Windows:

A screenshot of a computer program

Description automatically generated

# Read the JSON file

iot\_devices = spark.read.json("iot\_devices.json")

# Create a temporary view

iot\_devices.createOrReplaceTempView("iot\_devices\_view")

A screenshot of a computer screen

Description automatically generated

spark.sql("""SELECT (COUNT(CASE WHEN lcd = 'red' THEN 1 END) \* 100.0 / COUNT(\*)) AS red\_percentage FROM iot\_devices\_view""").show()

red\_percentage= spark.sql("""SELECT (COUNT(CASE WHEN lcd = 'red' THEN 1 END) \* 100.0 / COUNT(\*)) AS red\_percentage FROM iot\_devices\_view""")

Text file didn’t support decimal so convert to string:

from pyspark.sql.functions import col

>>> red\_percentage\_str = red\_percentage.withColumn("red\_percentage", col("red\_percentage").cast("string"))

>>> red\_percentage\_str.coalesce(1).write \

.mode("overwrite") \

.format("text") \

.option("header", "false") \

.save("file:/output/red\_percentage\_1.txt")

A screenshot of a computer

Description automatically generated

spark.sql("""SELECT cn AS country, COUNT(\*) AS red\_device\_count FROM iot\_devices\_view WHERE lcd = 'red' GROUP BY cn ORDER BY red\_device\_count DESC""").show()

A screenshot of a computer

Description automatically generated

result = spark.sql("""SELECT cn AS country, COUNT(\*) AS red\_device\_count FROM iot\_devices\_view WHERE lcd = 'red' GROUP BY cn ORDER BY red\_device\_count DESC""")

# Collect the result to the driver

result\_data = result.collect()

# Open a text file for writing

with open("country\_most\_red-devices.txt", "w") as f:

for row in result\_data:

f.write(f"{row['country']}: {row['red\_device\_count']}\n")

A screenshot of a computer program

Description automatically generated

A black screen with a black border

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

spark.sql("""CREATE TABLE red\_devices\_by\_country AS SELECT cn AS country, COUNT(\*) AS red\_device\_count FROM iot\_devices\_view WHERE lcd = 'red' GROUP BY cn ORDER BY red\_device\_count DESC""")

A screenshot of a computer

Description automatically generated

Should use docker to solve

DROP TABLE red\_devices\_by\_country;

Docker Steps:

1.Open docker app

In the cmd as admin:

2.docker start db43bc55d3fe

3.docker exec -it db43bc55d3fe /bin/bash

(or)

2.docker start hadoop-cluster

3.docker exec -it hadoop-cluster /bin/bash

4.Installed python, pyspark

apt-get update

apt-get install -y python3 python3-pip

pip3 install pyspark

5.copy the input file to docker in cmd:  
docker cp iot\_devices.json hadoop-cluster:/input/

Pyspark

iot\_devices = spark.read.json("iot\_devices.json")

iot\_devices.createOrReplaceTempView("iot\_devices\_view")

spark.sql("""CREATE TABLE red\_devices\_by\_country AS SELECT cn AS country, COUNT(\*) AS red\_device\_count FROM iot\_devices\_view WHERE lcd = 'red' GROUP BY cn ORDER BY red\_device\_count DESC""")

Drop table:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated