 18:30:00.000000+0000 | 65.3 | Isha Jaganwal

(4 rows)

**Queries:**

1. cqlsh:students> select \*from student\_info where rollno in(2,3);

rollno | dateofjoining | lastexampercent | stuname

--------+---------------------------------+-----------------+---------------

2 | 2014-07-17 18:30:00.000000+0000 | 70.2 | Albert inglan

3 | 2013-09-11 18:30:00.000000+0000 | 65.3 | Isha Jaganwal

(2 rows)

1. cqlsh:students> create index on student\_info(stuname);

cqlsh:students> select \*from student\_info where stuname='Isha Jaganwal';

rollno | dateofjoining | lastexampercent | stuname

--------+---------------------------------+-----------------+---------------

3 | 2013-09-11 18:30:00.000000+0000 | 65.3 | Isha Jaganwal

(1 rows)

1. cqlsh:students> create index on student\_info(lastexampercent);

cqlsh:students> select \*from student\_info where lastexampercent=80.8;

rollno | dateofjoining | lastexampercent | stuname

--------+---------------------------------+-----------------+---------------

4 | 2018-02-24 18:30:00.000000+0000 | 80.8 | Eshaan Pandey

(1 rows)

1. cqlsh:students> select rollno from student\_info limit 2;

rollno

--------

1

2

(2 rows)

1. cqlsh:students> update student\_info set stuname='David' where rollno=1;

cqlsh:students> update student\_info set stuname='John',lastexampercent=90.1 where rollno=1;

cqlsh:students> delete lastexampercent from student\_info where rollno=2;

cqlsh:students> create table Project\_deatils(

... project-id int,

... project\_name text,

... stu\_name text,

... rating double,

... duration int,

... PRIMARY KEY(project-id,project\_name);

1. cqlsh:students> create table Project\_deatils(project\_id int, project\_name text, stu\_name text, rating double, duration int, PRIMARY KEY(project\_id,project\_name));

cqlsh:students> create table Project\_details(project\_id int, project\_name text, stu\_name text, rating double, duration int, PRIMARY KEY(project\_id,project\_name));

cqlsh:students> drop table project\_deatils

... ;

cqlsh:students> insert into project\_details(project\_id,project\_name,stu\_name,rating,duration) values(1,'MS data migration','David sheen',5,720) ;

1. cqlsh:students> begin batch

... insert into project\_details(project\_id,project\_name,stu\_name,rating,duration) values(2,'computer networking','John',3.5,1440)

... insert into project\_details(project\_id,project\_name,stu\_name,rating,duration) values(3,'Data warehouse','Isha Jaganwal',4.5,880)

... insert into project\_details(project\_id,project\_name,stu\_name,rating,duration) values(3,'MS Data warehouse','lingdal',5.5,550);

... apply batch;

1. cqlsh:students> select \*from project\_details;

project\_id | project\_name | duration | rating | stu\_name

------------+---------------------+----------+--------+---------------

1 | MS data migration | 720 | 5 | David sheen

2 | computer networking | 1440 | 3.5 | John

3 | Data warehouse | 880 | 4.5 | Isha Jaganwal

3 | MS Data warehouse | 550 | 5.5 | lingdal

(4 rows)

1. cqlsh:students> alter table student\_info ADD hobbies set<text>;

cqlsh:students> update student\_info set hobbies=hobbies+{'chess,table tennis'} where rollno=1;

cqlsh:students> alter table student\_info add language list<text>

... ;

1. cqlsh:students> update student\_info set language=language+['Hindi,English'] where rollno=1;

cqlsh:students> select \*from student\_info;

rollno | dateofjoining | hobbies | language | lastexampercent | stuname

--------+---------------------------------+------------------------+-------------------+-----------------+---------------

1 | 2012-03-28 18:30:00.000000+0000 | {'chess,table tennis'} | ['Hindi,English'] | 90.1 | John

2 | 2014-07-17 18:30:00.000000+0000 | null | null | null | Albert inglan

4 | 2018-02-24 18:30:00.000000+0000 | null | null | 80.8 | Eshaan Pandey

3 | 2013-09-11 18:30:00.000000+0000 | null | null | 65.3 | Isha Jaganwal

(4 rows)

1. cqlsh:students> student\_info(rollno,dateofjoining,hobbies,language,lastexampercent,stuname)copy TO 'd:\student\_details.csv';

Using 11 child processes

Starting copy of students.student\_info with columns [rollno, dateofjoining, hobbies, language, lastexampercent, stuname].

Processed: 4 rows; Rate: 33 rows/s; Avg. rate: 33 rows/s

4 rows exported to 1 files in 0.148 seconds.

1. cqlsh:students> truncate student\_info;

Copystudent\_info(rollno,dateofjoining,hobbies,language,lastexampercent,stuname)TO 'd:\student\_details.csv';

Using 11 child processes

Starting copy of students.student\_info with columns [rollno, dateofjoining, hobbies, language, lastexampercent, stuname].

