



ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY



KHOA TOÁN - TIN
Faculty of Mathematics and Informatics

TRANSPORTATION Delivery Center

Kho dữ liệu và kinh doanh thông minh

Giảng viên hướng dẫn: ThS. Nguyễn Danh Tú

Đỗ Trung Quân	20216873
Nguyễn Minh Dương	20216917
Nguyễn Thị Nhã Linh	20210526
Hoàng Thị Ngân	20216860
Khổng Thị Hoài Phương	20216871

Nhóm thực hiện: Nhóm 17 - Lớp 150333 - Học kỳ 2023.2

Ngày 21 tháng 6 năm 2024


Mục lục

Lời mở đầu	3
1 Tổng quan về Data Warehouse	5
1.1 Khái niệm Data Warehouse	5
1.2 Phân loại Data Warehouse	5
1.3 Kiến trúc Data Warehouse	6
1.3.1 Quá trình phát triển	6
1.3.2 Thuộc tính	8
1.3.3 Phân loại kiến trúc Data Warehouse	9
1.4 Mô hình dữ liệu đa chiều	11
1.4.1 Khái niệm	11
1.4.2 Cách hoạt động	11
1.4.3 Phân loại OLAP	12
1.4.4 Lập mô hình dữ liệu trong OLAP	12
1.5 Ưu, nhược điểm của Data Warehouse	14
1.5.1 Ưu điểm của kho dữ liệu	14
1.5.2 Nhược điểm của kho dữ liệu	14
2 Tổng quan về Business Intelligence	15
2.1 Khái niệm Business Intelligence	15
2.2 Các thành phần của Business Intelligence	15
2.3 Các hoạt động chính của Business Intelligence	16
2.4 Lợi ích của Business Intelligence	16
2.5 Công cụ trực quan hóa dữ liệu Power BI	17
2.5.1 Giới thiệu chung	17
2.5.2 Kiến trúc Power BI	18
2.5.3 Các chức năng vượt trội của Power BI	18
3 Ứng dụng DW&BI trong Transportation	19
3.1 Khảo sát	19
3.1.1 Tổng quan về dịch vụ vận tải	19
3.1.2 Quy trình nghiệp vụ	20
3.1.3 Yêu cầu phân tích	25
3.1.4 Quy mô dữ liệu	29
3.1.5 ERD hệ thống OLTP	36
3.2 Phân tích và thiết kế	37
3.2.1 Khám phá dữ liệu	37
3.2.2 Kiến trúc Data Warehouse	48
3.2.3 Nội dung ETL	49
3.2.4 Hệ thống chiều khái niệm (Voi Dimension)	87
3.2.5 Data Model	89
3.3 Xây dựng báo cáo trực quan	95
3.3.1 Báo cáo về đơn đặt hàng	95
3.3.2 Báo cáo về đơn giao hàng	96
3.3.3 Báo cáo về doanh thu	97
3.3.4 Báo cáo về chi phí	98

3.3.5	Báo cáo về khoảng cách giao vận	99
3.3.6	Báo cáo về thời gian giao vận	100
3.4	Tổng kết	101
3.4.1	Những nội dung đã thực hiện được	101
3.4.2	Những hạn chế cần khắc phục	101
3.4.3	Hướng phát triển và bài học rút ra	102
	Kết luận	103
	Tài liệu tham khảo	103

Lời mở đầu

Trước sự bùng nổ của dữ liệu và sự phát triển không ngừng của công nghệ thông tin, vai trò của kho dữ liệu và kinh doanh thông minh ngày càng trở nên quan trọng và cần thiết trong các doanh nghiệp hiện đại. Đây không chỉ là nền tảng cung cấp kiến thức về cách tổ chức và quản lý dữ liệu một cách hiệu quả, mà còn mở rộng sang việc áp dụng kho dữ liệu để đưa ra các quyết định chiến lược thông minh trong kinh doanh.

Kho dữ liệu không chỉ đơn giản là một nơi lưu trữ thông tin mà còn là còn có thể hỗ trợ cho việc phân tích và dự đoán xu hướng thị trường, từ đó giúp doanh nghiệp nắm bắt cơ hội và đối phó với thách thức trong một môi trường kinh doanh thay đổi nhanh chóng. Trong học phần "Kho dữ liệu và kinh doanh thông minh", chúng em đã được học về những công cụ và kỹ năng cần thiết để xây dựng và quản lý các hệ thống kho dữ liệu hiệu quả, đồng thời khai thác dữ liệu để tối ưu hóa quyết định chiến lược trong các tổ chức.

Với sự hướng dẫn của thầy Nguyễn Danh Tú, nhóm chúng em đã thực hiện một dự án về chủ đề "Transportation", cụ thể là khám phá, phân tích dữ liệu của ngành dịch vụ vận tải. Qua quá trình nghiên cứu và áp dụng các kỹ thuật kho dữ liệu và kinh doanh thông minh, chúng em đã phần nào nắm bắt được các xu hướng quan trọng và đưa ra những đề xuất cụ thể nhằm tối ưu hóa hoạt động và cải thiện hiệu quả trong lĩnh vực này. Dưới đây là bài báo cáo về những gì chúng em đã làm được trong việc phân tích dữ liệu của ngành dịch vụ vận tải, mong thầy và các bạn đọc và đóng góp ý kiến để chúng em có thể hoàn thiện bài báo cáo của nhóm mình.

Hà Nội, Ngày 21 tháng 6 năm 2024
Nhóm 17

Bảng đánh giá thành viên nhóm 17

BẢNG ĐÁNH GIÁ THÀNH VIÊN

MÔN HỌC: Kho dữ liệu và kinh doanh thông minh

HỌ VÀ TÊN:	Đỗ Trung Quân
LỚP: (Ví dụ: K63 - Toán Tin 1)	K66 - Toán Tin 02
NHÓM: (Ví dụ: N2)	N17

STT	Tên thành viên	Làm tốt phần việc được giao Min 1 điểm; Max: 5 điểm	Liên hệ được khi cần	Khả năng đóng góp sáng kiến, ý kiến cho hoạt động nhóm	Sẵn sàng giúp đỡ	Đóng góp chung vào kết quả của nhóm	Tổng điểm
1	Đỗ Trung Quân	5	5	5	5	5	25
2	Khổng Thị Hoài Phương	5	5	5	5	5	25
3	Hoàng Thị Ngân	5	5	5	5	5	25
4	Nguyễn Minh Dương	5	5	5	5	5	25
5	Nguyễn Thị Nhã Linh	5	5	5	5	5	25

Chú ý: Bảng đánh giá này yêu cầu trưởng nhóm phải làm

Điểm đánh giá của mỗi thành viên trong nhóm phải có tối thiểu 4 bậc điểm

Điểm từ 1 đến 5 (Cao nhất là 5, thấp nhất là 1)

Bảng đánh giá của giảng viên

Đề tài	SL thành viên	Tuần trình bày	Nội dung	Đánh giá										Mức độ thu hút	Điểm bô sung	
				Khảo sát Quy trình nghiệp vụ	Require ment	Hệ thống qu ý mô đồ liệu	ERD	OLTP	Phân tích & thiết kế	Data Explorati on	Kiến trúc Dataware house	ETL	Dimensio n	Data model OLAP	Xây dựng chương trình	Dashboard bài học tổng kết
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Transportation		Tuần 13		5	4.5	4.5	5	5	5	4.5	4.5	5	5	4		2

1.1 Khái niệm Data Warehouse

Data warehouse: là quy trình thu thập và quản lý dữ liệu từ nhiều nguồn khác nhau để cung cấp thông tin chi tiết có ý nghĩa về doanh nghiệp. Kho dữ liệu thường được sử dụng để kết nối và phân tích dữ liệu kinh doanh từ các nguồn không đồng nhất. Chúng lưu trữ dữ liệu lịch sử và hiện tại ở một nơi duy nhất được sử dụng để tạo báo cáo phân tích cho người lao động trong toàn doanh nghiệp. Kho dữ liệu là cốt lõi của hệ thống BI được xây dựng để phân tích và báo cáo dữ liệu.

a. Hướng chủ đề (Subject – Oriented):

- Được tổ chức quanh các chủ đề như: customer, product, sales.
- Tập trung bài việc mô hình và phân tích dữ liệu cho việc ra quyết định.
- Cung cấp một góc nhìn đơn giản và xúc tích quanh một chủ đề cụ thể bằng cách loại bỏ các dữ liệu không hữu dụng trong tiến trình hỗ trợ quyết định.

b. Tích hợp (Integrated):

- Tích hợp dữ liệu từ nhiều nguồn dữ liệu không đồng nhất (cơ sở dữ liệu, các cấu trúc file hay các bản ghi giao dịch trực tuyến).
- Áp dụng các kỹ thuật làm sạch và tích hợp dữ liệu. (Đảm bảo sự nhất quán giữa các nguồn dữ liệu trong việc đặt tên, cấu trúc mã hóa, các thuộc tính đo đạc ...; Chuyển đổi dữ liệu khi thu thập dữ liệu.)

c. Dữ liệu theo thời gian (Time Variant):

- Dữ liệu trong DW phải thống nhất theo thời gian, từ đó làm tăng quy mô dữ liệu lên đáng kể so với hệ thống tác nghiệp (từ 5 tới 10 năm trong khi các hệ thống hoạt động quản lý dữ liệu chỉ từ 60 tới 90 ngày).

d. Bền vững (Non-Volatile):

- Lưu trữ tách biệt với cơ sở dữ liệu tác nghiệp.
- Không xảy ra việc sửa chữa dữ liệu trong môi trường data warehouse (Không đòi hỏi xử lý giao dịch, không phục và cơ chế điều khiển truy cập đồng thời Data Warehouse là cốt lõi của hệ thống BI được xây dựng để phân tích và báo cáo dữ liệu.)

1.2 Phân loại Data Warehouse

Ba loại Data Warehouse chính là:

a. Data Warehouse doanh nghiệp (Enterprise Data Warehouse):

Data Warehouse doanh nghiệp hay còn gọi kho dữ liệu doanh nghiệp là một kho tập trung. Chức năng cung cấp dịch vụ hỗ trợ quyết định trên toàn doanh nghiệp. Ngoài ra cung cấp một cách tiếp cận thống nhất để tổ chức và đại diện dữ liệu. Và thêm nữa là cung cấp khả năng phân loại dữ liệu theo chủ đề và cấp quyền truy cập theo các bộ phận đó.

b. Kho lưu trữ dữ liệu hoạt động (Operational Data Store - ODS):

Kho lưu trữ dữ liệu hoạt động không có gì ngoài kho lưu trữ dữ liệu cần thiết khi cả Data Warehouse và hệ thống OLTP không hỗ trợ các tổ chức báo cáo nhu cầu. Trong ODS, kho dữ liệu được làm mới theo thời gian. Do đó, nó được ưa thích rộng rãi cho các hoạt động thường ngày như lưu trữ hồ sơ của nhân viên.

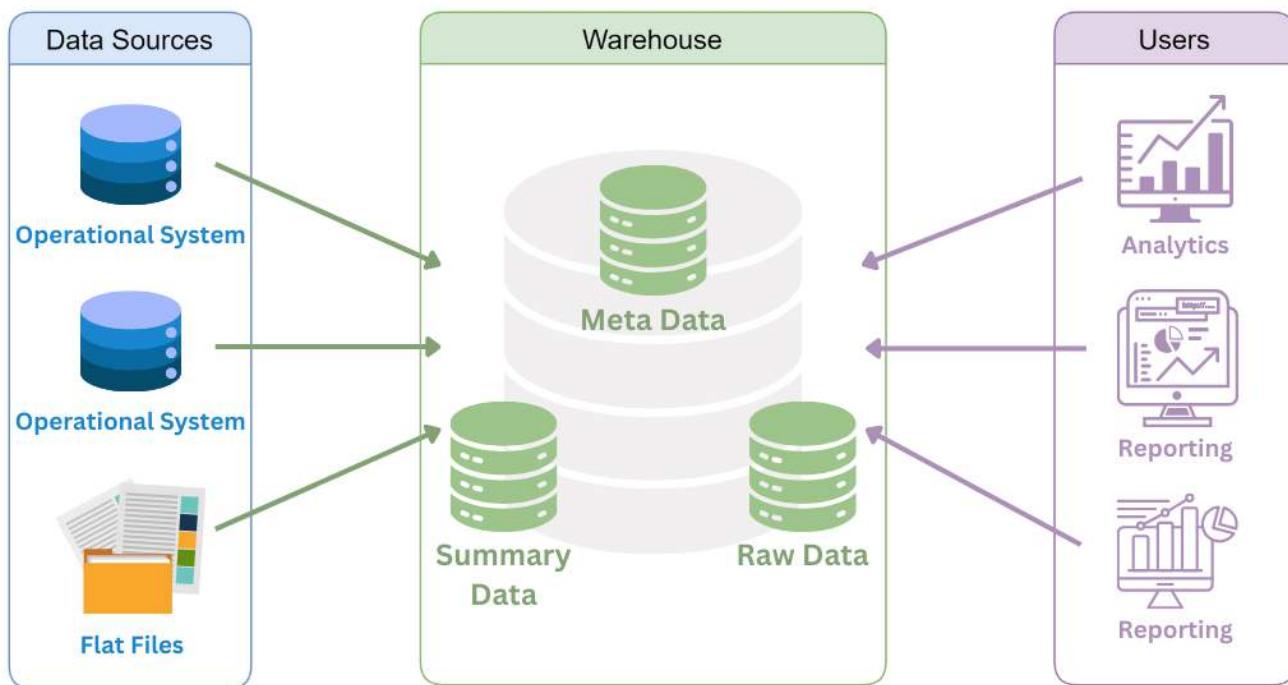
c. Data Mart:

Một Data Mart là một tập hợp con của Data Warehouse, được thiết kế đặc biệt cho một ngành kinh doanh cụ thể, chẳng hạn như bán hàng, tài chính, bán hàng hoặc tài chính. Có Data Mart độc lập và Data Mart phụ thuộc.

1.3 Kiến trúc Data Warehouse

1.3.1 Quá trình phát triển

a. Data Warehouse Architecture: Basic



- Operational System là một phương pháp được sử dụng trong kho dữ liệu để chỉ một hệ thống được sử dụng để xử lý các giao dịch hàng ngày của một tổ chức.
- Flat Files là một hệ thống tệp trong đó dữ liệu giao dịch được lưu trữ và mọi tệp trong hệ thống phải có một tên khác.
- Meta Data là tập hợp dữ liệu xác định và cung cấp thông tin về dữ liệu khác.

Dữ liệu meta được sử dụng trong Kho dữ liệu cho nhiều mục đích khác nhau, bao gồm:

Dữ liệu meta tóm tắt thông tin cần thiết về dữ liệu, có thể giúp việc tìm kiếm và làm việc với các trường hợp dữ liệu cụ thể dễ tiếp cận hơn. Ví dụ: tác giả, bản dựng dữ liệu và dữ liệu đã thay đổi và kích thước tệp là những ví dụ về siêu dữ liệu tài liệu rất cơ bản.

Siêu dữ liệu được sử dụng để hướng một truy vấn đến nguồn dữ liệu thích hợp nhất.

- Dữ liệu tóm tắt nhẹ nhàng và cao

Khu vực của kho dữ liệu lưu trữ cả các dữ liệu được xác định trước (tổng hợp) nhẹ nhàng và cao do người quản lý kho tạo ra.

Mục tiêu của thông tin tóm tắt là để tăng tốc hiệu suất truy vấn. Bản ghi tóm tắt được cập nhật liên tục khi thông tin mới được tải vào kho.

- End-User access Tools

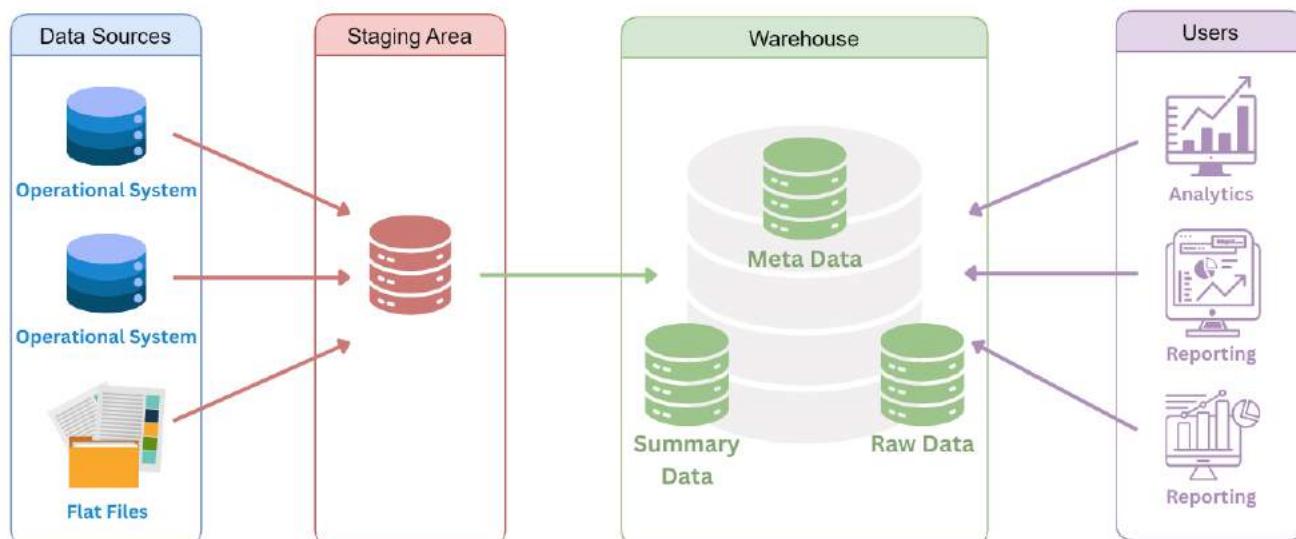
Mục đích chính của kho dữ liệu là cung cấp thông tin cho các nhà quản lý doanh nghiệp để ra quyết định chiến lược. Những khách hàng này tương tác với nhà kho bằng các công cụ truy cập khách hàng cuối.

