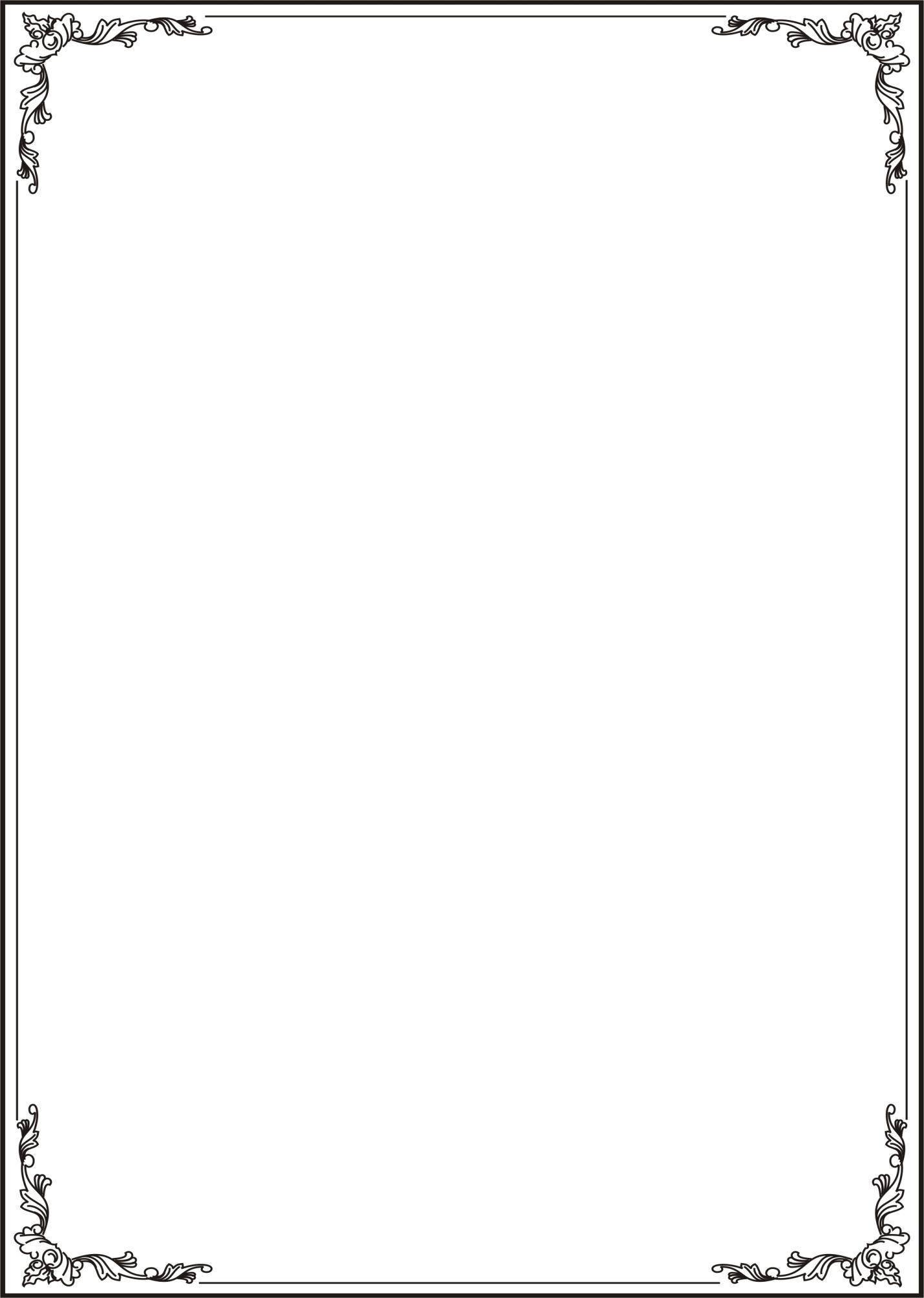
.

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**MÔN ĐIỆN TOÁN ĐÁM MÂY**



|  |  |
| --- | --- |
| 1. Đinh Tấn Phúc Huy | 20110129 |
| 1. Đỗ Trung Đức | 20110459 |
| 1. Nguyễn Văn Giang | 20110463 |

**Đề Tài:**

**TÌM HIỂU APACHE HIVE**

**VÀ VIẾT ỨNG DỤNG DEMO**

**GIẢNG VIÊN HƯỚNG DẪN: TS. HUỲNH XUÂN PHỤNG**

**TP HCM, tháng 12 năm 2022**

**KHÓA 2020 - 2024**

|  |  |
| --- | --- |
| **ĐH SƯ PHẠM KỸ THUẬT TP. HCM**  **KHOA CNTT**  \*\*\*\*\*\*\* | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh Phúc**  \*\*\*\*\*\*\* |

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Họ tên sv 1:**Nguyễn Văn Giang** Mã số sinh viên 1: **20110463**

Họ tên sv 2:**Đỗ Trung Đức** Mã số sinh viên 2: **20110459**

Họ tên sv 3:**Đinh Tấn Phúc Huy** Mã số sinh viên 3: **20110129**

Ngành: **Công Nghệ Thông TIN**

Đề tài: **Tìm hiểu Apache Hive và viết ứng dụng demo**

Giáo viên hướng dẫn: **TS. Huỳnh Xuân Phụng**

**LỜI NHẬN XÉT CỦA GIẢNG VIÊN**

1. Về nội dung đề tài & khối lượng thực hiện được trong đề tài:

2. Ưu điểm của đề tài:

3. Khuyết điểm cần sửa đổi trong đề tài:

5. Đánh giá xếp loại:

6. Điểm:

|  |  |  |
| --- | --- | --- |
| *Tp. HCM, ngày tháng năm 2022* | | |
|  | Giáo viên hướng dẫn  *(Ký & ghi rõ họ tên)* |

# **LỜI CẢM ƠN**

*Đầu tiên nhóm chúng em xin được phép gửi lời cảm ơn chân thành nhất và sâu sắc nhất đến với Khoa Đào Tạo Chất Lượng Cao – Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh đã tạo điều kiện tốt nhất cho nhóm chúng em được học tập, phát triển nền tảng kiến thức sâu sắc từ đó có thể được đề tài kết thúc môn học này.*

*Bên cạnh đó nhóm chúng em xin gửi đến thầy Huỳnh Xuân Phụng lời cảm ơn sâu sắc nhất trong suốt thời gian dài học tập cũng như thực hiện đề tài trong thời gian qua.*

*Với sự chỉ bảo nhiệt tình, giảng dạy tận tình và đầy đủ kiến thức của thầy Huỳnh Xuân Phụng, nhóm chúng em đã học tập và hiểu được những kiến thức cơ bản về Apache Hive – một dạng kho dữ liệu được sử dụng để quản lý và phân tích khối lượng dữ liệu lớn.*

*Tuy nhiên lượng kiến thức là vô tận và với khả năng hạn hẹp chúng em đã rất cố gắng để hoàn thành một cách tốt nhất đề tài này. Chính vì vậy việc xảy ra những thiếu sót là điều khó có thể tránh khỏi. Nhóm chúng em hi vọng nhận được sự góp ý, nhận xét và phê bình tận tình của quý thầy (cô) qua đó chúng em có thể rút ra được các bài học kinh nghiệm từ đó hoàn thiện và cải thiện nâng cấp lại đề tài của mình một cách tốt nhất có thể.*

*Nhóm chúng em xin chân thành cảm ơn quý thầy(cô)!*

***Sinh viên thực hiện đồ án***

**Nguyễn Văn Giang – 20110463**

**Đỗ Trung Đức - 20110459**

**Đinh Tấn Phúc Huy - 20110129**

**MỤC LỤC**

[LỜI CẢM ƠN 3](#_Toc120306958)

[PHẦN MỞ ĐẦU 5](#_Toc120306959)

[**1.** **Tính cấp thiết của đồ án** 5](#_Toc120306960)

[**2.** **Đối tượng nghiên cứu** 5](#_Toc120306961)

[**3.** **Phạm vi nghiên cứu** 5](#_Toc120306962)

[**4.** **Kết quả dự kiến đạt được** 5](#_Toc120306963)

[PHẦN NỘI DUNG 5](#_Toc120306964)

[**Chương 1 : CƠ SỞ LÝ THUYẾT VỀ APACHE HIVE** 5](#_Toc120306965)

[**1.** **Hive là gì ?** 5](#_Toc120306966)

[**2.** **Ra đời nhằm giải quyết những vấn đề gì ?** 6](#_Toc120306967)

[**3.** **Các tính năng của Apache Hive** 6](#_Toc120306968)

[**4.** **Kiến trúc của Apache Hive** 7](#_Toc120306969)

[**5.** **Quy trình làm việc của Apache Hive** 7](#_Toc120306970)

[**Chương 2 : CÀI ĐẶT, THIẾT KẾ VÀ XÂY DỰNG DATAWAREHOUSE** 9](#_Toc120306971)

[PHẦN KẾT LUẬN 22](#_Toc120306972)

[**1.** **Kết quả thực hiện và thực nghiệm được** 22](#_Toc120306973)

[**2.** **Ưu điểm** 22](#_Toc120306974)

[**3.** **Nhược điểm** 22](#_Toc120306975)

[**4.** **Hướng phát triển** 22](#_Toc120306976)

[TÀI LIỆU THAM KHẢO 23](#_Toc120306977)

# **PHẦN MỞ ĐẦU**

## **Tính cấp thiết của đồ án**

Hiện tại, dữ liệu lớn đã và đang dần phát triển rộng rãi trên thế giới, nhằm giải quyết khối lượng lớn dữ liệu này, Apache Software Foundation đã giới thiệu một khuôn khổ được gọi là Hadoop và bên trong nó là Apache Hive để giải quyết các thách thức về quản lý và xử lý dữ liệu lớn.