Processed: 4 rows; Rate: 33 rows/s; Avg. rate: 33 rows/s

4 rows exported to 1 files in 0.148 seconds.

cqlsh:students> truncate student\_info;

cqlsh:students>

cqlsh:students> describe table student\_info;

1. CREATE TABLE students.student\_info (

rollno int PRIMARY KEY,

dateofjoining timestamp,

hobbies set<text>,

language list<text>,

lastexampercent double,

stuname text

) WITH bloom\_filter\_fp\_chance = 0.01

AND caching = {'keys': 'ALL', 'rows\_per\_partition': 'NONE'}

AND comment = ''

AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max\_threshold': '32', 'min\_threshold': '4'}

AND compression = {'chunk\_length\_in\_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

AND crc\_check\_chance = 1.0

AND dclocal\_read\_repair\_chance = 0.1

AND default\_time\_to\_live = 0

AND gc\_grace\_seconds = 864000

AND max\_index\_interval = 2048

AND memtable\_flush\_period\_in\_ms = 0

AND min\_index\_interval = 128

AND read\_repair\_chance = 0.0

AND speculative\_retry = '99PERCENTILE';

1. CREATE INDEX student\_info\_lastexampercent\_idx ON students.student\_info (lastexampercent);

CREATE INDEX student\_info\_stuname\_idx ON students.student\_info (stuname);

1. cqlsh:students>copystudent\_info(rollno,dateofjoining,hobbies,language,lastexampercent,stuname) FROM 'd:\student\_details.csv';

Using 11 child processes

Starting copy of students.student\_info with columns [rollno, dateofjoining, hobbies, language, lastexampercent, stuname].

Processed: 4 rows; Rate: 6 rows/s; Avg. rate: 9 rows/s

4 rows imported from 1 files in 0.440 seconds (0 skipped).

cqlsh:students> select \*from student\_info;

rollno | dateofjoining | hobbies | language | lastexampercent | stuname

--------+---------------------------------+------------------------+-------------------+-----------------+---------------

1 | 2012-03-28 18:30:00.000000+0000 | {'chess,table tennis'} | ['Hindi,English'] |90.1 | John

2 | 2014-07-17 18:30:00.000000+0000 | null | null | null | Albert inglan

4 | 2018-02-24 18:30:00.000000+0000 | null | null | 80.8 | Eshaan Pandey

3 | 2013-09-11 18:30:00.000000+0000 | null | null | 65.3 | Isha Jaganwal

**Queries Executed on Employee Database using Casandra**

**Creation keyspace employee**

cqlsh:employee> CREATE KEYSPACE employee WITH REPLICATION={ 'class' : 'SimpleStrategy', 'replication\_factor' : 1};

cqlsh:employee> USE employee;

cqlsh:employee> create table employee\_info(emp\_id int PRIMARY KEY, emp\_name text,

... designation text, date\_of\_joining timestamp, salary double PRIMARY KEY, dept\_name text);

**Creation of table**

cqlsh:employee> CREATE TABLE employee\_info(emp\_id int, emp\_name text, designation text, date\_of\_joining timestamp, salary double, dept\_name text, PRIMARY KEY(emp\_id, salary));

**Inserting values into table**

cqlsh:employee> BEGIN BATCH INSERT INTO

... employee\_info(emp\_id,emp\_name,designation,date\_of\_joining,salary,dept\_name)

... VALUES(100,'Raj','MANAGER','2021-09-11',30000,'TESTING');

... INSERT INTO

... employee\_info(emp\_id,emp\_name,designation,date\_of\_joining,salary,dept\_name)

... VALUES(101,'Ravi','ASSOCIATE','2021-06-22',25000,'DEVELOPING');

... INSERT INTO

... employee\_info(emp\_id,emp\_name,designation,date\_of\_joining,salary,dept\_name)

... VALUES(102,'Ajay','HR','2021-03-30',35000,'HR');

... INSERT INTO

... employee\_info(emp\_id,emp\_name,designation,date\_of\_joining,salary,dept\_name)

... VALUES(103,'Sheela','CEO','2021-12-30',20000,'DEVELOPING');

... INSERT INTO

... employee\_info(emp\_id,emp\_name,designation,date\_of\_joining,salary,dept\_name)

... VALUES(104,'Sharat','ASSOCIATE','2021-06-25',25000,'TESTING');

... APPLY BATCH;

**Display of table**

cqlsh:employee> SELECT \* FROM employee\_info;

emp\_id | salary | date\_of\_joining | dept\_name | designation | emp\_name

--------+--------+---------------------------------+------------+-------------+----------

100 | 25000 | 2021-09-11 18:30:00.000000+0000 | TESTING | MANAGER | Raj

101 | 25000 | 2021-06-22 18:30:00.000000+0000 | DEVELOPING | ASSOCIATE | Ravi

102 | 35000 | 2021-03-30 18:30:00.000000+0000 | HR | HR | Ajay

103 | 20000 | 2021-12-30 18:30:00.000000+0000 | DEVELOPING | CEO | Sheela

104 | 30000 | 2021-06-25 18:30:00.000000+0000 | TESTING | ASSOCIATE | Sharat

(5 rows)