Ví dụ về một số công cụ truy cập người dùng cuối có thể là:

- Reporting and Query Tools
- Application Development Tools
- Executive Information Systems Tools
- Online Analytical Processing Tools
- Data Mining Tools

Cấu trúc cơ bản cho phép người dùng cuối của kho truy cập trực tiếp vào dữ liệu tóm tắt có nguồn gốc từ hệ thống nguồn và thực hiện phân tích, báo cáo và khai thác dữ liệu đó. Cấu trúc này hữu ích khi các nguồn dữ liệu xuất phát từ cùng loại hệ thống cơ sở dữ liệu.

b. Data Warehouse Architecture: With Staging Area

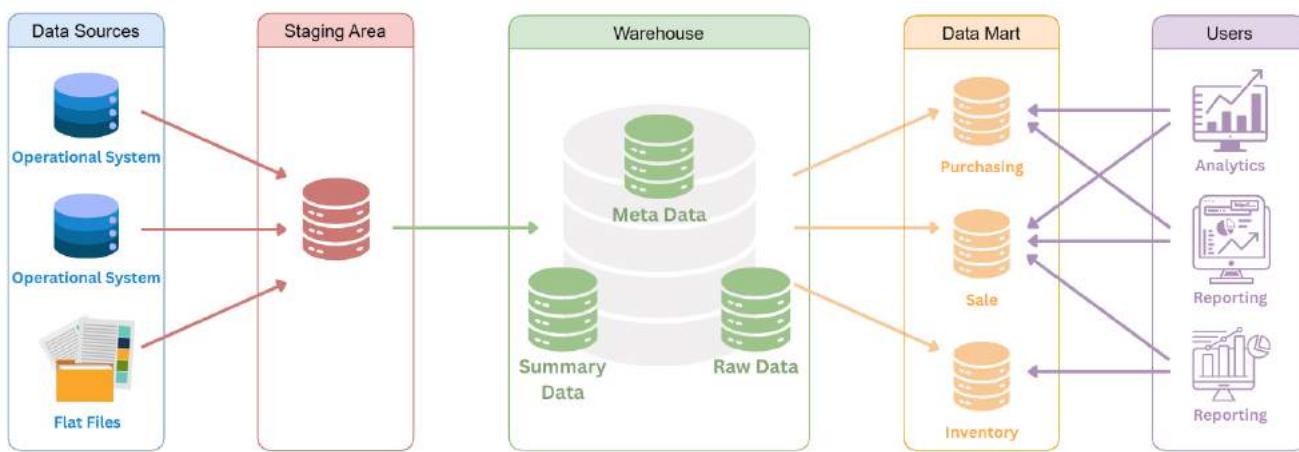


Dữ liệu cần được làm sạch và xử lý trước khi đưa vào kho. Lúc này nó sẽ được đưa vào khu vực tổ chức.

Khu vực tổ chức kho dữ liệu (Staging Area) là một vị trí tạm thời, nơi một bản ghi từ hệ thống nguồn được sao chép.

Khu vực tổ chức đơn giản hóa việc làm sạch và hợp nhất thông tin từ những nguồn dữ liệu khác nhau với nhiều loại và định dạng dữ liệu khác nhau. Khu vực này sẽ chuyển đổi dữ liệu thành định dạng có cấu trúc tóm tắt để dễ truy vấn hơn bằng các công cụ phân tích và báo cáo.

c. Data Warehouse Architecture: With Staging Area and Data Marts



Để có thể muôn tùy chỉnh kiến trúc nhà kho của mình cho nhiều nhóm trong tổ chức, ta thêm các ổ chứa dữ liệu.

Data Mart là một phần của kho dữ liệu, tập trung vào một mảng cụ thể của doanh nghiệp như bán hàng, tài chính, hay sản xuất. Nó lưu trữ dữ liệu tóm tắt và hỗ trợ các phân tích chuyên sâu liên quan đến lĩnh vực đó. Ví dụ, Data Mart có thể giúp nhà phân tích tài chính dễ dàng truy vấn và dự đoán hành vi khách hàng dựa trên dữ liệu bán hàng.

Data Mart giúp việc phân tích dễ dàng hơn bằng cách điều chỉnh dữ liệu cụ thể để đáp ứng nhu cầu của người dùng cuối.

1.3.2 Thuộc tính

- **Separation:** Quá trình xử lý phân tích và giao dịch phải càng xa nhau càng tốt.
- **Scalability:** Kiến trúc phần cứng và phần mềm phải đơn giản để xử lý khối lượng dữ liệu ngày càng lớn và số lượng yêu cầu truy vấn tăng dần từ người dùng mà không gặp trở ngại.
- **Extensibility:** Kiến trúc phải có thể thực hiện các hoạt động và công nghệ mới mà không cần thiết kế lại toàn bộ hệ thống.
- **Security:** Việc giám sát các truy cập là cần thiết vì dữ liệu chiến lược được lưu trữ trong các kho dữ liệu.
- **Administerability:** Quản lý Kho dữ liệu không nên phức tạp.

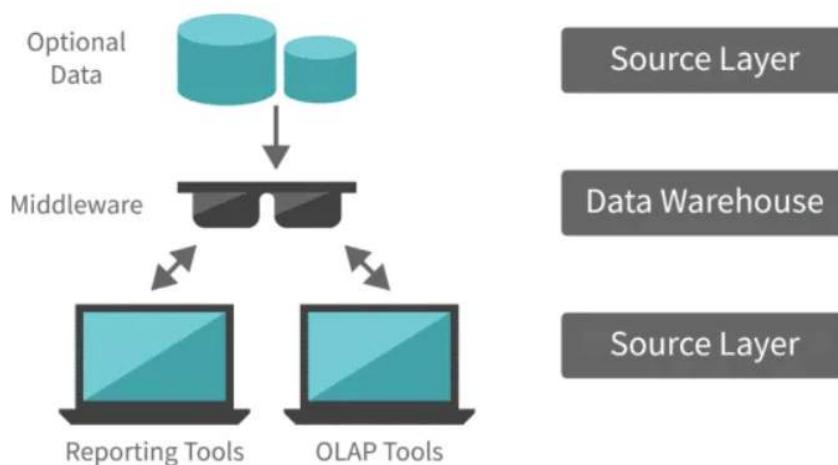
1.3.3 Phân loại kiến trúc Data Warehouse

a. Single-Tier Architecture

Kiến trúc một tầng không được sử dụng định kỳ trong thực tế. Mục đích của nó là giảm thiểu lượng dữ liệu được lưu trữ để đạt được mục tiêu này; nó loại bỏ dư thừa dữ liệu.

Hình cho thấy lớp duy nhất có sẵn về mặt vật lý là lớp nguồn. Trong phương pháp này, kho dữ liệu là ảo. Điều này có nghĩa là kho dữ liệu được thực hiện dưới dạng một cái nhìn đa chiều về dữ liệu hoạt động được tạo bởi phần mềm trung gian cụ thể hoặc một lớp xử lý trung gian.

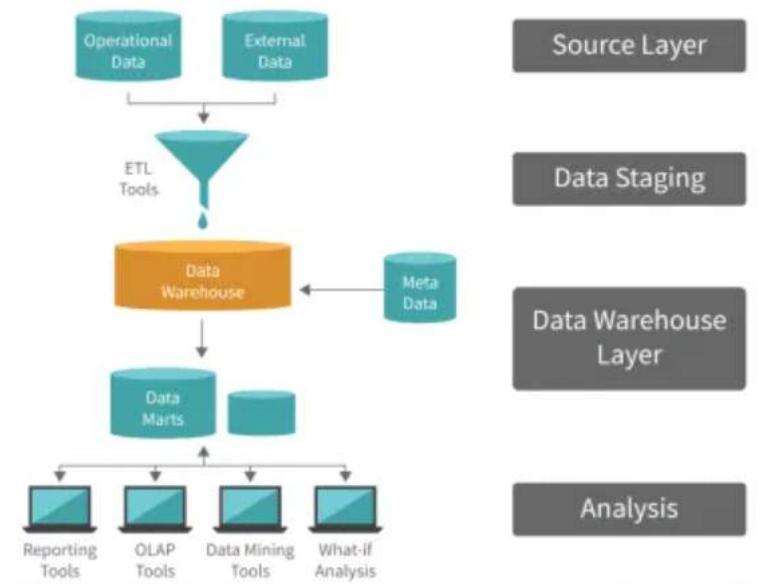
Single-Tier Data Warehouse Architecture



b. Two-Tier Architecture

Yêu cầu phân tách đóng một vai trò thiết yếu trong việc xác định kiến trúc hai tầng cho hệ thống kho dữ liệu, như thể hiện trong hình:

Two-Tier Data Warehouse Architecture



Mặc dù nó thường được gọi là kiến trúc hai lớp để làm nổi bật sự tách biệt giữa các nguồn có sẵn vật lý và kho dữ liệu, trên thực tế, bao gồm bốn giai đoạn luồng dữ liệu tiếp theo:

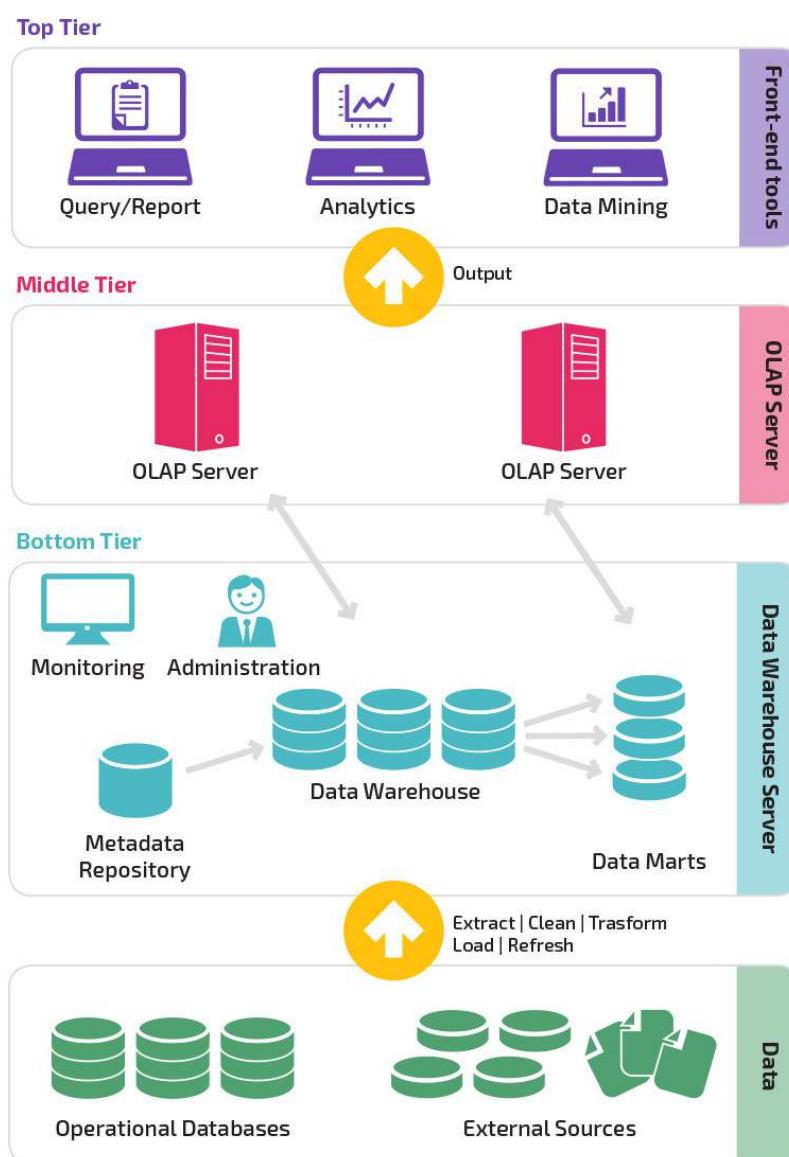
Lớp nguồn: Một hệ thống kho dữ liệu sử dụng một nguồn dữ liệu không đồng nhất. Dữ liệu đó ban đầu được lưu trữ vào cơ sở dữ liệu quan hệ của công ty hoặc cơ sở dữ liệu kế thừa, hoặc nó có thể đến từ một hệ thống thông tin bên ngoài các bức tường của công ty.

Giai đoạn dữ liệu: Dữ liệu được lưu trữ vào nguồn phải được trích xuất, làm sạch để loại bỏ sự mâu thuẫn và lấp đầy khoảng trống, đồng thời tích hợp để hợp nhất các nguồn không đồng nhất thành một lược đồ tiêu chuẩn. Công cụ trích xuất, chuyển đổi và tải (ETL) có tên như vậy có thể kết hợp các schemata không đồng nhất, trích xuất, chuyển đổi, làm sạch, xác thực, lọc và tải dữ liệu nguồn vào kho dữ liệu.

Lớp Kho dữ liệu: Thông tin được lưu vào một kho lưu trữ riêng lẻ tập trung hợp lý: kho dữ liệu. Các kho dữ liệu có thể được truy cập trực tiếp, nhưng nó cũng có thể được sử dụng như một nguồn để tạo các data mart, một phần sao chép nội dung kho dữ liệu và được thiết kế cho các bộ phận doanh nghiệp cụ thể. Kho lưu trữ siêu dữ liệu lưu trữ thông tin về nguồn, thủ tục truy cập, tổ chức dữ liệu, người dùng, lược đồ trung tâm dữ liệu, v.v.

Phân tích: Trong lớp này, dữ liệu tích hợp được truy cập hiệu quả và linh hoạt để đưa ra báo cáo, phân tích động thông tin và mô phỏng các tình huống kinh doanh giả định. Nó phải có tính năng điều hướng thông tin tổng hợp, trình tối ưu hóa truy vấn phức tạp và GUI thân thiện với khách hàng.

c. Three-Tier Architecture



Kiến trúc ba tầng bao gồm lớp nguồn (chứa nhiều hệ thống nguồn), lớp đối chiếu và lớp kho dữ liệu (chứa cả kho dữ liệu và ổ chứa dữ liệu). Lớp đối chiếu nằm giữa dữ liệu nguồn và kho dữ liệu.

Ưu điểm chính của lớp đối chiếu là nó tạo ra một mô hình dữ liệu tham chiếu tiêu chuẩn cho toàn bộ doanh nghiệp. Đồng thời, nó tách biệt các vấn đề khai thác và tích hợp dữ liệu nguồn với các vấn đề của tổng thể kho dữ liệu. Trong một số trường hợp, lớp đối chiếu cũng được sử dụng trực tiếp để hoàn thành tốt hơn một số nhiệm vụ hoạt động, chẳng hạn như tạo báo cáo hàng ngày mà không thể chuẩn bị thỏa đáng bằng cách sử dụng các ứng dụng của công ty hoặc tạo luồng dữ liệu để cung cấp các quy trình bên ngoài theo định kỳ để hưởng lợi từ việc làm sạch và tích hợp.

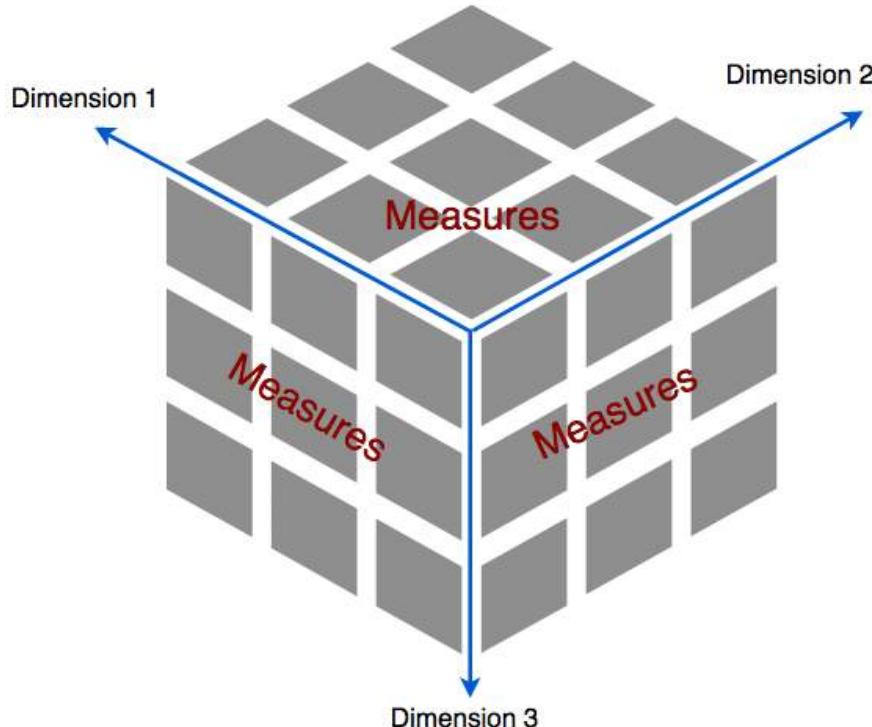
Kiến trúc này đặc biệt hữu ích cho các hệ thống mở rộng, toàn doanh nghiệp. Một nhược điểm của cấu trúc này là không gian lưu trữ tệp bổ sung được sử dụng thông qua lớp điều hòa dư thừa. Nó cũng làm cho các công cụ phân tích xa hơn một chút so với thời gian thực.

1.4 Mô hình dữ liệu đa chiều

1.4.1 Khái niệm

OLAP (Online Analytical Processing) là một phương pháp để phân tích dữ liệu từ một góc nhìn kinh doanh. Nó cho phép người dùng truy cập thông tin từ nhiều nguồn dữ liệu khác nhau và xem nó theo nhiều cách khác nhau. OLAP thường được sử dụng cho các nhiệm vụ như phân tích xu hướng, báo cáo tài chính và dự báo doanh số.

Hệ thống OLAP thường lưu trữ dữ liệu dưới dạng đa chiều, cho phép người dùng xem dữ liệu từ các góc nhìn khác nhau. Ví dụ, một hệ thống OLAP về doanh số có thể cho phép người dùng xem dữ liệu bán hàng theo sản phẩm, vùng miền hoặc khoảng thời gian. OLAP thường được sử dụng kết hợp với các công cụ khai thác dữ liệu và thông tin kinh doanh.



1.4.2 Cách hoạt động

Một hệ thống xử lý phân tích trực tuyến (OLAP) hoạt động bằng cách thu thập, tổ chức, tổng hợp và phân tích dữ liệu theo các bước sau:

1. Máy chủ OLAP thu thập dữ liệu từ nhiều nguồn dữ liệu, bao gồm cơ sở dữ liệu quan hệ và kho dữ liệu.
2. Sau đó, các công cụ trích xuất, chuyển đổi và tải (ETL) làm sạch, tổng hợp, tính toán trước và lưu trữ dữ liệu trong một khối OLAP theo số lượng chiều được chỉ định.
3. Các chuyên viên phân tích kinh doanh sử dụng công cụ OLAP để truy vấn và lập báo cáo từ dữ liệu đa chiều trong khối OLAP.

OLAP sử dụng ngôn ngữ truy vấn đa chiều (MDX) để truy vấn khối OLAP. MDX là một truy vấn, tương tự như SQL, cung cấp một tập các hướng dẫn để thao tác cơ sở dữ liệu.

1.4.3 Phân loại OLAP

Các hệ thống xử lý phân tích trực tuyến (OLAP) hoạt động theo ba cách chính.

1. MOLAP

Xử lý phân tích trực tuyến đa chiều (MOLAP) liên quan đến việc tạo ra một khối dữ liệu đại diện cho dữ liệu đa chiều từ một kho dữ liệu. Hệ thống MOLAP lưu trữ dữ liệu được tính toán trước trong siêu khối. Các kỹ sư dữ liệu sử dụng MOLAP vì loại công nghệ OLAP này cung cấp phân tích tốc độ cao.

2. ROLAP

Thay vì sử dụng một khối dữ liệu, xử lý phân tích trực tuyến quan hệ (ROLAP) cho phép các kỹ sư dữ liệu thực hiện phân tích dữ liệu đa chiều trên một cơ sở dữ liệu quan hệ. Nói cách khác, các kỹ sư dữ liệu sử dụng truy vấn SQL để tìm kiếm và truy xuất thông tin cụ thể dựa trên các chiều yêu cầu. ROLAP phù hợp cho phân tích dữ liệu rộng và chi tiết. Tuy nhiên, ROLAP có hiệu suất truy vấn chậm so với MOLAP.

3. HOLAP

Xử lý phân tích trực tuyến lai (HOLAP) kết hợp MOLAP và ROLAP để mang tới những ưu điểm tốt nhất của cả hai kiến trúc. HOLAP cho phép các kỹ sư dữ liệu nhanh chóng lấy kết quả phân tích từ một khối dữ liệu và trích xuất thông tin chi tiết từ cơ sở dữ liệu quan hệ.

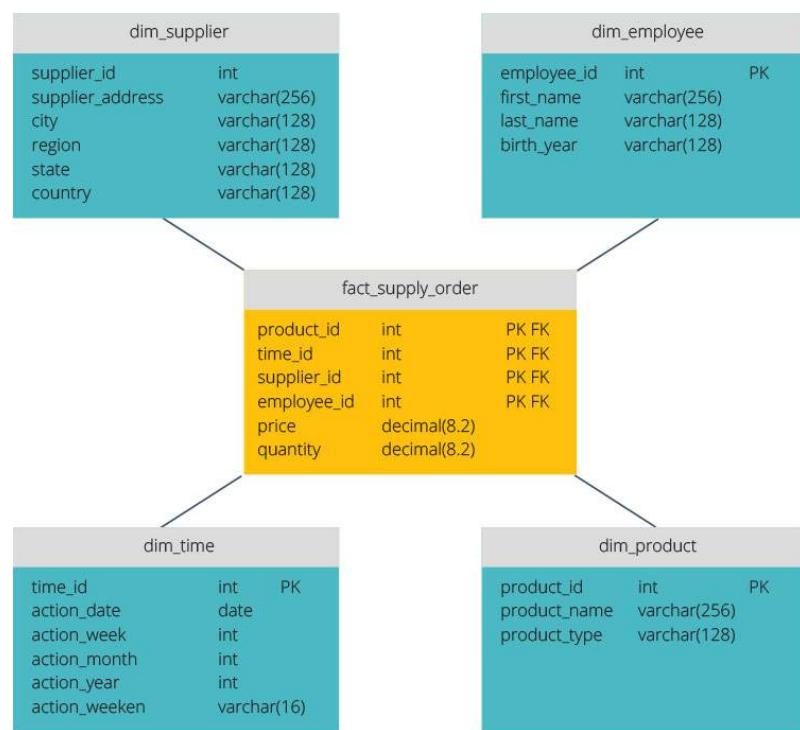
1.4.4 Lập mô hình dữ liệu trong OLAP

Lập mô hình dữ liệu là việc biểu diễn dữ liệu trong kho dữ liệu hoặc cơ sở dữ liệu xử lý phân tích trực tuyến (OLAP). Lập mô hình dữ liệu đóng vai trò rất quan trọng đối với xử lý phân tích trực tuyến quan hệ (ROLAP) vì nó phân tích dữ liệu trực tiếp từ cơ sở dữ liệu quan hệ. Nó lưu trữ dữ liệu đa chiều như một lược đồ ngôi sao hoặc bông tuyết.

Lược đồ ngôi sao

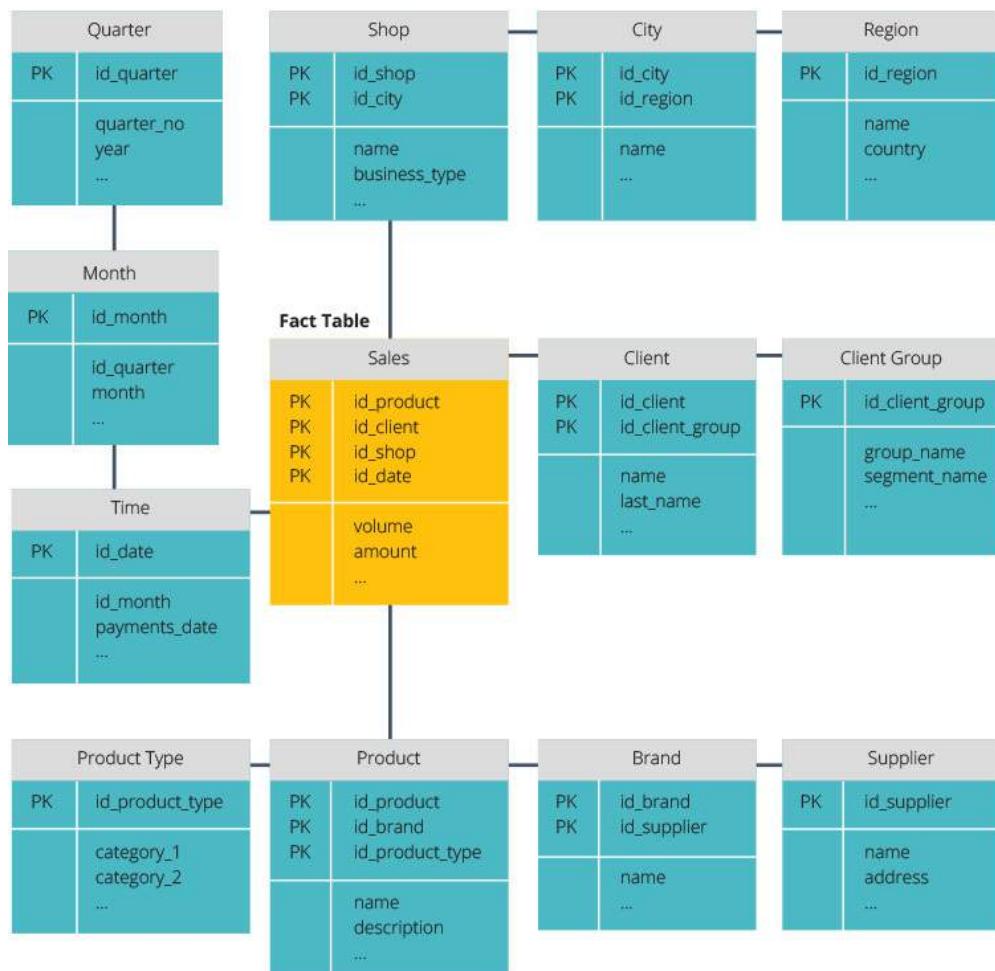
Lược đồ ngôi sao bao gồm một bảng dữ kiện và nhiều bảng thứ nguyên. Bảng dữ kiện là một bảng dữ liệu chứa các giá trị số liên quan đến quy trình kinh doanh và thứ nguyên chứa các giá trị mô tả mỗi thuộc tính trong bảng dữ kiện. Bảng dữ kiện đề cập đến các bảng thứ nguyên với khóa ngoại, chúng là mã định danh duy nhất tương quan với thông tin tương ứng trong bảng thứ nguyên.

Trong lược đồ ngôi sao, một bảng dữ kiện liên kết với một số bảng thứ nguyên khiêm cho mô hình dữ liệu trông giống một ngôi sao.



Lược đồ bông tuyết

Lược đồ bông tuyết là bản mở rộng của lược đồ ngôi sao. Một số bảng thứ nguyên có thể dẫn đến một hoặc nhiều bảng thứ nguyên phụ. Điều này tạo ra một hình dạng giống như bông tuyết khi các bảng thứ nguyên được kết hợp với nhau.



1.5 Ưu, nhược điểm của Data Warehouse

1.5.1 Ưu điểm của kho dữ liệu

a. Hợp nhất dữ liệu

- Loại bỏ việc lưu trữ dữ liệu ở các môi trường khác nhau do lưu trữ đa dạng kiểu dữ liệu.
- Cung cấp một trung tâm dữ liệu duy nhất.

b. Cung cấp thông tin kinh doanh cao hơn báo cáo

- Cầu nối giữa dữ liệu giao dịch khổng lồ và thông tin để ra quyết định.
- Cho phép trả lời các câu hỏi phức tạp.

c. Tăng tốc độ phân tích dữ liệu & phân tích kinh doanh

- Thiết kế mô hình đa chiều tăng tốc độ truy vấn.
- Không mất thời gian thu thập và tiền xử lý dữ liệu.
- Dữ liệu đã xử lý nên kết nối dễ dàng với các công cụ phân tích & BI.

d. Cải thiện quy trình ra quyết định tổng thể

- Có dữ liệu lịch sử.
- Dánh giá được rủi ro.
- Hiểu về nhu cầu khách hàng.
- Có thông tin để cải tiến sản phẩm và dịch vụ.

1.5.2 Nhược điểm của kho dữ liệu

a. Thiếu tính linh hoạt của dữ liệu

- Gặp khó khăn với các định dạng dữ liệu bán cấu trúc và phi cấu trúc như phân tích nhật ký, phát trực tuyến và dữ liệu truyền thông xã hội.
- Khó khăn khi áp dụng máy học (ML) và trí tuệ nhân tạo (AI).

b. Chi phí triển khai và bảo trì lớn

- Chi phí hàng năm của một kho dữ liệu nội bộ với một terabyte dung lượng lưu trữ và 100K truy vấn/tháng là \$450K.
- Kho dữ liệu thường không tinh; nó trở nên lỗi thời và yêu cầu bảo trì thường xuyên.

2.1 Khái niệm Business Intelligence

Thị trường kinh doanh đang ngày càng phát triển, mở rộng một cách nhanh chóng, thị trường phát triển đòi hỏi các công ty phải liên tục cập nhật, trau dồi và đổi mới liên tục để phù hợp với thị trường. Với sự tiến bộ của công nghệ, các công ty, các doanh nghiệp có thể dễ dàng tiếp cận đến khách hàng một cách thông minh và nhanh chóng.

Kinh doanh thông minh (BI) bao gồm các chiến lược và công nghệ được các doanh nghiệp sử dụng để phân tích dữ liệu, thông tin kinh doanh. Công nghệ BI cung cấp các quan điểm lịch sử, hiện tại và dự đoán về hoạt động kinh doanh. Các chức năng phổ biến của công nghệ thông minh kinh doanh bao gồm báo cáo, xử lý phân tích trực tuyến, phân tích, phát triển bảng điều khiển, khai thác dữ liệu, khai thác quy trình, xử lý sự kiện phức tạp, quản lý hiệu suất kinh doanh, đo điểm chuẩn, khai thác văn bản, phân tích dự đoán và phân tích mô tả.

2.2 Các thành phần của Business Intelligence

- **Data Sources (Nguồn dữ liệu):** Thành phần đầu tiên và cũng là một bước quan trọng để có được dữ liệu trong hệ thống chính là thu thập dữ liệu từ nhiều nguồn đa dạng đến từ nhiều định dạng khác nhau như hệ thống quản lý khách hàng (CRM), hệ thống quản trị nhân sự (HRM), khảo sát, thông tin khách hàng trên các nền tảng thương mại, v.v.
- **Data Warehousing (Kho dữ liệu):** Là cơ sở dữ liệu được thiết kế theo mô hình khác với CSDL OLTP thông thường (Online Transaction Processings – OLTP là thiết kế CSDL dành cho việc đọc ghi thường xuyên, lượng dữ liệu cho mỗi lần đọc ghi ít) và là nơi lưu trữ dữ liệu lâu dài của tổ chức. Dữ liệu của DWH chỉ có thể đọc, không được sử dụng để ghi hay update bởi ứng dụng thông thường, nó chỉ được cập nhật/ghi bởi công cụ ETL (Extract Transform Load), công cụ chuyển đổi dữ liệu từ Data Sources vào Data Warehouse.
- **Integrating Server (Tích hợp máy chủ):** Hỗ trợ quá trình vận hành công cụ ELT (viết tắt của Extract, Transform, Load) để trích xuất, chuyển đổi tất cả dữ liệu từ Data sources và sau đó tải dữ liệu vào trong Data warehouse.
- **Analysis Server (Máy chủ phân tích):** Chịu trách nhiệm thực thi các cube được thiết kế dựa trên các chiều dữ liệu và tri thức nghiệp vụ. Cube chịu trách nhiệm nhận dữ liệu đầu vào từ DWH và thực thi theo nghiệp vụ định nghĩa sẵn để trả về kết quả.
- **Reporting Server (Máy chủ báo cáo):** Thực thi các report với output nhận được từ Analysis Server. Đây là nơi quản trị tập trung các report trên nền web, các report này có thể được attach vào ứng dụng web, hay application.
- **Data Mining (Khai thác dữ liệu):** Là quá trình trích xuất thông tin dữ liệu đã qua xử lý (phù hợp với yêu cầu riêng của doanh nghiệp) từ Data Warehouse rồi kết hợp với các thuật toán để đưa ra (hoặc dự đoán) các quyết định có lợi cho việc kinh doanh của doanh nghiệp. Đây là một quá trình quan trọng trong BI, thông thường một doanh nghiệp muốn sử dụng giải pháp BI thường kèm theo về Data Mining.
- **Data Presentation (Trình bày dữ liệu):** Hệ thống BI sẽ xử lý và tổng hợp dữ liệu từ quá trình Data mining để tạo thành biểu đồ/ sơ đồ phục vụ cho việc trình bày đến các nhà hoạch định chính sách và bên ra quyết định.

2.3 Các hoạt động chính của Business Intelligence

- **Hỗ trợ doanh nghiệp đưa ra quyết định (Decision support):** Mục đích mà BI thu thập dữ liệu chính là giúp doanh nghiệp nhận ra những vấn đề còn tồn đọng. Do đó, BI đóng vai trò ảnh hưởng tích cực đến việc đưa ra quyết định chiến lược kinh doanh mà doanh nghiệp đang hướng đến.
- **Truy vấn và báo cáo (Query and reporting):** Từ những dữ liệu đã thu thập được, hệ thống sẽ tự động phân tích, ghi chú và lưu trữ những chi tiết quan trọng. Sau đó, tiến hành xây dựng báo cáo dưới dạng mô hình, cho thấy bức tranh tổng quan và dễ hiểu nhất. Với hoạt động này, người dùng có thể trích xuất thông tin một cách nhanh chóng và dễ dàng.
- **Phân tích thống kê (Statistical analysis):** là hoạt động nhằm phân tích, giải thích dữ liệu để phát hiện ra mẫu và xu hướng, đây là một khâu quan trọng trong việc phân tích dữ liệu.
- **Dự đoán (Forecasting):** BI có thể hỗ trợ doanh nghiệp dự đoán tương lai, trong tất cả mọi loại ngành nghề. Giúp doanh nghiệp chuẩn bị tốt hơn cho các vấn đề, biến động xấu có thể xảy ra, đồng thời định hình chiến lược kinh doanh trong dài hạn.
- **Khai thác dữ liệu (Data Mining):** Khai thác dữ liệu là quá trình thu thập data từ đa dạng các nguồn, sau đó phân tích và tổng hợp chúng thành các thông tin liên quan. Mục đích của khai thác dữ liệu là tìm ra giải pháp để giải quyết cho các vấn đề kinh doanh cụ thể của doanh nghiệp.

2.4 Lợi ích của Business Intelligence

Business Intelligence (BI) mang lại nhiều lợi ích cho các tổ chức bằng cách cho phép họ đưa ra quyết định dựa trên dữ liệu, tối ưu hóa hoạt động và duy trì lợi thế cạnh tranh. Một số lợi ích chính của BI bao gồm:

- **Cải thiện việc ra quyết định:** Bằng cách cung cấp thông tin chi tiết kịp thời và chính xác, BI giúp những người ra quyết định đưa ra những lựa chọn sáng suốt hơn, giảm sự phụ thuộc vào cảm tính hoặc phỏng đoán.
- **Nâng cao hiệu quả hoạt động:** Các công cụ BI có thể xác định sự thiếu hiệu quả, tắc nghẽn hoặc hạn chế tài nguyên, cho phép các tổ chức hợp lý hóa các quy trình, giảm chi phí và tối ưu hóa phân bổ tài nguyên.
- **Tăng doanh thu và lợi nhuận:** Với thông tin chi tiết về sở thích của khách hàng, xu hướng thị trường và hiệu suất bán hàng, các doanh nghiệp có thể điều chỉnh các dịch vụ, chiến lược giá và chiến dịch tiếp thị của mình, cuối cùng là thúc đẩy tăng trưởng doanh thu và lợi nhuận cao hơn.
- **Hiểu khách hàng tốt hơn:** BI cho phép các tổ chức phân tích dữ liệu khách hàng, xác định các mẫu và xu hướng giúp điều chỉnh các sản phẩm, dịch vụ và nỗ lực tiếp thị để đáp ứng nhu cầu của khách hàng và nâng cao sự hài lòng của khách hàng.
- **Lợi thế cạnh tranh:** Bằng cách cung cấp thông tin chi tiết về xu hướng thị trường, hoạt động của đối thủ cạnh tranh và động lực của ngành, BI cho phép doanh nghiệp thích nghi với môi trường luôn thay đổi và duy trì lợi thế cạnh tranh.

- **Dự báo và quản lý rủi ro:** Khả năng dự đoán của BI giúp các tổ chức dự đoán các xu hướng trong tương lai, xác định các rủi ro tiềm ẩn và phát triển các kế hoạch dự phòng, giúp họ chuẩn bị tốt hơn cho những điều không chắc chắn.
- **Văn hóa định hướng:** Triển khai BI khuyến khích văn hóa ra quyết định dựa trên dữ liệu, thúc đẩy sự hợp tác và nâng cao hiệu suất tổng thể của tổ chức.
- **Tuân thủ quy định và báo cáo:** Các công cụ BI có thể tạo các báo cáo chính xác và kịp thời, giúp các tổ chức tuân thủ các yêu cầu quy định và đảm bảo tính minh bạch.
- **Trao quyền cho nhân viên:** Bằng cách cung cấp quyền truy cập vào dữ liệu và thông tin chuyên sâu có liên quan, BI trao quyền cho nhân viên đưa ra quyết định tốt hơn trong vai trò tương ứng của họ, thúc đẩy quyền sở hữu và trách nhiệm giải trình.
- **Đổi mới và tăng trưởng:** Những hiểu biết sâu sắc từ BI có thể châm ngòi cho những ý tưởng mới, cho phép các tổ chức xác định cơ hội đổi mới, mở rộng hoặc đa dạng hóa, thúc đẩy tăng trưởng dài hạn.

2.5 Công cụ trực quan hóa dữ liệu Power BI

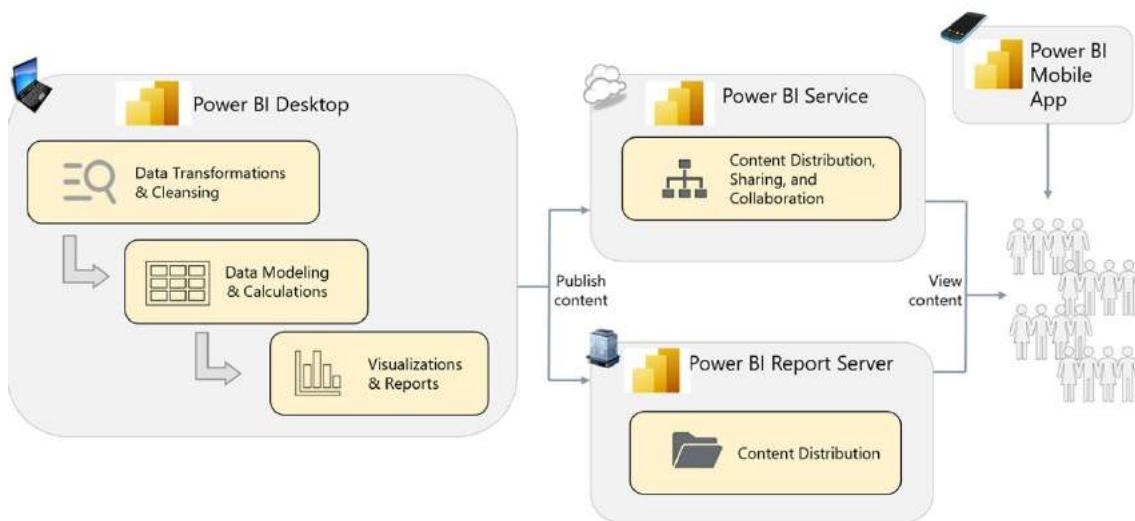
2.5.1 Giới thiệu chung

Power BI là công cụ phân tích và trực quan hóa dữ liệu dành cho lĩnh vực Business Intelligence (BI) của Microsoft. Đây là công cụ thông minh hỗ trợ doanh nghiệp trong việc tạo ra báo cáo quản trị đầy đủ và trực quan, giúp các nhà lãnh đạo có thể đưa ra quyết định chính xác dựa vào kết quả phân tích tình hình kinh doanh.

Power BI có một lịch sử phong phú và phát triển từ năm 2010. Ban đầu, nó được gọi là "Project Crescent" và là một công cụ trực quan hóa dữ liệu tích hợp với SQL Server Reporting Services (SSRS)¹. Sau đó, vào tháng 9 năm 2013, Microsoft giới thiệu Power BI cho Office 365. Phiên bản đầu tiên của Power BI dựa trên các tiện ích dựa trên Microsoft Excel như Power Query, Power Pivot và Power View. Với thời gian, Microsoft đã bổ sung nhiều tính năng khác như câu hỏi và trả lời, kết nối dữ liệu cấp doanh nghiệp và tùy chọn bảo mật thông qua Power BI Gateways². Power BI chính thức ra mắt công chúng vào ngày 24 tháng 7 năm 2015 và đã trở thành một công cụ phổ biến cho việc phân tích dữ liệu và thực hiện các biểu đồ tương tác.

Power BI có thể kết nối với nhiều dịch vụ phần mềm, hoạt động song song, từ đó có thể kết nối nhiều nguồn dữ liệu với nhau và tạo ra mô hình dữ liệu bao gồm các biểu đồ, con số và các thông tin được tự động tính toán chi tiết, liền mạch và được thể hiện một cách trực quan hóa. Mô hình này có thể được chia sẻ với bất cứ ai trong tổ chức, hoặc những người có tài khoản Power BI. Power BI cũng tối ưu hóa mức tiêu thụ dữ liệu Azure, giúp tiết kiệm đáng kể cho các công ty dựa vào Power BI cho nhu cầu báo cáo hàng ngày.

2.5.2 Kiến trúc Power BI



Các Thành phần chính của Power BI là:

- **Power BI Desktop:** Đây là một phần mềm trên hệ điều hành Window, có vai trò xử lý, tập hợp và xây dựng mô hình dữ liệu để trực quan hóa cho các báo cáo.
- **Power BI Apps:** Đây là dạng được sử dụng trên hệ điều hành IOS hoặc Android như điện thoại, máy tính bảng.
- **Power BI Online:** Đây là một dịch vụ lưu trữ dữ liệu đám mây cho phép người dùng có thể lưu trữ báo cáo, dashboard mọi lúc mọi nơi.
- **Power BI Report Server:** Cho phép người dùng có thể xuất báo cáo sau khi hoàn thành thao tác trên hệ thống

2.5.3 Các chức năng vượt trội của Power BI

Chức năng vượt trội của Power BI là tạo ra báo cáo hoặc các dashboard cho doanh nghiệp. Đây là công cụ hữu hiệu cho doanh nghiệp phải xử lý đồng bộ thông tin với số lượng lớn.

So với báo cáo dữ liệu thông thường, Power BI có nhiều điểm vượt trội hơn, cụ thể như sau:

- Cho phép người dùng tổng hợp được kết quả chung từ đa dạng nguồn dữ liệu khác nhau (dạng văn bản: File Excel, File PDF,...; từ các cơ sở lưu trữ và kho dữ liệu; dữ liệu từ platform; dữ liệu từ Azure; dữ liệu trên Online Service;...)
- Khả năng chuyển đổi phân tích dữ liệu lớn với khả năng làm việc từ 8 – 10 triệu dòng dữ liệu một lần.
- Nâng cao trực quan hóa dữ liệu qua mô hình nhờ việc kết hợp dữ liệu từ nhiều nguồn.
- Dùng Biểu thức phân tích (DAX) để phân tích dữ liệu (DAX là biểu thức phân tích vô cùng mạnh mẽ với tốc độ xử lý nhanh, hiệu quả).
- Người dùng có thể tạo lịch trình để cập nhật dữ liệu tự động thay vì tốn thời gian thao tác thủ công.
- Các thông tin dữ liệu kết nối luôn được hệ thống bảo mật tuyệt đối.

3

Ứng dụng DW&BI trong Transportation

3.1 Khảo sát

3.1.1 Tổng quan về dịch vụ vận tải



Dịch vụ Vận tải gồm hoạt động giao nhận, vận chuyển hàng hóa giữa các địa điểm, tỉnh thành trong phạm vi khu vực của một quốc gia hay vùng lãnh thổ. Nó đóng vai trò vô cùng quan trọng trong việc kết nối con người và hàng hóa trên toàn thế giới và giúp thúc đẩy thương mại, tạo điều kiện cho giao lưu văn hóa và nâng cao chất lượng cuộc sống.

Với các trung tâm hoạt động khác nhau trải rộng khắp Brazil, Delivery Center là nền tảng kết nối giữa các hubs, stores đến tay người tiêu dùng, tạo ra một hệ sinh thái lành mạnh để bán hàng hóa và thực phẩm trong ngành bán lẻ Brazil. Họ cung cấp một số đăng ký với hơn 900 nghìn mặt hàng và thực hiện hàng nghìn đơn đặt hàng và giao hàng hàng ngày thông qua một mạng lưới rộng lớn của cửa hàng và đối tác giao hàng trải rộng khắp Brazil. Do số lượng lớn dữ liệu mà họ thu thập từ các hoạt động này, họ đang tập trung vào việc sử dụng dữ liệu để hỗ trợ quyết định kinh doanh và định hướng cho tương lai, thấy rằng việc này có thể là một yếu tố quan trọng để tạo sự khác biệt trên thị trường.

Nhóm khách hàng chính của Delivery Center là: Người tiêu dùng, Nhà bán lẻ và Đối tác giao hàng.

Các dịch vụ của Delivery Center bao gồm:

- Vận chuyển đa dạng các mặt hàng: Thực phẩm, đồ uống, hàng tiêu dùng, hàng điện máy, thiết bị công nghiệp,...

- Giao hàng nhanh chóng và an toàn: Hệ thống vận tải hiện đại, đội ngũ lái xe chuyên nghiệp, đảm bảo hàng hóa được vận chuyển an toàn và đúng thời hạn.
- Theo dõi đơn hàng online: Khách hàng có thể dễ dàng theo dõi tình trạng đơn hàng của mình thông qua hệ thống website hoặc ứng dụng di động.
- Hỗ trợ khách hàng 24/7: Đội ngũ nhân viên tư vấn nhiệt tình, sẵn sàng giải đáp mọi thắc mắc của khách hàng.
- Giá cả cạnh tranh: Cung cấp dịch vụ với mức giá hợp lý, phù hợp với nhu cầu của mọi khách hàng.

3.1.2 Quy trình nghiệp vụ

a. Business Canvas Model



Mô hình kinh doanh của trung tâm giao hàng sẽ được thể hiện qua 9 yếu tố:

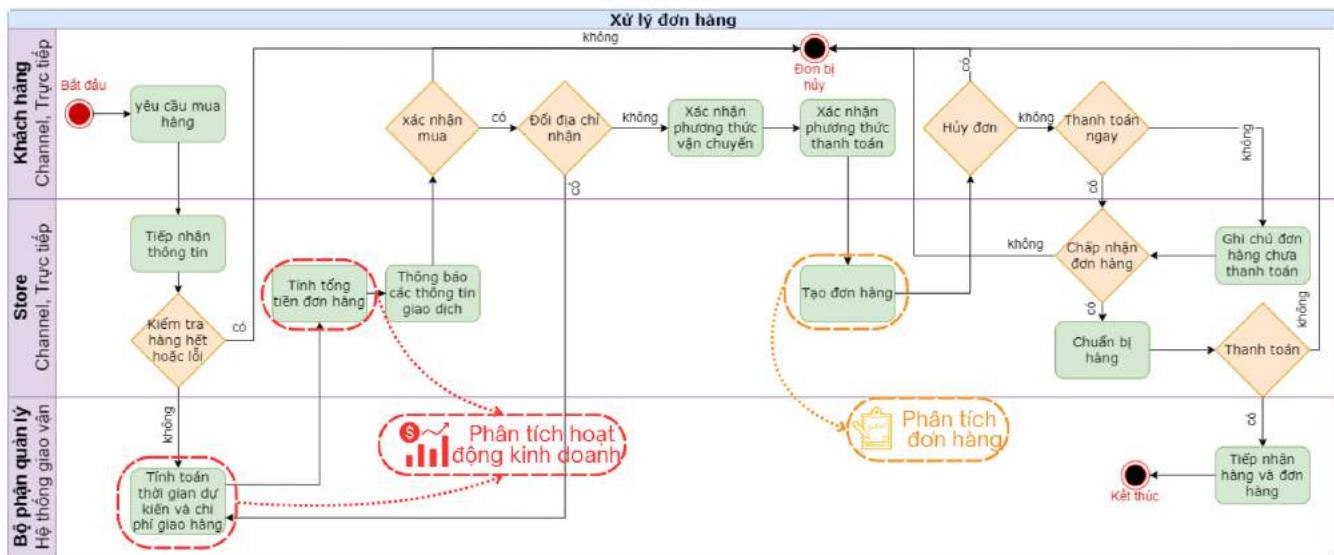
- Key partners (Đối tác chính):** Trung tâm giao vận hợp tác với các tài xế chuyên nghiệp, nhà cung cấp dịch vụ thanh toán uy tín, đối tác công nghệ tiên tiến, đối tác bảo hiểm hàng đầu và nhà cung cấp vật liệu đóng gói chất lượng. Những đối tác này giúp doanh nghiệp duy trì và nâng cao chất lượng dịch vụ, đảm bảo mọi khía cạnh của quá trình giao nhận hàng hóa đều được thực hiện một cách hiệu quả và tin cậy.
- Key Activities (Hoạt động chính):** Trung tâm giao vận chú trọng vào các hoạt động:
 - Quản lý đặt và giao đơn
 - Quản lý lịch trình di chuyển
 - Quản lý kho phân loại và lưu trữ

3. Key Resources (*Tài nguyên chính*): Để duy trì hoạt động hiệu quả, doanh nghiệp đầu tư vào hệ thống máy tính và phần mềm quản lý hiện đại, mạng lưới tài xế chuyên nghiệp và dữ liệu cùng hệ thống quản lý dữ liệu tiên tiến. Những nguồn lực này giúp doanh nghiệp quản lý tốt các đơn hàng, theo dõi và tối ưu hóa quá trình giao nhận hàng hóa.
4. Value Proposition (*Đề xuất giá trị*): Trung tâm giao vận kỳ vọng mang lại những dịch vụ giúp nâng cao trải nghiệm khách hàng, đảm bảo sự hài lòng và tin cậy như:
 - Giao hàng tận tay
 - Vận chuyển nhanh chóng
 - Chất lượng đảm bảo
 - Theo dõi hàng hóa trực tuyến
 - Chính sách hoàn trả linh hoạt
5. Customer Relationships (*Quan hệ khách hàng*): Doanh nghiệp xây dựng mối quan hệ với khách hàng thông qua dịch vụ hỗ trợ trực tuyến, dịch vụ chăm sóc khách hàng và chương trình khách hàng thân thiết. Những dịch vụ này không chỉ giúp giải quyết các vấn đề của khách hàng một cách nhanh chóng mà còn tạo ra sự gắn kết và lòng trung thành từ phía khách hàng.
6. Channels (*Kênh phân phối*): Doanh nghiệp tiếp cận khách hàng thông qua website, ứng dụng di động, dịch vụ trực tiếp, dịch vụ tư vấn trực tuyến và mạng xã hội. Những kênh này giúp doanh nghiệp tiếp cận và phục vụ khách hàng một cách toàn diện và tiện lợi.
7. Customer Segments (*Phân khúc khách hàng*): Trung tâm giao vận phục vụ các khách hàng tại Brazil, cửa hàng bán lẻ, kho phân loại và công ty thương mại điện tử. Doanh nghiệp cũng hiểu rõ nhu cầu của từng phân khúc khách hàng và cung cấp các giải pháp giao vận phù hợp, hiệu quả.
8. Cost Structure (*Cấu trúc chi phí*): Để vận hành doanh nghiệp, trung tâm cần trả những chi phí:
 - Chi phí vận hành và quản lý các kho phân loại và lưu trữ
 - Chi phí thuê và đào tạo nhân viên
 - Chi phí thuê mặt bằng
 - Chi phí quảng cáo và marketing
 - Chi phí phát triển và bảo trì hệ thống
9. Revenue Streams (*Dòng doanh thu*): Bên cạnh đó, doanh nghiệp cũng có những nguồn thu này giúp duy trì và phát triển dịch vụ, đáp ứng nhu cầu ngày càng cao của khách hàng:
 - Phí dịch vụ từ cửa hàng và khách hàng
 - Phí vận đơn
 - Phí từ quảng cáo
 - Doanh thu từ dịch vụ bảo hiểm hàng hóa
 - Phí lưu trữ hàng hóa

Trong bài báo cáo này, nhóm chúng em sẽ dựa trên các dữ liệu về *hoạt động quản lý* của doanh nghiệp để **phân tích hiệu quả giao vận**. Những giá trị như *vận chuyển nhanh chóng và chất lượng đảm bảo* sẽ được thể hiện qua dashboard **phân tích đơn hàng**. Báo cáo cũng sẽ **phân tích các hoạt động kinh doanh** dựa trên các dữ liệu về *chi phí vận hành, chi phí thuê, phí dịch vụ và phí vận đơn*.

b. Luồng nghiệp vụ

Xử lý đơn hàng trước khi giao



Hình 1: Quy trình nghiệp vụ trước khi giao hàng

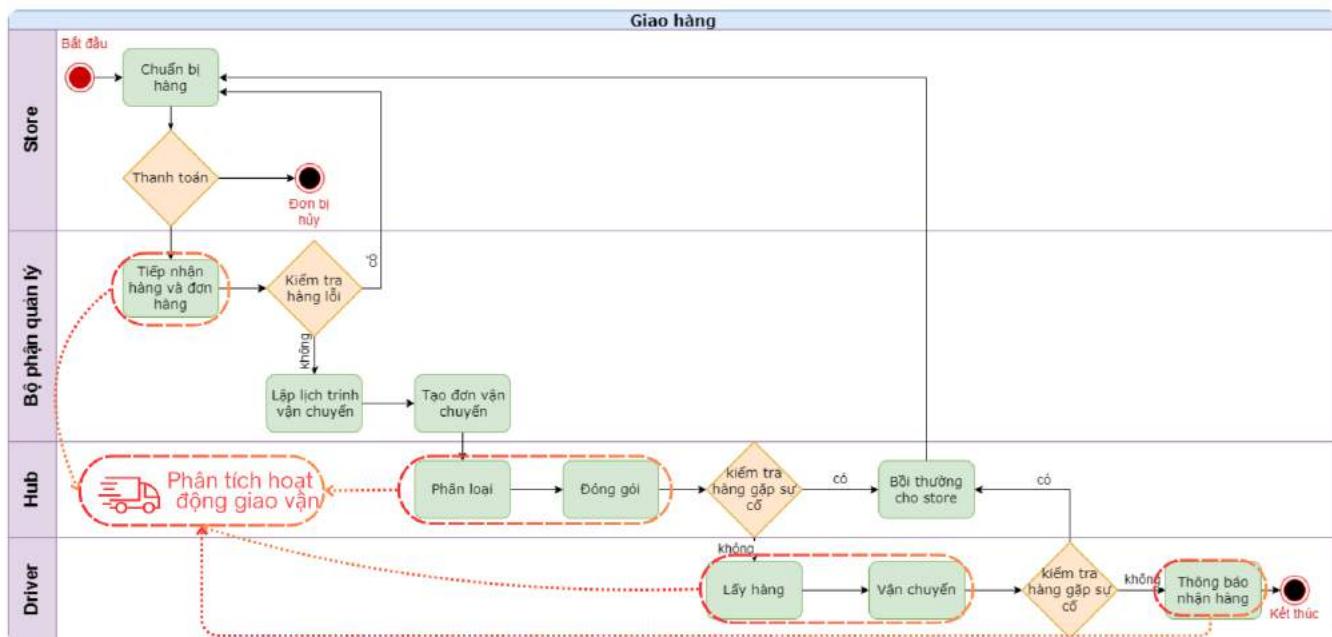
Trên các nền tảng mua sắm, khi khách hàng đặt mua hàng, cửa hàng sẽ nhận thông tin yêu cầu mua hàng và thông tin địa chỉ của khách hàng. Cửa hàng sẽ kiểm tra lại kho hàng của mình xem còn hàng hay có bị vắng đề gì không. Nếu có thì yêu cầu mua hàng sẽ bị hủy. Nếu không thì bộ phận quản lý của trung tâm giao vận sẽ tính toán thời gian dự kiến và chi phí giao hàng.

Sau đó cửa hàng sẽ tính tổng tiền đơn hàng, phí vận chuyển và gửi thông tin giao dịch đến khách hàng. Lúc này, nếu khách hàng không xác nhận mua thì yêu cầu mua hàng sẽ bị hủy. Nếu khách hàng đồng ý mua, họ sẽ xác nhận lại địa chỉ nhận hàng, nếu có thay đổi thì bộ phận quản lý sẽ tính toán lại phí vận chuyển và cửa hàng cập nhật lại thông tin giao dịch cho khách hàng.

Khi khách hàng hoàn tất xác nhận phương thức vận chuyển và phương thức thanh toán, đơn mua hàng sẽ được tạo. Lúc này, khách hàng vẫn còn lựa chọn hủy đơn hay không. Nếu khách hàng không thể thanh toán ngay, cửa hàng sẽ ghi chú lại đơn hàng và nhờ bên vận chuyển thu tiền hộ, sau đó xem xét chấp nhận đơn hàng và bắt đầu chuẩn bị hàng. Khi chuẩn bị hàng xong, cửa hàng sẽ phải thanh toán phí vận chuyển đơn hàng cho bên quản lý trung tâm giao vận. Bộ phận quản lý sẽ tiếp nhận hàng và đơn mua hàng.

Những dữ liệu ở khâu *tính toán thời gian dự kiến và chi phí giao hàng* và *tính tổng tiền đơn hàng* sẽ được phân tích rõ hơn trong **báo cáo về hoạt động kinh doanh**. Những thông tin về *đơn hàng* được tạo sẽ được báo cáo trong dashboard **phân tích đơn hàng**.

Xử lý đơn hàng trong khi giao



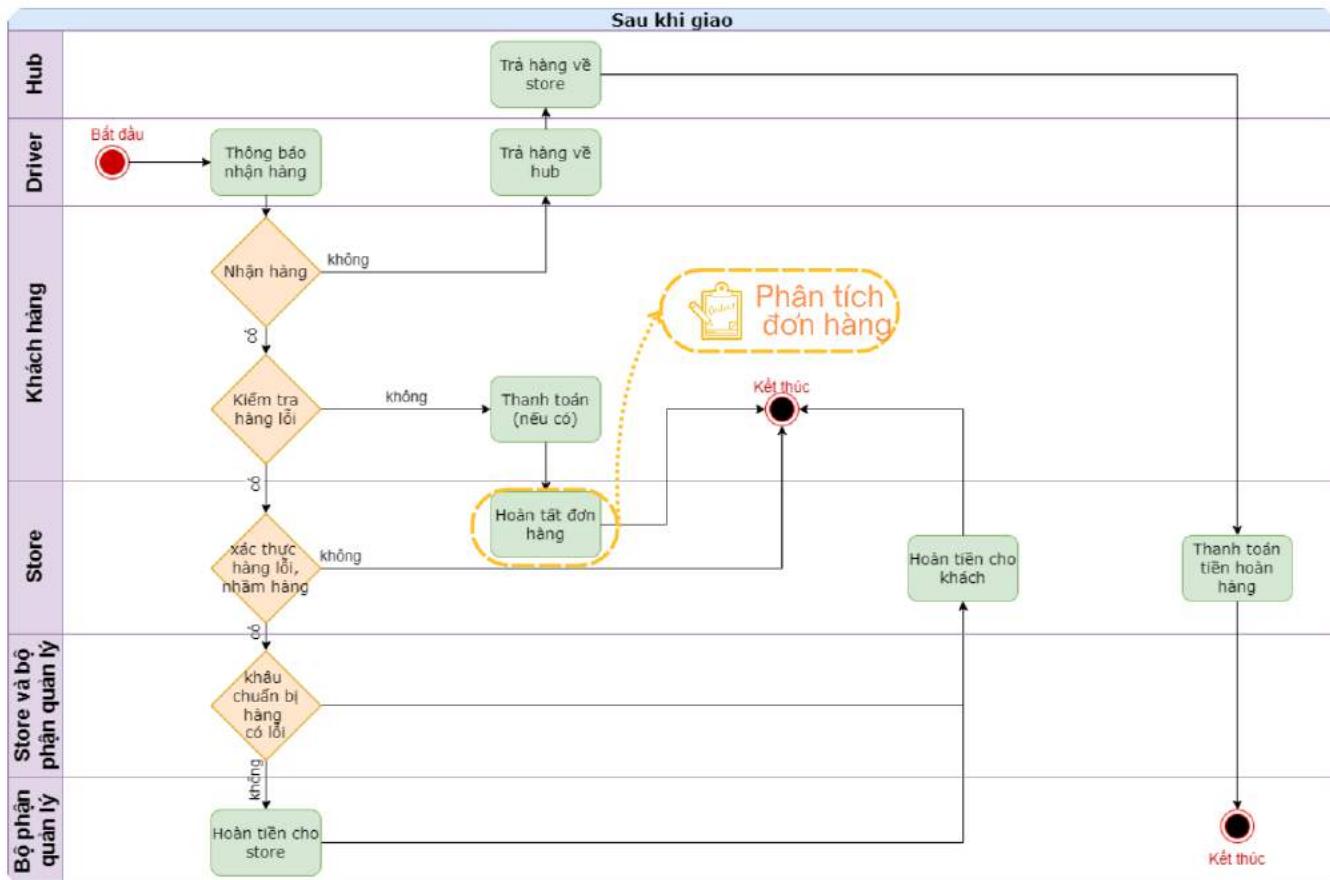
Hình 2: Quy trình nghiệp vụ trong khi giao hàng

Sau khi cửa hàng đã thanh toán phí vận chuyển, bộ phận quản lý sẽ tiếp nhận hàng hóa, đơn mua hàng và kiểm tra lại hàng hóa một lần nữa. Nếu hàng hóa không có lỗi nào, bộ phận quản lý sẽ bắt đầu lập lịch trình vận chuyển và tạo đơn vận chuyển.

Đơn hàng sau đó sẽ được chuyển đến các kho phân loại để thực hiện phân loại và đóng gói. Nếu lúc này hàng hóa gặp sự cố, doanh nghiệp sẽ phải bồi thường lại cho cửa hàng giá trị đơn hàng và phí vận chuyển. Ngược lại, các tài xế sẽ lấy hàng và bắt đầu giao hàng. Nếu hàng hóa bị hỏng hóc hay vỡ, doanh nghiệp phải bồi thường lại cho cửa hàng.

Những thông tin về khoảng cách, thời gian, số lượng trong khâu *tiếp nhận hàng, phân loại, đóng gói, lấy hàng, vận chuyển và thông báo nhận hàng* sẽ được tìm hiểu trong báo cáo **phân tích hoạt động giao vận**.

Xử lý đơn hàng sau khi giao



Hình 3: Quy trình nghiệp vụ sau khi giao hàng

Sau khi tài xế thông báo nhận hàng, khách hàng nếu không nhận thì đơn hàng sẽ hoàn lại về cửa hàng và cửa hàng sẽ phải trả thêm tiền vận chuyển hoàn hàng.

Khi khách nhận hàng mà phát hiện hàng có lỗi sẽ thông báo cho cửa hàng. Lúc này, cửa hàng sẽ phản hồi lại cho khách có đúng là hàng lỗi hay bị nhầm hàng không. Nếu không phải thì khách hàng sẽ cần thanh toán cho tài xế (nếu trước đó cửa hàng nhờ thu hộ).

Nếu thực sự có lỗi về hàng hóa, cửa hàng và bộ phận quản lý phải kiểm tra lại khâu chuẩn bị hàng. Nếu thấy không có lỗi thì hàng hóa bị hỏng trong quá trình vận chuyển và doanh nghiệp sẽ phải hoàn tiền lại cho cửa hàng và cửa hàng hoàn tiền lại cho khách hàng.

Khi đơn hàng được hoàn tất, những thông tin về số lượng, thời gian hoàn thành đơn hàng sẽ được phân tích trong dashboard về đơn hàng.

Sau khi xác định được các nghiệp vụ, ta có được cây nghiệp vụ như sau



Hình 4: Cây nghiệp vụ

3.1.3 Yêu cầu phân tích

a. Các báo cáo cần đưa ra

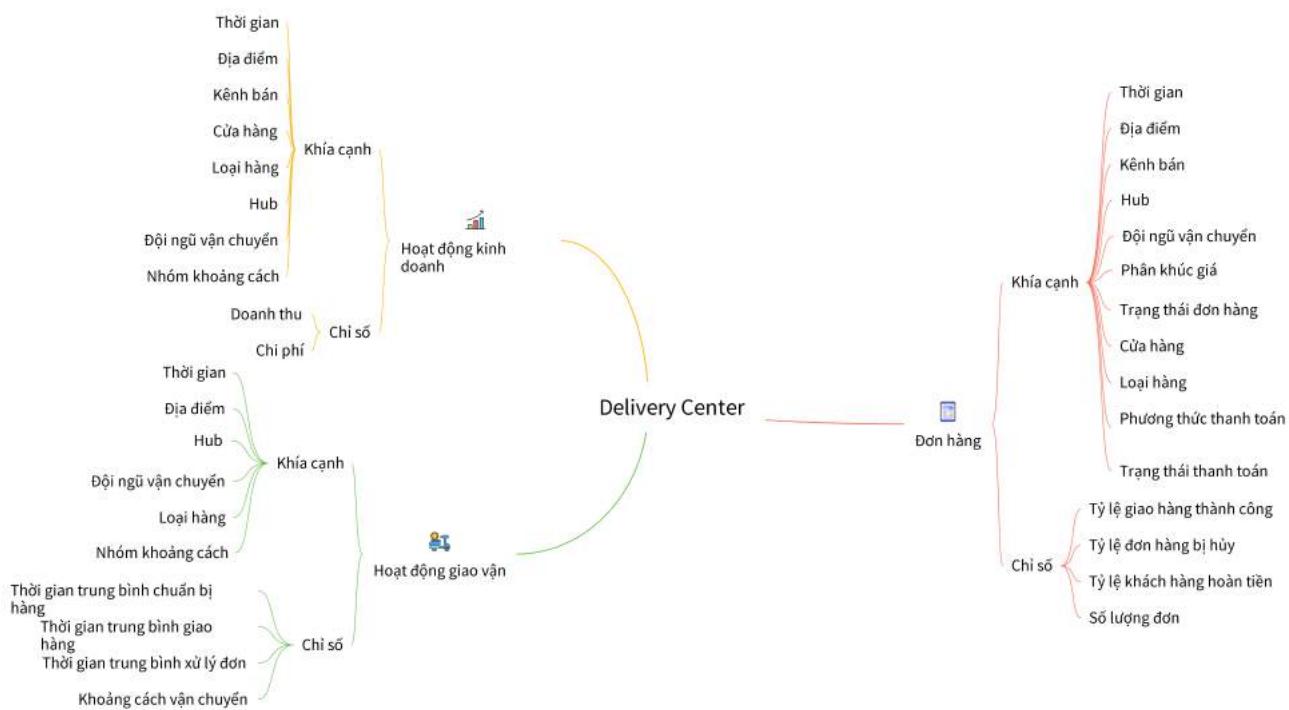
- Hoạt động kinh doanh:** Báo cáo doanh thu và chi phí theo thời gian (giờ, tháng, năm), địa điểm, kênh bán, cửa hàng, loại hàng kho phân loại, đội ngũ vận chuyển và nhóm khoảng cách.
- Hoạt động giao vận:** Báo cáo về khoảng thời gian và khoảng cách vận chuyển các đơn hàng theo giờ, tháng năm, địa điểm, hub, đội ngũ vận chuyển, loại hàng và nhóm khoảng cách.
- Đơn hàng:** Báo cáo về tỷ lệ giao hàng và số lượng đơn giao theo thời gian, địa điểm, kênh bán, hub, đội ngũ vận chuyển, phân khúc giá, trạng thái đơn hàng, cửa hàng, loại hàng và phương thức thanh toán và trạng thái thanh toán.

b. Chủ điểm phân tích

Chủ điểm	Hoạt động kinh doanh	Hoạt động giao vận	Đơn hàng
Chỉ số	<ul style="list-style-type: none"> Doanh thu Chi phí 	<ul style="list-style-type: none"> Thời gian trung bình chuẩn bị hàng Thời gian trung bình giao hàng Thời gian trung bình xử lý đơn Khoảng cách vận chuyển 	<ul style="list-style-type: none"> Tỷ lệ giao hàng thành công Tỷ lệ đơn hàng bị hủy Tỷ lệ khách hàng hoàn tiền Số lượng đơn
Khía cạnh	<ul style="list-style-type: none"> Thời gian Địa điểm Kênh bán Cửa hàng Loại hàng Hub Đội ngũ vận chuyển Nhóm khoảng cách 	<ul style="list-style-type: none"> Thời gian Địa điểm Hub Đội ngũ vận chuyển Loại hàng Nhóm khoảng cách 	<ul style="list-style-type: none"> Thời gian Địa điểm Kênh bán Hub Đội ngũ vận chuyển Phân khúc giá Trạng thái đơn hàng Cửa hàng Loại hàng Phương thức thanh toán Trạng thái thanh toán

Hình 5: Chủ điểm phân tích

c. Cây phân tích chủ điểm



Hình 6: Cây phân tích chủ điểm

d. Cây phân tích Dashboard

Ta phân tích sâu hơn vào từng chủ điểm theo các yêu cầu phân tích như sau

d.1. Chủ điểm Hoạt động kinh doanh



- Tổng doanh thu của đơn vị vận chuyển là bao nhiêu trong một khoảng thời gian cụ thể?
- Tổng chi phí vận chuyển trong các tháng vừa qua là bao nhiêu?

