Regine C. Palomo
CMSC 178 Lab Activity

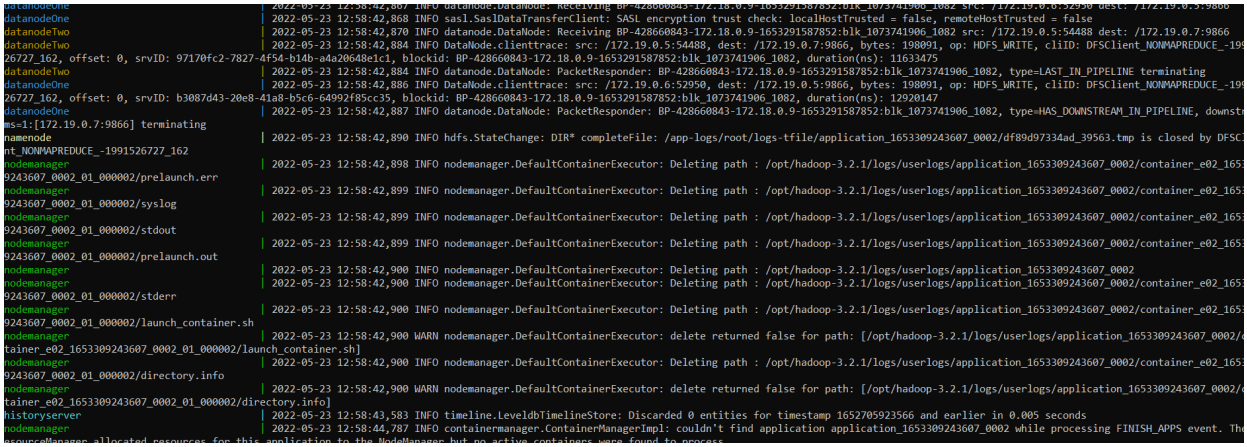
May 23, 2022

Big Data Activity 3: Intro to Docker, Hadoop, and Hive

Docker Setup

Containers were created by running the following commands. These containers will be running in the background.

```
docker-compose up
Docker-compose down
```



Docker containers created

Dataset

Download Dataset. Obtain the heart disease dataset using the link: `https://archive.ics.uci.edu/ml/machine-learning-databases/heartdisease/processed.switzerland.data`.

Attribute Information: (only 14 attributes were used)

1. **Age** - age in years
2. **Sex** - (1 = male; 0 = female)
3. **CP -** chest pain type
    a. typical angina (1)
    b. atypical angina (2)

```
32,1,1,95,0,?,0,127,0,.7,1,?,?,1
34,1,4,115,0,?,?,154,0,.2,1,?,?,1
35,1,4,?,0,?,0,130,1,?,?,?,7,3
36,1,4,110,0,?,0,125,1,1,2,?,6,1
38,0,4,105,0,?,0,166,0,2.8,1,?,?,2
38,0,4,110,0,0,0,156,0,0,2,?,3,1
38,1,3,100,0,?,0,179,0,-1.1,1,?,?,0
38,1,3,115,0,0,0,128,1,0,2,?,7,1
38,1,4,135,0,?,0,150,0,0,?,?,3,2
38,1,4,150,0,?,0,120,1,?,?,?,3,1
40,1,4,95,0,?,1,144,0,0,1,?,?,2
41,1,4,125,0,?,0,176,0,1,6,1,?,?,2
```

    c. non-anginal pain (3)
    d. asymptomatic (4)
4. **Trestbps -** resting blood pressure (in mm Hg on admission to the hospital)
5. **Chol** - serum cholestoral in mg/dl
6. **Fbs -** (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
7. **Restecg -** resting electrocardiographic results
8. **Thalac -** maximum heart rate achieved
9. **Exang-** exercise induced angina (1 = yes; 0 = no)
10. **Oldpeak -** ST depression induced by exercise relative to rest
11. **Slope-** the slope of the peak exercise ST segment
12. **Ca**- number of major vessels (0-3) colored by flourosopy
13. **thal-** (3 = normal; 6 = fixed defect; 7 = reversable defect)
14. **Num** - the predicted value

Putting Files into HDFS. HDFS provides a distributed file system designed to run on commodity hardware. Here, files are broken into blocks and distributed across the Hadoop physical system.

