Q2.. Part 1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

npbdh login: bigcdac432557

bigcdac432557@npbdh.cloudloka.com's password:

Last login: Wed Dec 14 08:58:31 2022 from ec2-65-1-45-35.ap-south-1.compute.amazonaws.com

[bigcdac432557@ip-10-1-1-204 ~]$ hadoop fs -get /user/bigcdac432557/txns1.txt

[bigcdac432557@ip-10-1-1-204 ~]$ hadoop fs -put txns1.txt /bigcdac432557

put: Permission denied: user=bigcdac432557, access=WRITE, inode="/":hdfs:supergroup:drwxr-xr-x

[bigcdac432557@ip-10-1-1-204 ~]$ hadoop fs -put txns1.txt one

[bigcdac432557@ip-10-1-1-204 ~]$ hadoop fs -put txns1.txt

put: `txns1.txt': File exists

[bigcdac432557@ip-10-1-1-204 ~]$ hadoop fs -get /user/bigcdac432557/custs.txt

get: `/user/bigcdac432557/custs.txt': No such file or directory

[bigcdac432557@ip-10-1-1-204 ~]$ hadoop fs -get /user/bigcdac432557/custs.txt

get: `/user/bigcdac432557/custs.txt': No such file or directory

[bigcdac432557@ip-10-1-1-204 ~]$ ls

**Kalimba.mp3** txns1.txt

[bigcdac432557@ip-10-1-1-204 ~]$ hadoop fs -get /user/bigcdac432557/custs.txt

get: `/user/bigcdac432557/custs.txt': No such file or directory

[bigcdac432557@ip-10-1-1-204 ~]$ hadoop fs -get /user/bigcdac432557/custs.txt

get: `/user/bigcdac432557/custs.txt': No such file or directory

[bigcdac432557@ip-10-1-1-204 ~]$ hadoop fs -get /user/bigcdac432557/custs.txt

[bigcdac432557@ip-10-1-1-204 ~]$ hadoop fs -put txns1.txt

put: `txns1.txt': File exists

[bigcdac432557@ip-10-1-1-204 ~]$ hadoop fs -put custs.txt

put: `custs.txt': File exists

[bigcdac432557@ip-10-1-1-204 ~]$ hadoop fs -put custs.txt one

[bigcdac432557@ip-10-1-1-204 ~]$ hive

WARNING: Use "yarn jar" to launch YARN applications.

SLF4J: Class path contains multiple SLF4J bindings.

SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/jars/log4j-slf4j-impl-2.8.2.jar!/org/slf4j/impl/StaticLoggerB

inder.class]

SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/jars/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBin

der.class]

SLF4J: See <http://www.slf4j.org/codes.html#multiple_bindings> for an explanation.

SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

2022-12-14 10:18:59,071 main WARN JNDI lookup class is not available because this JRE does not support JNDI. JNDI string lookups will not be availabl

e, continuing configuration. Ignoring java.lang.ClassNotFoundException: org.apache.logging.log4j.core.lookup.JndiLookup

Logging initialized using configuration in jar:file:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/jars/hive-common-2.1.1-cdh6.2.1.jar!/hive-l

og4j2.properties Async: false

WARNING: Hive CLI is deprecated and migration to Beeline is recommended.

hive> use Kushal;

OK

Time taken: 0.316 seconds

hive> show tables;

OK

custom

custom\_orc

Time taken: 0.216 seconds, Fetched: 2 row(s)

hive> create table custom2(custid int, fname string, lname string,age int, profession string)

> row format delimited

> fields terminated by ',' stored as textfile

> location

> '/user/bigcdac432557';

OK

Time taken: 0.349 seconds

hive> create table custom2\_orc(custid int, fname string, lname string,age int, profession string)

> stored as orc;

OK

Time taken: 0.342 seconds

hive> show tables;

OK

custom

custom2

custom2\_orc

custom\_orc

Time taken: 0.053 seconds, Fetched: 4 row(s)

hive> load data local inpath 'custs.txt' overwrite into table Custom2;

