

Kingdom of Saudi Arabia
Ministry of Education
University of Jeddah
College of Science and Computer
Engineering



المملكة العربية السعودية
وزارة التعليم
جامعة جدة
كلية علوم و هندسة الحاسب

Writing Hadoop Jobs!

Big Data

Amal Almutairi

Nouf Mansour

Noor Balfas

Muna Saleh Dini

Step 1: Create two Datasets!

Python program for creating the dataset:

We're using the **Faker** library to generate the dataset, which makes the process much easier.

Dataset Generation

The Buyers dataset

```
import csv
from faker import Faker
import random
fake = Faker()
with open('buyers.csv', mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['BuyerID', 'BuyerName', 'BuyerAge', 'BuyerGender', 'BuyerSalary'])
    for buyer_id in range(1, 10000 + 1):
        writer.writerow([
            buyer_id,
            fake.name_male() if (gender := fake.random_element(elements=('male', 'female'))) == 'male' else fake.name_female(),
            random.randint(12, 75),
            gender,
            round(random.uniform(3500, 11000), 2)
        ])
```

The Purchases dataset

```
import csv
import random

with open('Purchases.csv', mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['purchID', 'BuyerID', 'purchPrice', 'purchNumItems'])
    # purchase from 1 to 1,000,000
    for purch_id in range(1, 1000000+1):
        buyer_id = (purch_id - 1) // 100 + 1 # Every buyer gets 100 purchases
        purch_price = round(random.uniform(10, 100), 2) # Random purchase price
        num_items = random.randint(1, 10) # Random number of items

        # Write the purchase data to the CSV file
        writer.writerow([purch_id, buyer_id, purch_price, num_items])
```

Results after running the code and save it as a CSV files

	A	B	C	D	E
1	BuyerID	BuyerName	BuyerAge	BuyerGender	BuyerSalary
2	1	Robert Sol	15	male	5420.65
3	2	Kristin Lev	49	female	4930.35
4	3	Nicolas La	67	male	10379.51
5	4	Rebecca S	26	female	7364.33
6	5	Sonya Slo	23	female	10416.85
7	6	Eric Hart	27	male	8865.11

	A	B	C	D	E
1	purchID	BuyerID	purchPrice	purchNumItems	
2	1	1	17.08	5	
3	2	1	42.34	8	
4	3	1	13.81	10	
5	4	1	13.59	5	
6	5	1	66.94	1	
7	6	1	77.24	1	
8	7	1	71.12	5	
9	8	1	20.64	1	

Step 2: Upload the created dataset into HDFS!

Copying files from host into docker container

```
C:\Users\MANSOUR>docker cp C:\Users\MANSOUR\Downloads\Buyers.csv
cdss321container:/home/cdss321/dataFromHost/
Successfully copied 374kB to cdss321container:/home/cdss321/data
FromHost/

C:\Users\MANSOUR>docker cp C:\Users\MANSOUR\Downloads\Purchases
.csv cdss321container:/home/cdss321/dataFromHost/
Successfully copied 20.8MB to cdss321container:/home/cdss321/dat
aFromHost/

C:\Users\MANSOUR>|
```

```
-rwxr-xr-x 1 root root 364K Oct 22 17:52 Buyers.csv  
-rwxr-xr-x 1 root root 20M Oct 23 14:57 Purchases.csv  
→ dataFromHost
```

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	cdss321	supergroup	363.8 KB	10/26/2024, 2:40:39 AM	1	128 MB	Buyers.csv
-rw-r--r--	cdss321	supergroup	19.82 MB	10/26/2024, 2:36:39 AM	1	128 MB	Purchases.csv

Step 3: Writing mapReduce Jobs!

First Job (mapOnly) Mapper:

```
12 public class BuyersAge {  
13     public static class AgeFilter extends Mapper<Object, Text, Text, Text>{  
14         private boolean isHeader = true; // To handle header row  
15         public void map(Object key, Text value, Context context) throws IOException, InterruptedException {  
16             String record = value.toString();  
17             if (isHeader) {  
18                 isHeader = false; // Skip the first row (header)  
19                 return;  
20             }  
21  
22             String[] data = record.split(",");  
23             String id =data[0].trim();  
24             String name =data[1].trim();  
25             String gender =data[3].trim();  
26             String salary =data[4].trim();  
27  
28             try {  
29                 int age = Integer.parseInt(data[2].trim());  
30                 if (age >= 20 && age <= 50) {  
31                     context.write(new Text(id),  
32                         new Text(name + "," + age + "," + gender + "," + salary));  
33                 }  
34             } catch (NumberFormatException e) {  
35                 System.out.println("Number Format Exception");  
36             }  
37         }  
38     }  
39 }
```

