

Họ và tên: Nguyễn Gia Huy

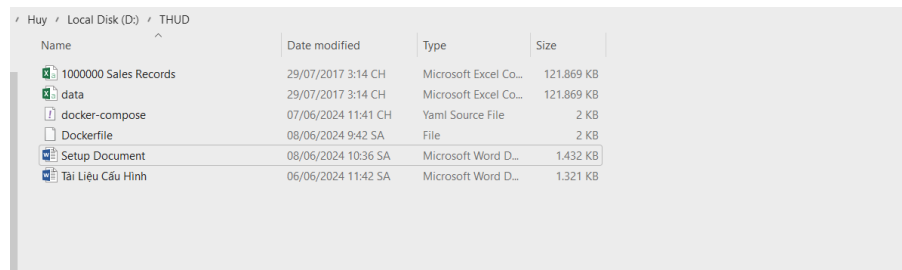
Mã Sinh Viên: 21002207

Lớp: K66 Kỹ Thuật Điện Tử Tin Học

CẤU HÌNH VÀ CHẠY SPARKSQL TRUY VẤN CƠ SỞ DỮ LIỆU

1. Docker Setup

- Thực hiện tải xuống Docker Desktop cho hệ máy Window
- Tạo một thư mục có tên là THUD
- Tạo một File với tên Dockerfile không có hậu tố
- Sử dụng trình soạn thảo Notepad thêm vào File đoạn code như ảnh 1
- Ngoài ra, tôi còn bổ sung thêm 1 tệp YAML giúp định nghĩa và quản lý các ứng dụng đa container (multi-container) một cách dễ dàng hơn
- Tạo một images bằng lệnh: "**docker build -t my-task.**"
- Sau đó build images **my-task** rồi chạy container bằng lệnh: "**docker run -it my-task bash**"



Thư mục lưu trữ

```
Dockerfile - Notepad
File Edit Format View Help
FROM docker.io/bitnami/spark:3.3.2
USER root

# Install prerequisites
RUN apt-get update && apt-get install -y curl


# Create the missing directory
RUN mkdir -p /var/lib/apt/lists/partial

# Update package list and install SQLite3
RUN apt-get update && apt-get install -y sqlite3

RUN curl -O https://repo1.maven.org/maven2/software/amazon/awssdk/s3/2.18.41/s3-2.18.41.jar \
&& curl -O https://repo1.maven.org/maven2/com/amazonaws/aws-java-sdk/1.12.367/aws-java-sdk-1.12.367.jar \
&& curl -O https://repo1.maven.org/maven2/com/amazonaws/aws-java-sdk-bundle/1.11.1026/aws-java-sdk-bundle-1.11.1026.jar \
&& curl -O https://repo1.maven.org/maven2/io/delta/delta-core_2.12/2.3.0/delta-core_2.12-2.3.0.jar \
&& curl -O https://repo1.maven.org/maven2/mysql/mysql-connector-java/8.0.19/mysql-connector-java-8.0.19.jar \
&& curl -O https://repo1.maven.org/maven2/org/apache/hadoop/hadoop-aws/3.3.2/hadoop-aws-3.3.2.jar \
&& mv s3-2.18.41.jar /opt/bitnami/spark/jars \
&& mv aws-java-sdk-1.12.367.jar /opt/bitnami/spark/jars \
&& mv aws-java-sdk-bundle-1.11.1026.jar /opt/bitnami/spark/jars \
&& mv delta-core_2.12-2.3.0.jar /opt/bitnami/spark/jars \
&& mv delta-storage-2.3.0.jar /opt/bitnami/spark/jars \
&& mv mysql-connector-java-8.0.19.jar /opt/bitnami/spark/jars \
&& mv hadoop-aws-3.3.2.jar /opt/bitnami/spark/jars
```

