**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙠🙟🕮🙝🙢



**MÔN HỌC: BIG DATA ANALYST**

**ĐỒ ÁN CUỐI KÌ**

**ĐỀ TÀI: ỨNG DỤNG APACHE SPARK VÀ APACHE PIG VÀO PHÂN TÍCH DỮ LIỆU**

**GVHD**: Lê Thị Minh Châu

**Nhóm sinh viên thực hiện:** Nhóm 2

1. Nguyễn Trí Dũng 20133029
2. Nguyễn Khoa Quang Thắng 20133090

**Mã môn học**: BDAN333977\_22\_2\_01

Thành phố Hồ Chí Minh, Tháng 5 năm 2023

MỤC LỤC

[BẢNG PHÂN CÔNG 4](#_Toc136151925)

[PHẦN 1 – GIỚI THIỆU: 5](#_Toc136151926)

[1.1) Các công cụ được dùng trong đề tài: 5](#_Toc136151927)

[1.2) Apache Pig: 6](#_Toc136151928)

[1.3) Apache Spark: 9](#_Toc136151929)

[PHẦN 2 – DỮ LIỆU 12](#_Toc136151930)

[2.1) Nguồn gốc của tập dữ liệu: 12](#_Toc136151931)

[2.2) Thông tin về tập dữ liệu: 12](#_Toc136151932)

[PHẦN 3 – APACHE PIG: 14](#_Toc136151933)

[3.1) Chuẩn bị: 14](#_Toc136151934)

[3.2) Thống kê 5 quốc gia mua hàng nhiều nhất: 16](#_Toc136151935)

[3.3) Thống kê 5 transaction có số tiền nhiều nhất: 20](#_Toc136151936)

[3.4) Thống kê 5 sản phẩm có số tiền nhiều nhất: 24](#_Toc136151937)

[3.5) Thống kê doanh thu theo ngày: 27](#_Toc136151938)

[3.6) Thống kê 5 khách hàng chi tiền nhiều nhất: 31](#_Toc136151939)

[3.7) Tạo bảng khách hàng với thông tin số tiền khách hàng đã chi và số lượng hàng khách hàng đã mua từ trước tới giờ: 34](#_Toc136151940)

[PHẦN 4 – APACHE SPARK: 38](#_Toc136151941)

[4.1) Miêu tả: 38](#_Toc136151942)

[4.2) Thuật toán gom cụm Kmeans: 38](#_Toc136151943)

[4.2.1) Giới thiệu thuật toán Kmeans: 38](#_Toc136151944)

[4.2.2) Thực thi thuật toán: 39](#_Toc136151945)

[PHẦN 5 – HIỂN THỊ OUTPUT TRÊN WEB: 44](#_Toc136151946)

[5.1) Top 5 quốc gia: 44](#_Toc136151947)

[5.2) Top 5 sản phẩm được mua nhiều nhất: 45](#_Toc136151948)

[5.3) Top 5 giao dịch nhiều nhất: 47](#_Toc136151949)

[5.4) Tổng doanh thu 10 ngày cao nhất: 48](#_Toc136151950)

[5.5) Top 5 khách hàng mua nhiều nhất: 50](#_Toc136151951)

[5.6) Gom cụm khách hàng: 51](#_Toc136151952)

[TÀI LIỆU THAM KHẢO 54](#_Toc136151953)

# BẢNG PHÂN CÔNG

|  |  |  |  |
| --- | --- | --- | --- |
| Công việc | Chi tiết | Nguyễn Trí Dũng | Nguyễn Khoa Quang Thắng |
| Phân tích data với Apache Pig | Top 5 quốc gia mua nhiều nhất,  Top 5 sản phẩm được mua nhiều nhất |  | X |
| Top 5 giao dịch nhiều nhất,  Tổng doanh thu 10 ngày cao nhất,  Top 5 khách hàng mua nhiều nhất | X |  |
| Phân cụm Kmean với Apache Spark Machine Learing | Gom nhóm data theo từng customer |  | X |
| Phân cụm customer theo thuật toán Kmean sử dụng PySpark | X |  |
| Giao diện, tạo file scripts .bat | Lấy output từ HDFS, làm giao diện |  | X |
| Tạo các file .bat | X |  |

# PHẦN 1 – GIỚI THIỆU:

## 1.1) Các công cụ được dùng trong đề tài:

* **Ngôn ngữ Python:** Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.
* **Visual studio code:** Visual Studio Code là một trình soạn thảo mã nguồn được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có chức năng nổi bật cú pháp (syntax highlighting), tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn.
* **PySpark** là API Python dành cho Apache Spark, một khung điện toán phân tán, mã nguồn mở và bộ thư viện để xử lý dữ liệu quy mô lớn, thời gian thực. Nếu bạn đã quen thuộc với Python và các thư viện như Pandas, thì PySpark là một ngôn ngữ tốt để học để tạo ra các phân tích và quy trình có thể mở rộng hơn.
* **Apache Spark** về cơ bản là một công cụ tính toán hoạt động với các tập dữ liệu khổng lồ bằng cách xử lý chúng theo hệ thống song song và hàng loạt. Spark được viết bằng Scala và PySpark được phát hành để hỗ trợ sự hợp tác giữa Spark và Python. Ngoài việc cung cấp API cho Spark, PySpark giúp bạn giao tiếp với Bộ dữ liệu phân tán đàn hồi (RDD) bằng cách tận dụng thư viện Py4j.
* **Pig** là một dự án nghiên cứu được bắt đầu bởi Yahoo và năm 2006. Hai năm sau, vào năm 2008 Apache Pig được phát hành phiên bản đầu tiên. Apache Pig là một nền tảng ngôn ngữ cấp cao. Để phân tích và truy vấn dữ liệu lớn được lưu trữ trong HDFS của một Hadoop Cluster. Ngoài ra, Pig còn cung cấp một ngôn ngữ cấp cao để viết các chương trình phân tích dữ liệu gọi là Pig Latin, khá giống với SQL. Pig latin cho phép Data Analyst viết các câu truy vấn và phân tích. Thậm chí có thể sử dụng Pig latin để tạo ra ETL pipeline.

## 1.2) Apache Pig:

Các tính năng của Apache Pig:

* Để thực hiện một số thao tác, Apache Pig cung cấp các bộ toán tử phong phú như filtering, joining, sorting, aggregation etc. Đặc biệt đối với lập trình viên SQL, Apache Pig là một phần bổ sung.
* Apache Pig có thể mở rộng để có thể tạo quy trình của riêng mình và các hàm do người dùng xác định (UDF) được viết bằng python, java hoặc các ngôn ngữ lập trình khác.
* Toán tử join rất dễ dàng trong Apache Pig.
* Apache Pig cho phép chia tách trong pipeline.
* Bằng cách tích hợp với các thành phần khác của hệ sinh thái Apache Hadoop, chẳng hạn như Apache Hive, Apache Spark và Apache ZooKeeper, Apache Pig cho phép người dùng tận dụng các khả năng của các thành phần này trong khi chuyển đổi dữ liệu.
* Cấu trúc dữ liệu đa giá trị, lồng nhau và phong phú hơn.
* Pig có thể xử lý việc phân tích cả dữ liệu có cấu trúc và phi cấu trúc.

Pig chạy trong Hadoop có hai chế độ thực thi:

* **Chế độ cục bộ:** Trong chế độ này, ngôn ngữ Hadoop Pig chạy trong một JVM duy nhất và sử dụng hệ thống tệp cục bộ. Chế độ này chỉ thích hợp để phân tích các tập dữ liệu nhỏ bằng Pig trong Hadoop
* **Chế độ Map Reduce:** Trong chế độ này, các truy vấn được viết bằng Pig Latin được dịch thành các công việc MapReduce và được chạy trên một cụm Hadoop (cụm có thể là giả hoặc được phân phối đầy đủ). Chế độ MapReduce với cụm được phân phối đầy đủ rất hữu ích khi chạy Pig trên các tập dữ liệu pig.

## 1.3) Apache Spark:

Apache Spark gồm có 5 thành phần chính : Spark Core, Spark Streaming, Spark SQL, MLlib và GraphX, trong đó:

* **Spark Core** là nền tảng cho các thành phần còn lại và các thành phần này muốn khởi chạy được thì đều phải thông qua Spark Core do Spark Core đảm nhận vai trò thực hiện công việc tính toán và xử lý trong bộ nhớ (In-memory computing) đồng thời nó cũng tham chiếu các dữ liệu được lưu trữ tại các hệ thống lưu trữ bên ngoài.
* **Spark SQL** cung cấp một kiểu data abstraction mới (SchemaRDD) nhằm hỗ trợ cho cả kiểu dữ liệu có cấu trúc (structured data) và dữ liệu nửa cấu trúc (semi-structured data – thường là dữ liệu dữ liệu có cấu trúc nhưng không đồng nhất và cấu trúc của dữ liệu phụ thuộc vào chính nội dung của dữ liệu ấy). Spark SQL hỗ trợ DSL (Domain-specific language) để thực hiện các thao tác trên DataFrames bằng ngôn ngữ Scala, Java hoặc Python và nó cũng hỗ trợ cả ngôn ngữ SQL với giao diện command-line và ODBC/JDBC server.
* **Spark Streaming** được sử dụng để thực hiện việc phân tích stream bằng việc coi stream là các mini-batches và thực hiệc kỹ thuật RDD transformation đối với các dữ liệu mini-batches này. Qua đó cho phép các đoạn code được viết cho xử lý batch có thể được tận dụng lại vào trong việc xử lý stream, làm cho việc phát triển lambda architecture được dễ dàng hơn. Tuy nhiên điều này lại tạo ra độ trễ trong xử lý dữ liệu (độ trễ chính bằng mini-batch duration) và do đó nhiều chuyên gia cho rằng Spark Streaming không thực sự là công cụ xử lý streaming giống như Storm hoặc Flink.
* **MLlib** (Machine Learning Library): MLlib là một nền tảng học máy phân tán bên trên Spark do kiến trúc phân tán dựa trên bộ nhớ. Theo các so sánh benchmark Spark MLlib nhanh hơn 9 lần so với phiên bản chạy trên Hadoop (Apache Mahout).
* **GrapX:** Grapx là nền tảng xử lý đồ thị dựa trên Spark. Nó cung cấp các Api để diễn tảcác tính toán trong đồ thị bằng cách sử dụng Pregel Api.

Trong đề tài cuối kì này, nhóm sẽ sử dụng SparkMLlib để hỗ trợ thực hiện đồ án cuối kì.

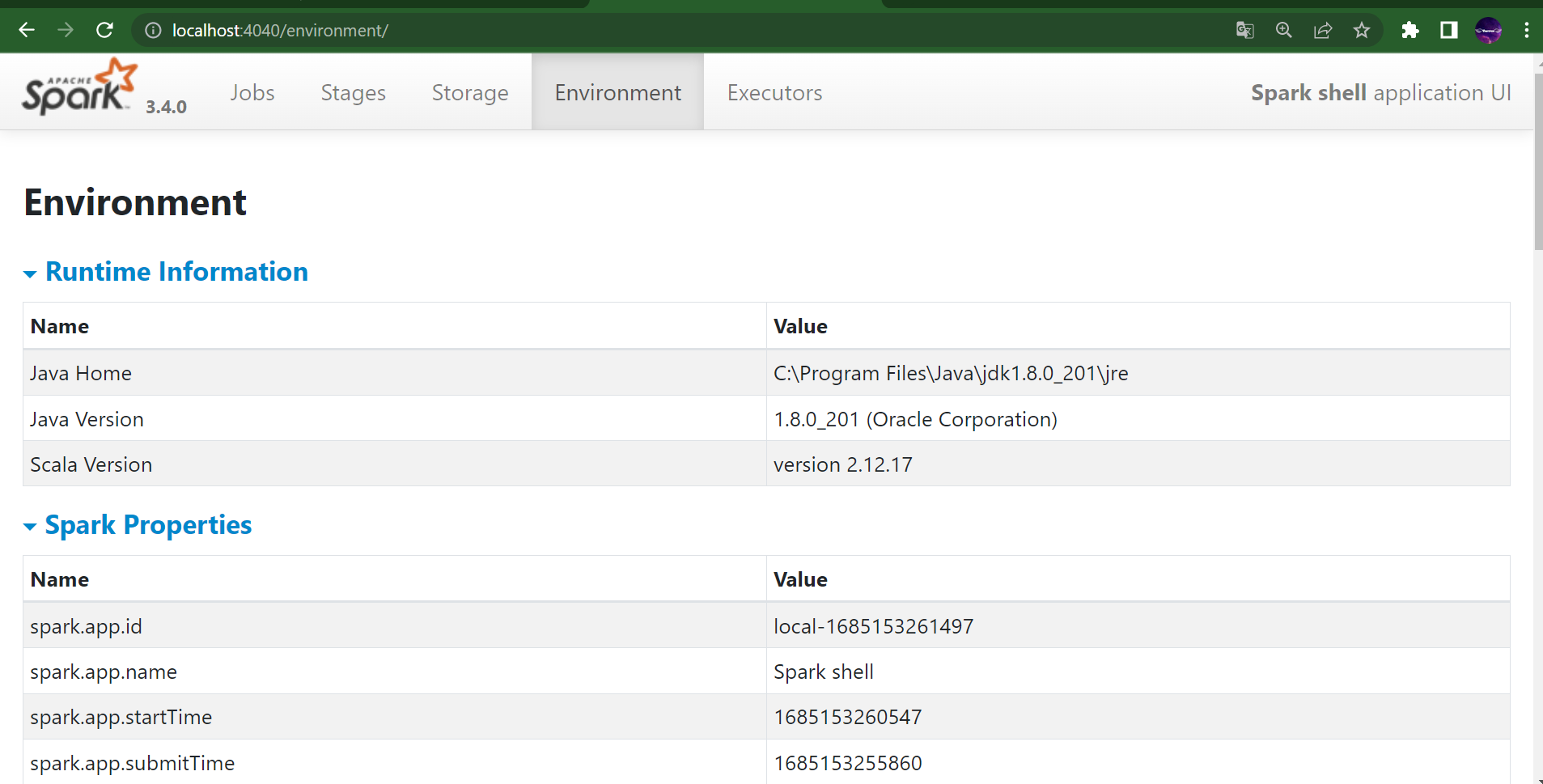
**Chế độ Standalone:**

Muốn chạy ở chế độ này phải bật cụm Spark lên và mỗi khi chạy spark-shell hay spark-submit thì phải set master của bạn tương ứng

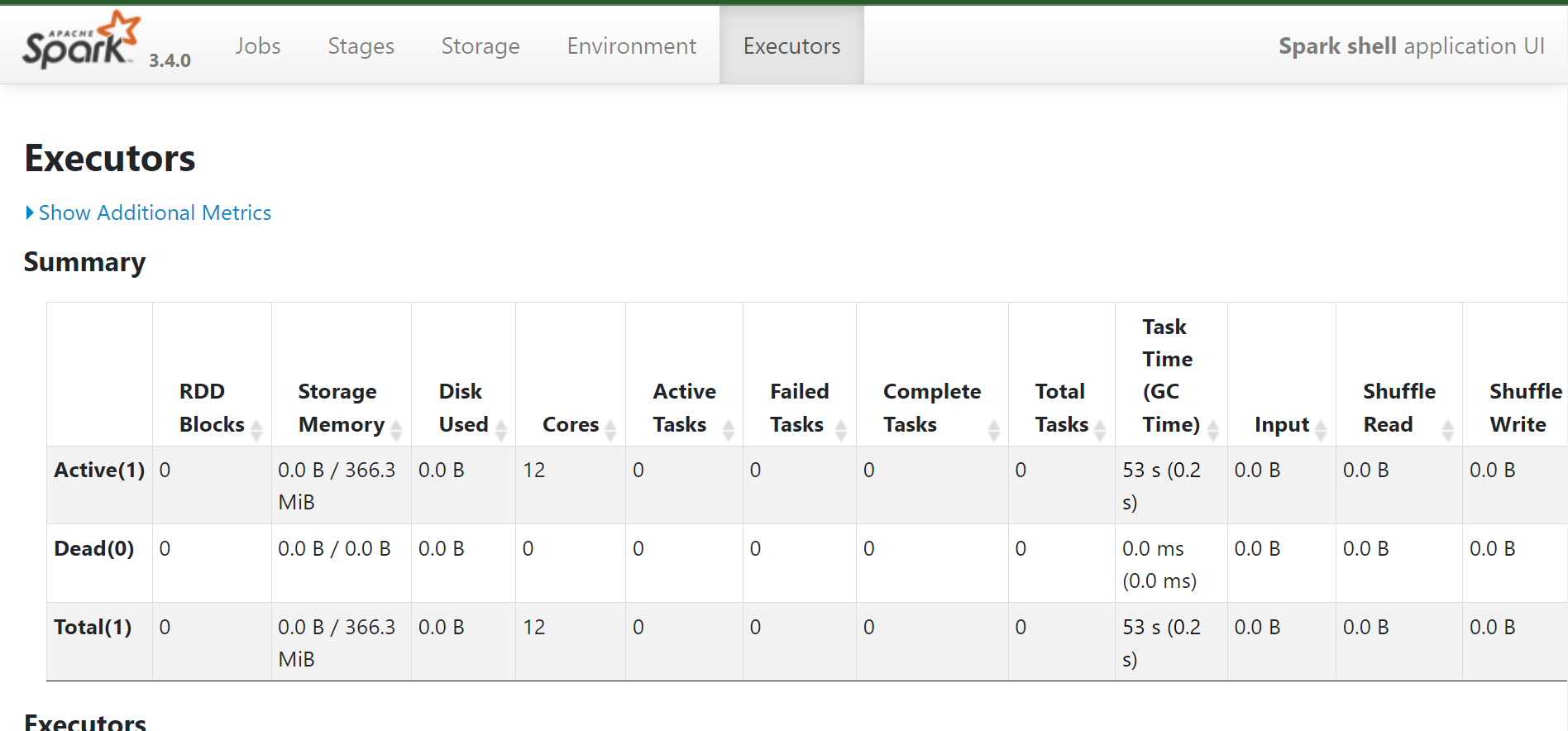
Khi chạy ở chế độ này các bạn có thể thoải mái xem các cấu hình cụm, thông tin job qua các cổng UI của Spark như là cổng 4040.

Truy cập vào localhost:4040 để xem được giao diện UI của Spark:

* Xem enviroment của Apache Spark:



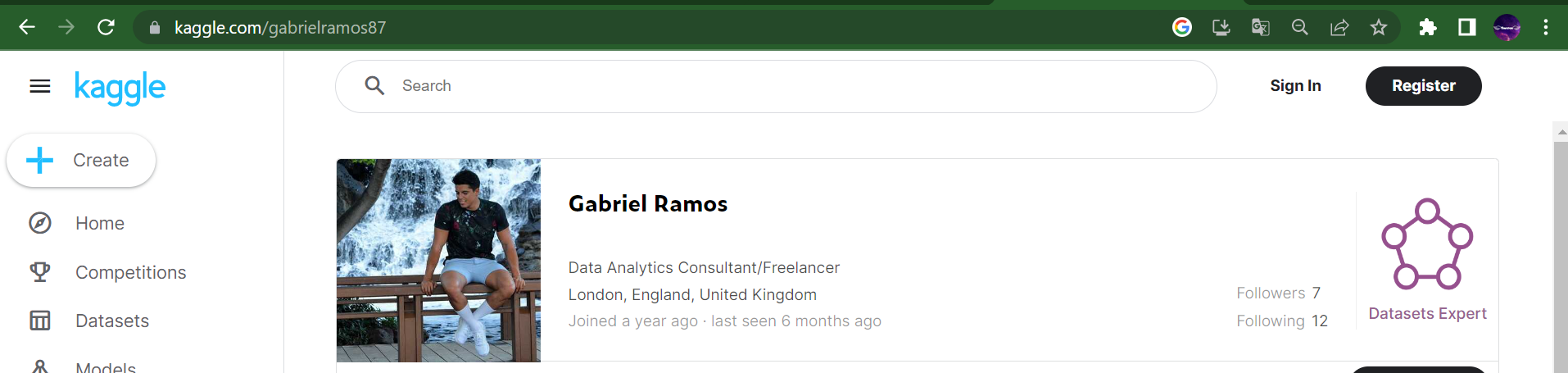
* Xem những task mà Apache Spark đã hoàn thành trong Executors:



**PHẦN 2 – DỮ LIỆU**

## 2.1) Nguồn gốc của tập dữ liệu:

Trong đề tài này, dataset được nhóm 2 thống nhất sử dụng được lấy từ Kaggle và tác giả của tập dữ liệu là Gabriel Ramos.