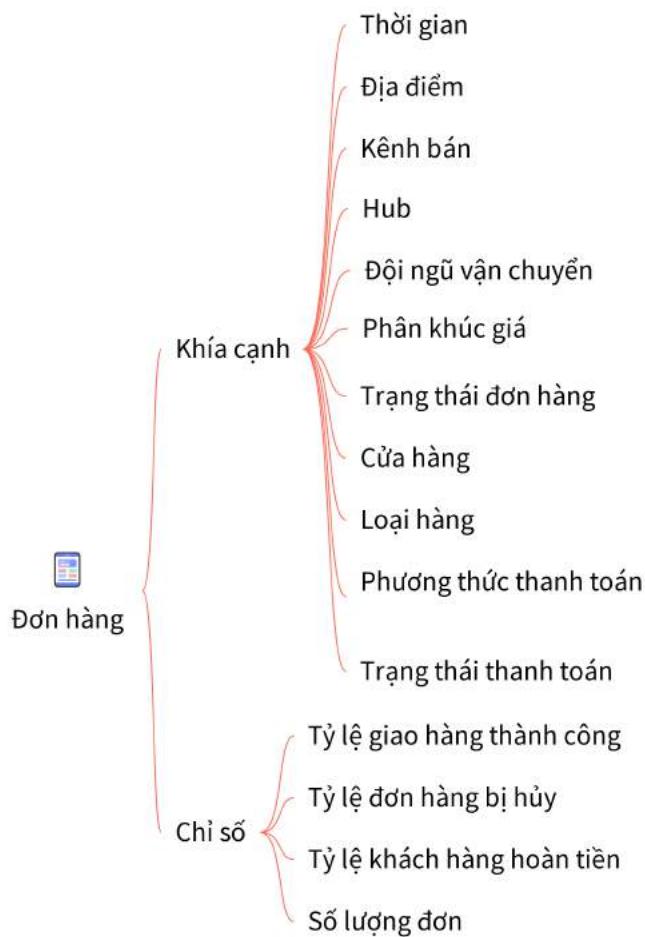
- Doanh thu theo tháng/quý/năm có xu hướng tăng hay giảm như thế nào?
- Thời điểm nào trong tuần là thời gian cao điểm về doanh số?
- Các cửa hàng ở khu vực nào có doanh thu cao nhất?
- Các địa điểm nào phát sinh chi phí vận chuyển lớn nhất?
- Kênh bán hàng nào mang lại doanh thu cao nhất?
- Loại hàng nào bán chạy nhất trong hệ thống?
- Hub nào có hiệu suất vận hành cao nhất (số lượng hàng xử lý, thời gian giao hàng, vv.)?

d.2. Chủ điểm Hoạt động giao vận



- Thời gian trung bình để hoàn tất một đơn hàng từ lúc đặt đến khi giao thành công là bao nhiêu?
- Thời gian nào trong ngày/tuần/tháng có số lượng giao vận cao nhất?
- Có sự khác biệt đáng kể nào về thời gian giao hàng giữa các loại sản phẩm khác nhau không?
- Những khu vực nào có thời gian giao hàng trung bình nhanh nhất?
- Khu vực nào thường gặp phải sự chậm trễ trong giao hàng?
- Thời gian xử lý đơn hàng tại mỗi hub là bao lâu?
- Đội ngũ vận chuyển nào có thời gian hoàn tất giao hàng nhanh nhất?
- Thời gian giao hàng thay đổi như thế nào với các khoảng cách khác nhau?

d.3. Chủ điểm Đơn hàng

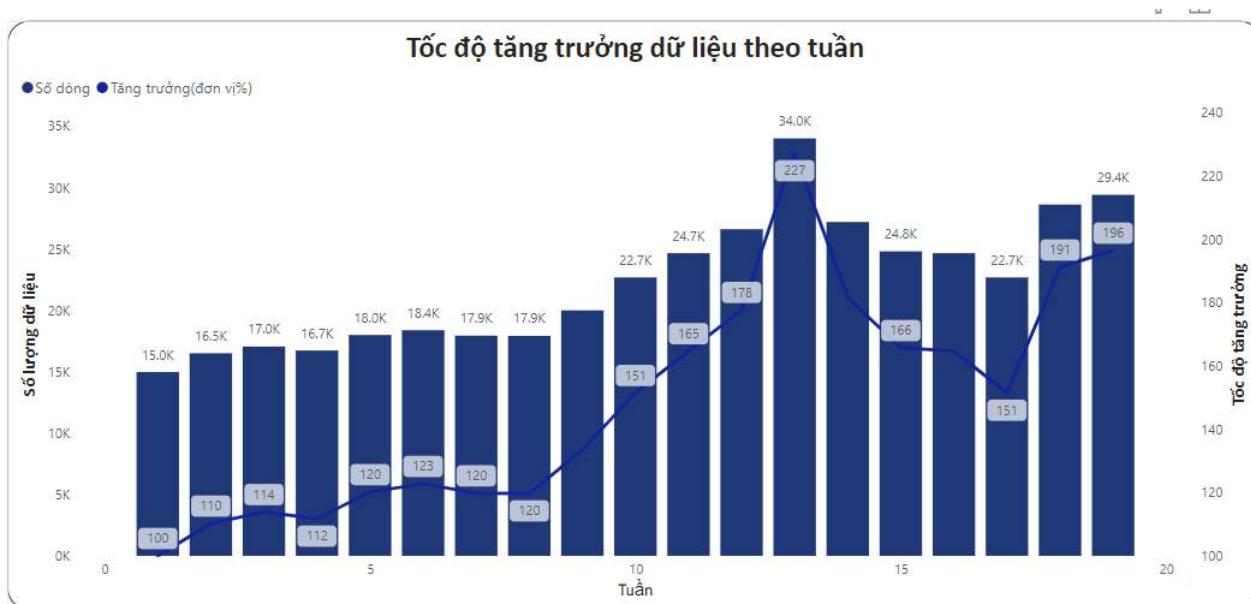


- Số lượng đơn đặt, giao hàng thay đổi như thế nào theo tuần, tháng?
- Khu vực địa lý nào có số lượng đơn đặt hàng cao nhất?
- Địa điểm nào thường gặp phải sự hủy bỏ đơn hàng nhiều nhất?
- Khu vực nào có số lượng đơn hàng giao thành công cao nhất?
- Số lượng đơn đặt hàng thay đổi như thế nào theo các phân khúc giá khác nhau (giá thấp, trung bình, cao)?
- Phần trăm đơn đặt hàng thành công so với tổng số đơn đặt hàng là bao nhiêu?
- Thời gian trung bình để một đơn hàng chuyển từ trạng thái đặt hàng đến trạng thái giao hàng là bao nhiêu?
- Thời gian giao hàng trung bình từ lúc đặt đến lúc giao thành công là bao nhiêu?
- Cửa hàng nào có số lượng đơn đặt hàng cao nhất?
- Phương thức thanh toán nào được sử dụng nhiều nhất trong các đơn đặt hàng?
- Phần trăm các đơn đặt hàng đã hoàn tất thanh toán là bao nhiêu?

3.1.4 Quy mô dữ liệu

- Bộ dữ liệu mà chúng em sử dụng trong báo cáo này có tên :" *Delivery Center: Food & Goods orders in Brazil*" có nguồn từ : [kaggle.com](https://www.kaggle.com)
- Bộ dữ liệu dữ này có nội dung dữ liệu về đặt hàng và giao hàng đã được Trung tâm giao hàng (Delivery Center) xử lý trong khoảng thời gian từ tháng 1 đến tháng 4 năm 2021.
- Quy mô bộ dữ liệu như sau :
 - Kích thước bộ dữ liệu : 129 MB.
 - Dịnh dạng file : csv.
 - Dòng thời gian : từ 1/1/2021 đến 30/4/2021.
 - Tần suất cập nhật dữ liệu : Phụ thuộc vào dữ liệu cần cập nhật.Ví dụ như thông tin thanh toán theo thời gian thực, thông tin giao hàng theo thời gian thực.....
 - Số bảng dữ liệu : 7
 - Số trường dữ liệu : 60

a. Tốc độ tăng trưởng dữ liệu



Ta thấy dữ liệu tăng trưởng mạnh vào tuần thứ 13 và giảm vào 3 tuần tiếp những tuần cuối có xu hướng hồi phục. Hơn nữa tốc độ tăng trưởng dữ liệu là xấp xỉ 40%.

Sự gia tăng dữ liệu đáng kể đặt ra nhiều thách thức và cơ hội cho các doanh nghiệp. Dưới đây là một số vấn đề chính mà các doanh nghiệp có thể gặp phải do sự tăng trưởng dữ liệu nhanh chóng:

- Quản lý và Lưu trữ Dữ liệu.
- Bảo mật và Quyền riêng tư.
- Hiệu suất Hệ thống.
- Khả năng mở rộng.

b. Tổng quan dữ liệu

Tên bảng	Số cột	Số hàng	Số bản ghi (Null)
channels	3	40	0
deliveries	5	382,384	850
drivers	3	4,825	0
hubs	6	32	0
orders	29	369,009	105,512
payments	6	412,420	0
stores	7	951	147

1. Bảng channels

Gồm: Thông tin về các kênh bán hàng (kênh tiếp thị) nơi bán hàng hóa và thực phẩm của nhà bán lẻ.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40 entries, 0 to 39
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   channel_id   40 non-null    int64  
 1   channel_name 40 non-null    object  
 2   channel_type  40 non-null    object  
dtypes: int64(1), object(2)
memory usage: 1.1+ KB
```

- channel_id: Mỗi kênh bán hàng có một channel_id duy nhất.
- channel_name: Tên của kênh bán hàng.
- channel_type: Kiểu của kênh bán hàng.

2. Bảng deliveries

Gồm: Thông tin về việc giao hàng được thực hiện bởi người chuyển phát.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   hub_id      32 non-null    int64  
 1   hub_name    32 non-null    object  
 2   hub_city    32 non-null    object  
 3   hub_state   32 non-null    object  
 4   hub_latitude 32 non-null    float64 
 5   hub_longitude 32 non-null    float64 
dtypes: float64(2), int64(1), object(3)
memory usage: 1.6+ KB
```

- delivery_id: Mỗi đơn giao hàng có một mã định danh duy nhất.
- delivery_order_id: Mã định danh giao hàng liên quan đến đơn đặt hàng.

- driver_id: Mã định danh người giao hàng phụ trách đơn vận chuyển đó.
- delivery_distance: Khoảng cách giao hàng.
- delivery_status: Trạng thái giao hàng.

3. Bảng drivers

Gồm: Thông tin về người giao hàng.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4825 entries, 0 to 4824
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype  
---  --          -----          ---  
 0   driver_id    4825 non-null   int64  
 1   driver_modal 4825 non-null   object  
 2   driver_type   4825 non-null   object  
dtypes: int64(1), object(2)
memory usage: 113.2+ KB
```

- driver_id: Mỗi người giao hàng có một mã định danh duy nhất.
- driver_modal: Phương thức giao hàng của tài xế.
- driver_type: Kiểu người giao hàng.

4. Bảng hubs

Gồm: Thông tin về các hub(trung tâm giao hàng) của Delivery Center.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4825 entries, 0 to 4824
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype  
---  --          -----          ---  
 0   driver_id    4825 non-null   int64  
 1   driver_modal 4825 non-null   object  
 2   driver_type   4825 non-null   object  
dtypes: int64(1), object(2)
memory usage: 113.2+ KB
```

- hub_id: Mỗi trung tâm có một mã định danh duy nhất.
- hub_name: Tên của trung tâm phân phối.
- hub_city: Tên thành phố.
- hub_state: Tên bang.
- hub_latitude: Vĩ độ.
- hub_longitude: Kinh độ.

5. Bảng orders

Gồm: Thông tin về doanh số bán hàng được xử lý thông qua nền tảng Delivery Center.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 369009 entries, 0 to 369008
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   order_id         369009 non-null   int64  
 1   store_id         369009 non-null   int64  
 2   channel_id       369009 non-null   int64  
 3   payment_order_id 369009 non-null   int64  
 4   delivery_order_id 369009 non-null   int64  
 5   order_status     369009 non-null   object  
 6   order_amount     369009 non-null   float64 
 7   order_delivery_fee 369009 non-null   float64 
 8   order_delivery_cost 369009 non-null   float64 
 9   order_created_hour 369009 non-null   int64  
 10  order_created_minute 369009 non-null   int64  
 11  order_created_day 369009 non-null   int64  
 12  order_created_month 369009 non-null   int64  
 13  order_created_year 369009 non-null   int64  
 14  order_moment_created 369009 non-null   object  
 15  order_moment_accepted 360935 non-null   object  
 16  order_moment_ready 360935 non-null   object  
 17  order_moment_collected 360935 non-null   object  
 18  order_moment_in_expedition 360935 non-null   object  
 19  order_moment_delivering 360935 non-null   object  
 20  order_moment_delivered 352311 non-null   object  
 21  order_moment_finished 369009 non-null   object  
 22  order_metric_collected_time 360935 non-null   float64 
 23  order_metric_paused_time 360935 non-null   float64 
 24  order_metric_production_time 360935 non-null   float64 
 25  order_metric_walking_time 360935 non-null   float64 
 26  order_metric_expedition_speed_time 360935 non-null   float64 
 27  order_metric_transit_time 360935 non-null   float64 
 28  order_metric_cycle_time 369009 non-null   int64  
dtypes: float64(9), int64(11), object(9)
memory usage: 81.6+ MB
```

Một số thuộc tính quan trọng :

- order_id: Mỗi đơn hàng có một mã định danh duy nhất.
- order_amount: Giá trị đơn hàng.
- order_delivery_fee: Phí vận chuyển đơn hàng.
- order_delivery_cost: Chi phí vận chuyển đơn hàng.
- order_status: Trạng thái đặt hàng.
- order_created_day: Ngày tạo đơn hàng.
- order_created_month: Tháng tạo đơn hàng.
- order_created_year: Năm tạo đơn hàng.
- order_moment_created: Thời điểm tạo đơn hàng.
- order_moment_collected: Thời điểm đơn hàng được tiếp nhận để vận chuyển.
- order_moment_finished: Thời điểm hoàn thành việc giao đơn hàng.

6. Bảng payments

Gồm: Thông tin về các khoản thanh toán được thực hiện cho Delivery Center.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 412420 entries, 0 to 412419
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   payment_id       412420 non-null   int64  
 1   payment_order_id 412420 non-null   int64  
 2   payment_amount    412420 non-null   float64 
 3   payment_fee       412420 non-null   float64 
 4   payment_method    412420 non-null   object  
 5   payment_status    412420 non-null   object  
dtypes: float64(2), int64(2), object(2)
memory usage: 18.9+ MB
```

- payment_id: Mã định danh duy nhất cho mỗi thanh toán.
- payment_order_id: Mã định danh đơn hàng liên quan đến thanh toán.
- payment_amount: Giá trị thanh toán.
- payment_fee: Phí thanh toán.
- payment_method: Phương thức thanh toán.
- payment_status: Trạng thái thanh toán.

7. Bảng stores

Gồm: Thông tin về các nhà bán lẻ sử dụng Nền tảng Delivery Center để bán các mặt hàng của họ (hàng hóa và/hoặc thực phẩm) trên các chợ.