Cấu hình Dockerfile

 docker-compose.yaml X

D: > THUD >  docker-compose.yaml

```
1  version: "3.9"
2
3  services:
4    spark-master:
5      build:
6        context: ./docker_image/spark
7        dockerfile: ./Dockerfile
8        container_name: "spark-master"
9      ports:
10       - "7077:7077" # Spark master port
11       - "8081:8080" # Spark master web UI port
12      expose:
13       - "7077"
14      environment:
15       - SPARK_MODE=master
16       - SPARK_RPC_AUTHENTICATION_ENABLED=no
17       - SPARK_RPC_ENCRYPTION_ENABLED=no
18       - SPARK_LOCAL_STORAGE_ENCRYPTION_ENABLED=no
19       - SPARK_SSL_ENABLED=no
20       - SPARK_USER=spark
21      volumes:
22       - ./docker_image/spark/conf/spark-defaults.conf:/opt/bitnami/spark/conf/spark-defaults.conf
23       - ./docker_image/spark/conf/log4j.properties:/opt/bitnami/spark/conf/log4j.properties
24       - ./data:/opt/spark
25      networks:
26       - data_network
27
28    spark-worker-1:
29      image: docker.io/bitnami/spark:3.3.2
30      container_name: "spark-worker-1"
31      env_file:
32       - .env
33      depends_on:
34       - spark-master
35      networks:
36       - data_network
```

Docker-Compose

Start a build

PS D:\THUD>

>>

PS D:\THUD> docker build -t my-task .

[+] Building 7.7s (9/9) FINISHED

docker:default

=> [internal] load build definition from Dockerfile

0.1s

=> => transferring dockerfile: 1.53kB

0.1s

=> [internal] load metadata for docker.io/bitnami/spark:3.3.2

7.0s

=> [internal] load .dockerignore

0.0s

=> => transferring context: 2B

0.0s

=> [1/5] FROM docker.io/bitnami/spark:3.3.2@sha256:8892966efaa1896e64c14c38a807abda3f3d202bd7c0b02c0759cc5557a07aad

0.0s

=> CACHED [2/5] RUN apt-get update && apt-get install -y curl

0.0s

=> CACHED [3/5] RUN mkdir -p /var/lib/apt/lists/partial

0.0s

=> CACHED [4/5] RUN apt-get update && apt-get install -y sqlite3

0.0s

=> CACHED [5/5] RUN curl -O https://repo1.maven.org/maven2/software/amazon/awssdk/s3/2.18.41/s3-2.18.41.jar && curl -O https://repo1.maven.org/maven2/com/amazonaws

0.0s

=> exporting to image

0.1s

=> => exporting layers

0.0s

=> => writing image sha256:eefb7ea95282695009b46e434dbbe3458ffbe792ae3b16f3baeedc1d884c0742

0.0s

=> => naming to docker.io/library/my-task

0.0s

What's Next?

View a summary of image vulnerabilities and recommendations -> docker scout quickview

Build images

```
PS D:\THUD> docker run -it my-task bash
spark 04:02:31.65
spark 04:02:31.65 Welcome to the Bitnami spark container
spark 04:02:31.66 Subscribe to project updates by watching https://github.com/bitnami/containers
spark 04:02:31.66 Submit issues and feature requests at https://github.com/bitnami/containers/issues
spark 04:02:31.67

root@0682e78009b1:/opt/bitnami/spark#
```

Run container

2. Spark Installation

Trong Dockerfile ở trên tôi đã khởi tạo các câu lệnh cho việc cài đặt các công cụ cần thiết cho việc sử dụng và chạy Spark trên container. Ngoài ra tôi cũng bổ sung thêm 1 số công cụ và thư viện để dễ dàng hơn trong quá trình làm việc với Docker như sau:

- SQLite3: Một hệ quản trị cơ sở dữ liệu nhúng (embedded database) nhẹ, không cần server, thường được sử dụng cho các ứng dụng nhỏ hoặc phát triển.
- AWS SDK for Java (s3-2.18.41.jar): Bộ công cụ phát triển phần mềm của Amazon cho phép Spark tương tác với các dịch vụ lưu trữ Amazon S3.
- AWS Java SDK (aws-java-sdk-1.12.367.jar): Bộ công cụ toàn diện hơn của Amazon cho Java, bao gồm nhiều dịch vụ AWS khác ngoài S3.
- Delta Lake (delta-core_2.12-2.3.0.jar & delta-storage-2.3.0.jar): Một framework mã nguồn mở giúp xây dựng hồ dữ liệu (data lake) trên nền tảng Apache Spark, cung cấp các tính năng ACID và khả năng xử lý dữ liệu theo thời gian thực (real-time).
- MySQL Connector/J (mysql-connector-java-8.0.19.jar): Trình điều khiển JDBC để kết nối Spark với cơ sở dữ liệu MySQL.
- Hadoop AWS (hadoop-aws-3.3.2.jar): Thư viện Hadoop cung cấp tích hợp giữa Hadoop và các dịch vụ AWS, cho phép Spark đọc và ghi dữ liệu từ các dịch vụ lưu trữ AWS như S3.

Sau đó, thực hiện **build** lại images với tên my-task và chạy container và cuối cùng, nhập lệnh "**spark-shell**" để xác minh Spark đã chạy

```
root@0682e78009b1:/opt/bitnami/spark#  
root@0682e78009b1:/opt/bitnami/spark# spark-shell  
Setting default log level to "WARN".  
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).  
24/06/08 04:13:11 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
Spark context Web UI available at http://0682e78009b1:4040  
Spark context available as 'sc' (master = local[*], app id = local-1717819993334).  
Spark session available as 'spark'.  
Welcome to
```

```
      ____  
    /   _ \   /--_       __\   /__  
   /\     -   V     ___/\___/'/_/  
  /_____\_.-/\/\_/___/\___\'/_/  
          |_|              version 3.3.2  
        |_|
```

```
Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 17.0.8)  
Type in expressions to have them evaluated.  
Type :help for more information.
```

Xác minh Spark đã chạy

3. Database Setup

Trong Dockerfile, chúng ta đã có câu lệnh thực hiện việc update và install Sqlite để đảm bảo việc Sqlite được cài đặt khi chúng ta thực hiện tạo Image và chạy Container

Tiếp theo, chúng ta thực hiện việc copy 1 file có tên 1000000 Sales Records.csv vào thư mục 'tmp' với tên 'data.csv'

Truy cập vào terminal của container bằng câu lệnh `docker exec -it 9fe4620bf22c /bin/bash`, sau đó tạo và chạy cơ sở dữ liệu sqlite 3

```
PS D:\THUD> docker cp "D:\THUD\data.csv" 9fe4620bf22c:/tmp/data.csv
>>
Successfully copied 125MB to 9fe4620bf22c:/tmp/data.csv
```

```
root@9fe4620bf22c:/opt/bitnami/spark# sqlite3 database.db
SQLite version 3.34.1 2021-01-20 14:10:07
Enter ".help" for usage hints.
```

Tạo file data.csv và cơ sở dữ liệu sqlite 3

Sau khi thực hiện bước trên, giao diện sẽ hiển thị tương tác với database, lúc này ta sẽ thực hiện 2 lệnh:

- `.mode csv`
- `.import data.csv my_table`

Quá trình này import file data.csv gồm 1 triệu dòng vào my table

```
sqlite>
sqlite> .mode csv
sqlite> .import /tmp/data.csv my_table
```

-Thử truy vấn hiển thị 10 dòng dữ liệu đầu

```
sqlite> .mode csv
sqlite> .import data.csv my_table

sqlite>
sqlite> sqlite> SELECT COUNT(*) FROM my_table;
Error: near "sqlite": syntax error
sqlite> SELECT COUNT(*) FROM my_table;
1000000
sqlite> SELECT * FROM my_table LIMIT 10;
"Sub-Saharan Africa","South Africa",Fruits,Offline,M,7/27/2012,443368995,7/28/2012,1593,9.33,6.92,14862.69,11023.56,3839.13
"Middle East and North Africa",Morocco,Clothes,Online,M,9/14/2013,667593514,10/19/2013,4611,109.28,35.84,503890.00,165258.24,338631.84
"Australia and Oceania","Papua New Guinea",Meat,Offline,M,5/15/2015,940995585,6/4/2015,360,421.89,364.69,151880.40,131288.40,20592.00
"Sub-Saharan Africa",Djibouti,Clothes,Offline,H,5/17/2017,880811536,7/2/2017,562,109.28,35.84,61415.36,20142.08,41273.28
Europe,Slovakia,Beverages,Offline,L,10/26/2016,174590194,12/4/2016,3973,47.45,31.79,188518.85,126301.67,62217.18
Asia,"Sri Lanka",Fruits,Online,L,11/7/2011,830192887,12/18/2011,1379,9.33,6.92,12866.07,9542.68,3323.39
"Sub-Saharan Africa","Seychelles ",Beverages,Online,M,1/18/2013,425793445,2/16/2013,597,47.45,31.79,28327.65,18978.63,9349.02
"Sub-Saharan Africa",Tanzania,Beverages,Online,L,11/30/2016,659878194,1/16/2017,1476,47.45,31.79,70036.20,46922.04,23114.16
"Sub-Saharan Africa",Ghana,"Office Supplies",Online,L,3/23/2017,601245963,4/15/2017,896,651.21,524.96,583484.16,470364.16,113120.00
"Sub-Saharan Africa",Tanzania,Cosmetics,Offline,L,5/23/2016,739008080,5/24/2016,7768,437.20,263.33,3396169.60,2045547.44,1350622.16
sqlite>
```

4. Dependencies

Chúng ta sẽ thêm 2 thành phần cần thiết là PySpark và JDBC cho sqlite để giúp kết nối và thực hiện truy vấn từ Spark đến cơ sở dữ liệu. Để thực hiện được tôi đã sử dụng 3 câu lệnh sau:

```
pip install --upgrade setuptools
```

```
pip install pyspark
```

```
wget https://repo1.maven.org/maven2/org/xerial/sqlite-jdbc/3.34.0/sqlite-jdbc-3.34.0.jar
```

```
PS D:\THUD> pip install pyspark
Requirement already satisfied: pyspark in c:\users\pro 2004\appdata\local\programs\python\python311\lib\site-packages (3.5.1)
Requirement already satisfied: py4j==0.10.9.7 in c:\users\pro 2004\appdata\local\programs\python\python311\lib\site-packages (from pyspark) (0.10.9.7)

[notice] A new release of pip is available: 23.3.1 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
PS D:\THUD> pip install pyspark
Requirement already satisfied: pyspark in c:\users\pro 2004\appdata\local\programs\python\python311\lib\site-packages (3.5.1)
Requirement already satisfied: py4j==0.10.9.7 in c:\users\pro 2004\appdata\local\programs\python\python311\lib\site-packages (from pyspark) (0.10.9.7)

[notice] To update, run: python.exe -m pip install --upgrade pip
PS D:\THUD> python -c "import pyspark; print(pyspark.__version__)"
>>
3.5.1
PS D:\THUD> wget https://repo1.maven.org/maven2/org/xerial/sqlite-jdbc/3.34.0/sqlite-jdbc-3.34.0.jar

StatusCode      : 200
StatusDescription : OK
Content         : {80, 75, 3, 4...}
RawContent      : HTTP/1.1 200 OK
                  Connection: keep-alive
                  X-Checksum-MD5: 743bacfa02e66cad1027e80b065c45ad
                  X-Checksum-SHA1: fd29bb0124e3f79c80b2753162a6a3873c240bcf
                  Age: 2656032
                  X-Served-By: cache-iad-kiad7000141-I...
Headers         : {[Connection, keep-alive], [X-Checksum-MD5, 743bacfa02e66cad1027e80b065c45ad], [X-Checksum-SHA1, fd29bb0124e3f79c80b2753162a6a3873c240bcf], [Age, 2656032].
                  ...}
RawContentLength : 7296329
```

5. Configuration

Để Spark có thể kết nối đến cơ sở dữ liệu, ta cần thực hiện di chuyển JDBC vào thư mục jars của Spark, đây là thư mục giúp Spark tự động phát hiện và chạy các thư viện phụ trợ nếu cần thiếu. Ta có thể nhập lệnh:

```
mv sqlite-jdbc-3.35.0.jar /opt/spark/jars
```

hoặc trực tiếp tải về thư mục jars thông qua câu lệnh:

```
curl -L -o /opt/bitnami/spark/jars/sqlite-jdbc-3.34.0.jar
```

<https://repo1.maven.org/maven2/org/xerial/sqlite-jdbc/3.34.0/sqlite-jdbc-3.34.0.jar>

Sau khi di chuyển hoàn tất, ta có thể thử truy cập và tương tác với cơ sở dữ liệu bằng PySpark

```
root@9fe4620bf22c:/opt/bitnami/spark# spark-shell --master local[*]
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/06/08 09:20:05 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Spark context Web UI available at http://9fe4620bf22c:4040
Spark context available as 'sc' (master = local[*], app id = local-1717838409031).
Spark session available as 'spark'.
Welcome to

    ____
   /__ \__  _--_____/ __\
  _ \| |/_/ - \_|'_ \_/_/
 /___/\ .--/\_,_/_/ /_/\_\
      |_/

        version 3.3.2


Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 17.0.8)
Type in expressions to have them evaluated.
Type :help for more information.
```

Sau đó nhập lệnh:

val spark =

```
SparkSession.builder().appName("SQLiteApp").getOrCreate()
```

```
val df = spark.read.format("jdbc").option("url",  
"jdbc:sqlite:/opt/bitnami/spark/database.db").option("dbtable",  
"my_table").option("driver", "org.sqlite.JDBC").load()
```

df.show()

Dưới đây là thử nghiệm kết nối đến cơ sở dữ liệu database.db, bảng my_table, hiển thị 20 dòng đầu tiên


```
scala> df.show()
```

only showing top 20 rows

Ở mục 6 này, tôi sẽ thực hiện các bước sau để thực hiện các truy vấn SQL sử dụng SparkSQL trên dữ liệu , bao gồm các thao tác CRUD (Create, Read, Update, Delete) và đánh giá hiệu suất của mệnh đề WHERE như sau:

- ```
scala> df.createOrReplaceTempView("sales")
```

- ```
scala> df.show(10)
```

only showing top 10 rows

- ```
scala> spark.sql("SELECT * FROM sales WHERE Country = 'Vietnam' AND \"Order Priority\" = 'H'").show()
```

| Region | Country | Item Type | Sales Channel | Order Priority | Order Date | Order ID | Ship Date | Units Sold | Unit Price | Unit Cost | Total Revenue | Total Cost | Total Profit |
|--------|---------|-----------|---------------|----------------|------------|----------|-----------|------------|------------|-----------|---------------|------------|--------------|
|--------|---------|-----------|---------------|----------------|------------|----------|-----------|------------|------------|-----------|---------------|------------|--------------|

## ➤ Kiểm tra lại kết quả sau khi lọc điều kiện WHERE

```
scala> spark.sql("SELECT COUNT(*) FROM sales WHERE Country = 'Vietnam' AND \"Order Priority\" = 'H').show()
+-----+
|count(1)|
+-----+
| 0|
+-----+
```

## ➤ Sắp xếp (Order by): Hiện thị 10 hàng có lợi nhuận cao nhất

```
scala> spark.sql("SELECT * FROM sales ORDER BY `Total Profit` DESC LIMIT 10").show()
```