Theo tác giả, đây là bộ dữ liệu giao dịch bán hàng của thương mại điện tử (bán lẻ trực tuyến) có trụ sở tại Vương quốc Anh trong một năm. Cửa hàng có trụ sở tại Luân Đôn này đã bán quà tặng và đồ gia dụng cho người lớn và trẻ em thông qua trang web từ năm 2007. Khách hàng của họ đến từ khắp nơi trên thế giới và thường mua hàng trực tiếp cho chính họ. Ngoài ra còn có các doanh nghiệp nhỏ mua với số lượng lớn và bán cho các khách hàng khác thông qua các kênh đại lý bán lẻ.

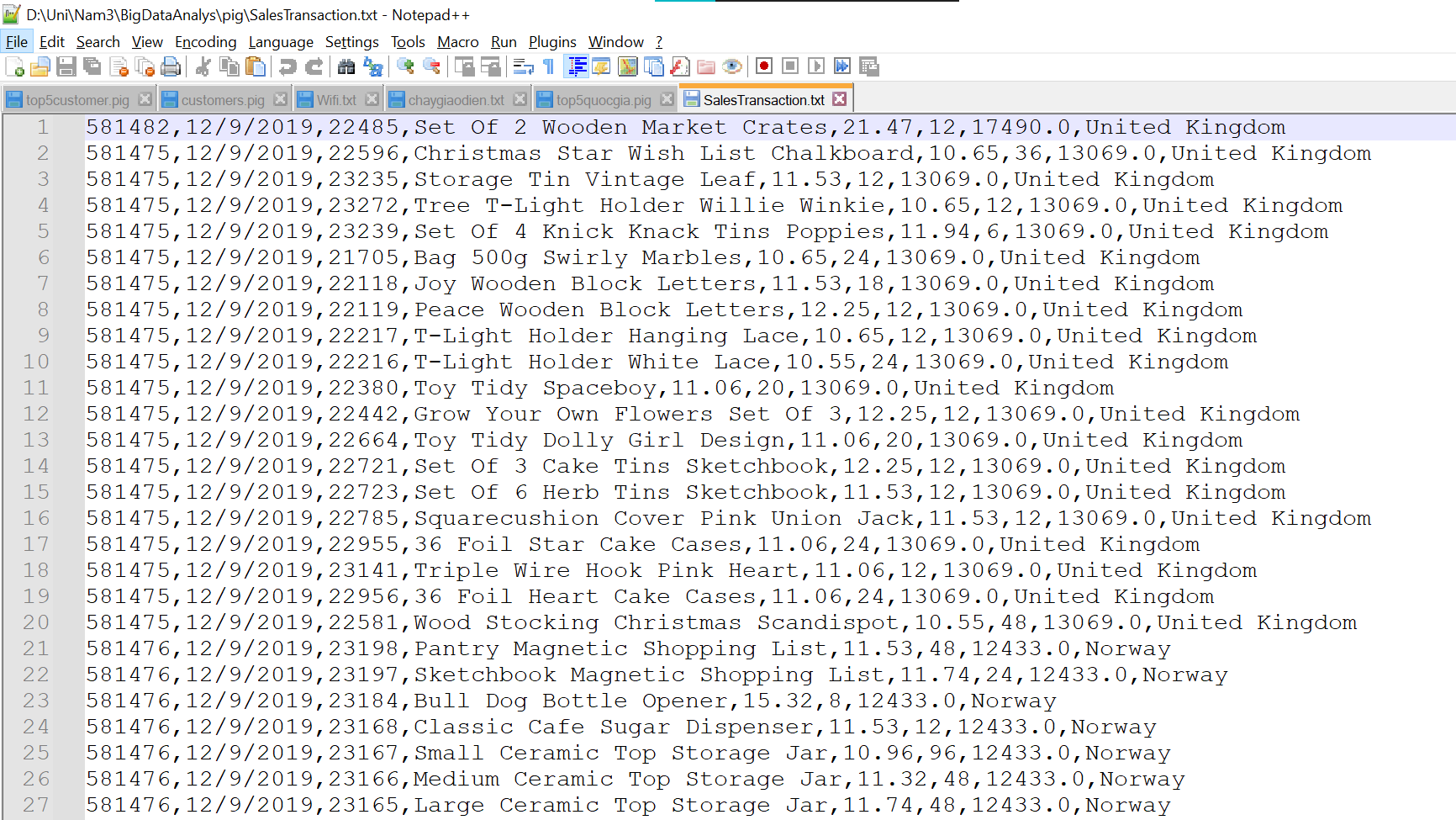
## 2.2) Thông tin về tập dữ liệu:

**Kích thước của tập dữ liệu:** 536350 dòng x 8 cột

**Các biến của tập dữ liệu:**

1. ***TransactionNo (categorical):*** a six-digit unique number that defines each transaction. The letter “C” in the code indicates a cancellation.
2. ***Date (numeric):*** the date when each transaction was generated.
3. ***ProductNo (categorical):*** a five or six-digit unique character used to identify a specific product.
4. ***Product (categorical):*** product/item name.
5. Price (numeric): the price of each product per unit in pound sterling (£).
6. ***Quantity (numeric):*** the quantity of each product per transaction. Negative values related to cancelled transactions.
7. ***CustomerNo (categorical):*** a five-digit unique number that defines each customer.
8. ***Country (categorical):*** name of the country where the customer resides.

Một vài dòng đầu của tập dữ liệu:

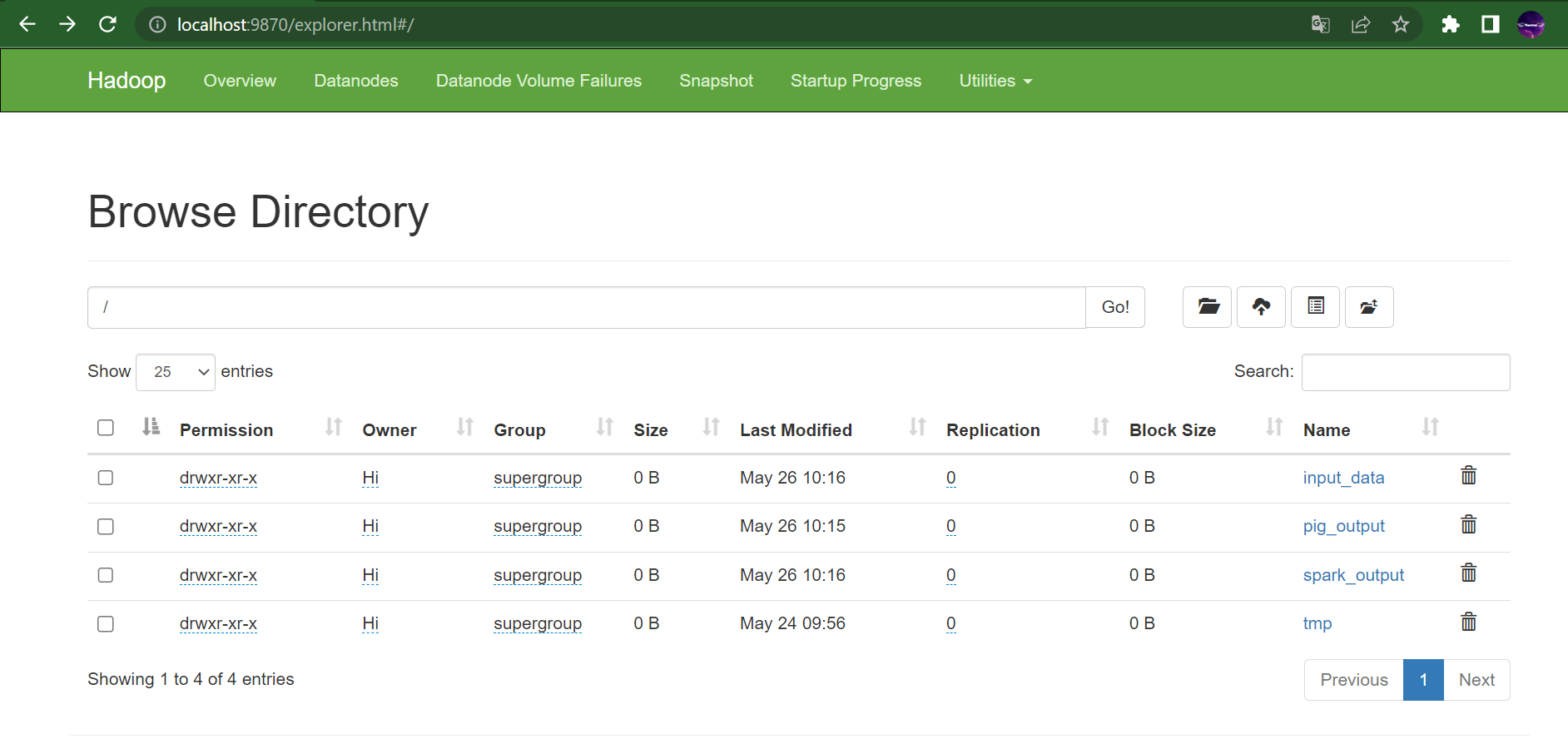


# PHẦN 3 – APACHE PIG:

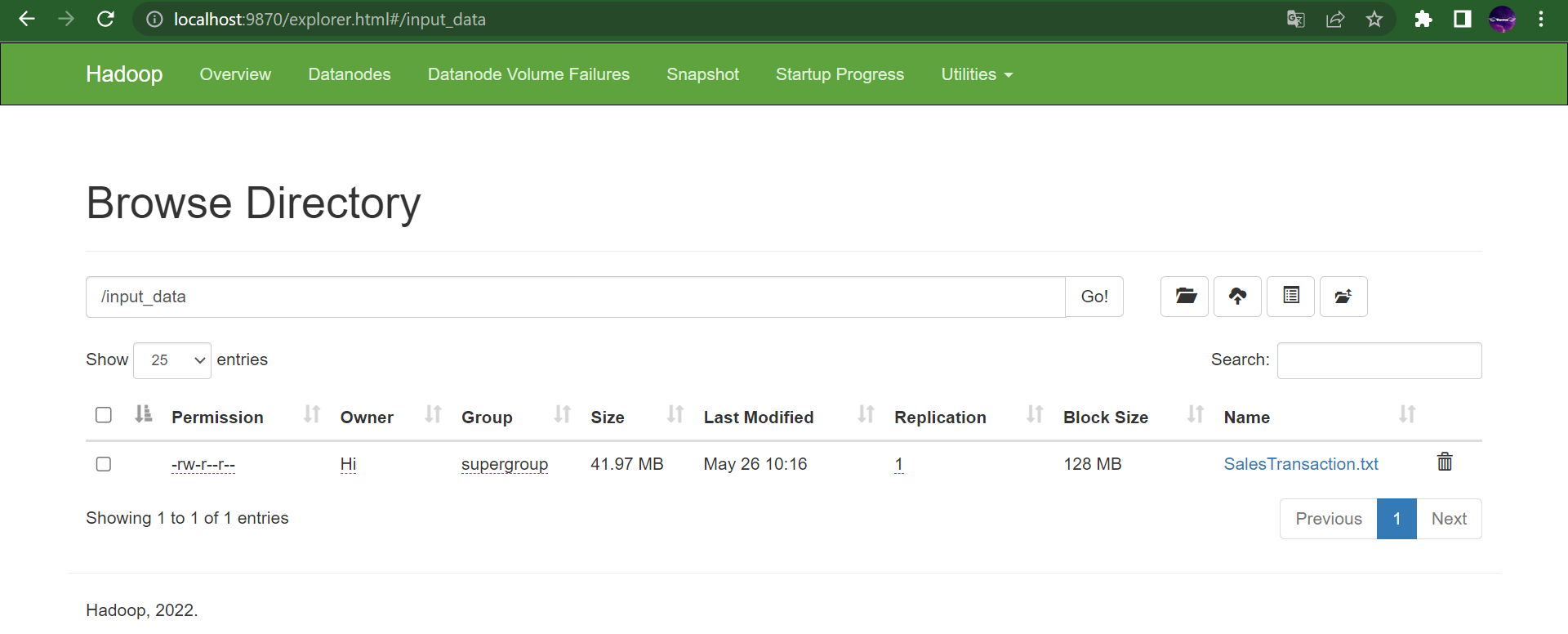
## 3.1) Chuẩn bị:

Tạo 3 thư mục trong HDFS:

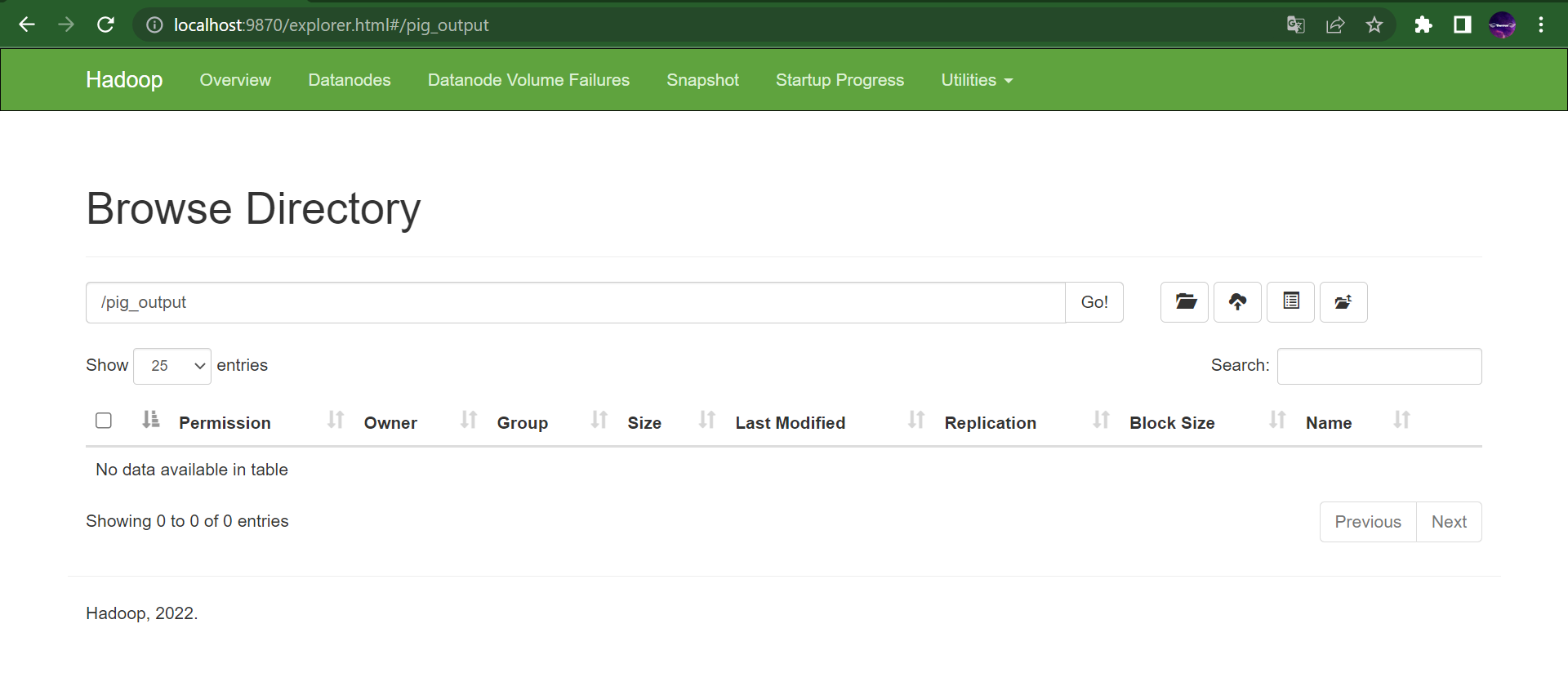
* input\_data: Chứa dataset gốc SalesTransaction.txt.
* pig\_output: Chứa kết quả sau khi thực thi Apache Pig.
* spark\_output: Chứa kết quả sau khi thực thi Apache Pig.



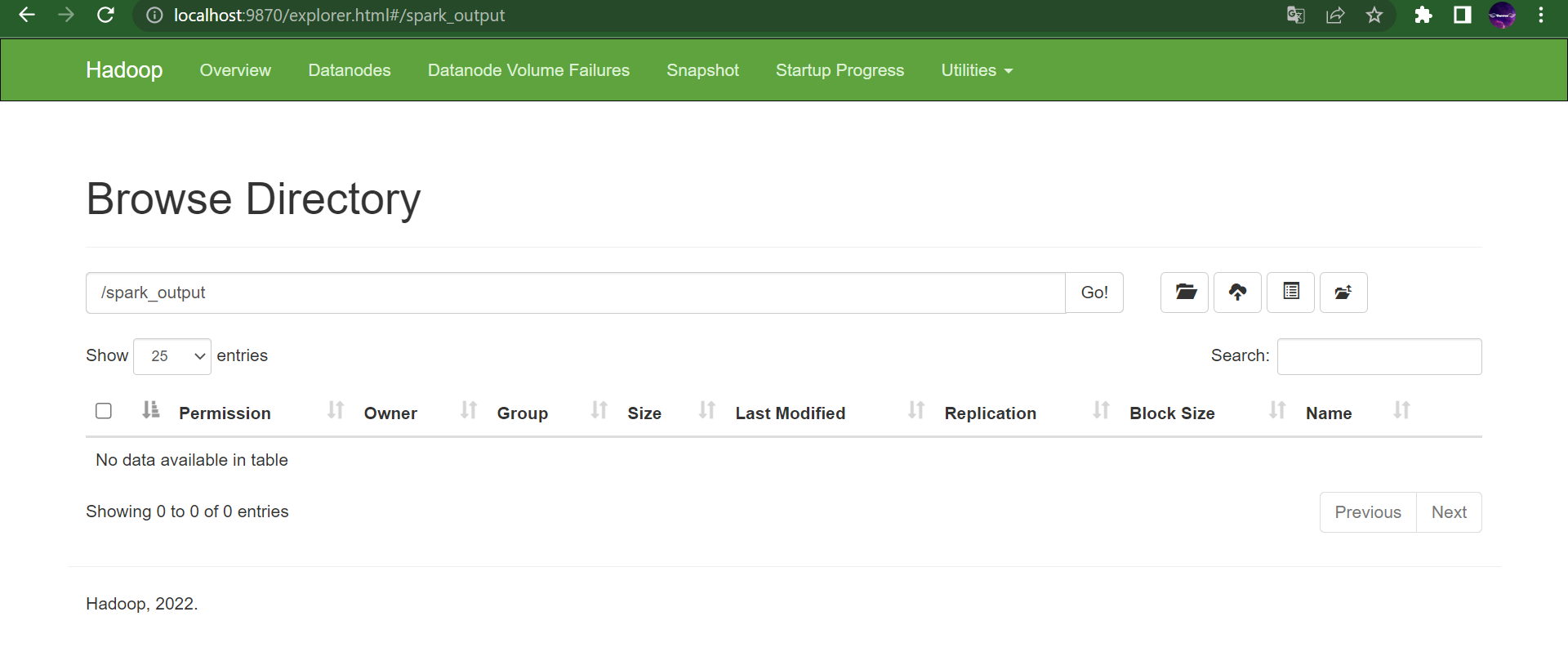
Trong input\_data chứa dataset gốc SalesTransaction.txt



Kết quả của quá trình Pig sẽ được lưu vào folder /pig\_output

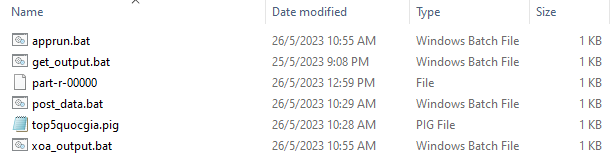


Kết quả của quá trình Spark sẽ được lưu vào folder /spark\_output



## 3.2) Thống kê 5 quốc gia mua hàng nhiều nhất:

File top5quocgia.pig chứa nội dung như bên dưới: lấy dataset nguồn từ HDFS, xử lý chọn ra 5 quốc gia mua hàng nhiều nhất rồi lưu kết quả vào thư mục /pig\_output trong HDFS



File top5quocgia.pig:

inputdata = LOAD 'hdfs://localhost:9000/input\_data/SalesTransaction.txt' USING PigStorage(',')

as (id:chararray,date:chararray,productid:chararray,productname:chararray,price:double,quantity:int,user:chararray,country:chararray);

inputdata = FILTER inputdata BY quantity>0.0;

inputdata = foreach inputdata Generate $0 .. $7, (price \* quantity) AS totalprice;

inputdata\_group = group inputdata by country;

inputdata\_count = foreach inputdata\_group Generate FLATTEN(inputdata.country) AS country, SUM(inputdata.price) AS total;

