Demo sản phẩm

A. Stream Layer

Step 1: Chạy kafka
>>> Chạy apache zookeeper

/usr/local/kafka/bin/zookeeper-server-start.sh /usr/local/kafka/config/zookeeper.properties

>>> Chạy kafka server

/usr/local/kafka/bin/kafka-server-start.sh /usr/local/kafka/config/server.properties

>>> Chạy create kafka topic

/usr/local/kafka/bin/kafka-topics.sh --create --topic smartphoneTopic --bootstrap-server localhost:9092

>>> Chạy kafka producer cho stream layer

/usr/local/kafka/bin/kafka-console-producer.sh --topic smartphoneTopic --bootstrap-server localhost:9092

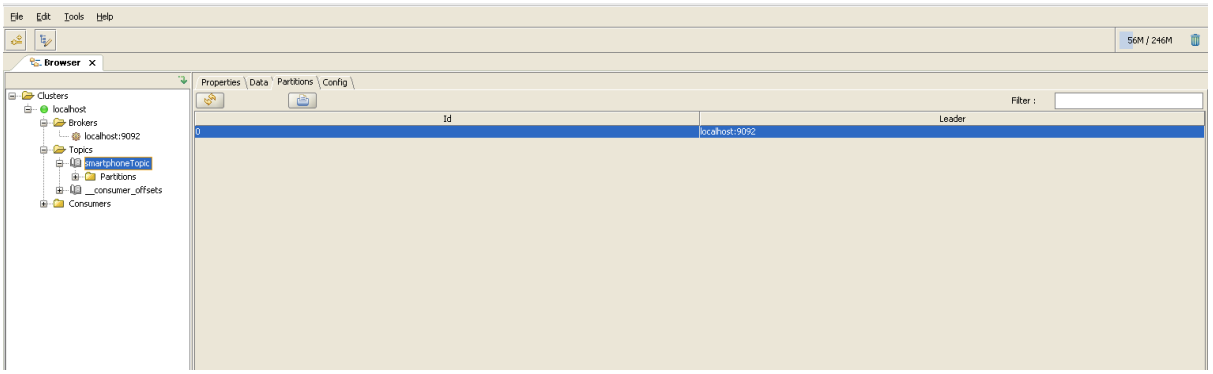
>>> Chạy kafka consumer cho stream layer

/usr/local/kafka/bin/kafka-console-consumer.sh --topic smartphoneTopic --from-beginning --bootstrap-server localhost:9092

>>> Sau khi chạy xong có thể kiểm tra kafka trên Kafka tool Offset Explorer 3.0
>>> Mở kafka tool:
cd /
cd ~/.wine/drive_c/'Program Files'/OffsetExplorer3
wine offsetexplorer.exe

>>> thêm cluster : localhost - 127.0.0.1:9092

>>> Đây là hình ảnh minh họa kafka đã hoạt động, topic “smartphoneTopic” đã được tạo ra, và có 1 partition id = 0 (do cấu hình)