**Queries:**

1. **Update**

cqlsh:employee> UPDATE employee\_info SET emp\_name = 'Sumith', dept\_name = 'TESTING' WHERE emp\_id = 102 AND salary = 35000;

cqlsh:employee> SELECT \* FROM employee\_info ;

emp\_id | salary | date\_of\_joining | dept\_name | designation | emp\_name

--------+--------+---------------------------------+------------+-------------+----------

100 | 25000 | 2021-09-11 18:30:00.000000+0000 | TESTING | MANAGER | Raj

101 | 25000 | 2021-06-22 18:30:00.000000+0000 | DEVELOPING | ASSOCIATE | Ravi

102 | 35000 | 2021-03-30 18:30:00.000000+0000 | HR | TESTING | Sumith

103 | 20000 | 2021-12-30 18:30:00.000000+0000 | DEVELOPING | CEO | Sheela

104 | 30000 | 2021-06-25 18:30:00.000000+0000 | TESTING | ASSOCIATE | Sharat

(5 rows)

1. **Order by**

cqlsh:employee> SELECT \* FROM employee\_info WHERE emp\_id in (100, 101, 102, 103, 104) order by salary;

emp\_id | salary | date\_of\_joining | dept\_name | designation | emp\_name

--------+--------+---------------------------------+------------+-------------+----------

103 | 20000 | 2021-12-30 18:30:00.000000+0000 | DEVELOPING | CEO | Sheela

100 | 25000 | 2021-09-11 18:30:00.000000+0000 | TESTING | MANAGER | Raj

101 | 25000 | 2021-06-22 18:30:00.000000+0000 | DEVELOPING | ASSOCIATE | Ravi

104 | 30000 | 2021-06-25 18:30:00.000000+0000 | TESTING | ASSOCIATE | Sharat

102 | 35000 | 2021-03-30 18:30:00.000000+0000 | HR | TESTING | Sumith

(5 rows)

**3)Alter**

cqlsh:employee> ALTER TABLE employee\_info ADD projects text;

cqlsh:employee> UPDATE employee\_info SET projects = 'Android' WHERE emp\_id = 101;

cqlsh:employee> UPDATE employee\_info SET projects = 'IOT' WHERE emp\_id = 102 and salary = 35000;

cqlsh:employee> UPDATE employee\_info SET projects = 'Deep learning' WHERE emp\_id = 103 and salary = 20000;

cqlsh:employee> UPDATE employee\_info SET projects = 'ML' WHERE emp\_id = 104 and salary = 30000;

cqlsh:employee> UPDATE employee\_info SET projects = 'AI' WHERE emp\_id = 100 and salary = 25000;

cqlsh:employee> SELECT \* FROM employee\_info

... ;

cqlsh:employee> SELECT \* FROM employee\_info ;

emp\_id | salary | date\_of\_joining | dept\_name | designation | emp\_name |Projects

--------+--------+---------------------------------+------------+-------------+----------------------------------

103 | 20000 | 2021-12-30 18:30:00.000000+0000 | DEVELOPING | CEO | Sheela |Deep learning

100 | 25000 | 2021-09-11 18:30:00.000000+0000 | TESTING | MANAGER | Raj |AI

101 | 25000 | 2021-06-22 18:30:00.000000+0000 | DEVELOPING | ASSOCIATE | Ravi |Android

104 | 30000 | 2021-06-25 18:30:00.000000+0000 | TESTING | ASSOCIATE | Sharat |ML

102 | 35000 | 2021-03-30 18:30:00.000000+0000 | HR | TESTING | SumitH |IOT

(5 rows)

1. **Time to live(TTL)**

cqlsh:employee>INSERTINTOemployee\_info(emp\_id,emp\_name,designation,date\_of\_joining,salary,dept\_name)VALUES(110,'SAMARTH','OWNER','2021-01-11',90000,'LEADER') USING TTL 15;

cqlsh:employee> SELECT TTL(emp\_name) from employee\_info WHERE emp\_id = 110;

ttl(emp\_name)

---------------

3

**HDFS**

hduser@bmsce-Precision-T1700:~$ start-all.sh

This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh

Starting namenodes on [localhost]

hduser@localhost's password:

localhost: namenode running as process 6088. Stop it first.

hduser@localhost's password:

localhost: datanode running as process 6269. Stop it first.