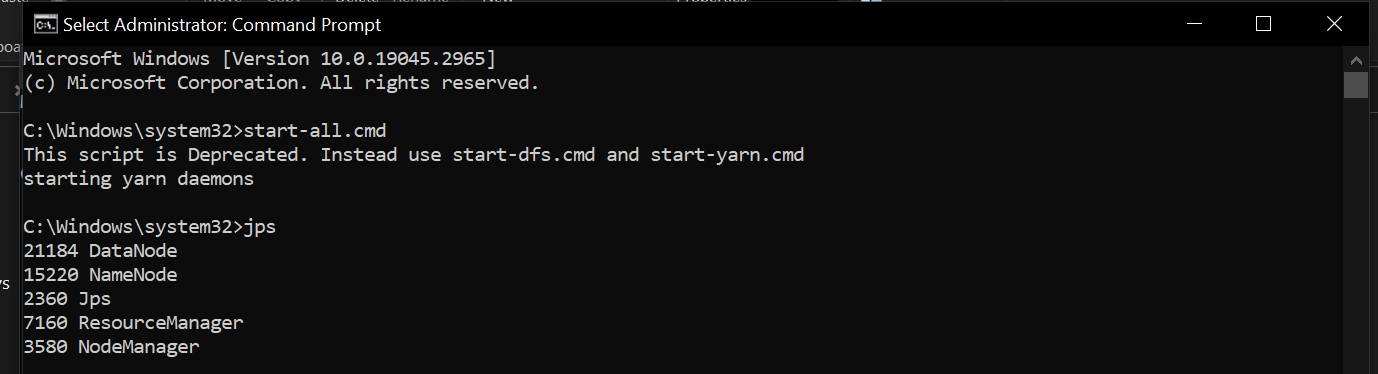
inputdata\_dist = distinct inputdata\_count;

inputdata\_sapxep = ORDER inputdata\_dist BY total DESC;

inputdata\_gioihan = LIMIT inputdata\_sapxep 5;

STORE inputdata\_gioihan INTO 'hdfs://localhost:9000/pig\_output/top5quocgia';

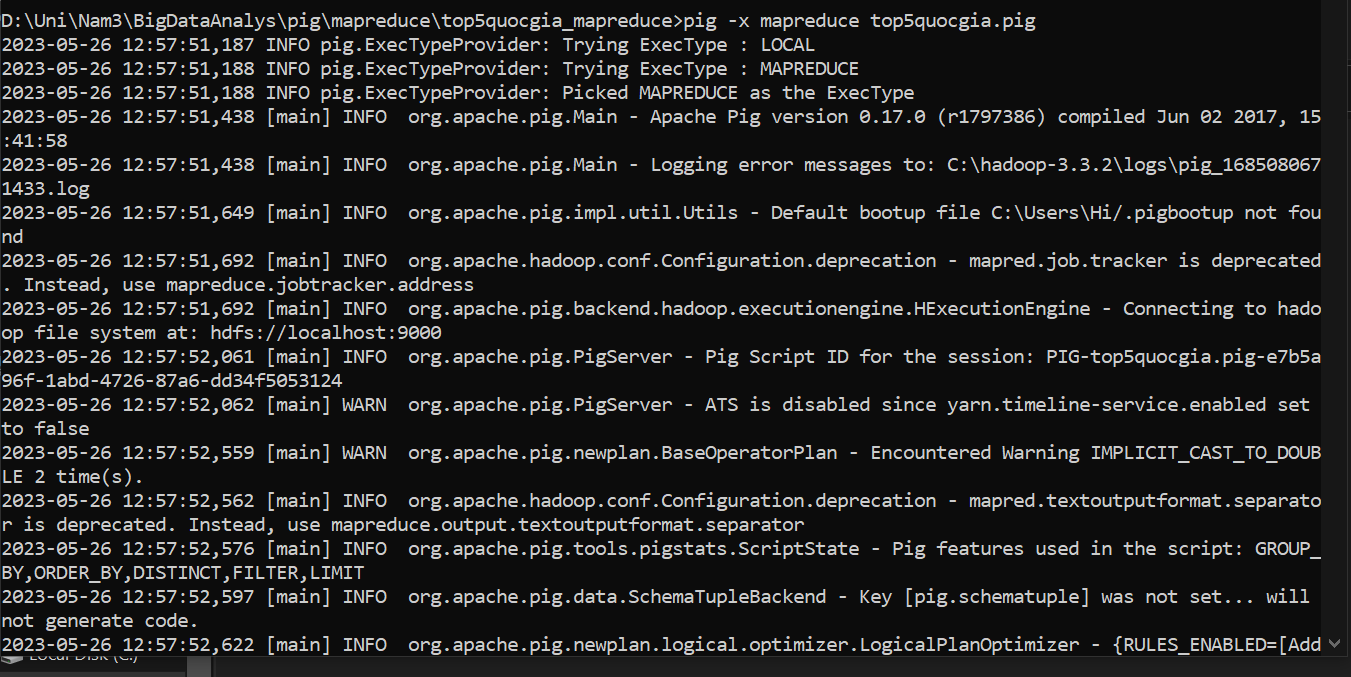
Bước 1: start-all.cmd: lệnh để chạy hadoop cluster



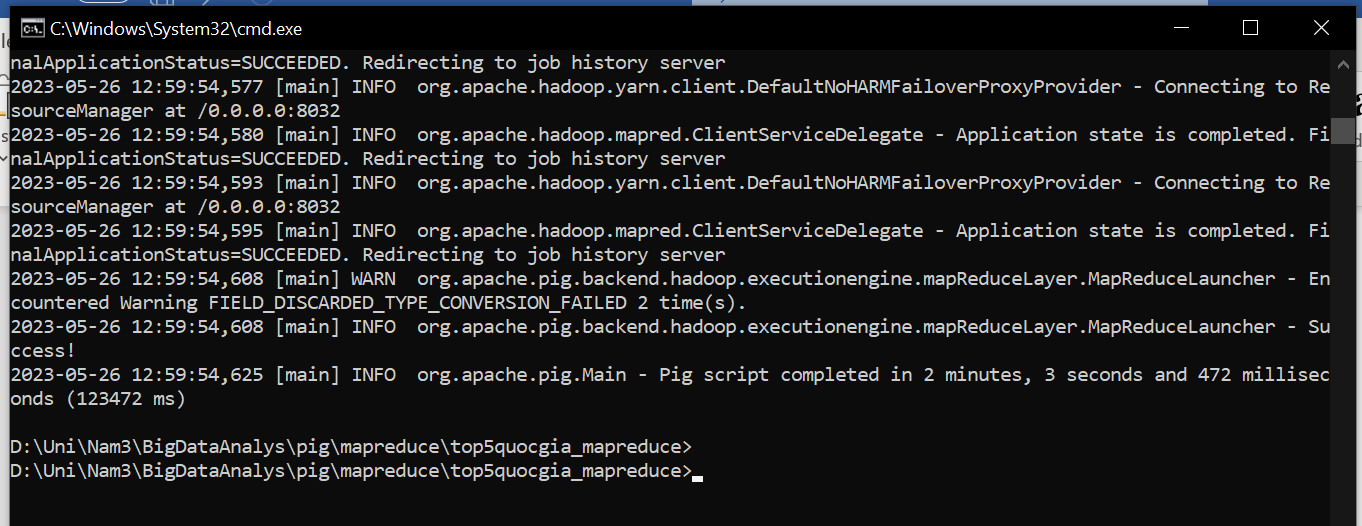
Bước 2: mapred.cmd historyserver: chạy thêm tác vụ JobHistoryServer giúp cải thiện tốc độ chạy Pig chế độ MapReduce

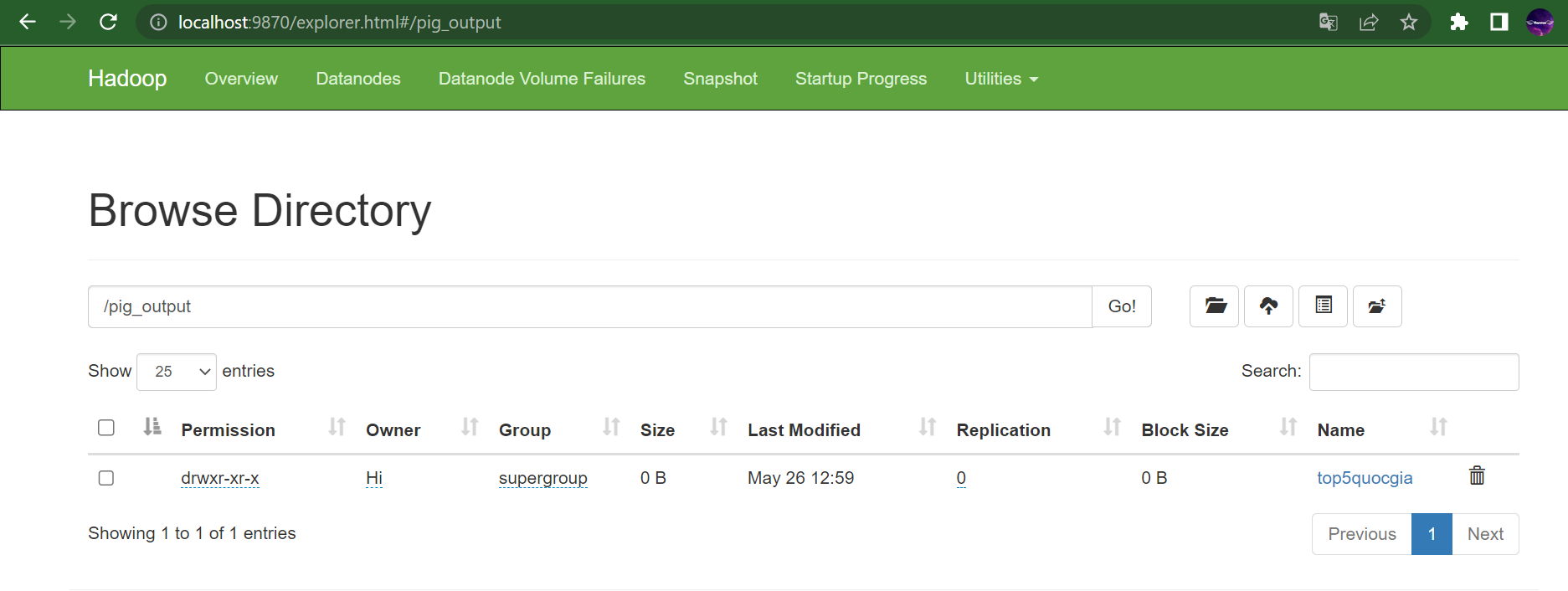


Bước 3: pig -x mapreduce top5quocgia.pig: chạy scripts với pig ở chế độ mapreduce

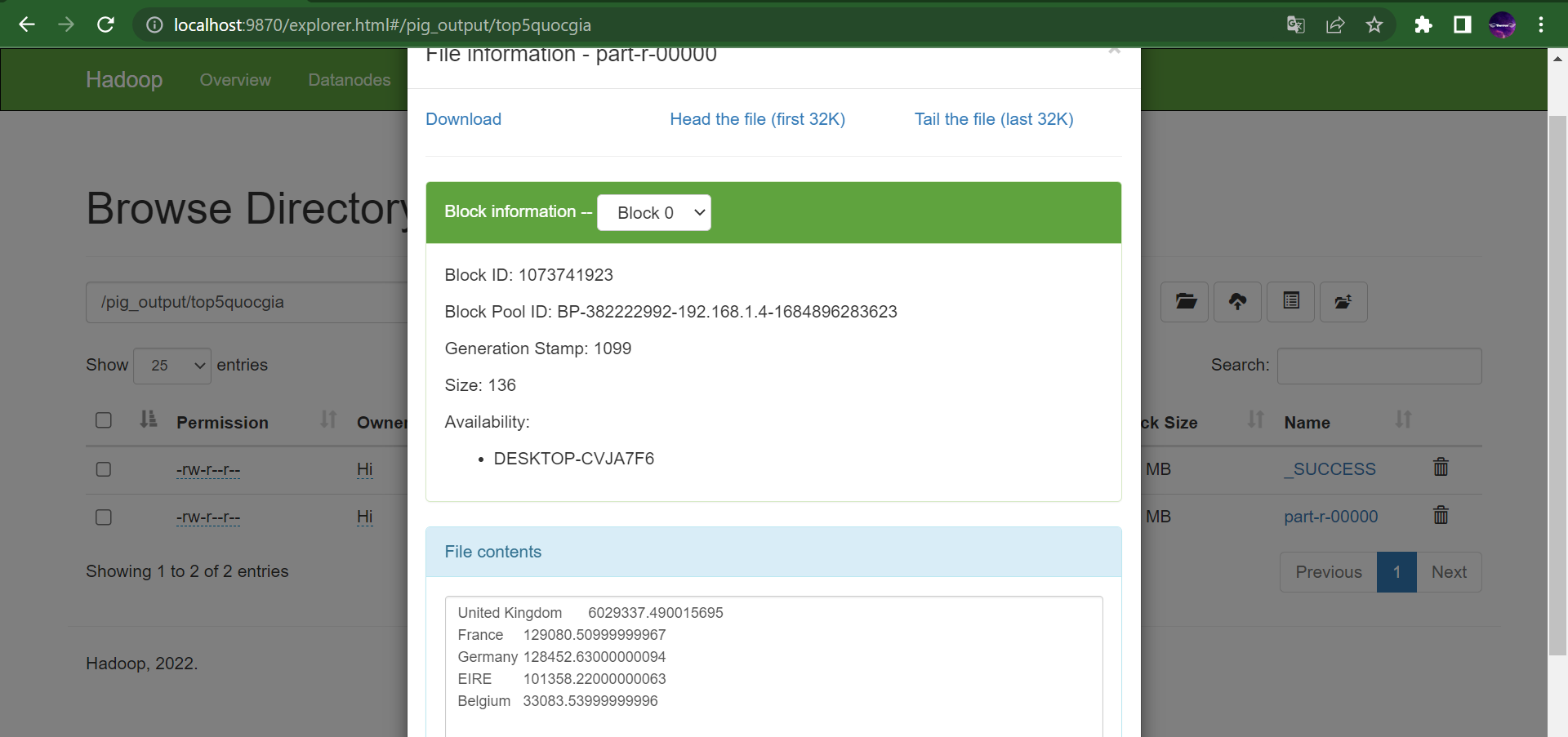


Hoàn thành: sau khi hoàn thành chương trình, file output được lưu vào HDFS ở thư mục /pig\_output



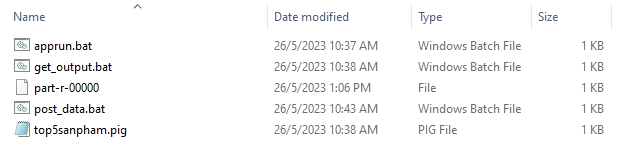


Part-r-000000: xuất ra những dòng đầu tiên của kết quả



## 3.3) Thống kê 5 transaction có số tiền nhiều nhất:

File top5transaction.pig chứa nội dung như bên dưới: lấy dataset nguồn từ HDFS, xử lý chọn ra 5 giao dịch được mua nhiều nhất rồi lưu kết quả vào thư mục /pig\_output trong HDFS



File top5transaction.pig:

inputdata = LOAD 'hdfs://localhost:9000/input\_data/SalesTransaction.txt' USING PigStorage(',')

as (id:chararray,date:chararray,productid:chararray,productname:chararray,price:double,quantity:int,user:chararray,country:chararray);

inputdata = FILTER inputdata BY quantity>0.0;

inputdata = foreach inputdata Generate $0 .. $7, (price \* quantity) AS totalprice;

inputdata\_group = GROUP inputdata BY id;

inputdata\_sum = foreach inputdata\_group Generate FLATTEN(inputdata.id), SUM(inputdata.quantity),SUM(inputdata.totalprice) as totalprice;

inputdata\_sum = distinct inputdata\_sum;

inputdata\_sapxep = ORDER inputdata\_sum BY totalprice DESC;

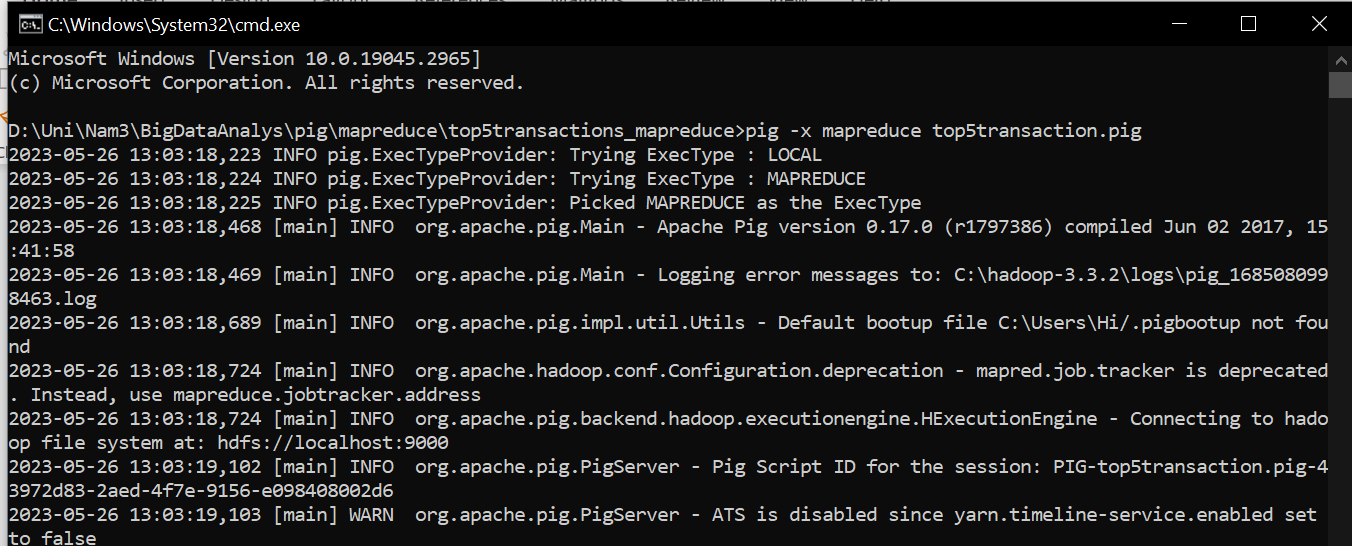
inputdata\_gioihan = LIMIT inputdata\_sapxep 5;

STORE inputdata\_gioihan INTO 'hdfs://localhost:9000/pig\_output/top5transaction';

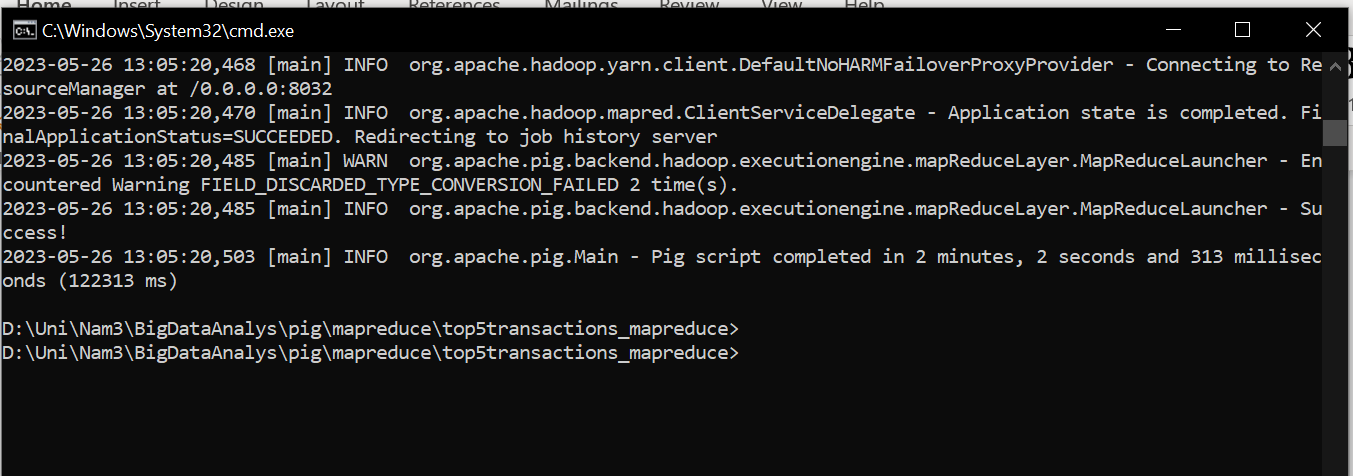
Bước 1: start-all.cmd (Đã thực hiện)