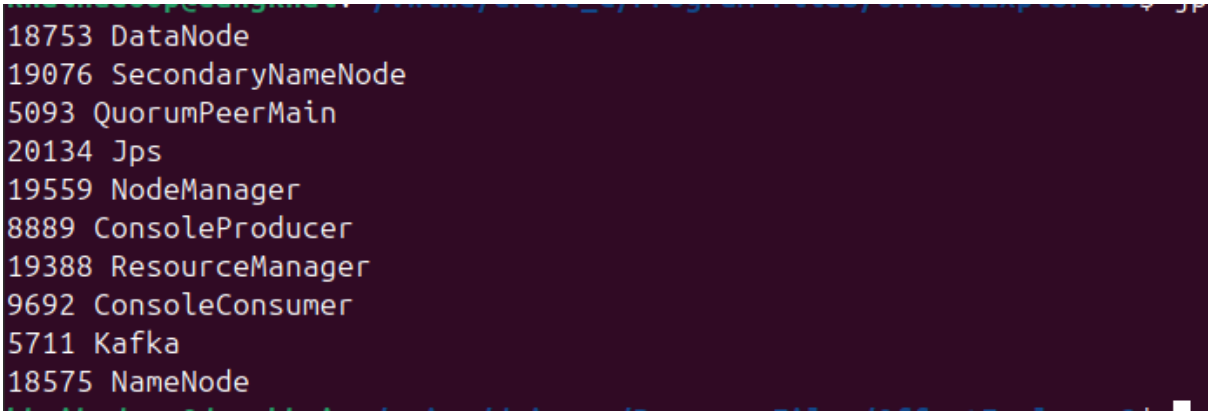


Step 2: start HDFS, YARN

start-all.sh

>>> Minh chứng là HDFS, YARN đã chạy

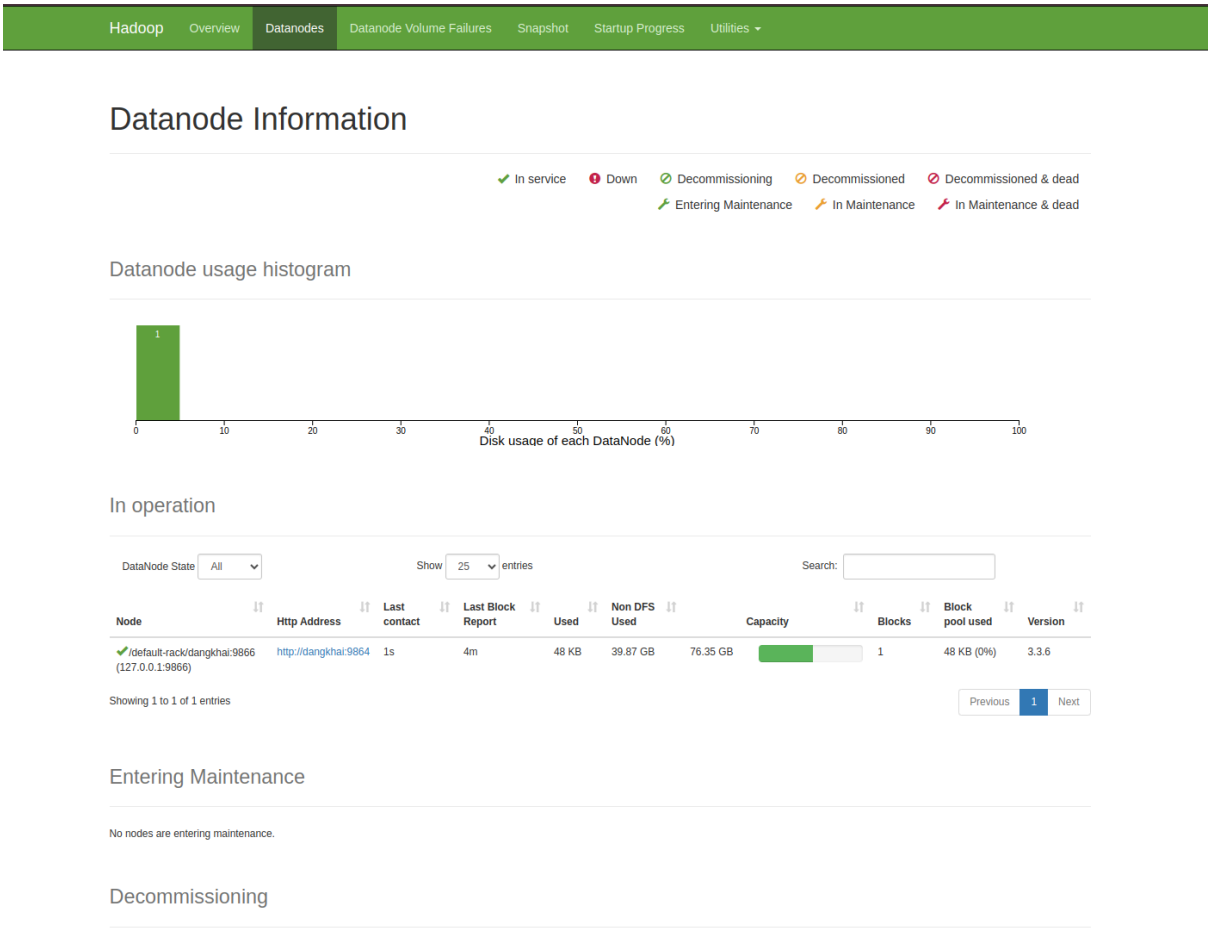
jps



>>> Mở web UI của Hadoop

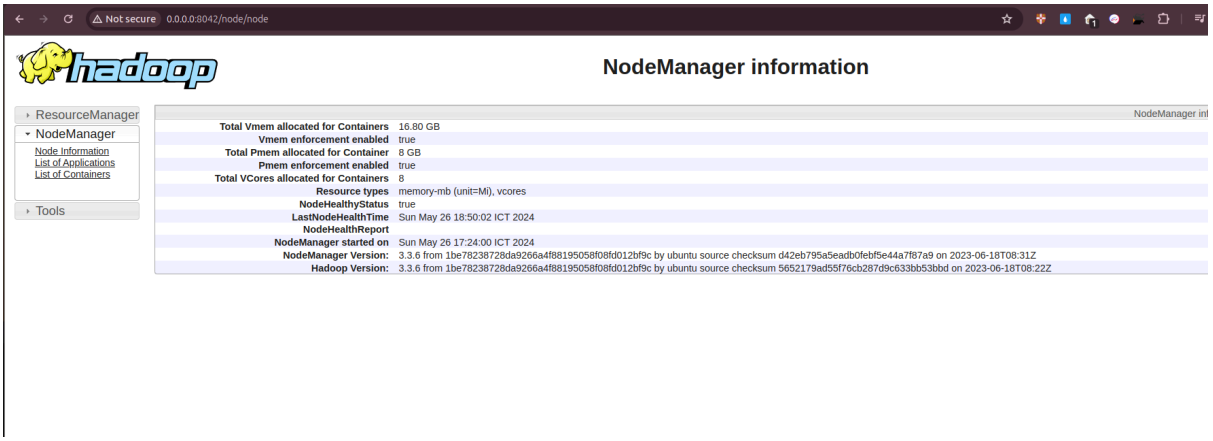
https://localhost:9870 ⇒ Namenode information

>>> Last Contact (1s), Bộ nhớ chiếm dụng - Used, ..



>>> Thong tin Nodemanager

https://localhost:8042/



Step 3 : Mở hbase, Run thrift server (for Hbase)

/usr/local/hbase/bin/start-hbase.sh

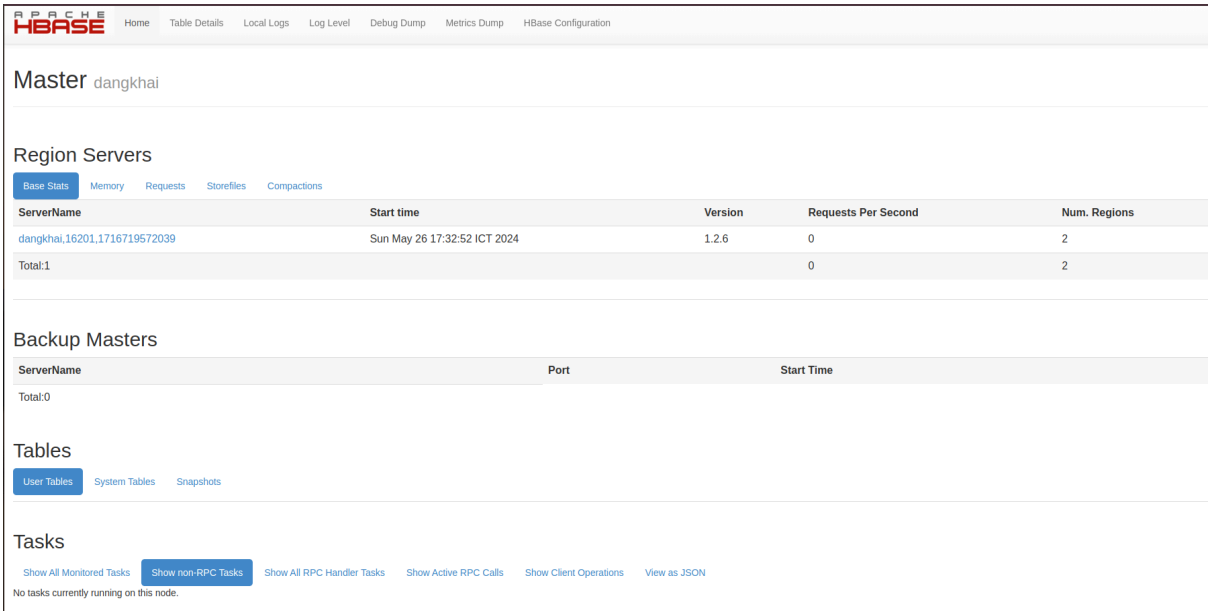
/usr/local/hbase/bin/hbase thrift start

```
khaihadoop@dangkhai:~$ jps
18753 DataNode
19076 SecondaryNameNode
5093 QuorumPeerMain
25478 HMaster
19559 NodeManager
25640 HRegionServer
8889 ConsoleProducer
19388 ResourceManager
9692 ConsoleConsumer
26254 Jps
5711 Kafka
18575 NameNode
```

>>> Web UI của hbase

0.0.0.0:16010

>>> Num region = 2



>>> Kiểm tra hbase trên terminal

1. Mở hbase
   hbase shell
2. Kiểm tra có table nào không
   list

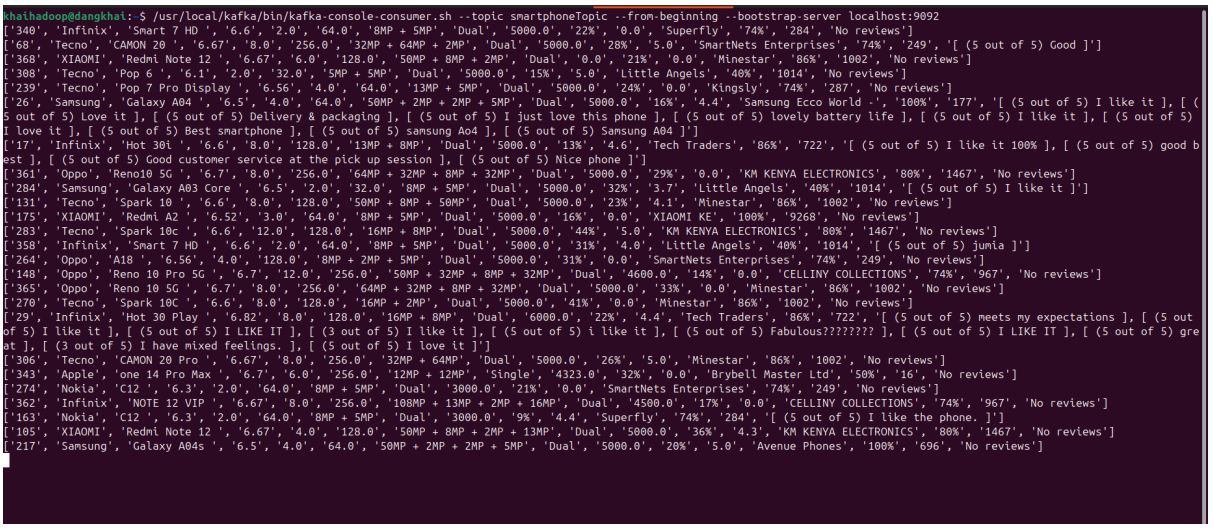
```
hbase(main):001:0> list
TABLE
0 row(s) in 0.1820 seconds

=> []
hbase(main):002:0>
```

>>> như vậy là chưa có table nào được tạo trên hbase

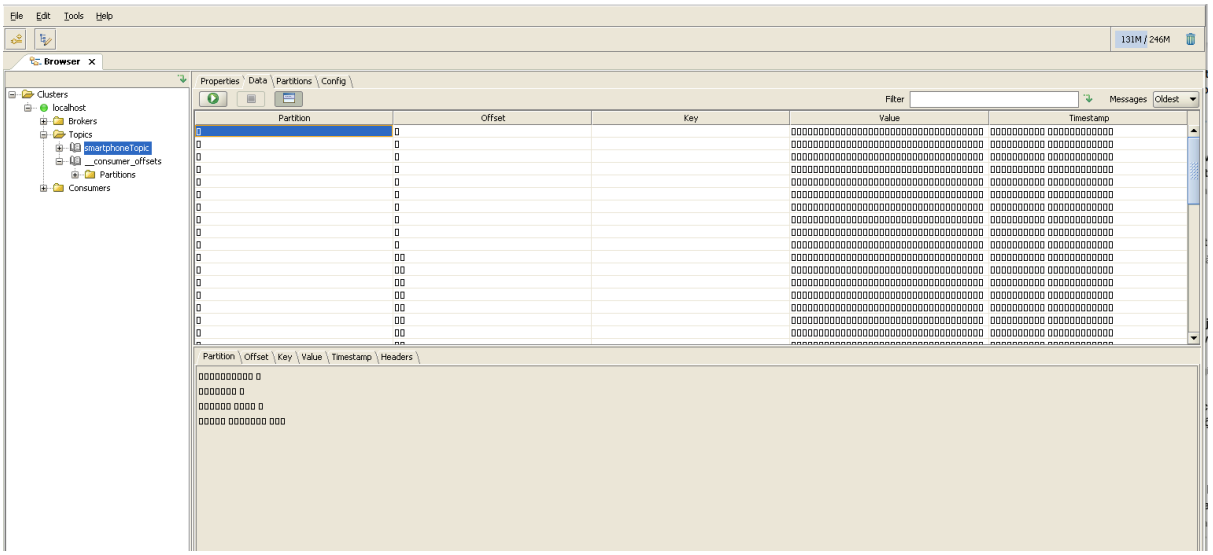
Step 4: Sau tất cả, chạy `stream_pipeline.py` script.

>>>Data đã bắt đầu được đẩy lên “smartphoneTopic” của kafka, tiếp đến là qua Hbase

>>> Đây là terminal của consumer kafka:



>>> kafka ở trên tool đã thay đổi, có vẻ bị mã hóa



>>> kiểm tra hbase đã thêm được table “smartphone”

list

```
hbase(main):002:0> list
TABLE
smartphone
1 row(s) in 0.0160 seconds

=> ["smartphone"]
hbase(main):003:0>
```

>>> Table liên tục được cập nhật, trong COLUMN+CELL đã xuất hiện key-value của price ⇒ data đã đi qua thành công model XGBoost