Job Configuration:

```
54 public static void main(String[] args) throws Exception {  
55     Configuration conf = new Configuration();  
56     Job job = Job.getInstance(conf, "Buyers whose Age between 20 and 50");  
57     job.setJarByClass(BuyersAge.class);  
58     job.setMapperClass(AgeFilter.class);  
59     job.setOutputKeyClass(Text.class);  
60     job.setOutputValueClass(Text.class);  
61     FileInputFormat.addInputPath(job, new Path(args[0]));  
62     FileOutputFormat.setOutputPath(job, new Path(args[1]));  
63     System.exit(job.waitForCompletion(true) ? 0 : 1);  
64 }  
65 }
```

Second Job (mapReduce)

Mapper:

```
11 public class Purchases {  
12  
13     public static class PurchasesMapper extends Mapper<Object, Text, Text, Text> {  
14         private boolean isHeader = true; // To handle header row  
15  
16         public void map(Object key, Text value, Context context) throws IOException, InterruptedException {  
17  
18             String[] data = value.toString().split(",");  
19  
20             if (isHeader) {  
21                 isHeader = false; // Skip the first row (header)  
22                 return;  
23             }  
24  
25             String BuyerID = data[1];  
26             float purchPrice;  
27  
28             try {  
29                 purchPrice = Float.parseFloat(data[2]);  
30             } catch (NumberFormatException e) {  
31                 System.out.println("Number Format Exception");  
32                 return;  
33             }  
34  
35             context.write(new Text(BuyerID), new Text("1,"+String.format("%.2F", purchPrice)));  
36  
37         }  
38     }  
39 }
```

Reducer:

```
40 public static class PurchaseReducer extends Reducer<Text,Text, Text, Text> {  
41     private Text result = new Text();  
42  
43     public void reduce(Text key, Iterable<Text> values, Context context) throws IOException  
44     , InterruptedException {  
45         int CountPurch = 0;  
46         float purchPrice = 0;  
47  
48  
49         for (Text value : values) {  
50             String[] parts = value.toString().split(",");  
51             CountPurch += Integer.parseInt(parts[0]);  
52             purchPrice += Float.parseFloat(parts[1]);  
53  
54         }  
55  
56  
57         context.write(key, new Text( CountPurch+","+ String.format("%.2f", purchPrice)));  
58     }  
59 }
```

Job Configuration:

```
public static void main(String[] args) throws Exception {  
    Configuration conf = new Configuration();  
    Job job = Job.getInstance(conf, "Number of purchases and Count");  
    job.setJarByClass(Purchases.class);  
  
    job.setMapperClass(PurchasesMapper.class);  
    job.setCombinerClass(PurchaseReducer.class);  
    job.setReducerClass(PurchaseReducer.class);  
  
    job.setOutputKeyClass(Text.class);  
    job.setOutputValueClass(Text.class);  
  
    FileInputFormat.addInputPath(job, new Path(args[0]));  
    FileOutputFormat.setOutputPath(job, new Path(args[1]));  
    System.exit(job.waitForCompletion(true) ? 0 : 1);  
}
```

Combiner

Then package the Hadoop projects into a JAR file and place it in the containers

```
-rwxr-xr-x 1 root    root 4.8K Oct 26 10:37 job1-0-1.0-SNAPSHOT.jar
-rwxr-xr-x 1 root    root 6.1K Oct 26 10:38 job2-0-1.0-SNAPSHOT.jar
```

Running Jobs on Hadoop Cluster

Creating Folders:

```
-mkdir: UNKNOWN command
➔ ~ hdfs dfs -mkdir /user/cdss321/AssigInput
➔ ~ hdfs dfs -mkdir /user/cdss321/AssigOutput
➔ ~ |
```

Uploading Files from local storage to HDFS:

```
→ ~ hdfs dfs -put /home/cdss321/dataFromHost/Buyers.csv /user/c  
dss321/AssigInput  
→ ~ hdfs dfs -put /home/cdss321/dataFromHost/Purchases.csv /use  
r/cdss321/AssigInput  
→ ~ hdfs dfs -put /home/cdss321/dataFromHost/job1-0-1.0-SNAPSHO  
T.jar /user/cdss321/AssigInput  
→ ~ hdfs dfs -put /home/cdss321/dataFromHost/job2-0-1.0-SNAPSHO  
T.jar /user/cdss321/AssigInput  
→ ~ |
```