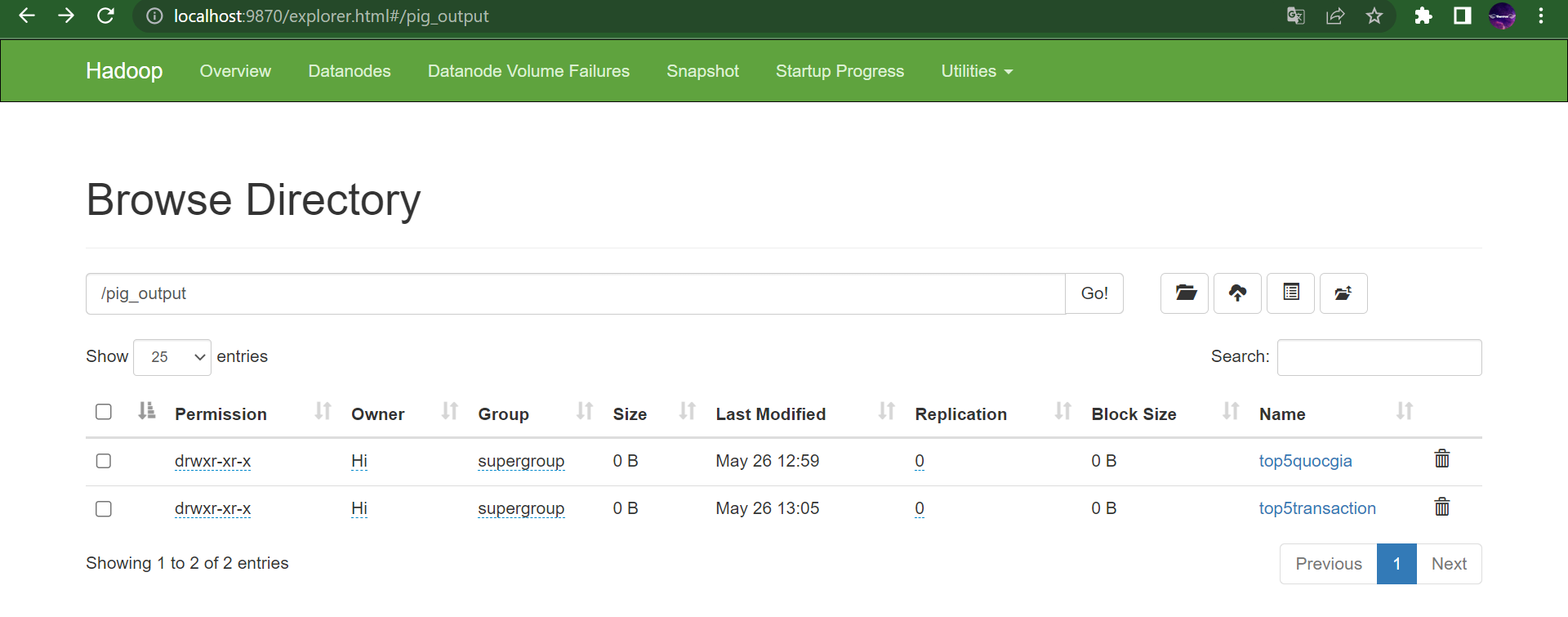
Bước 2: mapred.cmd historyserver (Đã thực hiện): JobHistoryServer giúp cải thiện tốc độ chạy Pig chế độ MapReduce

Bước 3: pig -x mapreduce top5transaction.pig:

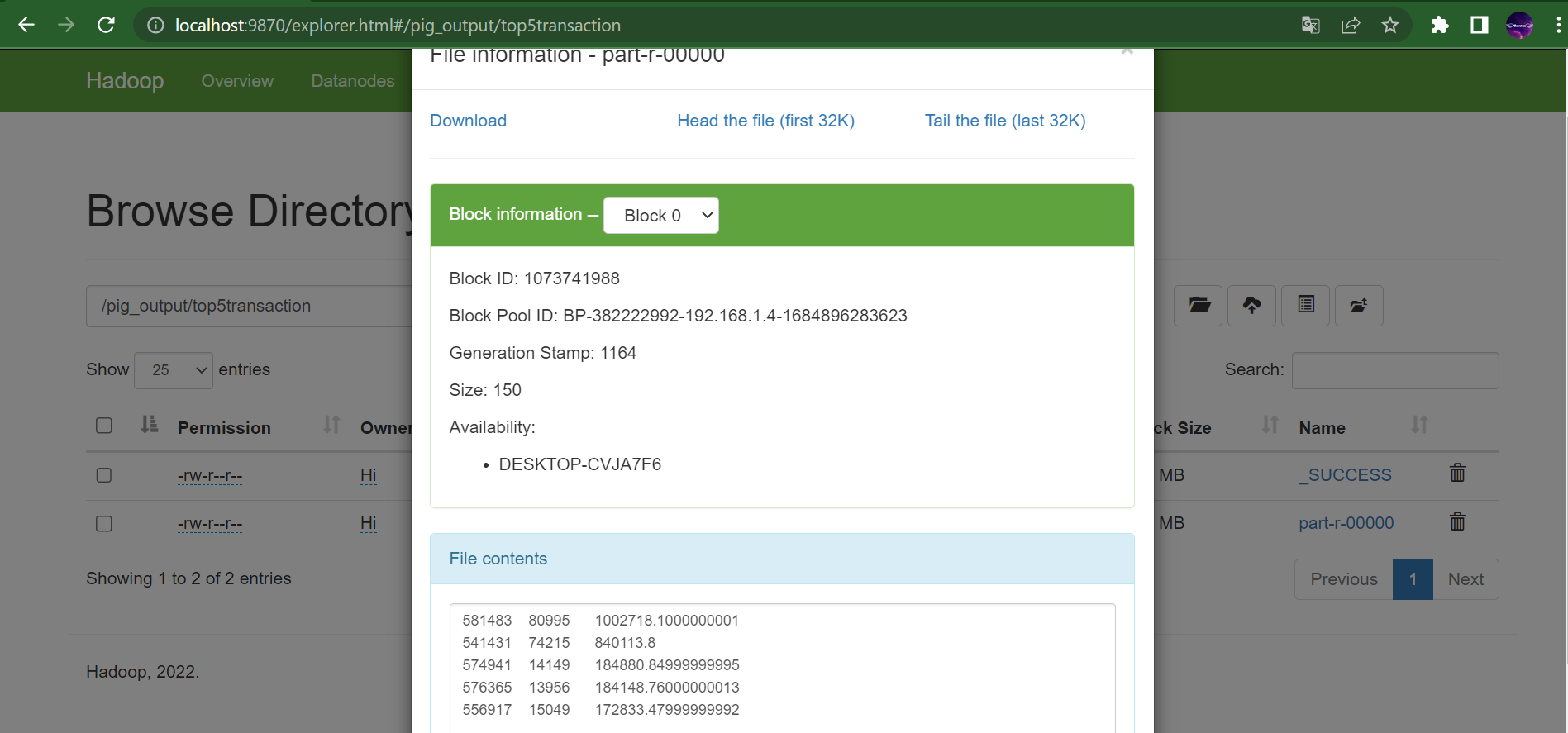


Hoàn thành: sau khi hoàn thành chương trình, file output được lưu vào HDFS ở thư mục /pig\_output



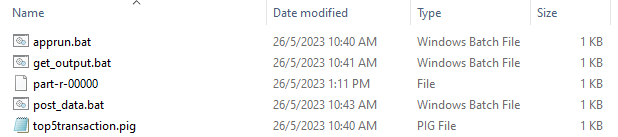


Part-r-000000: xuất ra những dòng đầu tiên của kết quả



## 3.4) Thống kê 5 sản phẩm có số tiền nhiều nhất:

File top5sanpham.pig chứa nội dung như bên dưới: lấy dataset nguồn từ HDFS, xử lý chọn ra 5 sản phẩm được mua nhiều nhất rồi lưu kết quả vào thư mục /pig\_output trong HDFS



File top5sanpham.pig:

inputdata = LOAD 'hdfs://localhost:9000/input\_data/SalesTransaction.txt' USING PigStorage(',')

as (id:chararray,date:chararray,productid:chararray,productname:chararray,price:double,quantity:int,user:chararray,country:chararray);

inputdata = FILTER inputdata BY quantity>0.0;

inputdata = foreach inputdata Generate $0 .. $7, (price \* quantity) AS totalprice;

inputdata\_group = GROUP inputdata BY productname;

inputdata\_sum = foreach inputdata\_group Generate FLATTEN(inputdata.productname), SUM(inputdata.quantity) as totalquantity, SUM(inputdata.totalprice) as totalprice;

inputdata\_sum = distinct inputdata\_sum;

inputdata\_sapxep = ORDER inputdata\_sum BY totalprice DESC;

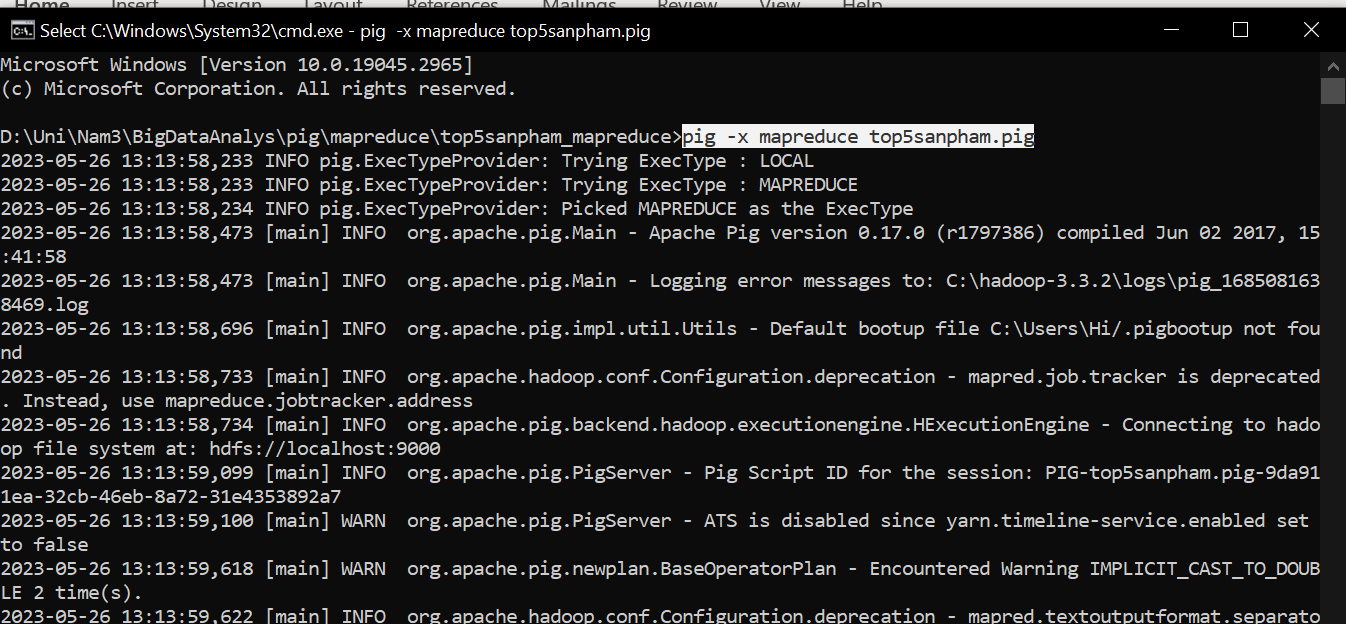
inputdata\_gioihan = LIMIT inputdata\_sapxep 5;

STORE inputdata\_gioihan INTO 'hdfs://localhost:9000/pig\_output/top5sanpham';

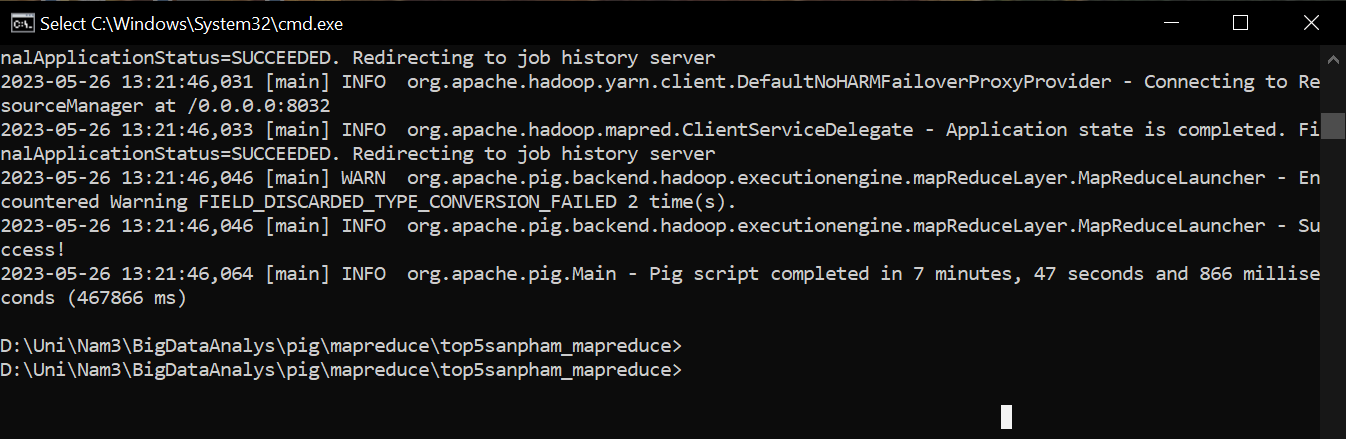
Bước 1: start-all.cmd (Đã thực hiện)

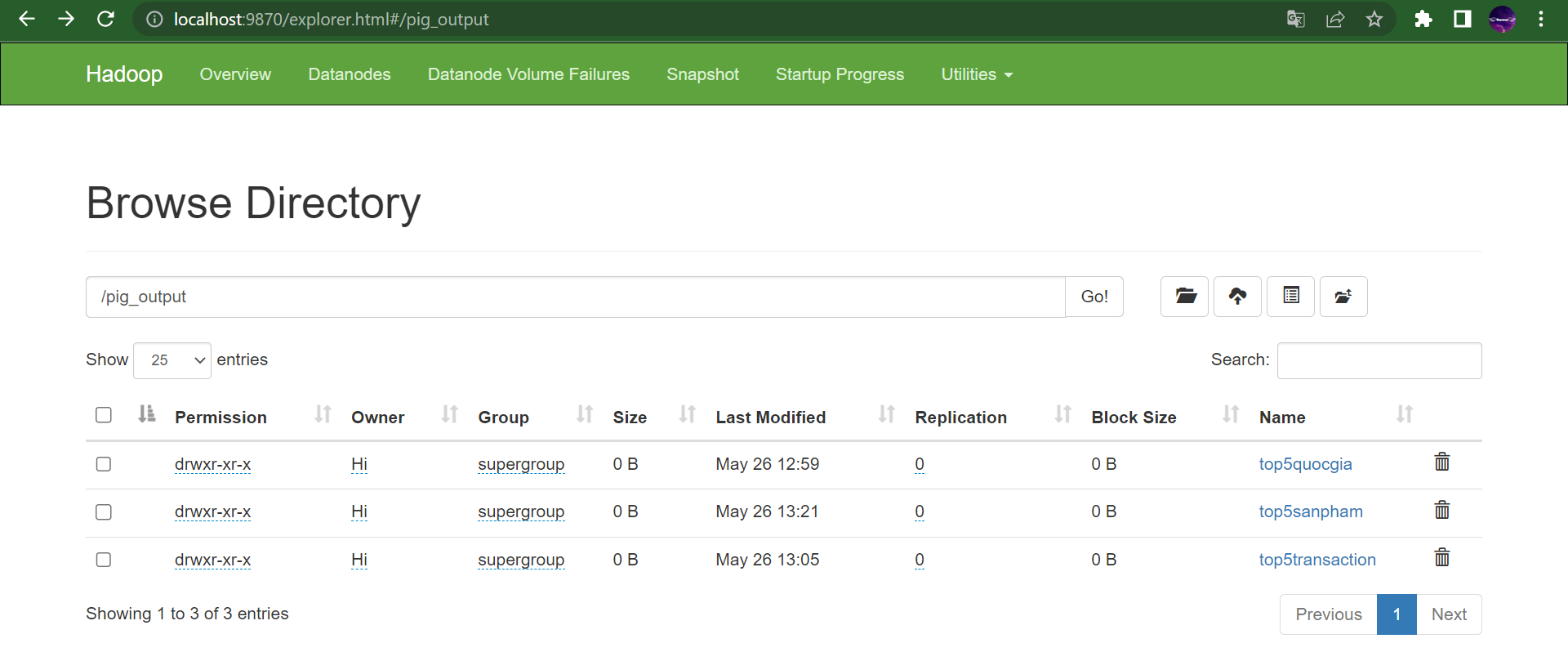
Bước 2: mapred.cmd historyserver (Đã thực hiện): JobHistoryServer giúp cải thiện tốc độ chạy Pig chế độ MapReduce

Bước 3: pig -x mapreduce top5sanpham.pig:

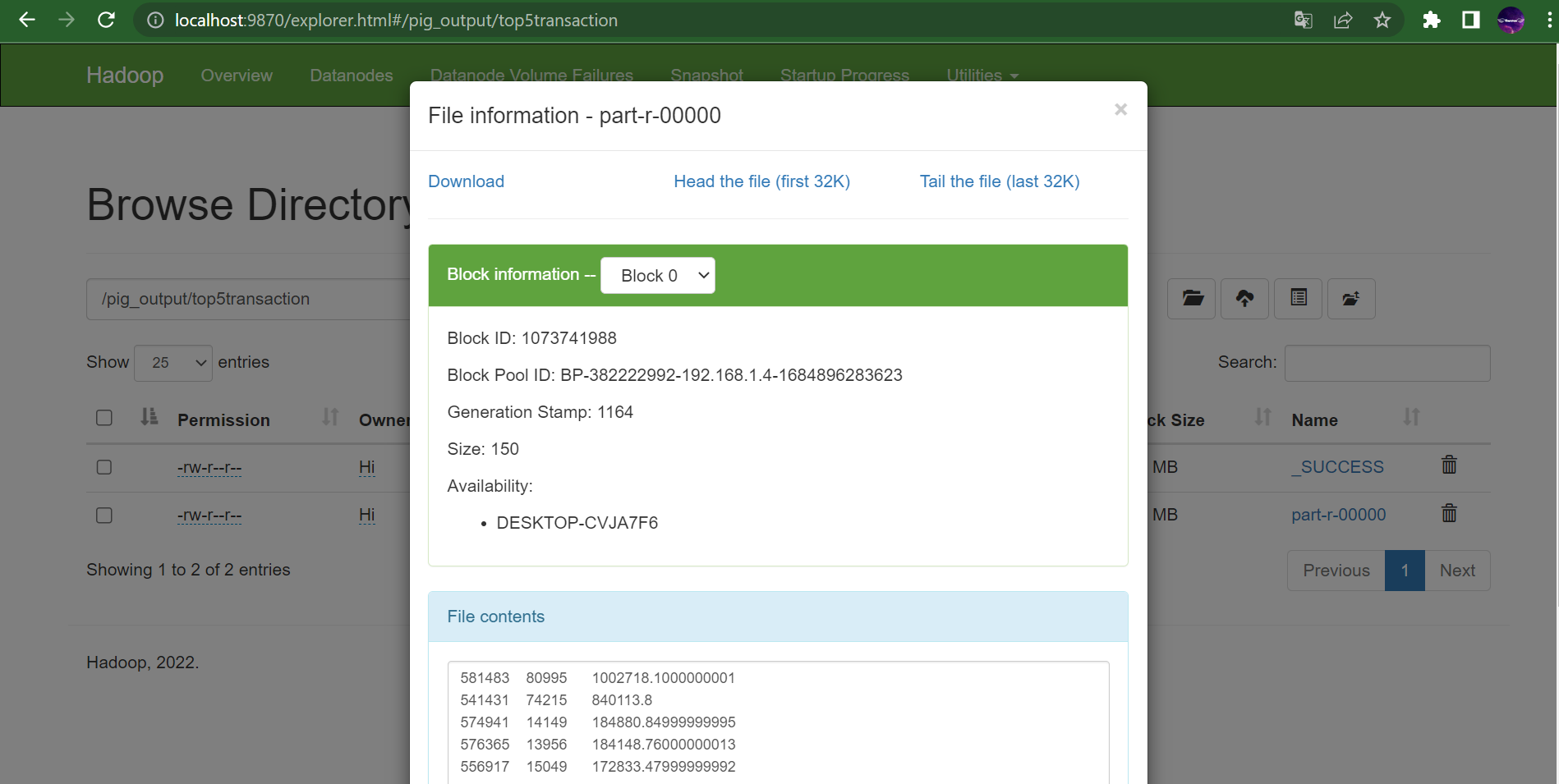


Hoàn thành: sau khi hoàn thành chương trình, file output được lưu vào HDFS ở thư mục /pig\_output



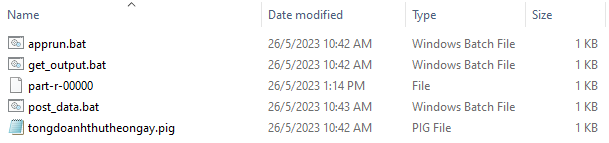


Part-r-000000: xuất ra những dòng đầu tiên của kết quả



## 3.5) Thống kê doanh thu theo ngày:

File tongdoanhthutheongay.pig chứa nội dung như bên dưới: lấy dataset nguồn từ HDFS, xử lý chọn ra 10 ngày được mua nhiều nhất rồi lưu kết quả vào thư mục /pig\_output trong HDFS



File tongdoanhthutheongay.pig:

inputdata = LOAD 'hdfs://localhost:9000/input\_data/SalesTransaction.txt' USING PigStorage(',')

as (transactionid:chararray,date:chararray,productid:chararray,productname:chararray,price:double,quantity:int,customer:chararray,country:chararray);

inputdata = FILTER inputdata BY quantity>0.0;

inputdata = foreach inputdata Generate $0 .. $7, (price \* quantity) AS totalprice;

inputdata\_group = GROUP inputdata BY date;

inputdata\_sum = foreach inputdata\_group Generate Flatten(inputdata.date), SUM(inputdata.totalprice)as totalprice;

inputdata\_sum = distinct inputdata\_sum;

inputdata\_sapxep = ORDER inputdata\_sum BY totalprice DESC;

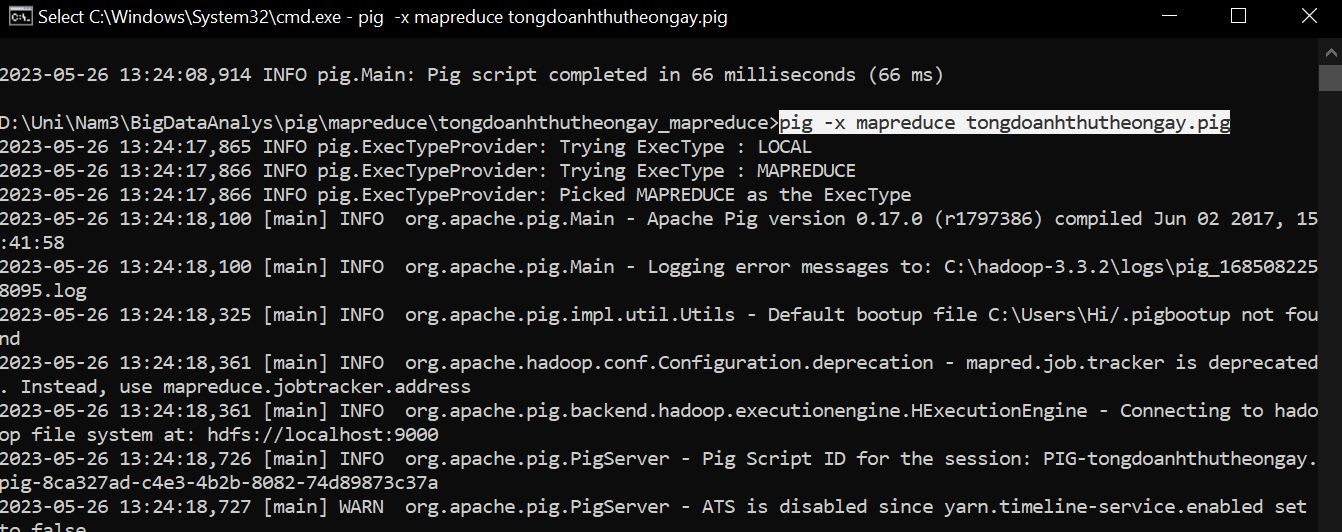
inputdata\_gioihan = LIMIT inputdata\_sapxep 10;

STORE inputdata\_gioihan INTO 'hdfs://localhost:9000/pig\_output/tongdoanhthutheongay';

Bước 1: start-all.cmd (Đã thực hiện)

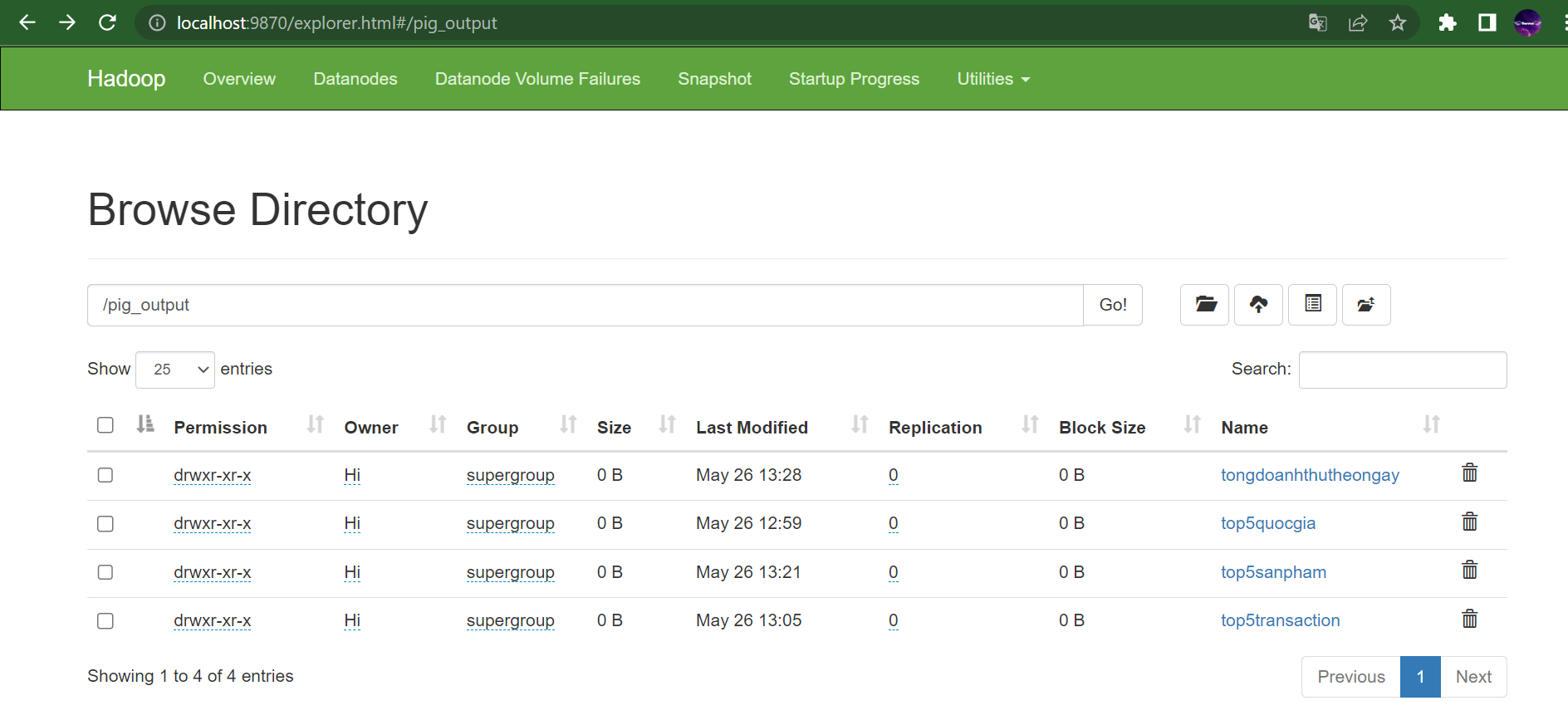
Bước 2: mapred.cmd historyserver (Đã thực hiện): JobHistoryServer giúp cải thiện tốc độ chạy Pig chế độ MapReduce

Bước 3: pig -x mapreduce top5sanpham.pig:

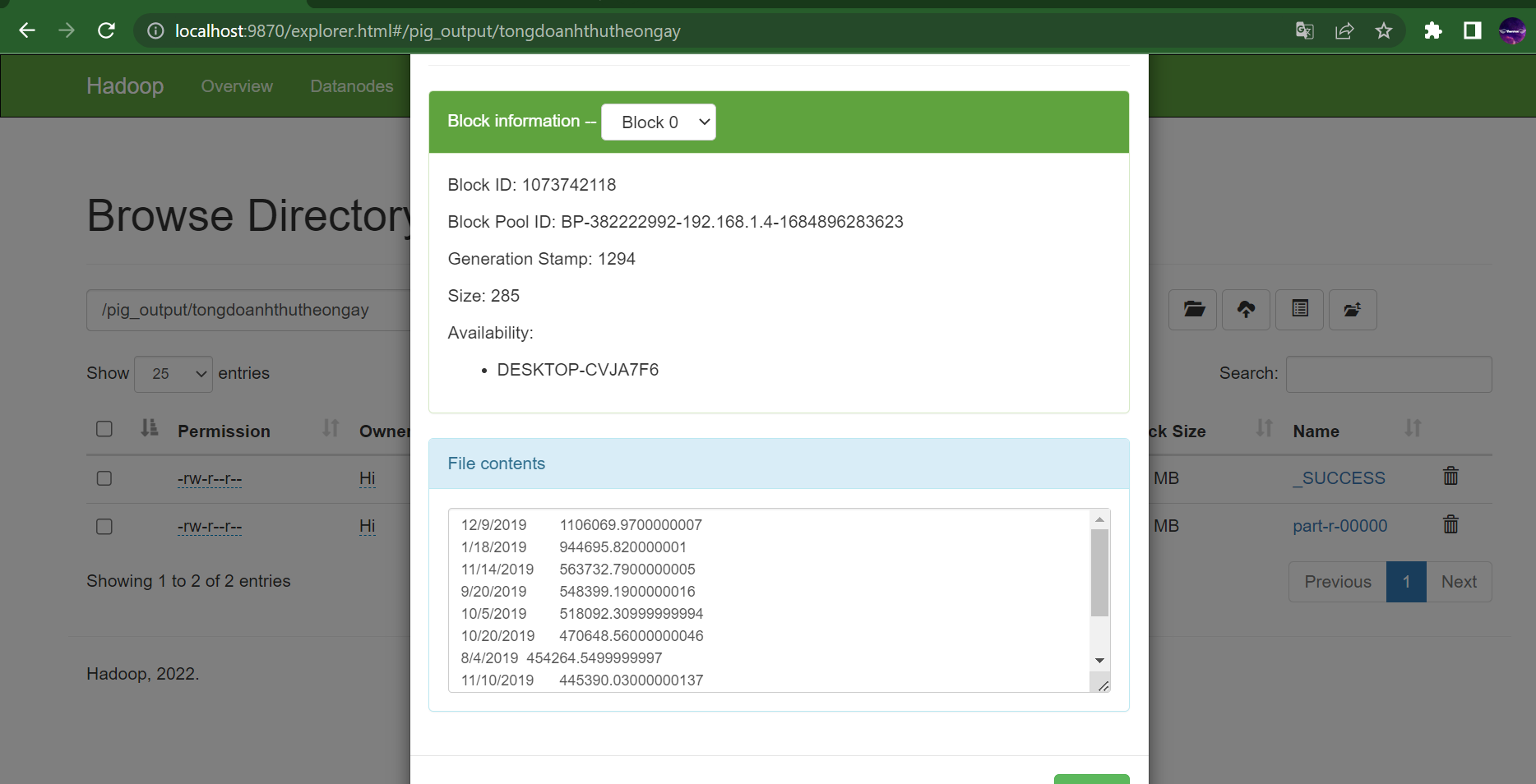


Hoàn thành: sau khi hoàn thành chương trình, file output được lưu vào HDFS ở thư mục /pig\_output



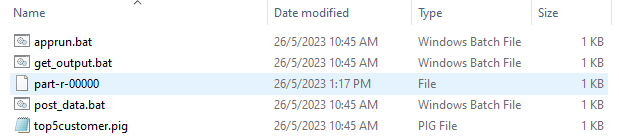


Part-r-000000: xuất ra những dòng đầu tiên của kết quả



## 3.6) Thống kê 5 khách hàng chi tiền nhiều nhất:

File top5khachhang.pig chứa nội dung như bên dưới: lấy dataset nguồn từ HDFS, xử lý chọn ra 5 khách hàng mua nhiều nhất rồi lưu kết quả vào thư mục /pig\_output trong HDFS



File top5khachhang.pig:

inputdata = LOAD 'hdfs://localhost:9000/input\_data/SalesTransaction.txt' USING PigStorage(',')

as (id:chararray,date:chararray,productid:chararray,productname:chararray,price:double,quantity:int,customer:chararray,country:chararray);

inputdata = FILTER inputdata BY quantity>0.0;

inputdata = foreach inputdata Generate $0 .. $7, (price \* quantity) AS totalprice;

inputdata\_group = GROUP inputdata BY customer;

inputdata\_sum = foreach inputdata\_group Generate FLATTEN(inputdata.customer), SUM(inputdata.quantity) as totalquantity, SUM(inputdata.totalprice) as totalprice;

inputdata\_sum = distinct inputdata\_sum;

inputdata\_sapxep = ORDER inputdata\_sum BY totalprice DESC;

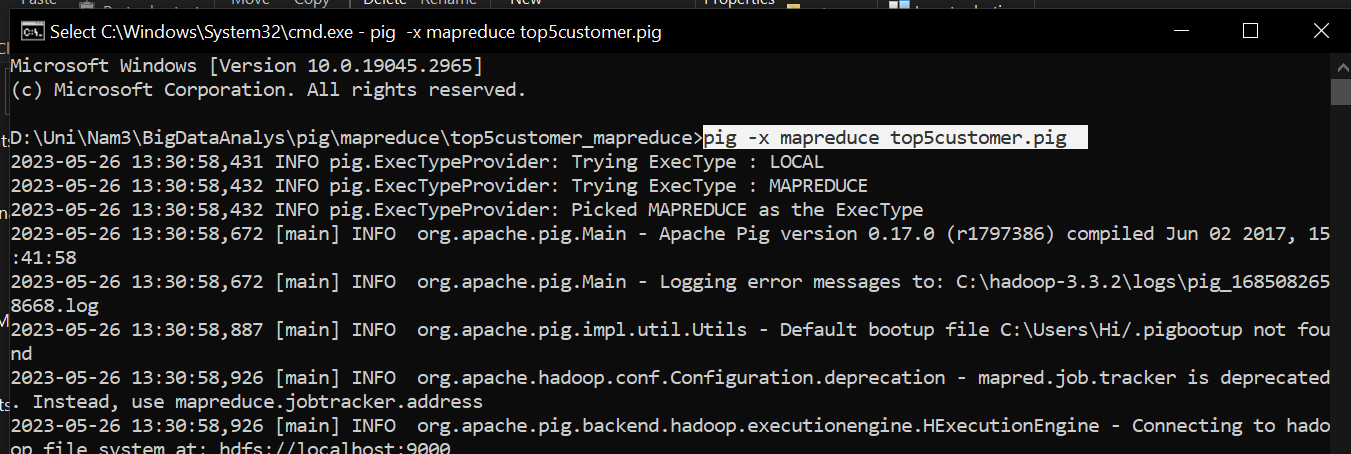
inputdata\_gioihan = LIMIT inputdata\_sapxep 5;

STORE inputdata\_gioihan INTO 'hdfs://localhost:9000/pig\_output/top5customer';

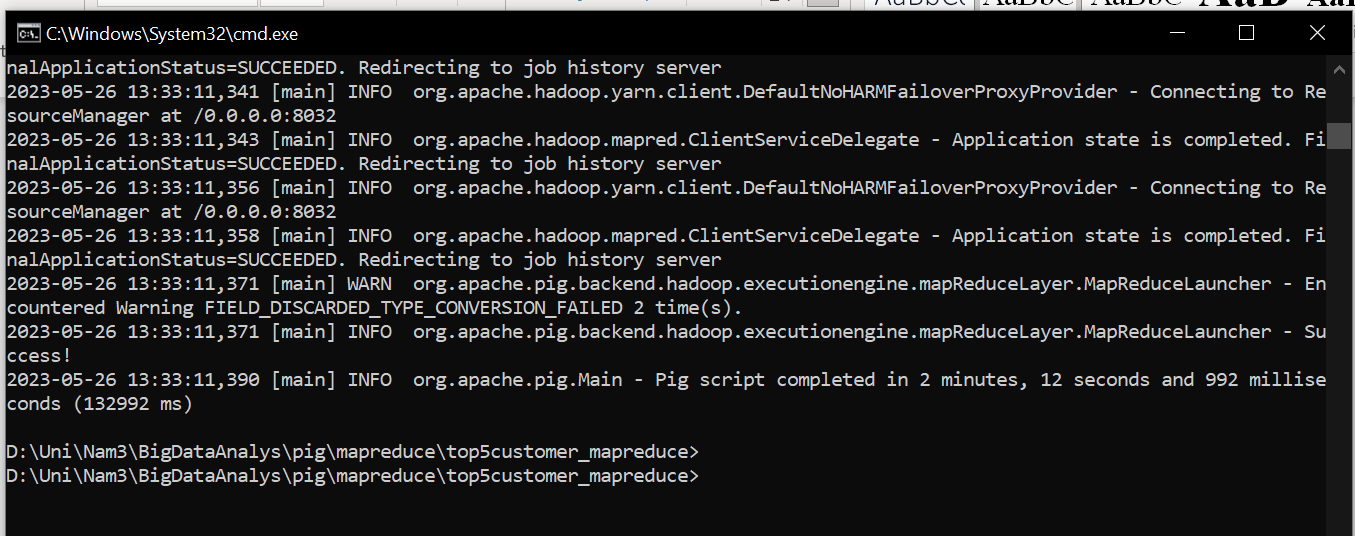
Bước 1: start-all.cmd (Đã thực hiện)

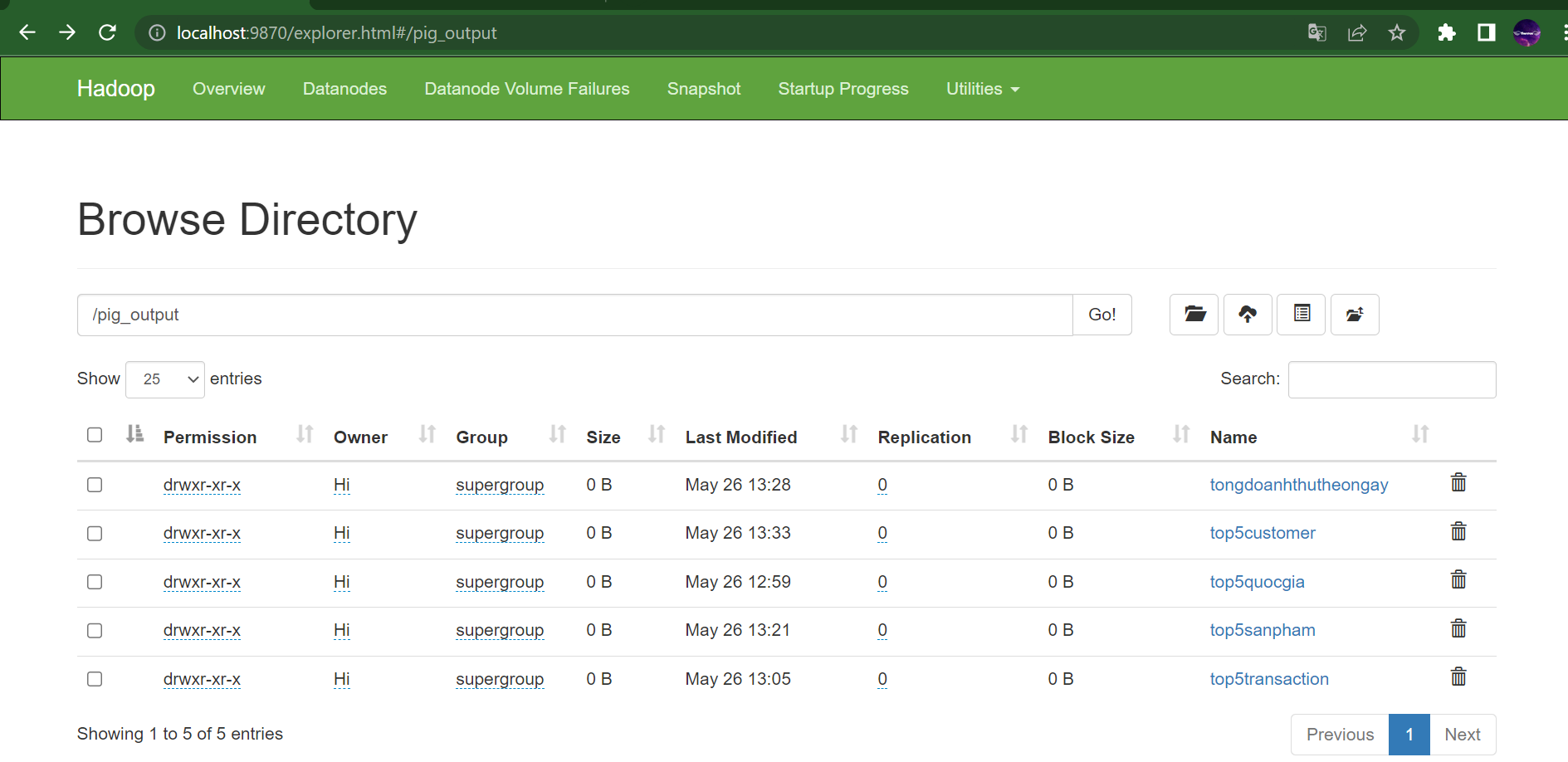
Bước 2: mapred.cmd historyserver (Đã thực hiện): JobHistoryServer giúp cải thiện tốc độ chạy Pig chế độ MapReduce

