



**TẬP ĐOÀN CÔNG NGHIỆP – VIỄN THÔNG QUÂN ĐỘI**

# **BÁO CÁO MINI-PROJECT**

## **Dashboard phân tích dữ liệu**

**ĐÀO ANH QUÂN**

anhquan7303qqq@gmail.com

**Chương trình Viettel Digital Talent 2025**

**Lĩnh vực: Data Engineering**

**Mentor:** Nguyễn Tuấn Anh

**Đơn vị:** VTT

**HÀ NỘI, 06/2025**

## **Lời mở đầu**

Xin gửi lời cảm ơn chân thành đến Ban Tổ chức Chương trình Viettel Digital Talent 2025 và các anh chị trong Lĩnh vực Data Engineering đã tạo cơ hội để em tham gia và học hỏi. Đặc biệt, em rất vinh dự khi được hướng dẫn thực hiện Mini Project với đề tài “Dashboard phân tích dữ liệu” – một đề tài rất có ý nghĩa trong thực tế. Đây là một lĩnh vực đầy tiềm năng và thử thách, mở ra nhiều hướng nghiên cứu và ứng dụng thực tiễn.

Sự hỗ trợ và hướng dẫn tận tình của các anh mentor đã giúp em có cái nhìn sâu sắc và toàn diện hơn, từ đó góp phần nâng cao năng lực chuyên môn và định hướng nghề nghiệp. Một lần nữa, xin chân thành cảm ơn các anh chị trong Ban Tổ chức Chương trình Viettel Digital Talent 2025 nói riêng và Tập đoàn Công nghiệp - Viễn thông Quân đội Viettel nói chung.

## **Tóm tắt nội dung và đóng góp**

Báo cáo này trình bày quy trình triển khai một hệ thống trực quan hóa dữ liệu từ dữ liệu hành vi người tiêu dùng trên nền tảng thương mại điện tử. Dự án hướng tới việc xây dựng một công cụ hỗ trợ phân tích dữ liệu có tính linh hoạt cao, giúp người dùng có thể theo dõi các chỉ số kinh doanh và xu hướng tiêu dùng dưới dạng biểu đồ tương tác.

Trong quá trình thực hiện, dữ liệu thô được xử lý qua một pipeline ETL cơ bản nhằm đảm bảo tính toàn vẹn và dễ khai thác. Các bước xử lý bao gồm làm sạch dữ liệu, tính toán các chỉ số quan trọng và chuẩn hóa định dạng thông tin. Các công nghệ chủ đạo được sử dụng trong pipeline bao gồm Apache Spark để xử lý phân tán, Apache Airflow để tự động hóa quy trình và Hadoop HDFS làm hệ thống lưu trữ dữ liệu.

Sản phẩm cuối cùng là một dashboard được xây dựng bằng Power BI, với khả năng kết nối dữ liệu tự động và cung cấp bộ công cụ lọc nâng cao, giúp người dùng có thể truy xuất thông tin theo nhiều góc nhìn khác nhau. Báo cáo này không chỉ minh họa khả năng tích hợp giữa xử lý dữ liệu lớn và trực quan hóa, mà còn khẳng định tính ứng dụng cao của các công cụ Data Engineering trong việc hỗ trợ ra quyết định tại các tổ chức, doanh nghiệp.

Sinh viên thực hiện

Quân

Đào Anh Quân

## MỤC LỤC

<b>DANH MỤC HÌNH VẼ .....</b>	<b>5</b>
<b>DANH MỤC BẢNG.....</b>	<b>6</b>
<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>7</b>
1.1 Đặt vấn đề .....	7
1.2 Mục tiêu và phạm vi.....	7
<b>CHƯƠNG 2. NỘI DUNG VÀ PHƯƠNG PHÁP .....</b>	<b>8</b>
2.1 Công nghệ sử dụng.....	8
2.1.1 Apache Airflow .....	8
2.1.2 Hadoop HDFS .....	8
2.1.3 Apache Spark.....	9
2.1.4 PostgreSQL.....	9
2.1.5 Power BI .....	9
2.2 Phương pháp thực hiện.....	10
<b>CHƯƠNG 3. TRIỂN KHAI VÀ KẾT QUẢ THỰC HIỆN.....</b>	<b>11</b>
3.1 Kiến trúc tổng quan.....	11
3.2 Triển khai dự án .....	11
3.2.1 Phân tích yêu cầu và nguồn dữ liệu .....	11
3.2.2 Xây dựng pipeline ETL xử lý dữ liệu.....	12
3.2.3 Xây dựng và phát triển Dashboard .....	17
3.2.4 Triển khai các thành phần lên Docker .....	18
3.3 Kết quả thực hiện .....	19
<b>CHƯƠNG 4. KẾT LUẬN.....</b>	<b>23</b>
4.1 Kết luận .....	23
4.2 Hướng phát triển trong tương lai .....	23
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>24</b>

## DANH MỤC HÌNH VẼ

<b>Hình 3.1.</b> Kiến trúc tổng quan của dự án.....	11
<b>Hình 3.2.</b> DAG của quá trình ETL .....	16
<b>Hình 3.3.</b> Các bảng dữ liệu được import từ PostgreSQL .....	17
<b>Hình 3.4.</b> Triển khai các thành phần lên Docker.....	18
<b>Hình 3.5.</b> Kết quả xử lý dữ liệu trên Spark .....	19
<b>Hình 3.6.</b> Kết quả thực thi định kỳ bằng Airflow.....	20
<b>Hình 3.7.</b> Lưu trữ dữ liệu trên HDFS .....	20
<b>Hình 3.8.</b> Mô hình dữ liệu trong Power BI sử dụng kiến trúc star schema.....	21
<b>Hình 3.9.</b> Dashboard phân tích hành vi và doanh thu trong Power BI.....	22

## DANH MỤC BẢNG

<b>Bảng 3.1.</b> Các trường thông tin của dữ liệu nguồn.....	12
<b>Bảng 3.2.</b> Thông tin bảng dim_date .....	13
<b>Bảng 3.3.</b> Thông tin bảng dim_time .....	13
<b>Bảng 3.4.</b> Thông tin bảng dim_event_type.....	14
<b>Bảng 3.5.</b> Thông tin bảng dim_product.....	14
<b>Bảng 3.6.</b> Thông tin bảng dim_category .....	14
<b>Bảng 3.7.</b> Thông tin bảng fact_events .....	15
<b>Bảng 3.8.</b> Thông tin bảng fact_summary .....	15
<b>Bảng 3.9.</b> Thông tin bảng predicted_revenue.....	15

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

Trong bối cảnh thương mại điện tử phát triển mạnh mẽ, việc khai thác và phân tích hành vi người dùng từ dữ liệu giao dịch trở thành một công cụ quan trọng giúp các doanh nghiệp đưa ra quyết định chính xác và kịp thời. Khối lượng dữ liệu phát sinh mỗi ngày từ hàng triệu lượt xem sản phẩm, thêm vào giỏ hàng, hoặc đặt mua trên các nền tảng bán lẻ trực tuyến đòi hỏi phải có các phương pháp phân tích hiệu quả, trực quan và dễ sử dụng.

Một trong những cách tiếp cận phổ biến và hiệu quả hiện nay là sử dụng các công cụ Business Intelligence (BI) như Power BI, Tableau hoặc Google Data Studio để xây dựng các dashboard phân tích dữ liệu. Các dashboard này giúp người dùng cuối dễ dàng truy xuất thông tin, theo dõi chỉ số quan trọng và đưa ra quyết định kịp thời thông qua giao diện trực quan và tương tác.