scan “smartphone”

```
hbase(main):003:0> scan "smartphone"
ROW                    COLUMN+CELL
 20240526181951901395 column=info:Battery, timestamp=1716722391980, value=5000.0
 20240526181951901395 column=info:Brand, timestamp=1716722391980, value=Tecno
 20240526181951901395 column=info:Price, timestamp=1716722391980, value=2450.357
                      7
 20240526181951901395 column=info:RAM, timestamp=1716722391980, value=8.0
 20240526181951901395 column=info:Screen_size, timestamp=1716722391980, value=6.
                      67
 20240526181951901395 column=info:Sim_type, timestamp=1716722391980, value=Dual
 20240526181951901395 column=info:Storage, timestamp=1716722391980, value=256.0
 20240526181951901395 column=info:date, timestamp=1716722391980, value=2024-05-2
                      6 18:19:51
 20240526181953255358 column=info:Battery, timestamp=1716722393266, value=0.0
 20240526181953255358 column=info:Brand, timestamp=1716722393266, value=XIAOMI
 20240526181953255358 column=info:Price, timestamp=1716722393266, value=2120.245
                      6
 20240526181953255358 column=info:RAM, timestamp=1716722393266, value=6.0
 20240526181953255358 column=info:Screen_size, timestamp=1716722393266, value=6.
                      67
 20240526181953255358 column=info:Sim_type, timestamp=1716722393266, value=Dual
 20240526181953255358 column=info:Storage, timestamp=1716722393266, value=128.0
 20240526181953255358 column=info:date, timestamp=1716722393266, value=2024-05-2
                      6 18:19:53
 20240526181958321898 column=info:Battery, timestamp=1716722398333, value=5000.0
 20240526181958321898 column=info:Brand, timestamp=1716722398333, value=Tecno
 20240526181958321898 column=info:Price, timestamp=1716722398333, value=875.1683
                      3
 20240526181958321898 column=info:RAM, timestamp=1716722398333, value=2.0
 20240526181958321898 column=info:Screen_size, timestamp=1716722398333, value=6.
                      1
 20240526181958321898 column=info:Sim_type, timestamp=1716722398333, value=Dual
 20240526181958321898 column=info:Storage, timestamp=1716722398333, value=32.0
 20240526181958321898 column=info:date, timestamp=1716722398333, value=2024-05-2
                      6 18:19:58
 20240526182003252062 column=info:Battery, timestamp=1716722403270, value=5000.0
 20240526182003252062 column=info:Brand, timestamp=1716722403270, value=Tecno
 20240526182003252062 column=info:Price, timestamp=1716722403270, value=1185.370
                      6
 20240526182003252062 column=info:RAM, timestamp=1716722403270, value=4.0
 20240526182003252062 column=info:Screen_size, timestamp=1716722403270, value=6.
                      56
```

>>> Trên Web UI (0.0.0.0:16010) của Hbase, số lượng request liên tục thay đổi

Table smartphone

Table Attributes

Attribute Name	Value	Description
Enabled	true	Is the table enabled
Compaction	NONE	Is the table compacting

Table Regions

Name	Region Server	Start Key	End Key	Locality	Requests
smartphone,,1716722389610.4ff9cfdc0d0f7a7aeb5a001e14c7c9a4.	dangkhai:16301			0.0	259

Regions by Region Server

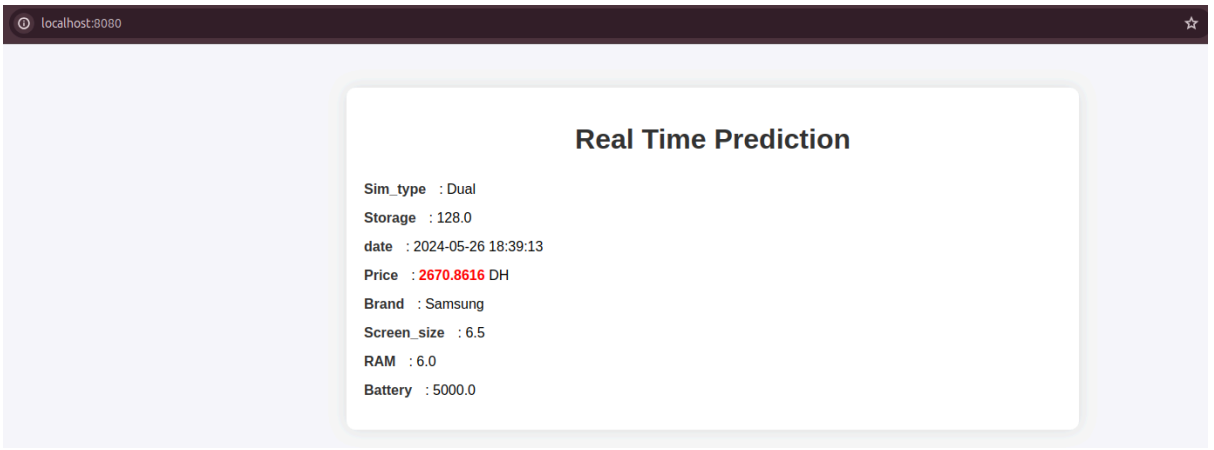
Region Server	Region Count
dangkhai:16301	1

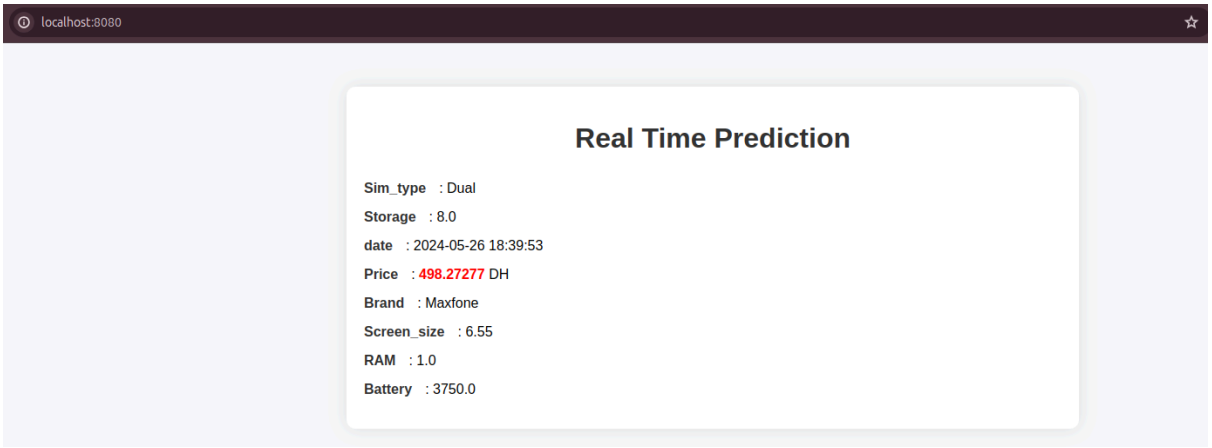
Step 6: Mở ứng dụng Spring Boot Suit 4 để chạy giao diện của lớp Stream layer

- Mở app spring boot suit
- Run as ‘spring boot app”
- Giao diện spring boot suit

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.2.3)

