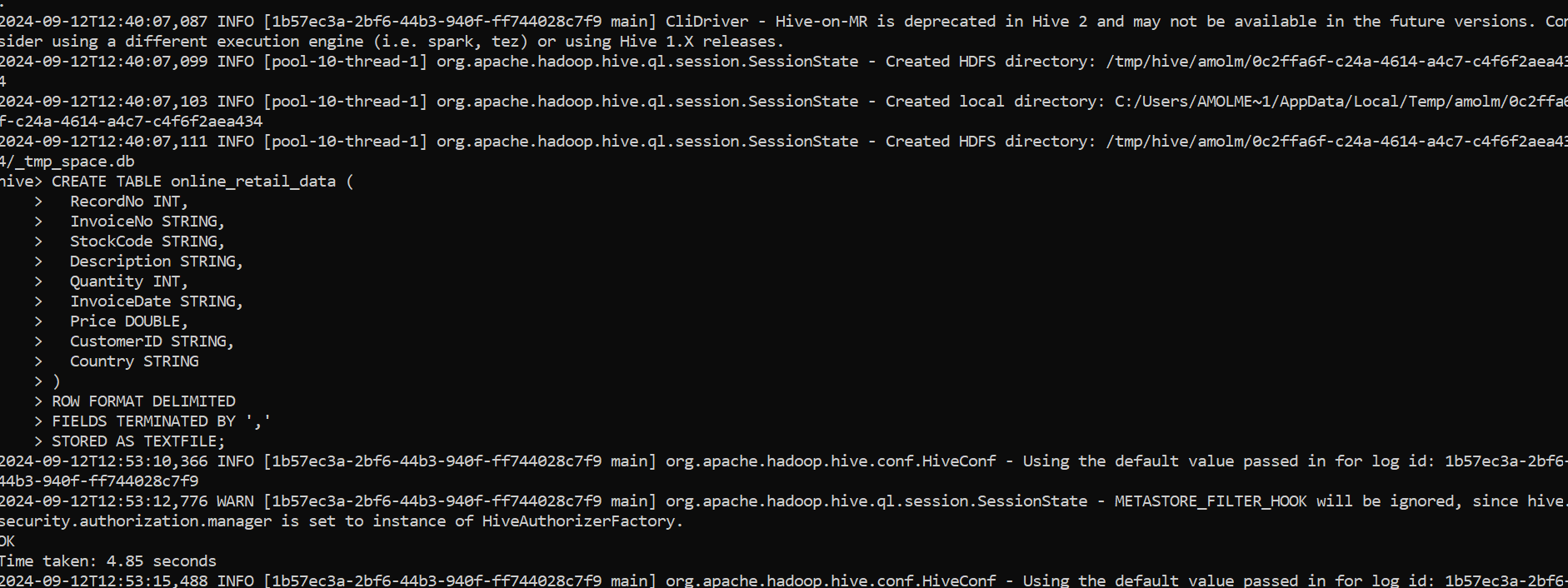
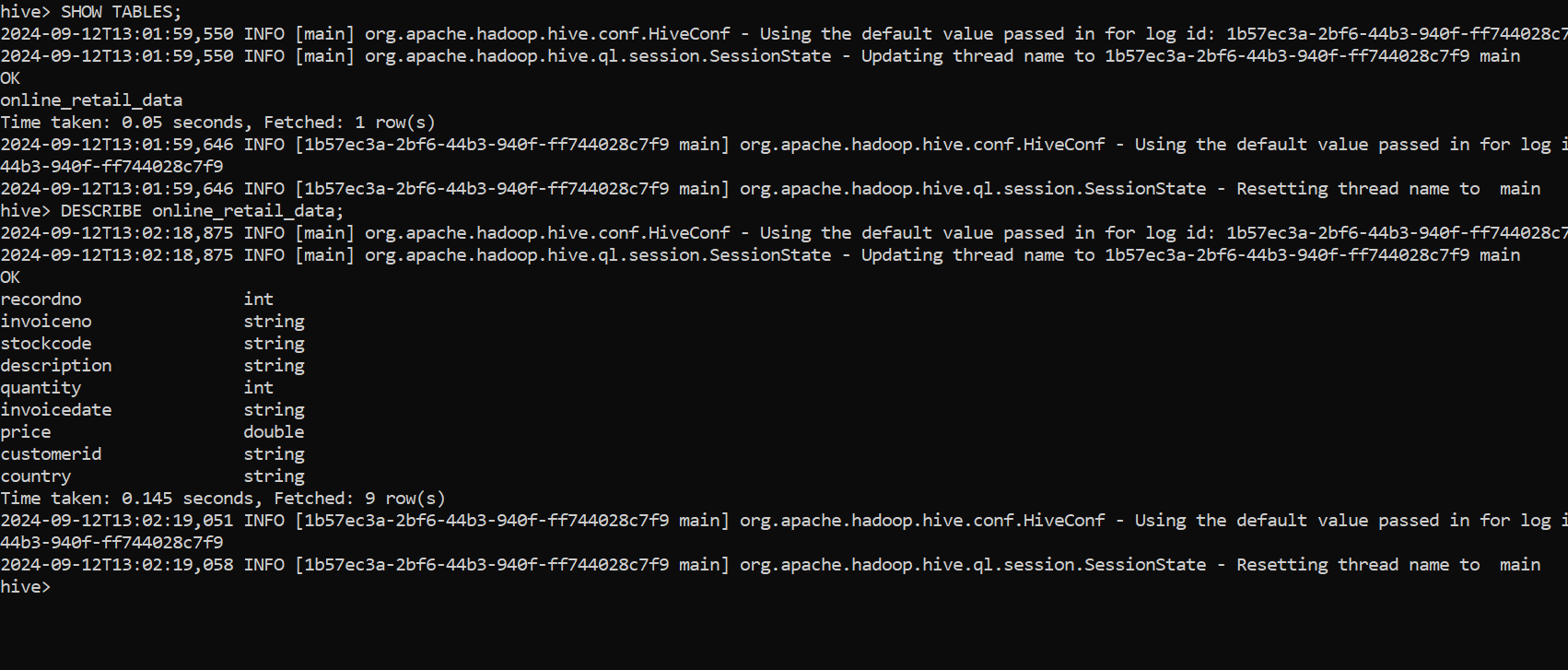
**ASSIGNMENT 2**

|  |  |  |
| --- | --- | --- |
| **BDS GROUP 32** | | |
| **NAME** | **BITS ID** | **CONTRIBUTION** |
| AMOL MEHRA | 2022DC04286 | 100% |
| SURAPANENI MANOGNI | 2022dc04257 | 0% |
| HARSHAVARDHAN MOGALGIDDI | 2022dc04166 | 0% |

* The assignment has been performed with Hadoop and queries are run through Hive. The configuration files of hadoop and hive are attached with the assignment.
* Installation Requirements:-
  + JDK 1.8
  + Hadoop 3.2.4
  + Derby 10.4.2.0
  + Hive 3.1.2
* Commands to run for starting Hadoop(in Admin mode):
  + hdfs namenode -format
  + start-all.cmd
* Command to run derby(in Admin mode):
  + StartNetworkServer -h 0.0.0.0
* Command to run hive(in Admin mode):
  + hive --service schematool --dbType derby -initSchema
* The data afterwards is loaded to hdfs using command:
  + hdfs dfs -put C:/data/online\_retail\_data.csv /user/<username>/
* The table structureis created under hive to load data.

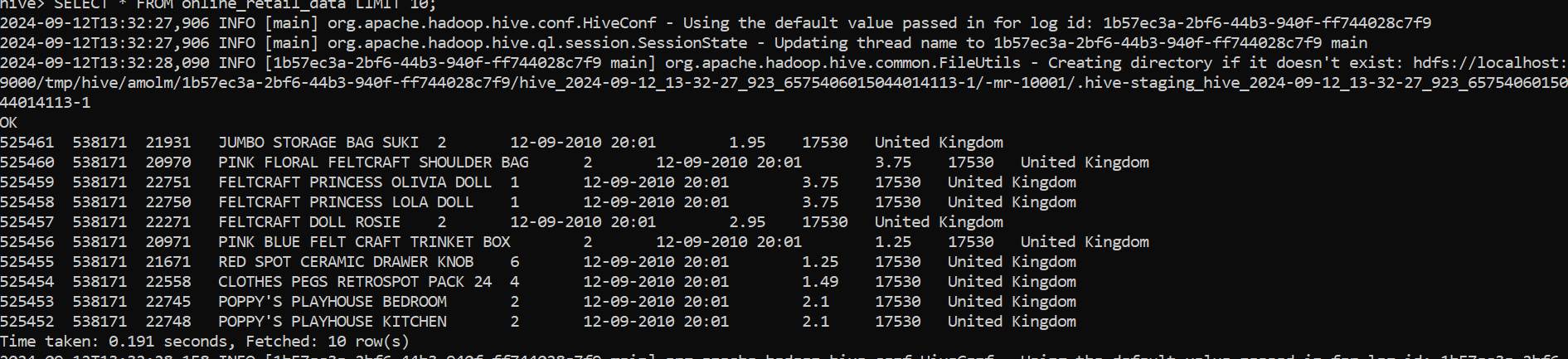
CREATE TABLE online\_retail\_data (RecordNo INT,InvoiceNo STRING,StockCode STRING,Description STRING,Quantity INT,InvoiceDate STRING,Price DOUBLE,CustomerID STRING,Country STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ','STORED AS TEXTFILE;





