Big Data Course

Capstone Project   
Final Report

For students (instructor review required)

ⓒ2023 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of this document.

This document is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this document other than the curriculum of Samsung Innovation Campus, you must receive written consent from copyright holder.

| Airline Dataset Analysis using Hadoop, Hive, Spark and Tableau |
| --- |

18/08/2024

Team Name: DHT

Trần Văn Tiến (Team leader)

Trương Thị Thuỳ Dung

Đặng Xuân Huyền

Content

**[1. Introduction 4](#_heading=h.gjdgxs)**

**[1.1. Background Information 4](#_heading=h.30j0zll)**

**[1.2. Motivation and Objective 4](#_heading=h.1fob9te)**

**[1.3. Members and Role Assignments 4](#_heading=h.3znysh7)**

**[1.4. Schedule and Milestones 5](#_heading=h.2et92p0)**

**[2. Project Execution 8](#_heading=h.tyjcwt)**

**[2.1. Simulated Scenario Description 8](#_heading=h.3dy6vkm)**

**[2.2. Datasets Selection and Description 10](#_heading=h.1t3h5sf)**

**[2.3. Data Ingestion Pipeline 12](#_heading=h.4d34og8)**

**[2.4. Data Transformation Processing 15](#_heading=h.2s8eyo1)**

**[2.5. Data Query and Insight 16](#_heading=h.17dp8vu)**

**[3. Results 17](#_heading=h.3rdcrjn)**

**[3.1 Đưa các file chứa dữ liệu được lưu trữ tại cục bộ lên HDFS 17](#_heading=h.26in1rg)**

**[3.2 Tạo cơ sở dữ liệu trong hive. Dữ liệu được lấy từ các file đã được đẩy lên HDFS 20](#_heading=h.lnxbz9)**

**[3.3. Tải dữ liệu vào các bảng từ các file csv đã được lưu trên HDFS 25](#_heading=h.1ksv4uv)**

**[3.4. Tiến hành quá trình ETL dùng Apache Spark 29](#_heading=h.44sinio)**

**[4. Projected Impact 42](#_heading=h.z337ya)**

**[4.1. Accomplishments and Benefits 42](#_heading=h.3j2qqm3)**

**[4.2. Future Improvements 42](#_heading=h.1y810tw)**

**[5. Team Member Review and Comment 43](#_heading=h.4i7ojhp)**

[**6. Instructor Review and Comment 43**](#_heading=h.1ci93xb)

# **1. Introduction**

## **1.1. Background Information**

Ngành hàng không là một lĩnh vực phức tạp, đòi hỏi sự quản lý và phân tích dữ liệu khổng lồ để đảm bảo hoạt động hiệu quả và an toàn. Các hãng hàng không, sân bay, và các cơ quan quản lý cần xử lý một lượng lớn dữ liệu hàng ngày, bao gồm thông tin về chuyến bay, hành khách, máy bay, thời tiết, và nhiều yếu tố khác. Việc phân tích chính xác các dữ liệu này có thể giúp cải thiện hiệu suất hoạt động, giảm thiểu chậm trễ, tối ưu hóa lịch trình bay, và nâng cao trải nghiệm của hành khách.

Trong bối cảnh này, việc sử dụng các công cụ Big Data như Hadoop, Hive, Spark và Tableau trở nên cực kỳ quan trọng.

- Hadoop cho phép lưu trữ và xử lý một lượng dữ liệu khổng lồ một cách phân tán, đảm bảo tính sẵn sàng và khả năng mở rộng.

- Hive giúp quản lý và truy vấn dữ liệu một cách có cấu trúc, dễ dàng chuyển đổi dữ liệu thô thành dạng tối ưu cho phân tích.

- Spark cung cấp khả năng xử lý dữ liệu nhanh chóng, cho phép thực hiện các tính toán phức tạp và phân tích dữ liệu theo thời gian thực.

- Tableau là công cụ mạnh mẽ để trực quan hóa dữ liệu, giúp trình bày kết quả phân tích một cách dễ hiểu và tương tác.

Lý do chọn phân tích dữ liệu hàng không sử dụng bộ công cụ này là để tận dụng sức mạnh của Big Data trong việc giải quyết các thách thức lớn của ngành hàng không. Qua đó, giúp cải thiện hiệu quả vận hành, dự đoán và xử lý các vấn đề tiềm ẩn, cũng như tối ưu hóa trải nghiệm của hành khách. Việc này không chỉ có ý nghĩa với các nhà quản lý và điều hành trong ngành hàng không, mà còn mang lại lợi ích kinh tế và xã hội đáng kể..

## **1.2. Motivation and Objective**

Dự án Airline Dataset Analysis được thúc đẩy bởi nhu cầu cấp thiết trong việc cải thiện hiệu suất hoạt động, tối ưu hóa quy trình bay, và nâng cao trải nghiệm khách hàng trong ngành hàng không. Sự chậm trễ, hủy chuyến và các sự cố vận hành không chỉ gây thiệt hại kinh tế lớn mà còn làm suy giảm lòng tin của hành khách. Phân tích dữ liệu chi tiết giúp nhận diện và giảm thiểu các yếu tố gây ra những vấn đề này, từ đó cải thiện hiệu quả hoạt động. Dự án nhằm mục tiêu phân tích và giảm thiểu chậm trễ, tối ưu hóa lịch trình bay, dự đoán và giảm thiểu hủy chuyến, cũng như cải thiện quản lý nguồn lực. Quan trọng hơn, việc nâng cao trải nghiệm khách hàng thông qua phân tích dữ liệu sẽ giúp các hãng hàng không đáp ứng tốt hơn nhu cầu và mong đợi của hành khách, tạo ra dịch vụ chất lượng cao và tăng cường tính cạnh tranh trong thị trường. Nhờ đạt được những mục tiêu này, dự án không chỉ mang lại lợi ích lớn cho các hãng hàng không mà còn tạo ra giá trị tích cực cho hành khách và toàn ngành.

## **1.3. Members and Role Assignments**

- Cả nhóm

· Tìm hiểu tổng quan về đề tài

· Tìm kiếm dữ liệu và xây dựng mô hình dữ liệu

· Xây dựng Action plan cơ bản

- Thành viên 1: Trương Thị Thuỳ Dung

· Tìm hiểu về Hadoop

· Tìm kiếm dữ liệu và xây dựng mô hình dữ liệu

· Trực quan hoá dữ liệu

· Thực hiện power point báo cáo final report

· Thực hiện power point báo cáo action plan

- Thành viên 2: Trần Văn Tiến

· Tìm hiểu về Spark

· Data processing với Spark

· Hiểu với cách ETL của Spark

· Trực quan hoá dữ liệu

· Thực hiện word báo cáo Action plan hoàn chỉnh

- Thành viên 3: Đặng Xuân Huyền

. Tìm hiểu về Hive, Tableau

· EDA cơ bản với Hive

· Hive partitioning and clustering

· Thực hiện word báo cáo final report

## **1.4. Schedule and Milestones**

Bảng hoặc biểu đồ mô tả lịch trình thực hiện và các mốc quan trọng của dự án từ ngày 21/07/2024 đến ngày 10/08/2024, bao gồm các giai đoạn lập kế hoạch, thu thập và hiểu dữ liệu, tiền xử lý và ETL dữ liệu, trực quan hóa và triển khai.

| TIME | ACTIVITY | | | | COMMENT | |
| --- | --- | --- | --- | --- | --- | --- |
| Day 1 | Tổng quan về dự án Capstone | | | | * + Giới thiệu xem sơ bộ về dự án | |
| Bài giảng | | | | * + “Bắt đầu một dự án Big Data” | |
| Tạo đội nhóm | | | | * + Gồm 3 thành viên | |
| Day 2 | Bài giảng | | | | * + “Hướng dẫn dự án Capstone dữ liệu lớn” | |
| Xác định các yêu cầu và mục tiêu cụ thể của dự án. | | | | * + Tìm hiểu về tổng quan về đề tài “Phân tích dữ liệu hang không sử dụng Hadoop, Hive, Spark, Tableau” | |
| Thảo luận và điền mẫu action plan sơ bộ. | | | |
| Day 3 | | Thực hiện Design Thinking: Empathize, Define, Ideate. | * Empathize: Tìm hiểu sâu về người dùng, quan sát và trải nghiệm. * Define: Xác định vấn đề cốt lõi cần giải quyết dựa trên thông tin thu thập được từ giai đoạn Empathize. * Ideate: Sáng tạo nhiều ý tưởng và giải pháp khác nhau để giải quyết vấn đề đã xác định. | | |
| Lập kế hoạch chi tiết cho từng thành viên | * Phân công nhiệm vụ cho từng thành viên cho nhóm | | |
| Day 4 | | Thiết lập môi trường làm việc: cài đặt và cấu hình Hadoop, Hive, Spark. | * Cài đặt cấu hình cho từng dịch vụ | | |
| Thực hành các lệnh cơ bản của Unix. | * Làm quen với hệ thống và quản lý tệp tin, thư mục, quyền truy cập thông qua terminal. | | |
| Day 5 | | Hoàn thiện action plan chi tiết. |  | | |
| Day 6 ~ Day 9 | | * Thu thập dữ liệu từ các nguồn khác nhau. Lưu trữ dữ liệu vào HDFS * Xây dựng và hiểu rõ mô hình dữ liệu (star schema) * Nghiên cứu và hiểu rõ về các dịch vụ và công cụ sẽ được sử dụng | * Hệ thống đặt vé, lịch trình bay, dữ liệu hành khách và dữ liệu hoạt động. * Xác định các bảng, quan hệ giữa các bảng và các trường dữ liệu chính * Hadoop, Hive, Spark, Tableau | | |
| Day 10 | | * Nghiên cứu Spark | * Nghiên cứu kiến trúc của Spark và các loại dữ liệu Spark hỗ trợ | | |
| Day 11 | | * Sử dụng Spark để làm sạch và tiền xử lý dữ liệu. | * Loại bỏ các giá trị thiếu, chuẩn hóa dữ liệu và chuyển đổi định dạng dữ liệu. | | |
| * Phản hồi và tổng kết | * Phản hồi của nhóm và sự tổng kết của người hướng dẫn/cố vấn | | |
| Day 12 ~ Day 13 | | * Thực hiện các bước ETL * Tải dữ liệu đã xử lý vào một cơ sở dữ liệu khác trong Hive | | * Trích xuất dữ liệu thô từ Hive. * Kiểm tra và đảm bảo dữ liệu đã được lưu trữ đúng cách | |
| Day 14 ~ Day 15 | | * Sử dụng Hive để thực hiện các truy vấn * Sử dụng các kỹ thuật phân vùng và phân cụm trong Hive | | * Truy vấn SQL cơ bản nhằm khám phá và hiểu rõ dữ liệu (EDA cơ bản). * Tối ưu hóa truy vấn và cải thiện hiệu suất lưu trữ. | |
| Day 16 | | * Sử dụng Tableau để tạo các biểu đồ | | * Tạo các biểu đồ, báo cáo và dashboard trực quan từ kết quả phân tích. | |
| Day 17 | | * Thực hiện điều phối các công việc của pipeline | | * Lập kế hoạch và triển khai các bước xử lý dữ liệu theo trình tự hợp lý. | |
| Day 18 | | Báo cáo dự án Capstone | | * Sử dụng mẫu được cung cấp | |
| Day 19 | | Chuẩn bị bài thuyết trình cuối cùng | | * Trang trình bày + dữ liệu + mã * Trang trình bày ở định dạng PowerPoint, PDF hoặc Word * Mẫu WBS của dự án Capstone | |
|  | | Tham khảo ý kiến của giáo viên hướng dẫn | | * Điều chỉnh theo sự góp ý của giáo viên hướng dẫn | |
| Day 20 | | Báo cáo dự án | | * Sử dụng máy chiếu * 20 phút thuyết trình + 10 phút hỏi đáp cho mỗi đội | |

# **2. Project Execution**

## **2.1. Simulated Scenario Description**

Trong dự án này, nhóm mô phỏng một kịch bản thực tế của một hãng hàng không quốc tế có mạng lưới đường bay rộng khắp và lượng hành khách lớn. Phân tích dữ liệu hãng hàng không bao gồm việc trích xuất những thông tin và mẫu có ý nghĩa từ lượng dữ liệu khổng lồ được thu thập trong quá trình hoạt động của các hãng hàng không. Phân tích này giúp các hãng hàng không đưa ra quyết định thông minh, tối ưu hóa quy trình, nâng cao hiệu quả, cải thiện trải nghiệm khách hàng và cuối cùng là tăng lợi nhuận. Dưới đây là một số lĩnh vực chính mà phân tích dữ liệu đóng vai trò quan trọng trong ngành hàng không:

Phân tích hiệu suất chuyến bay:Các hãng hàng không phân tích dữ liệu liên quan đến lịch trình chuyến bay, hiệu suất đúng giờ, trì hoãn và hủy bỏ. Phân tích này giúp xác định nguyên nhân gốc rễ của các sự cố trì hoãn, đánh giá hiệu quả của các tuyến bay và điều chỉnh để cải thiện hiệu suất tổng thể.

Dự báo nhu cầu: Bằng cách phân tích dữ liệu đặt chỗ lịch sử, các hãng hàng không có thể dự đoán nhu cầu hành khách cho các tuyến đường và thời gian khác nhau. Dự báo nhu cầu chính xác giúp tối ưu hóa lịch trình chuyến bay, đặt giá vé và phân bổ tài nguyên hiệu quả.

Lợi nhuận tuyến đường: Phân tích dữ liệu cho phép các hãng hàng không đánh giá lợi nhuận của các tuyến đường khác nhau bằng cách xem xét các yếu tố như tải hành khách, chi phí vận hành và giá vé. Thông tin này giúp các hãng hàng không đưa ra quyết định thông minh về việc mở rộng hoặc giảm tuyến đường.

Quản lý doanh thu: Các hãng hàng không sử dụng phân tích dữ liệu để triển khai các chiến lược quản lý doanh thu hiệu quả. Bằng cách điều chỉnh giá vé linh hoạt dựa trên nhu cầu và mẫu đặt chỗ, các hãng hàng không có thể tối đa hóa doanh thu và lấp đầy chỗ ngồi có sẵn hiệu quả hơn.

Hiệu suất và lịch trình phi hành đoàn: Phân tích dữ liệu giúp đánh giá hiệu suất phi hành đoàn, theo dõi giờ làm việc của họ và tối ưu hóa lịch trình phi hành đoàn. Điều này đảm bảo tuân thủ quy định, cải thiện sự hài lòng của phi hành đoàn và giảm thiểu sự không hiệu quả trong hoạt động.

Dự báo bảo trì: Các hãng hàng không phân tích dữ liệu bảo trì để dự đoán các vấn đề tiềm ẩn với các thành phần máy bay. Phương pháp tiếp cận chủ động này giúp ngăn chặn bảo trì không dự kiến và giảm thời gian ngừng hoạt động của máy bay.

Hiệu suất hoạt động: Phân tích dữ liệu được sử dụng để đánh giá hiệu quả của các quy trình hoạt động khác nhau, chẳng hạn như xử lý hành lý, quy trình lên máy bay và thời gian quay vòng. Việc xác định các nút thắt và khu vực cần cải thiện có thể dẫn đến hoạt động trơn tru hơn.

Để thực hiện các phân tích này, các hãng hàng không sử dụng các công cụ phân tích dữ liệu, thuật toán học máy và các công nghệ tiên tiến khác. Hadoop và HDFS cung cấp một khung lưu trữ và xử lý phân tán, cho phép xử lý các bộ dữ liệu lớn. Hive và Spark phục vụ như các ngôn ngữ truy vấn và công cụ xử lý dữ liệu, đơn giản hóa các tác vụ thao tác dữ liệu. Tableau cung cấp các giải pháp trực quan hóa dữ liệu để trình bày kết quả phân tích một cách trực quan và dễ hiểu.

Bằng cách sử dụng các công nghệ này, dự án nhằm trình bày cách các hãng hàng không có thể tận dụng các giải pháp dữ liệu lớn để trích xuất các thông tin có giá trị, tối ưu hóa hoạt động và cải thiện hiệu suất tổng thể trong ngành. Điều quan trọng là các hãng hàng không phải có các thực hành quản trị dữ liệu mạnh mẽ để đảm bảo chất lượng, an toàn và tuân thủ các quy định liên quan.

## **2.2. Datasets Selection and Description**

Nhóm đã chọn 4 tập dữ liệu chính để thực hiện phân tích trong dự án này:

***2.2.1. airports.csv***

- Thuộc tính chính:

· iata: Mã sân bay (IATA)

· airport: Tên sân bay

· city: Thành phố nơi sân bay tọa lạc

· state: Bang (nếu có)

· country: Quốc gia

· lat: Vĩ độ· longitude: Kinh độ

- Mô tả: Dữ liệu này cung cấp thông tin về các sân bay, giúp phân tích các chuyến bay dựa trên vị trí địa lý của sân bay.

***2.2.2. carriers.csv***

- Thuộc tính chính:

· Code: Mã của hãng hàng không

· Description: Tên của hãng hàng không

- Mô tả: Dữ liệu này chứa thông tin về các hãng hàng không, giúp xác định và phân tích hiệu suất của các hãng hàng không khác nhau.

***2.2.3. detailed\_data.csv***

- Thuộc tính chính:

· year: Năm

· month: Tháng

· dayofmonth: Ngày của tháng

· dayofweek: Ngày của tuần

· deptime: Giờ khởi hành (thực tế)

· crsdeptime: Giờ khởi hành theo lịch trình

· arrtime: Giờ đến (thực tế)

· crsarrtime: Giờ đến theo lịch trình

· uniquecarrier: Mã hãng hàng không duy nhất

· flightnum: Số hiệu chuyến bay

· tailnum: Số hiệu đuôi máy bay

· actualelapsedtime: Thời gian bay thực tế

· crselapsedtime: Thời gian bay theo lịch trình

· airtime: Thời gian bay (trong không khí)

· arrdelay: Độ trễ khi đến

· depdelay: Độ trễ khi khởi hành

· origin: Sân bay khởi hành

· dest: Sân bay đến

· distance: Khoảng cách bay

· taxiin: Thời gian taxi vào

· taxiout: Thời gian taxi ra

· cancelled: Trạng thái hủy chuyến

· cancellationcode: Mã hủy chuyến

· diverted: Trạng thái chuyển hướng

· carrierdelay: Độ trễ của hãng

· weatherdelay: Độ trễ do thời tiết

· nasdelay: Độ trễ do hệ thống không lưu quốc gia (NAS)

· securitydelay: Độ trễ do an ninh

· lateaircraftdelay: Độ trễ do máy bay đến muộn

- Mô tả: Dữ liệu này chứa thông tin chi tiết về các chuyến bay, giúp phân tích hiệu suất và xu hướng của các chuyến bay, bao gồm độ trễ, hủy chuyến, và chuyển hướng.

***2.2.4. plane-data.csv***

- Thuộc tính chính:

· tailnum: Số hiệu đuôi máy bay (tail number)

· type: Loại chủ sở hữu (ví dụ: Corporation)

· manufacturer: Nhà sản xuất máy bay (ví dụ: EMBRAER, AIRBUS INDUSTRIE, BOEING)

· issue\_date: Ngày cấp giấy chứng nhận (issue date)

· model: Mẫu máy bay (model) (ví dụ: EMB-145XR, A320-214, 737-3TO)

· status: Trạng thái của giấy chứng nhận (status) (ví dụ: Valid)

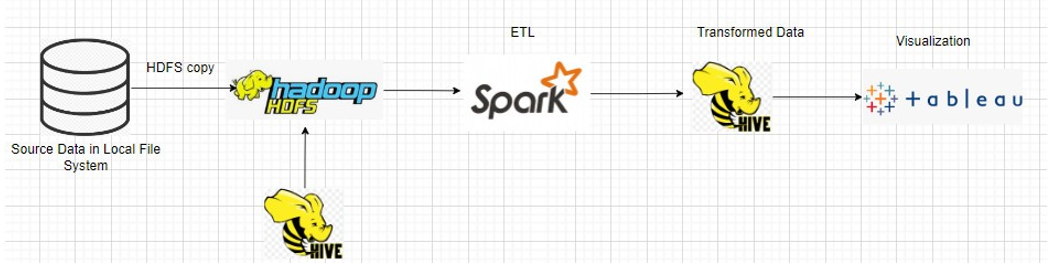
· aircraft\_type: Loại máy bay (aircraft type) (ví dụ: Fixed Wing Multi-Engine)

· engine\_type: Loại động cơ (engine type) (ví dụ: Turbo-Fan, Turbo-Jet)

· year: Năm sản xuất (year)

- Mô tả: Dữ liệu này cung cấp thông tin về các máy bay, giúp theo dõi và phân tích hiệu suất của từng máy bay dựa trên nhà sản xuất, mẫu máy bay, và loại động cơ.

## **2.3. Data Ingestion Pipeline**



Pipeline bao gồm các bước chính sau:

1. Thu thập và lưu trữ dữ liệu

2. Sao chép dữ liệu vào HDFS

3. Quản lý dữ liệu với Hive

4. Xử lý dữ liệu với Spark

5. Quản lý dữ liệu và tối ưu hóa

6. Trực quan hóa dữ liệu với Tableau

### *2.3.1 Thu thập và lưu trữ dữ liệu*

Dữ liệu hàng không sẽ được thu thập từ các nguồn dữ liệu công khai và từ các hãng hàng không. Dữ liệu bao gồm 4 file CSV chính:

· airports.csv: Tập dữ liệu này chứa thông tin về các sân bay.

· carriers.csv: Tập dữ liệu này chứa thông tin về các hãng hàng không.

· detailed\_data.csv: Tập dữ liệu này chứa thông tin chi tiết về các chuyến bay.

· plane-data.csv: Tập dữ liệu này chứa thông tin về các chuyến bay.

Các tệp CSV này sẽ được lưu trữ trong hệ thống tệp cục bộ (Local File System) trước khi được xử lý và chuyển đến các hệ thống quản lý dữ liệu phân tán.

2.3.2 Sao chép dữ liệu vào HDFS

Dữ liệu từ hệ thống tệp cục bộ được sao chép vào Hệ thống Tệp Phân Tán Hadoop (HDFS) để lưu trữ và quản lý phân tán. HDFS cung cấp khả năng lưu trữ dữ liệu lớn và hỗ trợ truy cập nhanh chóng đến dữ liệu.

### Lợi ích của HDFS:

· Khả năng lưu trữ lớn: HDFS có thể lưu trữ hàng petabyte dữ liệu, cho phép lưu trữ dữ liệu lớn từ nhiều nguồn khác nhau.

· Quản lý dữ liệu phân tán: HDFS tự động sao chép dữ liệu trên nhiều nút, đảm bảo tính sẵn sàng cao và khả năng chịu lỗi.

· Truy cập nhanh chóng: HDFS cung cấp khả năng truy cập dữ liệu nhanh chóng thông qua hệ thống phân tán, giảm thiểu thời gian truy vấn và xử lý dữ liệu.

Dữ liệu được sao chép vào HDFS sử dụng lệnh hadoop -put -f

***2.3.3 Quản lý dữ liệu với Hive***

Dữ liệu được nhập vào Hive, nơi nó được quản lý và tổ chức dưới dạng các bảng để dễ dàng truy vấn và thao tác. Hive cho phép tạo các bảng quản lý dữ liệu (Hive-managed tables) và các bảng ngoại vi (Hive-external tables) tùy thuộc vào cách dữ liệu được quản lý và lưu trữ. Điều này giúp tối ưu hóa việc lưu trữ và truy vấn dữ liệu trong Hive.

***2.3.3.1 Xây dựng mô hình dữ liệu***

Mô hình dữ liệu được xây dựng theo star schema bao gồm một bảng chính flight\_facts và bốn bảng dimension (dim). Bảng flight\_facts chứa các thông tin chi tiết về các chuyến bay, trong khi các bảng dimension chứa các thông tin phụ trợ giúp mô tả chi tiết hơn về dữ liệu trong bảng chính.

· Bảng flight\_facts: Chứa thông tin về chuyến bay như tháng, số hiệu chuyến bay, số đuôi máy bay, hãng hàng không, độ trễ trung bình khi đến và khi khởi hành, khoảng cách trung bình, tổng số chuyến bay bị hủy và bị chuyển hướng, tên hãng hàng không, nhà sản xuất máy bay, mã số đúng giờ và ngày giờ tải dữ liệu.

· Bảng carriers: Chứa thông tin về các hãng hàng không như mã số hãng hàng không và tên hãng hàng không.

· Bảng plane\_data: Chứa thông tin về các loại máy bay như số đuôi máy bay, nhà sản xuất, loại máy bay.

· Bảng airports: Chứa thông tin về các sân bay như mã số sân bay, tên sân bay, thành phố và quốc gia.

· Bảng detailed\_data: Chứa thông tin về thời gian như năm, tháng, ngày, giờ và thông tin về các chuyến bay.

Mô hình star schema giúp tối ưu hóa truy vấn bằng cách tách dữ liệu chính và dữ liệu mô tả, đồng thời giúp dễ dàng mở rộng và bảo trì dữ liệu.

### *2.3.4 Xử lý dữ liệu với Spark*

Spark được sử dụng để thực hiện quy trình ETL (Extract, Transform, Load). Quy trình này bao gồm việc trích xuất dữ liệu từ Hive, thực hiện các bước làm sạch và chuyển đổi dữ liệu, và sau đó lưu dữ liệu đã chuyển đổi trở lại Hive.

#### *2.3.4.1. Tiền xử lý dữ liệu*

Làm sạch và tiền xử lý dữ liệu là bước quan trọng trong quy trình xử lý dữ liệu. Các bước này bao gồm:

· Loại bỏ các giá trị thiếu: Xử lý các giá trị thiếu hoặc null trong dữ liệu để đảm bảo tính chính xác và đầy đủ.

· Chuẩn hóa dữ liệu: Chuyển đổi dữ liệu về cùng một định dạng và chuẩn hóa các giá trị để đảm bảo tính nhất quán.

· Chuyển đổi định dạng dữ liệu: Chuyển đổi dữ liệu từ định dạng ban đầu sang định dạng phù hợp với yêu cầu phân tích, chẳng hạn như từ CSV sang Parquet.

#### *2.3.4.2. ETL dữ liệu*

Thực hiện quy trình ETL bằng Spark giúp tối ưu hóa việc trích xuất, chuyển đổi và tải dữ liệu. Quy trình ETL bao gồm các bước sau:

· Trích xuất dữ liệu từ Hive: Sử dụng Spark SQL để truy vấn và trích xuất dữ liệu từ các bảng trong Hive.

· Chuyển đổi dữ liệu: Thực hiện các bước chuyển đổi dữ liệu như làm sạch, chuẩn hóa, tính toán các chỉ số và tổng hợp dữ liệu.

· Tải dữ liệu vào Hive: Lưu dữ liệu đã chuyển đổi trở lại Hive vào một database khác hoặc các bảng khác để dễ dàng truy vấn và phân tích.

Spark hỗ trợ xử lý dữ liệu song song và phân tán, giúp tăng tốc độ xử lý và tối ưu hóa hiệu suất.

### *2.3.5. Quản lý dữ liệu và tối ưu hóa*

#### *2.3.5.1. Sử dụng phân vùng*

Phân vùng dữ liệu: Giúp giảm thiểu thời gian truy vấn bằng cách chỉ truy vấn những phần dữ liệu cần thiết. Dữ liệu được phân vùng theo các tiêu chí như năm, tháng hoặc ngày.

#### *2.3.5.2. Quản lý dữ liệu với Hive*

Hive quản lý dữ liệu dưới dạng các bảng, cho phép người dùng truy vấn dữ liệu sử dụng ngôn ngữ SQL. Hive-managed tables và Hive-external tables cung cấp hai cách khác nhau để quản lý và lưu trữ dữ liệu, tùy thuộc vào nhu cầu cụ thể của người dùng.

· Hive-managed tables: Hive quản lý hoàn toàn dữ liệu và các file tương ứng. Khi bảng bị xóa, dữ liệu cũng sẽ bị xóa khỏi HDFS.

· Hive-external tables: Hive chỉ quản lý metadata, còn dữ liệu và các file tương ứng được quản lý bên ngoài Hive. Khi bảng bị xóa, dữ liệu vẫn được giữ lại trong HDFS.

### *2.3.6. Trực quan hóa dữ liệu với Tableau*

Dữ liệu đã chuyển đổi trong Hive được truy xuất và nhập vào Tableau. Tableau được sử dụng để tạo các biểu đồ, báo cáo, và dashboard trực quan, giúp trình bày kết quả phân tích một cách rõ ràng và dễ hiểu.

· Kết nối Tableau với Hive: Sử dụng các driver như ODBC hoặc JDBC để kết nối Tableau với Hive và truy xuất dữ liệu.

· Tạo biểu đồ và báo cáo: Tableau cung cấp các công cụ mạnh mẽ để tạo các biểu đồ và báo cáo trực quan, giúp người dùng dễ dàng hiểu và phân tích các xu hướng và mô hình trong dữ liệu.

· Tạo dashboard: Kết hợp nhiều biểu đồ và báo cáo vào một dashboard tổng hợp, giúp trình bày thông tin chi tiết và toàn diện về dữ liệu hàng không.

## **2.4. Data Transformation Processing**

Quy trình xử lý biến đổi dữ liệu (ETL - Extract, Transform, Load) là một phần quan trọng trong quản lý dữ liệu, đặc biệt trong các hệ thống dữ liệu lớn như kho dữ liệu (Data Warehouse). Quy trình này bao gồm ba bước chính: trích xuất dữ liệu từ các nguồn khác nhau, biến đổi dữ liệu theo yêu cầu, và tải dữ liệu vào hệ thống lưu trữ đích.

***2.4. Khởi tạo SparkSession:***

Quy trình bắt đầu bằng việc khởi tạo SparkSession, đây là điểm khởi đầu cho các ứng dụng Spark. SparkSession cung cấp một giao diện để kết nối với các nguồn dữ liệu và thực hiện các thao tác xử lý dữ liệu. Việc khởi tạo SparkSession với các cấu hình phù hợp giúp truy cập và xử lý dữ liệu từ Hive một cách hiệu quả. SparkSession không chỉ hỗ trợ việc kết nối mà còn cung cấp các công cụ và API để dễ dàng thao tác, biến đổi, và lưu trữ dữ liệu.

***2.4.2. Đọc dữ liệu từ các bảng nguồn:***

Dữ liệu được trích xuất từ các bảng nguồn trong cơ sở dữ liệu Hive. Các bảng này có thể chứa dữ liệu thô từ nhiều nguồn khác nhau, bao gồm thông tin về sân bay, hãng hàng không, máy bay và chi tiết các chuyến bay. Việc trích xuất dữ liệu này là bước đầu tiên để thu thập dữ liệu cần thiết cho các bước biến đổi tiếp theo. Quá trình này đòi hỏi sự tương tác với Hive để truy vấn và tải dữ liệu vào bộ nhớ của Spark, tạo ra các DataFrame để tiếp tục xử lý.

***2.4.3. Biến đổi dữ liệu:***

Biến đổi dữ liệu là bước quan trọng trong quy trình ETL, nơi các thao tác cần thiết được thực hiện để chuẩn bị dữ liệu cho phân tích. Các thao tác biến đổi dữ liệu có thể bao gồm:

Thêm các cột mới: Thêm các cột như ngày và thời gian tải dữ liệu ('load\_dt' và 'load\_dtm') để theo dõi thời điểm dữ liệu được tải. Điều này giúp cung cấp thông tin về tính cập nhật của dữ liệu và hỗ trợ các hoạt động quản lý dữ liệu sau này.

Chuyển đổi kiểu dữ liệu: Đảm bảo rằng các cột dữ liệu có kiểu dữ liệu phù hợp để dễ dàng xử lý và phân tích. Việc chuyển đổi kiểu dữ liệu giúp tránh các lỗi và đảm bảo dữ liệu có thể được sử dụng hiệu quả trong các tính toán và phân tích.

Lọc và chọn các cột cần thiết: Giảm bớt kích thước dữ liệu bằng cách chỉ giữ lại những cột thực sự cần thiết cho phân tích. Việc này giúp tối ưu hóa hiệu suất và giảm dung lượng lưu trữ.

Nhóm và tính toán thống kê: Nhóm dữ liệu theo các tiêu chí nhất định và tính toán các giá trị thống kê như độ trễ trung bình, khoảng cách trung bình, tổng số chuyến bay bị hủy, v.v. Đây là bước quan trọng để tạo ra các thông tin tổng hợp từ dữ liệu thô, giúp phân tích và ra quyết định.

***2.4.4. Lưu dữ liệu:***

Dữ liệu đã được biến đổi sẽ được lưu trữ vào hai nơi:

HDFS (Hadoop Distributed File System): Dữ liệu được lưu dưới định dạng Parquet trong các thư mục tương ứng. Định dạng Parquet giúp lưu trữ dữ liệu một cách hiệu quả và tối ưu cho các tác vụ đọc ghi. Việc lưu trữ trên HDFS đảm bảo rằng dữ liệu có thể được truy cập và xử lý song song bởi nhiều nút trong hệ thống.

Hive: Dữ liệu cũng được lưu vào các bảng trong cơ sở dữ liệu Hive, giúp dễ dàng truy vấn và tích hợp với các công cụ phân tích dữ liệu khác. Hive cung cấp một giao diện SQL để truy vấn dữ liệu lớn, cho phép tích hợp dễ dàng với các công cụ BI và phân tích khác.

***2.4.5. Kết thúc SparkSession:***

Sau khi hoàn tất các bước xử lý và lưu trữ dữ liệu, SparkSession được dừng lại để giải phóng tài nguyên hệ thống. Việc quản lý tài nguyên hiệu quả giúp đảm bảo rằng hệ thống luôn sẵn sàng cho các tác vụ xử lý dữ liệu tiếp theo. Dừng SparkSession đúng cách giúp ngăn chặn rò rỉ tài nguyên và tối ưu hóa hiệu suất của hệ thống.

## **2.5. Data Query and Insight**

Sau khi dữ liệu đã được chuyển đổi, nó sẽ được tải vào Apache Hive để truy vấn và phân tích. Hive cung cấp một nền tảng cho phép thực hiện các truy vấn tương tự SQL trên dữ liệu có cấu trúc, giúp trích xuất những thông tin có giá trị về hiệu suất và mức độ sử dụng của máy chủ web. Các truy vấn được thiết kế để khám phá các chỉ số chính như khối lượng lưu lượng, thời gian sử dụng cao điểm, phân phối mã phản hồi, và thời gian phản hồi trung bình.Quá trình này bao gồm việc khám phá và hiểu dữ liệu thông qua các truy vấn SQL cơ bản và sử dụng các kỹ thuật phân vùng trong Hive để tối ưu hóa hiệu suất truy vấn.

#### *2.5.1. Truy vấn Dữ liệu bằng Hive và nhận định*

Hive cho phép chúng ta viết và thực thi các truy vấn giống SQL để khám phá và hiểu dữ liệu. Các truy vấn này giúp phân tích xu hướng, tìm kiếm thông tin cụ thể, và tạo ra các báo cáo tổng hợp. Sau đây là một số truy vấn tiêu biểu:

* Phân tích Độ trễ Trung bình theo Hãng hàng không và Nhà sản xuất Máy bay: Đánh giá độ trễ trung bình của các hãng hàng không khác nhau và xem xét liệu nhà sản xuất máy bay có ảnh hưởng đến độ trễ hay không.
* Đánh giá Tính đúng giờ của Chuyến bay: Kiểm tra tính đúng giờ của các chuyến bay để xác định tỷ lệ chuyến bay đúng giờ so với những chuyến bay bị trễ.
* Phân tích Khoảng cách Bay Tối đa: Xác định các hãng hàng không có khoảng cách bay tối đa lớn nhất và phân tích các chỉ số liên quan đến độ trễ của các chuyến bay.
* Tỷ lệ Chuyến bay Đúng giờ và Không Đúng giờ theo Hãng hàng không và Nhà sản xuất Máy bay: Đánh giá tỷ lệ chuyến bay đúng giờ và không đúng giờ của các hãng hàng không và nhà sản xuất máy bay khác nhau.
* Phân tích Tỷ lệ Hủy chuyến Bay: Đánh giá tỷ lệ hủy chuyến bay theo từng tháng để hiểu rõ hơn về tình hình hoạt động của các chuyến bay.
* Phân tích Tỷ lệ Chuyển hướng Chuyến Bay: Đánh giá tỷ lệ chuyển hướng chuyến bay theo từng tháng.

Những nhận định rút ra từ các truy vấn này được trực quan hóa và báo cáo để cung cấp một cái nhìn tổng quan toàn diện về hiệu suất máy chủ và tương tác của người dùng. Các hình ảnh trực quan giúp dễ dàng truyền đạt các phát hiện và hỗ trợ quyết định dựa trên dữ liệu. Bằng cách hiểu rõ hiệu suất máy chủ và các mẫu sử dụng, các chiến lược tối ưu hóa có thể được thông tin và cải tiến tương lai có thể được lên kế hoạch để nâng cao trải nghiệm người dùng.

#### *2.5.2. Kỹ thuật Phân vùng*

* Tạo Bảng Phân vùng: Phân vùng bảng ‘flight\_facts’theo năm để tối ưu hóa hiệu suất truy vấn. Bằng cách phân vùng dữ liệu theo năm, Hive có thể bỏ qua các phân vùng không liên quan, giảm thời gian xử lý và tăng tốc độ truy vấn.
* Truy vấn Dữ liệu từ Bảng Phân vùng: Truy xuất dữ liệu cho các năm cụ thể mà không cần quét toàn bộ tập dữ liệu, cải thiện hiệu suất truy vấn,
* đặc biệt khi làm việc với các tập dữ liệu lớn.



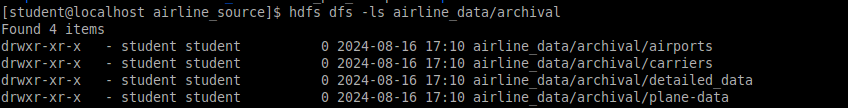
# **3. Results**

## **3.1 Đưa các file chứa dữ liệu được lưu trữ tại cục bộ lên HDFS**

* Các file dữ liệu bao gồm airports.csv, carriers.csv, detailed\_data.csv, plane\_data.csv được lưu trữ trong các thư mục trùng tên với tên của từng file tại thư mục gốc “airline\_source”



* File “hdfs\_put\_data.sh” nhằm mục đích đưa các file dữ liệu vừa nêu trên lên HDFS. Và các file sẽ được lưu trữ tại hdfs://localhost:9000/airline\_data/archival.



* Nội dung của file “hdfs\_put\_data.sh” như sau:

| #!/bin/bash  #hdfs put command  current\_date=$(date +%Y%m%d)  #Copying the existing data to archival  #Checking whether existing file is present or not  airports\_file\_count=$(hadoop fs -count /airline\_data/airports | awk '{print $2}')  carriers\_file\_count=$(hadoop fs -count /airline\_data/carriers | awk '{print $2}')  plane\_file\_count=$(hadoop fs -count /airline\_data/plane-data | awk '{print $2}')  detailed\_data\_file\_count=$(hadoop fs -count /airline\_data/detailed\_data | awk '{print $2}')  if [ $airports\_file\_count -eq 0 ]  then      hadoop fs -put -f airline\_source\_data/airports/airports.csv /airline\_data/airports  else      hadoop fs -mkdir -p airline\_data/archival/airports/$current\_date      hadoop fs -cp -f /airline\_data/airports/airports.csv airline\_data/archival/airports/$current\_date/      hadoop fs -put -f airline\_source\_data/airports/airports.csv /airline\_data/airports  fi  if [ $carriers\_file\_count -eq 0 ]  then      hadoop fs -put -f airline\_source\_data/carriers/carriers.csv /airline\_data/carriers  else      hadoop fs -mkdir -p airline\_data/archival/carriers/$current\_date      hadoop fs -cp -f /airline\_data/carriers/carriers.csv airline\_data/archival/carriers/$current\_date/      hadoop fs -put -f airline\_source\_data/carriers/carriers.csv  /airline\_data/carriers  fi  if [ $plane\_file\_count -eq 0 ]  then      hadoop fs -put -f airline\_source\_data/plane-data/plane-data.csv /airline\_data/plane-data  else      hadoop fs -mkdir -p airline\_data/archival/plane-data/$current\_date      hadoop fs -cp -f /airline\_data/plane-data/plane-data.csv airline\_data/archival/plane-data/$current\_date/      hadoop fs -put -f airline\_source\_data/plane-data/plane-data.csv  /airline\_data/plane-data  fi  if [ $detailed\_data\_file\_count -eq 0 ]  then      hadoop fs -put -f airline\_source\_data/detailed\_data/detailed\_data.csv /airline\_data/detailed\_data  else      hadoop fs -mkdir -p airline\_data/archival/detailed\_data/$current\_date      hadoop fs -cp -f /airline\_data/detailed\_data/detailed\_data.csv airline\_data/archival/detailed\_data/$current\_date/      hadoop fs -put -f airline\_source\_data/detailed\_data/detailed\_data.csv  /airline\_data/detailed\_data  fi |
| --- |

## **3.2 Tạo cơ sở dữ liệu trong hive. Dữ liệu được lấy từ các file đã được đẩy lên HDFS**

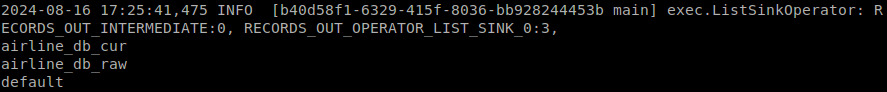
* Khởi chạy Hive CLI bằng cách dùng lệnh **“hive”**



***3.2.1. Tạo 2 cơ sở dữ liệu trong hive***

* Sau khi chạy lệnh “hive”, tiến hành tạo 2 cơ sở dữ liệu có tên lần lượt là **“airline\_db\_raw”** và **“airline\_db\_cur”**. Dùng các lệnh sau:

| create database airline\_db\_raw;  create database airline\_db\_cur; |
| --- |



***3.2.2. Tạo các bảng trong cơ sở dữ liệu “airline\_db\_raw”***

* Trong cơ sở dữ liệu “airline\_db\_raw” tạo 4 bảng sau:
  + airports
  + carriers
  + detailed\_data
  + plane\_data
* Tạo bảng **airports** với code như sau:

| CREATE EXTERNAL TABLE `airports`(  `iata` string COMMENT 'from deserializer',  `airport` string COMMENT 'from deserializer',  `city` string COMMENT 'from deserializer',  `state` string COMMENT 'from deserializer',  `country` string COMMENT 'from deserializer',  `lat` string COMMENT 'from deserializer',  `longitude` string COMMENT 'from deserializer')  ROW FORMAT SERDE  'org.apache.hadoop.hive.serde2.OpenCSVSerde'  WITH SERDEPROPERTIES (  'escapeChar'='\\',  'quoteChar'='\"',  'separatorChar'=',',  'skip.header.line.count'='1')  STORED AS INPUTFORMAT  'org.apache.hadoop.mapred.TextInputFormat'  OUTPUTFORMAT  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'  LOCATION  'hdfs://localhost:9000/airline\_data/archival/airports'; |
| --- |

* Tạo bảng carriers với code như sau:

| CREATE EXTERNAL TABLE `carriers`(  `code` string COMMENT 'from deserializer',  `description` string COMMENT 'from deserializer')  ROW FORMAT SERDE  'org.apache.hadoop.hive.serde2.OpenCSVSerde'  WITH SERDEPROPERTIES (  'escapeChar'='\\',  'quoteChar'='\"',  'separatorChar'=',',  'skip.header.line.count'='1')  STORED AS INPUTFORMAT  'org.apache.hadoop.mapred.TextInputFormat'  OUTPUTFORMAT  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'  LOCATION  'hdfs://localhost:9000/airline\_data/archival/carriers'; |
| --- |

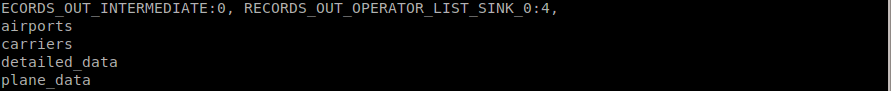
* Tạo bảng detailed\_data với code như sau:

| CREATE EXTERNAL TABLE `detailed\_data`(  `year` string COMMENT 'from deserializer',  `month` string COMMENT 'from deserializer',  `dayofmonth` string COMMENT 'from deserializer',  `dayofweek` string COMMENT 'from deserializer',  `deptime` string COMMENT 'from deserializer',  `crsdeptime` string COMMENT 'from deserializer',  `arrtime` string COMMENT 'from deserializer',  `crsarrtime` string COMMENT 'from deserializer',  `uniquecarrier` string COMMENT 'from deserializer',  `flightnum` string COMMENT 'from deserializer',  `tailnum` string COMMENT 'from deserializer',  `actualelapsedtime` string COMMENT 'from deserializer',  `crselapsedtime` string COMMENT 'from deserializer',  `airtime` string COMMENT 'from deserializer',  `arrdelay` string COMMENT 'from deserializer',  `depdelay` string COMMENT 'from deserializer',  `origin` string COMMENT 'from deserializer',  `dest` string COMMENT 'from deserializer',  `distance` string COMMENT 'from deserializer',  `taxiin` string COMMENT 'from deserializer',  `taxiout` string COMMENT 'from deserializer',  `cancelled` string COMMENT 'from deserializer',  `cancellationcode` string COMMENT 'from deserializer',  `diverted` string COMMENT 'from deserializer',  `carrierdelay` string COMMENT 'from deserializer',  `weatherdelay` string COMMENT 'from deserializer',  `nasdelay` string COMMENT 'from deserializer',  `securitydelay` string COMMENT 'from deserializer',  `lateaircraftdelay` string COMMENT 'from deserializer')  ROW FORMAT SERDE  'org.apache.hadoop.hive.serde2.OpenCSVSerde'  WITH SERDEPROPERTIES (  'escapeChar'='\\',  'quoteChar'='\"',  'separatorChar'=',',  'skip.header.line.count'='1')  STORED AS INPUTFORMAT  'org.apache.hadoop.mapred.TextInputFormat'  OUTPUTFORMAT  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'  LOCATION  'hdfs://localhost:9000/airline\_data/archival/detailed\_data'; |
| --- |

* Tạo bảng planed\_data với code như sau:

| CREATE EXTERNAL TABLE `plane\_data`(  `tailnum` string COMMENT 'from deserializer',  `type` string COMMENT 'from deserializer',  `manufacturer` string COMMENT 'from deserializer',  `issue\_date` string COMMENT 'from deserializer',  `model` string COMMENT 'from deserializer',  `status` string COMMENT 'from deserializer',  `aircraft\_type` string COMMENT 'from deserializer',  `engine\_type` string COMMENT 'from deserializer',  `year` string COMMENT 'from deserializer')  ROW FORMAT SERDE  'org.apache.hadoop.hive.serde2.OpenCSVSerde'  WITH SERDEPROPERTIES (  'escapeChar'='\\',  'quoteChar'='\"',  'separatorChar'=',',  'skip.header.line.count'='1')  STORED AS INPUTFORMAT  'org.apache.hadoop.mapred.TextInputFormat'  OUTPUTFORMAT  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'  LOCATION  'hdfs://localhost:9000/airline\_data/archival/plane-data'; |
| --- |

* Sau khi thực hiện việc tạo bảng trong cơ sở dữ liệu “airline\_db\_raw”, kết quả như hình dưới đây:



***3.2.3. Tạo các bảng trong cơ sở dữ liệu “airline\_db\_******cur”***

* Trong cơ sở dữ liệu “airline\_db\_ cur” tạo 5 bảng sau:
  + airports
  + carriers
  + detailed\_data
  + plane\_data
  + flight\_facts
* Tạo bảng “airports” với code như sau:

| CREATE EXTERNAL TABLE `airports`(  `iata` string,  `airport` string,  `city` string,  `state` string,  `country` string,  `lat` string,  `longitude` string,  `load\_dt` string,  `load\_dtm` string)  ROW FORMAT SERDE  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  STORED AS INPUTFORMAT  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'  OUTPUTFORMAT  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'  LOCATION  'hdfs://localhost:9000/airline\_data/cur/airports'; |
| --- |

* Tạo bảng “carriers” với code như sau:

| CREATE EXTERNAL TABLE `carriers`(  `code` string,  `description` string,  `load\_dt` string,  `load\_dtm` string)  ROW FORMAT SERDE  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  STORED AS INPUTFORMAT  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'  OUTPUTFORMAT  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'  LOCATION  'hdfs://localhost:9000/airline\_data/cur/carriers'; |
| --- |

* Tạo bảng “detailed\_data” với code như sau:

| CREATE EXTERNAL TABLE `detailed\_data`(  `month` string,  `dayofmonth` string,  `dayofweek` string,  `deptime` string,  `crsdeptime` string,  `arrtime` string,  `crsarrtime` string,  `uniquecarrier` string,  `flightnum` string,  `tailnum` string,  `actualelapsedtime` string,  `crselapsedtime` string,  `airtime` string,  `arrdelay` string,  `depdelay` string,  `origin` string,  `dest` string,  `distance` string,  `taxiin` string,  `taxiout` string,  `cancelled` string,  `cancellationcode` string,  `diverted` string,  `carrierdelay` string,  `weatherdelay` string,  `nasdelay` string,  `securitydelay` string,  `lateaircraftdelay` string,  `load\_dt` string,  `load\_dtm` string)  PARTITIONED BY (  `year` string)  ROW FORMAT SERDE  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  STORED AS INPUTFORMAT  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'  OUTPUTFORMAT  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'  LOCATION  'hdfs://localhost:9000/airline\_data/cur/detailed\_data'; |
| --- |

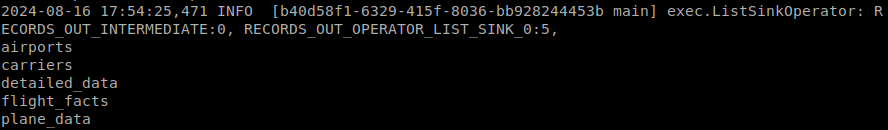
* Tạo bảng “flight\_facts” với code như sau:

| CREATE EXTERNAL TABLE `flight\_facts`(  `month` string,  `flightnum` string,  `tailnum` string,  `uniquecarrier` string,  `avg\_arr\_delay` double,  `avg\_dep\_delay` double,  `avg\_distance` double,  `max\_distance` int,  `tot\_cancelled\_flights` bigint,  `tot\_diverted\_flights` bigint,  `airline\_name` string,  `manufacturer` string,  `punctuality\_code` string,  `load\_dt` string,  `load\_dtm` string)  PARTITIONED BY (  `year` string)  ROW FORMAT SERDE  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  STORED AS INPUTFORMAT  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'  OUTPUTFORMAT  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'  LOCATION  'hdfs://localhost:9000/airline\_data/cur/flight\_facts'; |
| --- |

* Tạo bảng “plane\_data” với code như sau:

| CREATE EXTERNAL TABLE `plane\_data`(  `tailnum` string,  `type` string,  `manufacturer` string,  `issue\_date` string,  `model` string,  `status` string,  `aircraft\_type` string,  `engine\_type` string,  `year` string,  `load\_dt` string,  `load\_dtm` string)  ROW FORMAT SERDE  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  STORED AS INPUTFORMAT  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'  OUTPUTFORMAT  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'  LOCATION  'hdfs://localhost:9000/airline\_data/cur/plane\_data'; |
| --- |

* Sau khi thực hiện tạo các bảng trong cơ sở dữ liệu “airline\_db\_ cur”, kết quả như hình dưới đây:



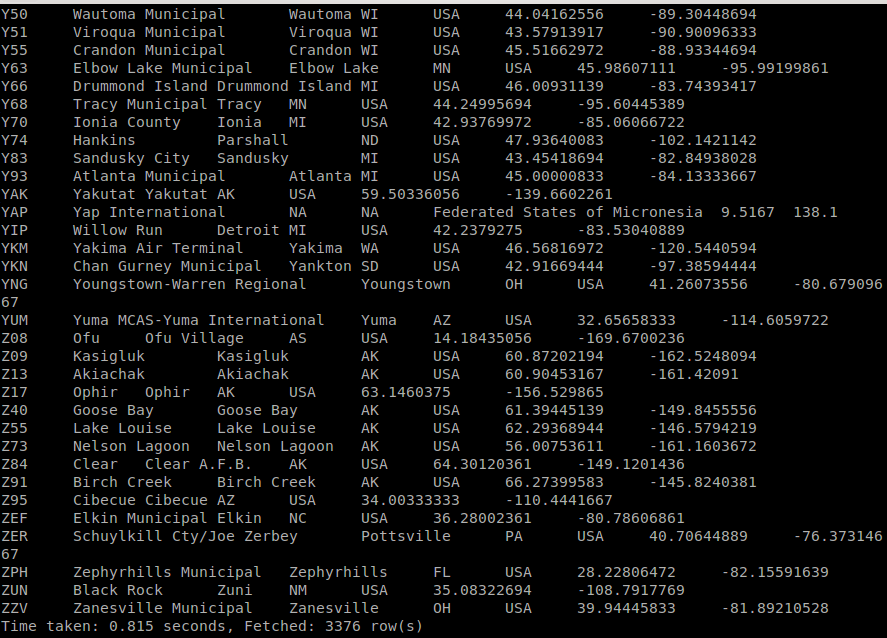
## **3.3. Tải dữ liệu vào các bảng từ các file csv đã được lưu trên HDFS**

***3.3.1. Tải dữ liệu vào các bảng trong cơ sở dữ liệu “airline\_db\_ raw”***

* Tải dữ liệu vào bảng “airports”
* Sử dụng câu lệnh:

| LOAD DATA INPATH '/airline\_data/archival/airports.csv' INTO TABLE airports; |
| --- |

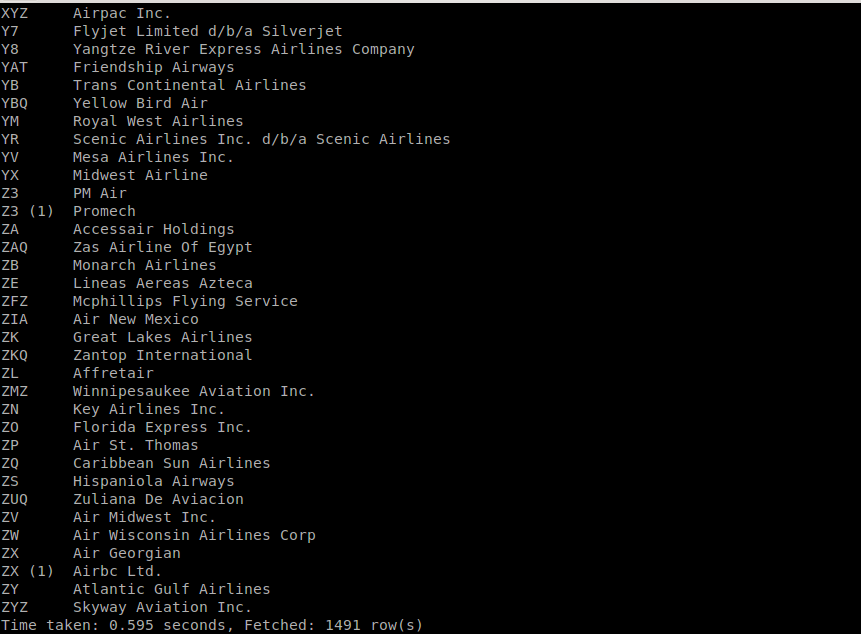
* Kết quả sau khi thực hiện:

****

* Tải dữ liệu vào bảng “carriers”
* Sử dụng câu lệnh:

| LOAD DATA INPATH '/airline\_data/archival/carriers.csv' INTO TABLE carriers; |
| --- |

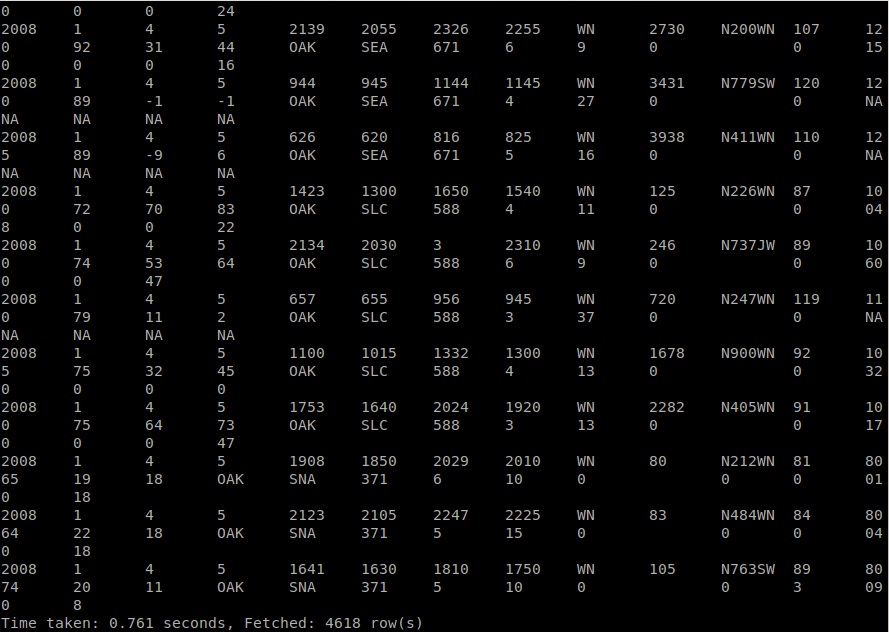
* Kết quả sau khi thực hiện:

****

* Tải dữ liệu vào bảng “detailed\_data”
* Sử dụng câu lệnh:

| LOAD DATA INPATH '/airline\_data/archival/detailed\_data.csv' INTO TABLE detailed\_data; |
| --- |

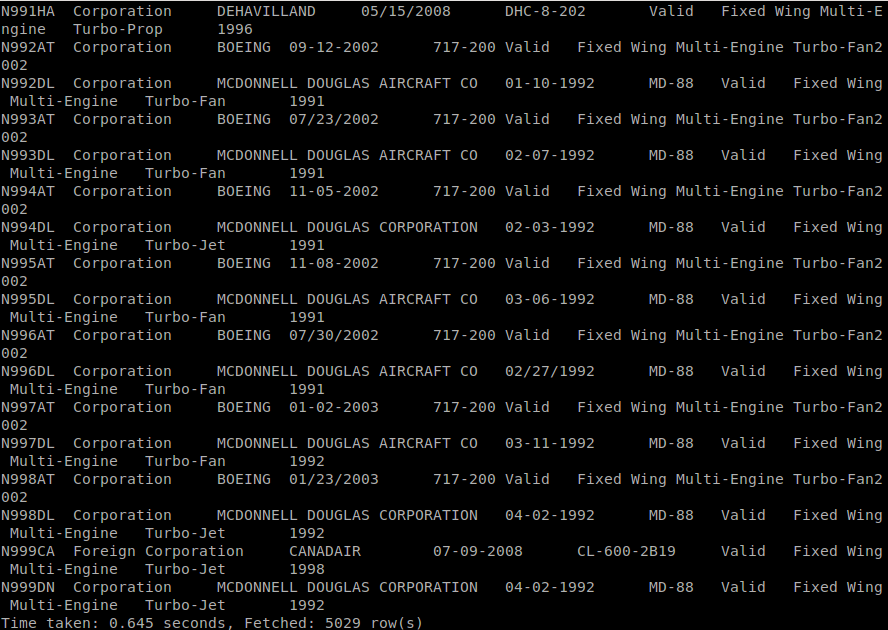
* Kết quả sau khi thực hiện:

****

* Tải dữ liệu vào bảng “plane\_data”
* Sử dụng câu lệnh:

| LOAD DATA INPATH '/airline\_data/archival/plane-data.csv' INTO TABLE plane\_data; |
| --- |

* Kết quả sau khi thực hiện:

****

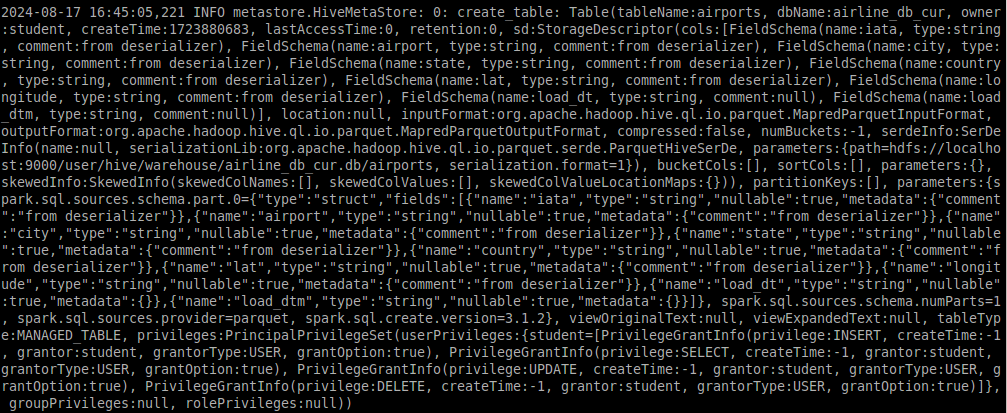
## **3.4. Tiến hành quá trình ETL dùng Apache Spark**

***3.4.1. Quá trình ETL cho bảng airports trong cơ sở dữ liệu “airline\_db\_cur”***

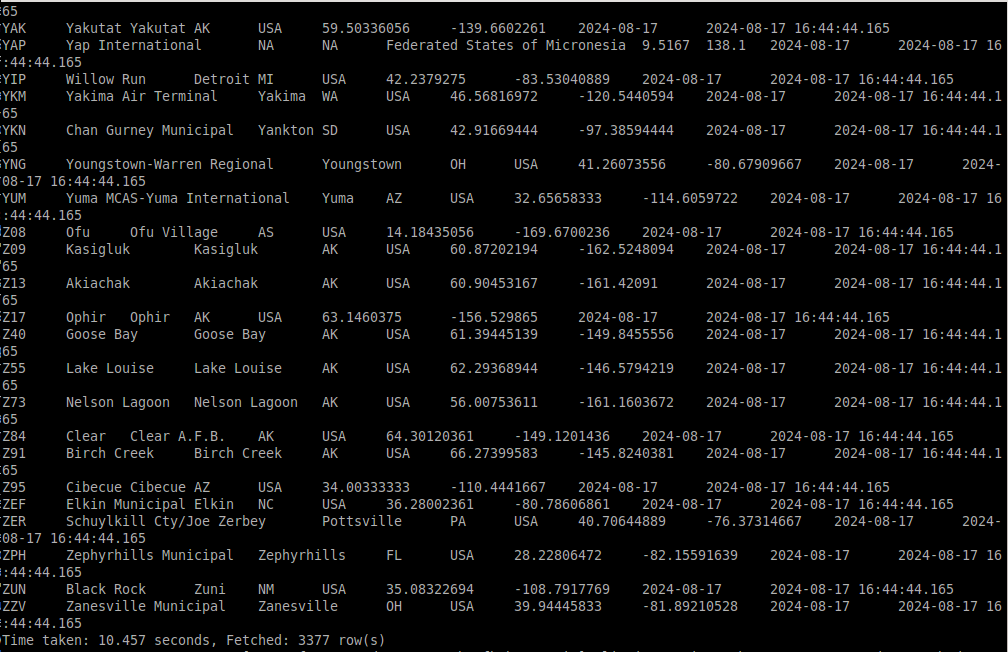
* Sử dụng code sau để tiến hành quá trình ETL trong bảng airports:

| from pyspark.sql import SparkSession  from pyspark.sql.functions import current\_date, current\_timestamp  # Tạo SparkSession  spark = SparkSession.builder \      .appName("AirportsDataProcessing") \      .enableHiveSupport() \      .getOrCreate()  # Đọc dữ liệu từ Hive table  airports\_df = spark.sql("SELECT \* FROM airline\_db\_raw.airports")  # Biến đổi dữ liệu  data\_transformed = airports\_df.withColumn("load\_dt", current\_date().cast("string")) \                                .withColumn("load\_dtm", current\_timestamp().cast("string"))  # Lưu dữ liệu vào Hive table khác  data\_transformed.write.mode("overwrite").saveAsTable("airline\_db\_cur.airports")  # Dừng SparkSession  spark.stop() |
| --- |

* Kết quả sau khi thực hiện

****

* Dữ liệu được lưu trữ trong hive ở bảng airports trong cơ sở dữ liệu “airline\_db\_cur”

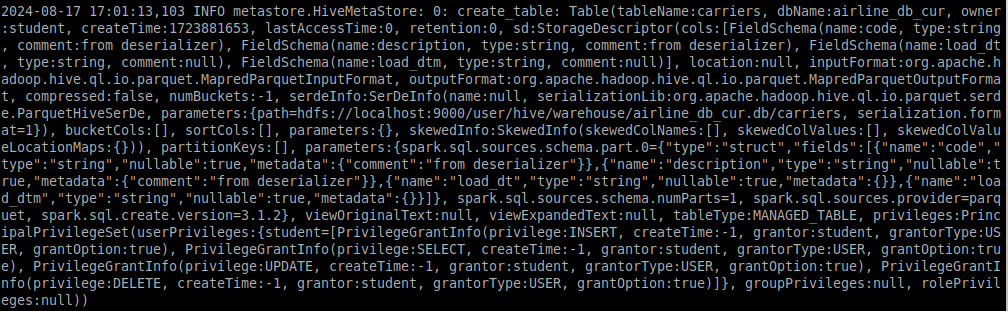
****

***3.4.2. Quá trình ETL cho bảng carriers trong cơ sở dữ liệu “airline\_db\_cur”***

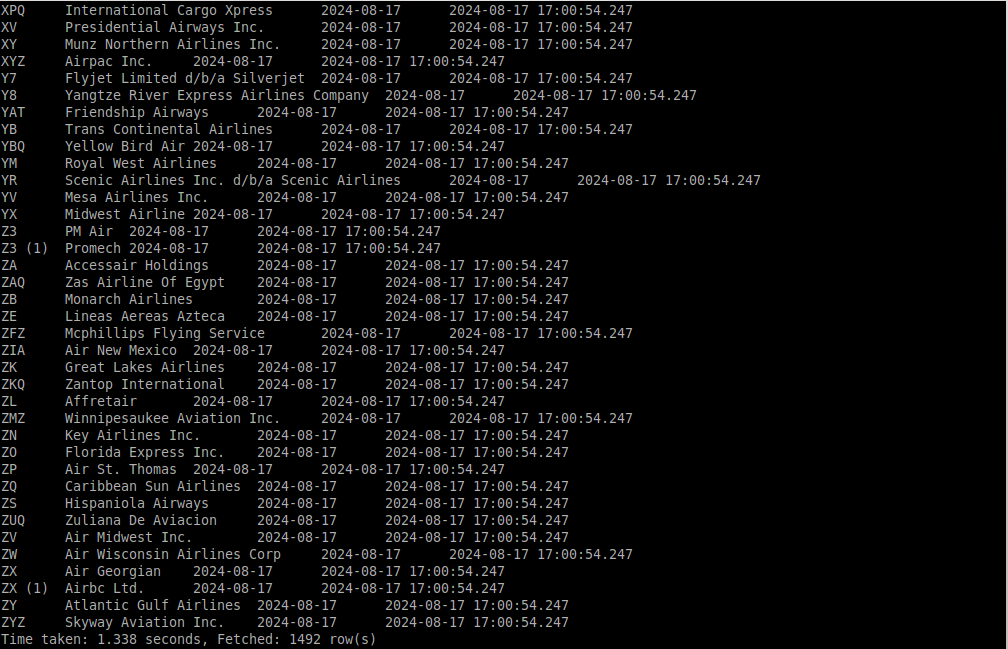
* Sử dụng code sau để tiến hành quá trình ETL trong bảng carriers:

| from pyspark.sql import SparkSession  from pyspark.sql.functions import current\_date, current\_timestamp  # Tạo SparkSession  spark = SparkSession.builder \      .appName("Transform Carriers Data") \      .enableHiveSupport() \      .getOrCreate()  # Tải dữ liệu từ bảng Hive  carriers\_df = spark.sql("SELECT \* FROM airline\_db\_raw.carriers")  # Biến đổi dữ liệu  data\_transformed = carriers\_df.withColumn("load\_dt", current\_date().cast("string")) \                                .withColumn("load\_dtm", current\_timestamp().cast("string"))  # Lưu dữ liệu vào bảng Hive khác  data\_transformed.write.mode("overwrite").saveAsTable("airline\_db\_cur.carriers")  # Dừng SparkSession  spark.stop() |
| --- |

* Kết quả sau khi thực hiện

****

* Dữ liệu được lưu trữ trong hive ở bảng carriers trong cơ sở dữ liệu “airline\_db\_cur”

****

***3.4.3. Quá trình ETL cho bảng detailed\_data trong cơ sở dữ liệu “airline\_db\_cur”***

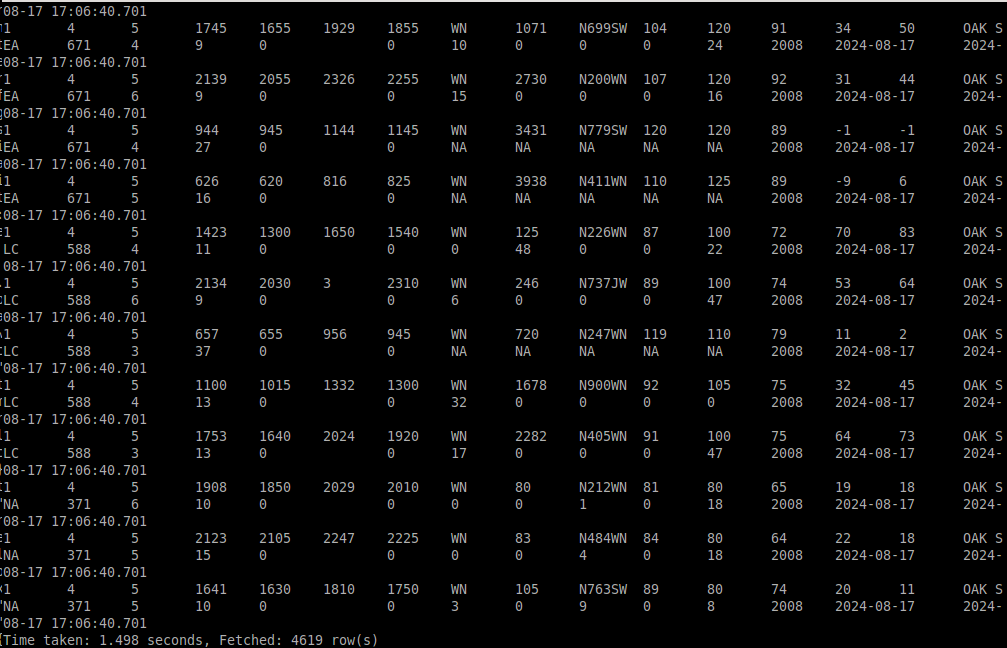
* Sử dụng code sau để tiến hành quá trình ETL trong bảng detailed\_data

| from pyspark.sql import SparkSession  from pyspark.sql.functions import current\_date, current\_timestamp  # Tạo SparkSession  spark = SparkSession.builder \      .appName("Transform Detailed Data") \      .enableHiveSupport() \      .getOrCreate()  # Tải dữ liệu từ bảng Hive  detailed\_data\_df = spark.sql("SELECT \* FROM airline\_db\_raw.detailed\_data")  # Biến đổi dữ liệu  data\_transformed = detailed\_data\_df.withColumn("load\_dt", current\_date().cast("string")) \                                     .withColumn("load\_dtm", current\_timestamp().cast("string"))  # Chọn các cột cần thiết và thêm các cột mới  selected\_columns = [      "month", "dayofmonth", "dayofweek", "deptime", "crsdeptime", "arrtime",      "crsarrtime", "uniquecarrier", "flightnum", "tailnum", "actualelapsedtime",      "crselapsedtime", "airtime", "arrdelay", "depdelay", "origin", "dest",      "distance", "taxiin", "taxiout", "cancelled", "cancellationcode",      "diverted", "carrierdelay", "weatherdelay", "nasdelay", "securitydelay",      "lateaircraftdelay", "year", "load\_dt", "load\_dtm"  ]  # Lấy các cột đã chọn và lưu vào bảng Hive khác  data\_transformed.select(\*selected\_columns).write.mode("overwrite").saveAsTable("airline\_db\_cur.detailed\_data")  # Dừng SparkSession  spark.stop() |
| --- |

* Kết quả sau khi thực hiện

****

* Dữ liệu được lưu trữ trong hive ở bảng detailed\_data trong cơ sở dữ liệu “airline\_db\_cur”

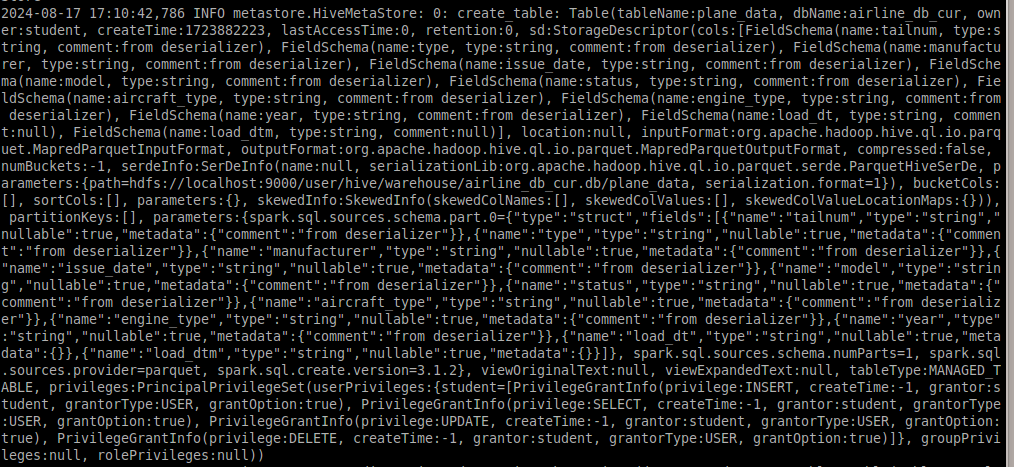
****

***3.4.4. Quá trình ETL cho bảng plane\_data trong cơ sở dữ liệu “airline\_db\_cur”***

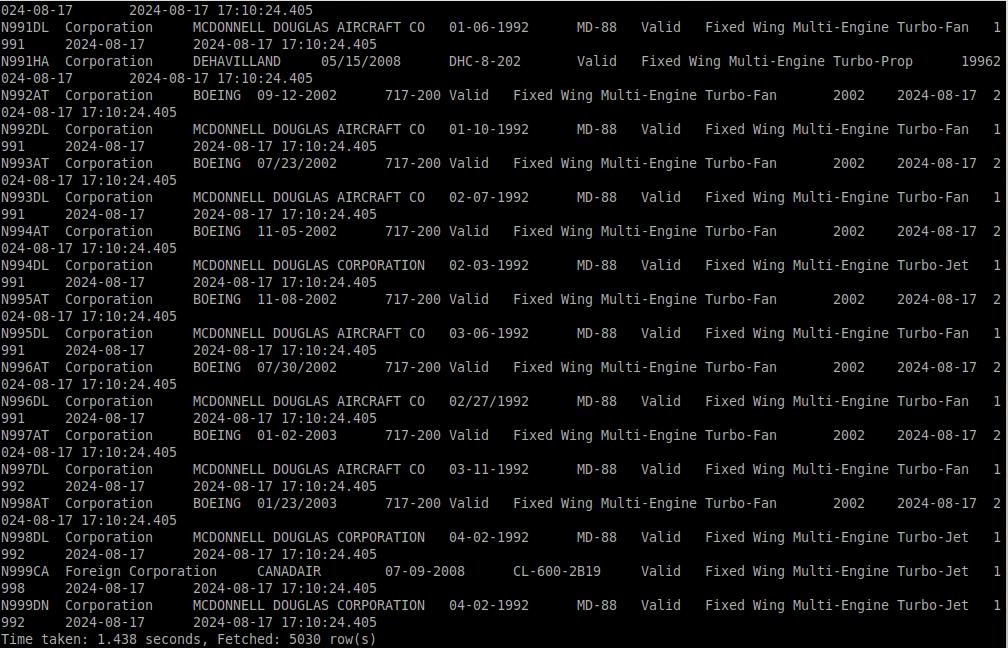
* Sử dụng code sau để tiến hành quá trình ETL trong bảng plane\_data

| from pyspark.sql import SparkSession  from pyspark.sql.functions import current\_date, current\_timestamp  # Tạo SparkSession  spark = SparkSession.builder \      .appName("Transform Plane Data") \      .enableHiveSupport() \      .getOrCreate()  # Tải dữ liệu từ bảng Hive  plane\_data\_df = spark.sql("SELECT \* FROM airline\_db\_raw.plane\_data")  # Biến đổi dữ liệu  data\_transformed = plane\_data\_df.withColumn("load\_dt", current\_date().cast("string")) \                                  .withColumn("load\_dtm", current\_timestamp().cast("string"))  # Chọn các cột cần thiết và thêm các cột mới  selected\_columns = [      "tailnum", "type", "manufacturer", "issue\_date", "model", "status",      "aircraft\_type", "engine\_type", "year", "load\_dt", "load\_dtm"  ]  # Lấy các cột đã chọn và lưu vào bảng Hive khác  data\_transformed.select(\*selected\_columns).write.mode("overwrite").saveAsTable("airline\_db\_cur.plane\_data")  # Dừng SparkSession  spark.stop() |
| --- |

* Kết quả sau khi thực hiện

****

* Dữ liệu được lưu trữ trong hive ở bảng plane\_data trong cơ sở dữ liệu “airline\_db\_cur”

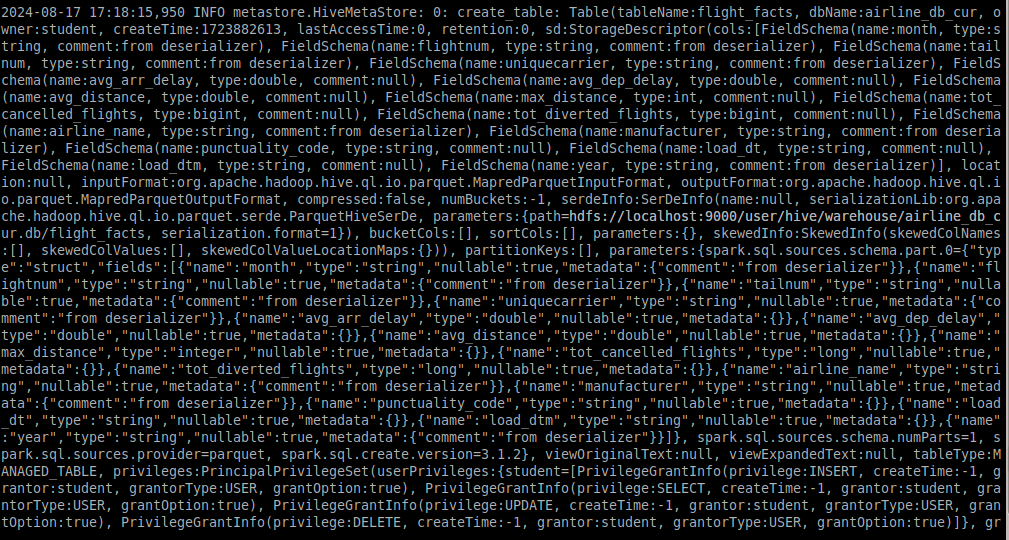
****

***3.4.5. Quá trình ETL cho bảng flight\_facts trong cơ sở dữ liệu “airline\_db\_cur”***

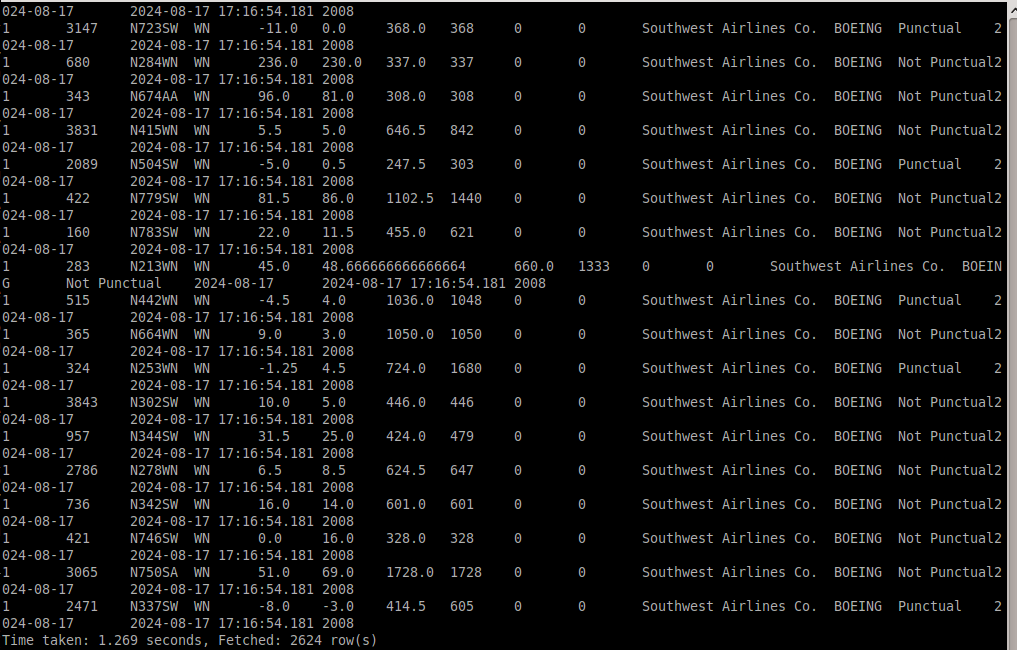
* Sử dụng code sau để tiến hành quá trình ETL trong bảng plane\_data

| from pyspark.sql import SparkSession  from pyspark.sql.functions import col, avg, max as max\_, sum as sum\_, current\_date, current\_timestamp, when  # Tạo SparkSession  spark = SparkSession.builder \      .appName("Transform Flight Data") \      .enableHiveSupport() \      .getOrCreate()  # Tải dữ liệu từ các bảng Hive  plane\_data\_df = spark.sql("SELECT \* FROM airline\_db\_cur.plane\_data")  carriers\_data\_df = spark.sql("SELECT \* FROM airline\_db\_cur.carriers")  airport\_data\_df = spark.sql("SELECT \* FROM airline\_db\_cur.airports")  detailed\_data\_df = spark.sql("SELECT \* FROM airline\_db\_cur.detailed\_data")  # Biến đổi dữ liệu  plane\_data\_manu = plane\_data\_df.select("tailnum", "manufacturer")  carriers\_man = carriers\_data\_df.select("code", "description")  derived\_data\_transformed = detailed\_data\_df.select(      "year", "month", "flightnum", "tailnum", "uniquecarrier",      col("arrdelay").cast("int"), col("depdelay").cast("int"),      col("distance").cast("int"), col("cancelled").cast("int"),      col("diverted").cast("int")  )  # Nhóm dữ liệu theo các cột cần thiết và tính các thống kê  derived\_grouped\_data = derived\_data\_transformed.groupBy("year", "month", "flightnum", "tailnum", "uniquecarrier").agg(      avg("arrdelay").alias("avg\_arr\_delay"),      avg("depdelay").alias("avg\_dep\_delay"),      avg("distance").alias("avg\_distance"),      max\_("distance").alias("max\_distance"),      sum\_("cancelled").alias("tot\_cancelled\_flights"),      sum\_("diverted").alias("tot\_diverted\_flights")  )  # Thực hiện join với bảng carriers\_data\_df để lấy tên hãng hàng không  derived\_avg\_arr\_flight = derived\_grouped\_data.join(carriers\_data\_df, derived\_grouped\_data.uniquecarrier == carriers\_data\_df.code)  # Chọn các cột cần thiết và đổi tên  result\_crr = derived\_avg\_arr\_flight.select(      "year", "month", "flightnum", "tailnum", "uniquecarrier",      "avg\_arr\_delay", "avg\_dep\_delay", "avg\_distance", "max\_distance",      "tot\_cancelled\_flights", "tot\_diverted\_flights",      col("description").alias("airline\_name")  )  # Thực hiện join với bảng plane\_data\_manu để lấy thông tin nhà sản xuất  result\_crr\_man\_join = result\_crr.join(plane\_data\_manu, "tailnum")  # Chọn các cột cần thiết, tính toán cột mới và thêm cột ngày hiện tại và thời gian hiện tại  final = result\_crr\_man\_join.select(      "month", "flightnum", "tailnum", "uniquecarrier",      "avg\_arr\_delay", "avg\_dep\_delay", "avg\_distance", "max\_distance",      "tot\_cancelled\_flights", "tot\_diverted\_flights", "airline\_name", "manufacturer",      when(col("avg\_arr\_delay") < 0, "Punctual").otherwise("Not Punctual").alias("punctuality\_code"),      current\_date().cast("string").alias("load\_dt"),      current\_timestamp().cast("string").alias("load\_dtm"),      "year"  )  # Lưu dữ liệu vào bảng Hive  final.write.mode("overwrite").saveAsTable("airline\_db\_cur.flight\_facts")  # Dừng SparkSession  spark.stop() |
| --- |

* Kết quả sau khi thực hiện

****

* Dữ liệu được lưu trữ trong hive ở bảng flight\_facts trong cơ sở dữ liệu “airline\_db\_cur”

****



# **4. Projected Impact**

## **4.1. Accomplishments and Benefits**

Dự án này nhằm khai thác tiềm năng của phân tích dữ liệu lớn trong ngành hàng không bằng cách sử dụng các công nghệ mã nguồn mở như Hadoop, Spark, Hive và Tableau. Với sự gia tăng liên tục của khối lượng dữ liệu từ các hoạt động hàng không, việc áp dụng các công cụ phân tích dữ liệu hiện đại là cần thiết để giúp các hãng hàng không đưa ra quyết định thông minh hơn, cải thiện hiệu suất và nâng cao trải nghiệm khách hàng. Bằng cách triển khai một quy trình ETL (Extract, Transform, Load) với Spark, dữ liệu từ các nguồn khác nhau sẽ được xử lý và chuyển đổi để tạo ra những thông tin có giá trị, sau đó được lưu trữ trong Hive và trực quan hóa với Tableau. Dự án này không chỉ tập trung vào việc xử lý dữ liệu hiệu quả mà còn vào việc hiểu rõ và tối ưu hóa quy trình quản lý dữ liệu trong môi trường hàng không.

## **4.2. Future Improvements**

Trong tương lai, dự án phân tích dữ liệu hàng không có thể được cải thiện để mang lại giá trị lớn hơn thông qua một số lĩnh vực quan trọng. Thứ nhất, việc tích hợp dữ liệu thời gian thực như cập nhật thời tiết, thông tin kiểm soát không lưu và trạng thái chuyến bay sẽ cải thiện đáng kể độ chính xác và kịp thời của phân tích, cho phép đưa ra quyết định và tối ưu hóa lịch trình bay và nguồn lực một cách linh hoạt hơn. Thứ hai, mở rộng sử dụng các thuật toán học máy và trí tuệ nhân tạo để dự đoán các sự cố như chậm trễ hay hủy chuyến sẽ giúp các hãng hàng không chủ động xử lý vấn đề trước khi chúng xảy ra. Thứ ba, nâng cao trải nghiệm khách hàng cá nhân hóa thông qua phân tích dữ liệu hành khách sẽ giúp các hãng hàng không tùy chỉnh dịch vụ, giao tiếp và ưu đãi phù hợp với nhu cầu từng người, tăng cường sự hài lòng và trung thành của khách hàng. Thứ tư, tích hợp các chỉ số bền vững vào phân tích sẽ hỗ trợ các hãng hàng không tối ưu hóa hoạt động bay về hiệu quả nhiên liệu và giảm thiểu tác động môi trường, bao gồm việc phân tích mô hình tiêu thụ nhiên liệu và tối ưu hóa đường bay. Thứ năm, đảm bảo an ninh và quyền riêng tư dữ liệu là điều cần thiết, với việc áp dụng các phương pháp mã hóa tiên tiến, ẩn danh dữ liệu nhạy cảm và tuân thủ các quy định bảo vệ dữ liệu toàn cầu. Cuối cùng, tối ưu hóa hiệu suất và khả năng mở rộng của hạ tầng xử lý dữ liệu sẽ giúp dự án xử lý được lượng dữ liệu lớn hơn và các truy vấn phức tạp hơn, đồng thời cải thiện tốc độ lưu trữ và truy xuất dữ liệu. Bằng cách tập trung vào các cải tiến này, dự án có thể tiếp tục phát triển và cung cấp những hiểu biết sâu sắc hơn cho các hãng hàng không, giúp họ hoạt động hiệu quả hơn, nâng cao trải nghiệm khách hàng và duy trì tính cạnh tranh trong ngành công nghiệp dựa trên dữ liệu.

# **5. Team Member Review and Comment**

| Trần Văn Tiến    Trương Thị Thùy Dung    Đặng Xuân Huyền |
| --- |

| NAME | REVIEW and COMMENT |
| --- | --- |
| Trần Văn Tiến | Tiến phân công nhiệm vụ hợp lý và giám sát chặt chẽ tiến độ dự án. Tìm hiểu sâu về quy trình ETL với Spark, tối ưu hóa quy trình xử lý dữ liệu của nhóm. Thiết kế pipeline và kế hoạch hành động bằng Word rất chi tiết và có cấu trúc tốt. |
| Trương Thị Thùy Dung | Dung đã nghiên cứu về Hadoop, đóng góp quan trọng trong việc tìm kiếm và mô hình hóa dữ liệu. Các biểu đồ trực quan của Dung giúp làm rõ các kết quả, và các bản thuyết trình PowerPoint được thực hiện rất chuyên nghiệp. |
| Đặng Xuân Huyền | Huyền đã áp dụng hiệu quả Hive và Tableau, cải thiện phân tích và truy vấn dữ liệu. Báo cáo cuối cùng rất chi tiết và có tổ chức tốt, giúp nhóm hiểu rõ hơn về dữ liệu. |



# **6. Instructor Review and Comment**

| CATEGORY | SCORE | REVIEW and COMMENT |
| --- | --- | --- |
| IDEA | \_\_/10 |  |
| APPLICATION | \_\_/30 |  |
| RESULT | \_\_/30 |  |
| PROJECT MANAGEMENT | \_\_/10 |  |
| PRESENTATION & REPORT | \_\_/20 |  |
| TOTAL | \_\_/100 |  |