Bước 3: pig -x mapreduce top5customer.pig:

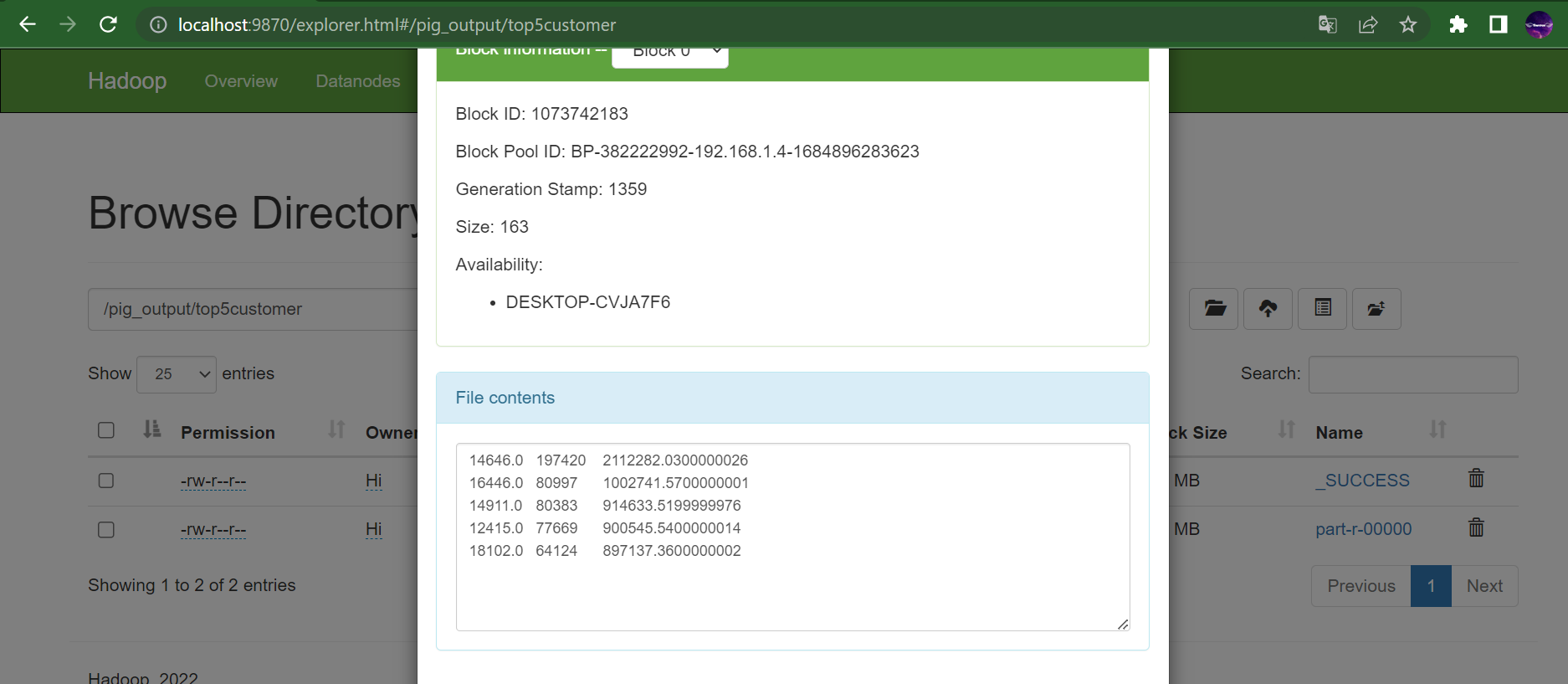


Hoàn thành: sau khi hoàn thành chương trình, file output được lưu vào HDFS ở thư mục /pig\_output



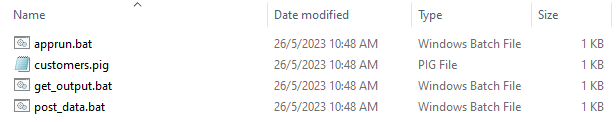


Part-r-00000: xuất ra những dòng đầu tiên của kết quả



## 3.7) Tạo bảng khách hàng với thông tin số tiền khách hàng đã chi và số lượng hàng khách hàng đã mua từ trước tới giờ:

File customers.pig chứa nội dung như bên dưới: lấy dataset nguồn từ HDFS, gom nhóm dataset theo từng khách hàng với số lượng mua bao nhiêu, tổng mua bao nhiêu rồi lưu kết quả vào thư mục /pig\_output trong HDFS



File customers.pig:

inputdata = LOAD 'hdfs://localhost:9000/input\_data/SalesTransaction.txt' USING PigStorage(',')

as (id:chararray,date:chararray,productid:chararray,productname:chararray,price:double,quantity:int,user:chararray,country:chararray);

inputdata = FILTER inputdata BY quantity>0.0;

inputdata = foreach inputdata Generate $0 .. $7, (price \* quantity) AS totalprice;

inputdata\_group = GROUP inputdata BY user;

inputdata\_sum = foreach inputdata\_group Generate FLATTEN(inputdata.user), SUM(inputdata.quantity),SUM(inputdata.totalprice) as totalprice;

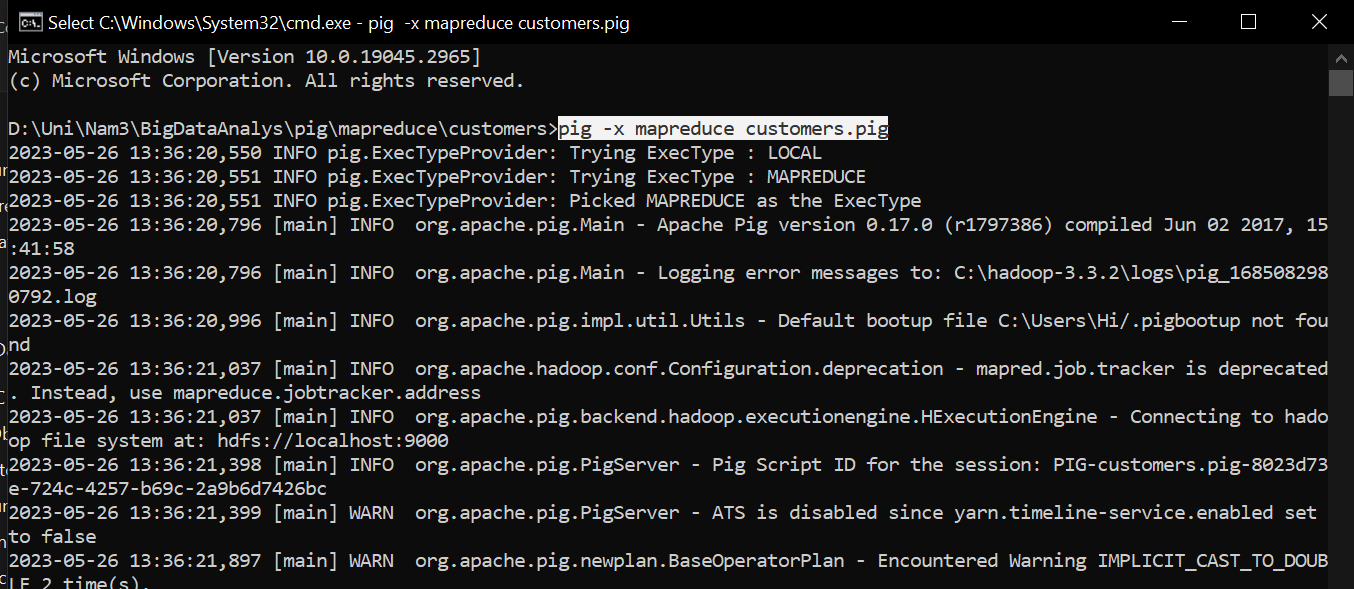
inputdata\_sum = distinct inputdata\_sum;

STORE inputdata\_sum INTO 'hdfs://localhost:9000/pig\_output/customers';

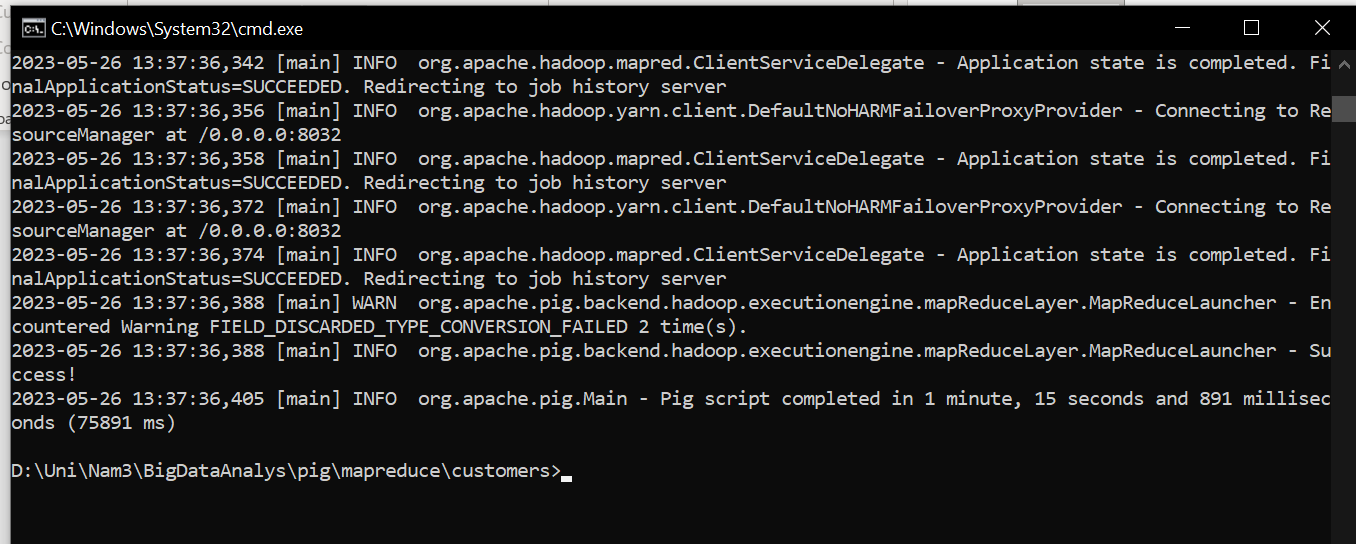
Bước 1: start-all.cmd (Đã thực hiện)

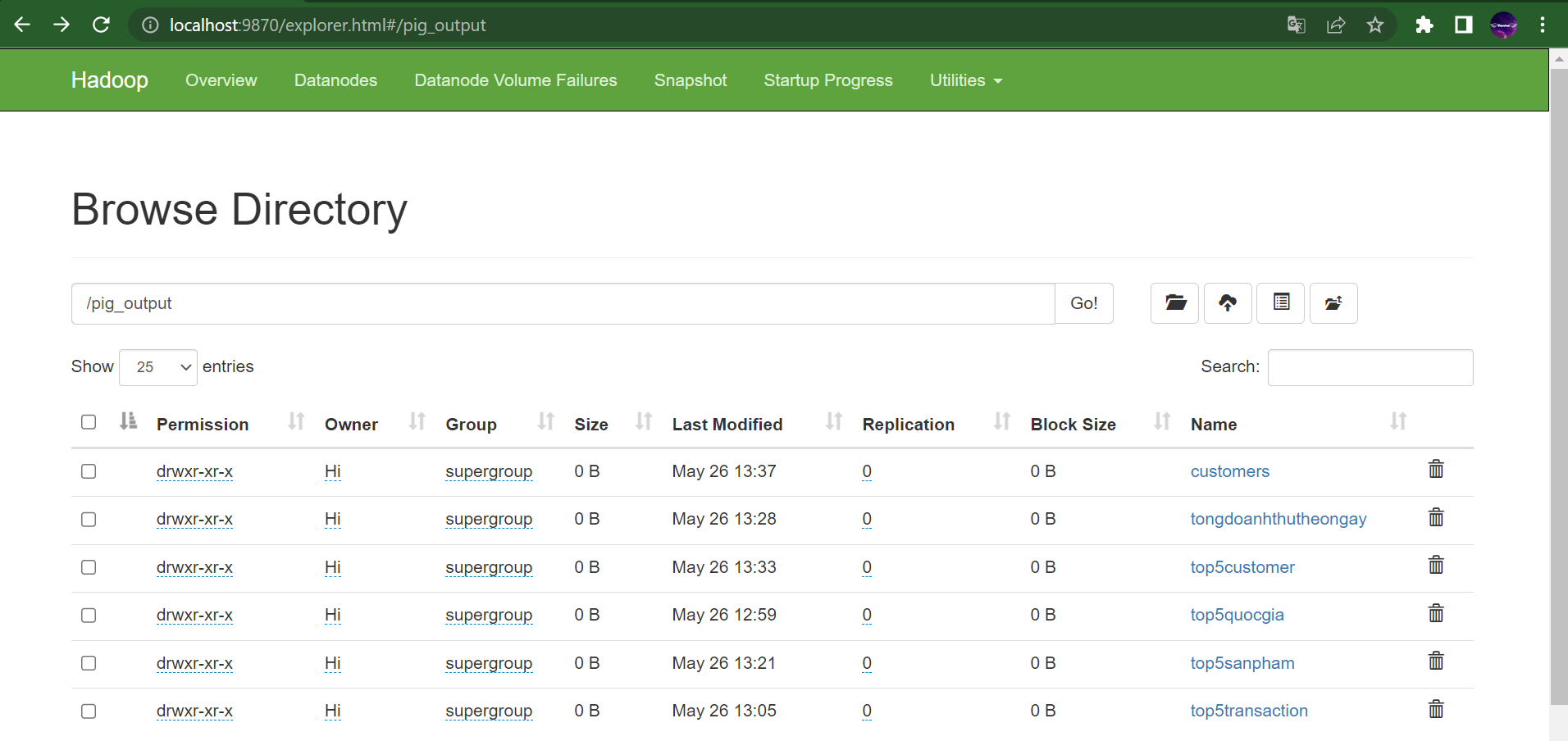
Bước 2: mapred.cmd historyserver (Đã thực hiện): JobHistoryServer giúp cải thiện tốc độ chạy Pig chế độ MapReduce

Bước 3: pig -x mapreduce customers.pig:

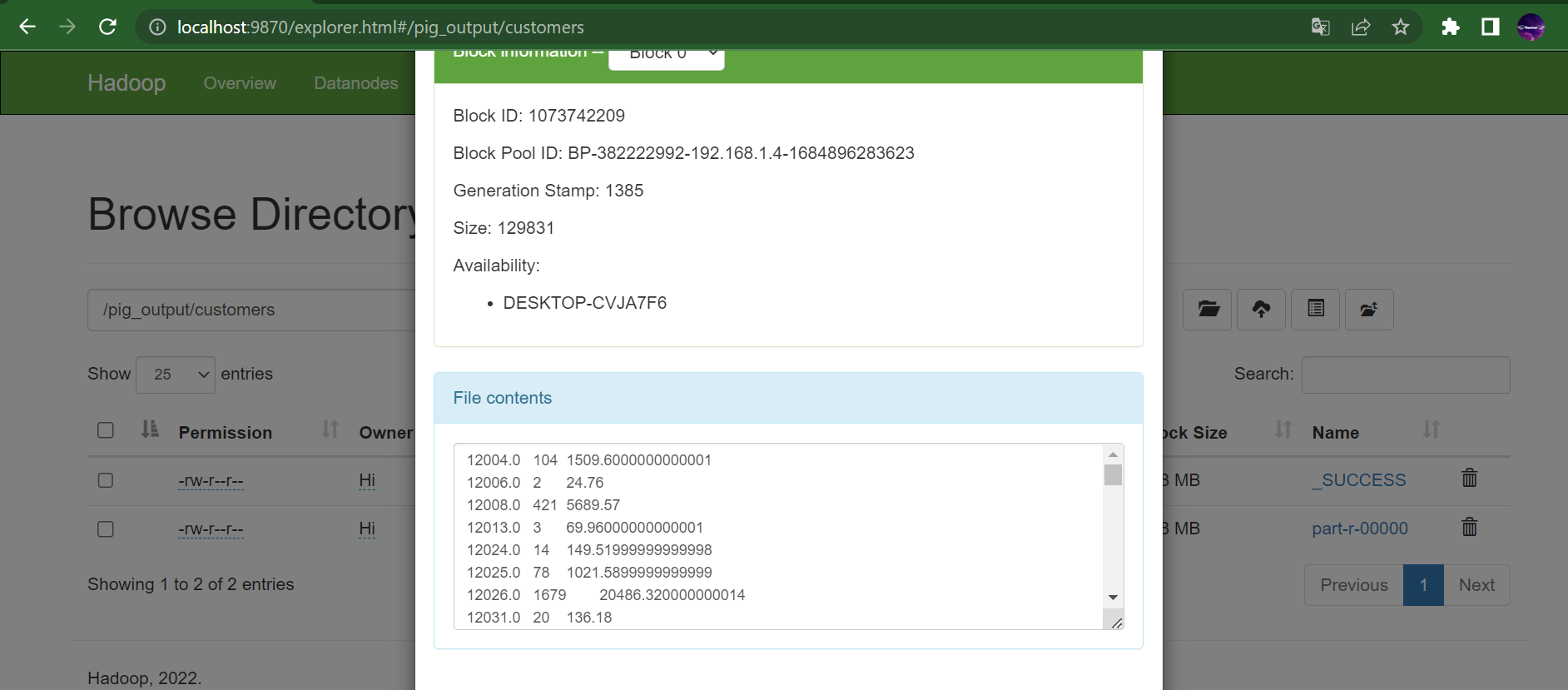


Hoàn thành: sau khi hoàn thành chương trình, file output được lưu vào HDFS ở thư mục /pig\_output





Part-r-00000: xuất ra những dòng đầu tiên của kết quả



# PHẦN 4 – APACHE SPARK:

## 4.1) Miêu tả:

Sau khi thực thi Apache Pig ở mục “*3.7) Tạo bảng khách hàng với thông tin số tiền khách hàng đã chi và số lượng hàng khách hàng đã mua từ trước tới giờ”*, nhóm 2 sẽ tiếp tục thực hiện Apache Spark để phân cụm khách hàng.

## 4.2) Thuật toán gom cụm Kmeans:

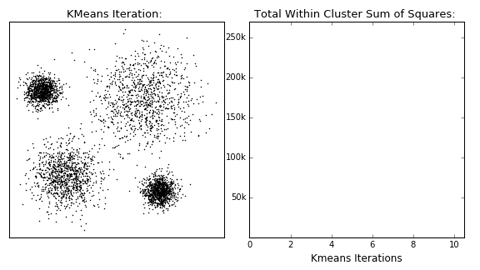
### 4.2.1) Giới thiệu thuật toán Kmeans:

**Đầu vào:** Dữ liệu X và số lượng cluster cần tìm K.

**Đầu ra:** Các center M và label vector cho từng điểm dữ liệu Y.

1. Chọn **K** điểm bất kỳ làm các center ban đầu.
2. Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất.
3. Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.
4. Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất các các điểm dữ liệu đã được gán vào cluster đó sau bước 2.
5. Quay lại bước 2.