* The data is loaded into the table by using below hive command:

LOAD DATA INPATH 'online\_retail\_data.csv' INTO TABLE online\_retail\_data;



**HIVE QUERIES**

**2. Country with Maximum Revenue in March 2010**

**SQL QUERY:**

SELECT

Country,

SUM(Quantity \* Price) AS total\_revenue

FROM

online\_retail\_data

WHERE

FROM\_UNIXTIME(UNIX\_TIMESTAMP(REGEXP\_REPLACE(InvoiceDate, '-', '/'), 'MM/dd/yyyy HH:mm')) BETWEEN '2010-03-01' AND '2010-03-31'

GROUP BY

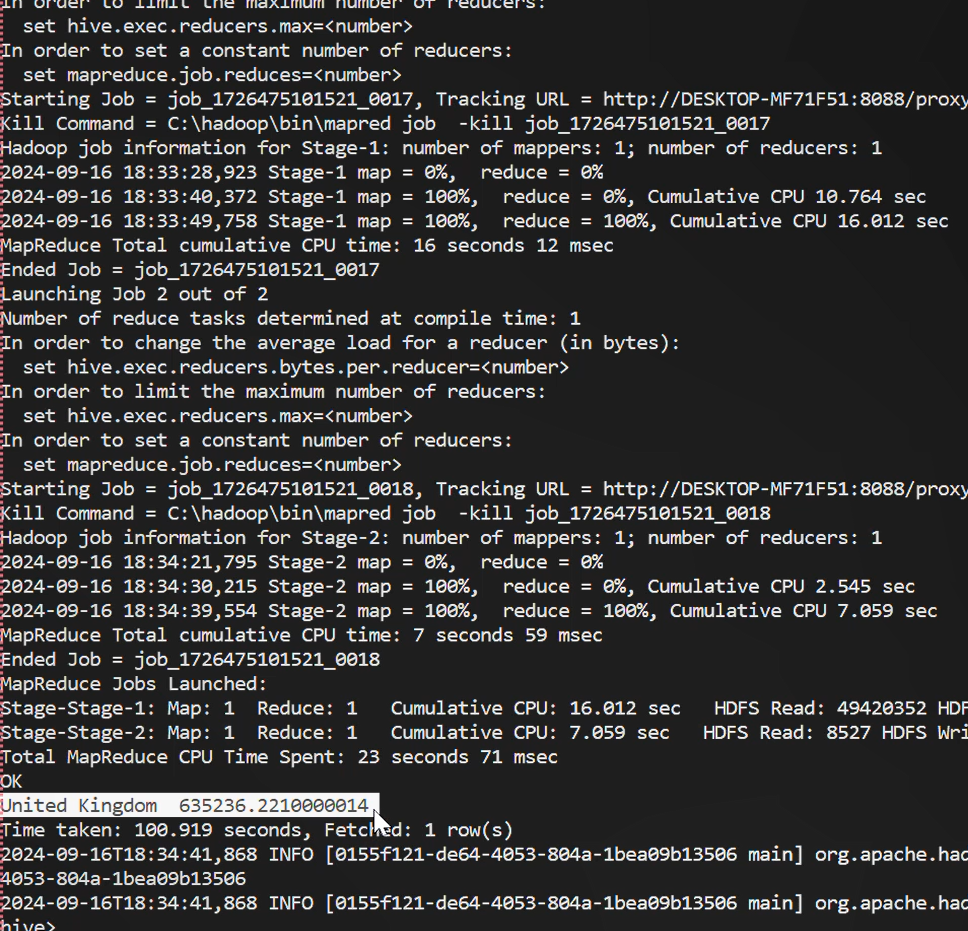
Country

ORDER BY

total\_revenue DESC

LIMIT 1;