| Region               | Country             | Item Type  | Sales Channel | Order Priority | Order Date | Order ID  | Ship Date | Units Sold | Unit Price | Unit Cost | Total Revenue | Total Cost | Total Profit |
|----------------------|---------------------|------------|---------------|----------------|------------|-----------|-----------|------------|------------|-----------|---------------|------------|--------------|
| Central America a... | Trinidad and Tobago | Vegetables | Online        | H              | 3/21/2010  | 535656726 | 4/1/2010  | 1584       | 154.06     | 90.93     | 244031.04     | 144033.12  | 99997.92     |
| Central America a... | Trinidad and Tobago | Vegetables | Online        | H              | 3/21/2010  | 535656726 | 4/1/2010  | 1584       | 154.06     | 90.93     | 244031.04     | 144033.12  | 99997.92     |
| Europe               | Poland              | Vegetables | Offline       | H              | 3/12/2010  | 658019959 | 4/28/2010 | 1584       | 154.06     | 90.93     | 244031.04     | 144033.12  | 99997.92     |
| Asia                 | Laos                | Cosmetics  | Offline       | H              | 3/5/2017   | 995090333 | 4/18/2017 | 5751       | 437.20     | 263.33    | 2514337.20    | 1514410.83 | 999926.37    |
| Sub-Saharan Africa   | Rwanda              | Cosmetics  | Online        | H              | 1/13/2017  | 829869735 | 1/21/2017 | 5751       | 437.20     | 263.33    | 2514337.20    | 1514410.83 | 999926.37    |
| Middle East and N... | Tunisia             | Cosmetics  | Online        | H              | 1/29/2017  | 585143268 | 3/7/2017  | 5751       | 437.20     | 263.33    | 2514337.20    | 1514410.83 | 999926.37    |
| Central America a... | Trinidad and Tobago | Cosmetics  | Offline       | H              | 2/16/2017  | 340416800 | 3/3/2017  | 5751       | 437.20     | 263.33    | 2514337.20    | 1514410.83 | 999926.37    |
| Asia                 | Singapore           | Cosmetics  | Online        | H              | 6/1/2017   | 672057998 | 7/9/2017  | 5751       | 437.20     | 263.33    | 2514337.20    | 1514410.83 | 999926.37    |
| Europe               | Poland              | Cosmetics  | Offline       | H              | 2/8/2017   | 462780034 | 2/8/2017  | 5751       | 437.20     | 263.33    | 2514337.20    | 1514410.83 | 999926.37    |

## ➤ Tính toán (Aggregate): Tính trung bình và tổng số sản phẩm bán được

```
scala> spark.sql("SELECT AVG(`Total Profit`) AS avg_profit, SUM(`Units Sold`) AS total_units_sold FROM sales").show()
+-----+-----+
|avg_profit|total_units_sold|
+-----+-----+
|392295.56164894794|4.998867302E9|
+-----+-----+
```

## ➤ Gom nhóm (Group by): Tính tổng lợi nhuận cho mỗi quốc gia

```
scala> spark.sql("SELECT Country, SUM(`Total Profit`) AS total_profit FROM sales GROUP BY Country").show()
+-----+-----+
|Country|total_profit|
+-----+-----+
Chad	2.1918871360499988E9
Russia	2.0804650325800033E9
Yemen	2.1180728529700022E9
Senegal	2.1525300459900036E9
Sweden	2.1418085992999995E9
Kiribati	2.1154151909000049E9
Eritrea	2.1127757483300009E9
Philippines	2.1169799002100031E9
Djibouti	2.1397777185099971E9
Tonga	2.07418477733E9
Singapore	2.1930081922700014E9
Malaysia	2.1437040150299988E9
Fiji	2.1279547660899978E9
Turkey	2.0681019684200046E9
Malawi	2.1430224149500017E9
Iraq	2.1339626060300026E9
Germany	2.0550807875399983E9
Comoros	2.1049547661700044E9
Cambodia	2.1028961437099986E9
Afghanistan	2.1336894335899992E9
+-----+-----+
only showing top 20 rows
```



- Thực hiện các thao tác CRUD khác:

- Tạo (Create): Giả sử tạo một bảng mới *top\_countries* từ dữ liệu hiện có

```
sqlite> CREATE TABLE top_countries AS
...> SELECT Country, SUM("Total Profit") AS total_profit
...> FROM my_table
...> GROUP BY Country
...> ORDER BY total_profit DESC
...> LIMIT 10;
```

Thông báo tạo thành công bảng mới có tên “top\_countries”

```
sqlite> SELECT * FROM top_countries;
Egypt|2203699711.79001
Singapore|2193008192.27
Chad|2191887136.05
New Zealand|2190794420.57
Serbia|2181042376.24
Panama|2179364731.62001
Maldives|2176744893.41
Australia|2174542209.71
San Marino|2174334249.71001
Mexico|2173364479.84001
```