Tuy nhiên, trong bối cảnh dữ liệu ngày càng nhiều và phức tạp, những công cụ xử lý dữ liệu thông thường sẽ khó đáp ứng được yêu cầu xử lý. Vì vậy, việc ứng dụng các công nghệ xử lý dành cho dữ liệu lớn sẽ là cần thiết và là xu hướng phát triển sau này.

## 1.2 Mục tiêu và phạm vi

Dự án “Dashboard phân tích dữ liệu” được thực hiện với mục tiêu trực quan hóa các hành vi tiêu dùng và biến động doanh thu trong một khoảng thời gian nhất định, từ đó hỗ trợ việc theo dõi, đánh giá hiệu quả hoạt động kinh doanh theo từng giai đoạn. Thông qua các biểu đồ tương tác và báo cáo động, dashboard giúp người dùng cuối dễ dàng truy xuất, lọc và phân tích dữ liệu theo nhiều tiêu chí như thời gian, khu vực, danh mục sản phẩm hoặc phân khúc khách hàng.

Đề tài tập trung vào việc xây dựng một pipeline ETL cơ bản (Extract – Transform – Load) để xử lý và chuẩn hóa dữ liệu, bao gồm loại bỏ lỗi, chuẩn hóa các trường thông tin, và tính toán các chỉ số quan trọng như tổng doanh thu, tỷ lệ chuyển đổi. Nguồn dữ liệu là dữ liệu có cấu trúc, được tổng hợp thành định dạng CSV. Quá trình xử lý dữ liệu được thực hiện thông qua các công cụ và công nghệ hiện đại trong lĩnh vực Data Engineering như Apache Spark, Apache Airflow và hệ thống tệp phân tán Hadoop HDFS. Dự án được triển khai trên môi trường cục bộ, đóng gói bằng Docker, dễ dàng cài đặt và sử dụng cho người dùng cuối.

## CHƯƠNG 2. NỘI DUNG VÀ PHƯƠNG PHÁP

### 2.1 Công nghệ sử dụng

#### 2.1.1 Apache Airflow

Apache Airflow [1] là một nền tảng mã nguồn mở dùng để lập lịch, điều phối và theo dõi trình tự xử lý dữ liệu. Được phát triển bởi Airbnb và hiện trực thuộc Apache Software Foundation, Airflow cho phép người dùng định nghĩa các luồng công việc (DAG – Directed Acyclic Graph) bằng ngôn ngữ Python, từ đó giúp tự động hóa các tác vụ ETL (Extract - Transform - Load), kiểm tra dữ liệu, huấn luyện mô hình, hoặc cập nhật dashboard.

Một DAG trong Airflow bao gồm nhiều tác vụ (task) có quan hệ phụ thuộc rõ ràng về thứ tự và logic. Airflow hỗ trợ nhiều loại tác vụ như chạy script Python, shell, Spark job, gửi email, hoặc thao tác với API bên ngoài. Bên cạnh đó, Airflow cung cấp khả năng lập lịch định kỳ (cron-like), thực thi lại tác vụ khi lỗi, gửi cảnh báo khi có sự cố và ghi log chi tiết cho từng bước xử lý.

Điểm mạnh nổi bật của Airflow là khả năng mở rộng và tích hợp tốt với hệ sinh thái Big Data: người dùng có thể sử dụng các Operator (thành phần thực thi tác vụ) có sẵn như BashOperator, PythonOperator, SparkSubmitOperator, DockerOperator,... hoặc tự xây dựng Operator riêng để phù hợp với hệ thống của mình. Giao diện web của Airflow cho phép quản lý, theo dõi trạng thái task theo thời gian thực và dễ dàng điều chỉnh khi cần.

#### 2.1.2 Hadoop HDFS

Hadoop Distributed File System (HDFS) [2] là hệ thống tệp phân tán được phát triển trong khuôn khổ của Apache Hadoop, nhằm lưu trữ dữ liệu khối lượng lớn trên nhiều máy chủ một cách an toàn và hiệu quả. HDFS tuân theo mô hình master-slave, trong đó NameNode chịu trách nhiệm quản lý metadata (thông tin về cấu trúc file và vị trí block), còn các DataNode lưu trữ trực tiếp dữ liệu người dùng. Mỗi file trong HDFS được chia thành các block có kích thước mặc định (thường là 128MB hoặc 256MB), và mỗi block được nhân bản trên nhiều node để đảm bảo tính sẵn sàng và độ tin cậy cao.

Một điểm mạnh quan trọng của HDFS là khả năng xử lý song song và phân tán: các tác vụ đọc/ghi có thể thực hiện đồng thời trên nhiều DataNode. Bên cạnh đó, HDFS được thiết kế tối ưu cho các ứng dụng đọc tuần tự dữ liệu lớn, phù hợp với các hệ thống xử lý theo lô như Spark hoặc MapReduce. Tuy nhiên, HDFS không phù hợp cho các ứng dụng yêu cầu đọc-ngẫu nhiên hoặc ghi dữ liệu thường xuyên với độ trễ thấp.

HDFS thường đóng vai trò là nơi lưu trữ trung tâm trong các hệ thống dữ liệu lớn, nơi dữ liệu thô (raw data) được lưu giữ lâu dài và phục vụ cho các tác vụ xử lý batch. Khi kết hợp với các hệ thống như YARN, Hive, hoặc Spark, HDFS



cung cấp một tầng lưu trữ mạnh mẽ, dễ mở rộng và có khả năng tích hợp cao trong các pipeline xử lý dữ liệu hiện đại.

### 2.1.3 Apache Spark

Apache Spark [3] là nền tảng xử lý dữ liệu phân tán mã nguồn mở, nổi bật với khả năng xử lý song song trong bộ nhớ (in-memory). Spark cung cấp các API cấp cao trong các ngôn ngữ như Java, Scala, Python và R, cùng với một bộ máy thực thi được tối ưu hóa hỗ trợ đồ thị thực thi tổng quát (general execution graphs). Ngoài ra, Spark còn hỗ trợ một tập hợp phong phú các công cụ cấp cao, bao gồm: Spark SQL, Pandas API, MLlib, GraphX, Spark Structured Streaming.

Trong dự án này, Spark SQL là nền tảng được sử dụng nhằm thực hiện phân tích dữ liệu. Spark SQL là thành phần xử lý dữ liệu có cấu trúc của Apache Spark, cho phép người dùng sử dụng ngôn ngữ SQL để truy vấn và phân tích dữ liệu. Nó hỗ trợ tích hợp với các định dạng dữ liệu phổ biến như Parquet, Avro, JSON và Hive, đồng thời tận dụng khả năng tối ưu hóa của Catalyst Optimizer để tăng hiệu suất xử lý.

### 2.1.4 PostgreSQL

PostgreSQL [4] là hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, được phát triển từ năm 1986 tại Đại học California tại Berkeley, và đã trở thành một trong những hệ thống cơ sở dữ liệu phổ biến và mạnh mẽ nhất hiện nay. PostgreSQL hỗ trợ đầy đủ chuẩn SQL và mở rộng thêm các tính năng nâng cao như xử lý JSON, indexing đa dạng (B-tree, GIN, GiST), stored procedures, và cơ chế giao dịch ACID.

Một điểm nổi bật của PostgreSQL là khả năng mở rộng mạnh mẽ và tính linh hoạt cao. Người dùng có thể định nghĩa kiểu dữ liệu riêng, hàm mở rộng, hoặc module xử lý song song. PostgreSQL cũng hỗ trợ đồng thời xử lý nhiều truy vấn phức tạp, phân vùng dữ liệu và thực thi các phép toán phân tích.

Trong các hệ thống phân tích dữ liệu hiện đại, PostgreSQL thường đóng vai trò là kho lưu trữ kết quả (analytical sink), nơi tổng hợp, lưu trữ và cung cấp dữ liệu cho các công cụ trực quan hóa như Power BI, Metabase hoặc Tableau. Với khả năng tích hợp tốt, bảo mật cao và cộng đồng hỗ trợ mạnh, PostgreSQL là lựa chọn hàng đầu trong nhiều hệ thống ETL và BI hiện đại.