**RESULT:** United Kingdom 635236.2210000014

****

**3. Month in 2010 with Maximum Number of Items Sold**

**SQL QUERY:**

SELECT

MONTH(FROM\_UNIXTIME(UNIX\_TIMESTAMP(REGEXP\_REPLACE(InvoiceDate, '-', '/'), 'MM/dd/yyyy HH:mm'))) AS month,

SUM(quantity) AS total\_items\_sold

FROM online\_retail\_data

WHERE

YEAR(FROM\_UNIXTIME(UNIX\_TIMESTAMP(REGEXP\_REPLACE(InvoiceDate, '-', '/'), 'MM/dd/yyyy HH:mm'))) = 2010

GROUP BY

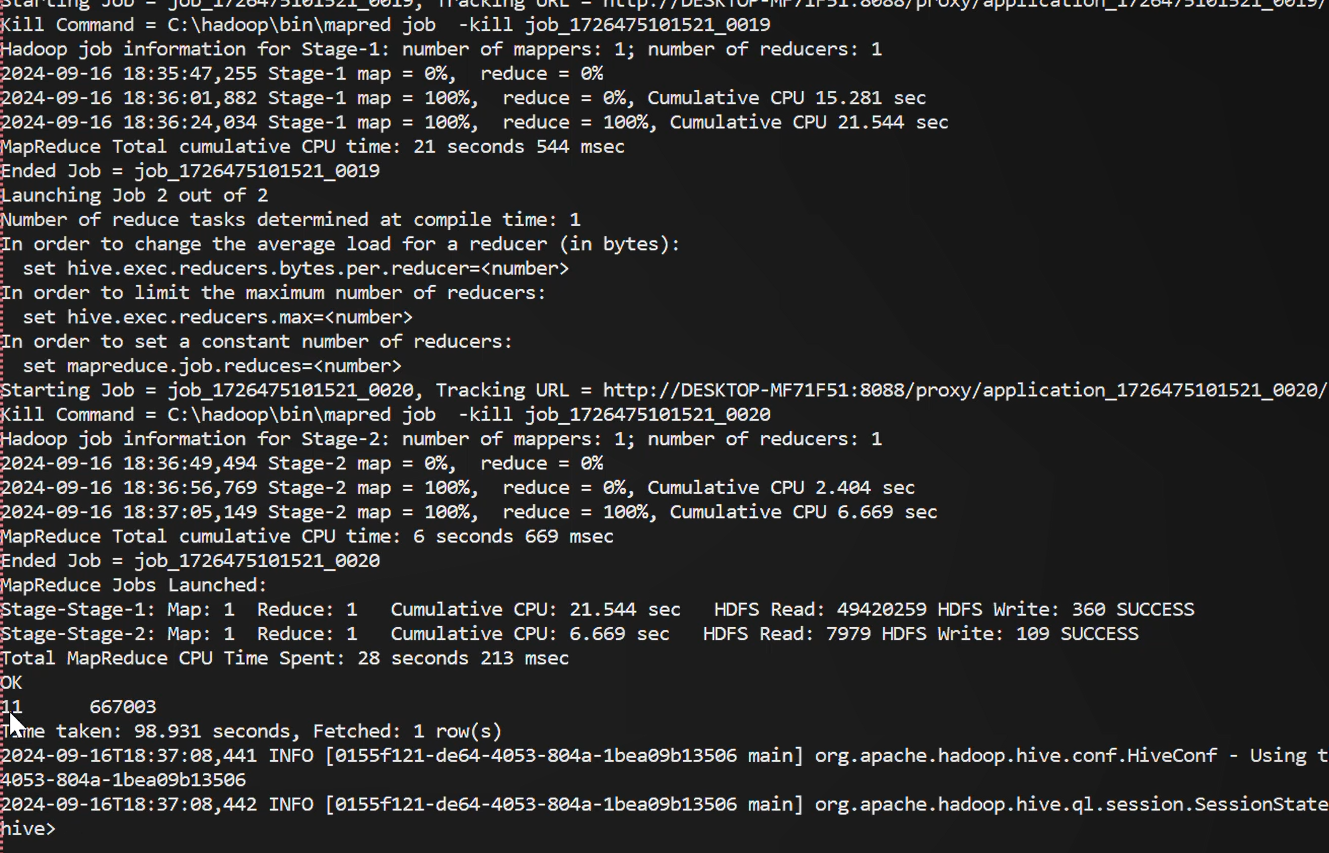
MONTH(FROM\_UNIXTIME(UNIX\_TIMESTAMP(REGEXP\_REPLACE(InvoiceDate, '-', '/'), 'MM/dd/yyyy HH:mm')))

ORDER BY

total\_items\_sold DESC

LIMIT 1;

**RESULT:** 11 , 667003

****

**4.Country with Maximum Number of Items Sold in January 2010**

**SQL QUERY:**

SELECT

Country,

SUM(Quantity) AS total\_items\_sold

FROM online\_retail\_data

WHERE

FROM\_UNIXTIME(UNIX\_TIMESTAMP(REGEXP\_REPLACE(InvoiceDate, '-', '/'), 'MM/dd/yyyy HH:mm')) BETWEEN '2010-01-01' AND '2010-01-31'

GROUP BY

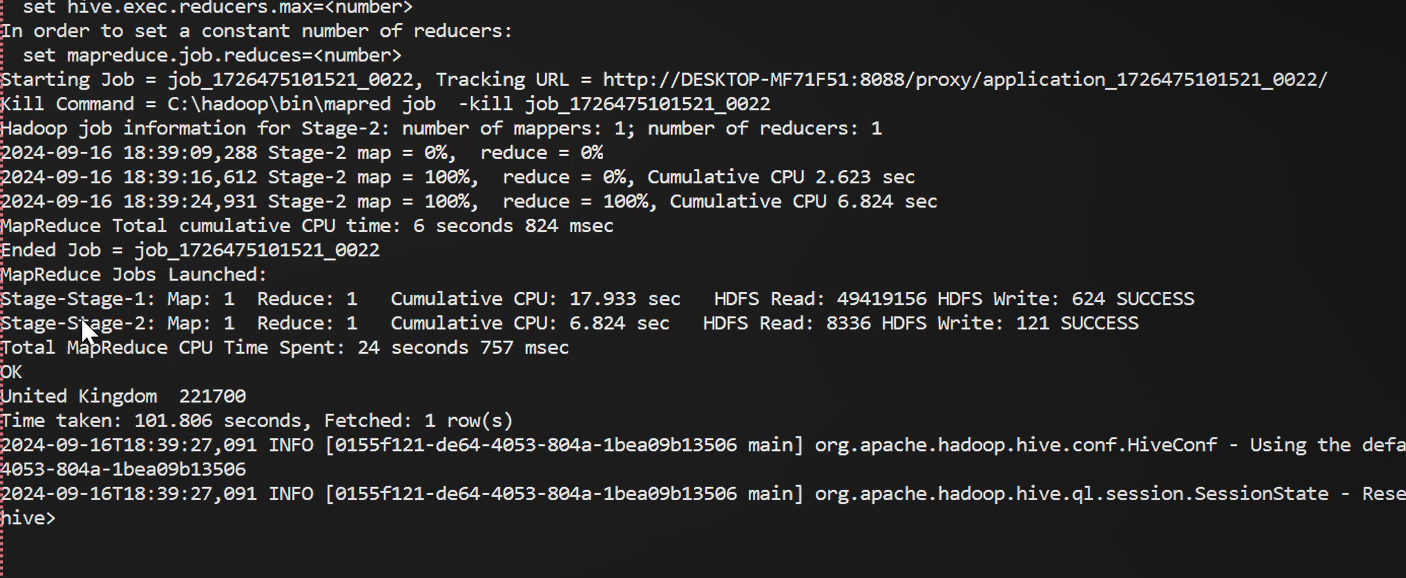
Country

ORDER BY

total\_items\_sold DESC

LIMIT 1;

**RESULT:** United Kingdom 221700

****

**5. StockCode of the Item with Highest Number of Sales in the Given Country in 2010**

**SQL QUERY:**

SELECT

StockCode,

SUM(Quantity) AS total\_items\_sold

FROM

online\_retail\_data

WHERE

Country = 'United Kingdom'

AND YEAR(FROM\_UNIXTIME(UNIX\_TIMESTAMP(REGEXP\_REPLACE(InvoiceDate, '-', '/'), 'MM/dd/yyyy HH:mm'))) = 2010