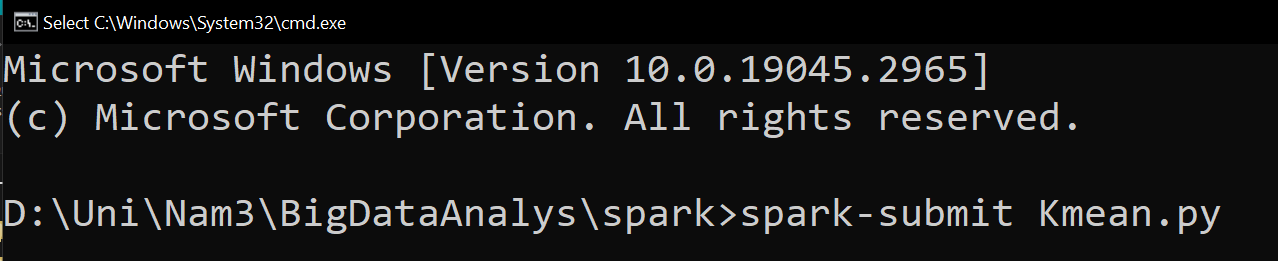
**Miêu tả thuật toán chạy:**



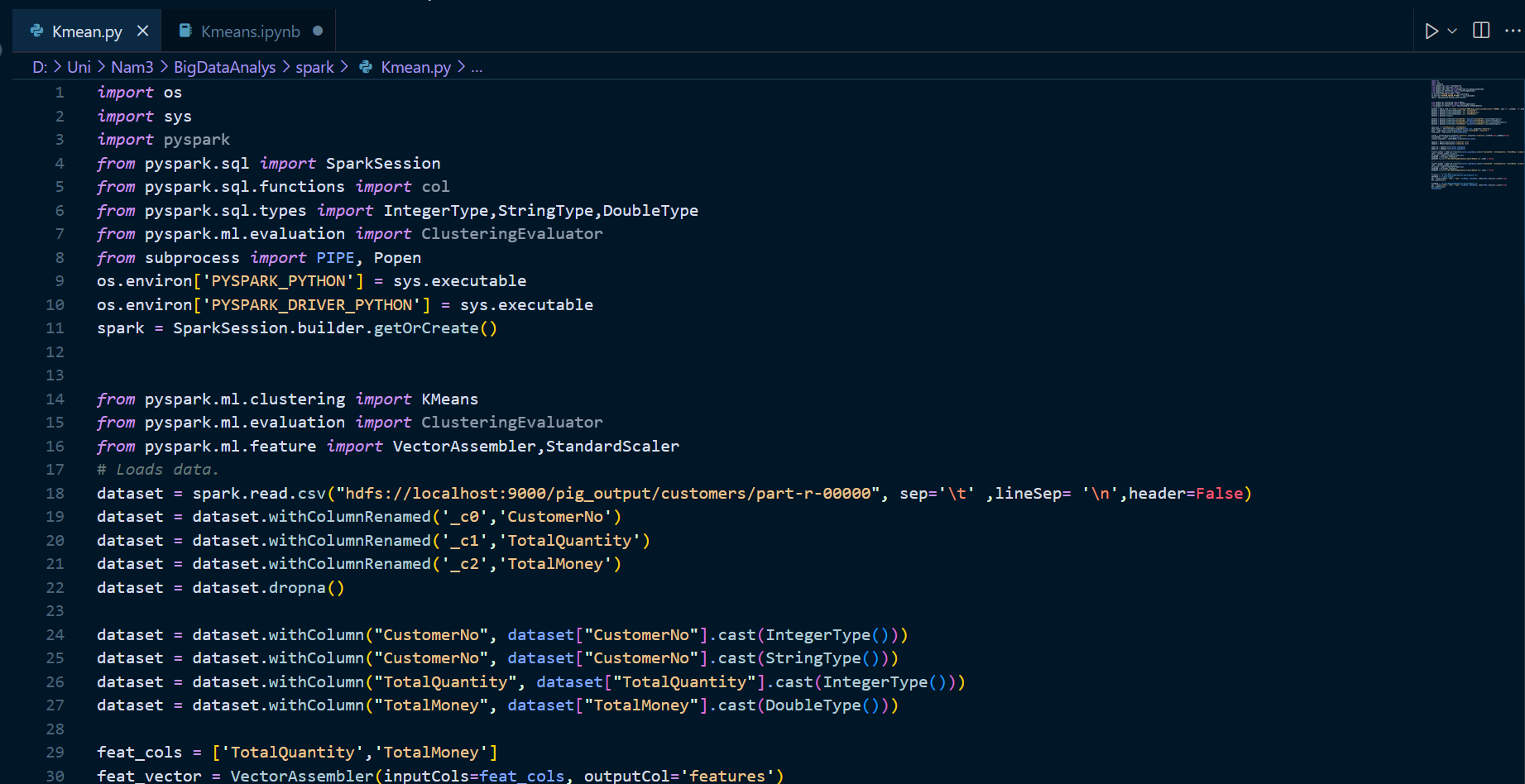
### 4.2.2) Thực thi thuật toán:

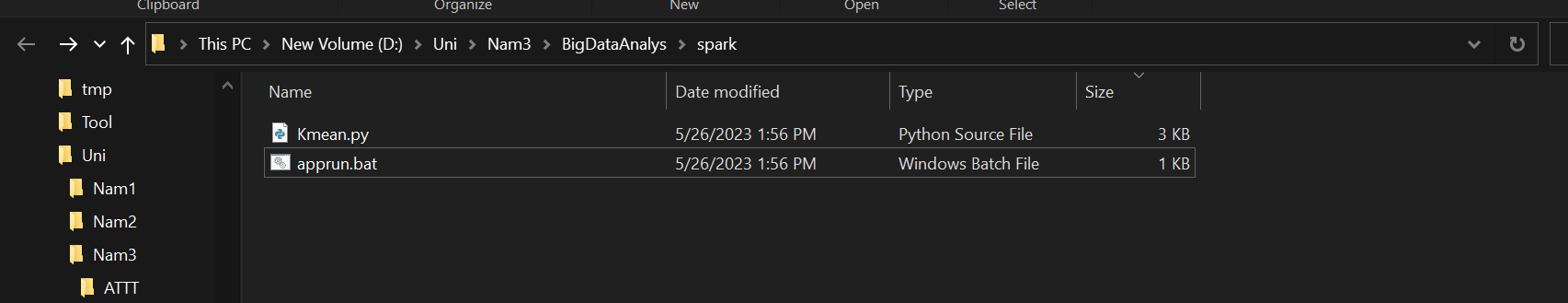
**Bước 1:** Chạy lệnh spark-submit Kmean.py:

Lệnh này sẽ chạy Apache Spark với chế độ Standalone và thực thi Kmean.py.

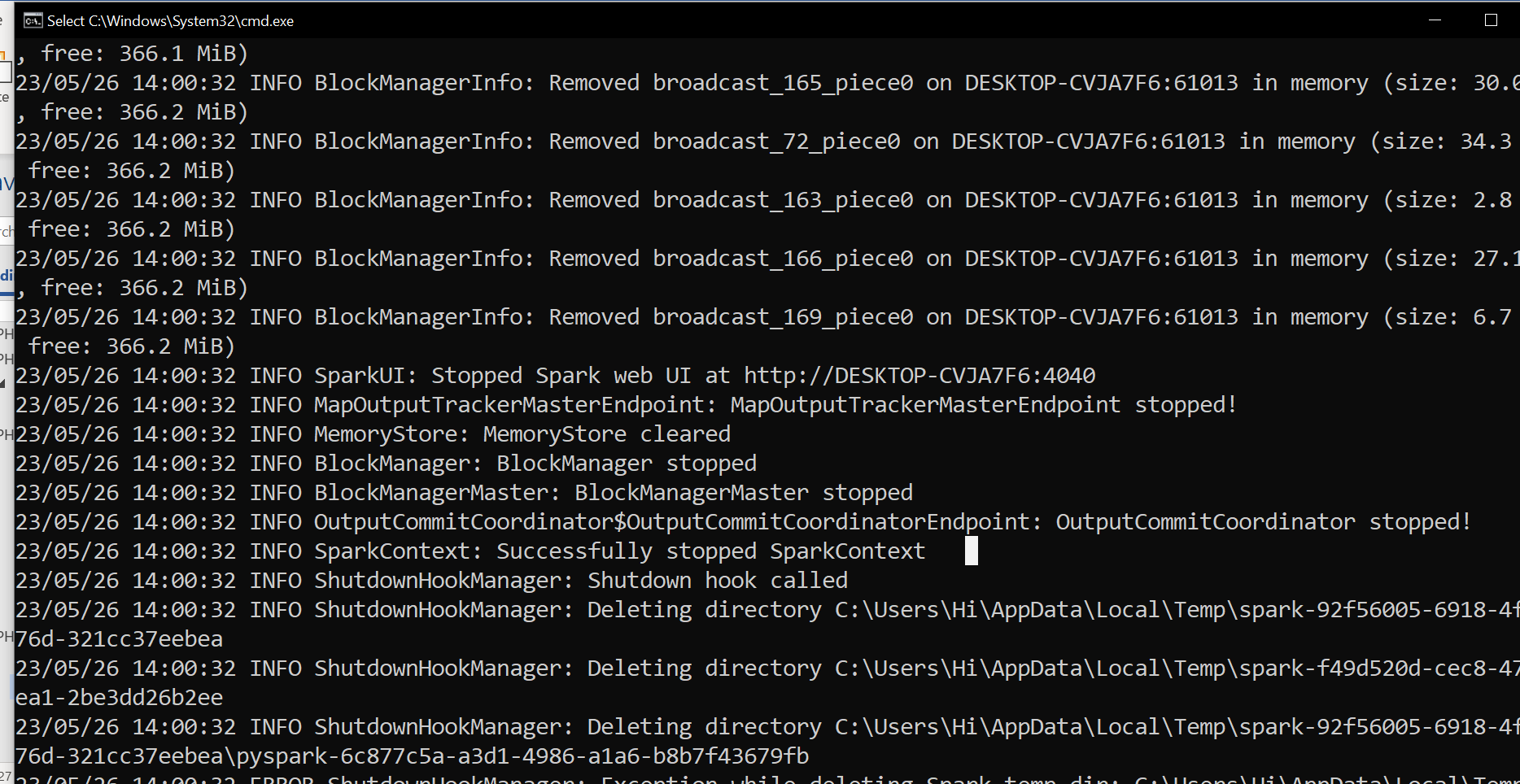


Kmean.py sẽ chạy thuật toán kmean với k = 3 và k = 4 với input là tổng số lượng sản phẩm và tổng tiền đã chi của từng khách hàng.

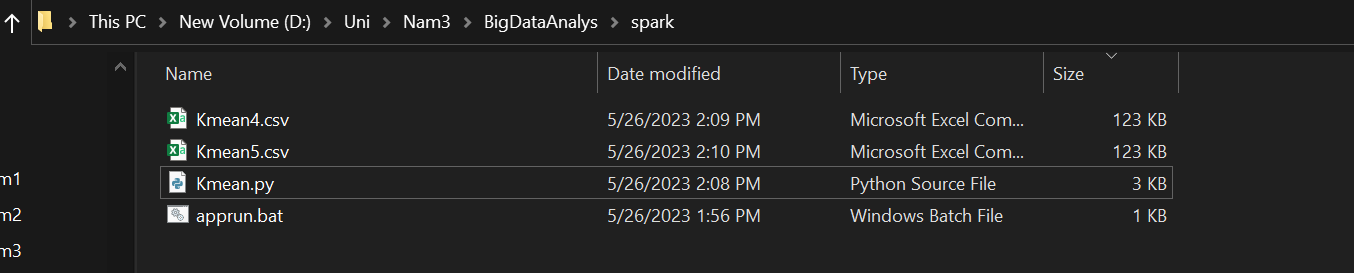




Hoàn thành:

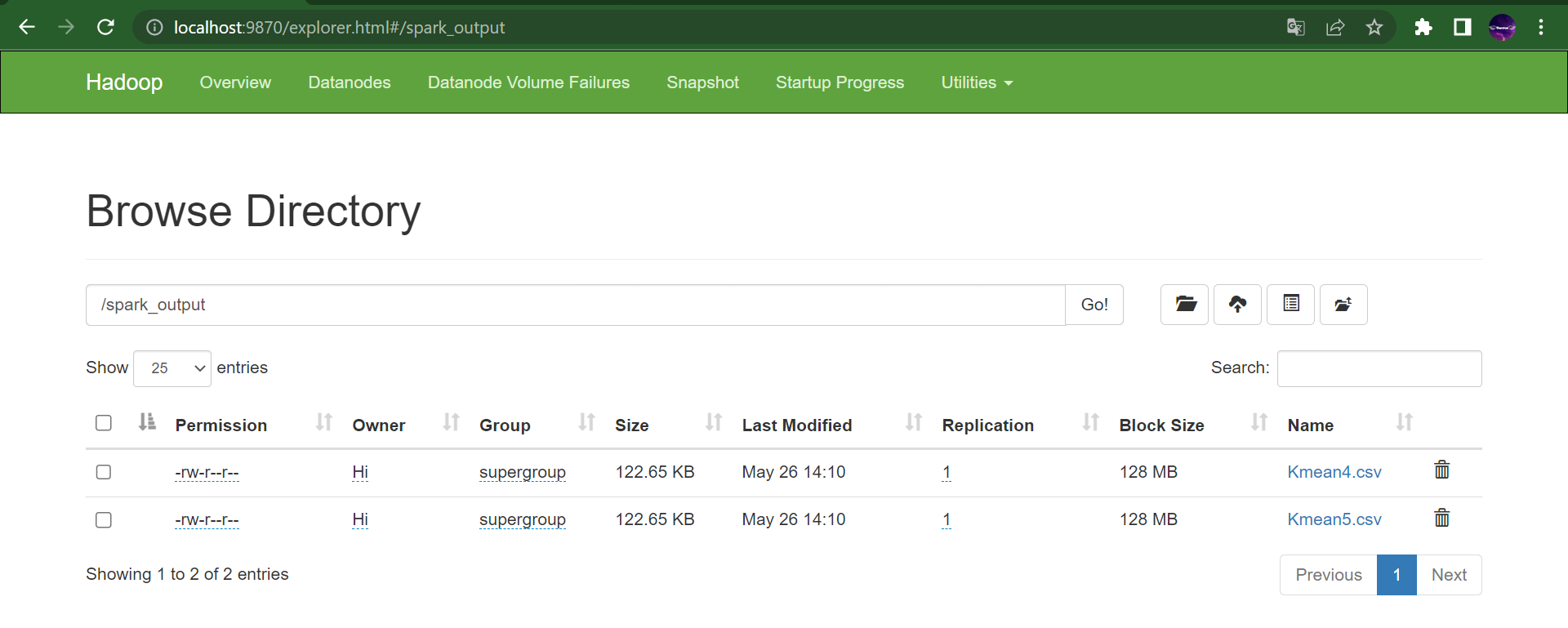


Sau khi chạy spark, sẽ xuất ra 2 file csv là Kmean4 và Kmean5, khách hàng sẽ được phân cụm theo thuật toán kmean với k = 4 và k = 5.

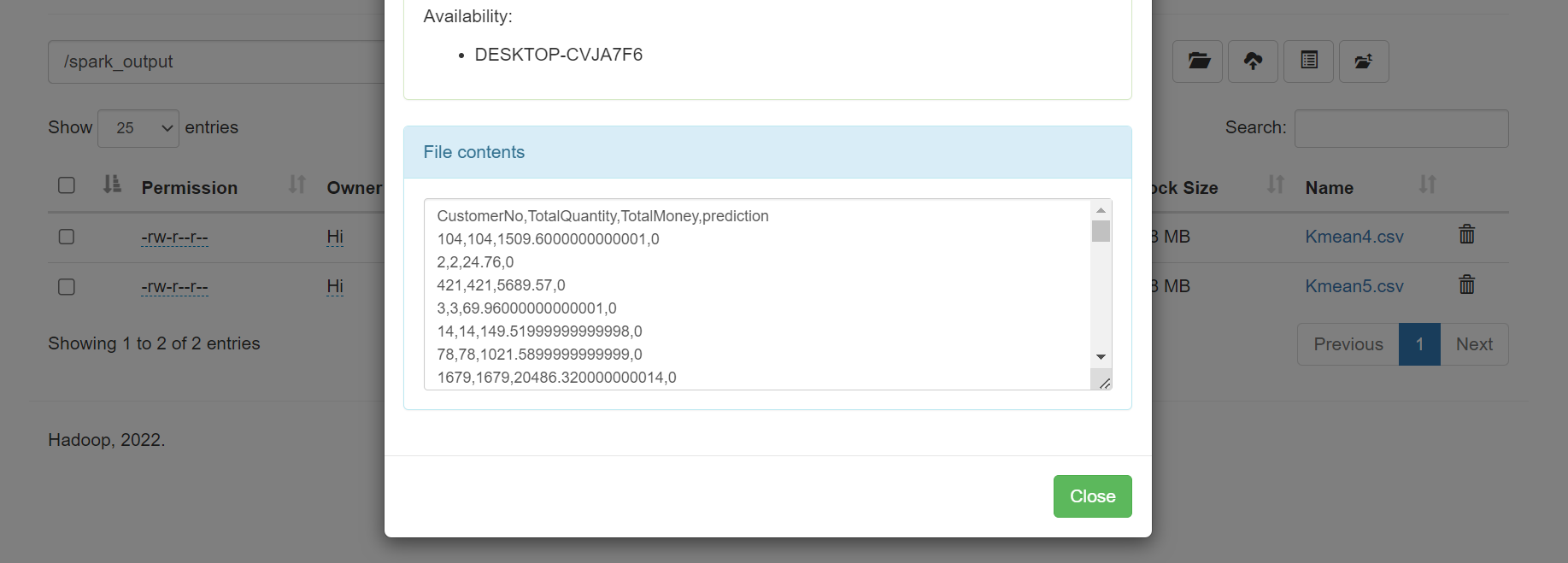


Code python sau đó sẽ tự động đẩy 2 file lên HDFS.

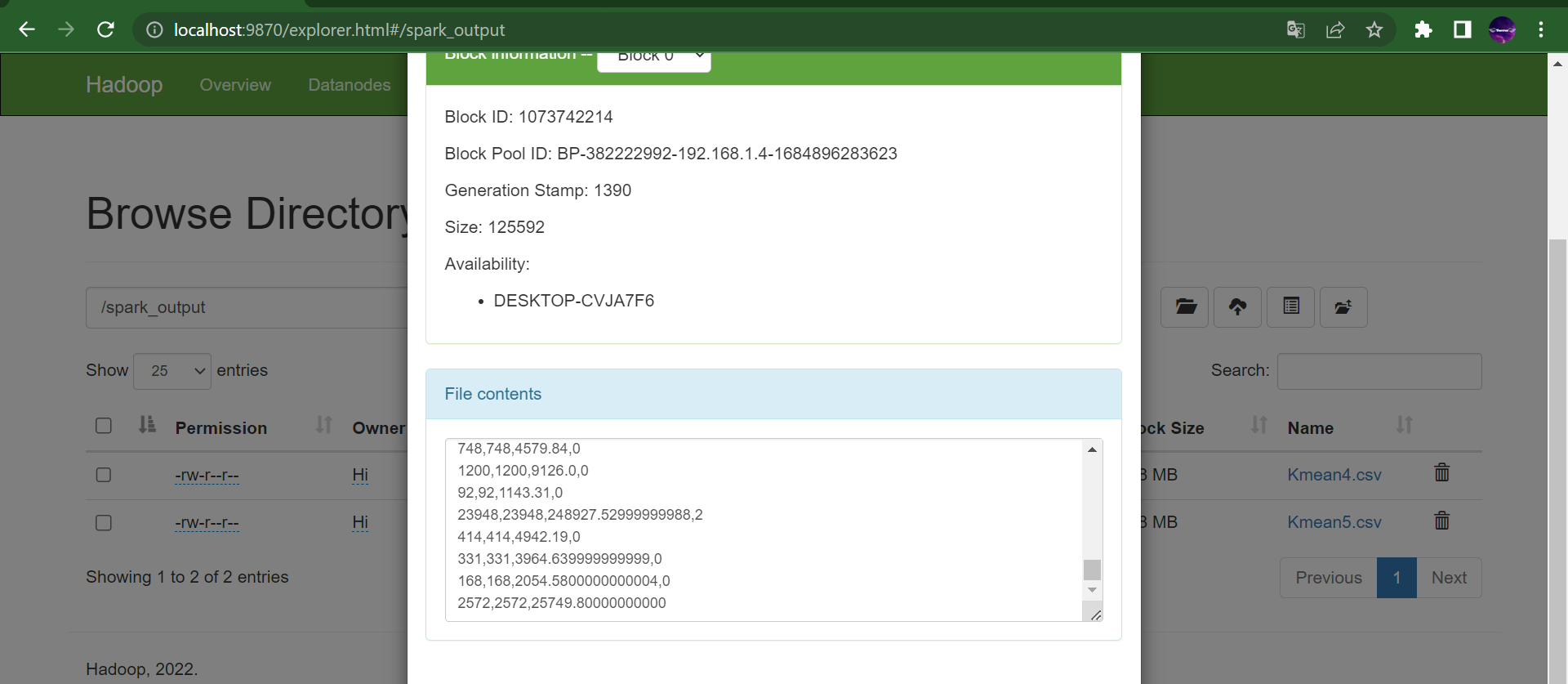
**Bước 2:** Truy cập vào localhost:8080, chọn file spark\_output để kiểm tra 2 file CSV đã được put lên HDFS chưa.