### 2.1.5 Power BI

Power BI [5] là một trong những công cụ Business Intelligence (BI) mạnh mẽ và phổ biến nhất hiện nay, do Microsoft phát triển. Với khả năng kết nối dữ liệu đa dạng, xử lý linh hoạt và giao diện thân thiện, Power BI cho phép người dùng dễ dàng xây dựng các báo cáo tương tác và trực quan hóa dữ liệu từ nhiều nguồn khác nhau như Excel, SQL Server, Azure, Web API, Hadoop hoặc các dịch vụ đám mây như Google Analytics, Salesforce.

Một trong những điểm nổi bật của Power BI là khả năng trực quan hóa dữ liệu một cách linh hoạt và chuyên nghiệp. Người dùng có thể sử dụng hàng chục

loại biểu đồ như bar chart, line chart, pie chart, heatmap, scatter plot... kết hợp với các công cụ phân tích như slicer, drill-down, drill-through để khám phá dữ liệu từ tổng quan đến chi tiết. Tính năng filter động theo thời gian, danh mục, khu vực, phân khúc khách hàng là yếu tố then chốt giúp người dùng cuối dễ dàng tùy biến góc nhìn.

Ngoài ra, Power BI còn hỗ trợ ngôn ngữ DAX (Data Analysis Expressions) để tạo ra các chỉ số tính toán (measure) hoặc cột mới (calculated column) phục vụ các phân tích nâng cao như tính doanh thu trung bình, tỷ lệ chuyển đổi, tổng lũy kế theo thời gian, v.v.

Với khả năng xuất bản online thông qua Power BI Service, người dùng có thể chia sẻ dashboard với các thành viên trong nhóm hoặc ban quản lý doanh nghiệp, đồng thời thiết lập cập nhật dữ liệu theo lịch trình tự động, giúp thông tin luôn được cập nhật kịp thời và chính xác.

## **2.2 Phương pháp thực hiện**

Để triển khai đề tài “Dashboard phân tích dữ liệu”, em đã lựa chọn một quy trình thực hiện có tính hệ thống, bao gồm các bước tuần tự từ phân tích dữ liệu nguồn đến trực quan hóa kết quả thông qua dashboard tương tác. Phương pháp tiếp cận của đề tài dựa trên mô hình kết hợp giữa kỹ thuật xử lý dữ liệu lớn (Big Data Engineering) và trực quan hóa dữ liệu (Data Visualization), nhằm đảm bảo tính đầy đủ, chính xác và dễ khai thác của thông tin đầu ra. Cụ thể, quá trình thực hiện được chia thành các giai đoạn chính như sau:

**Phân tích yêu cầu và nguồn dữ liệu:** Trước tiên, đề tài xác định rõ mục tiêu phân tích, các chỉ số quan trọng cần theo dõi và nhu cầu của người dùng cuối. Đồng thời, tiến hành khảo sát cấu trúc và đặc điểm của bộ dữ liệu hành vi khách hàng từ hệ thống thương mại điện tử.

**Thiết kế kiến trúc hệ thống và dashboard:** Xây dựng sơ đồ tổng quan về pipeline xử lý dữ liệu, lựa chọn công cụ phù hợp ở từng giai đoạn (Spark, Airflow, HDFS, Power BI). Thiết kế bố cục dashboard với các thành phần chính như biểu đồ theo thời gian, phân loại sản phẩm, nhóm khách hàng...

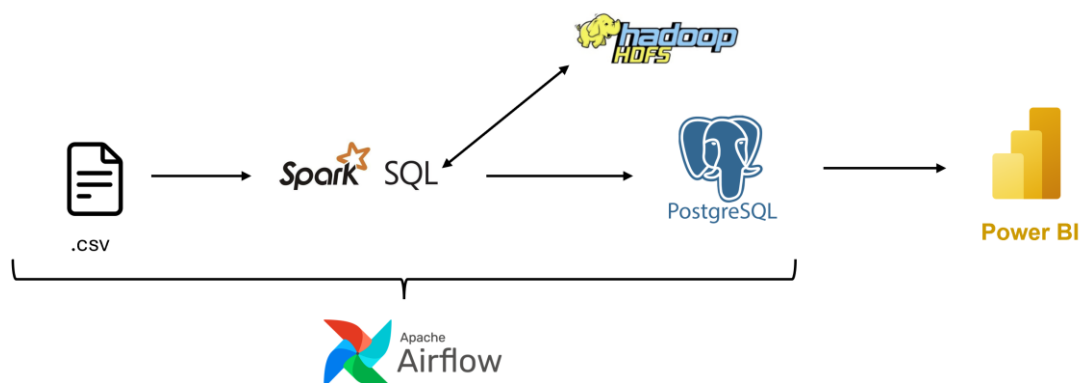
**Xây dựng pipeline ETL:** Thực hiện quy trình ETL gồm trích xuất dữ liệu từ file CSV, xử lý và chuyển đổi bằng Spark, lên lịch tự động bằng Airflow, và lưu trữ dữ liệu đã xử lý trên hệ thống phân tán HDFS.

**Phát triển dashboard phân tích:** Kết nối dữ liệu đã chuẩn hóa vào Power BI để xây dựng dashboard tương tác. Tích hợp các tính năng như bộ lọc theo thời gian, danh mục, thương hiệu; biểu đồ động thể hiện doanh thu, số lượt mua, tỷ lệ chuyển đổi...

**Kiểm thử và đánh giá:** Đảm bảo độ chính xác của các biểu đồ và chỉ số, kiểm tra tính tương tác và khả năng mở rộng của dashboard trong môi trường triển khai thực tế.

## CHƯƠNG 3. TRIỂN KHAI VÀ KẾT QUẢ THỰC HIỆN

### 3.1 Kiến trúc tổng quan



Hình 3.1. Kiến trúc tổng quan của dự án

Hình 3.1 là kiến trúc tổng quan của dự án. Quá trình thực hiện bắt đầu từ việc đọc dữ liệu tổng hợp trong 1 tháng của một nền tảng thương mại điện tử. Dữ liệu này sau đó được làm sạch, chuẩn hóa bằng Spark SQL và lưu trữ trên HDFS. Tiếp đó, dữ liệu sạch được đọc từ HDFS và thực hiện các phân tích, biến đổi. Dữ liệu này sẽ được lưu trữ trên PostgreSQL theo cấu trúc Star Schema, thuận lợi cho việc trực quan hóa trên PowerBI cũng như hiệu quả trong truy vấn và lưu trữ. Cuối cùng, dữ liệu sẽ được đọc từ PostgreSQL để tạo báo cáo phân tích trên PowerBI. Toàn bộ quá trình ETL sẽ được tự động hóa bằng Airflow.

### 3.2 Triển khai dự án

#### 3.2.1 Phân tích yêu cầu và nguồn dữ liệu

Trước khi xây dựng hệ thống phân tích và trực quan hóa, bước đầu tiên của dự án là phân tích yêu cầu nghiệp vụ và đánh giá đặc điểm của nguồn dữ liệu đầu vào. Mục tiêu của hệ thống dashboard là giúp người dùng cuối – có thể là nhà quản lý, chuyên viên kinh doanh hoặc phân tích dữ liệu – dễ dàng theo dõi các chỉ số kinh doanh, nắm bắt hành vi khách hàng và phát hiện xu hướng theo thời gian.