Browse Directory

/user/cdss321/AssigInput

Get

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	cdss321	supergroup	363.6 KB	10/26/2024, 2:51:04 PM	1	128 MB	Buyers.csv
-rw-r--r--	cdss321	supergroup	10.82 MB	10/26/2024, 2:52:42 PM	1	128 MB	Purchases.csv
-rw-r--r--	cdss321	supergroup	4.78 KB	10/26/2024, 2:54:36 PM	1	128 MB	job1-0-1.0-SNAPSHOT.jar
-rw-r--r--	cdss321	supergroup	6.09 KB	10/26/2024, 2:55:33 PM	1	128 MB	job2-0-1.0-SNAPSHOT.jar

Running Second Job on Cluster:

```

Bytes Written=180274
~
~ hadoop jar /home/cdss321/dataFromHost/job2-0-1.0-SNAPSHOT.
jar Purchases /user/cdss321/AssigInput/Purchases.csv /user/cdss3
21/AssigOutput/resultJob2
24/10/26 12:27:24 INFO Configuration.deprecation: session.id is
deprecated. Instead, use dfs.metrics.session-id
24/10/26 12:27:24 INFO jvm.JvmMetrics: Initializing JVM Metrics
with processName=JobTracker, sessionId=
24/10/26 12:27:24 WARN mapreduce.JobResourceUploader: Hadoop com
mand-line option parsing not performed. Implement the Tool inter
face and execute your application with ToolRunner to remedy this
24/10/26 12:27:24 INFO input.FileInputFormat: Total input paths
to process : 1
24/10/26 12:27:24 INFO mapreduce.JobSubmitter: number of splits:
1
24/10/26 12:27:25 INFO mapreduce.JobSubmitter: Submitting tokens
for job: job_local309071870_0001
24/10/26 12:27:25 INFO mapreduce.Job: The url to track the job:
http://localhost:8080/
24/10/26 12:27:25 INFO mapreduce.Job: Running job: job_local3090
71870_0001
24/10/26 12:27:25 INFO mapred.LocalJobRunner: OutputCommitter se
t in config null
24/10/26 12:27:25 INFO output.FileOutputCommitter: File Output C
ommitter Algorithm version is 1
24/10/26 12:27:25 INFO mapred.LocalJobRunner: OutputCommitter is
org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
24/10/26 12:27:25 INFO mapred.LocalJobRunner: Waiting for map ta
sks
24/10/26 12:27:25 INFO mapred.LocalJobRunner: Starting task: att
empt_local309071870_0001_m_000000_0
24/10/26 12:27:25 INFO output.FileOutputCommitter: File Output C

```

Browse Directory

/user/cdss321/AssigOutput/resultJob2 Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	cdss321	supergroup	0 B	10/26/2024, 3:27:28 PM	1	128 MB	_SUCCESS
-rw-r--r--	cdss321	supergroup	164.94 KB	10/26/2024, 3:27:28 PM	1	128 MB	part-r-000000

Hadoop, 2018.

Copying files from HDFS to the local Machine:

```

➤ ~ hdfs dfs -get /user/cdss321/AssigOutput/resultJob1 /home/cd
ss321/dataFromHost/
➤ ~ hdfs dfs -get /user/cdss321/AssigOutput/resultJob2 /home/cd
ss321/dataFromHost/
get: mkdir '/home/cd': Input/output error
zsh: no such file or directory: ss321/dataFromHost/
➤ ~ hdfs dfs -get /user/cdss321/AssigOutput/resultJob2 /home/cd
ss321/dataFromHost/
➤ ~ cd dataFromHost
➤ dataFromHost ll
total 21M
-rwxr-xr-x 1 root    root 364K Oct 26 11:27 Buyers.csv
-rwxr-xr-x 1 root    root 28M Oct 26 11:28 Purchases.csv
-rwxr-xr-x 1 root    root 5.5K Sep 13 11:50 WordCount.jar
-rwxr-xr-x 1 root    root 1.2K Sep 13 11:49 data.txt
-rwxr-xr-x 1 root    root 4.8K Oct 26 10:37 job1-0-1.0-SNAPSHOT.
jar
-rwxr-xr-x 1 root    root 6.1K Oct 26 10:38 job2-0-1.0-SNAPSHOT.
jar
drwxr-xr-x 2 cdss321 root 4.0K Sep 16 08:18 result
drwxr-xr-x 2 cdss321 root 4.0K Oct 26 12:30 resultJob1
drwxr-xr-x 2 cdss321 root 4.0K Oct 26 12:32 resultJob2
➤ dataFromHost |

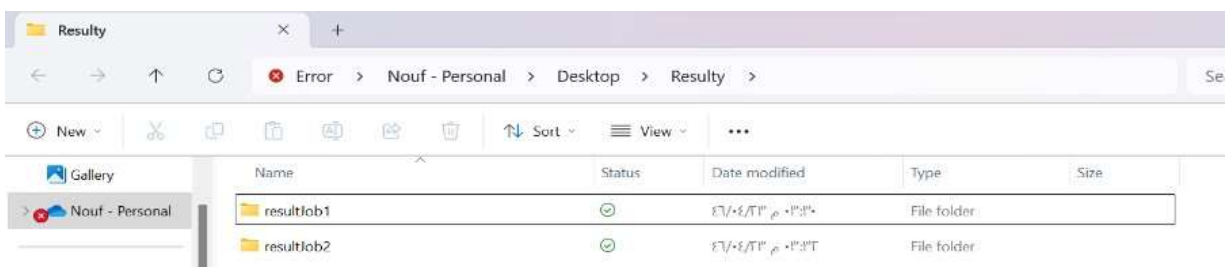
```