2024-05-26T18:36:50.983+07:00  INFO 67721 --- [real_time_app] [           main] c.example.demo.RealTimeAppApplication    : Starting RealTimeAppApplication using Java 17.0.11 with PID 67721 (/home/khaihad
2024-05-26T18:36:50.989+07:00  INFO 67721 --- [real_time_app] [           main] c.example.demo.RealTimeAppApplication    : No active profile set, falling back to 1 default profile: "default"
2024-05-26T18:36:52.219+07:00  INFO 67721 --- [real_time_app] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8080 (http)
2024-05-26T18:36:52.238+07:00  INFO 67721 --- [real_time_app] [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-05-26T18:36:52.239+07:00  INFO 67721 --- [real_time_app] [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.19]
2024-05-26T18:36:52.325+07:00  INFO 67721 --- [real_time_app] [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2024-05-26T18:36:52.326+07:00  INFO 67721 --- [real_time_app] [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1269 ms
2024-05-26T18:36:52.510+07:00  INFO 67721 --- [real_time_app] [           main] o.s.b.a.w.s.WelcomePageHandlerMapping    : Adding welcome page template: index
2024-05-26T18:36:52.858+07:00  INFO 67721 --- [real_time_app] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context path ''
2024-05-26T18:36:52.870+07:00  INFO 67721 --- [real_time_app] [           main] c.example.demo.RealTimeAppApplication    : Started RealTimeAppApplication in 2.365 seconds (process running for 3.599)
```

>>> Mở https://localhost:8080