*Hiện thị bảng*

- Xóa (Delete): Giả sử xóa bỏ những hàng có *total\_profit* < 5000

```
sqlite>
sqlite>
sqlite>
sqlite> SELECT COUNT(*) FROM my_table;
1000001
```

*My\_table trước khi bị xóa với hơn 1 triệu mẫu*

```
sqlite> SELECT COUNT(*) FROM my_table;
1000001
sqlite> DELETE FROM my_table WHERE "Total Profit" < 5000;
sqlite> SELECT COUNT(*) FROM my_table;
337062
```

*Sau khi thực hiện thành công lệnh xóa , bảng dữ liệu còn lại 337.062 mẫu*

- Cập nhật (Update): Giả sử cập nhật Order Priority thành 'H' (Cao) cho các đơn hàng có Total Profit > 5000 và Cập nhật Unit Price cho một số Item Type cụ thể ví dụ như Cosmetics:

```

sqlite> .mode csv
sqlite> .import /tmp/data.csv my_table
sqlite> UPDATE my_table SET `Unit Price` = `Unit Price` * 1.1 WHERE `Item Type` = 'Cosmetics';
sqlite> SELECT `Item Type`, `Unit Price` FROM my_table WHERE `Item Type` = 'Cosmetics' LIMIT 20;
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
Cosmetics,480.92
sqlite> SELECT * FROM my_table WHERE `Item Type` = 'Cosmetics' LIMIT 20;
"Sub-Saharan Africa",Tanzania,Cosmetics,Offline,L,5/23/2016,739008080,5/24/2016,7768,480.92,263.33,3396169.60,2045547.44,1350622.16
"Middle East and North Africa",Algeria,Cosmetics,Online,M,2/18/2011,761723172,2/24/2011,9669,480.92,263.33,4227286.80,2546137.77,1681149.03
"Sub-Saharan Africa",Ethiopia,Cosmetics,Online,M,7/7/2011,807785928,7/25/2011,662,480.92,263.33,289426.40,174324.46,115101.94
Europe,France,Cosmetics,Online,M,12/7/2015,324669444,1/18/2016,5758,480.92,263.33,2517397.60,1516254.14,1001413.46
Europe,"United Kingdom",Cosmetics,Online,L,5/1/2015,135178029,5/16/2015,1038,480.92,263.33,453813.60,273336.54,180477.06
"Sub-Saharan Africa",Uganda,Cosmetics,Online,M,2/28/2015,842238795,3/15/2015,6031,480.92,263.33,2636753.20,1588143.23,1048609.97
Europe,"Czech Republic",Cosmetics,Online,H,3/22/2014,726137769,4/26/2014,9157,480.92,263.33,4003440.40,2411312.81,1592127.59
"Middle East and North Africa",Oman,Cosmetics,Online,H,11/29/2010,358570849,12/28/2010,7937,480.92,263.33,3470056.40,2090050.21,1380006.19
"Sub-Saharan Africa",Togo,Cosmetics,Online,M,9/8/2015,563681733,9/9/2015,4806,480.92,263.33,2101183.20,1265563.98,835619.22
"North America",Canada,Cosmetics,Online,H,5/9/2011,368977391,6/2/2011,7464,480.92,263.33,3263260.80,1965495.12,1297765.68
"Middle East and North Africa",Tunisia,Cosmetics,Offline,H,7/1/2012,479969346,7/20/2012,2450,480.92,263.33,1071140.00,645158.50,425981.50
"Sub-Saharan Africa",Democratic Republic of the Congo,Cosmetics,Online,M,10/7/2012,584356629,10/25/2012,2967,480.92,263.33,1297172.40,781300.11,515872.29
"Sub-Saharan Africa",Democratic Republic of the Congo,Cosmetics,Offline,M,7/30/2013,641770064,8/25/2013,2878,480.92,263.33,1258261.60,757863.74,500397.86
"Central America and the Caribbean",El Salvador,Cosmetics,Offline,L,2/20/2016,219787776,3/5/2016,8942,480.92,263.33,3909442.40,2354696.86,1554745.54
Asia,Myanmar,Cosmetics,Online,C,1/21/2013,801426732,2/19/2013,8834,480.92,263.33,3862224.80,2326257.22,1535967.58
"Sub-Saharan Africa",Zambia,Cosmetics,Online,M,4/16/2011,849694049,5/29/2011,2206,480.92,263.33,964463.20,580905.98,383557.22
"North America",Canada,Cosmetics,Online,C,2/14/2012,140492665,2/18/2012,3757,480.92,263.33,1642560.40,989330.81,653229.59
Europe,Denmark,Cosmetics,Online,M,9/24/2012,113121688,9/26/2012,4076,480.92,263.33,1782027.20,1073333.08,708694.12
Europe,Netherlands,Cosmetics,Online,L,11/21/2013,200302493,11/25/2013,2814,480.92,263.33,1230280.80,741010.62,409270.18
"Sub-Saharan Africa",Kenya,Cosmetics,Online,H,2/27/2010,864372813,3/3/2010,5979,480.92,263.33,2614018.80,1574450.07,1039568.73
sqlite>