Từ yêu cầu thực tế đó, các nhóm thông tin cần được phân tích bao gồm:

- Doanh thu theo thời gian (ngày, giờ)
- Mức độ quan tâm của người dùng đến từng sản phẩm hoặc danh mục
- Tỷ lệ chuyển đổi từ xem sản phẩm đến hành vi mua hàng
- Phân bố hành vi khách hàng theo khung giờ trong ngày, nhóm sản phẩm hoặc thương hiệu

- Các chỉ số tổng hợp: số lượng giao dịch, tổng giá trị giao dịch, số lượng người dùng

Nguồn dữ liệu ở đây em sử dụng là dữ liệu về hành vi khách hàng trên nền tảng thương mại điện tử. Dữ liệu này lấy từ Kaggle với hơn 60 triệu bản ghi mỗi tệp csv, là dữ liệu tổng hợp của 1 tháng. Lượng dữ liệu lớn này phù hợp cho yêu cầu của dự án, đó là kết hợp việc phân tích dữ liệu lớn với trực quan hóa bằng các công cụ BI. Dữ liệu bao gồm các trường như Bảng 3.1, với ý nghĩa của từng bản ghi như sau: Người dùng *user\_id*, trong phiên giao dịch *user\_session*, đã thêm vào giỏ hàng (với thuộc tính *event\_type* bằng *cart*) sản phẩm *product\_id* thuộc thương hiệu *brand*, danh mục *category\_code*, với giá là *price* tại thời điểm *event\_time*.

**Bảng 3.1.** Các trường thông tin của dữ liệu nguồn

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa
1	event_time	Timestamp	Thời điểm xảy ra sự kiện
2	event_type	String	Loại sự kiện (cart, view, purchase)
3	product_id	String	Mã định danh của sản phẩm
4	category_id	String	Mã định danh của danh mục
5	category_code	String	Tên danh mục có phân cấp
6	brand	String	Tên thương hiệu
7	price	Double	Giá sản phẩm
8	user_id	String	Mã định danh khách hàng
9	user_session	String	Mã phiên truy cập của khách hàng

### 3.2.2 Xây dựng pipeline ETL xử lý dữ liệu

Để chuẩn hóa dữ liệu đầu vào và phục vụ quá trình phân tích, đề tài đã triển khai một pipeline ETL (Extract – Transform – Load) với cấu trúc rõ ràng, đảm bảo khả năng xử lý hiệu quả và dễ mở rộng trong các giai đoạn sau. Quá trình này bắt đầu từ bước Extract, trong đó dữ liệu được đọc từ file CSV chứa hành vi người dùng trong một tháng. Sau khi đọc vào, hệ thống thực hiện định nghĩa lại các trường dữ liệu, chuẩn hóa định dạng thời gian, loại bỏ các bản ghi trùng lặp hoặc bị thiếu thông tin quan trọng như *user\_id*, *event\_time*, *product\_id*, đồng thời chuyển đổi kiểu dữ liệu về dạng phù hợp để xử lý sau này.

Dữ liệu sau khi làm sạch được lưu trên HDFS dưới định dạng Parquet và được phân vùng (partition) theo từng ngày (year/month/day). Mục đích của việc phân vùng là để tối ưu hiệu suất truy vấn trong các bài toán phân tích theo thời gian, giúp hệ thống chỉ cần đọc đúng phần dữ liệu cần thiết thay vì phải tải toàn bộ. Ngoài ra, định dạng Parquet còn giúp giảm kích thước lưu trữ đáng kể và tăng tốc độ xử lý do được Spark hỗ trợ tốt.

Sau khi dữ liệu thô được lưu trên HDFS, hệ thống tiếp tục bước Transform nhằm xây dựng các bảng dữ liệu phục vụ trực quan hóa và phân tích. Hình 3.2

Dựa trên nguyên tắc mô hình dữ liệu sao (star schema), các bảng dim (chiều) và fact (sự kiện) được trích xuất và tổng hợp. Cụ thể, em đã xây dựng các bảng dim\_date và dim\_time để phục vụ phân tích theo ngày, giờ, ca làm việc (sáng – chiều – tối – đêm); bảng dim\_product chứa thông tin về sản phẩm và giá trung bình; bảng dim\_category lưu cấu trúc phân cấp danh mục từ cấp 1 đến cấp 4; bảng dim\_event\_type ánh xạ các loại sự kiện tương tác (view, cart, purchase) thành ID để thuận tiện cho việc join. Chi tiết các bảng dim được trình bày trong các Bảng 3.2, Bảng 3.3, Bảng 3.4, Bảng 3.5 và Bảng 3.6.

**Bảng 3.2.** Thông tin bảng dim\_date

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa
1	date_id	Date	Thời gian (định dạng dd-MM-yyyy)
2	month	Int	Tháng
3	day	Int	Ngày
4	quarter	Int	Quý
5	day_of_week	Int	Số thứ tự thứ trong tuần
6	day_name	String	Tên thứ trong tuần

**Bảng 3.3.** Thông tin bảng dim\_time

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa
1	hour	Int	Giá trị giờ
2	hour_label	String	Giờ theo định dạng HH:00
3	is_morning	Boolean	Giờ có thuộc buổi sáng hay không
4	is_afternoon	Boolean	Giờ có thuộc buổi chiều hay không
5	is_evening	Boolean	Giờ có thuộc buổi tối hay không
6	is_night	Boolean	Giờ có thuộc buổi đêm hay không
7	hour_group	String	Buổi trong ngày

**Bảng 3.4.** Thông tin bảng dim\_event\_type

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa
1	event_type_id	Int	Định danh loại sự kiện (1 -> view, 2 -> cart, 3 -> purchase)
2	event_type_name	String	Tên sự kiện (view, cart, purchase)

**Bảng 3.5.** Thông tin bảng dim\_product

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa
1	product_id	String	Mã định danh sản phẩm
2	category_id	String	Mã định danh danh mục
3	brand	String	Nhãn hiệu
4	price	Double	Giá sản phẩm

**Bảng 3.6.** Thông tin bảng dim\_category

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa
1	category_id	String	Mã định danh danh mục
2	category_level_1	String	Danh mục cấp 1
3	category_level_2	String	Danh mục cấp 2
4	category_level_3	String	Danh mục cấp 3
5	category_level_4	String	Danh mục cấp 4

Trong khi đó, bảng fact\_events ghi nhận toàn bộ sự kiện người dùng tương tác với sản phẩm, bao gồm thời gian, hành vi, sản phẩm, người dùng và doanh thu tương ứng nếu là hành vi mua (purchase). Đồng thời, em xây dựng thêm bảng fact\_summary tổng hợp số lượt tương tác theo từng loại sự kiện và tính toán tỷ lệ chuyển đổi (conversion rate) cũng như tổng doanh thu theo sản phẩm. Việc tạo các bảng tổng hợp giúp phân tích sâu hơn về hiệu suất của từng sản phẩm và hành vi tiêu dùng. Ngoài ra em còn thực hiện tạo dữ liệu dự đoán đơn giản dựa trên thuật toán Linear Regression để dự đoán xu hướng doanh thu của tháng kế tiếp. Chi tiết các bảng được trình bày trong Bảng 3.7, Bảng 3.8 và Bảng 3.9.