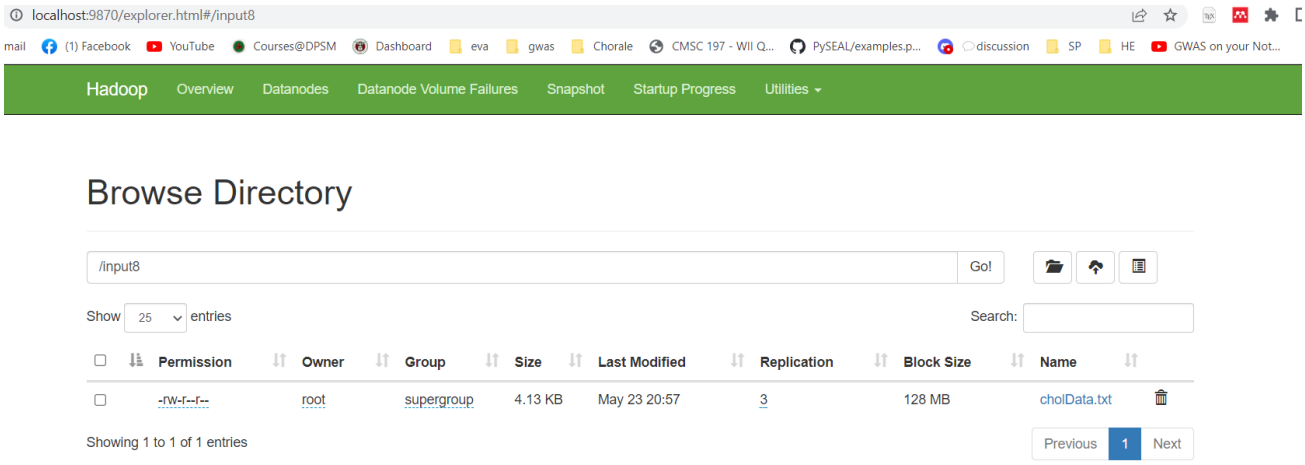
The dataset `'cholData.txt'` were added to HDFS using the following commands:

```
docker cp cholData.txt namenode:/
docker exec -it namenode /bin/bash
hdfs dfs -copyFromLocal cholData.txt /
hdfs dfs -ls /
```

```
C:\Users\Reg\Desktop\docker-hadoop>docker cp cholData.txt namenode:/

C:\Users\Reg\Desktop\docker-hadoop>docker exec -it namenode /bin/bash
root@ab1a173bc22b:/# hdfs dfs -copyFromLocal cholData.txt /
2022-05-23 12:57:02,982 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false,
ostTrusted = false
root@ab1a173bc22b:/# hdfs dfs -ls /
Found 15 items
drwxrwxrwt   - root root               0 2022-05-23 11:55 /app-logs
-rw-r--r--   3 root supergroup       4232 2022-05-23 12:57 /cholData.txt
drwxr-xr-x   - root supergroup          0 2022-05-23 10:14 /input1
drwxr-xr-x   - root supergroup          0 2022-05-23 11:19 /input3
drwxr-xr-x   - root supergroup          0 2022-05-23 11:20 /input4
drwxr-xr-x   - root supergroup          0 2022-05-23 11:44 /input5
drwxr-xr-x   - root supergroup          0 2022-05-23 12:17 /input6
drwxr-xr-x   - root supergroup          0 2022-05-23 12:36 /input7
drwxr-xr-x   - root supergroup          0 2022-05-23 09:35 /inputnew
drwxr-xr-x   - root supergroup          0 2022-05-23 11:56 /output5
drwxr-xr-x   - root supergroup          0 2022-05-23 12:29 /output6
drwxr-xr-x   - root supergroup          0 2022-05-23 12:37 /output7
drwxr-xr-x   - root supergroup          0 2022-05-23 07:40 /rmstate
drwxrwxr-x   - root supergroup          0 2022-05-23 11:55 /tmp
drwxr-xr-x   - root supergroup          0 2022-05-23 07:40 /user
```

