**TRƯỜNG ĐH SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**🕯✡🕮🕮✡🕯**

****

**MÔN HỌC: Big data Analysis**

**GVHD: Cô Lê Thị Minh Châu**

**Nhóm SV thực hiện: Nhóm 10**

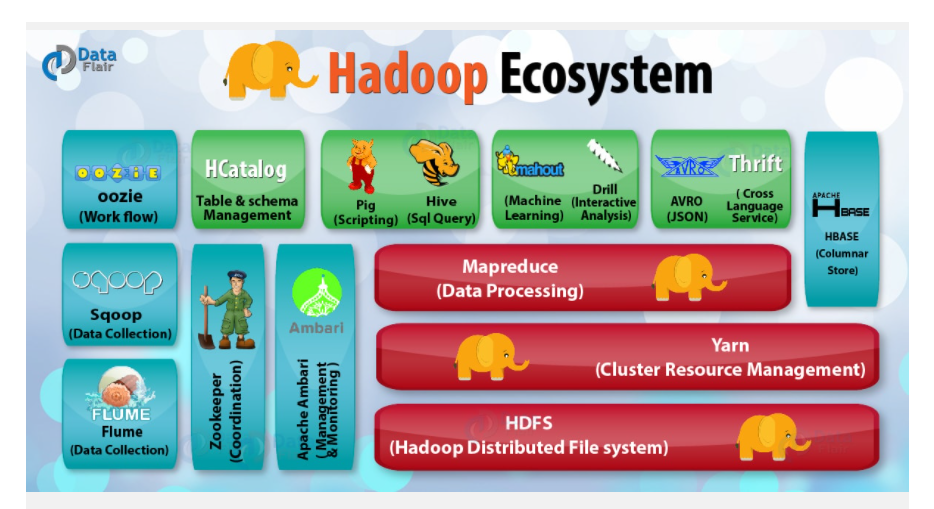
Hồ Văn Trường 18110384

Mai Đăng Khoa 18110306

**TP. Hồ Chí Minh, tháng 12 năm 2021**

|  |  |
| --- | --- |
| Bảng Phân Công | |
| Hồ Văn Trường | Làm topic đề xuất phim |
| Mai Đăng Khoa | Tìm hiểu về Apache Hive, cài đặt và demo |

1. **Tìm hiểu về Apache Hive**
2. **Tổng quan về hệ sinh thái Hadoop**



Hệ sinh thái Hadoop bao gồm: HDFS, MapReduce, Yarn, Apache Pig, Apache HBase, Apache Hive, Apache Sqoop, …

Trong đó HDFS là thành phần quan trọng nhất của Hệ sinh thái Hadoop, là hệ thống lưu trữ chính của Hadoop dựa trên java cung cấp khả năng mở rộng, khả năng chịu lỗi, lưu trữ dữ liệu và tiết kiệm chi phí cho Big Data.

MapReduce là thành phần cốt lõi của hệ sinh thái Hadoop cung cấp khả năng xử lý dữ liệu, cho phép xử lý song song tất cả dữ liệu được lưu trữ bởi HDFS.

YARN cũng là một trong những thành phần quan trọng nhất của Hệ sinh thái Hadoop, cung cấp công tác quản lý tài nguyên. Có thể coi YARN là hệ điều hành của Hadoop vì nó chịu trách nhiệm quản lý và giám sát khối lượng công việc.

Apache Pig là ngôn ngữ kịch bản cấp cao để viết các chương trình phân tích dữ liệu cho các tập dữ liệu khổng lồ trong cụm Hadoop.

Apache HBase là cơ sở dữ liệu hướng theo cột hỗ trợ hoạt động của HDFS trên cơ sở thời gian thực.

Apache Hive là một công cụ cơ sở hạ tầng kho dữ liệu để xử lý dữ liệu cấu trúc trong Hadoop.

Apache Sqoop là một ứng dụng giúp nhập và xuất dữ liệu từ Hadoop sang các hệ thống quản lý cơ sở dữ liệu quan hệ khác.

1. **Apache Hive**

Apache Hive là một công cụ cơ sở hạ tầng kho dữ liệu để xử lý dữ liệu cấu trúc trong Hadoop. Nằm trên đỉnh Hadoop để tóm tắt dữ liệu lớn và giúp truy vấn, phân tích dễ dàng.

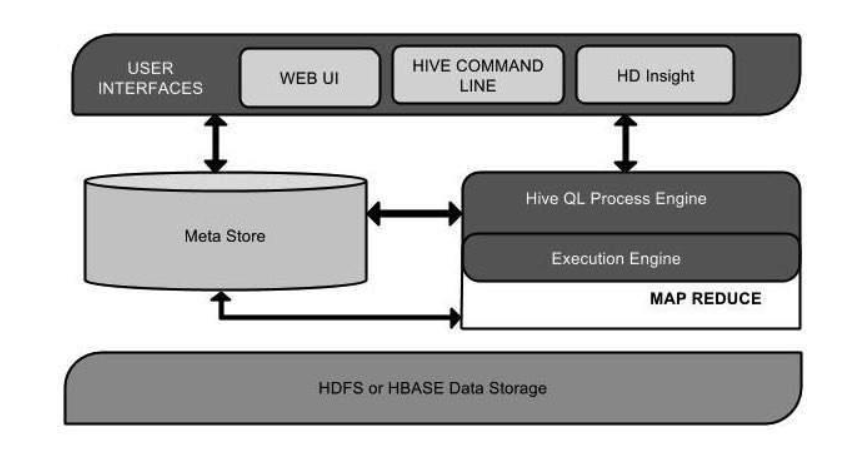
1. Đặc trưng

Lưu trữ lược đồ trong cơ sở dữ liệu và xử lý dữ liệu vào HDFS

Cung cấp ngôn ngữ kiểu SQL để truy vấn được gọi là HiveQL hoặc HQL

Nhanh chóng và có khả năng mở rộng

1. Kiến trúc



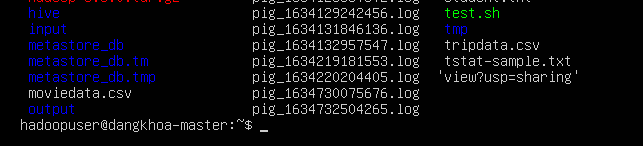
Kiến trúc của Apache Hive gồm:

* User Interface: Cung cấp giao diện web để tương tác với HDFS, tương tác command line.
* Meta Store: Lưu thông tin cơ bản về cấu trúc dữ liệu. Các thông tin gồm id của database(schema), id của table, id của index, định dạng của table, ...
* Hive QL: Tương tự như SQL, dùng để truy vấn dữ liệu dựa trên thông tin metastore cung cấp. Phương pháp thay thế cho việc phải viết trương trình mapreduce truyền thống.
* Execution Engine: Chuyển hóa các lệnh Hive QL thành MapReduce
* HDFS hoặc HBase: Lưu trữ dữ liệu.