## **Đối tượng nghiên cứu**

Đối tượng nghiên cứu của đề tài này là Dữ liệu lớn. Đồng thời kèm theo đó là các công nghệ áp dụng để xây dựng một warehouse đơn giản, cụ thể như:

- Apache spoop: là một công cụ được thiết kế để hỗ trợ xuất và nhập dữ liệu hàng loạt vào HDFS từ các kho dữ liệu có cấu trúc như cơ sở dữ liệu quan hệ, kho dữ liệu doanh nghiệp và hệ thống NoSQL. Nó là một công cụ di chuyển dữ liệu dựa trên kiến ​​trúc trình kết nối hỗ trợ các plugin để cung cấp kết nối với các hệ thống bên ngoài mới.

- Hadoop: Apache framework mã nguồn mở cho phép phát triển các ứng dụng phân tán để lưu trữ và quản lý các tập dữ liệu lớn.

- Apache Hive: Công cụ cơ sở hạ tầng kho dữ liệu để xử lý dữ liệu có cấu trúc trong Hadoop.

- Amazon S3: là dịch vụ lưu trữ đối tượng được xây dựng để lưu trữ và truy xuất bất kỳ lượng dữ liệu nào từ bất cứ nơi nào. Đây là dịch vụ lưu trữ đơn giản có độ bền, độ sẵn có, hiệu suất, tính bảo mật dẫn đầu ngành và khả năng thay đổi quy mô gần như không giới hạn với chi phí cực kỳ thấp.

- Amazon EMR: là giải pháp dữ liệu lớn trên đám mây dẫn đầu ngành trong việc xử lý dữ liệu ở cấp độ petabyte, phân tích tương tác và công nghệ máy học bằng các khung nguồn mở

## **Phạm vi nghiên cứu**

Đề tài này chủ yếu tập trung vào các việc xử lý và phân tích các câu truy vấn đối với dữ liệu lớn khi sử dụng Hive.

## **Kết quả dự kiến đạt được**

- Cài đặt được Hadoop và Apache Hive

- Xây dựng được một data warehouse đơn giản bằng các câu truy vấn.

- Phân tích, thống kê các dữ liệu từ các câu truy vấn tùy vào mục đích của người dùng.

# **PHẦN NỘI DUNG**

## **Chương 1 : CƠ SỞ LÝ THUYẾT VỀ APACHE HIVE**

### **Hive là gì ?**

Apache Hive là một dự án phần mềm kho dữ liệu được xây dựng dựa trên Apache Hadoop để cung cấp truy vấn và phân tích dữ liệu. Hive cung cấp giao diện giống như SQL để truy vấn dữ liệu được lưu trữ trong các cơ sở dữ liệu và hệ thống tệp khác nhau tích hợp với Hadoop. Các truy vấn SQL truyền thống phải được triển khai trong MapReduce Java API để thực thi các ứng dụng SQL và truy vấn trên dữ liệu phân tán.



Hive không phải là :

* Một Cơ sở dữ liệu quan hệ
* Một thiết kế để xử lý các giao dịch trực tuyến (OnLine Transaction Processing - OLTP)
* Một ngôn ngữ cho các truy vấn thời gian thực và cập nhật cấp hàng

### **Ra đời nhằm giải quyết những vấn đề gì ?**

Hive là một dự án phần mềm cung cấp các truy vấn và phân tích dữ liệu. Nó tạo điều kiện thuận lợi cho việc đọc, viết và xử lý các tập dữ liệu rộng được lưu trữ trong bộ lưu trữ phân tán và được truy vấn bằng các cú pháp ngôn ngữ truy vấn cấu trúc (SQL).

Hive không được xây dựng cho khối lượng công việc xử lý giao dịch trực tuyến (OLTP). Nó thường được sử dụng cho các tác vụ lưu trữ dữ liệu như đóng gói dữ liệu, truy vấn đặc biệt và phân tích các tập dữ liệu khổng lồ.

Nó được thiết kế để nâng cao khả năng mở rộng, khả năng mở rộng, hiệu suất, khả năng chịu lỗi và kết hợp lỏng lẻo với các định dạng đầu vào của nó.

### **Các tính năng của Apache Hive**

• Hive lưu trữ lược đồ trong CSDL và xử lý các dữ liệu vào HDFS.

• Hive được thiết kế giành cho OLAP.

• Hive cung cấp các ngôn ngữ kiểu SQL dùng để truy vấn được gọi là HiveQL hoặc là HQL.

• Hive là quen thuộc, nhanh chóng và có khả năng mở rộng cao

### **Kiến trúc của Apache Hive**

***Sơ đồ sau đây miêu tả kiến trúc của Hive :***

Diagram

Description automatically generated

Sơ đồ thành phần này miêu tả kiến trúc của Hive chứa các đơn vị khác nhau.

• User Interface: Hive là một trong những phần mềm cơ sở hạ tầng kho dữ liệu có thể tạo ra được sự tương tác giữa người dùng và HDFS. Các giao diện thông dùng được người dùng sử dụng mà Hive hỗ trợ hiện nay là Web UI, command line, HD Insight (máy chủ Window).

• Meta Store: Hive chọn các máy chủ cơ sở dữ liệu tương ứng để lưu trữ lược đồ hoặc metadata của các bảng, cơ sở dữ liệu, các cột ở trong một bảng dữ liệu, các loại dữ liệu của chúng và ánh xạ lên HDFS.

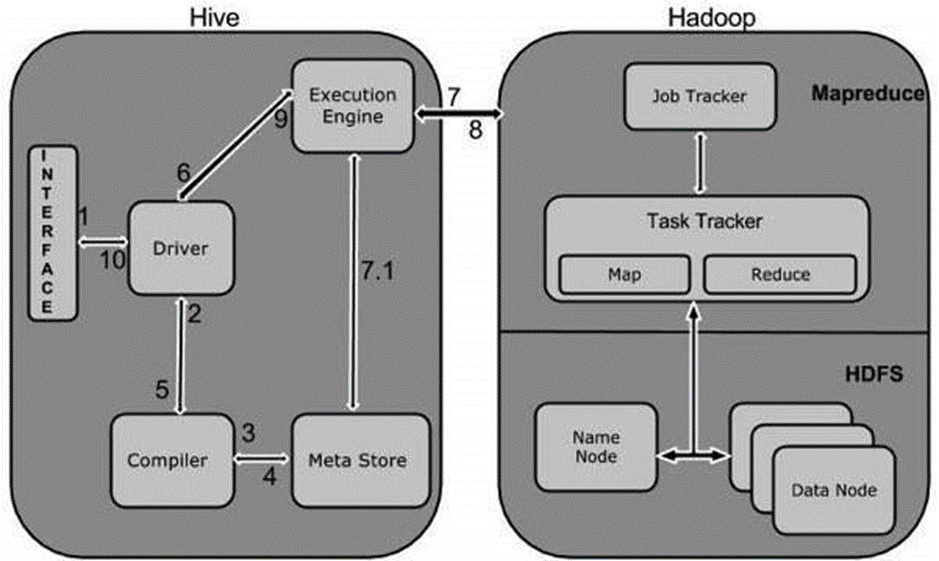
• HiveQL Process Engine: HiveQL tương tự như SQL để truy vấn thông tin lược đồ trên Metastore. Đây thay thế của chương trình MapReduce. Thay vì chúng ta phải viết chương trình MapReduce bằng Java, thay vào đó có thể viết một truy vấn cho MapReduce và xử lý nó một cách nhanh chóng.

• Execution Engine: Phần kết hợp của công cụ xử lý HiveQL và MapReduce là Công cụ thực thi Hive (Hive Execution Engine). Một công cụ thực thi xử lý truy vấn và tạo ra các kết quả giống như kết quả của chương trình MapReduce.

• HDFS hoặc HBASE: Hệ thống tệp phân tán Hadoop hoặc HBASE là các kỹ thuật lưu trữ dữ liệu để lưu trữ dữ liệu vào hệ thống tệp.

### **Quy trình làm việc của Apache Hive**

***Sơ đồ sau mô tả quy trình làm việc giữa Hive và Hadoop.***

******

Cách Hive tương tác với framework Hadoop:

• Thực thi query: Giao diện Hive như command line hoặc giao diện người dùng web gửi truy vấn đến trình điều khiển để thực thi.

• Nhận kế hoạch: Trình điều khiển có sự trợ giúp của trình biên dịch truy vấn để phân tích cú pháp truy vấn để kiểm tra cú pháp và kế hoạch để thực hiện truy vấn hoặc các yêu cầu của truy vấn.

• Nhận metadata: Trình biên dịch gửi yêu cầu metadata đến Metastore (bất kỳ cơ sở dữ liệu nào).

• Gửi metadata: Metastore gửi metadata như một phản hồi cho trình biên dịch.

• Gửi kế hoạch: Trình biên dịch gửi kiểm tra yêu cầu và gửi lại các kế hoạch cho trình điều khiển thực hiện. Ở đây, việc phân tích các cú pháp và biên dịch một truy vấn đã hoàn thành.

• Kế hoạch thực hiện: Trình điều khiển gửi các kế hoạch thực hiện đến các công cụ thực thi.

• Thực thi công việc: Trong nội bộ, các quá trình thực thi công việc là một công việc MapReduce. Công cụ thực thi gửi các công việc đến JobTracker, trong các node Name và ở đây nó gán công việc này cho TaskTracker xử lý, trong node Data. Đến đây, truy vấn thực thi công việc MapReduce. Hoạt động metadata: Trong khi xử lý, công cụ thực thi có thể thực thi các hoạt động metadata đối với Metastore.