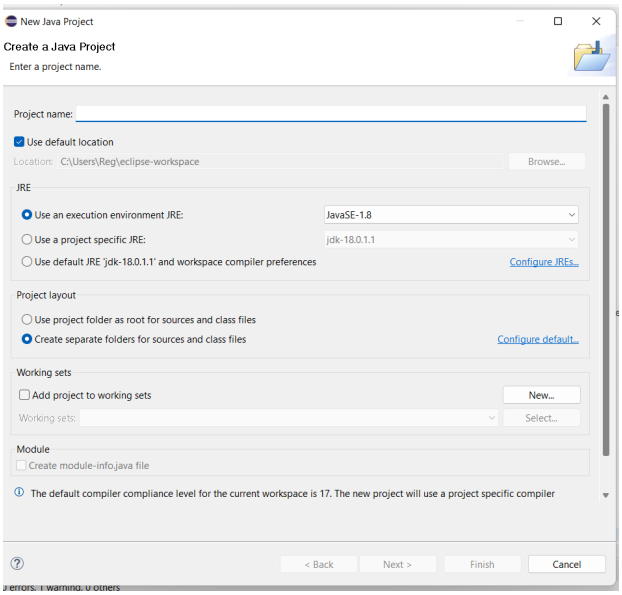
Adding `'cholData.txt'` to HDFS

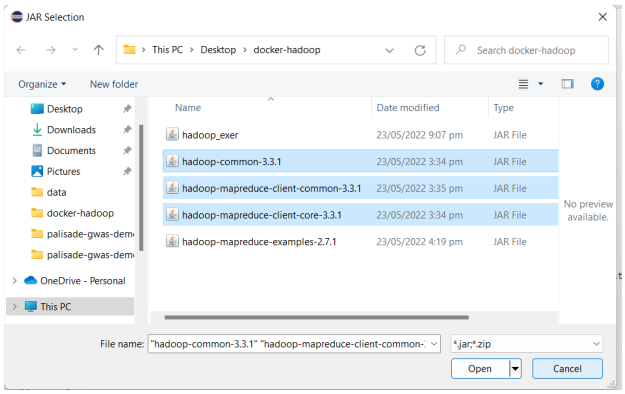


Checking if the dataset were added using the Browser Directory

Java

Eclipse IDE Setup. In creating a new project, make sure the execution environment JRE is set to JavaSE-1.8. Add the jar files as an external library before you start programming.





Creating a new Java Project

Java Map Reduce Function. In this activity, we need to compute the average cholesterol based on sex(male or female). The schema of input data is `sex` at position 1st and `cholesterol` at 4th position.

cholAvg.java

```
1  package org.example.hadoopcodes;
2  import java.io.IOException;
3
4  import org.apache.hadoop.conf.Configuration;
5  import org.apache.hadoop.fs.Path;
6  import org.apache.hadoop.io.FloatWritable;
7  import org.apache.hadoop.io.LongWritable;
8  import org.apache.hadoop.io.Text;
9
10 import org.apache.hadoop.mapreduce.Job;
11 import org.apache.hadoop.mapreduce.Mapper;
12 import org.apache.hadoop.mapreduce.Reducer;
13 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
14 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
15
16
17 public class cholAvg {
18
19 public static class MapperClass extends Mapper<LongWritable, Text, Text, FloatWritable> {
20
21     public void map(LongWritable key, Text empRecord, Context con) throws IOException, InterruptedE
22       String[] word = empRecord.toString().split(",");
23       String sex = word[1];
24       try {
25        Float salary = Float.parseFloat(word[4]);
26        con.write(new Text(sex), new FloatWritable(salary));
27       } catch (Exception e) {
28        e.printStackTrace();
29       }
30     }
31 }
32
33 public static class ReducerClass extends Reducer<Text, FloatWritable, Text, Text> {
34
35     //private Text res = new FloatWritable();
36
37      public void reduce(Text key, Iterable<FloatWritable> valueList,
38        Context con) throws IOException, InterruptedException {
```

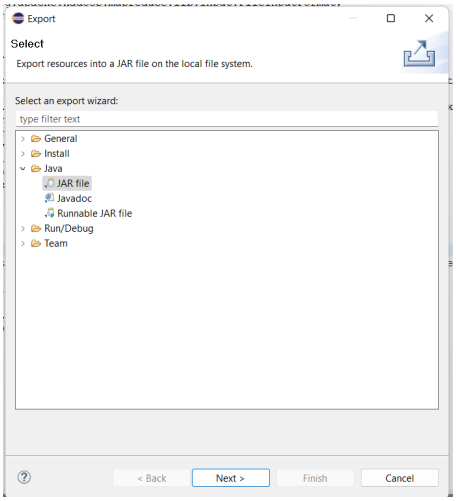
```
33 public static class ReducerClass extends Reducer<Text, FloatWritable, Text, Text> {
34
35     //private Text res = new FloatWritable();
36
37      public void reduce(Text key, Iterable<FloatWritable> valueList,
38        Context con) throws IOException, InterruptedException {
39          Float total = (float) 0;
40          int count = 0;
41          for (FloatWritable var : valueList) {
42              total += var.get();
43              System.out.println("reducer " + var.get());
44              count++;
45          }
46          Float avg = (Float) total / count;
47          //float avg = (Float) total / count;
48
```