Starting secondary namenodes [0.0.0.0]

hduser@0.0.0.0's password:

0.0.0.0: secondarynamenode running as process 6487. Stop it first.

starting yarn daemons

resourcemanager running as process 6646. Stop it first.

hduser@localhost's password:

localhost: nodemanager running as process 6979. Stop it first.

hduser@bmsce-Precision-T1700:~$ jps

6979 NodeManager

6646 ResourceManager

6487 SecondaryNameNode

12184 Jps

6088 NameNode

6269 DataNode

**Creation of Directory**

hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /Supriya

mkdir: `/Supriya': File exists

hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /Student

**View list of Directories and Files**

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /

Found 12 items

drwxr-xr-x - hduser supergroup 0 2022-06-04 09:47 /Employee

drwxr-xr-x - hduser supergroup 0 2022-06-04 09:53 /Student

drwxr-xr-x - hduser supergroup 0 2022-06-04 09:52 /Supriya

drwxr-xr-x - hduser supergroup 0 2022-06-01 09:55 /cs184

drwxr-xr-x - hduser supergroup 0 2022-06-03 15:24 /dee

drwxr-xr-x - hduser supergroup 0 2022-06-03 14:51 /dishabhaskar

drwxr-xr-x - hduser supergroup 0 2022-06-03 12:21 /final\_hari

drwxr-xr-x - hduser supergroup 0 2022-06-03 12:24 /final\_nisar

drwxr-xr-x - hduser supergroup 0 2022-06-03 12:46 /final\_nisarga

drwxr-xr-x - hduser supergroup 0 2022-06-03 12:06 /hari

drwxr-xr-x - hduser supergroup 0 2022-06-03 12:38 /nisargas

drwxr-xr-x - hduser supergroup 0 2022-06-01 14:56 /yay

**Use of PUT command**

hduser@bmsce-Precision-T1700:~$ hdfs dfs -put /home/hduser/Desktop/xyz.txt /Student/srs.txt

hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /Student/srs.txt

CSE

ISE

EC

ME

EE

**Use of copyFromLocal command**

hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyFromLocal /home/hduser/Desktop/xyz.txt /Student/zyx.txt

duser@bmsce-Precision-T1700:~$ hdfs dfs -cat /Student/zyx.txt

CSE

ISE

EC

ME

EE

duser@bmsce-Precision-T1700:~$ sudo nano hadoop.txt

[sudo] password for hduser:

**Use of GET command**

duser@bmsce-Precision-T1700:~$ hdfs dfs -get /Student/srs.txt /home/hduser/Downloads/new.txt

/////new file named new.txt is created in downloads with contents of srs.txt

**Use of MERGE command**

hdfs dfs -getmerge /Student/srs.txt /Student/zyx.txt /home/hduser/Desktop/Merge.txt

//output:merge.txt created in desktop

CSE

ISE

EC

ME

EE

CSE

ISE

EC

ME

EE

hduser@bmsce-Precision-T1700:~$ hadoop fs -getfacl /Student/

# file: /Student

# owner: hduser

# group: supergroup

user::rwx

group::r-x

other::r-x

**Use of copyToLocal command**

hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Student/srs.txt /home/hduser/Desktop/sss.txt

//content of sss.txt created in desktop

CSE

ISE

EC

ME

EE

**Display contents using CAT command**

hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /Student/srs.txt

CSE

ISE

EC

ME

EE

**HADOOP**

**TopN program**

**WCMapper class**

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;

import org.apache.hadoop.mapred.Mapper;

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reporter;

public class WCMapper extends MapReduceBase implements Mapper&lt;LongWritable,

Text, Text,

IntWritable&gt; {

// Map function

public void map(LongWritable key, Text value, OutputCollector&lt;Text,

IntWritable&gt; output, Reporter rep) throws IOException

{

String line = value.toString();

// Splitting the line on spaces

for (String word : line.split(&quot; &quot;))

{

if (word.length() &gt; 0)

{

output.collect(new Text(word), new IntWritable(1));

} } } }

**WCReducer class**

import java.io.IOException;

import java.util.Iterator;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reducer;

import org.apache.hadoop.mapred.Reporter;

public class WCReducer extends MapReduceBase implements Reducer&lt;Text,

IntWritable, Text, IntWritable&gt; {

public void reduce(Text key, Iterator&lt;IntWritable&gt; value,

OutputCollector&lt;Text, IntWritable&gt; output,

Reporter rep) throws IOException

{

int count = 0;

// Counting the frequency of each words

while (value.hasNext())

{

IntWritable i = value.next();

count += i.get();

}

output.collect(key, new IntWritable(count));

} }

**WCDriver class**

import java.io.IOException;

import org.apache.hadoop.conf.Configured;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.FileInputFormat;

import org.apache.hadoop.mapred.FileOutputFormat;

import org.apache.hadoop.mapred.JobClient;

import org.apache.hadoop.mapred.JobConf;

import org.apache.hadoop.util.Tool;

import org.apache.hadoop.util.ToolRunner;

public class WCDriver extends Configured implements Tool {

public int run(String args[]) throws IOException

{

if (args.length &lt; 2)

{

System.out.println(&quot;Please give valid inputs&quot;);

return -1;

}

JobConf conf = new JobConf(WCDriver.class);

FileInputFormat.setInputPaths(conf, new Path(args[0]));

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

conf.setMapperClass(WCMapper.class);

conf.setReducerClass(WCReducer.class);

conf.setMapOutputKeyClass(Text.class);

conf.setMapOutputValueClass(IntWritable.class);

conf.setOutputKeyClass(Text.class);

conf.setOutputValueClass(IntWritable.class);

JobClient.runJob(conf);

return 0;

}

// Main Method

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

{

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

System.out.println(exitCode);

}

}

**Wordcount**

**WCDriver class**

import java.io.IOException;

import org.apache.hadoop.conf.Configured;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.FileInputFormat;

import org.apache.hadoop.mapred.FileOutputFormat;

import org.apache.hadoop.mapred.JobClient;

import org.apache.hadoop.mapred.JobConf;

import org.apache.hadoop.util.Tool;

import org.apache.hadoop.util.ToolRunner;

public class WCDriver extends Configured implements Tool {

public int run(String args[]) throws IOException

{

if (args.length &lt; 2)

{

System.out.println(&quot;Please give valid inputs&quot;);

return -1;

}

JobConf conf = new JobConf(WCDriver.class);

FileInputFormat.setInputPaths(conf, new Path(args[0]));

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

conf.setMapperClass(WCMapper.class);

conf.setReducerClass(WCReducer.class);

conf.setMapOutputKeyClass(Text.class);

conf.setMapOutputValueClass(IntWritable.class);

conf.setOutputKeyClass(Text.class);

conf.setOutputValueClass(IntWritable.class);

JobClient.runJob(conf);

return 0;

}

// Main Method

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

{

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

System.out.println(exitCode);

}

}

**WCMapper class**

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;

import org.apache.hadoop.mapred.Mapper;

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reporter;

public class WCMapper extends MapReduceBase implements Mapper&lt;LongWritable,

Text, Text,

IntWritable&gt; {

// Map function

public void map(LongWritable key, Text value, OutputCollector&lt;Text,

IntWritable&gt; output, Reporter rep) throws IOException

{

String line = value.toString();

// Splitting the line on spaces

for (String word : line.split(&quot; &quot;))

{

if (word.length() &gt; 0)

{

output.collect(new Text(word), new IntWritable(1));

} } } }

**WCReducer class**

import java.io.IOException;

import java.util.Iterator;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reducer;

import org.apache.hadoop.mapred.Reporter;

public class WCReducer extends MapReduceBase implements Reducer&lt;Text,

IntWritable, Text, IntWritable&gt; {

// Reduce function

public void reduce(Text key, Iterator&lt;IntWritable&gt; value,

OutputCollector&lt;Text, IntWritable&gt; output,

Reporter rep) throws IOException

{

int count = 0;

// Counting the frequency of each words

while (value.hasNext())

{

IntWritable i = value.next();

count += i.get();

}

output.collect(key, new IntWritable(count));

} }

**Output:**

hduser@bmsce-Precision-T1700:~$ start-all.sh

This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh

Starting namenodes on [localhost]

hduser@localhost's password:

localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namenode-bmsce-Precision-T1700.out

hduser@localhost's password:

localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datanode-bmsce-Precision-T1700.out

Starting secondary namenodes [0.0.0.0]

hduser@0.0.0.0's password:

0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hduser-secondarynamenode-bmsce-Precision-T1700.out

starting yarn daemons

starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourcemanager-bmsce-Precision-T1700.out

hduser@localhost's password:

localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-nodemanager-bmsce-Precision-T1700.out

hduser@bmsce-Precision-T1700:~$ jps

5428 Jps

4422 NameNode

4598 DataNode

5304 NodeManager

4812 SecondaryNameNode

4972 ResourceManager

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /

Found 6 items

drwxr-xr-x - hduser supergroup 0 2022-06-15 09:57 /lab7

drwxr-xr-x - hduser supergroup 0 2019-10-23 11:06 /output

drwxr-xr-x - hduser supergroup 0 2019-10-23 15:50 /rgs

drwxrwxr-x - hduser supergroup 0 2019-10-23 15:53 /tmp

drwxr-xr-x - hduser supergroup 0 2019-08-01 16:03 /user

drwxr-xr-x - hduser supergroup 0 2019-10-23 10:47 /yp

hduser@bmsce-Precision-T1700:~$ hadoop fs -mkdir /ss

hduser@bmsce-Precision-T1700:~$ hadoop fs -copyFromLocal Desktop/Input\_data.txt /ss/test.txt

hduser@bmsce-Precision-T1700:~$ hadoop jar /home/hduser/Desktop/shilpa12.jar WCDriver input output

Exception in thread "main" java.lang.ClassNotFoundException: WCDriver

at java.net.URLClassLoader.findClass(URLClassLoader.java:382)

at java.lang.ClassLoader.loadClass(ClassLoader.java:418)

at java.lang.ClassLoader.loadClass(ClassLoader.java:351)

at java.lang.Class.forName0(Native Method)

at java.lang.Class.forName(Class.java:348)

at org.apache.hadoop.util.RunJar.run(RunJar.java:214)

at org.apache.hadoop.util.RunJar.main(RunJar.java:136)