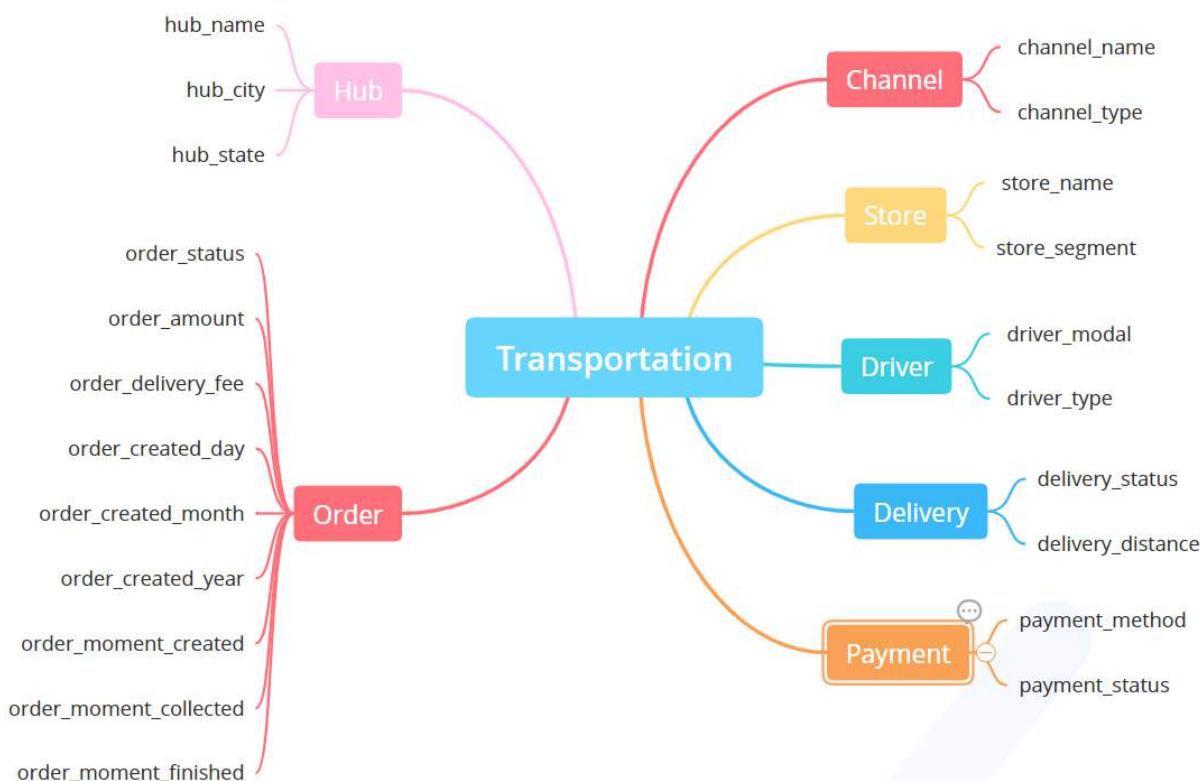
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 951 entries, 0 to 950
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   store_id         951 non-null   int64  
 1   hub_id           951 non-null   int64  
 2   store_name       951 non-null   object  
 3   store_segment    951 non-null   object  
 4   store_plan_price 836 non-null   float64 
 5   store_latitude   935 non-null   float64 
 6   store_longitude  935 non-null   float64 
dtypes: float64(3), int64(2), object(2)
memory usage: 52.1+ KB
```

- hub_id: Mã định danh trung tâm liên quan đến nhà bán lẻ.

- store_name: Tên cửa hàng bán lẻ.
- store_segment: Phân khúc bán lẻ.
- store_plan_price: Giá trị kế hoạch của nhà bán lẻ.
- store_latitude: Vĩ độ của store.
- store_longitude: Kinh độ của store.

c. Data taxonomy

Data taxonomy là quá trình tổ chức dữ liệu thành cấu trúc phân cấp, nhóm dữ liệu tương tự lại với nhau và cung cấp mô tả và hiểu mối quan hệ giữa các thành phần dữ liệu với nhau.



Transportation (chủ điểm trung tâm)

- **Hub** (Trung tâm) : Danh mục chính trong đó danh mục phụ gồm :

- hub_name
- hub_city
- hub_state

- **Order** (Đơn hàng) : Danh mục chính trong đó danh mục phụ gồm :

- order_status
- order_amount
- order_delivery_fee,....

- **Payment** (Thanh toán) :Danh mục chính trong đó danh mục phụ gồm :

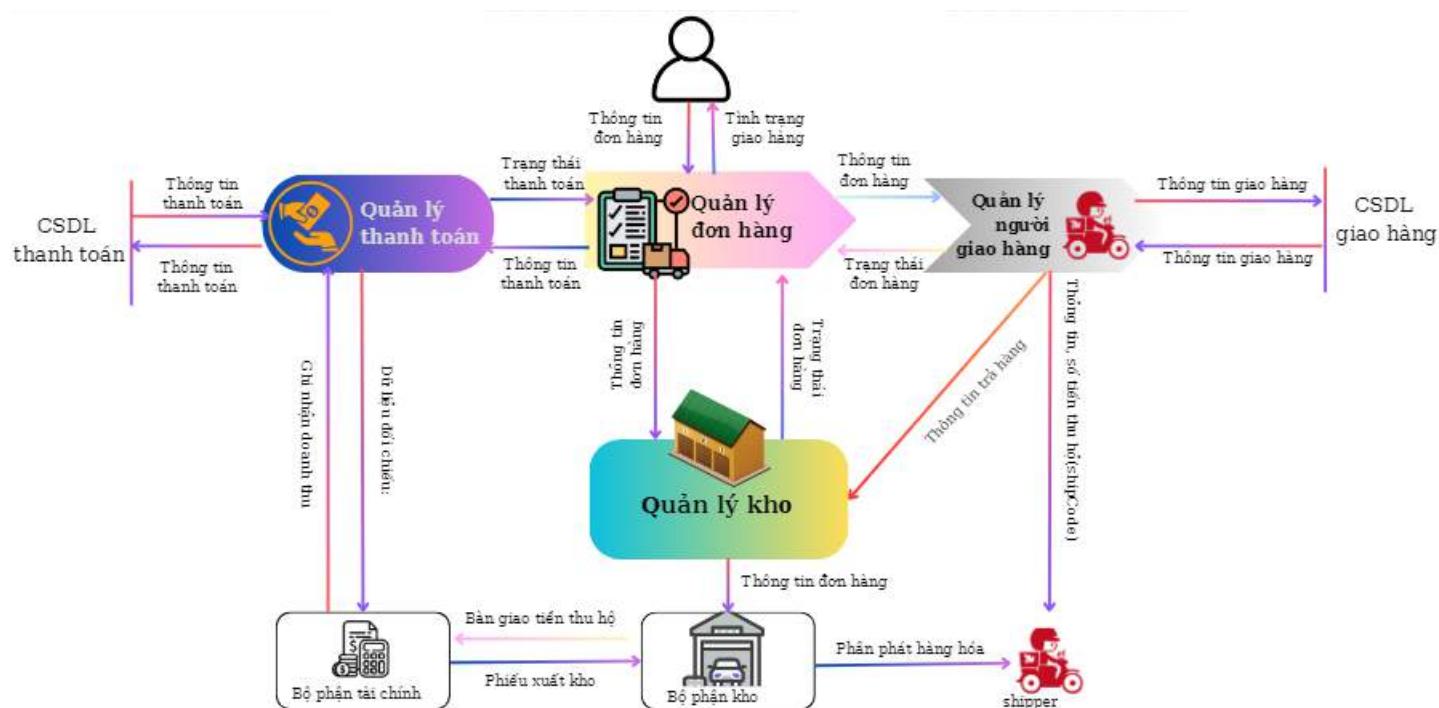
- payment_method

- payment_status
- **Channel** (Kênh) : Danh mục chính trong đó danh mục phụ gồm :
 - channel_name
 - channel_type
- **Store** (Cửa hàng) : Danh mục chính trong đó danh mục phụ gồm :
 - store_name
 - store_segment
- **Driver** (Tài xế) : Danh mục chính trong đó danh mục phụ gồm :
 - driver_modal
 - driver_type
- **Delivery** (Giao hàng) : Danh mục chính trong đó danh mục phụ gồm :
 - delivery_status
 - delivery_distance

d. Data flow

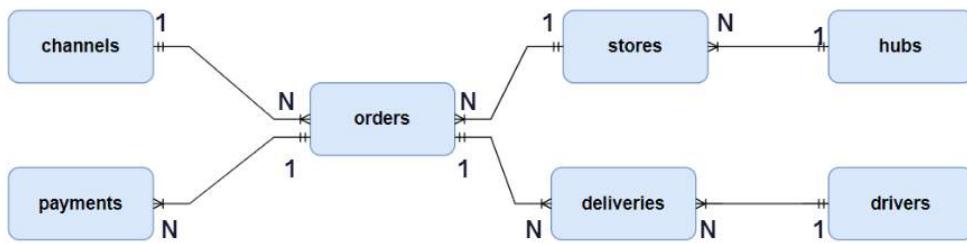
Hệ thống *giao vận* được thành 4 nhóm hệ thống chính: Hệ thống quản lý thanh toán, Hệ thống quản lý người giao hàng, Hệ thống quản lý kho, Hệ thống quản lý đơn hàng.

Dựa trên tương tác của khách hàng với hệ thống, sơ đồ luồng dữ liệu được mô tả như hình dưới đây:



3.1.5 ERD hệ thống OLTP

a. ER



Mối quan hệ giữa các thực thể :

- Quan hệ giữa thực thể *channels* và *orders*.

Một kênh tiếp thị sẽ có nhiều đơn hàng=> quan hệ *channels* và *orders* là 1-N.

- Quan hệ giữa thực thể *payments* và *orders*.

Một đơn hàng sẽ có nhiều thanh toán => quan hệ giữa bảng *orders* và *payments* là 1-N(do người dùng có thể thanh toán 1 phần).

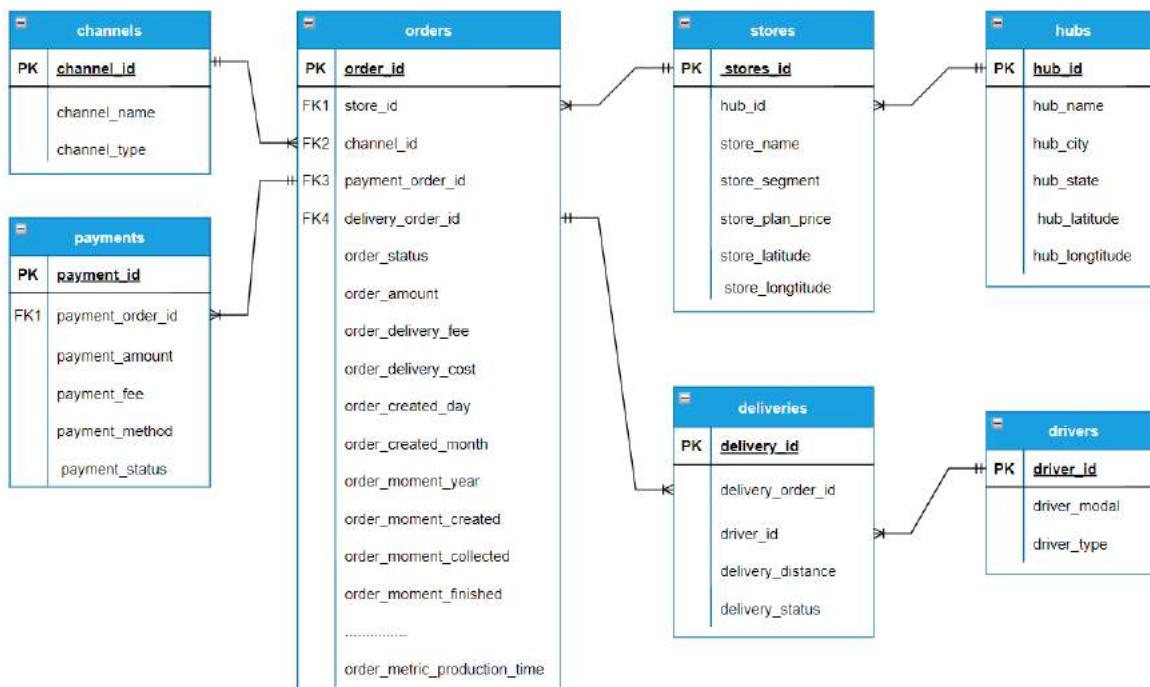
- Quan hệ giữa thực thể *hubs* với *stores* và *stores* với *orders*.

Một trung tâm giao hàng có nhiều stores gửi hàng đến, một cửa hàng thì có nhiều đơn hàng để bán=> quan hệ giữa thực thể *hubs* và *stores*, *stores* và *orders* là 1-N.

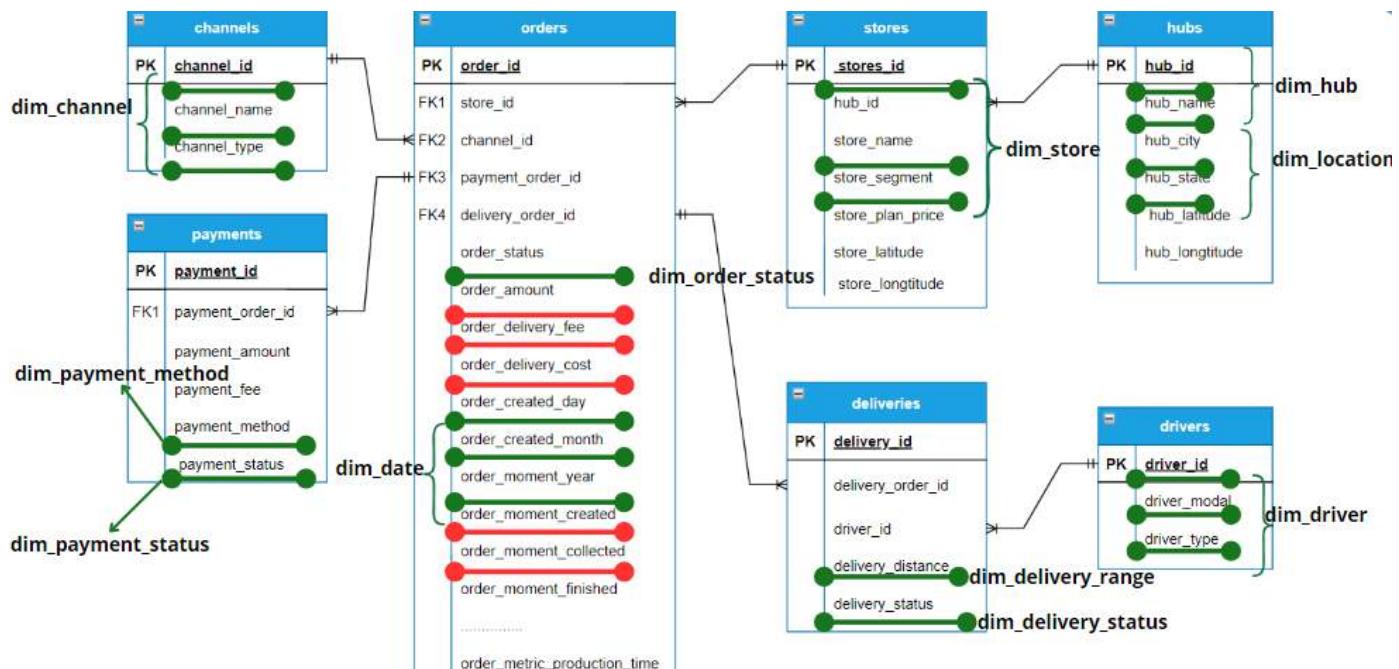
- Quan hệ giữa thực thể *deliveries* với *drivers* và *deliveries* với *orders*.

Một driver có thể có nhiều deliveries, một đơn hàng có nhiều deliveries(do không giao được phải giao cho ca sau người giao hàng giao lại)=> quan hệ giữa thực thể *drivers* và *deliveries* là 1-N, giữa *orders* và *deliveries* là 1-N.

b. Entity Relationship Diagram (ERD). OLTP



c. Mapping tương ứng khi chuyển sang OLAP

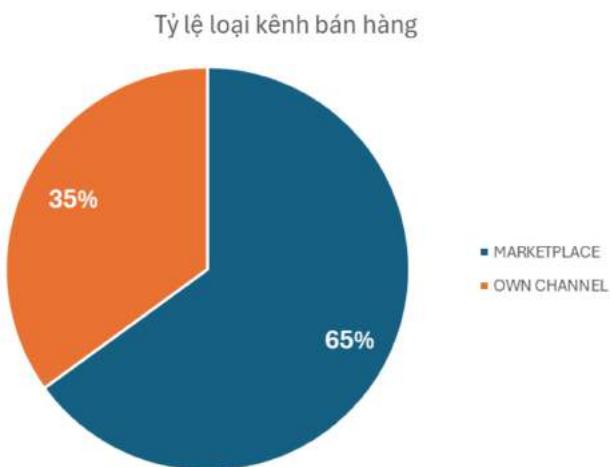


Các thuộc tính ở diagram được gạch màu xanh sẽ trở thành dimension, còn thuộc tính được gạch chân màu đỏ sẽ trở thành fact.

3.2 Phân tích và thiết kế

3.2.1 Khám phá dữ liệu

a. Tỷ lệ loại kênh bán hàng

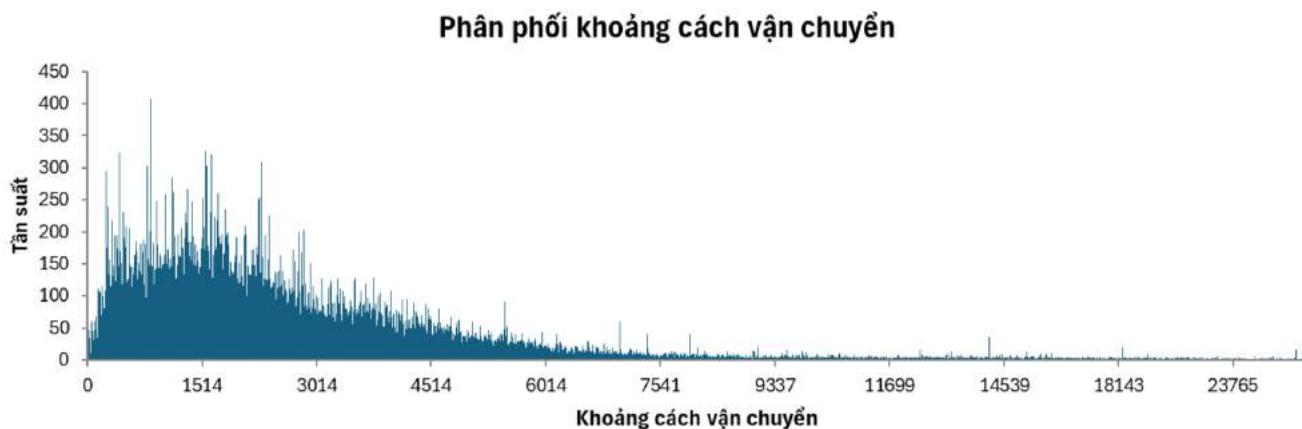


Tổng cộng 40 kênh bán hàng:

- MARKETPLACE chiếm 65% (26/40)
- OWN CHANNEL chiếm 35% (14/40)

=> Doanh nghiệp phụ thuộc nhiều hơn vào các nền tảng MARKETPLACE để bán hàng.

b. Phân phối khoảng cách vận chuyển



- Phần lớn các khoảng cách vận chuyển tập trung ở phía bên trái của biểu đồ, cho thấy phần lớn các chuyến giao hàng có khoảng cách ngắn.
- Tần suất giảm dần khi khoảng cách vận chuyển tăng lên=>có ít chuyến vận chuyển hơn với khoảng cách lớn hơn.
- Sau đỉnh cao nhất, tần suất giảm dần khi khoảng cách vận chuyển tăng lên. Điều này cho thấy rằng các chuyến vận chuyển với khoảng cách lớn hơn ít phổ biến hơn.

c. Lượng đơn theo tình trạng giao hàng

Lượng đơn theo tình trạng giao hàng

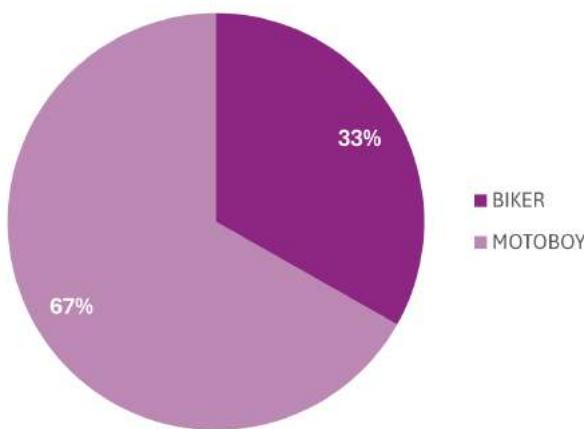


- Số lượng đơn hàng đã giao là 372,960. Đây là con số rất lớn, chiếm phần lớn trong tổng số đơn hàng. Số lượng đơn hàng đã hủy là 9,424.
- Tình trạng giao hàng DELIVERED có lượng đơn gấp 39,6 lần CANCELLED.
- Hiệu suất giao hàng tốt với tỷ lệ đơn hàng thành công (đã giao) rất cao, trong khi tỷ lệ đơn hàng thất bại (đã hủy) là rất thấp.