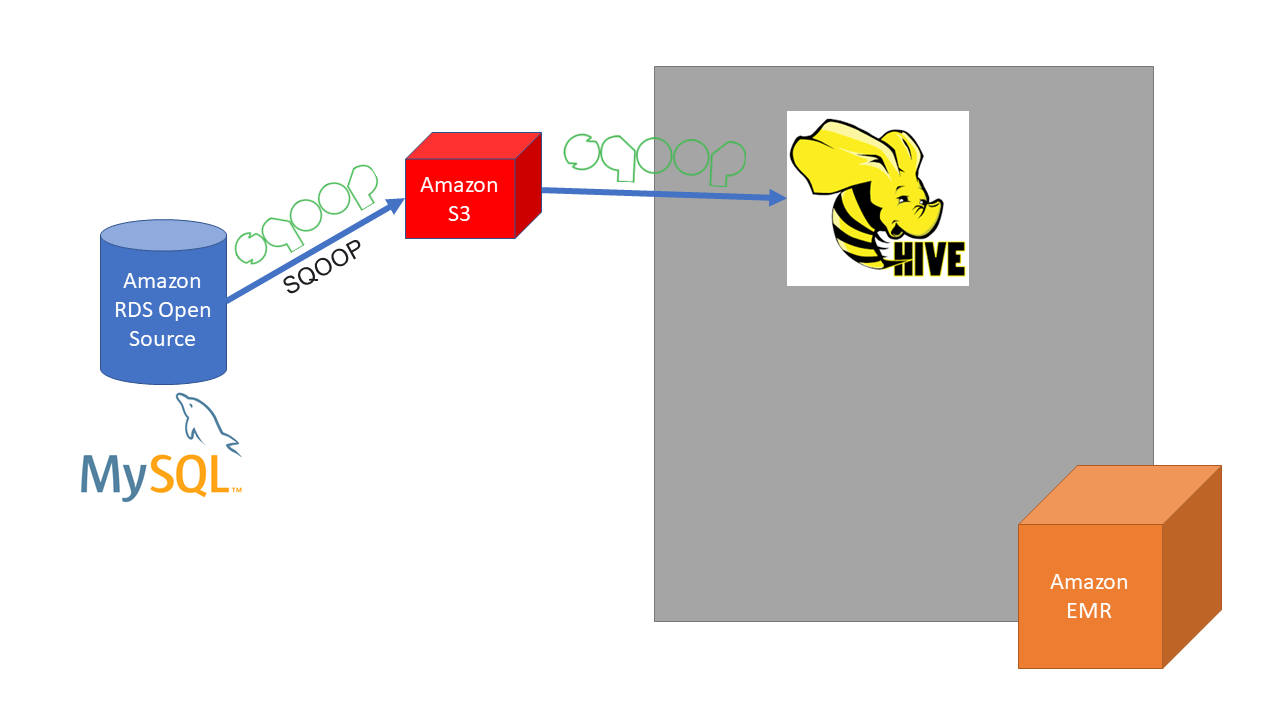
• Lấy kết quả: Công cụ thực thi ghi nhận kết quả từ các node Data.

• Gửi kết quả: Công cụ thực thi gửi các giá trị kết quả đó đến trình điều khiển.

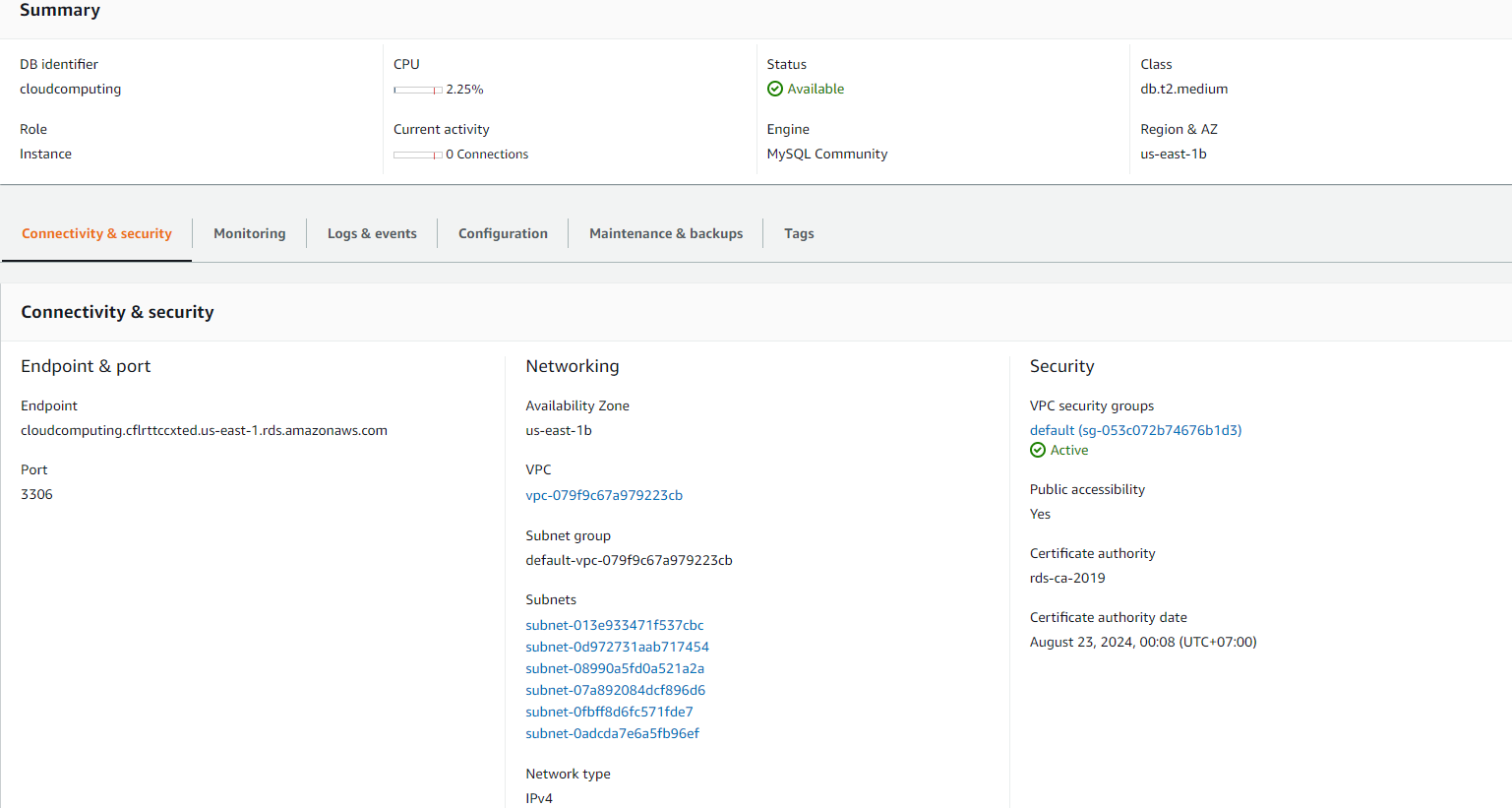
• Gửi kết quả: Trình điều khiển gửi kết quả đến Giao diện người dùng Hive.

## **Chương 2 : CÀI ĐẶT, THIẾT KẾ VÀ XÂY DỰNG DATA****WAREHOUSE**

Hướng xây dựng data warehouse:



**Bước 1:** Cấu hình và tạo database có tên là cloudcomputing trên RDS.



**Bước 2:** Cấu hình rules từ MySQL kết nối đến server.

Graphical user interface, application, Word

Description automatically generated

Copy Endpoint

Graphical user interface, text, application, email

Description automatically generated

**Bước 3:** Kết nối tới database trên RDS bằng MySQL:

Graphical user interface, text, application

Description automatically generated

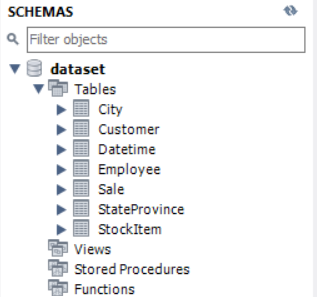
Thêm dòng opt local infile = 1

Graphical user interface, text, application

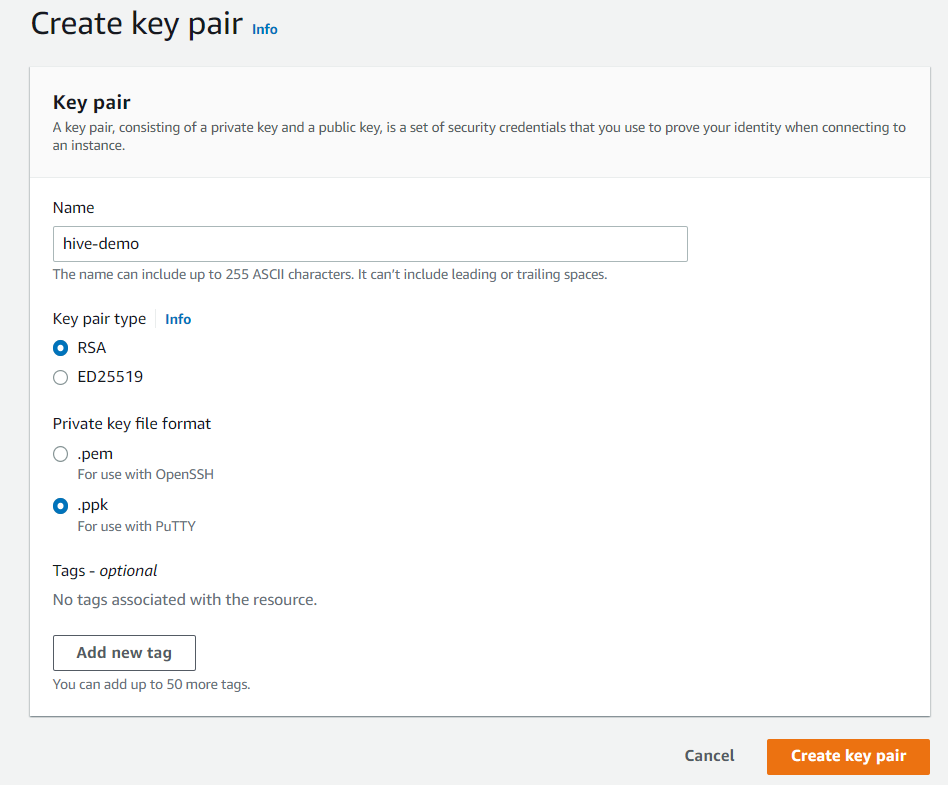
Description automatically generated

**Bước 4:** Load data từ file .csv vào các table đã tạo trong MySQL:

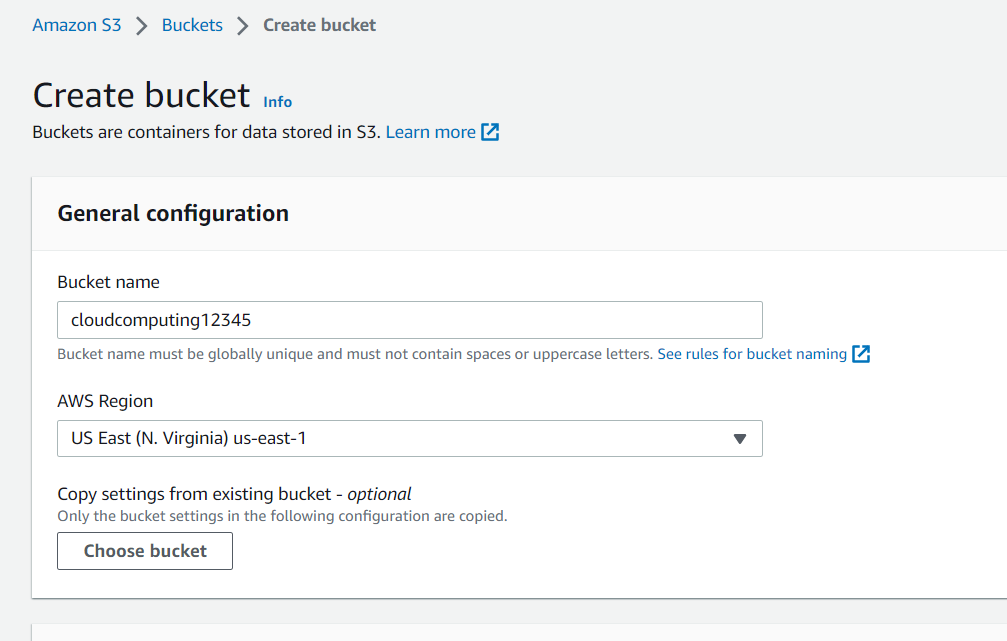




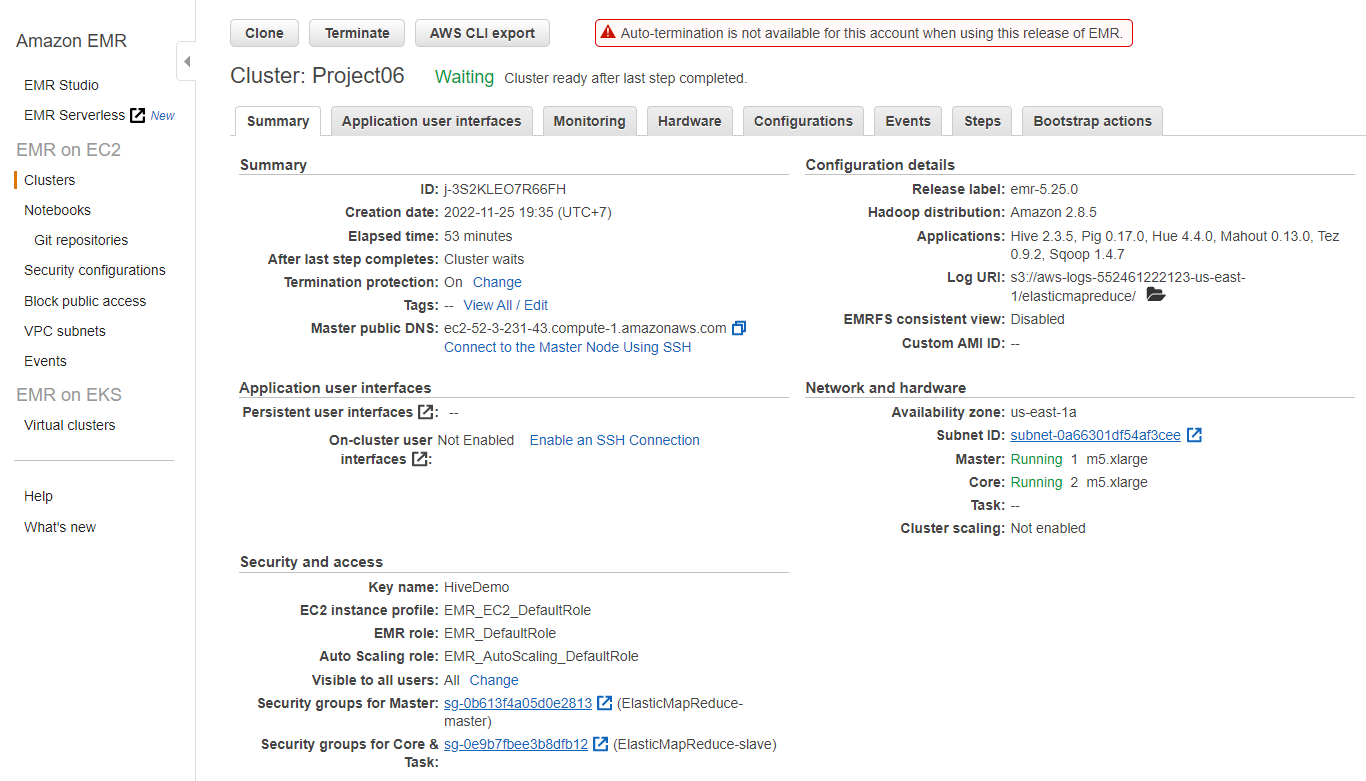
**Bước 5:** Tạo EC2 keypairs và lưu file hive-demo.ppk về máy.



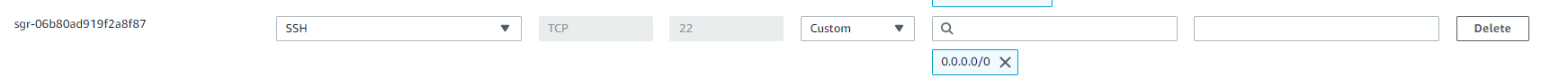
**Bước 6:** Tạo 1 bucket trên S3



**Bước 7:** Cấu hình và tạo EMR Cluster.

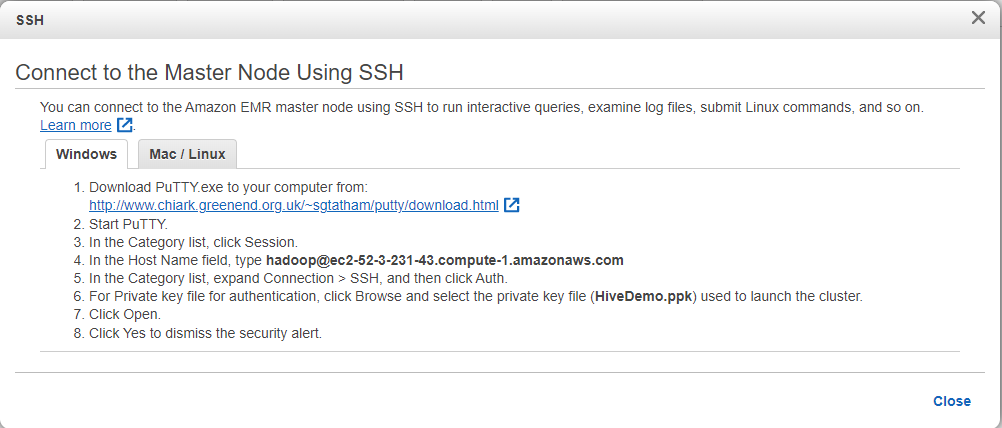


**Bước 8:** Cấu hình inbound rules SSH kết nối đến EMR.

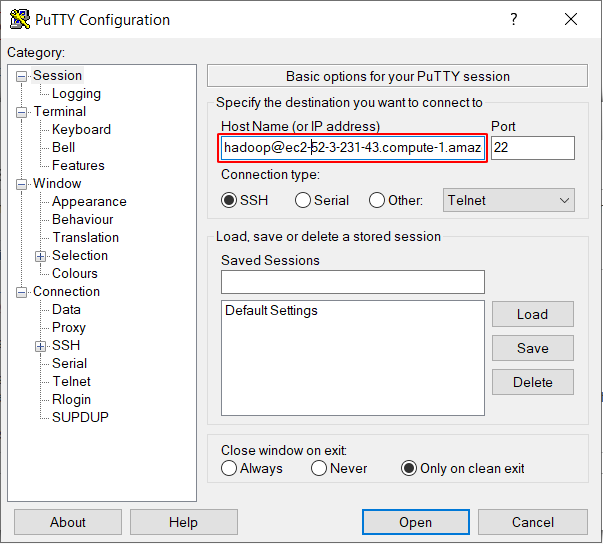


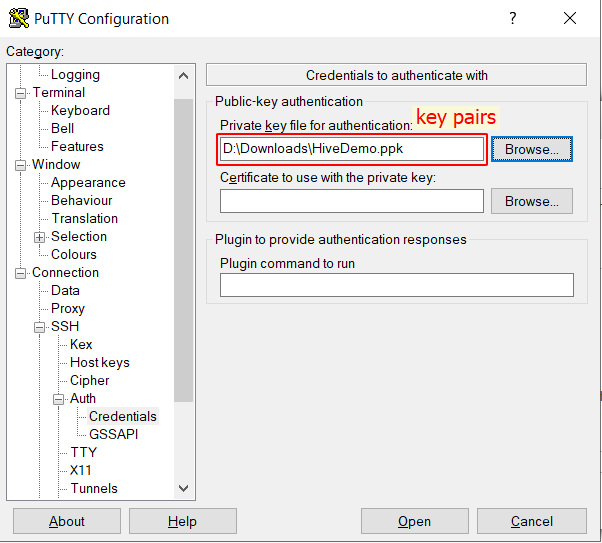
**Bước 9:** Kết nối tới master node:

Tải PuTTY.exe về máy theo link <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

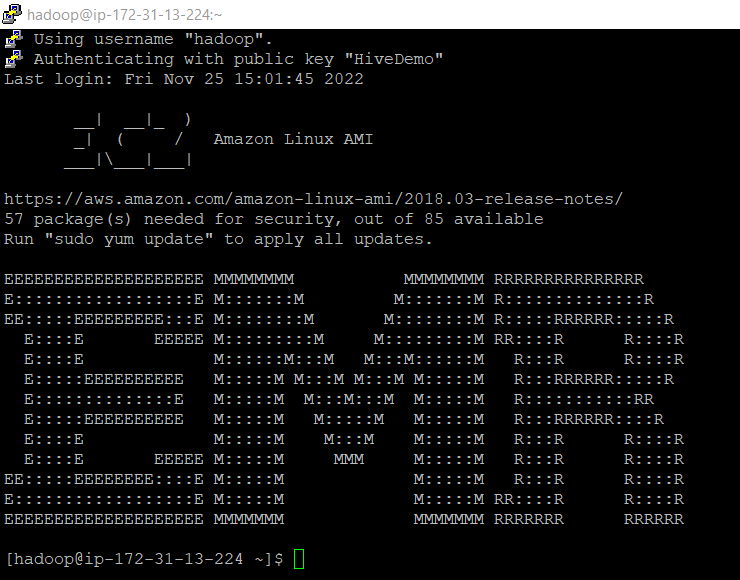


Kết nối tới Master Node sử dụng SSH





Chọn keypair là file hive-demo.ppk đã lưu từ trước.



Click Open và kết nối thành công đến Master Cluster.

**Bước 10:** Cài đặt jdbc driver:

Tải tệp JDBC Driver trực tiếp xuống Master Node theo đường dẫn: <http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.46.tar.gz>

Text

Description automatically generated

Sau đó giải nén tệp JDBC mới tải.

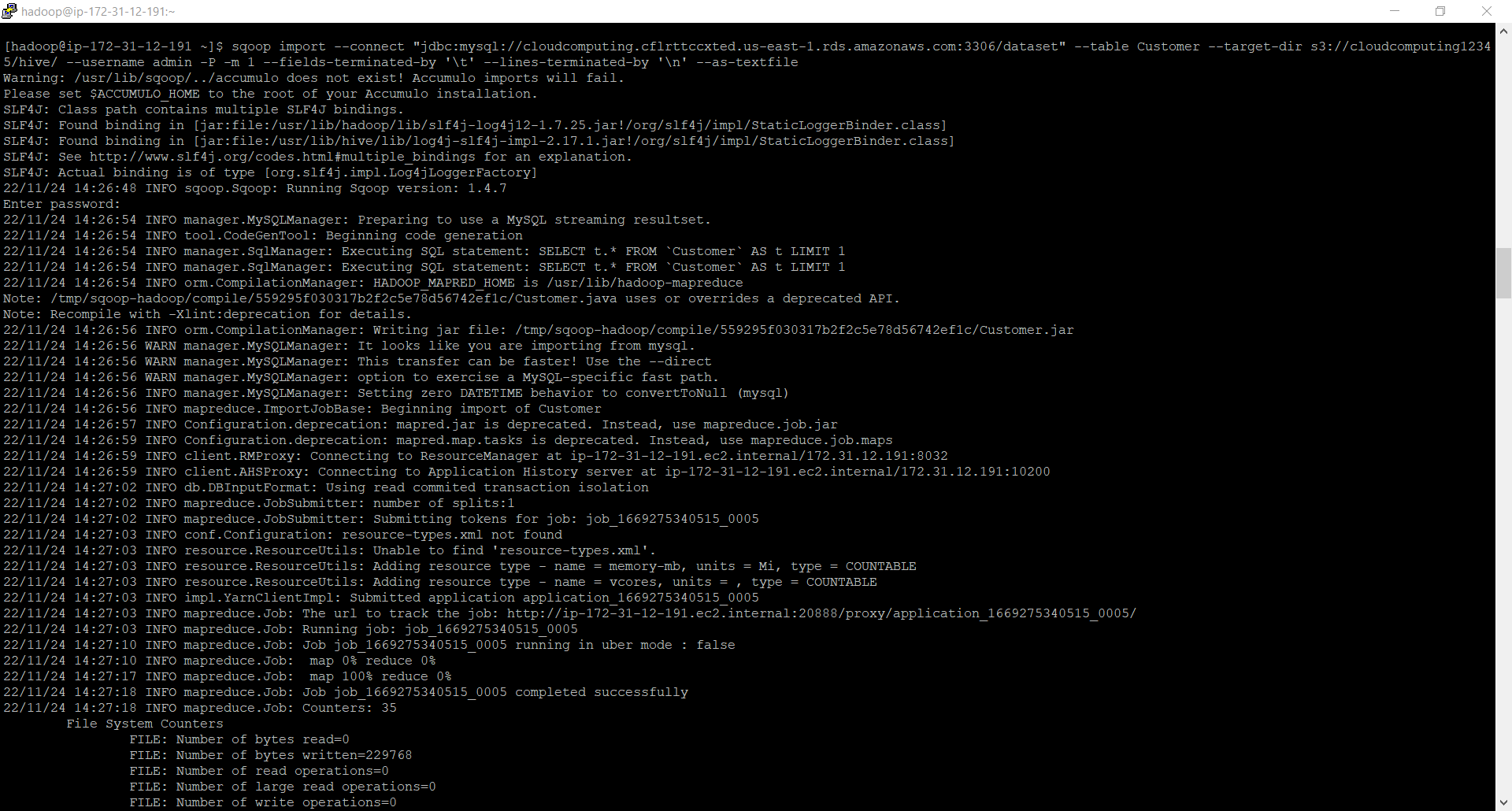


Copy file JDBC vừa giải nén vào thư mục var/lib/sqoop của EMR Master Cluster

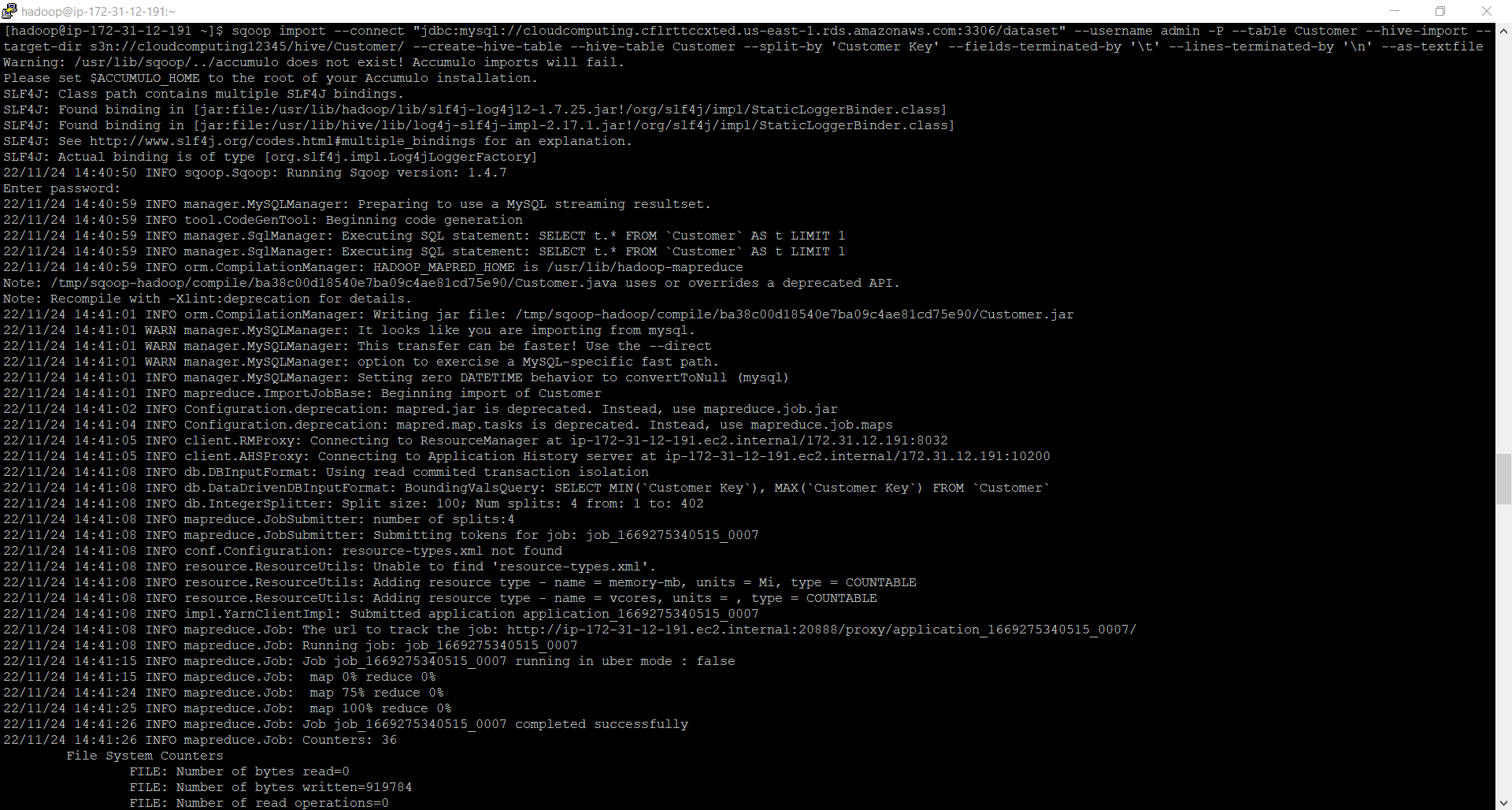


**Bước 11:** Import dữ liệu từ rds server lên datalake.

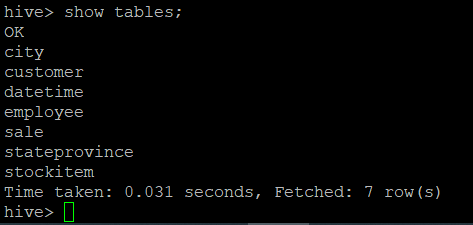
Sử dụng spoop để import dữ liệu từ RDS Sever lên datalake là S3



**Bước 12:** Import dữ liệu từ S3 lên hive table.

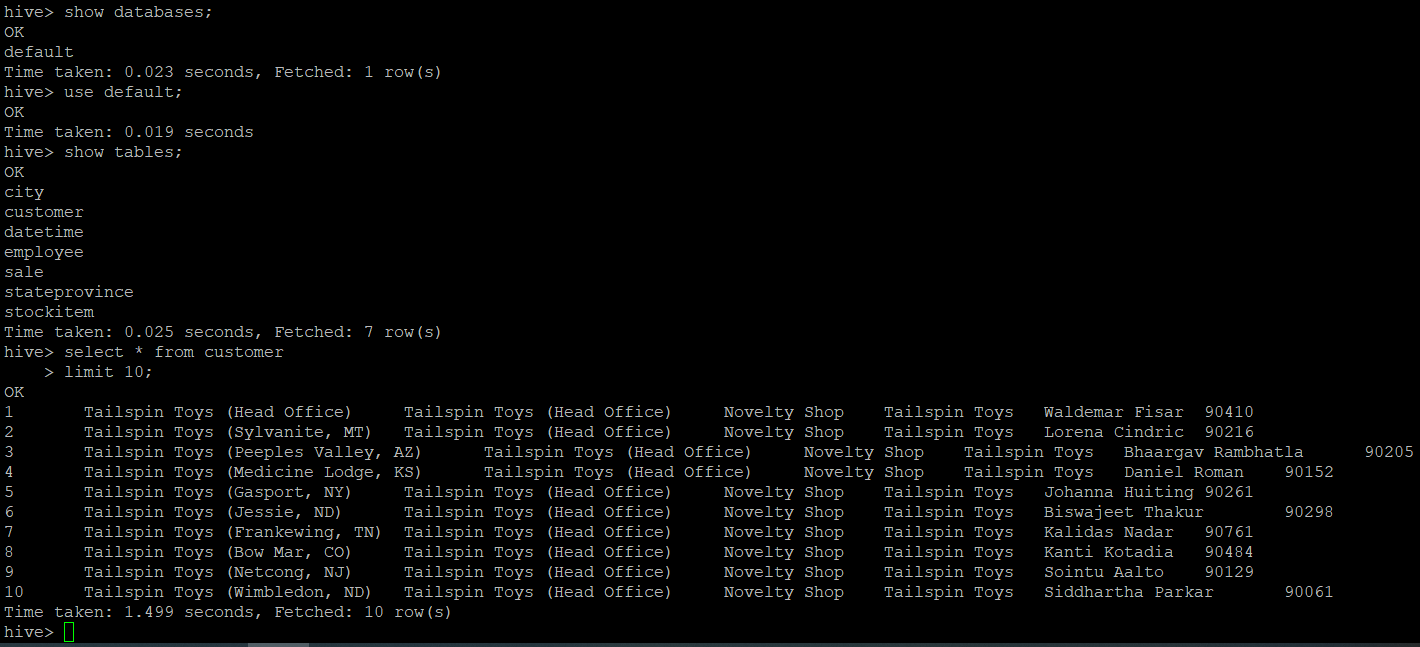
Sử dụng spoop để import dữ liệu từ S3 lên hive table

Tạo và import dữ liệu cho 7 tables.

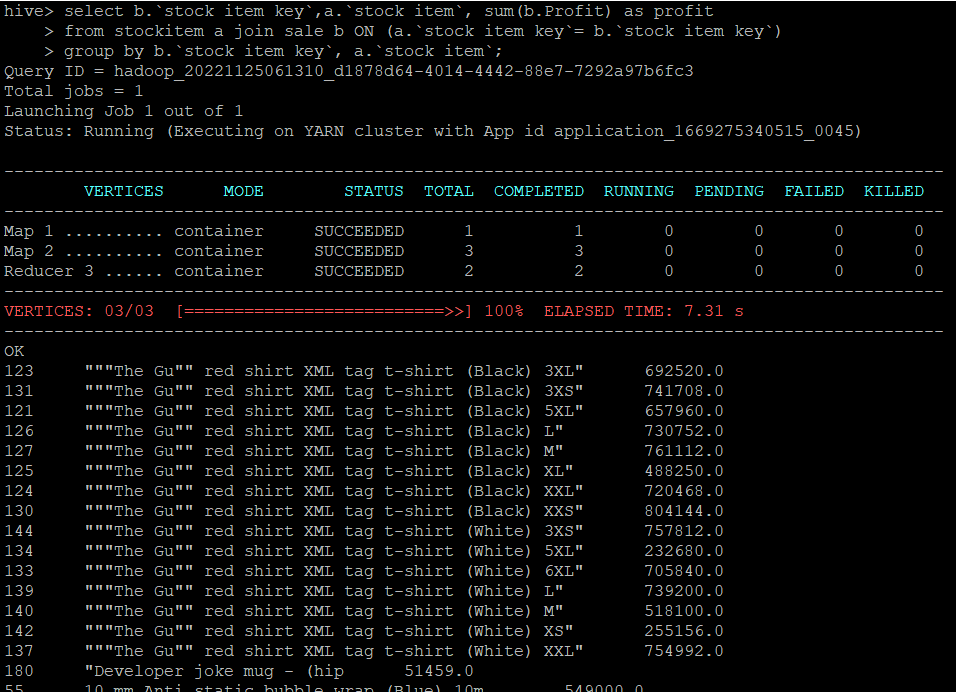


**Bước 13:** Test truy vấn dữ liệu bằng hive.

Xuất 10 rows từ bảng customer:



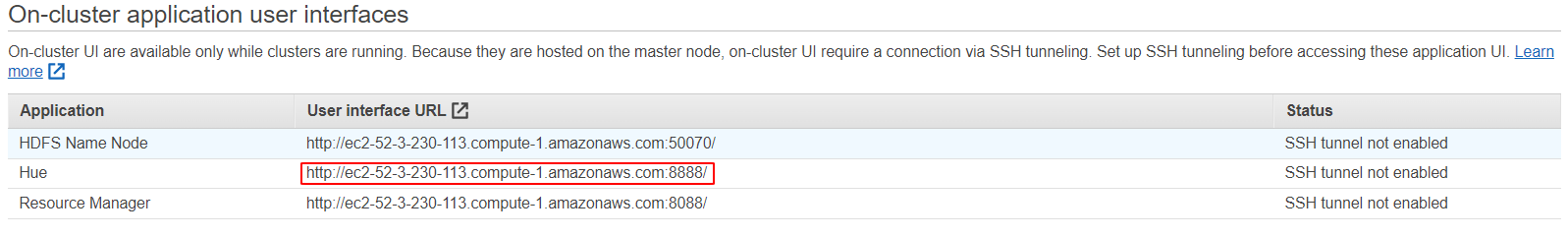
Tạo truy vấn tính tính tổng lợi nhuận thu được của từng sản phẩm:



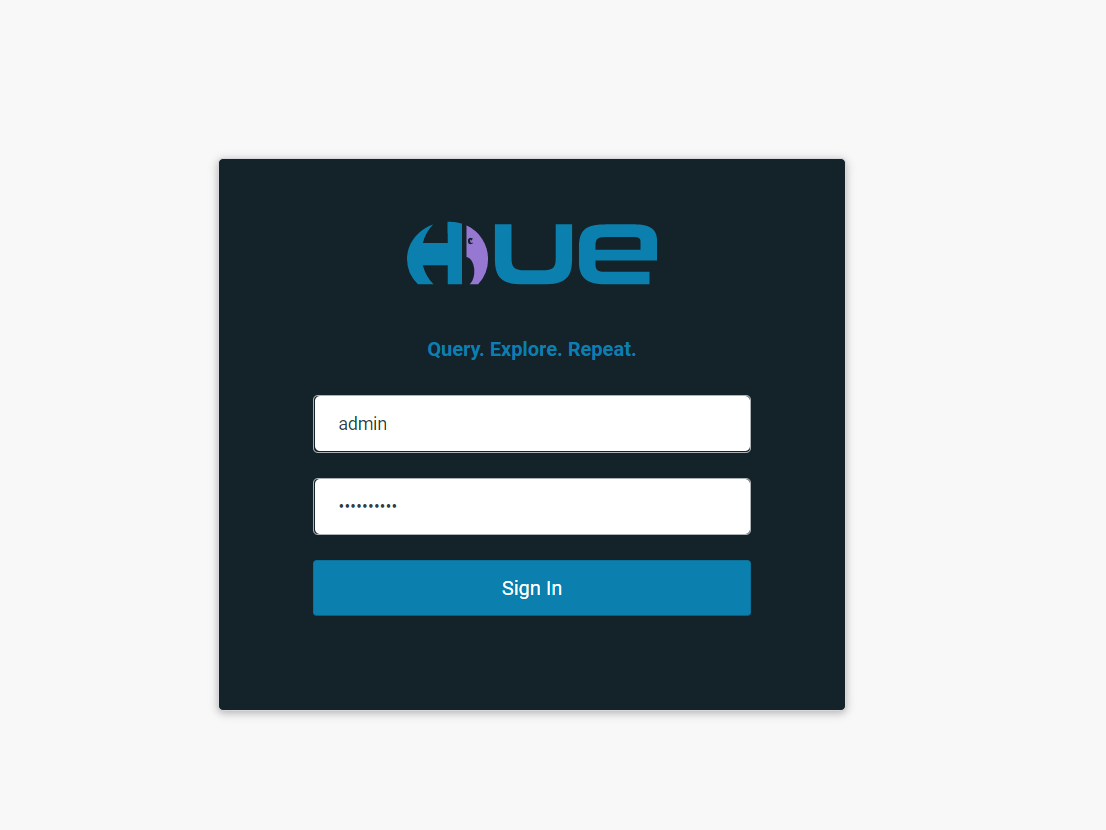
**Bước 14:** Sử dụng hive trên HUE Application.

Để chạy trên server ta cần mở các port tương ứng với từng application. Vì vậy, ta mở port 8888 để chạy application HUE thực hiện tạo database và truy vấn trên hive.



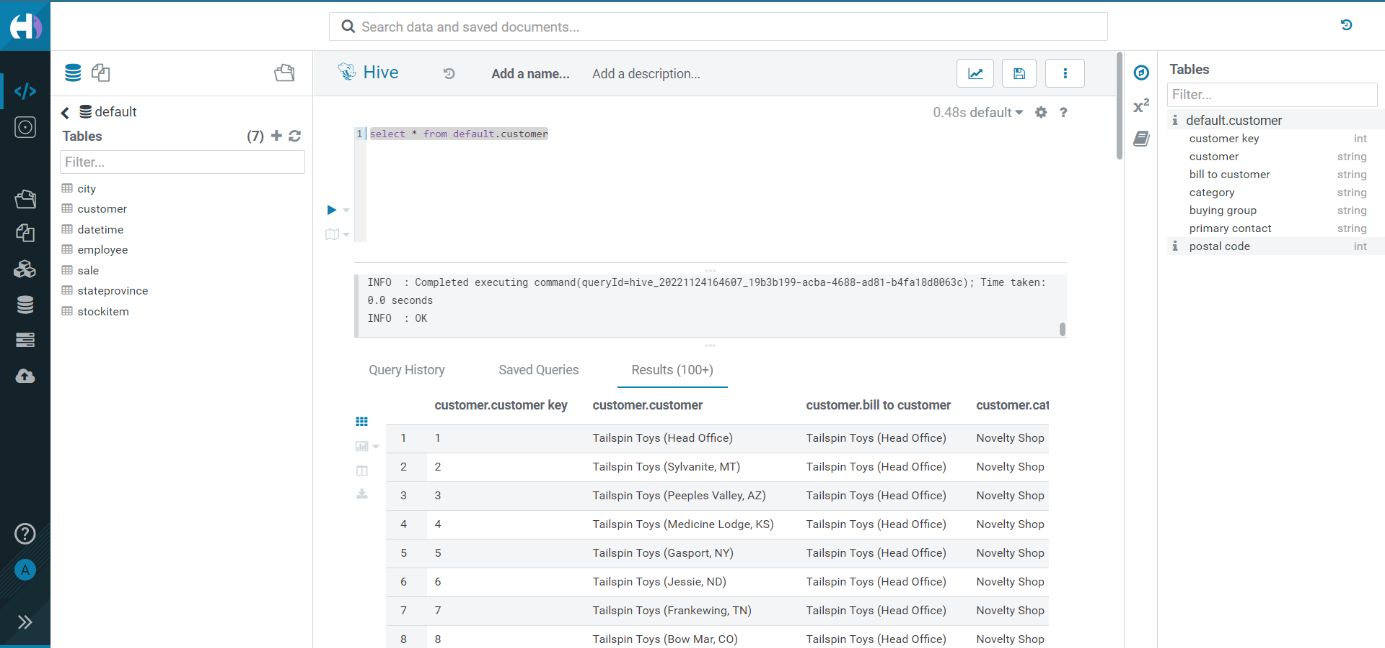


Ta mở HUE app theo đường dẫn trong trang Application user interfaces trên EMR



HUE yêu cầu tạo tài khoản để đăng nhập.

Thử truy vấn hive trên HUE.



Truy vấn bằng HQL trên HUE giúp dễ dàng quan sát kết quả truy vấn hơn:

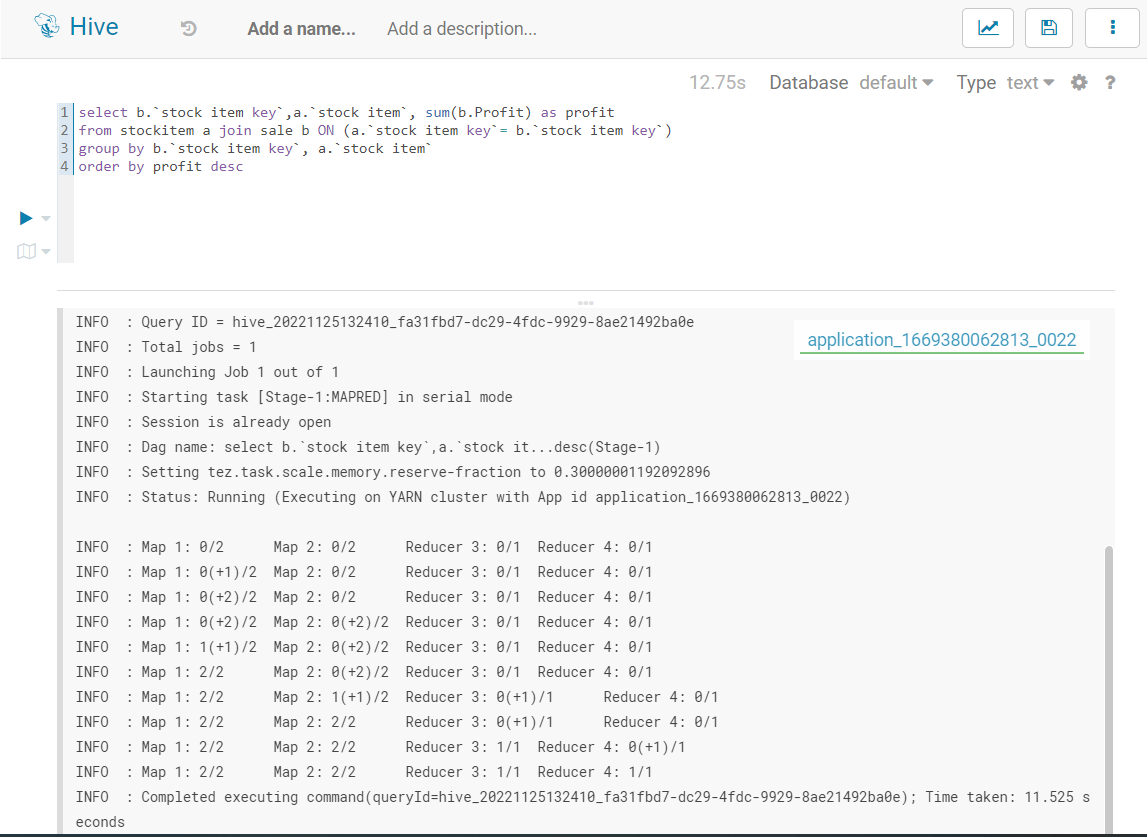
**+ Tổng lợi nhuận của từng loại sản phẩm:**