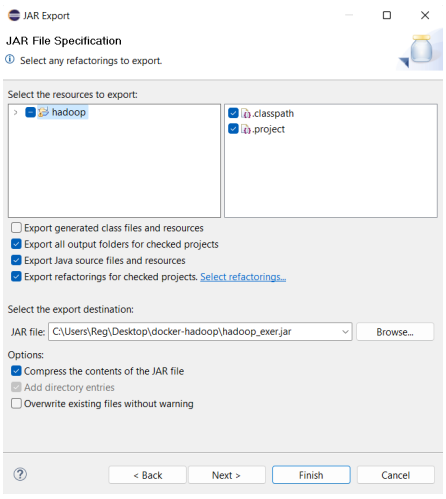
```
48
49
50          //res.set(avg);
51          String out = "Total: " + total + " :: " + "Average: " + avg;
52          con.write(key, new Text(out));
53      }
54  }
55
56      public static void main(String[] args) throws Exception{
57          Configuration conf = new Configuration();
58          Job job = Job.getInstance(conf, "cholavg");
59          job.setJarByClass(cholAvg.class);
60          job.setMapperClass(MapperClass.class);
61          //job.setCombinerClass(ReducerClass.class);
62          job.setReducerClass(ReducerClass.class);
63          job.setOutputKeyClass(Text.class);
64          job.setOutputValueClass(FloatWritable.class);
65          FileInputFormat.addInputPath(job, new Path(args[0]));
66          FileOutputFormat.setOutputPath(job, new Path(args[1]));
67          System.exit(job.waitForCompletion(true)? 0 : 1);
68
69      }
```

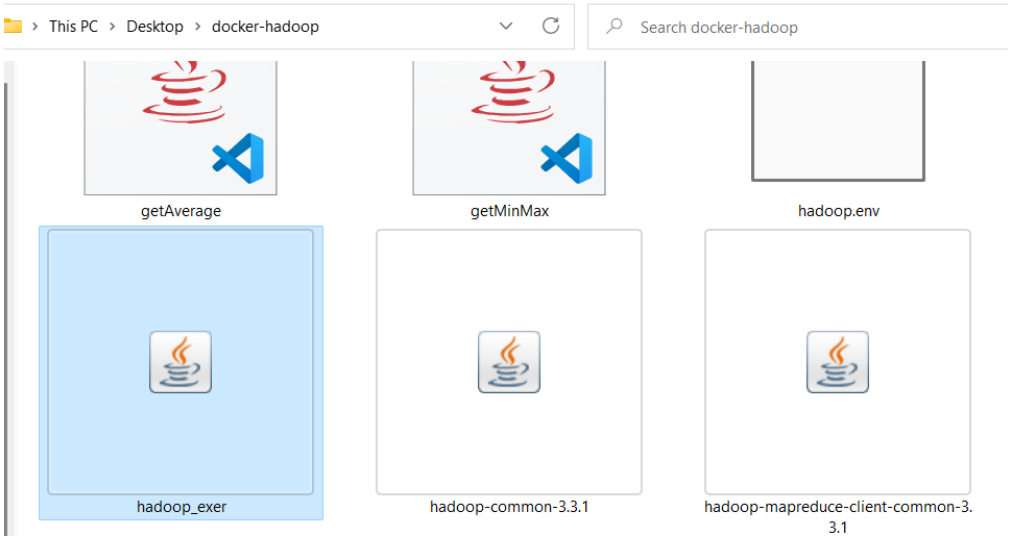
Java Map Reduce Function

In the context of map/reduce, we have to write the mapper(map method) and reducer (reduce method) class. In the map method, process input file line by line split the given input line, and extract sex and cholesterol. Then we write extracted sex and cholesterol in the context object. The output of the mapper is key as sex(M=1 or F=0).