Loading data to table kushal.custom2

OK

Time taken: 0.951 seconds

hive> select profession from Custom2

> group by profession;

Query ID = bigcdac432557\_20221214102718\_afa00e97-52b2-4b3b-9d27-c124827c0004

Total jobs = 1

Launching Job 1 out of 1

Number of reduce tasks not specified. Estimated from input data size: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

22/12/14 10:27:19 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032

22/12/14 10:27:19 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032

Starting Job = job\_1663041244711\_22996, Tracking URL = <http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1663041244711_22996/>

Kill Command = /opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/bin/hadoop job -kill job\_1663041244711\_22996

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1

2022-12-14 10:27:37,297 Stage-1 map = 0%, reduce = 0%

2022-12-14 10:28:02,127 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.49 sec

2022-12-14 10:28:15,582 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.58 sec

MapReduce Total cumulative CPU time: 5 seconds 580 msec

Ended Job = job\_1663041244711\_22996

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.58 sec HDFS Read: 399761 HDFS Write: 1382 HDFS EC Read: 0 SUCCESS

Total MapReduce CPU Time Spent: 5 seconds 580 msec

OK

Accountant

Actor

Agricultural and food scientist

Architect

Artist

Athlete

Automotive mechanic

Carpenter

Chemist

Childcare worker

Civil engineer

Coach

Computer hardware engineer

Computer software engineer

Computer support specialist

Dancer

Designer

Doctor

Economist

Electrical engineer

Electrician

Engineering technician

Environmental scientist

Farmer

Financial analyst

Firefighter

Human resources assistant

Judge

Lawyer

Librarian

Loan officer

Musician

Nurse

Pharmacist

Photographer

Physicist

Pilot

Police officer

Politician

Psychologist

Real estate agent

Recreation and fitness worker

Reporter

Secretary

Social Worker

Social worker

Statistician

Teacher

Therapist

Veterinarian

Writer

hive> select profession,count(profession) from Custom2

> group by profession;

Query ID = bigcdac432557\_20221214102902\_a4ea1dc3-04b1-43fc-b004-d10cfe0e1cf2

Total jobs = 1

Launching Job 1 out of 1

Number of reduce tasks not specified. Estimated from input data size: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

22/12/14 10:29:02 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032

22/12/14 10:29:02 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032

Starting Job = job\_1663041244711\_22999, Tracking URL = <http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1663041244711_22999/>

Kill Command = /opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/bin/hadoop job -kill job\_1663041244711\_22999

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1

2022-12-14 10:29:33,115 Stage-1 map = 0%, reduce = 0%

2022-12-14 10:29:48,940 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.77 sec

2022-12-14 10:30:13,720 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.81 sec

MapReduce Total cumulative CPU time: 5 seconds 810 msec

Ended Job = job\_1663041244711\_22999

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.81 sec HDFS Read: 400494 HDFS Write: 1584 HDFS EC Read: 0 SUCCESS

Total MapReduce CPU Time Spent: 5 seconds 810 msec

OK

Accountant 199

Actor 202

Agricultural and food scientist 195

Architect 203

Artist 175

Athlete 196

Automotive mechanic 193

Carpenter 181

Chemist 209

Childcare worker 207

Civil engineer 193

Coach 201

Computer hardware engineer 204

Computer software engineer 216

Computer support specialist 222

Dancer 185

Designer 205

Doctor 197

Economist 189

Electrical engineer 192

Electrician 194

Engineering technician 204

Environmental scientist 176

Farmer 201

Financial analyst 198

Firefighter 217

Human resources assistant 212

Judge 196

Lawyer 212

Librarian 218

Loan officer 221

Musician 205

Nurse 192

Pharmacist 213

Photographer 222

Physicist 201

Pilot 211

Police officer 210

Politician 228

Psychologist 194

Real estate agent 191

Recreation and fitness worker 210

Reporter 200

Secretary 200

Social Worker 1

Social worker 212

Statistician 196

Teacher 204

Therapist 187

Veterinarian 208

Writer 101

Time taken: 73.822 seconds, Fetched: 51 row(s)