Copying files from docker to the host

```

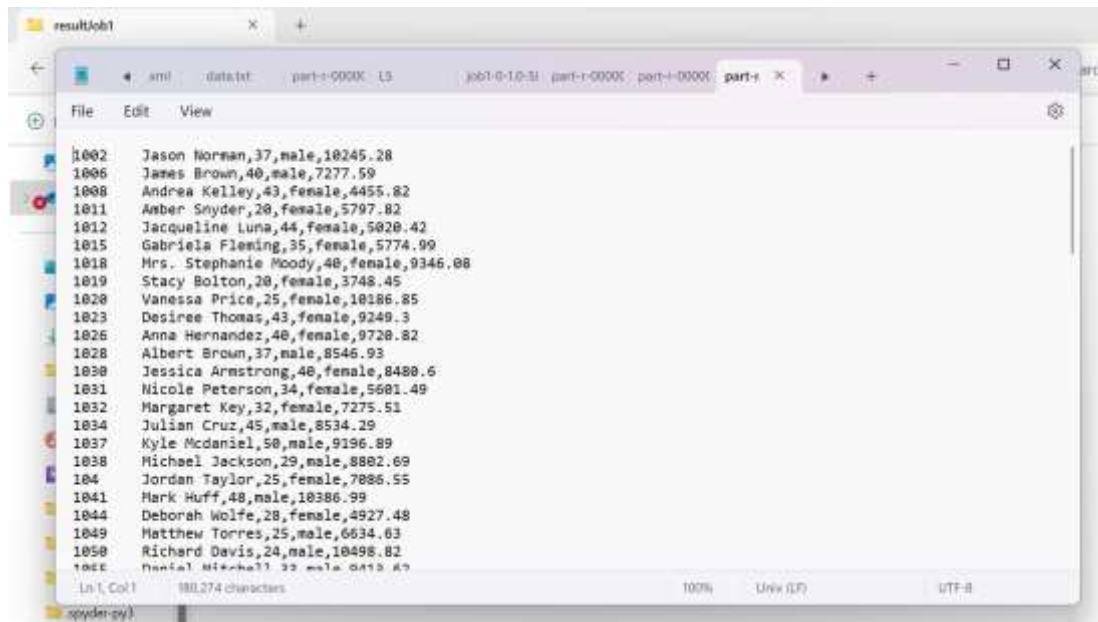
C:\Users\MANSOUR>docker cp cdss321container:/home/cdss321/dataFromHost/resultJob1/ "C:\Users\MANSOUR\OneDrive\المكتب\Resulty"
Successfully copied 183kB to C:\Users\MANSOUR\OneDrive\المكتب\Resulty
C:\Users\MANSOUR>docker cp cdss321container:/home/cdss321/dataFromHost/resultJob2/ "C:\Users\MANSOUR\OneDrive\المكتب\Resulty"
Successfully copied 172kB to C:\Users\MANSOUR\OneDrive\المكتب\Resulty
C:\Users\MANSOUR>