select b.`stock item key`,a.`stock item`, sum(b.Profit) as profit

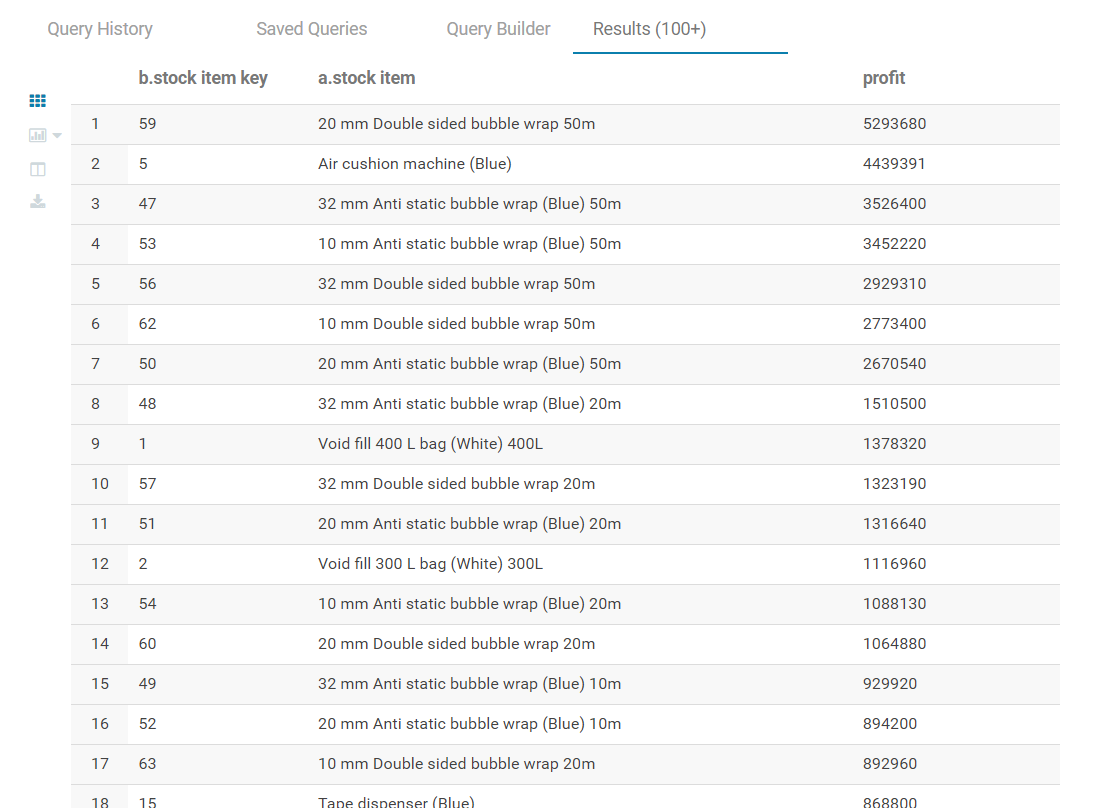
from stockitem a join sale b ON (a.`stock item key`= b.`stock item key`)

group by b.`stock item key`, a.`stock item`

order by profit desc



Kết quả:



**+Top 20 khách hàng có tổng số lần mua hàng nhiều nhất:**

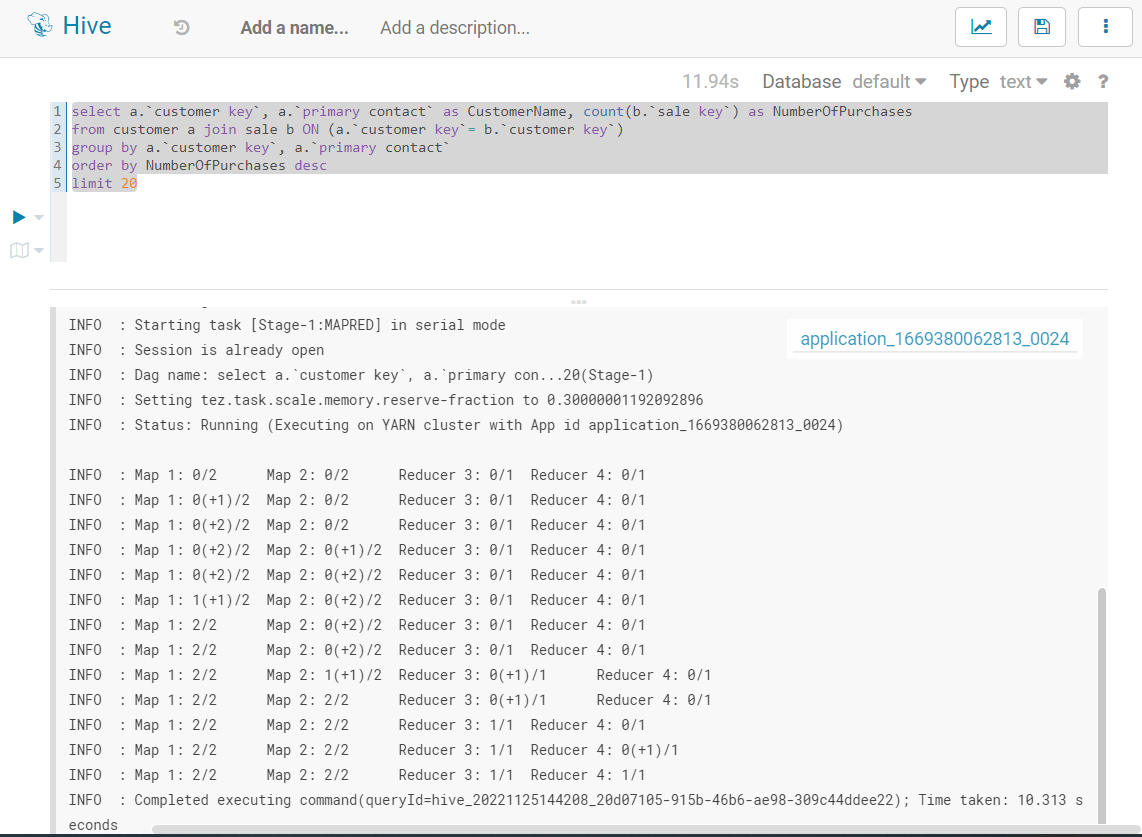
select a.`customer key`, a.`primary contact` as CustomerName, count(b.`sale key`) as NumberOfPurchases

from customer a join sale b ON (a.`customer key`= b.`customer key`)

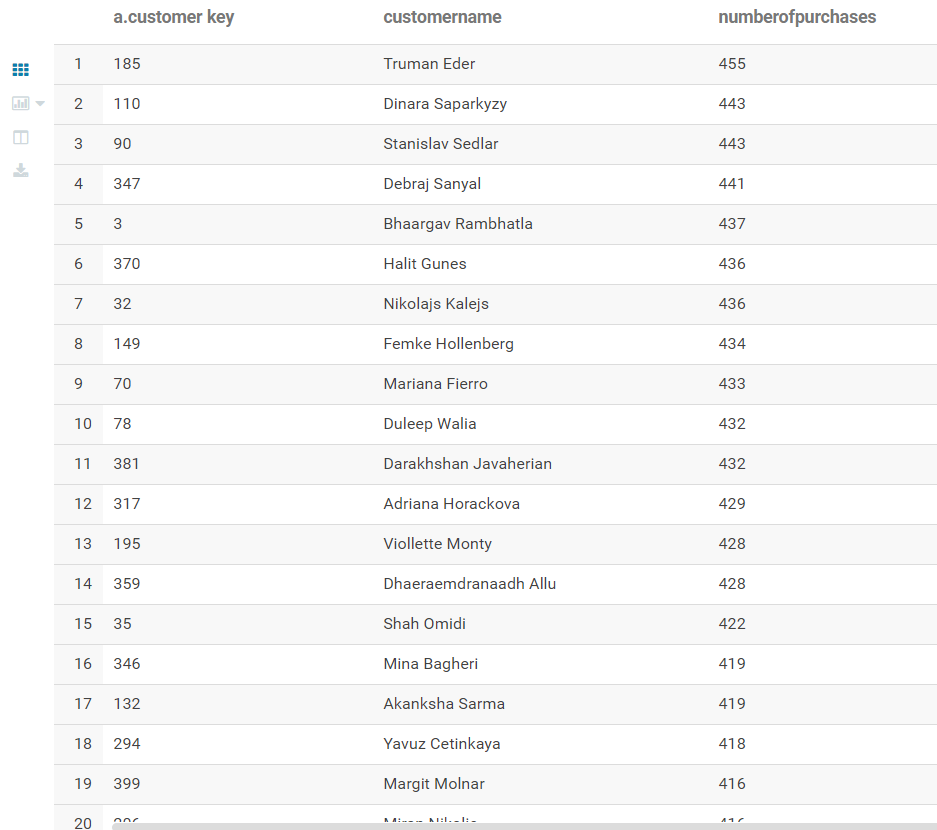
group by a.`customer key`, a.`primary contact`

order by NumberOfPurchases desc

limit 10



Kết quả:



# **PHẦN KẾT LUẬN**

## **Kết quả thực hiện và thực nghiệm được**

* Sau thời gian tìm hiểu,nghiên cứu và cũng như thực hiện đề *tài “Tìm hiểu Apache Hive và viết ứng dụng demo”,* nhóm chúng em đã đạt được những kết quả như sau:
* Nắm bắt được các kiến thức quan trọng về Apache Hive, truy vấn với HUE và áp dụng kiến thức để thiết kế và xây dựng một Data warehouse đơn giản. Biết được cách Apache Hive hoạt động trên Hadoop, luồng dữ liệu của Hive, kiến trúc, đặc trưng, cách tổ chức dữ liệu trong Hive.

## **Ưu điểm**

* Hoạt động của Hive diễn ra một cách chính xác, không xảy ra tình trạng lỗi trong hệ thống.
* Lưu trữ được một lượng dữ liệu lớn.
* Xử lý các thông tin, truy vấn dữ liệu chính xác và nhanh chóng.

## **Nhược điểm**

* Hiện tại nhóm đang làm load tất cả data trong một lần ( Full load )

## **Hướng phát triển**

* Có thể import từng thành phần của data (Incremental load)
* Tiếp tục hoàn thiện hơn các chức năng còn thiếu trong đề tài.
* Xây dựng và quản lý data warehouse lớn hơn nếu có thể.

# **TÀI LIỆU THAM KHẢO**

1. Migrate RDBMS or On-Premise data to EMR Hive, S3 using EMR – Sqoop: <https://aws.amazon.com/vi/blogs/big-data/migrate-rdbms-or-on-premise-data-to-emr-hive-s3-and-amazon-redshift-using-emr-sqoop/>
2. Transfer MySQL RDS data to S3 & Import S3 Data to Hive using SQOOP. <https://www.youtube.com/watch?v=zqyOOyOlcAk&t=796s>