1. **Cài Đặt Apache Hive**

* Giải nén và đổi tên thư mục Hive





* Giải nén và đổi tên thư mục Derby



* Thiết lập môi trường cho Hive và Derby

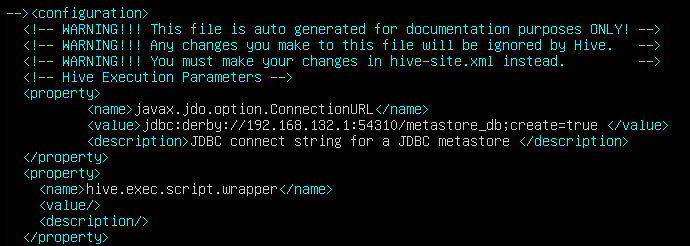


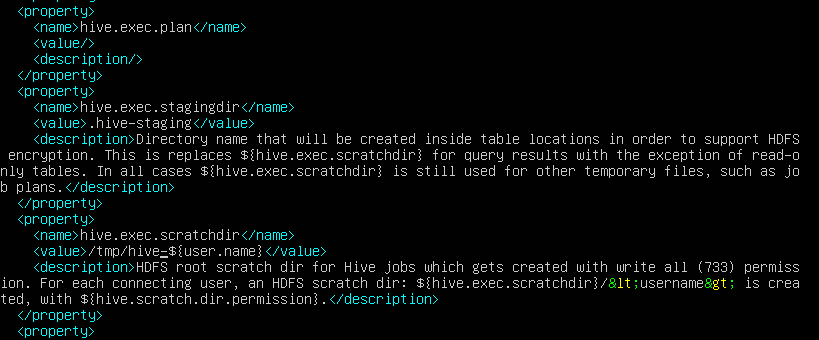


* Áp dụng tất cả các thay đổi vào hệ thống đang chạy hiện tại

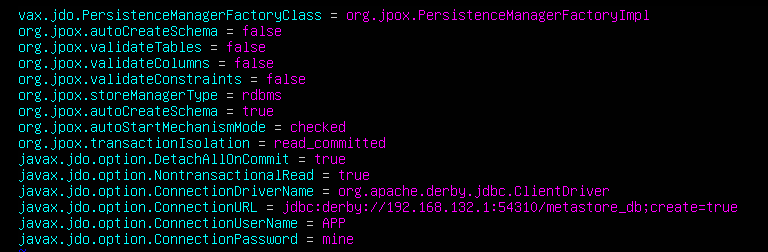


* Chỉnh sửa hive-site.xml

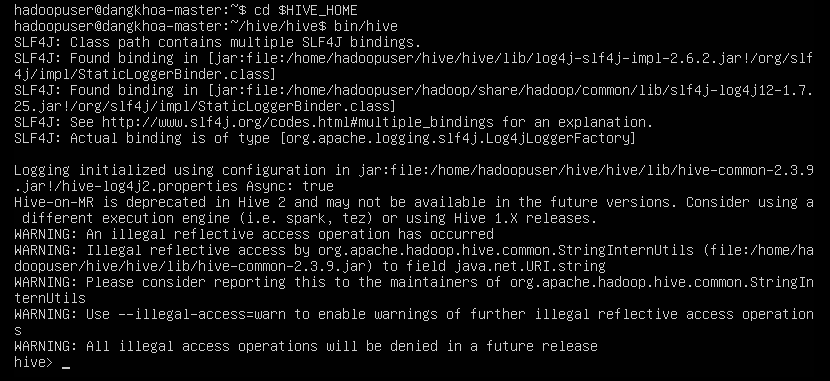


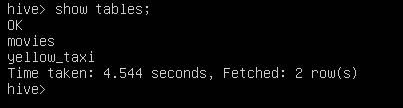


* Tạo tệp jpox.properties và thêm vào đoạn mã như sau



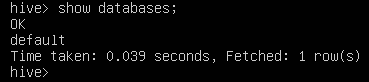
Chạy Hive



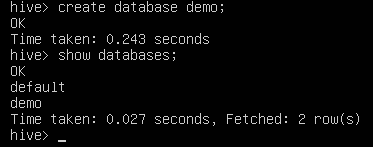


1. **Một số thao tác cơ bản**

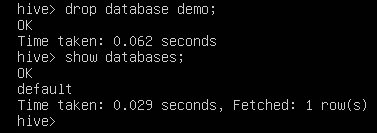
show databases;



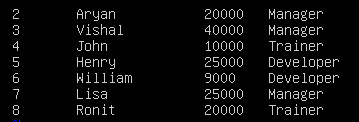
create database demo;



drop database demo;

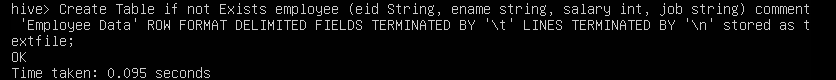


Tạo file employee.txt

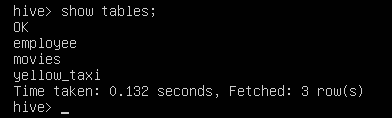




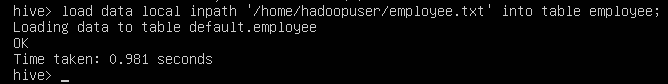
Create Table if not Exists employee (eid String, ename string, salary int, job string) comment 'Employee Data' ROW FORMAT DELIMITED FIELDS terminated by '/t' Lines terminated by '\n' stored as textfile;



show tables;



load data local inpath '/home/hadoopuser/employee.txt' into table employee;

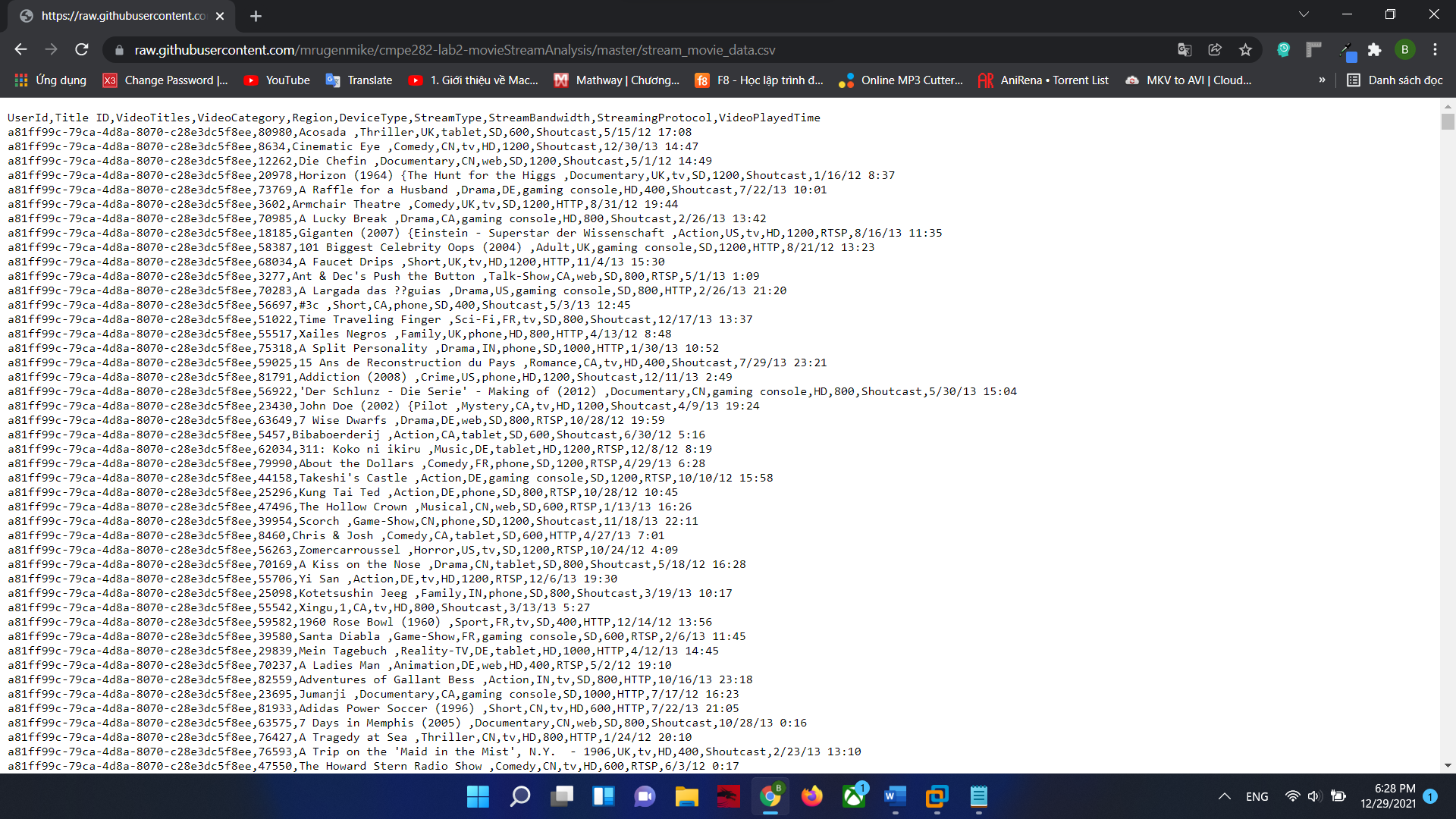


Select \* from employee;