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /output/

Found 2 items

-rw-r--r-- 1 hduser supergroup 0 2019-10-23 11:06 /output/\_SUCCESS

-rw-r--r-- 1 hduser supergroup 23 2019-10-23 11:06 /output/part-00000

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /output/part-00000

-rw-r--r-- 1 hduser supergroup 23 2019-10-23 11:06 /output/part-00000

hduser@bmsce-Precision-T1700:~$ hadoop jar /home/hduser/Desktop/shilpa12.jar shilpa\_g.WCDriver /ss/test.txt /lab7/out123

22/06/18 10:30:02 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id

22/06/18 10:30:02 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=

22/06/18 10:30:02 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized

22/06/18 10:30:02 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.

22/06/18 10:30:03 INFO mapred.FileInputFormat: Total input paths to process : 1

22/06/18 10:30:03 INFO mapreduce.JobSubmitter: number of splits:1

22/06/18 10:30:03 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_local737984032\_0001

22/06/18 10:30:03 INFO mapreduce.Job: The url to track the job: http://localhost:8080/

22/06/18 10:30:03 INFO mapred.LocalJobRunner: OutputCommitter set in config null

22/06/18 10:30:03 INFO mapreduce.Job: Running job: job\_local737984032\_0001

22/06/18 10:30:03 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapred.FileOutputCommitter

22/06/18 10:30:03 INFO mapred.LocalJobRunner: Waiting for map tasks

22/06/18 10:30:03 INFO mapred.LocalJobRunner: Starting task: attempt\_local737984032\_0001\_m\_000000\_0

22/06/18 10:30:03 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]

22/06/18 10:30:03 INFO mapred.MapTask: Processing split: hdfs://localhost:54310/ss/test.txt:0+30

22/06/18 10:30:03 INFO mapred.MapTask: numReduceTasks: 1

22/06/18 10:30:03 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)

22/06/18 10:30:03 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100

22/06/18 10:30:03 INFO mapred.MapTask: soft limit at 83886080

22/06/18 10:30:03 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600

22/06/18 10:30:03 INFO mapred.MapTask: kvstart = 26214396; length = 6553600

22/06/18 10:30:03 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer

22/06/18 10:30:03 INFO mapred.LocalJobRunner:

22/06/18 10:30:03 INFO mapred.MapTask: Starting flush of map output

22/06/18 10:30:03 INFO mapred.MapTask: Spilling map output

22/06/18 10:30:03 INFO mapred.MapTask: bufstart = 0; bufend = 62; bufvoid = 104857600

22/06/18 10:30:03 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26214368(104857472); length = 29/6553600

22/06/18 10:30:03 INFO mapred.MapTask: Finished spill 0

22/06/18 10:30:03 INFO mapred.Task: Task:attempt\_local737984032\_0001\_m\_000000\_0 is done. And is in the process of committing

22/06/18 10:30:03 INFO mapred.LocalJobRunner: hdfs://localhost:54310/ss/test.txt:0+30

22/06/18 10:30:03 INFO mapred.Task: Task 'attempt\_local737984032\_0001\_m\_000000\_0' done.

22/06/18 10:30:03 INFO mapred.LocalJobRunner: Finishing task: attempt\_local737984032\_0001\_m\_000000\_0

22/06/18 10:30:03 INFO mapred.LocalJobRunner: map task executor complete.

22/06/18 10:30:03 INFO mapred.LocalJobRunner: Waiting for reduce tasks

22/06/18 10:30:03 INFO mapred.LocalJobRunner: Starting task: attempt\_local737984032\_0001\_r\_000000\_0

22/06/18 10:30:03 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]

22/06/18 10:30:03 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin: org.apache.hadoop.mapreduce.task.reduce.Shuffle@1f00f307

22/06/18 10:30:03 INFO reduce.MergeManagerImpl: MergerManager: memoryLimit=334338464, maxSingleShuffleLimit=83584616, mergeThreshold=220663392, ioSortFactor=10, memToMemMergeOutputsThreshold=10

22/06/18 10:30:03 INFO reduce.EventFetcher: attempt\_local737984032\_0001\_r\_000000\_0 Thread started: EventFetcher for fetching Map Completion Events

22/06/18 10:30:03 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of map attempt\_local737984032\_0001\_m\_000000\_0 decomp: 80 len: 84 to MEMORY

22/06/18 10:30:03 INFO reduce.InMemoryMapOutput: Read 80 bytes from map-output for attempt\_local737984032\_0001\_m\_000000\_0

22/06/18 10:30:03 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of size: 80, inMemoryMapOutputs.size() -> 1, commitMemory -> 0, usedMemory ->80

22/06/18 10:30:03 INFO reduce.EventFetcher: EventFetcher is interrupted.. Returning

22/06/18 10:30:03 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/18 10:30:03 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory map-outputs and 0 on-disk map-outputs

22/06/18 10:30:03 INFO mapred.Merger: Merging 1 sorted segments

22/06/18 10:30:03 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 74 bytes

22/06/18 10:30:03 INFO reduce.MergeManagerImpl: Merged 1 segments, 80 bytes to disk to satisfy reduce memory limit

22/06/18 10:30:03 INFO reduce.MergeManagerImpl: Merging 1 files, 84 bytes from disk

22/06/18 10:30:03 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into reduce

22/06/18 10:30:03 INFO mapred.Merger: Merging 1 sorted segments

22/06/18 10:30:03 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 74 bytes

22/06/18 10:30:03 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/18 10:30:03 INFO mapred.Task: Task:attempt\_local737984032\_0001\_r\_000000\_0 is done. And is in the process of committing

22/06/18 10:30:03 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/18 10:30:03 INFO mapred.Task: Task attempt\_local737984032\_0001\_r\_000000\_0 is allowed to commit now

22/06/18 10:30:03 INFO output.FileOutputCommitter: Saved output of task 'attempt\_local737984032\_0001\_r\_000000\_0' to hdfs://localhost:54310/lab7/out123/\_temporary/0/task\_local737984032\_0001\_r\_000000

22/06/18 10:30:03 INFO mapred.LocalJobRunner: reduce > reduce

22/06/18 10:30:03 INFO mapred.Task: Task 'attempt\_local737984032\_0001\_r\_000000\_0' done.

22/06/18 10:30:03 INFO mapred.LocalJobRunner: Finishing task: attempt\_local737984032\_0001\_r\_000000\_0

22/06/18 10:30:03 INFO mapred.LocalJobRunner: reduce task executor complete.

22/06/18 10:30:04 INFO mapreduce.Job: Job job\_local737984032\_0001 running in uber mode : false

22/06/18 10:30:04 INFO mapreduce.Job: map 100% reduce 100%

22/06/18 10:30:04 INFO mapreduce.Job: Job job\_local737984032\_0001 completed successfully

22/06/18 10:30:04 INFO mapreduce.Job: Counters: 38

File System Counters

FILE: Number of bytes read=8620

FILE: Number of bytes written=507746

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: Number of write operations=0

HDFS: Number of bytes read=60

HDFS: Number of bytes written=23

HDFS: Number of read operations=13

HDFS: Number of large read operations=0

HDFS: Number of write operations=4

Map-Reduce Framework

Map input records=2

Map output records=8

Map output bytes=62

Map output materialized bytes=84

Input split bytes=86

Combine input records=0

Combine output records=0

Reduce input groups=4

Reduce shuffle bytes=84

Reduce input records=8

Reduce output records=4

Spilled Records=16

Shuffled Maps =1

Failed Shuffles=0

Merged Map outputs=1

GC time elapsed (ms)=1

CPU time spent (ms)=0

Physical memory (bytes) snapshot=0

Virtual memory (bytes) snapshot=0

Total committed heap usage (bytes)=478150656

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=30

File Output Format Counters

Bytes Written=23

0

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /lab7/out123

Found 2 items

-rw-r--r-- 1 hduser supergroup 0 2022-06-18 10:30 /lab7/out123/\_SUCCESS

-rw-r--r-- 1 hduser supergroup 23 2022-06-18 10:30 /lab7/out123/part-00000

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /lab7/out123/part-00000

-rw-r--r-- 1 hduser supergroup 23 2022-06-18 10:30 /lab7/out123/part-00000

hduser@bmsce-Precision-T1700:~$ hadoop fs -cat /lab7/out123/part-00000

are 2

hi 2

how 2

you 2

**SPARK (SCALA)**

### Scala program to print your name.

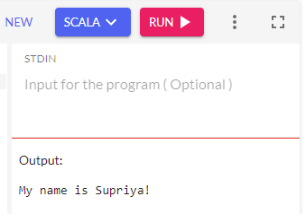
object ExPrintName {

def main(args: Array[String]) {

println("My name is Supriya!")

}

}



### Scala program to find largest number among two numbers.

object ExFindLargest {

def main(args: Array[String]) {

var number1=2500;

var number2=300;

var x = 10;

if( number1>number2){

println("Largest number is:" + number1);

}

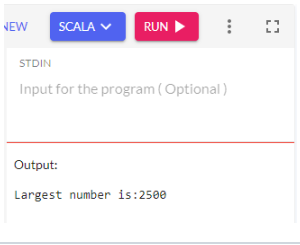
else{

println("Largest number is:" + number2);

}

}

}



### Scala program to find a number is positive, negative or positive.

object ExCheckNumber {

def main(args: **Array**[**String**]) {

/\*\*declare a variable\*/

var number= (-100);

**if**(number==0){

println("number is zero");

}

**else** **if**(number>0){

println("number is positive");