Q2.. Part 2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<DATABASES ALREADY CREATED>

hive> select product,sum(amt) from custom

> group by product

> order by amt

> limit 10;

FAILED: SemanticException [Error 10004]: Line 3:9 Invalid table alias or column reference 'amt': (possible column names are: product, \_c1)

hive> select product,sum(amt) from custom

> group by product

> order by sum(amt)

> limit 10;

FAILED: SemanticException [Error 10004]: Line 3:13 Invalid table alias or column reference 'amt': (possible column names are: product, \_c1)

hive> select product,sum(amt) from Custom

> group by product

> limit 10;

Query ID = bigcdac432557\_20221214103817\_2003ead6-f797-4cdf-8a2d-0352792472a2

Total jobs = 1

Launching Job 1 out of 1

Number of reduce tasks not specified. Estimated from input data size: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

22/12/14 10:38:17 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032

22/12/14 10:38:17 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032

Starting Job = job\_1663041244711\_23037, Tracking URL = <http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1663041244711_23037/>

Kill Command = /opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/bin/hadoop job -kill job\_1663041244711\_23037

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1

2022-12-14 10:39:18,801 Stage-1 map = 0%, reduce = 0%

2022-12-14 10:40:19,318 Stage-1 map = 0%, reduce = 0%

2022-12-14 10:41:20,055 Stage-1 map = 0%, reduce = 0%

2022-12-14 10:42:20,074 Stage-1 map = 0%, reduce = 0%

2022-12-14 10:43:20,436 Stage-1 map = 0%, reduce = 0%

Q1.. ECLIPSE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

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;

public class "classname" {

   public static class "mapperclassname" extends Mapper<LongWritable, Text, Text, IntWritable>{

 }

  public static class "reducerclassname"  extends Reducer<Text,IntWritable,Text,IntWritable> {

  }

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

    Configuration conf = new Configuration();

    Job job = Job.getInstance(conf, " ");

    job.setJarByClass();

    job.setMapperClass();

    job.setReducerClass();

    job.setNumReduceTasks(1);

    job.setMapOutputKeyClass();

    job.setMapOutputValueClass();

    job.setOutputKeyClass();

    job.setOutputValueClass(;

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

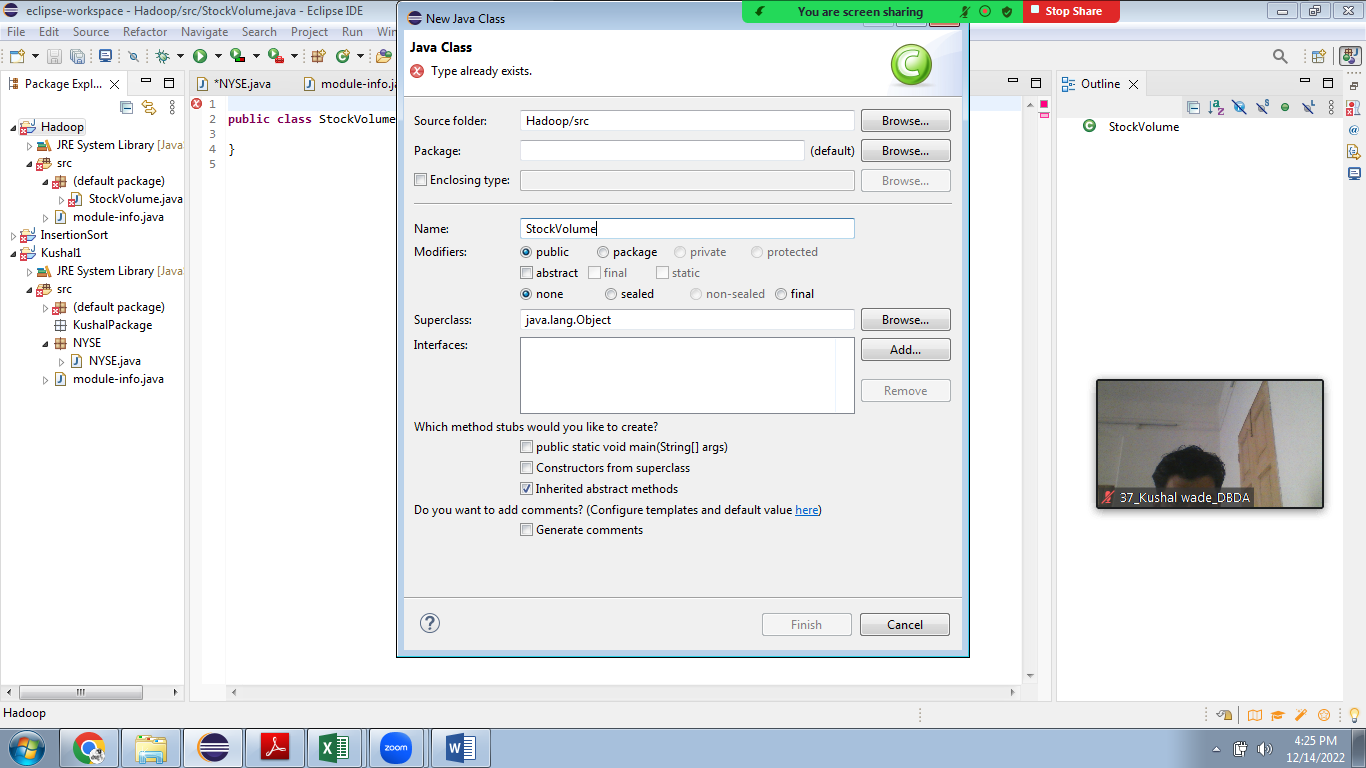
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