```



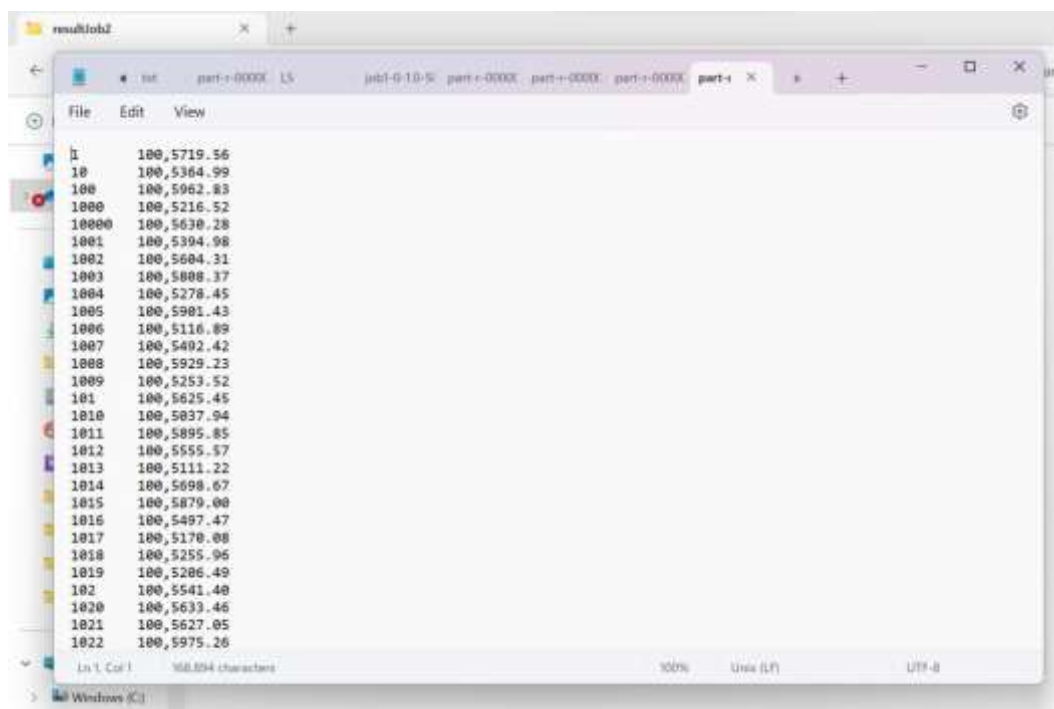
Name	Status	Date modified	Type	Size
resultJob1	✓	٤٦/١٠/٢٣ م ١٢:٣٠	File folder	
resultJob2	✓	٤٦/١٠/٢٣ م ١٢:٣٢	File folder	

Job one Output:



```
1002 Jason Norman,37,male,10245.28
1006 James Brown,40,male,7277.59
1008 Andrea Kelley,43,female,4455.82
1011 Amber Snyder,20,female,5797.82
1012 Jacqueline Luna,44,female,5020.42
1015 Gabriela Fleming,35,female,5774.99
1018 Mrs. Stephanie Moody,40,female,9346.08
1019 Stacy Bolton,20,female,3748.45
1020 Vanessa Price,25,female,10186.85
1023 Desiree Thomas,43,female,9249.3
1026 Anna Hernandez,40,female,9720.82
1028 Albert Brown,37,male,8546.93
1030 Jessica Armstrong,40,female,8480.6
1031 Nicole Peterson,34,female,5601.49
1032 Margaret Key,32,female,7275.51
1034 Julian Cruz,45,male,8534.29
1037 Kyle McDaniel,50,male,9196.89
1038 Michael Jackson,29,male,8802.69
104 Jordan Taylor,25,female,7086.55
1041 Mark Huff,48,male,10386.99
1044 Deborah Wolfe,28,female,4927.48
1049 Matthew Torres,25,male,6634.63
1050 Richard Davis,24,male,10498.82
1056 Daniel Mitchell,32,male,9413.67
```

Job two Output:



```
1 100,5719.56
10 100,5364.99
100 100,5962.83
1000 100,5216.52
10000 100,5630.28
1001 100,5394.98
1002 100,5604.31
1003 100,5808.37
1004 100,5278.45
1005 100,5901.43
1006 100,5116.89
1007 100,5492.42
1008 100,5929.23
1009 100,5253.52
101 100,5625.45
1010 100,5037.94
1011 100,5895.85
1012 100,5555.57
1013 100,5111.22
1014 100,5698.67
1015 100,5879.00
1016 100,5497.47
1017 100,5170.08
1018 100,5255.96
1019 100,5206.49
102 100,5541.40
1020 100,5633.46
1021 100,5627.05
1022 100,5975.26
```