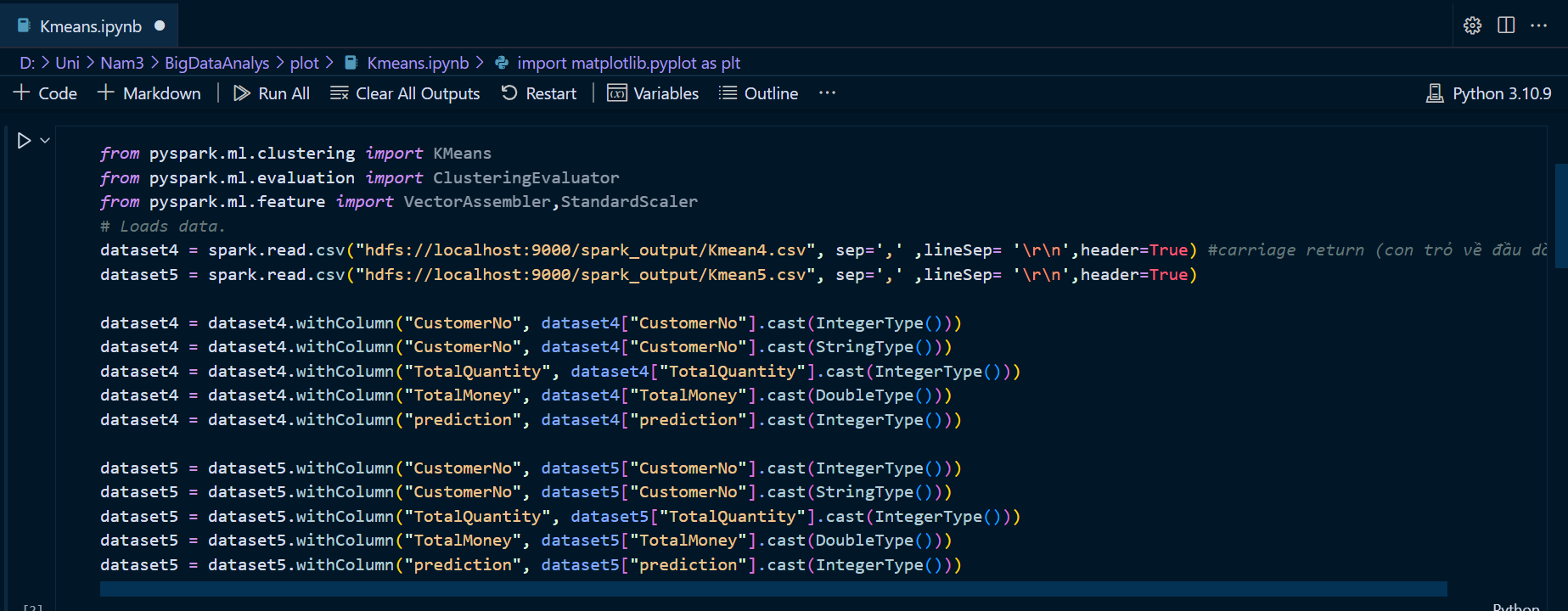
In ra file Kmean4.csv:



In ra file Kmean5.csv:



**Bước 3:** Thực thi Kmeans.ipynb bằng VSCode:

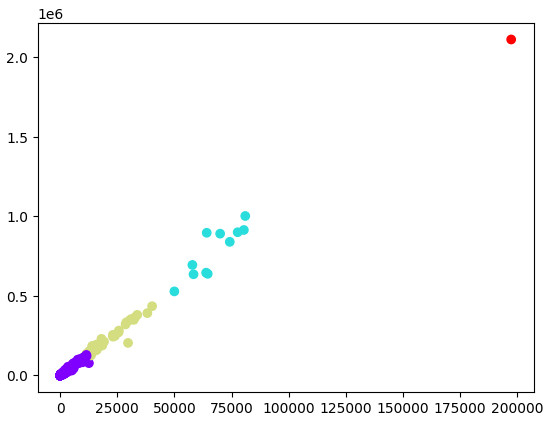


Miêu tả:

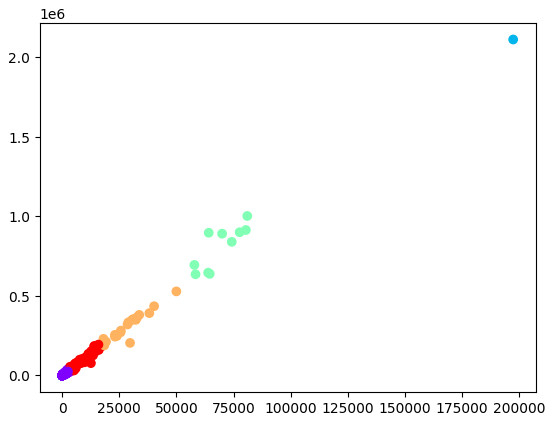
* Kmeans.ipynb sẽ lấy file Kmean4.csv và Kmean5.csv qua đường dẫn “hdfs://localhost:9000/spark\_output/Kmean4.csv” và “hdfs://localhost:9000/spark\_output/Kmean5.csv”
* Chỉnh định dạng của dữ liệu dataset Kmean4.csv và Kmean5.csv.
* Vẽ biểu đồ.

**Kết quả:**

Vẽ biểu đồ với k = 4:



Vẽ biểu đồ với k = 5:

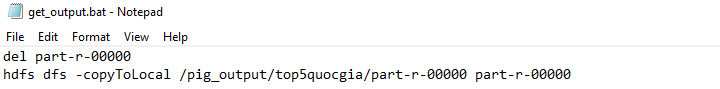


# PHẦN 5 – HIỂN THỊ OUTPUT TRÊN WEB:

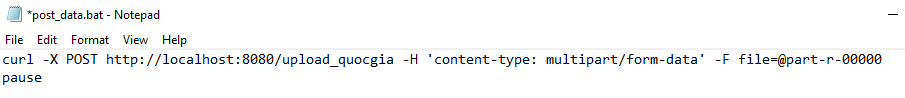
## 5.1) Top 5 quốc gia:

Bước 1: Gửi file kết quả cho top5quocgia cho Spring Project:

copyToLocal sẽ lấy file từ HDFS và copy về local.



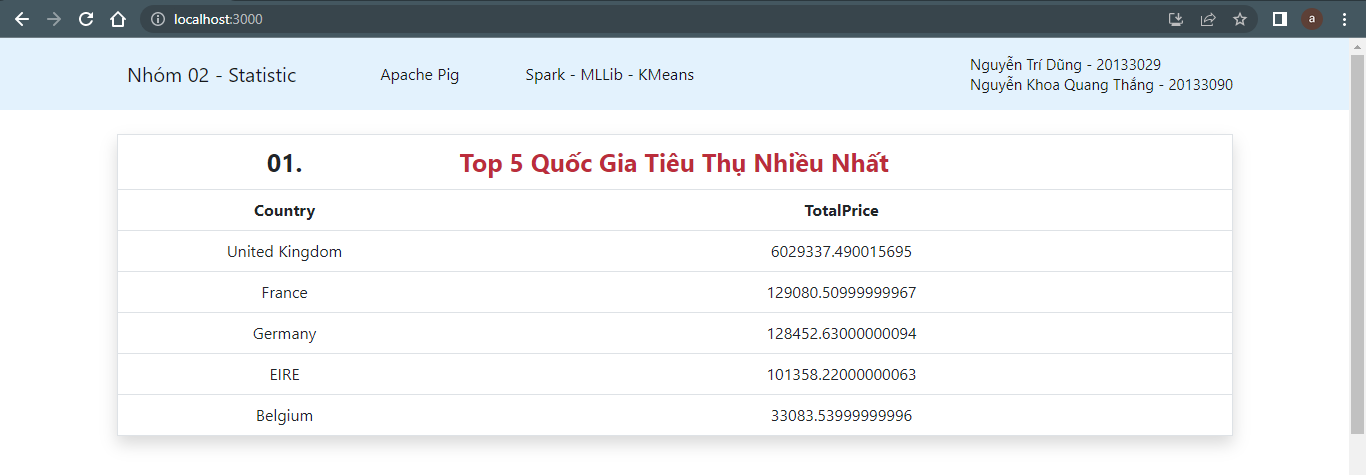
Bước 2: Gửi file vừa lấy được từ HDFS về và gửi vào đường dẫn http://localhost:8080/upload\_quocgia



Bước 3: Sring Project sẽ nhận được file và lưu trữ nội dung của file đó



Bước 4: Hiển thị output trên trang web: ứng dụng ReactJS chạy ở port 3000 sẽ lấy kết quả từ Spring Project và hiển thị lên màn hình



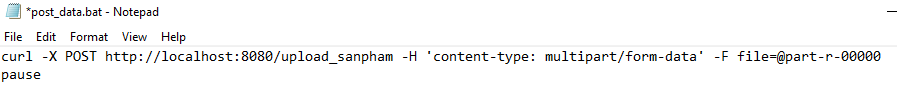
## 5.2) Top 5 sản phẩm được mua nhiều nhất:

Bước 1: Gửi file kết quả cho top5sanpham cho Spring Project:

copyToLocal sẽ lấy file từ HDFS và copy về local.



Bước 2: Gửi file vừa lấy được từ HDFS về và gửi vào đường dẫn http://localhost:8080/upload\_sanpham



Bước 3: Sring Project sẽ nhận được file và lưu trữ nội dung của file đó



Bước 4: Hiển thị output trên trang web: ứng dụng ReactJS chạy ở port 3000 sẽ lấy kết quả từ Spring Project và hiển thị lên màn hình



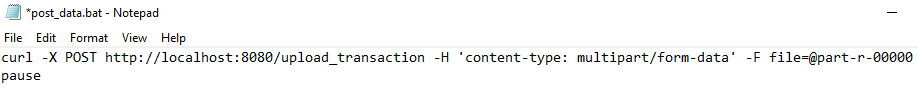
## 5.3) Top 5 giao dịch nhiều nhất:

Bước 1: Gửi file kết quả cho top5transaction cho Spring Project:

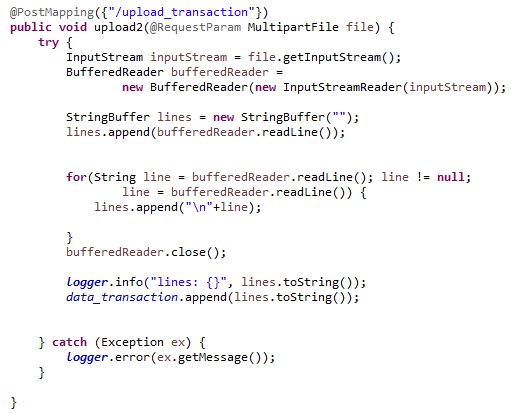
copyToLocal sẽ lấy file từ HDFS và copy về local



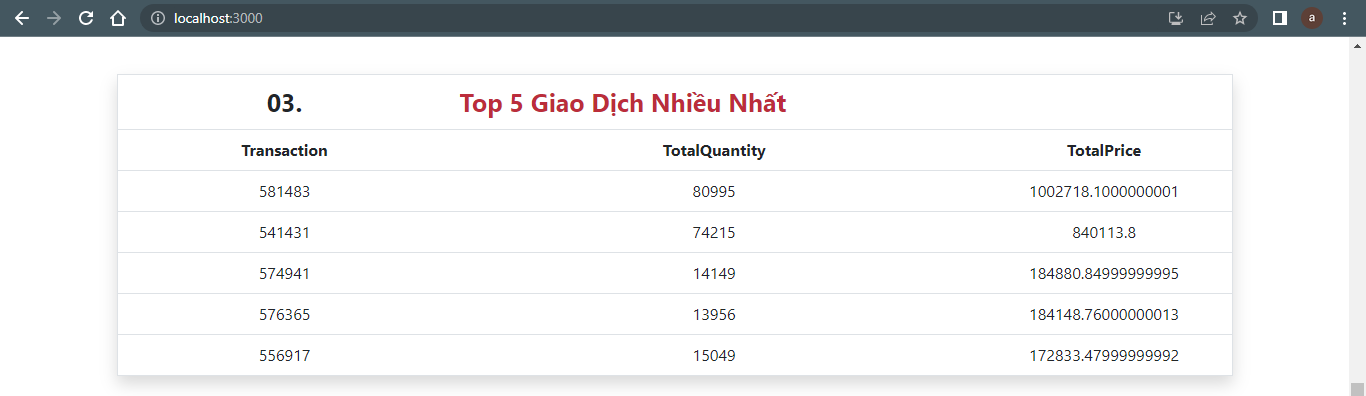
Bước 2: Gửi file vừa lấy được từ HDFS về và gửi vào đường dẫn http://localhost:8080/upload\_transaction



Bước 3: Sring Project sẽ nhận được file và lưu trữ nội dung của file đó



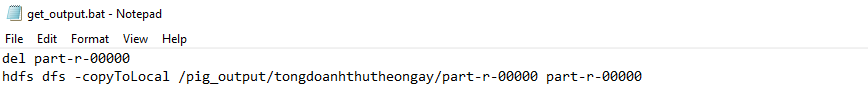
Bước 4: Hiển thị output trên trang web: ứng dụng ReactJS chạy ở port 3000 sẽ lấy kết quả từ Spring Project và hiển thị lên màn hình



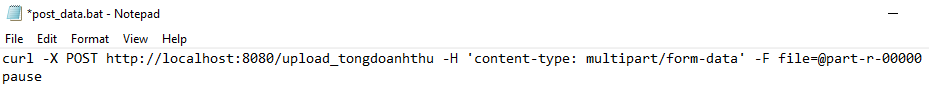
## 5.4) Tổng doanh thu 10 ngày cao nhất:

Bước 1: Gửi file kết quả cho tongdoanhthutheongay cho Spring Project:

copyToLocal sẽ lấy file từ HDFS và copy về local



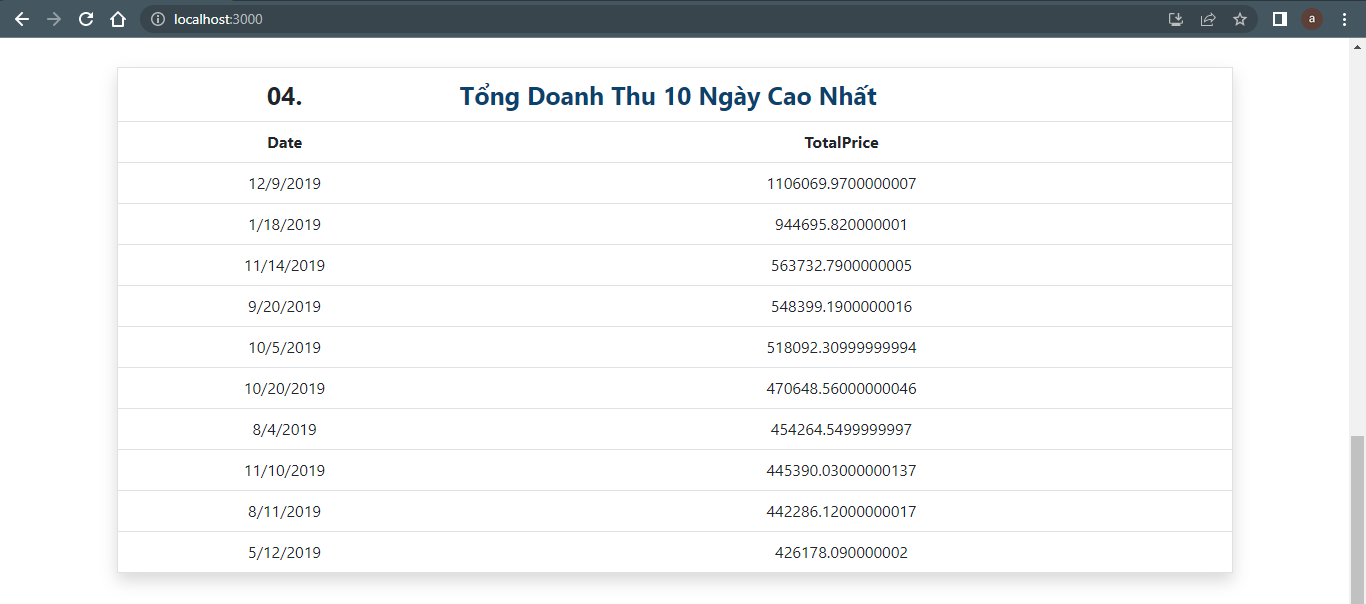
Bước 2: Gửi file vừa lấy được từ HDFS về và gửi vào đường dẫn http://localhost:8080/upload\_tongdoanhthu



Bước 3: Sring Project sẽ nhận được file và lưu trữ nội dung của file đó



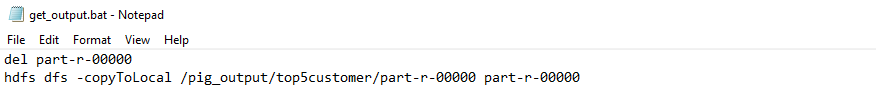
Bước 4: Hiển thị output trên trang web: ứng dụng ReactJS chạy ở port 3000 sẽ lấy kết quả từ Spring Project và hiển thị lên màn hình



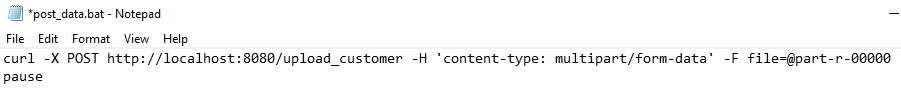
## 5.5) Top 5 khách hàng mua nhiều nhất:

Bước 1: Gửi file kết quả cho top5customer cho Spring Project:

copyToLocal sẽ lấy file từ HDFS và copy về local



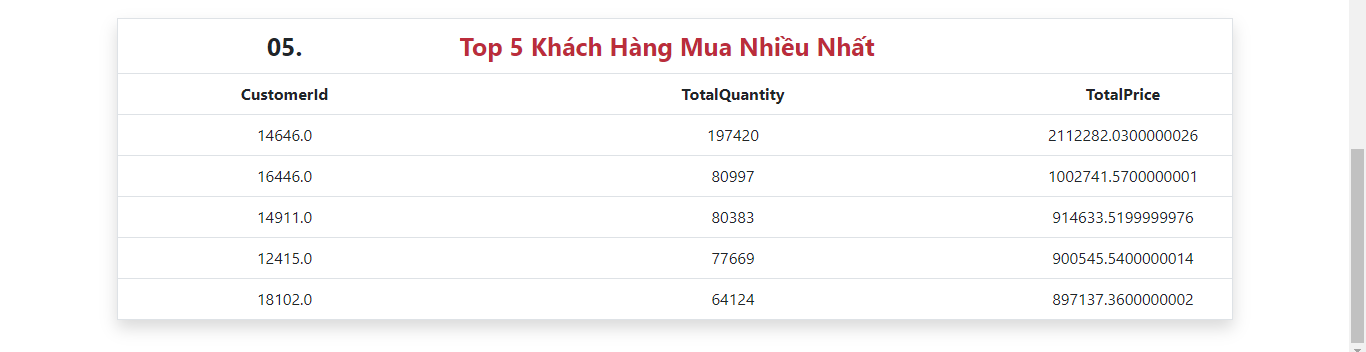
Bước 2: Gửi file vừa lấy được từ HDFS về và gửi vào đường dẫn http://localhost:8080/upload\_customer



Bước 3: Sring Project sẽ nhận được file và lưu trữ nội dung của file đó



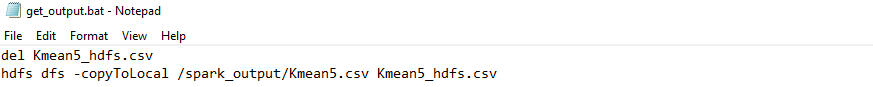
Bước 4: Hiển thị output trên trang web: ứng dụng ReactJS chạy ở port 3000 sẽ lấy kết quả từ Spring Project và hiển thị lên màn hình



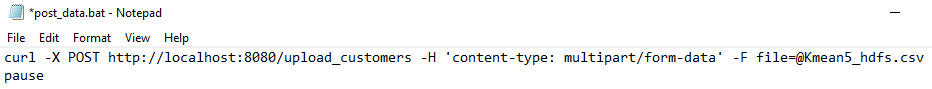
## 5.6) Gom cụm khách hàng:

Bước 1: Gửi file kết quả cho phân cụm customer cho Spring Project:

copyToLocal sẽ lấy file từ HDFS và copy về local



Bước 2: Gửi file vừa lấy được từ HDFS về và gửi vào đường dẫn http://localhost:8080/upload\_quocgia



Bước 3: Sring Project sẽ nhận được file và lưu trữ nội dung của file đó



Bước 4: Hiển thị output trên trang web: ứng dụng ReactJS chạy ở port 3000 sẽ lấy kết quả từ Spring Project và hiển thị lên màn hình



# TÀI LIỆU THAM KHẢO

1. Giới thiệu VSCode: <https://vi.wikipedia.org/wiki/Visual_Studio_Code>
2. Giới thiệu Python: <https://aws.amazon.com/vi/what-is/python/>
3. Giới thiệu PySpark và Apache Spark: <https://www.dominodatalab.com/data-science-dictionary/pyspark>
4. Giới thiệu Apache Pig: <https://renovacloud.com/phan-tich-du-lieu-lon-voi-apache-pig-tren-aws-elastic-mapreduce/>
5. Apache Pig Project: <https://www.cloudduggu.com/pig/project-01/>
6. Kmean Spark: <https://www.youtube.com/watch?v=etE5hQfzd9o>