**Bảng 3.7.** Thông tin bảng fact\_events

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa
1	event_id	Int	Mã định danh sự kiện
2	date_id	Date	Định danh ngày tháng
3	event_type_id	Int	Mã định danh sự kiện
4	hour	Int	Giờ xảy ra sự kiện
5	product_id	String	Mã định danh sản phẩm
6	quantity	Int	Đếm sự kiện (=1)
7	revenue	Double	Doanh thu (> 0 với sự kiện là purchase)
8	user_id	String	Mã định danh khách h
9	user_session	String	Định danh phiên truy cập

**Bảng 3.8.** Thông tin bảng fact\_summary

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa
1	product_id	String	Mã định danh sản phẩm
2	cart	Int	Số lượt thêm vào giỏ
3	view	Int	Số lượt xem sản phẩm
4	purchase	Int	Số lượt mua hàng
5	total_events	Int	Tổng số sự kiện
6	total_revenue	Double	Tổng doanh thu
7	purchase_conversion	Double	Tỷ lệ chuyển đổi (xem - > mua hàng)

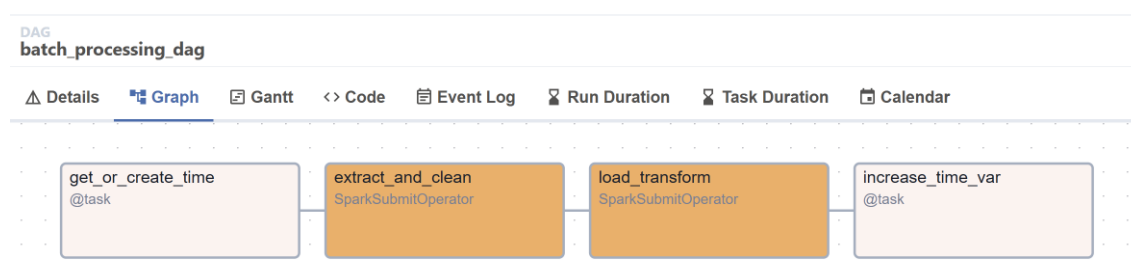
**Bảng 3.9.** Thông tin bảng predicted\_revenue

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa
1	forecast_date	Date	Ngày dự đoán
2	day	Int	Giá trị ngày
3	day_of_week	Int	Ngày trong tuần (1 – 7)
4	is_weekend	Boolean	Có phải cuối tuần hay không
5	predicted_revenue	Double	Doanh thu dự đoán

Cuối cùng, tất cả các bảng đã xử lý được ghi vào cơ sở dữ liệu PostgreSQL để phục vụ việc kết nối với Power BI. Dữ liệu được ghi thông qua giao thức JDBC với các tham số kết nối cấu hình sẵn. Việc phân tách dữ liệu

thành các bảng rõ ràng và chuẩn hóa theo cấu trúc phù hợp đã giúp quá trình kết nối, trực quan hóa và phân tích trong các bước tiếp theo diễn ra nhanh chóng và chính xác.

Toàn bộ quá trình này được tự động hóa bằng công cụ Apache Airflow – một nền tảng quản lý workflow mạnh mẽ trong lĩnh vực xử lý dữ liệu lớn. Pipeline ETL được tổ chức thành một DAG (Directed Acyclic Graph) với các task thực hiện tuần tự từ việc đọc dữ liệu gốc, xử lý, tính toán các bảng dim, fact, đến việc ghi kết quả vào PostgreSQL. Airflow giúp kiểm soát luồng công việc một cách trực quan, dễ theo dõi trạng thái từng bước và có khả năng thiết lập lịch chạy định kỳ theo ngày hoặc theo tháng. Hình 3.2 dưới đây là DAG của quá trình ETL. DAG bao gồm 4 task, lần lượt để khởi tạo biến thời gian, thực hiện extract, transform, load dữ liệu và tăng giá trị biến thời gian.



**Hình 3.2.** DAG của quá trình ETL

Để có thể thực thi được Spark Job từ cụm Spark, cần phải tạo kết nối từ Airflow đến Spark. Hình 3.3 sau đây là giao diện tạo kết nối giữa Airflow và cụm Spark.

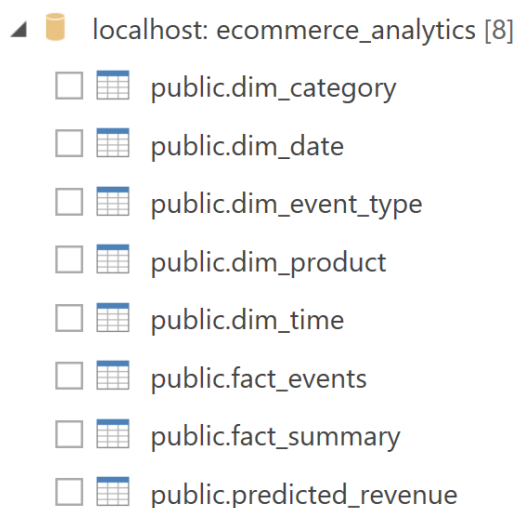
**Hình 3.3.** Tạo kết nối giữa Airflow và Spark



### 3.2.3 Xây dựng và phát triển Dashboard

Sau khi dữ liệu đã được xử lý và lưu trữ đầy đủ trong cơ sở dữ liệu PostgreSQL dưới dạng các bảng dim và fact, em tiến hành xây dựng dashboard phân tích bằng công cụ Power BI. Mục tiêu chính của dashboard là cung cấp cái nhìn trực quan, tương tác về hành vi tiêu dùng, hiệu suất sản phẩm và xu hướng doanh thu trong từng khoảng thời gian cụ thể.

Quá trình phát triển dashboard bắt đầu bằng việc kết nối Power BI với cơ sở dữ liệu PostgreSQL thông qua giao thức kết nối trực tiếp (DirectQuery hoặc Import). Các bảng dim và fact được tải vào Power BI và thiết lập mối quan hệ khóa – ngoại khóa giữa các bảng, đảm bảo đúng mô hình dữ liệu sao (star schema). Việc thiết kế dữ liệu theo cấu trúc này giúp việc tạo biểu đồ, lọc và phân tích diễn ra nhanh chóng và nhất quán. Hình 3.4 là danh sách các bảng dữ liệu sau khi kết nối kết nối PowerBI với PostgreSQL, các bảng này chính là các bảng được xử lý từ dữ liệu thô sinh ra.



**Hình 3.4.** Các bảng dữ liệu được import từ PostgreSQL

Trên giao diện Power BI, em xây dựng các trang dashboard theo từng nhóm chủ đề cụ thể. Trang tổng quan hiển thị các chỉ số chính như tổng doanh thu, số lượt mua hàng, tỷ lệ chuyển đổi, và xu hướng mua theo thời gian (line chart). Ngoài ra, em thiết kế các biểu đồ bar chart thể hiện doanh thu theo danh mục sản phẩm, thương hiệu, hoặc nhóm khách hàng. Các biểu đồ pie chart được sử dụng để phân tích phân bố hành vi theo loại sự kiện (view, cart, purchase).

Đặc biệt, dashboard được bổ sung tính năng lọc động theo thời gian (ngày, tháng), danh mục sản phẩm, thương hiệu hoặc khoảng giá. Nhờ đó, người dùng có thể chủ động lựa chọn góc nhìn phù hợp với nhu cầu phân tích, đồng thời dễ dàng phát hiện những bất thường hoặc xu hướng thay đổi theo thời gian.