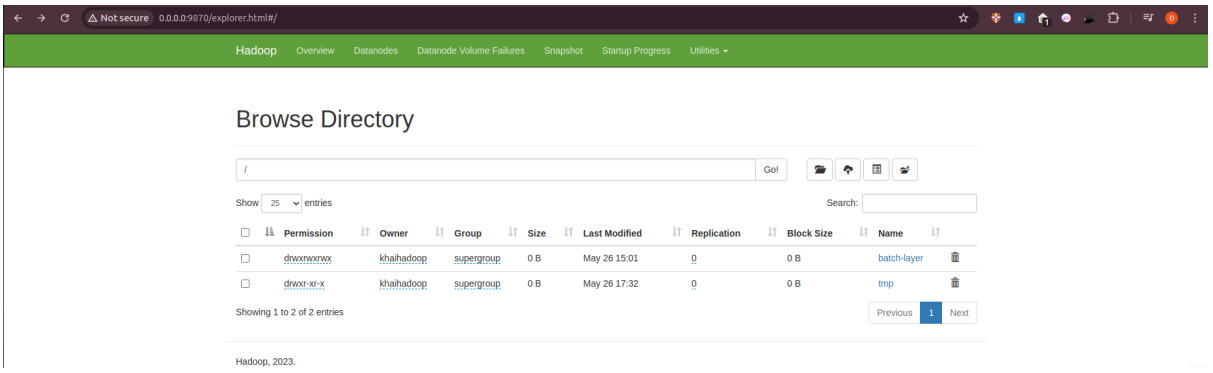




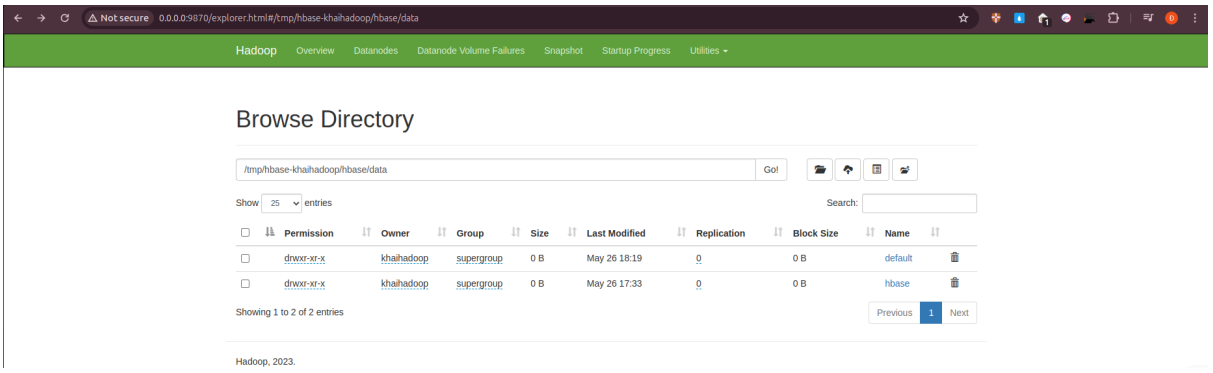
⇒ OK, data đã xuất hiện liên tục theo thời gian (real-time)

>>> Data steam đã được lưu trên hdfs trong thu muc /tmp

https://localhost:9870/



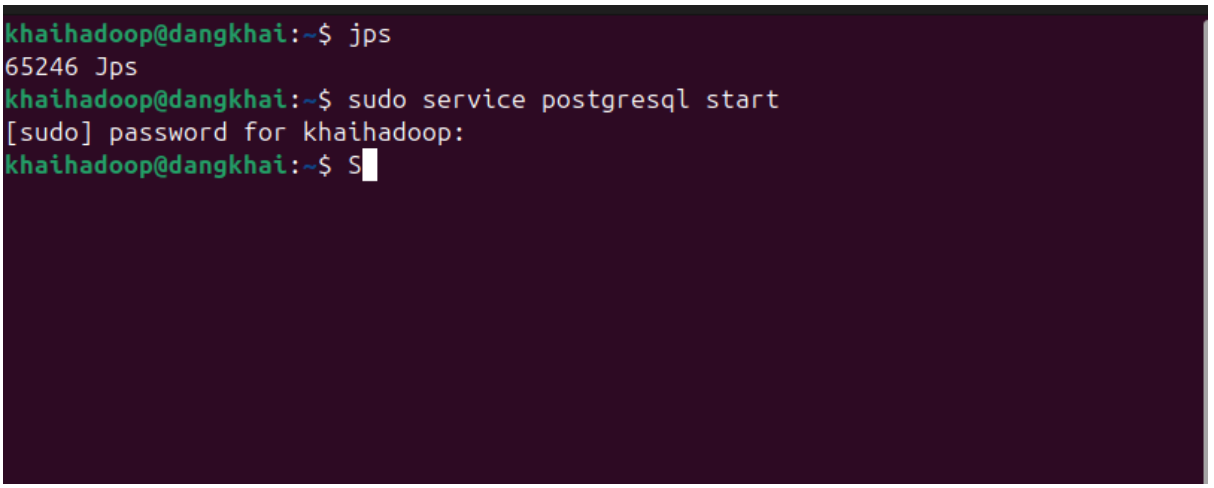
>>> vào /tmp/hbase-khaihadoop/hbase/data/



B. Batch-layer

Step 1: Chắc chắn là postgresQL đang chạy

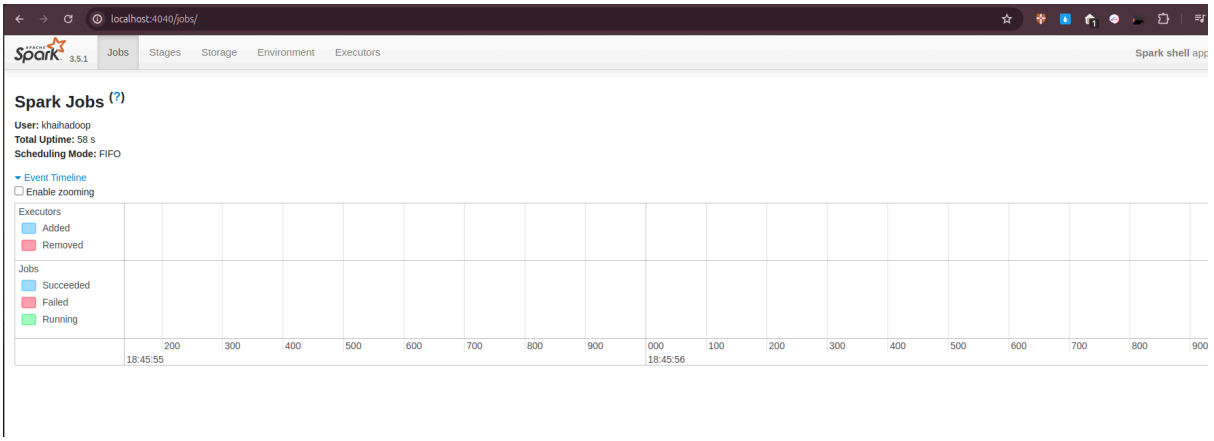
sudo service postgresql start



Step 2 : start apache spark

spark-shell

>>> Mở https://localhost:4040/
>>> kiểm tra



Step 3: chạy kafka, mở ra producers, consumers cho batch layer

/usr/local/kafka/bin/zookeeper-server-start.sh /usr/local/kafka/config/zookeeper.properties

/usr/local/kafka/bin/kafka-server-start.sh /usr/local/kafka/config/server.properties

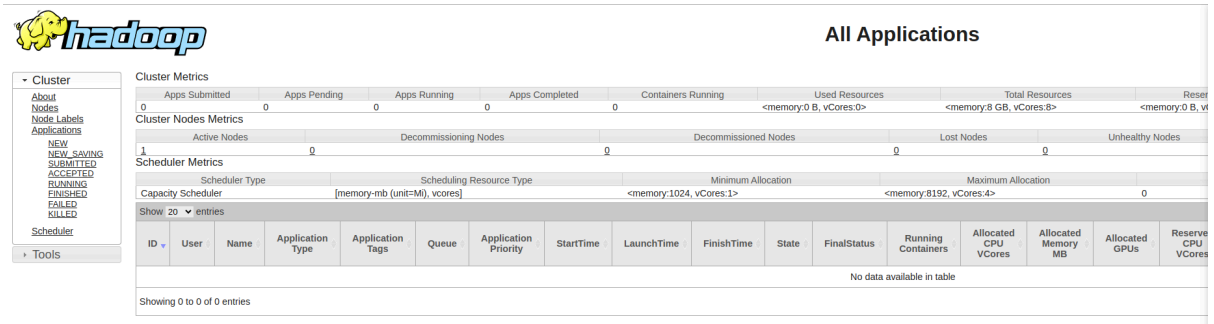
/usr/local/kafka/bin/kafka-console-producer.sh --topic smartphoneTopic --bootstrap-server localhost:9092

/usr/local/kafka/bin/kafka-console-consumer.sh --topic smartphoneTopic --from-beginning --bootstrap-server localhost:9092

Step 4 : chạy hdfs, yarn

start-all.sh

>>> cluster: http://127.0.0.1:8088/ ⇒ chưa có gì



Step 5 : Trước khi chạy kiểm tra tra Postgresql

>>> command vao terminal của Postgres:

sudo -i -u postgres

psql

>>> Kiểm tra xem có database nào

\l

>>> Ở đây đã có sẵn database “Big-Data-Project” cho dự án này

\c Big-Data-Project

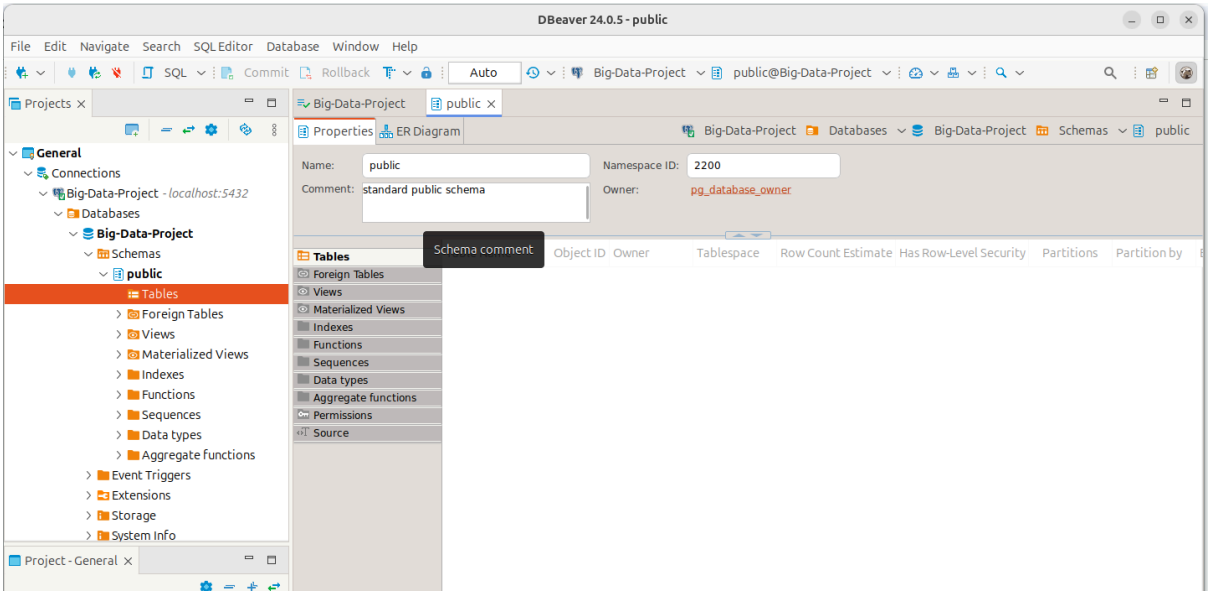
>>> Hiển thị danh sách các bảng trong cơ sở dữ liệu

\dt

>>> không có bảng nào