```

- Đánh giá hiệu năng của WHERE: So sánh thời gian thực thi  
Tạo tempview để truy vấn dữ liệu

```
scala> val newDF = spark.read.format("jdbc")
newDF: org.apache.spark.sql.DataFrameReader = org.apache.spark.sql.DataFrameReader@7b2f5b91

scala> .option("url", "jdbc:sqlite:/opt/bitnami/spark/database.db")
res8: org.apache.spark.sql.DataFrameReader = org.apache.spark.sql.DataFrameReader@7b2f5b91

scala> .option("dbtable", "my_table")
res9: org.apache.spark.sql.DataFrameReader = org.apache.spark.sql.DataFrameReader@7b2f5b91

scala> .option("driver", "org.sqlite.JDBC")
res10: org.apache.spark.sql.DataFrameReader = org.apache.spark.sql.DataFrameReader@7b2f5b91

scala> .load().createOrReplaceTempView("sales")
```

### ➤ Không có WHERE

```
scala> val startTime1 = System.currentTimeMillis()
startTime1: Long = 1718180635181

scala> spark.sql("SELECT * FROM sales").count()
res12: Long = 1000001

scala> val endTime1 = System.currentTimeMillis()
endTime1: Long = 1718180974744

scala> println(s"Thời gian thực thi không có WHERE: ${endTime1 - startTime1} ms")
Thời gian thực thi không có WHERE: 339563 ms
```

### ➤ Có WHERE

```
scala> val startTime2 = System.currentTimeMillis()
startTime2: Long = 1718181109894

scala> spark.sql("SELECT * FROM sales WHERE Country = 'Vietnam']").count()
res14: Long = 5367

scala> val endTime2 = System.currentTimeMillis()
endTime2: Long = 1718181125609

scala> println(s"Thời gian thực thi có WHERE: ${endTime2 - startTime2} ms")
Thời gian thực thi có WHERE: 15715 ms
```

#### • Truy vấn không có WHERE:

- Spark phải đọc và xử lý **toàn bộ dữ liệu** trong bảng sales.
- Thời gian thực thi sẽ **tăng lên** khi kích thước của bảng sales tăng.
- Không hiệu quả khi bạn chỉ quan tâm đến một phần nhỏ dữ liệu.

#### • Truy vấn có WHERE:

- Spark chỉ đọc và xử lý các dòng **thỏa mãn điều kiện** trong mệnh đề WHERE.

- Thời gian thực thi thường **nhANH hơn** so với truy vấn không có `WHERE`, đặc biệt khi điều kiện lọc giúp giảm đáng kể lượng dữ liệu cần xử lý.

**Kết luận:** Có thể thấy, `WHERE` là một công cụ quan trọng để tối ưu hóa hiệu suất truy vấn trong Spark SQL. Bằng cách sử dụng `WHERE`, ta có thể giảm đáng kể lượng dữ liệu cần xử lý, giúp truy vấn chạy nhanh hơn và tiết kiệm tài nguyên.