GROUP BY

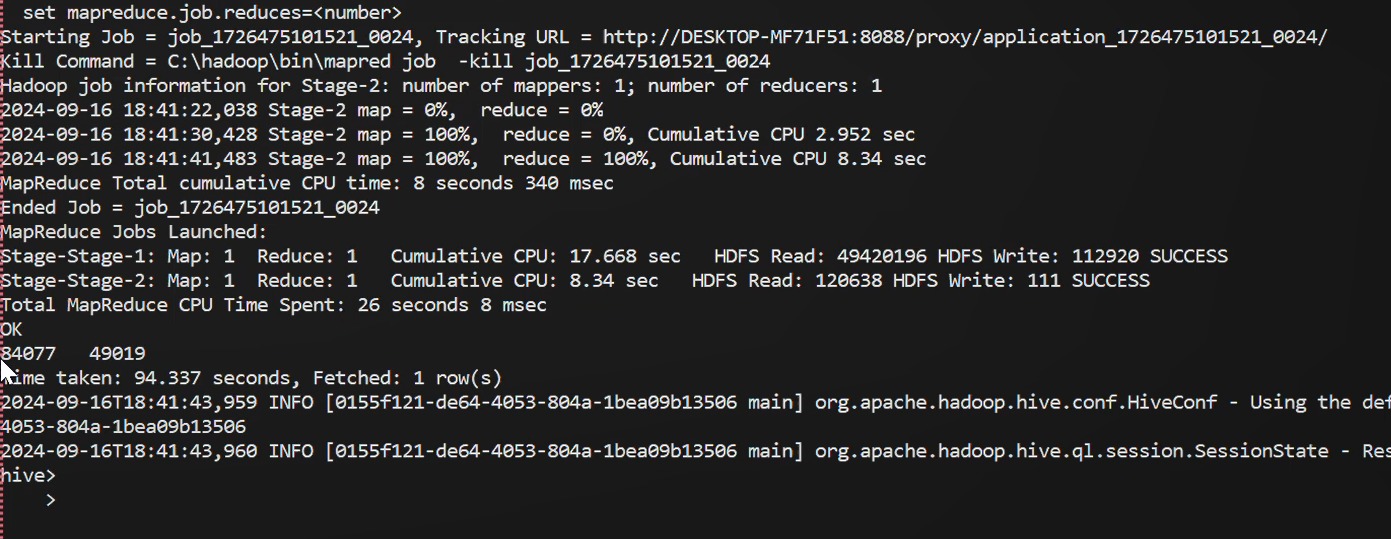
StockCode

ORDER BY

total\_items\_sold DESC

LIMIT 1;

**RESULT:** 84077, 49019

****

**6. StockCode of the Item for Which Maximum Revenue was Received in December 2010**

**SQL QUERY:**

SELECT

StockCode,

SUM(Quantity \* Price) AS total\_revenue

FROM online\_retail\_data

WHERE

FROM\_UNIXTIME(UNIX\_TIMESTAMP(REGEXP\_REPLACE(InvoiceDate, '-', '/'), 'MM/dd/yyyy HH:mm')) BETWEEN '2010-12-01' AND '2010-12-31'

GROUP BY

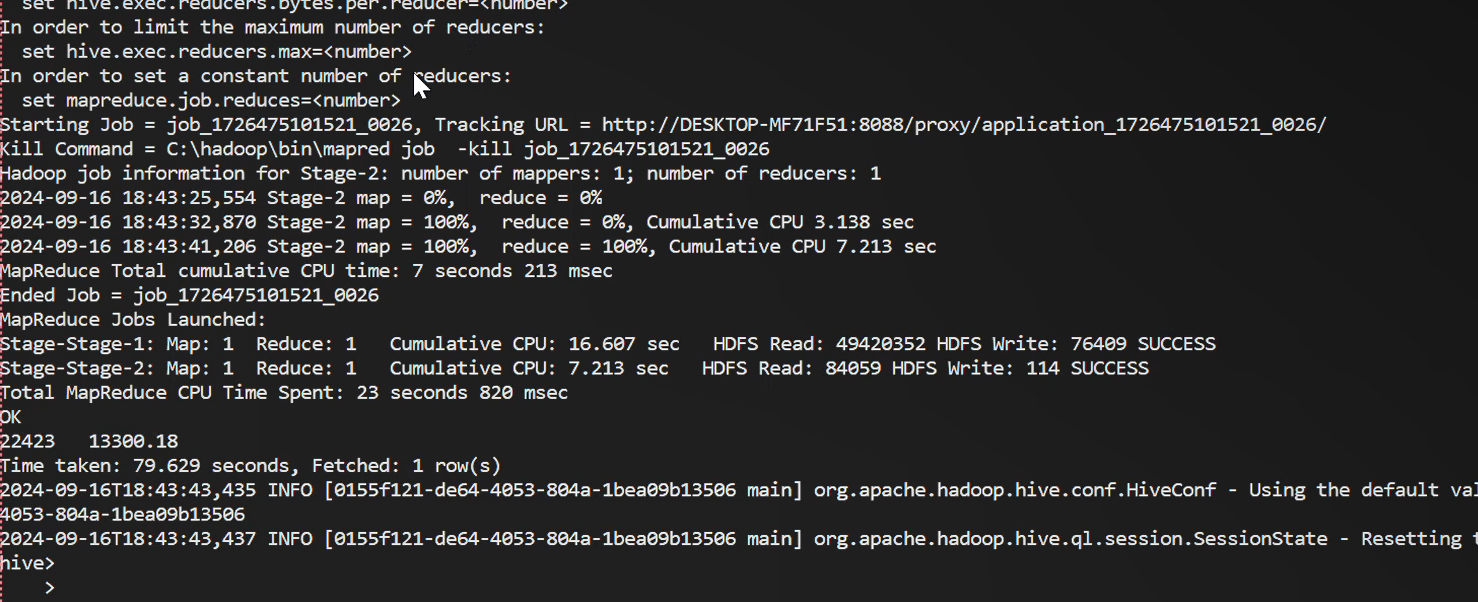
StockCode

ORDER BY

total\_revenue DESC

LIMIT 1;

**RESULT:** 22423 13300.18

****

**7. Country with Minimum Number of Sales in 2010**

**SQL QUERY:**

SELECT

Country,

SUM(Quantity) AS total\_items\_sold

FROM

online\_retail\_data

WHERE

YEAR(FROM\_UNIXTIME(UNIX\_TIMESTAMP(REGEXP\_REPLACE(InvoiceDate, '-', '/'), 'MM/dd/yyyy HH:mm'))) = 2010

GROUP BY

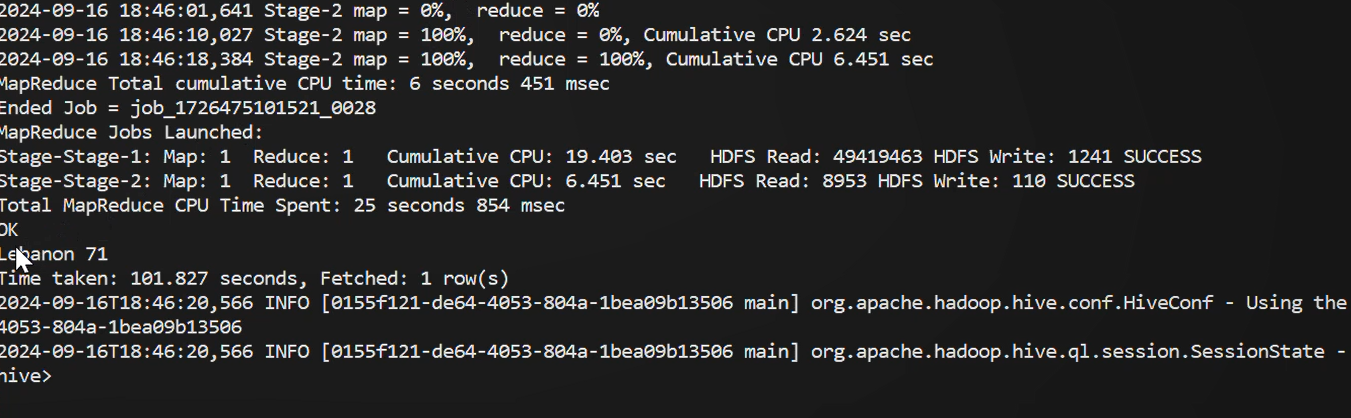
Country

ORDER BY

total\_items\_sold ASC

LIMIT 1;

**RESULT: Lebanon 71**

****

**MAPREDUCE FUNCTION**

1. **Total Number of Unique Customers in the Given Country.**

**MAPREDUCE FUNCTION IN PYTHON:**

**MAPPER.PY**

#!/usr/bin/env python

import sys

for line in sys.stdin:

line = line.strip()

fields = line.split(',')

# Skip header row

if fields[0] == 'Record number':

continue

country = fields[7]

invoice = fields[1]

# Emit (country, invoice) if the country is 'United Kingdom'

if country == 'United Kingdom':

print(f'{invoice}\t1')

**REDUCER.PY**

#!/usr/bin/env python

import sys

current\_invoice = None

unique\_customers = set()

for line in sys.stdin:

line = line.strip()

invoice, count = line.split('\t')

# Track unique invoices (customers)

if invoice != current\_invoice:

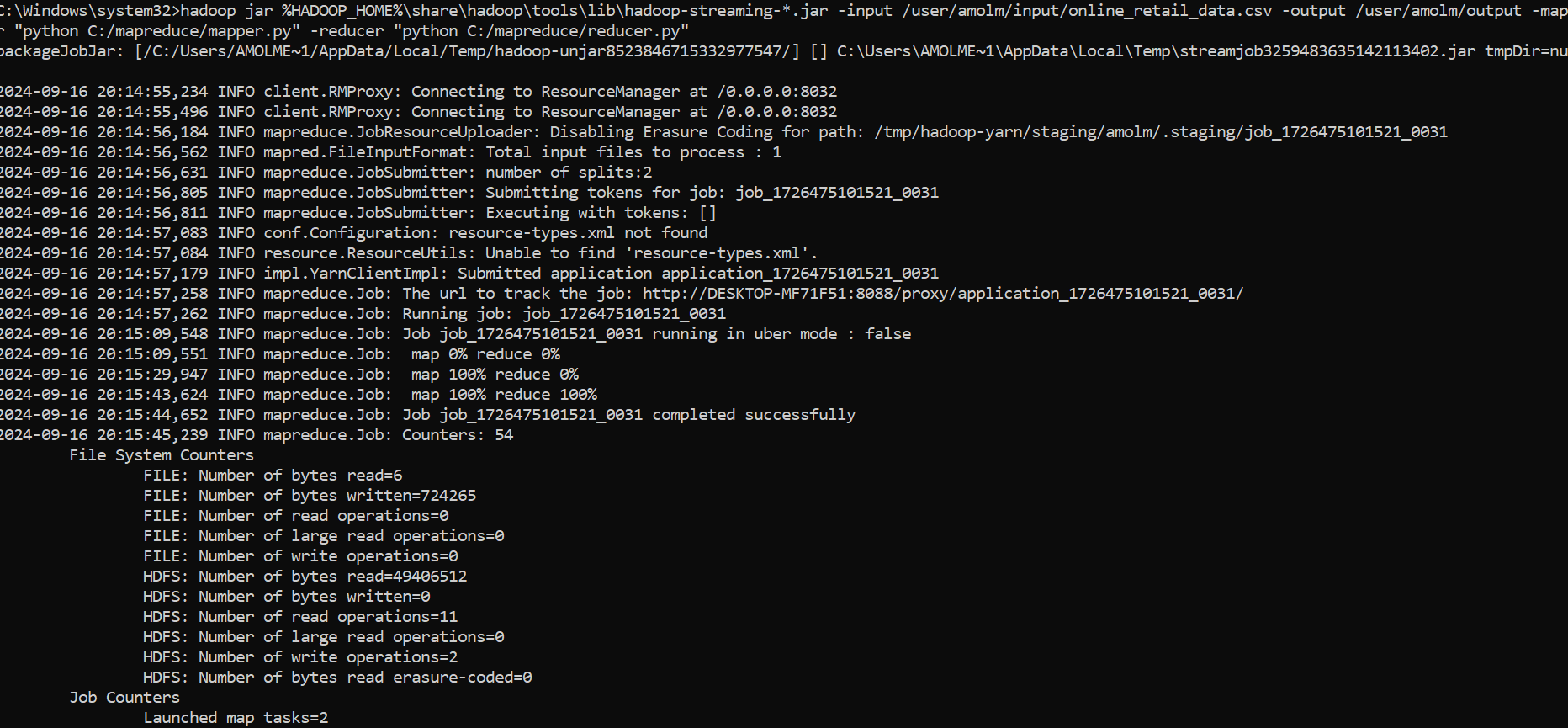
unique\_customers.add(invoice)

current\_invoice = invoice

# Output the total number of unique invoices (customers)

print(len(unique\_customers))

* **Commands:**
  + hadoop fs -mkdir /user/<username>/input
  + hadoop fs -mkdir /user/<username>/output
  + hadoop fs -put C:/data/online\_retail\_data.csv /user/<username>/input
  + hadoop jar %HADOOP\_HOME%\share\hadoop\tools\lib\hadoop-streaming-\*.jar -input /user/<username>/input/online\_retail\_data.csv -output /user/<username>/output -mapper "python C:/mapreduce/mapper.py" -reducer "python C:/mapreduce/reducer.py"
  + hadoop fs -cat /user/amolm/output/part-00000

****

**Output: 4035**