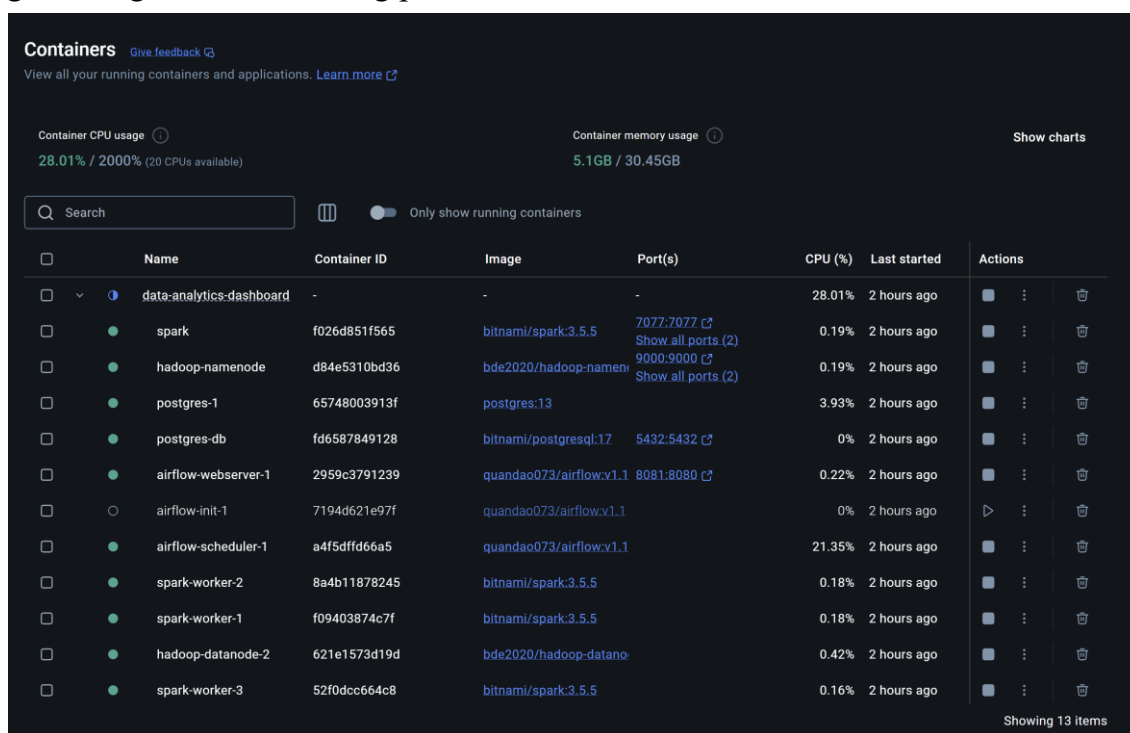
Để tăng tính chuyên nghiệp và dễ sử dụng, em đã thiết kế giao diện dashboard theo phong cách hiện đại, phân vùng rõ ràng, sử dụng màu sắc nhất quán và đặt tiêu đề, đơn vị đo lường rõ ràng cho từng biểu đồ. Dashboard có thể được xuất bản trực tuyến thông qua Power BI Service, hỗ trợ chia sẻ tới các đối

tương liên quan như quản lý, nhà phân tích hoặc nhóm vận hành. Ngoài ra, Power BI còn cho phép thiết lập lịch cập nhật dữ liệu định kỳ, đảm bảo thông tin luôn phản ánh đúng tình trạng thực tế.

Việc xây dựng dashboard không chỉ giúp trực quan hóa kết quả phân tích, mà còn đóng vai trò quan trọng trong hỗ trợ ra quyết định, đồng thời thể hiện khả năng ứng dụng công nghệ BI trong xử lý và khai thác dữ liệu thực tế một cách hiệu quả.

### 3.2.4 Triển khai các thành phần lên Docker

Để đảm bảo tính tái sử dụng, dễ dàng thiết lập môi trường và triển khai toàn bộ hệ thống một cách đồng bộ, em đã sử dụng Docker [6] kết hợp với Docker Compose để triển khai các thành phần chính của dự án. Việc container hóa giúp đơn giản hóa quá trình cấu hình, cô lập môi trường và đảm bảo khả năng mở rộng, di động cao cho hệ thống phân tích dữ liệu.



<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	data-analytics-dashboard	-	-	-	28.01%	2 hours ago	
<input type="checkbox"/>	spark	f026d851f565	bitnami/spark:3.5.5	7077:7077 Show all ports (2)	0.19%	2 hours ago	
<input type="checkbox"/>	hadoop-namenode	d84e5310bd36	bde2020/hadoop-namenode	9000:9000 Show all ports (2)	0.19%	2 hours ago	
<input type="checkbox"/>	postgres-1	65748003913f	postgres:13		3.93%	2 hours ago	
<input type="checkbox"/>	postgres-db	fd6587849128	bitnami/postgresql:17	5432:5432	0%	2 hours ago	
<input type="checkbox"/>	airflow-webserver-1	2959c3791239	quandao073/airflow:v1.1	8081:8080	0.22%	2 hours ago	
<input type="checkbox"/>	airflow-init-1	7194d621e97f	quandao073/airflow:v1.1		0%	2 hours ago	
<input type="checkbox"/>	airflow-scheduler-1	a4f5dffd66a5	quandao073/airflow:v1.1		21.35%	2 hours ago	
<input type="checkbox"/>	spark-worker-2	8a4b11878245	bitnami/spark:3.5.5		0.18%	2 hours ago	
<input type="checkbox"/>	spark-worker-1	f09403874c7f	bitnami/spark:3.5.5		0.18%	2 hours ago	
<input type="checkbox"/>	hadoop-datanode-2	621e1573d19d	bde2020/hadoop-datanode		0.42%	2 hours ago	
<input type="checkbox"/>	spark-worker-3	52f0dcc664c8	bitnami/spark:3.5.5		0.16%	2 hours ago	

Hình 3.5. Triển khai các thành phần lên Docker

Hình 3.5 mô tả các thành phần sau khi triển khai trên Docker. Cụ thể, em đã xây dựng một file docker-compose.yml với cấu trúc đầy đủ, bao gồm các thành phần chính sau:

Thành phần xử lý dữ liệu: sử dụng cụm Spark gồm một container master và ba container worker, sử dụng image bitnami/spark:3.5.5. Cụm Spark cho phép chạy các job xử lý và dự đoán doanh thu trên tập dữ liệu lớn theo mô hình phân tán.

Hệ thống lưu trữ phân tán: triển khai Hadoop HDFS với một container namenode và hai datanode từ image bde2020/hadoop. HDFS đóng vai trò lưu trữ dữ liệu thô (dạng Parquet), phục vụ cho Spark đọc/ghi hiệu quả.

Hệ quản trị cơ sở dữ liệu: sử dụng PostgreSQL để lưu các bảng phân tích đã xử lý (dim, fact, predicted\_revenue). Có hai instance PostgreSQL: một dành cho Airflow metadata (postgres) và một dành cho dữ liệu phân tích (postgres-db).


Hệ thống điều phối và tự động hóa: triển khai Apache Airflow gồm các container airflow-webserver, airflow-scheduler, airflow-init và airflow-cli. Airflow sử dụng LocalExecutor để chạy các DAG xử lý dữ liệu. Em sử dụng một image tùy chỉnh quanda073/airflow:v1.1 đã cài sẵn gói kết nối Spark để sử dụng SparkSubmitOperator, cho phép kết nối Airflow với cụm Spark.

Tất cả các container được liên kết trong một mạng ảo data-analytics-dashboard-network, cho phép các dịch vụ trao đổi dữ liệu nội bộ mà không cần mở cổng ra ngoài. Các biến môi trường như thông tin kết nối PostgreSQL, HDFS, Spark master URL được cấu hình thông qua các biến môi trường POSTGRES\_\_URI, HDFS\_\_URI, SPARK\_\_MASTER\_\_URL, giúp các dịch vụ có thể kết nối với nhau dễ dàng. Đồng thời, em gắn các thư mục chứa mã nguồn (./airflow/code), DAG (./dags) và dữ liệu (./data) vào các container tương ứng để đảm bảo dễ cập nhật và theo dõi.

Việc sử dụng Docker Compose giúp em có thể khởi động toàn bộ hệ thống chỉ với một lệnh duy nhất (docker compose up), đồng thời kiểm soát trạng thái của các container thông qua các công cụ quản lý như Docker Desktop hoặc dòng lệnh. Ngoài ra, với cấu trúc rõ ràng và tách biệt theo vai trò, hệ thống có thể dễ dàng mở rộng, triển khai thực tế trên môi trường cloud hoặc tích hợp CI/CD.