1. **Ví dụ minh họa**

Data Movies: <https://raw.githubusercontent.com/mrugenmike/cmpe282-lab2-movieStreamAnalysis/master/stream_movie_data.csv>



Các field: userid, titleid, videotitles, videocategory, region, devicetype

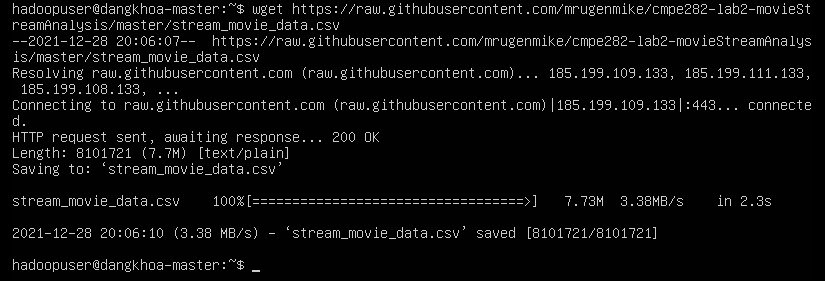
streamtype, streambandwidth, streamingprotocol, videoplayedtime

(id người dùng, id tiêu đề, tiêu đề video, thể loại, khu vực, loại thiết bị,

loại luồng, độ rộng băng tần của luồng, luồng trong giao thức, thời gian phát video)

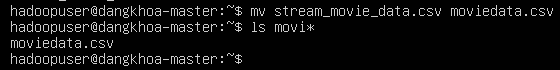
* Download data



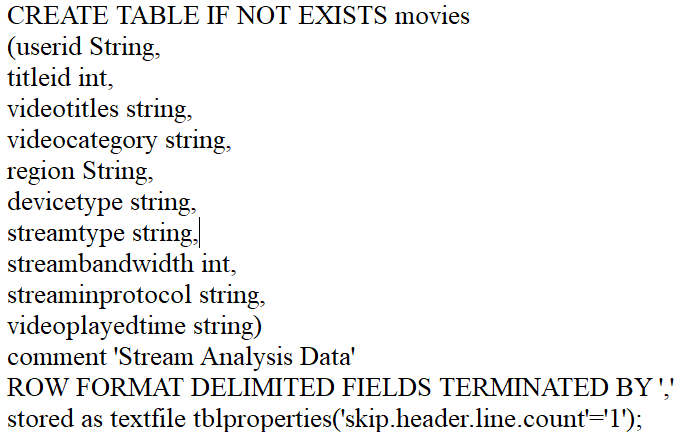


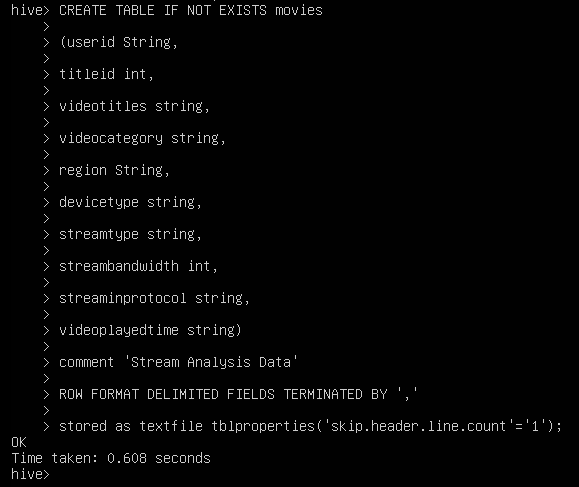


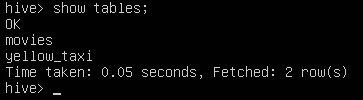
* Đổi tên file thành moviedata.csv



* Create table movies

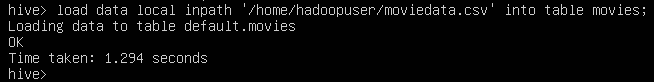






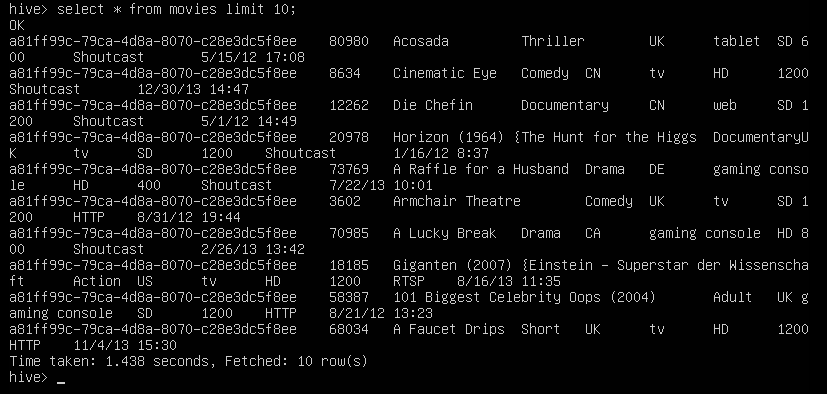
* Load data





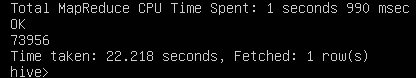
* Kiểm tra thử 10 dòng dữ liệu

select \* from movies limit 10;

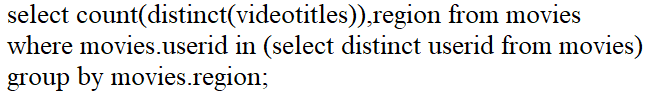


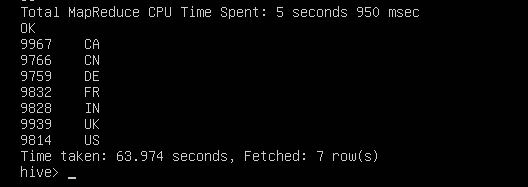
* Tổng số dòng dữ liệu (chiếu theo userid)





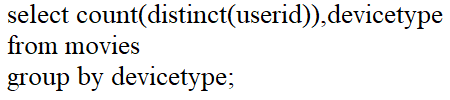
* Truy vấn số lượng video từ các khu vực

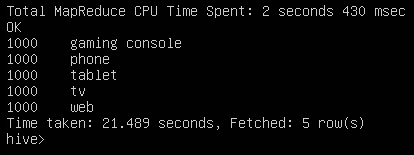




Mệnh đề distinct được dùng để loại bỏ trùng lặp.

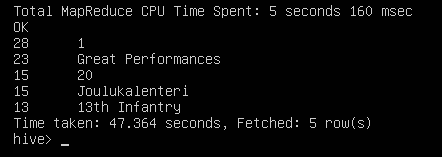
* Truy vấn số lượng userid theo loại thiết bị



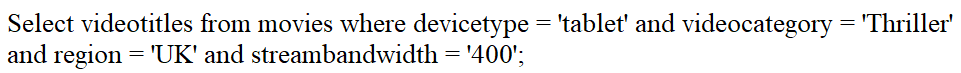


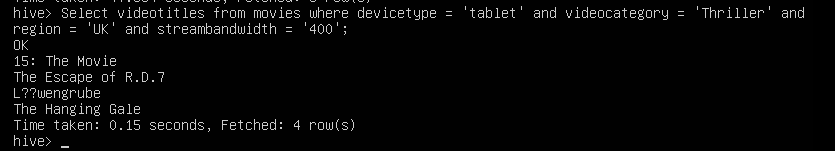
* Truy vấn số lượng userid theo videotitles, sắp xếp theo thứ tự giảm dần với giới hạn là 5





* Truy vấn videotitles có devicetype = 'tablet', videocategory = 'Thriller', region = 'UK' và streambandwidth = '400';



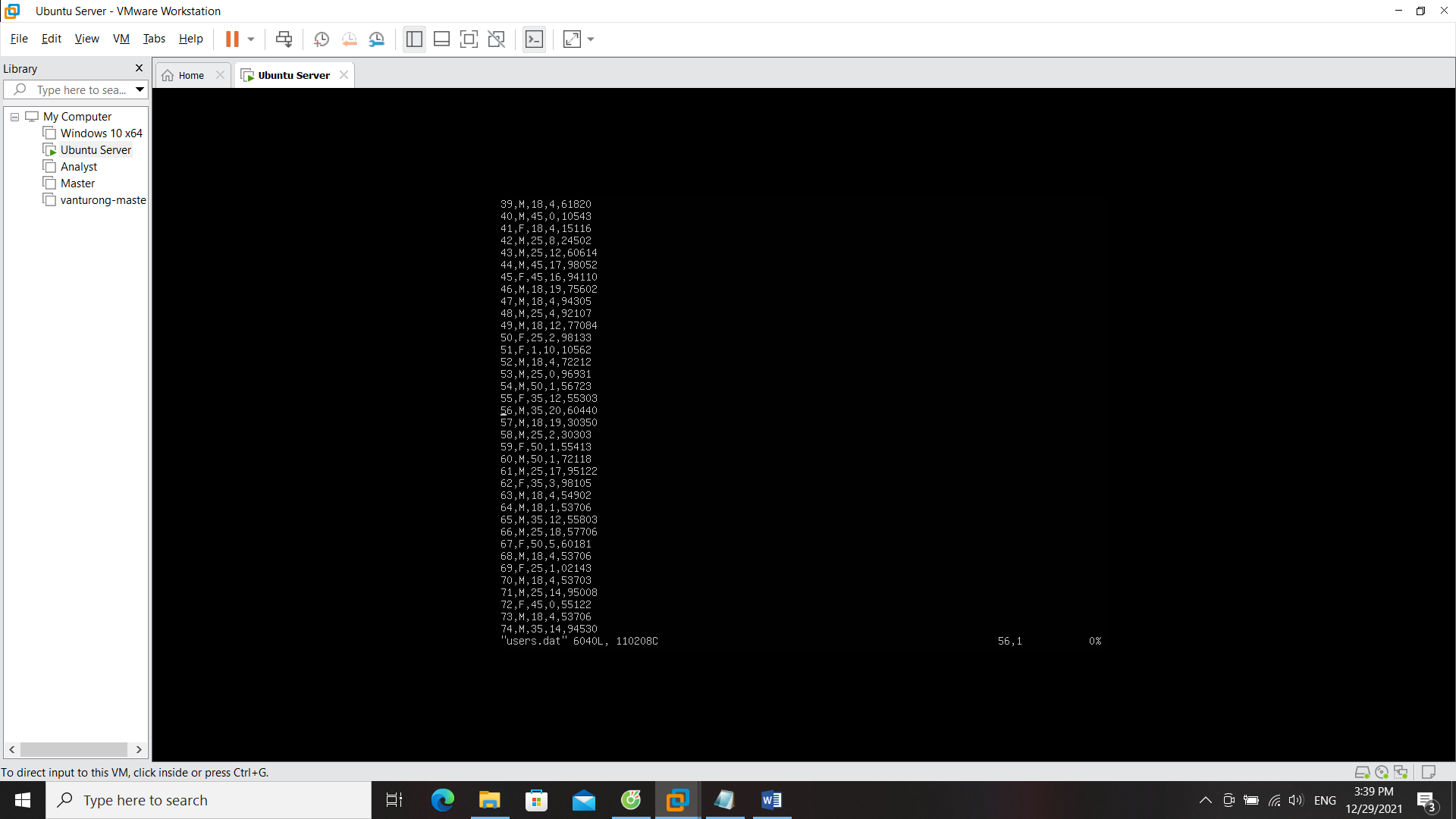


1. **Gợi ý Phim**
2. Giới thiệu về project
3. Mô tả dữ liệu

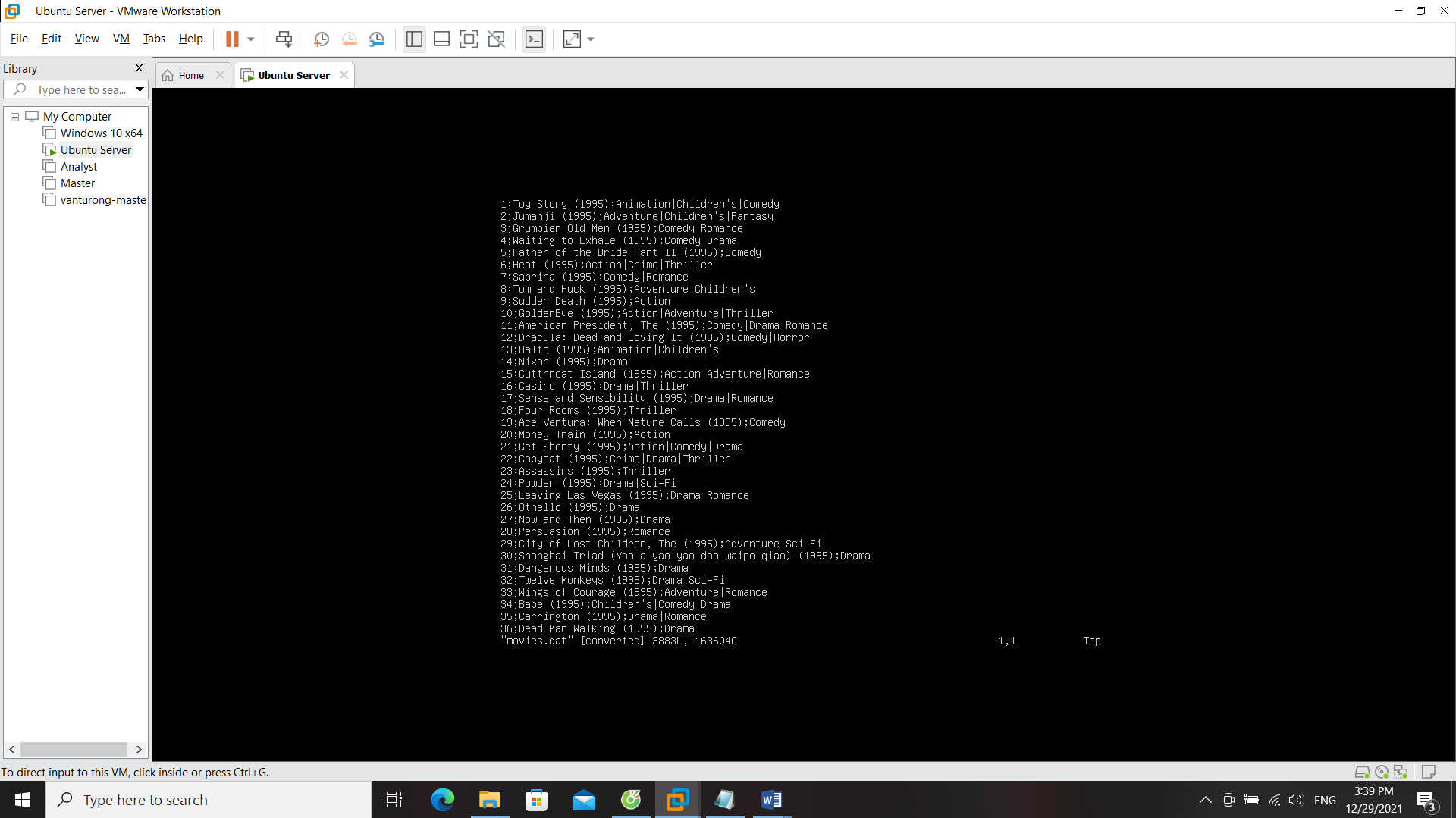
Dữ liệu chứa 1.000.209 xếp hạng ẩn danh của khoảng 3.900 phim được thực hiện bởi 6.040 người dùng.

Nguồn: Kaggle

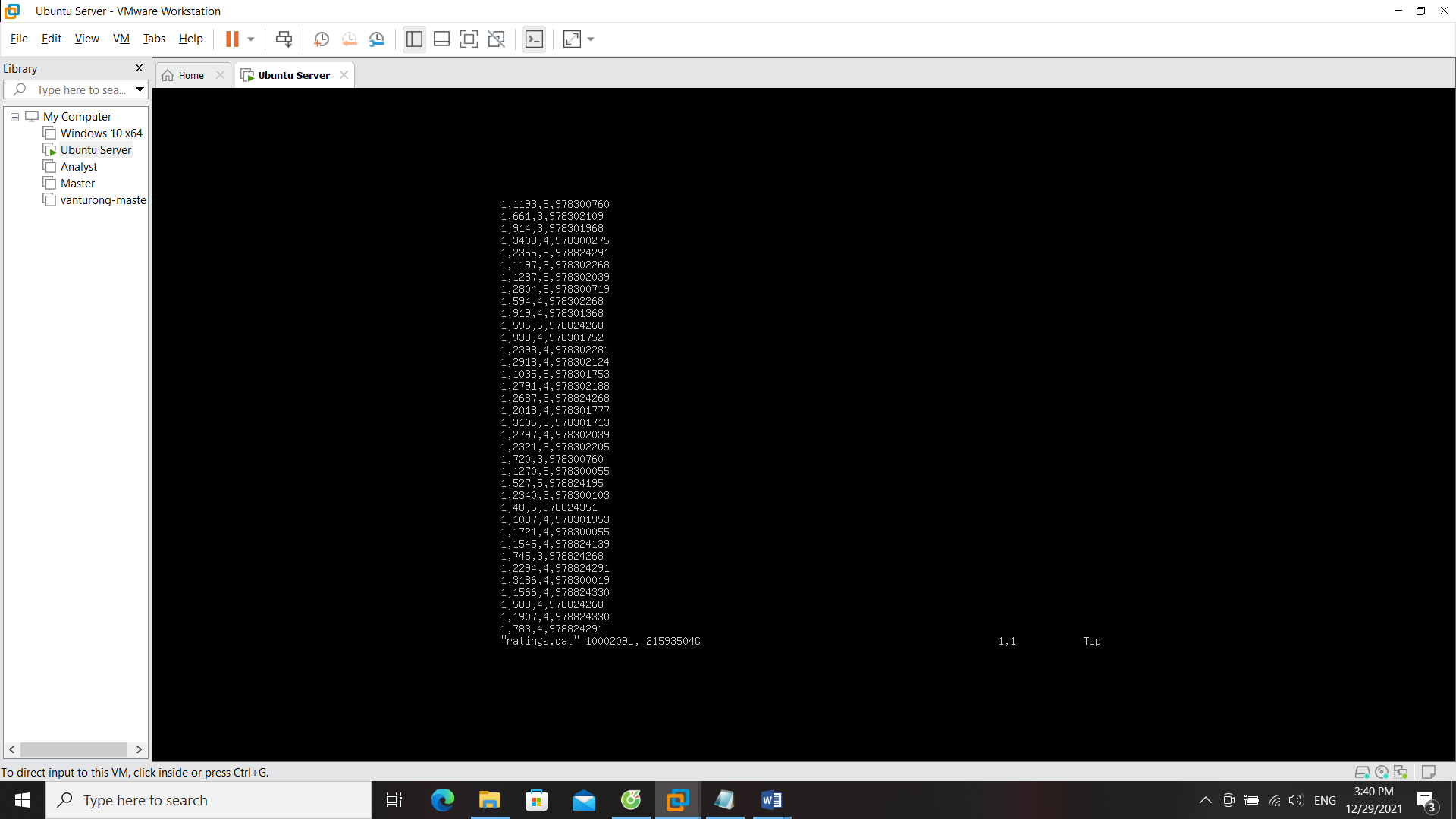
* User: có 6040 dòng, gồm các thông tin như id, gender, age, occupation, zipcode



* Movie: có 3883 dòng, gồm id, title, genres



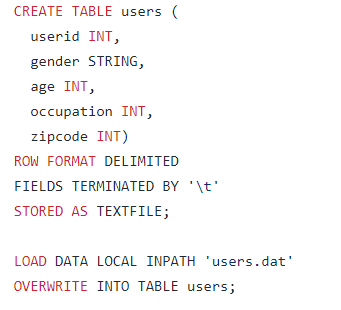
* Rating: có 1000209 dòng, gồm userid, movieid, rating, timestamp

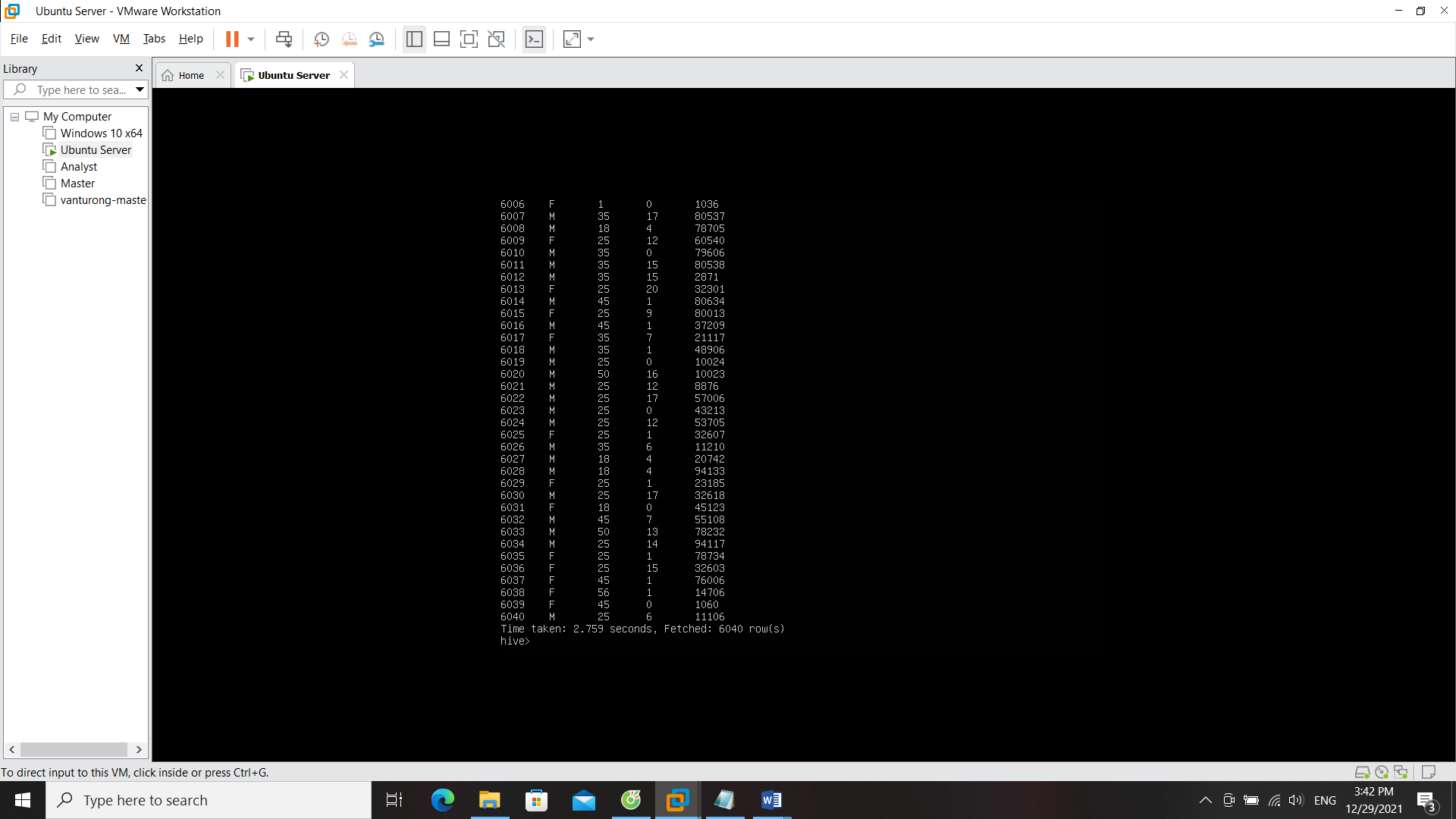


1. Mô tả project

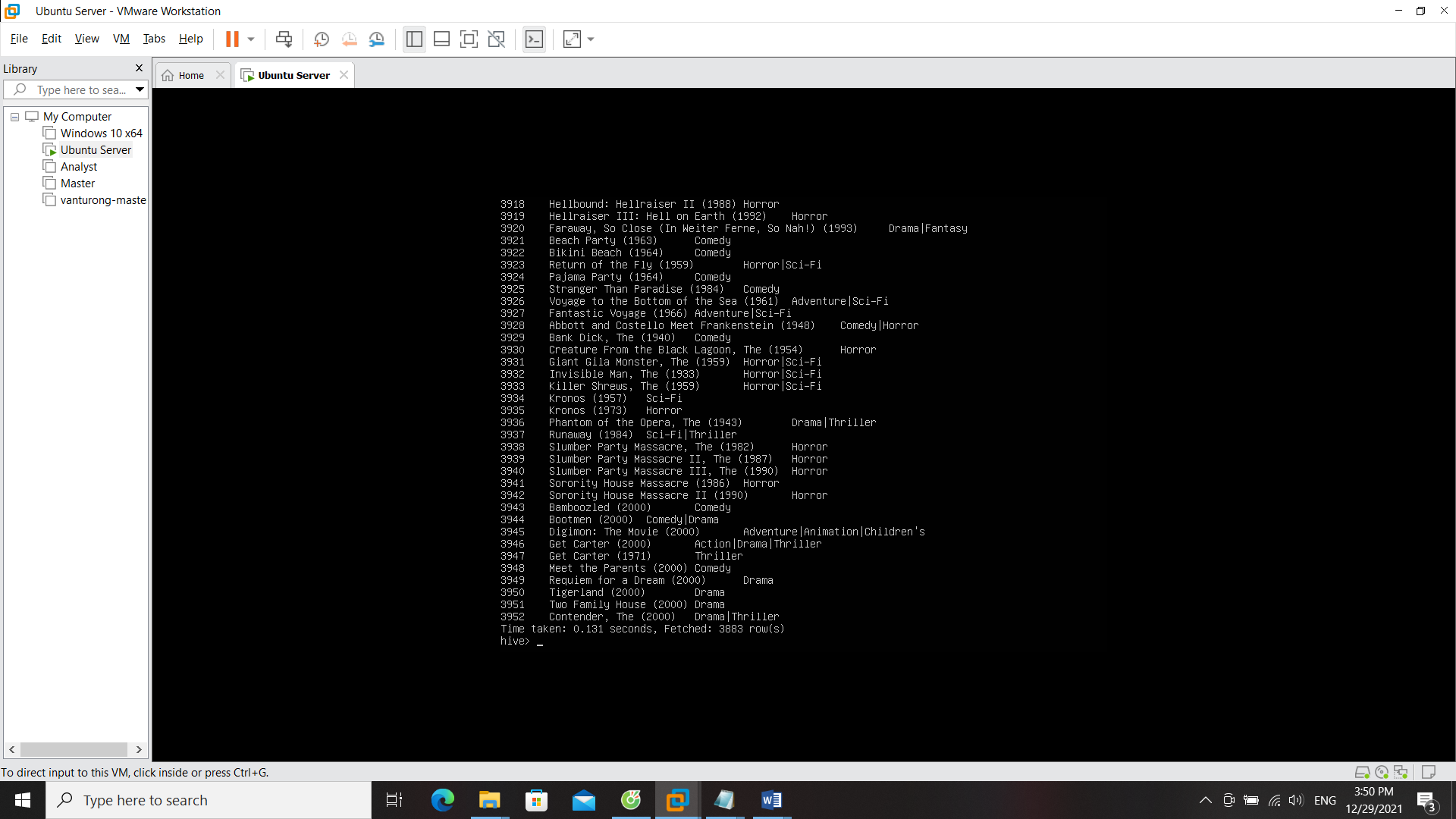
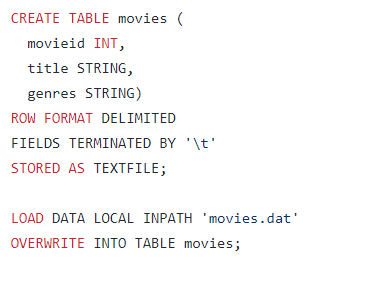
Đầu tiên, nhóm trích xuất dữ liệu từ tập dữ liệu thô thông qua Hive.

* Bảng User

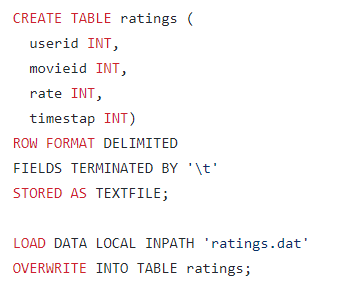


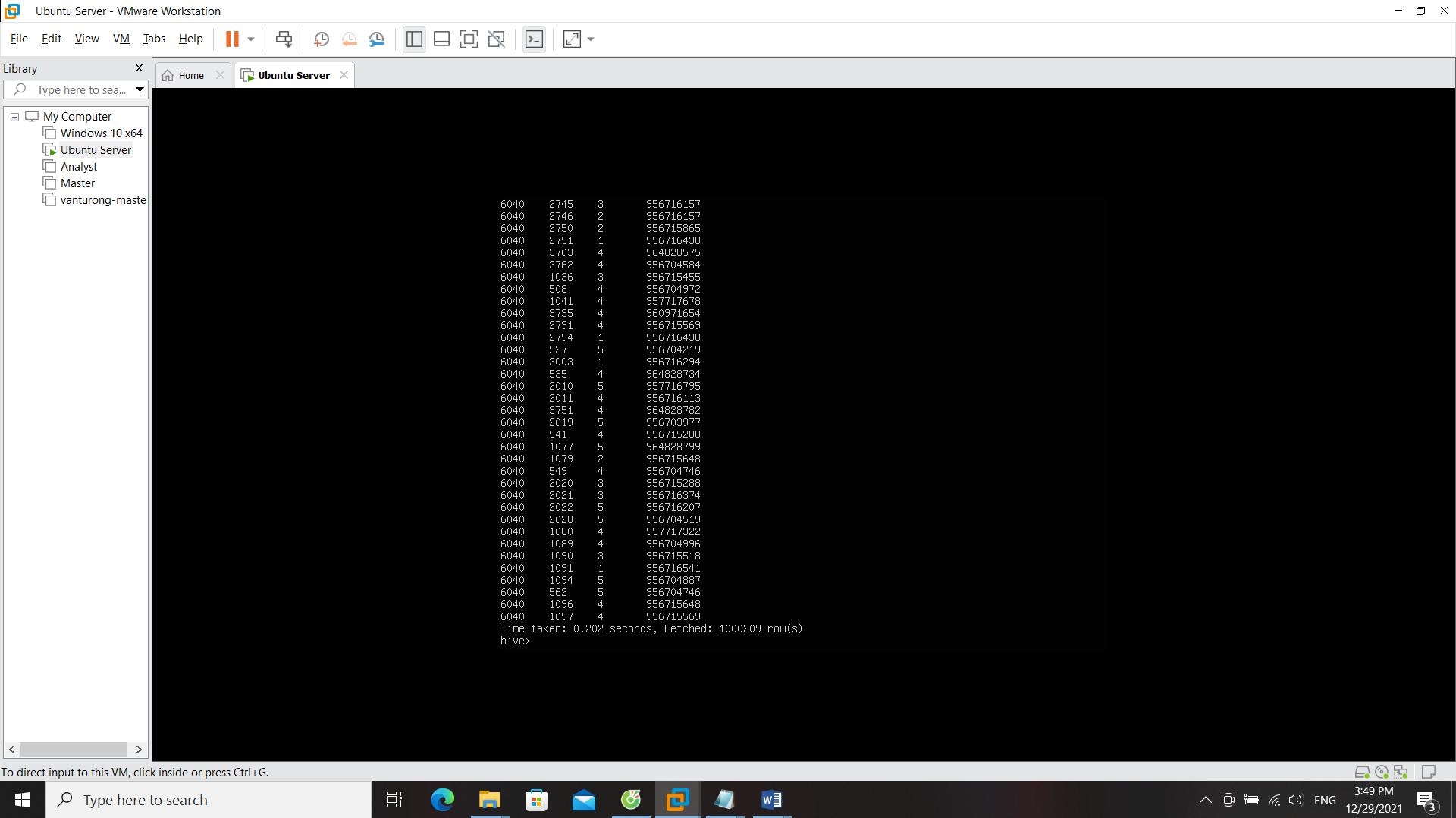


* Bảng Movie

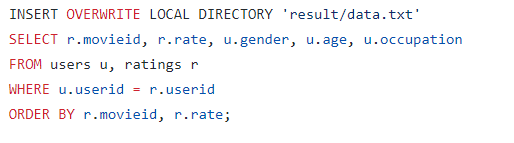


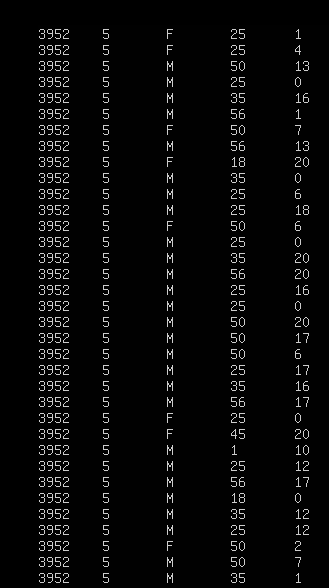
* Bảng Rating





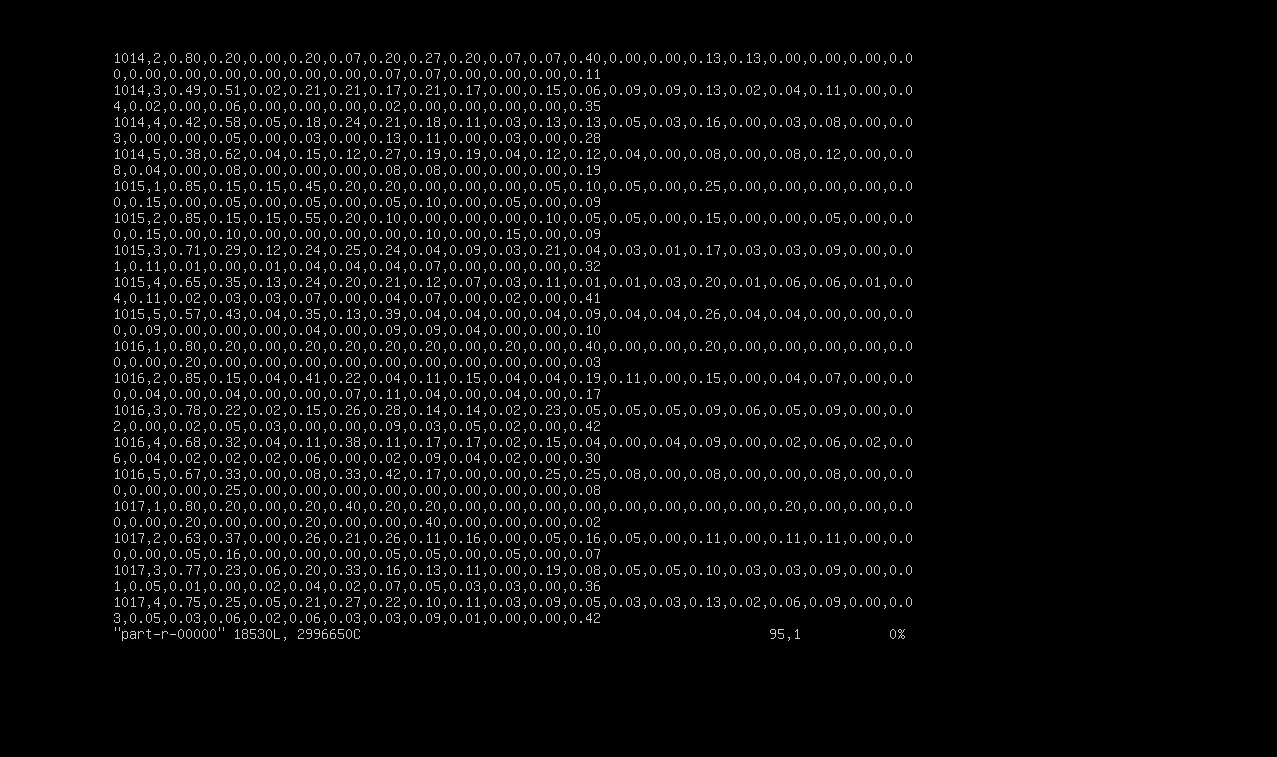
* Chọn các cột như movieid, rate, gender, age, occupation để dùng làm dữ liệu.



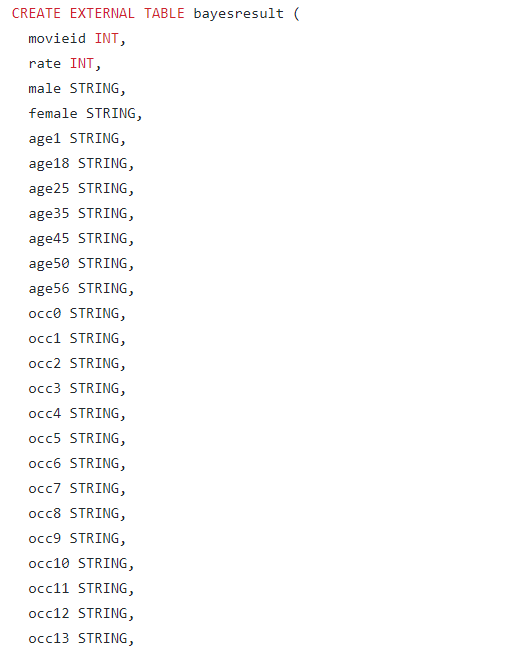


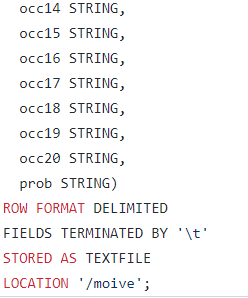
Sau đó, bằng cách áp dụng Naive Bayes, đối với mỗi bộ phim (được phân biệt bằng "Id"), ta phải ước tính nó là P(g = Gi | r = k), P(a = Ai | r = k), P(o = Oi | r = k) và P(r) từ dữ liệu training. Set Gi cho giới tính “Nam” hoặc “Nữ”. Set Ai cho độ tuổi từ “1,18,20…65”. Set Oi cho nghề nghiệp từ “0” đến “20” Set r là cho tỷ lệ phim nằm trong khoảng từ “1” đến “5”. Để làm như vậy, cần tính tổng g = Gi, a = Ai, o = Oi cho mỗi nhãn r trong dữ liệu training để tính P(g = Gi | r = k), P(a = Ai | r = k), P(o = Oi | r = k).

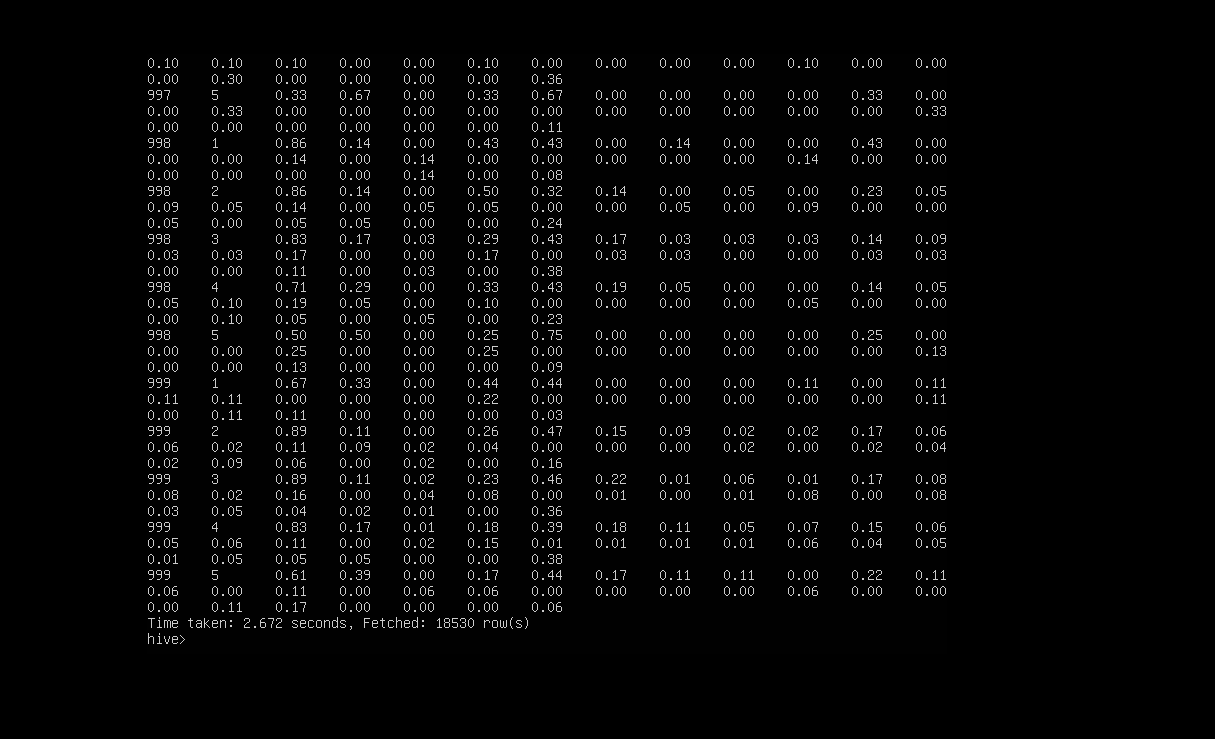
hadoop jar ./Bayes.jar hw6.MultiMovieRecommender /hadoop/ /hadoop/temp /hadoop/output/ chạy file Bayes.jar



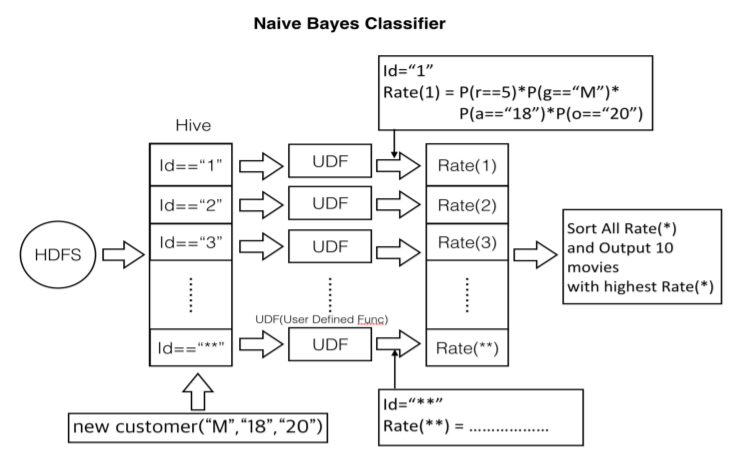
Sau khi tính toán tất cả các khả năng, dữ liệu sẽ được lưu trữ vào HDFS, và sau đó được tải vào Hive để tiếp tục phân loại Naive Bayes để đọc các hoạt động.







Khi một customer mới đến với thông tin cá nhân của họ, ta có thể truy vấn cơ sở dữ liệu Hive bằng cách Mapreduce song song hóa. Để sử dụng kết quả truy vấn và quyết định phim nào phù hợp với sở thích của người dùng nhất, ta đã xác định UDF (Hàm do người dùng xác định) để tính toán xác suất người dùng thích phim. Đây là một trong những chức năng mạnh mẽ nhất của Hive, cho phép triển khai logic nghiệp vụ của riêng họ trong giao diện java. UDF này có thể được gọi khi chúng ta thực hiện truy vấn và lấy kết quả truy vấn làm tham số. Quan trọng hơn, quá trình này cũng được thực hiện thông qua Hadoop Mapreduce.



1. Hướng phát triển.

Kết hợp với hệ thống website để phát triển một hệ thống đề xuất phim hoàn chỉnh

Tài liệu tham khảo

<https://data-flair.training/blogs/hadoop-ecosystem-components/>

<https://viblo.asia/p/gioi-thieu-ve-hive-4P856kvaKY3>

<https://www.tutorialspoint.com/hive/hive_installation.htm>

<https://www.tutorialspoint.com/hive/hive_create_table.htm>