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

  }

}

}



**import** java.io.\*;

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

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

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

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

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

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

**import** org.apache.hadoop.conf.\*;

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

**import** org.apache.hadoop.mapreduce.lib.input.\*;

**import** org.apache.hadoop.mapreduce.lib.output.\*;

**public** **class** StockVolume {

**public** **static** **class** MapClass **extends** Mapper<LongWritable,Text,Text,DoubleWritable>

{

**public** **void** map(LongWritable key, Text value, Context context)

{

**try**{

String[] str = value.toString().split(",");

**double** high = Double.*parseDouble*(str[4]);

context.write(**new** Text(str[1]),**new** DoubleWritable(high));

}

**catch**(Exception e)

{

System.***out***.println(e.getMessage());

}

}

}

**public** **static** **class** ReduceClass **extends** Reducer<Text,DoubleWritable,Text,DoubleWritable>

{

**private** DoubleWritable result = **new** DoubleWritable();

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

**double** max = 0.00;

**for** (DoubleWritable val : values)

{

**if** (val.get() > max) {

max = val.get();

}

}

result.set(max);

context.write(key, result);

//context.write(key, new LongWritable(sum));

}

}

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

Configuration conf = **new** Configuration();

conf.set("mapreduce.output.textoutputformat.separator",",");

//conf.set("name", "value")

conf.set("mapreduce.input.fileinputformat.split.maxsize", "28311552");

Job job = Job.getInstance(conf, "All Time High Price for each stock");

job.setJarByClass(StockVolume.**class**);

job.setMapperClass(MapClass.**class**);

job.setCombinerClass(ReduceClass.**class**);

job.setReducerClass(ReduceClass.**class**);

job.setNumReduceTasks(1);

job.setOutputKeyClass(Text.**class**);

job.setOutputValueClass(DoubleWritable.**class**);

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

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

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

}

}

Q3.. PySpark\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

npbdh login: bigcdac432557

bigcdac432557@npbdh.cloudloka.com's password:

Last login: Wed Dec 14 10:04:33 2022 from ec2-65-1-45-35.ap-south-1.compute.amazonaws.com

[bigcdac432557@ip-10-1-1-204 ~]$ pyspark

Python 3.7.6 (default, Jan 8 2020, 19:59:22)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

Setting default log level to "WARN".

To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).