### 3.3 Kết quả thực hiện

Hình 3.6 và Hình 3.7 là kết quả của luồng ETL xử lý dữ liệu. Giao diện Spark Master cho thấy cụm xử lý gồm 3 worker đang hoạt động ổn định. Hai ứng dụng chính đã hoàn tất là *Extract and Clean Data* và *TransformLoad*, cho thấy pipeline hoạt động đúng với yêu cầu đặt ra.

 **Spark Master at spark://78300828af72:7077**

URL: spark://78300828af72:7077  
Alive Workers: 3  
Cores in use: 12 Total, 0 Used  
Memory in use: 6.0 GiB Total, 0.0 B Used  
Resources in use:  
Applications: 0 Running, 2 Completed  
Drivers: 0 Running, 0 Completed  
Status: ALIVE

▼ Workers (3)

Worker Id	Address	State	Cores	Memory	Resources
worker-20250618074539-172.20.0.10-34899	172.20.0.10:34899	ALIVE	4 (0 Used)	2.0 GiB (0.0 B Used)	
worker-20250618074539-172.20.0.7-46523	172.20.0.7:46523	ALIVE	4 (0 Used)	2.0 GiB (0.0 B Used)	
worker-20250618074539-172.20.0.8-41097	172.20.0.8:41097	ALIVE	4 (0 Used)	2.0 GiB (0.0 B Used)	

▼ Running Applications (0)

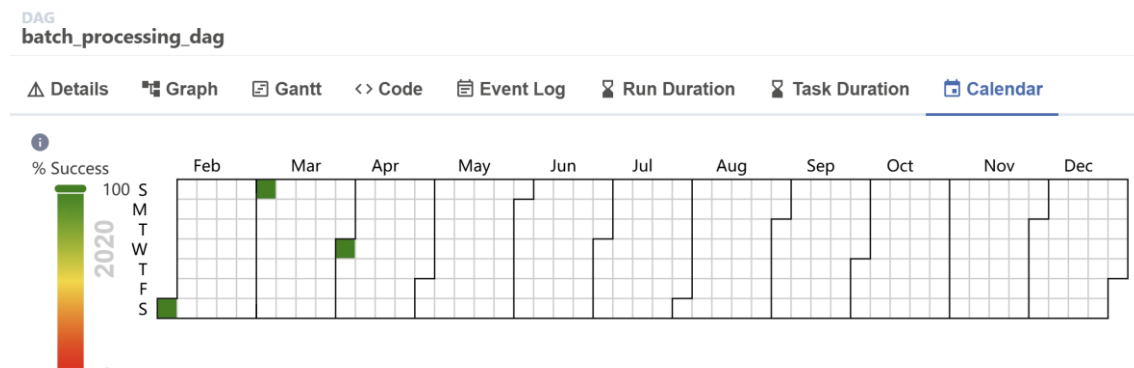
Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

▼ Completed Applications (2)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20250618084228-0001	TransformLoad	9	1536.0 MiB		2025/06/18 08:42:28	default	FINISHED	2.9 min
app-20250618083632-0000	Extract and Clean Data	9	1536.0 MiB		2025/06/18 08:36:32	default	FINISHED	5.6 min

Hình 3.6. Kết quả xử lý dữ liệu trên Spark

Trên giao diện Airflow, DAG `batch_processing_dag` đã được thực thi định kỳ theo từng tháng, thể hiện qua lịch chạy liên tục trong 3 tháng đầu năm 2020. Mỗi lần chạy hoàn tất thành công, xác nhận hệ thống hoạt động ổn định và có khả năng xử lý dữ liệu định kỳ tự động.



**Hình 3.7.** Kết quả thực thi định kỳ bằng Airflow

Hình 3.8 là kết quả của việc lưu trữ dữ liệu trên HDFS. Dữ liệu đã được lưu trữ thành công trên HDFS được phân vùng lưu trữ theo từng năm, tháng và ngày (`day=1, day=2, ..., day=31`). Việc phân vùng dữ liệu theo ngày giúp tối ưu hiệu suất truy xuất trong các truy vấn Spark sau này, đồng thời chứng minh pipeline ETL đã thực hiện đầy đủ bước lưu trữ dữ liệu thô theo cấu trúc phân vùng đúng chuẩn.

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities

Browse Directory

/cleaned\_data/year=2020/month=1

Go!

Show

25

entries

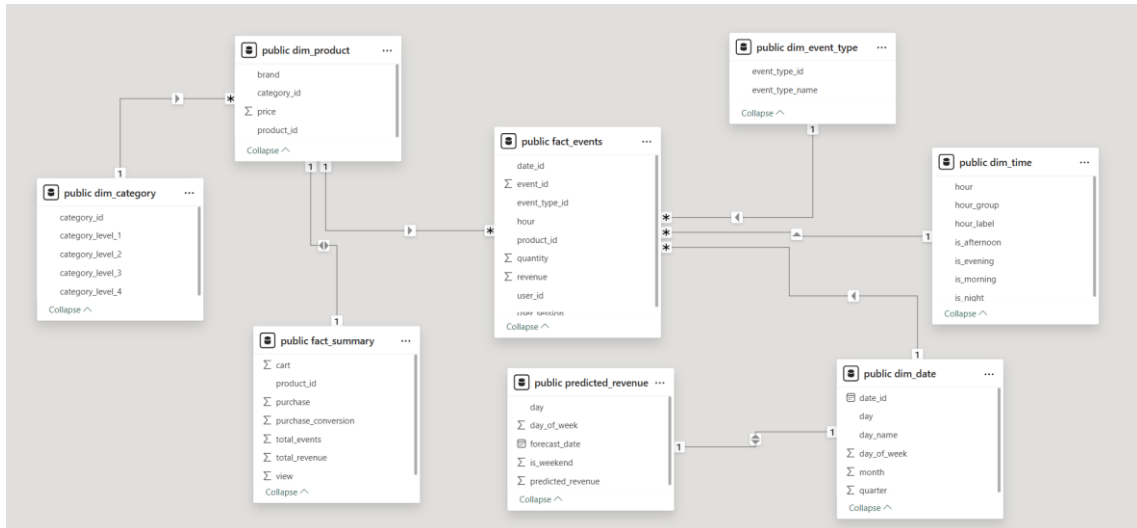
Search:

<input type="checkbox"/>	<div></div> Permission	<div></div> Owner	<div></div> Group	<div></div> Size	<div></div> Last Modified	<div></div> Replication	<div></div> Block Size	<div></div> Name	<div></div>
<input type="checkbox"/>	drwxr-xr-x	default	supergroup	0 B	Jun 18 12:22	0	0 B	day=1	<div></div>
<input type="checkbox"/>	drwxr-xr-x	default	supergroup	0 B	Jun 18 12:23	0	0 B	day=10	<div></div>
<input type="checkbox"/>	drwxr-xr-x	default	supergroup	0 B	Jun 18 12:23	0	0 B	day=11	<div></div>
<input type="checkbox"/>	drwxr-xr-x	default	supergroup	0 B	Jun 18 12:23	0	0 B	day=12	<div></div>
<input type="checkbox"/>	drwxr-xr-x	default	supergroup	0 B	Jun 18 12:23	0	0 B	day=13	<div></div>
<input type="checkbox"/>	drwxr-xr-x	default	supergroup	0 B	Jun 18 12:23	0	0 B	day=14	<div></div>
<input type="checkbox"/>	drwxr-xr-x	default	supergroup	0 B	Jun 18 12:23	0	0 B	day=15	<div></div>
<input type="checkbox"/>	drwxr-xr-x	default	supergroup	0 B	Jun 18 12:24	0	0 B	day=16	<div></div>
<input type="checkbox"/>	drwxr-xr-x	default	supergroup	0 B	Jun 18 12:24	0	0 B	day=17	<div></div>
<input type="checkbox"/>	drwxr-xr-x	default	supergroup	0 B	Jun 18 12:24	0	0 B	day=18	<div></div>
<input type="checkbox"/>	drwxr-xr-x	default	supergroup	0 B	Jun 18 12:24	0	0 B	day=19	<div></div>
<input type="checkbox"/>	drwxr-xr-x	default	supergroup	0 B	Jun 18 12:22	0	0 B	day=20	<div></div>
<input type="checkbox"/>	drwxr-xr-x	default	supergroup	0 B	Jun 18 12:24	0	0 B	day=2	<div></div>