```
Big-Data-Project=# \dt
Did not find any relations.
Big-Data-Project=#
```

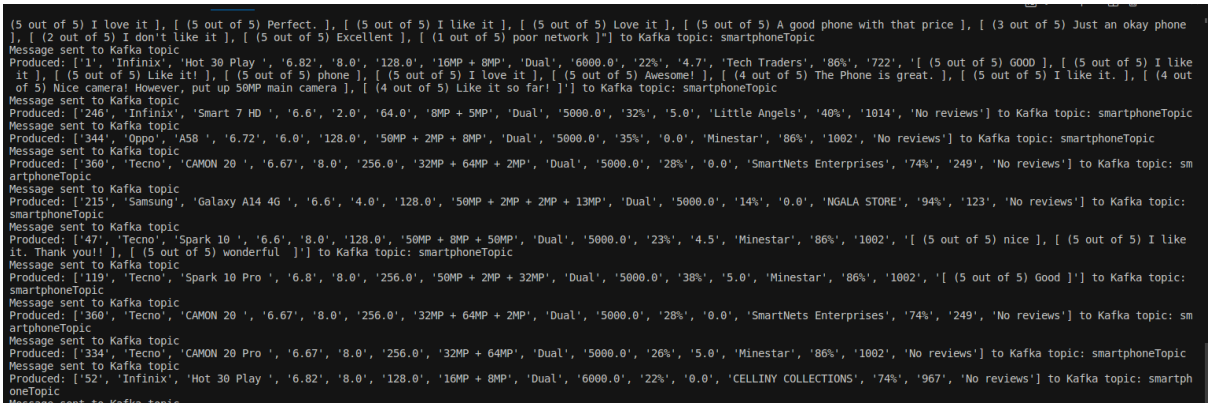
>>> Hoặc có thể kiểm tra trên công cụ DBMS - DBeaver-ce liên kết với database “Big-Data-Project”



>>> Đang không có bảng nào

Step 6: Bắt đầu chạy code

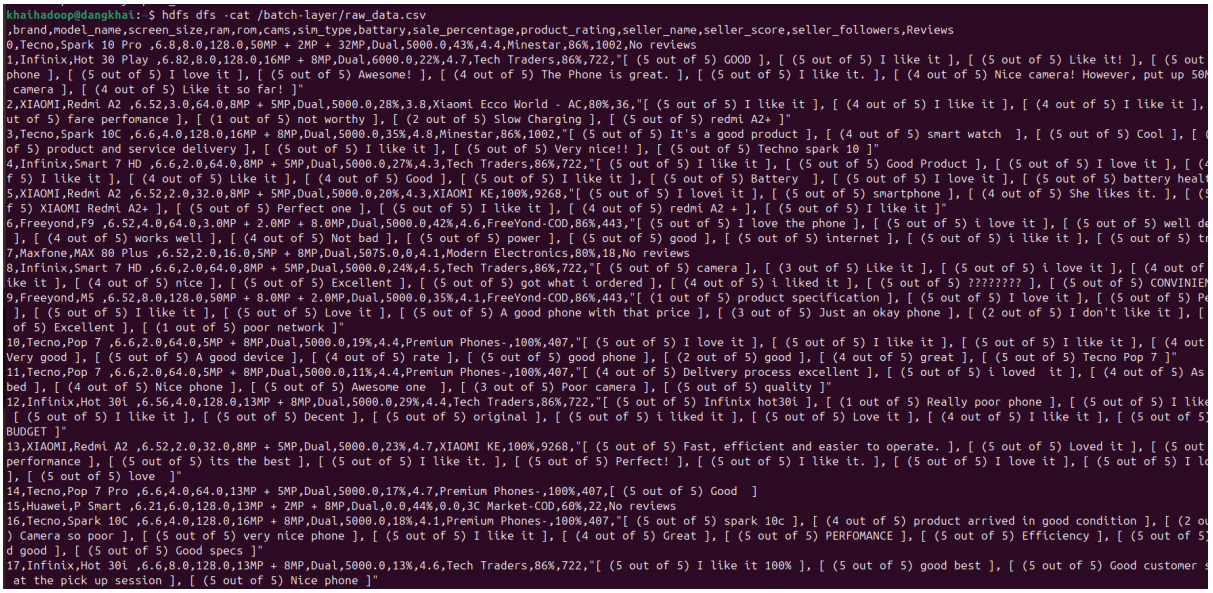
>>> chạy file batch_pipline.py ⇒ chuyển data lên kafka, lưu vào hdfs



>>> kiểm tra file /batch-layer/raw_data.csv trên hdfs

>>> new terminal

hdfs dfs -cat /batch-layer/raw_data.csv



>>> chạy file syc_batch_pipline.py để lấy data lưu ở hdfs đi qua spark transformation (đi qua model xgboost) rồi cuối cùng đẩy lên postgres:

```
/home/khaihadoop/.local/lib/python3.12/site-packages/xgboost/core.py:160: UserWarning: [22:26:30] WARNING: /workspace/
 pickle in Python, RDS in R) or
configuration generated by an older version of XGBoost, please export the model by calling
`Booster.save_model` from that version first, then load it back in current version. See:

    https://xgboost.readthedocs.io/en/stable/tutorials/saving_model.html

for more details about differences between saving model and serializing.

  warnings.warn(smsg, UserWarning)
Data transformed successfully
Data stored in PostgreSQL
begin
Data transformed successfully
Data stored in PostgreSQL
begin
Data transformed successfully
Data stored in PostgreSQL
begin
Data transformed successfully
Data stored in PostgreSQL
begin
Data transformed successfully
Data stored in PostgreSQL
begin
Data transformed successfully
Data stored in PostgreSQL
begin
Data transformed successfully
```

>>> Kiem tra web Spark

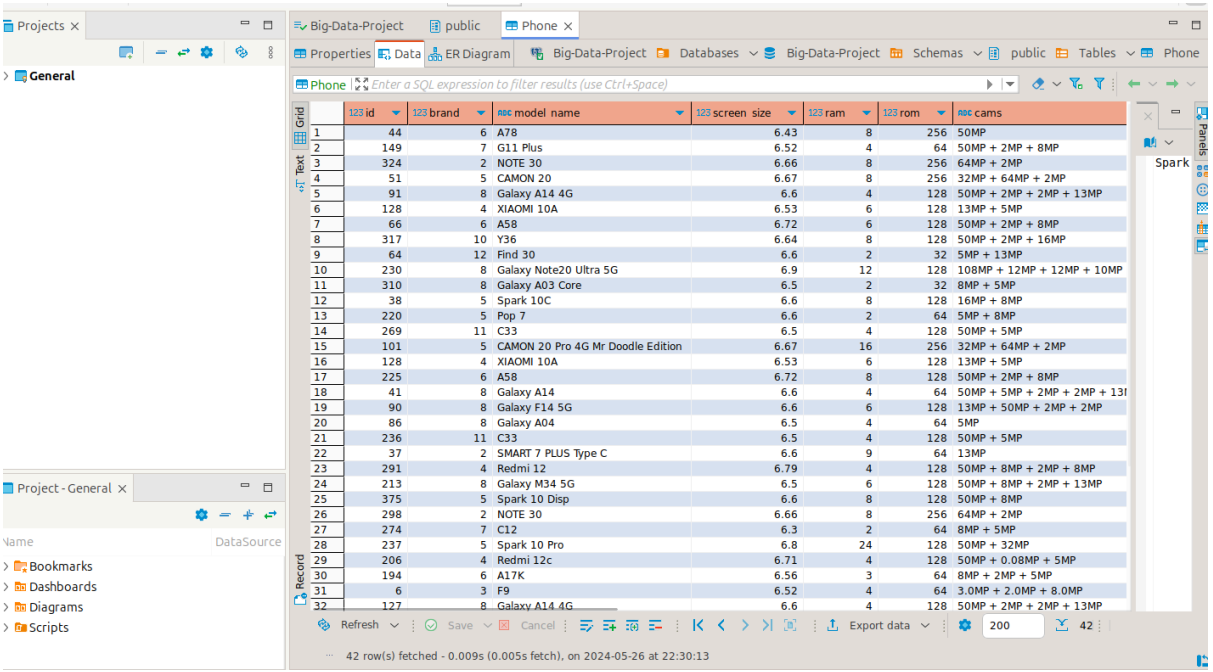
Completed Jobs (14)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
13	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8	2024/05/27 21:13:14	0.6 s	1/1	16/16
12	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45	2024/05/27 21:13:13	0.6 s	1/1	16/16
11	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8	2024/05/27 21:13:12	0.6 s	1/1	16/16
10	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45	2024/05/27 21:13:11	0.6 s	1/1	16/16
9	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8	2024/05/27 21:13:10	0.8 s	1/1	16/16
8	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45	2024/05/27 21:13:09	0.8 s	1/1	16/16
7	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8	2024/05/27 21:13:08	0.6 s	1/1	16/16
6	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45	2024/05/27 21:13:07	0.6 s	1/1	16/16
5	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8	2024/05/27 21:13:06	0.6 s	1/1	16/16
4	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45	2024/05/27 21:13:05	0.7 s	1/1	16/16
3	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8	2024/05/27 21:13:04	0.6 s	1/1	16/16
2	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45	2024/05/27 21:13:02	0.8 s	1/1	16/16
1	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/save_data_postgresql.py:8	2024/05/27 21:13:00	1 s	1/1	16/16
0	toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45 toPandas at /home/khaihadoop/Workspace/BigDataProject/Big-Data-Project/Main/Lambda/Batch_layer/spark_tranformation.py:45	2024/05/27 21:12:53	6 s	1/1	16/16

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

>>> data đã có trên postgresql



	id	brand	model_name	screen_size	ram	rom	cams
1	44	6	A78	6.43	8	256	50MP
2	149	7	G11 Plus	6.52	4	64	5MP + 2MP + 8MP
3	324	2	NOTE 30	6.66	8	256	64MP + 2MP
4	51	5	CAMON 20	6.67	8	256	32MP + 64MP + 2MP
5	91	8	Galaxy A14 4G	6.6	4	128	50MP + 2MP + 2MP + 13MP
6	128	4	XIAOMI 10A	6.53	6	128	13MP + 5MP
7	66	6	A58	6.72	6	128	50MP + 2MP + 8MP
8	317	10	Y36	6.64	8	128	50MP + 2MP + 16MP
9	64	12	Find 30	6.6	2	32	5MP + 13MP
10	230	8	Galaxy Note20 Ultra 5G	6.9	12	128	108MP + 12MP + 12MP + 10MP
11	310	8	Galaxy A03 Core	6.5	2	32	8MP + 5MP
12	38	5	Spark 10C	6.6	8	128	16MP + 8MP
13	220	5	Pop 7	6.6	2	64	5MP + 8MP
14	269	11	C33	6.5	4	128	50MP + 5MP
15	101	5	CAMON 20 Pro 4G Mr Doodle Edition	6.67	16	256	32MP + 64MP + 2MP
16	128	4	XIAOMI 10A	6.53	6	128	13MP + 5MP
17	225	6	A58	6.72	8	128	50MP + 2MP + 8MP
18	41	8	Galaxy A14	6.6	4	64	50MP + 5MP + 2MP + 2MP + 13I
19	90	8	Galaxy F14 5G	6.6	6	128	13MP + 50MP + 2MP + 2MP
20	86	8	Galaxy A04	6.5	4	64	5MP
21	236	11	C33	6.5	4	128	50MP + 5MP
22	37	2	SMART 7 PLUS Type C	6.6	9	64	13MP
23	291	4	Redmi 12	6.79	4	128	50MP + 8MP + 2MP + 8MP
24	213	8	Galaxy M34 5G	6.5	6	128	50MP + 8MP + 2MP + 13MP
25	375	5	Spark 10 Disp	6.6	8	128	50MP + 8MP
26	298	2	NOTE 30	6.66	8	256	64MP + 2MP
27	274	7	C12	6.3	2	64	8MP + 5MP
28	237	5	Spark 10 Pro	6.8	24	128	50MP + 32MP
29	206	4	Redmi 12c	6.71	4	128	50MP + 0.08MP + 5MP
30	194	6	A17K	6.56	3	64	8MP + 2MP + 5MP
31	6	3	F9	6.52	4	64	3.0MP + 2.0MP + 8.0MP
32	127	8	Galaxy A14 4G	6.6	4	128	50MP + 2MP + 2MP + 13MP

>>> đã xuất hiện cột price ⇒ data đã được xử lý, đi qua xgboost

Column Name	#	Data type	Identity
id	1	int8	
brand	2	int4	
model_name	3	text	
screen_size	4	float4	
ram	5	float4	
rom	6	float4	
cams	7	text	
sim_type	8	int4	
battary	9	float4	
sale_percenta	10	text	
product_ratir	11	float8	
seller_name	12	text	
seller_score	13	text	
seller_followe	14	int8	
Reviews	15	text	
price	16	text	