In reduce method, the cholesterol list is iterated, and the average is computed. Average cholesterol is written in context as Text with sex 0 or 1.







hadoop_exer.jar

Hadoop Interaction

The dataset `'hadoop_exer.jar'` were added to HDFS using the following commands:

```
docker cp hadoop_exer.jar namenode:/
docker exec -it namenode /bin/bash
hdfs dfs -mkdir /input9/
hdfs dfs -mv /cholData.txt /input9/
hadoop jar hadoop_exer.jar org.example.hadoopcodes.cholAvg /input9 /output9
```

```
C:\Users\Reg\Desktop\docker-hadoop>docker cp hadoop_exer.jar namenode:/

C:\Users\Reg\Desktop\docker-hadoop>docker exec -it namenode /bin/bash
root@ab1a173bc22b:/# hdfs dfs -mkdir /input8/
mkdir: `/input8': File exists
root@ab1a173bc22b:/# hdfs dfs -mkdir /input9/
root@ab1a173bc22b:/# hdfs dfs -mv /cholData.txt /input9/
root@ab1a173bc22b:/# hadoop jar hadoop_exer.jar org.example.hadoopcodes.cholAvg /input9 /output9
2022-05-23 13:10:25,936 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/172.19.0.3:8032
2022-05-23 13:10:26,184 INFO client.AHSProxy: Connecting to Application History server at historyserver/172.19.0.4:102
2022-05-23 13:10:26,406 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implemen
the Tool interface and execute your application with ToolRunner to remedy this.
2022-05-23 13:10:26,428 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/stagin
root/.staging/job_1653309243607_0003
2022-05-23 13:10:26,523 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remot
```

Hadoop_exer.jar added to HDFS

Here, we created a folder named `input9` and moved the data `cholData.txt` to it. In this way, only the files located in the folder will be affected/analyzed when we run the next line of codes.

```
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=32
                Reduce input records=123
                Reduce output records=2
                Spilled Records=246
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=186
                CPU time spent (ms)=2010
                Physical memory (bytes) snapshot=478547968
                Virtual memory (bytes) snapshot=13561450496
                Total committed heap usage (bytes)=400556032
                Peak Map Physical memory (bytes)=286019584
                Peak Map Virtual memory (bytes)=5106335744
                Peak Reduce Physical memory (bytes)=192528384
                Peak Reduce Virtual memory (bytes)=8455114752
        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=4232
        File Output Format Counters
                Bytes Written=58
```

The following commands will run the program and output the result:

```
hdfs dfs -get /output9 .
dir
cd output9
dir
cat part-r-00000
```

(1) Retrieves the folder to the current directory then we change the directory using command (2). Command (5) outputs the result of the program.

```
                Bytes Written=58
oot@ab1a173bc22b:/# hdfs dfs -get /output9 .
022-05-23 13:10:57,329 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteH
stTrusted = false
oot@ab1a173bc22b:/# dir
EYS  cholData.txt   etc          hadoop_exer.jar  lib    mnt      output8  root    sbin  tmp
in   dev            hadoop       home             lib64  opt      output9  run     srv   usr
oot  entrypoint.sh  hadoop-data  input7           media  output7  proc     run.sh  sys   var
oot@ab1a173bc22b:/# cd output9
oot@ab1a173bc22b:/output9# dir
SUCCESS  part-r-00000
```

Running the program

```
root@ab1a173bc22b:/output9# cat part-r-00000
        Total: 0.0 :: Average: 0.0
        Total: 0.0 :: Average: 0.0
```

Result

Based on the result, using the dataset, the average cholesterol for both males and females is 0.

HiveQL Version

Hive is an initiative started by Facebook to provide a traditional Data Warehouse interface for MapReduce programming. For writing queries for MapReduce in SQL fashion, the Hive compiler converts them in the background to be executed in the Hadoop cluster

Docker setup. Run the docker container for the Hive server using the following command:

```
docker ps
docker exec -it docker-hadoop-hive-server-1 /bin/bash
hive
```

```
C:\Users\Reg\Desktop\docker-hadoop>docker exec -it docker-hadoop-hive-server-1 /bin/bash
root@bcfa62ee5c0e:/opt# hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/Sta
ticLoggerBinder.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]
```