**Hình 3.8.** Lưu trữ dữ liệu trên HDFS

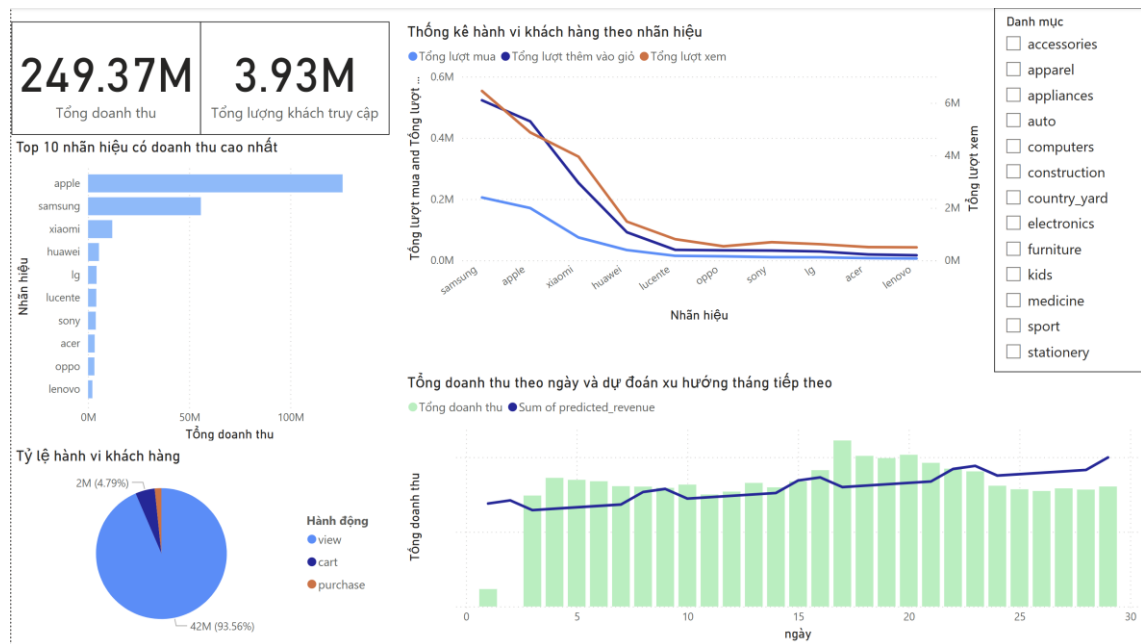
Hình 3.9 mô tả dữ liệu được xây dựng trong Power BI theo kiến trúc star schema, với bảng trung tâm `fact_events` liên kết đến các bảng chiều như

dim\_product, dim\_category, dim\_event\_type, dim\_date, và dim\_time. Ngoài ra, hai bảng fact\_summary và predicted\_revenue lần lượt lưu trữ dữ liệu tổng hợp hành vi người dùng và kết quả dự đoán doanh thu theo ngày. Cấu trúc này giúp hệ thống truy xuất dữ liệu hiệu quả, hỗ trợ lọc và phân tích đa chiều trong các biểu đồ dashboard.



**Hình 3.9.** Mô hình dữ liệu trong Power BI sử dụng kiến trúc star schema

Hình 3.10 là kết quả của báo cáo phân tích từ dữ liệu biến đổi. Dashboard được xây dựng bằng Power BI, hiển thị các chỉ số chính như tổng doanh thu (249.37M), tổng lượt truy cập (3.93M), top 10 thương hiệu có doanh thu cao nhất, và tỷ lệ hành vi người dùng. Biểu đồ đường thể hiện xu hướng xem, thêm giỏ hàng và mua theo từng thương hiệu. Biểu đồ cột phía dưới kết hợp đường dự đoán hiển thị doanh thu thực tế và xu hướng tháng tiếp theo theo từng ngày. Bộ lọc bên phải cho phép phân tích chi tiết theo danh mục sản phẩm.



Hình 3.10. Dashboard phân tích hành vi và doanh thu trong Power BI

## CHƯƠNG 4. KẾT LUẬN

### 4.1 Kết luận

Như vậy, mini-project đã xây dựng thành công một hệ thống phân tích dữ liệu thương mại điện tử, từ khâu xử lý dữ liệu thô đến trực quan hóa trên dashboard. Toàn bộ quy trình ETL được tự động hóa bằng Airflow, dữ liệu được lưu trữ trên HDFS và xử lý bằng Spark, sau đó được tổng hợp và kết nối vào Power BI để hiển thị các chỉ số quan trọng như doanh thu, hành vi khách hàng, và dự đoán xu hướng theo thời gian. Hệ thống có thể hỗ trợ các doanh nghiệp trong việc theo dõi thương hiệu bán chạy, tỷ lệ chuyển đổi hành vi và phân tích hiệu suất kinh doanh theo từng tháng.

Thông qua việc thực hiện đề tài, em đã có cơ hội làm việc thực tế với các công nghệ dữ liệu hiện đại như Apache Spark, Airflow, Hadoop HDFS, PostgreSQL và Power BI. Em cũng nâng cao đáng kể kỹ năng đọc hiểu tài liệu, triển khai các ứng dụng lên Docker và hoàn thiện kỹ năng viết báo cáo học thuật phục vụ cho học tập cũng như công việc sau này.

### 4.2 Hướng phát triển trong tương lai

Trong tương lai, hệ thống có thể được mở rộng theo nhiều hướng nhằm tăng tính ứng dụng thực tiễn và độ sâu phân tích.

Thứ nhất, tích hợp cơ chế thu thập dữ liệu định kỳ hoặc theo thời gian thực từ các hệ thống vận hành sẽ giúp dashboard luôn phản ánh chính xác trạng thái hiện tại, hỗ trợ ra quyết định kịp thời.

Thứ hai, mở rộng nguồn dữ liệu đầu vào như dữ liệu hồ sơ người dùng, phản hồi đánh giá, hoặc dữ liệu từ chiến dịch tiếp thị sẽ giúp phân tích toàn diện hơn về hành vi khách hàng và hiệu quả kinh doanh.

Thứ ba, hệ thống có thể tận dụng dữ liệu thô đã lưu trữ trên HDFS để phục vụ các bài toán học máy như dự đoán doanh thu, phân loại người dùng, phát hiện bất thường... Từ đó không chỉ dừng lại ở phân tích mô tả mà còn tiến tới phân tích dự đoán và ra quyết định thông minh.

Đây là những bước quan trọng để hệ thống có thể phát triển thành một nền tảng phân tích dữ liệu quy mô lớn, ứng dụng được trong thực tế doanh nghiệp.

## TÀI LIỆU THAM KHẢO

- [1] "Apache Airflow," [Online]. Available: <https://airflow.apache.org/docs/>.
- [2] "Hadoop HDFS," [Online]. Available: <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>.
- [3] "Apache Spark," [Online]. Available: <https://spark.apache.org/docs/latest/>.
- [4] "PostgreSQL," [Online]. Available: <https://www.postgresql.org/docs/>.
- [5] "Power BI," [Online]. Available: <https://learn.microsoft.com/en-us/power-bi/>.