}

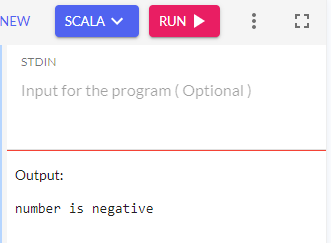
**else**{

println("number is negative");

}

}

}





### Scala program to declare string variable and print the string.

object ExampleString {

def main(args: Array[String]) {

//declare and assign string variable "text"

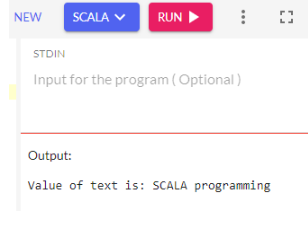
val text : String = "SCALA programming";

//print the value of string variable "text"

println("Value of text is: " + text);

}

}



### Scala program to demonstrate example of multiple variables declarations and assignments.

object ExampleVarDecAndAssin {

def main(args: Array[String]) {

var (name: String, age: Int) = ("Supriya",20);

//print values

println("Name: "+name);

println("Age: "+age);

//declaration without specifying data type

var (address,mobile) = ("Dharwad, India",1234567890);

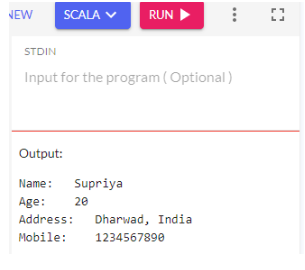
//print values

println("Address: "+address);

println("Mobile: "+mobile);

}

}



### Scala program to print numbers from 1 to 100 using for loop.

object ExampleForLoop1 {

def main(args: Array[String]) {

var counter: Int=0;

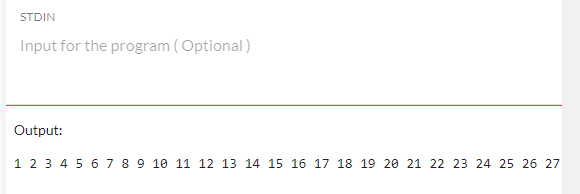
for(counter <- 1 to 100)

print(counter + " ");

// to print new line

println();

}



### Scala program to print numbers from 1 to 100 using for loop with until to determine loop range.

object ExampleForLoop2 {

def main(args: Array[String]) {

var counter: Int=0;

for(counter <- 1 until 200)

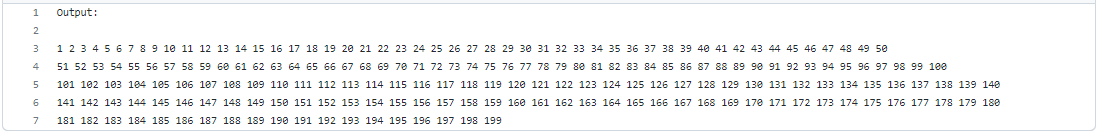
print(counter + " ");

// to print new line

println();

}

}



### 8)Scala program to demonstrate example of collection list and for loop.

object ExampleForAndCollection {

def main(args: Array[String]) {

//declare an integer

var N: Int=0;

//declare integer list

var numbers = List(100,200,300,400);

//to print all numbers using for loop

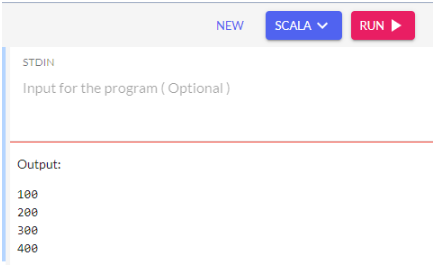
for(N<-numbers){

println(N);

}

}

}



### 9) Scala program to create a user define function to return largest number among two numbers.

object ExampleUDFToGetLargestNumber {

//function definition

def getLargestNumber(x: Int, y: Int) : Int ={

var largestNumber: Int=0;

if(x>y)

largestNumber=x;

else

largestNumber=y;

return largestNumber;

}

def main(args: Array[String]) {

var a: Int=100;

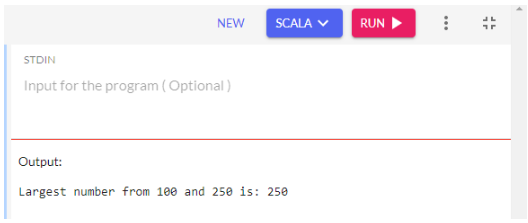
var b: Int=250;

//function calling

println("Largest number from "+ a+" and "+ b +" is: "+ getLargestNumber(a,b));

}

}



### Scala program of array - Declare, print and calculate sum of all elements.

object ExampleArray1 {

def main(args: Array[String]) {

var numbers = Array(10,20,30,40,50);

var N:Int=0;

//print all array elements

println("All array elements: ");

for ( N <- numbers ) {

println(N);

}

//calculating SUM of all elements

var sum: Int=0;

for ( N <- numbers ) {

sum+=N;

}

println("Sum of all array elements: "+sum);

}

}