Docker container running

The following commands will create a table in Hive and will overwrite the data stored in `input10` folder which is the `cholData.txt`:

```
create table testtable2 (x1 int, x2 int, x3 int, x4 int, x5 int, x6 int,
x7 int, x8 int, x9 int, x10 int, x11 int, x12 int, x13 int, x14 int)
row format delimited
fields terminated by ',' ;

LOAD DATA INPATH '/input10' OVERWRITE INTO TABLE testtable2;
```

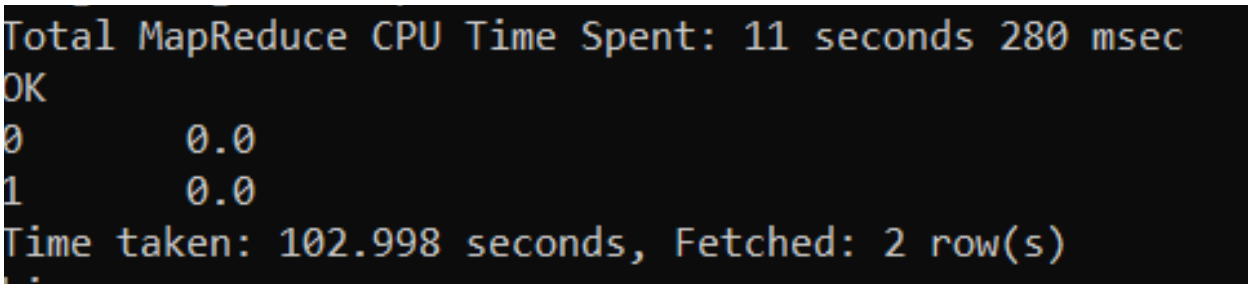
```
67      1       1       145     0       0       2       125     0       0       2       NULL    3
68      1       4       135     0       0       1       120     1       0       1       NULL    7
68      1       4       145     0       NULL    0       136     0       1       1       NULL    NULL
69      1       4       135     0       0       0       130     0       0       2       NULL    6
69      1       4       NULL    0       0       1       NULL    NULL    NULL    NULL    NULL    7
70      1       4       115     0       0       1       92      1       0       2       NULL    7
70      1       4       140     0       1       0       157     1       2       2       NULL    7
72      1       3       160     0       NULL    2       114     0       1       2       2       NULL
73      0       3       160     0       0       1       121     0       0       1       NULL    3
74      1       2       145     0       NULL    1       123     0       1       1       NULL    NULL
Time taken: 0.773 seconds, Fetched: 123 row(s)
```

Table created from the dataset

Command to get the average cholesterol level per gender/sex. Here x2 is assigned as sex while x5 was assigned as cholesterol.

```
SELECT x2, avg (x5) from testtable2 group by x2;
```

```
  set mapreduce.job.reduces=<number>
Starting Job = job_1653317428366_0002, Tracking URL = http://resourcemanager:8088/proxy/application_1653317428366_0002
Kill Command = /opt/hadoop-2.7.4/bin/hadoop job  -kill job_1653317428366_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-05-23 15:15:26,131 Stage-1 map = 0%,  reduce = 0%
2022-05-23 15:15:39,113 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.95 sec
2022-05-23 15:16:40,143 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 11.28 sec
MapReduce Total cumulative CPU time: 11 seconds 280 msec
Ended Job = job_1653317428366_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 11.28 sec   HDFS Read: 13805 HDFS Write: 123 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 280 msec
OK
0       0.0
1       0.0
Time taken: 102.998 seconds, Fetched: 2 row(s)
```



Result

We were able to obtain the same result as the one above. Using the same dataset, the average cholesterol for both males and females is 0.

Comparison:

- It is easier to perform SQL-like operations on HDFS using HIVE.
- It is easier to set up HIVE compare to MapReduce.
- We only need to write a few lines of code in HIVE compared to MapReduce.

- The code execution time in HIVE is longer but development effort is less while MapReduce is the opposite.
- Based on the result, the execution time in HIVE is 102.998 seconds.

Note the challenges you faced in doing the activity.

- Class not found error in MapReduce (we have to be mindful of the modules in eclipse)
- Error related to imports in Eclipse
- For MapReduce, it is difficult to debug and program in general since you have to do the process all over again from uploading a file in HDFS to generating a jar file.
- I also experience difficulty in overwriting data files in HIVE.
- Overall, I find it more difficult to execute the activity using the MapReduce version compared to the HIVE version.