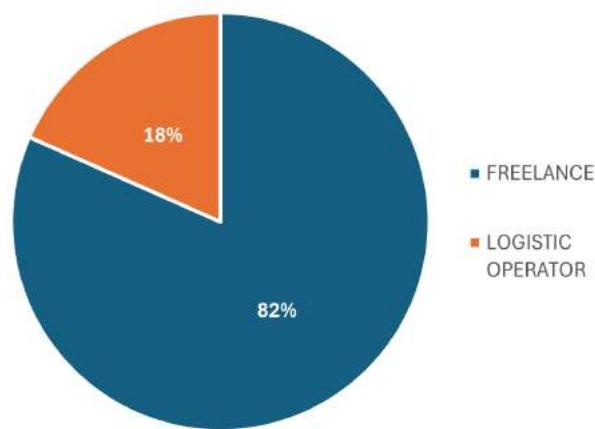
=> Doanh nghiệp hoạt động rất hiệu quả trong việc giao hàng, với tỷ lệ đơn hàng thành công cao và tỷ lệ hủy đơn thấp.

d. Tỷ lệ về lượng tài xế

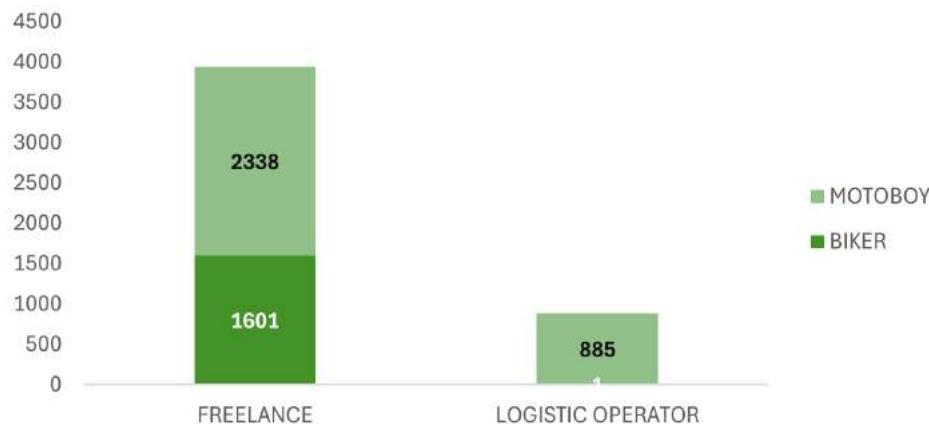
Tỷ lệ tài xế theo loại hình xe



Tỷ lệ tài xế theo đơn vị vận chuyển



Lượng tài xế theo đơn vị và hình thức vận chuyển



- Tỷ lệ giao hàng là FREELANCE gấp 4,5 lần LOGISTIC OPERATOR. Trong khi đó, lượng đơn được vận chuyển bởi FREELANCE lại chỉ gấp 3 lần LOGISTIC OPERATOR. \Rightarrow Phần lớn tài xế làm việc tự do, cho thấy số lượng tài xế tự do nhiều hơn đáng kể so với tài xế làm việc cho các đơn vị vận chuyển logistic.
- Lượng đơn giao bằng hình thức MOTOBOY nhiều gấp 2 lần BIKER. Số lượng MOTOBOY làm việc tự do là lớn nhất, tiếp theo là BIKER làm việc tự do. Logistic Operator có số lượng MOTOBOY thấp hơn nhiều so với freelancer.

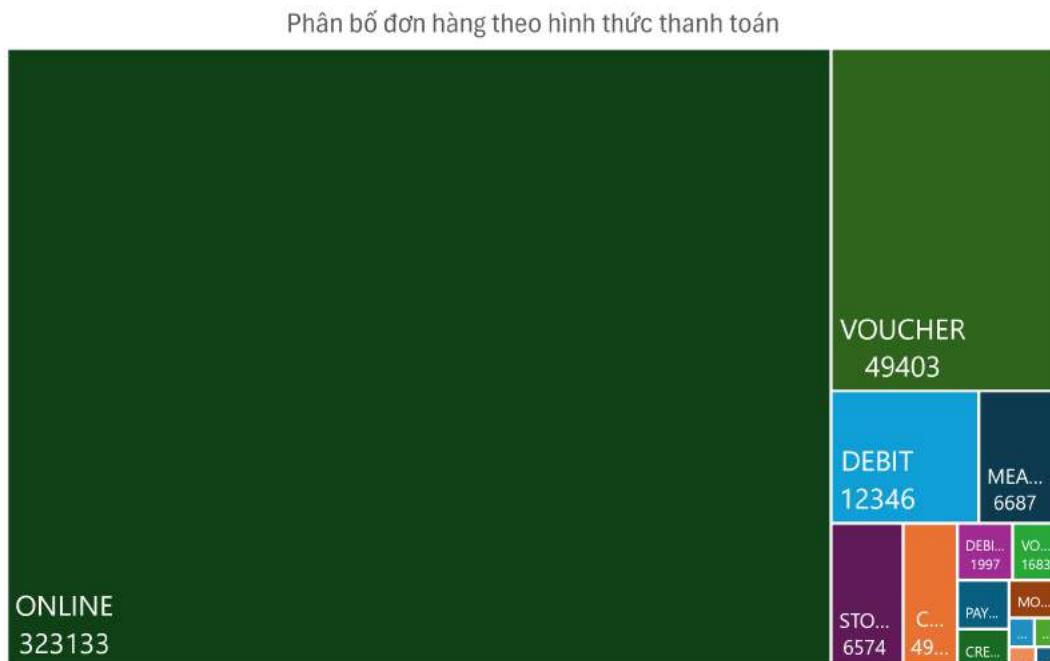
\Rightarrow Các đơn vị LOGISTIC và hình thức vận chuyển MOTOBOY giao hàng hiệu quả hơn. Nhìn chung, thị trường tài xế tự do rất phát triển, đặc biệt là nhóm MOTOBOY.

e. Chi phí và doanh thu



- Sự phân bố giá trị đơn hàng của Order amount không đồng đều.
- Phí giao hàng(Order delivery fee) chủ yếu nằm trong khoảng từ 0 đến 200, nhưng có một số ít đơn hàng có phí giao hàng rất cao, lên đến khoảng 1000.
- Chi phí giao hàng(Order delivery cost) phần lớn từ 0-20=> chi phí vận hàng tương đối thấp

f. Phân bố đơn hàng theo hình thức thanh toán

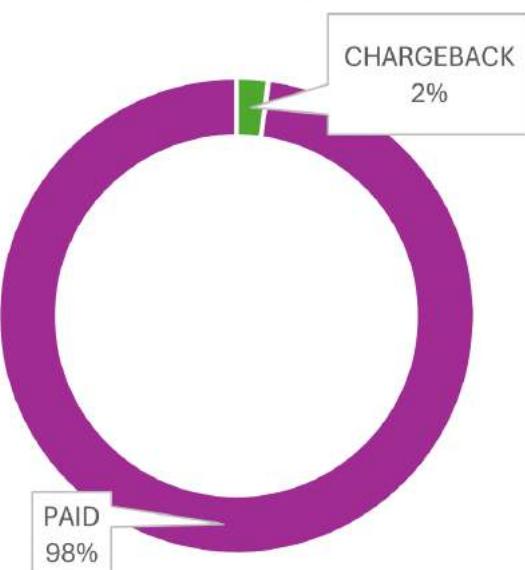


- ONLINE chiếm phần lớn nhất với 323,133 đơn hàng.
- VOUCHER đứng thứ hai với 49,403 đơn hàng.
- Hình thức thanh toán ONLINE chiếm phần lớn tổng số đơn hàng, gấp 6,5 lần hình thức thanh toán VOUCHER(với số lượng đơn hàng nhiều gấp 2).

=> Hình thức thanh toán ONLINE là xu hướng của người tiêu dùng. Bên cạnh đó VOUCHER đứng thứ 2 có thể do chương trình khuyến mãi

g. Phân bố tình trạng đơn hàng

Phân bố tình trạng đơn hàng



- Đơn hàng đã thanh toán chiếm 98% tổng số đơn hàng. Đây là một tỷ lệ rất cao, cho thấy phần lớn các giao dịch được hoàn tất và thanh toán thành công. Điều này phản ánh khả năng thu hồi tiền và mức độ tin cậy trong quá trình thanh toán của doanh nghiệp.
- Đơn hàng bị hoàn trả chiếm 2% tổng số đơn hàng. Mặc dù tỷ lệ này là nhỏ, nhưng nó vẫn có thể gây ra một số vấn đề cần giải quyết, chẳng hạn như lý do hoàn trả, quản lý rủi ro gian lận, hoặc cải thiện trải nghiệm khách hàng để giảm thiểu số lượng đơn hàng bị hoàn trả.

=> Doanh nghiệp đang hoạt động khá hiệu quả trong việc xử lý thanh toán với tỷ lệ đơn hàng thanh toán thành công rất cao.

h. Lượng giao dịch theo hình thức và tình trạng thanh toán

Lượng giao dịch theo hình thức và tình trạng thanh toán

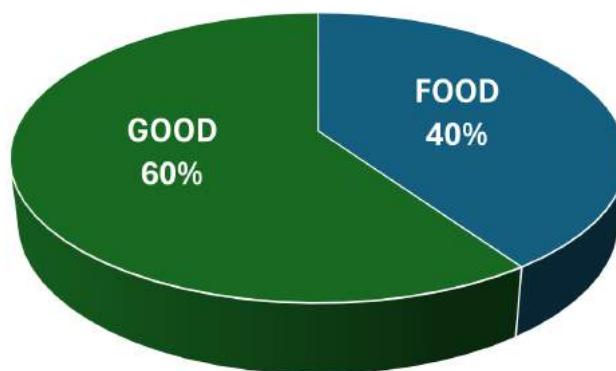


- ONLINE là hình thức thanh toán có lượng giao dịch PAID lớn nhất gấp 125 lần hình thức Voucher.
- Hầu hết các hình thức thanh toán khác đều không có giao dịch CHARGEBACK.
- Một số hình thức thanh toán có lượng PAID lớn như VOUCHER, STORE_DIRECT_PAYMENT, và DEBIT, nhưng không có giao dịch CHARGEBACK.

=> PAID khá cao nhưng không xuất hiện vấn đề về CHARGEBACK, điều này có thể cho thấy tính ổn định và ít rủi ro của các hình thức này so với thanh toán ONLINE.

i. Phân bố cửa hàng theo loại sản phẩm

Phân phối cửa hàng theo loại sản phẩm

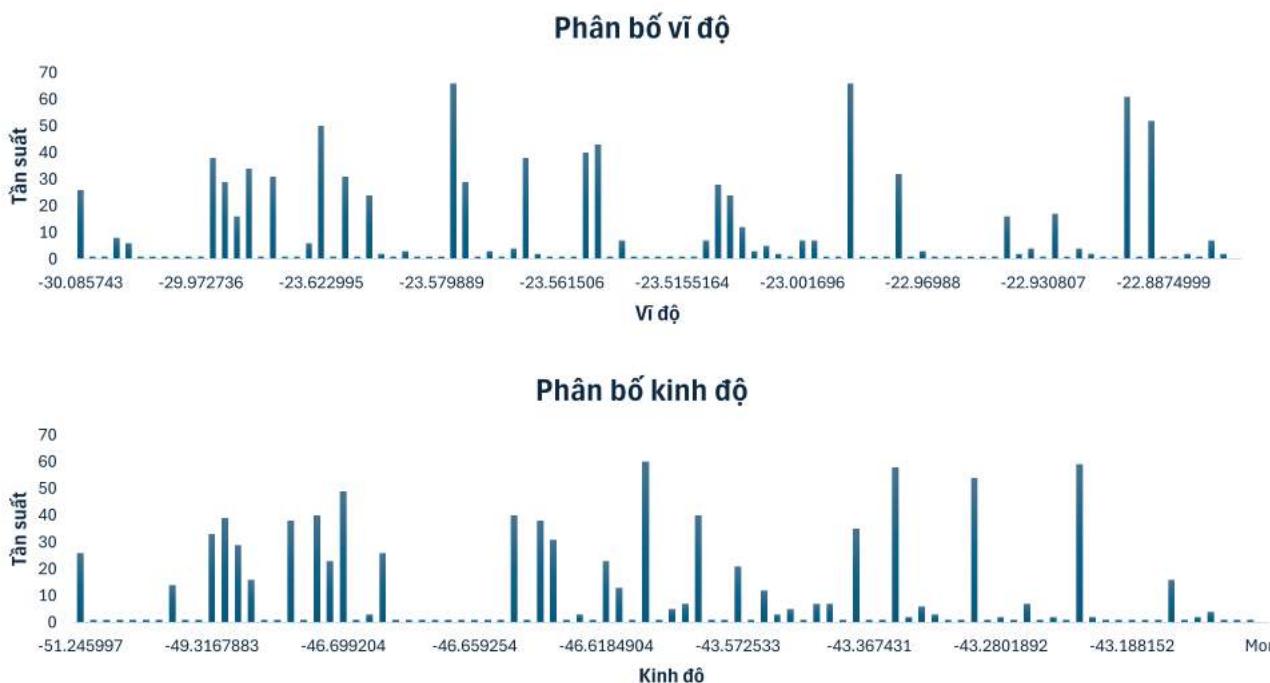


Tổng cộng có 2 loại mặt hàng vận chuyển :

- GOOD chiếm 60%.
- FOOD chiếm 40%.

=> Doanh nghiệp vận chuyển phần lớn mặt hàng GOOD.

j. Phân bố cửa hàng theo kinh độ và vĩ độ



- Khoảng kinh độ -46,6184904 và vĩ độ -23,62295 tập trung nhiều điểm dữ liệu, cho thấy phân bố cửa hàng ở khu vực này khá cao. Khả năng đây là khu thành phố lớn.
=> Doanh nghiệp có thể thấy được khu vực nào có lượng đơn hàng cao nhất, từ đó tập trung chiến dịch marketing và phân phối tại những khu vực này.

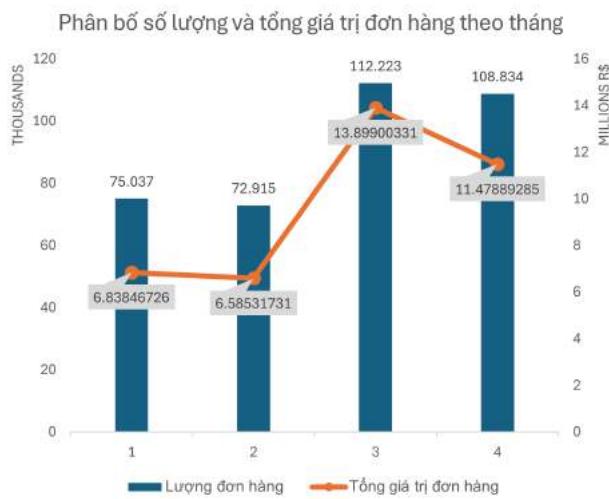
k. Phân bố kho theo bang và thành phố



- SP là nơi tập trung nhiều hub và kho nhất(15), RJ cũng có lượng hub và kho đáng kể(9).
 - 2 thành phố thuộc 2 bang này là Sao Paolo và Rio de Janeiro, và tập trung ít hơn tại 2 bang còn lại.

=> Doanh nghiệp tập trung mạnh mẽ vào SP và RJ về cả kho và hub. Từ đó thấy được sự quan trọng của 2 khu này.

1. Phân bố số lượng và tổng giá trị đơn hàng theo tháng

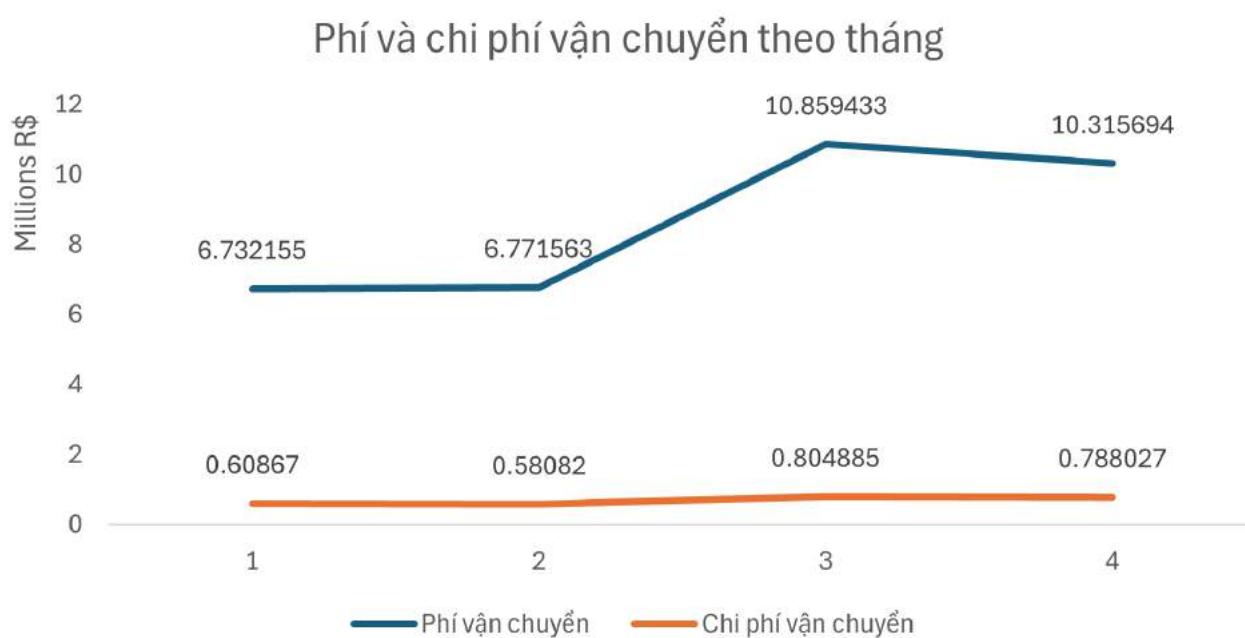


	Order amount			
	2021-1	2021-2	2021-3	2021-4
count	75037.0	72915.0	112223.0	108834.0
mean	91.13	90.31	123.85	105.47
std	114.54	131.61	5342.2	344.92
min	0.0	0.0	0.0	0.0
25%	40.7	39.3	39.4	41.4
50%	69.9	66.0	72.6	75.9
75%	114.2	111.6	127.7	128.8
max	17609.0	9999.99	1788306.11	100000.11

- Số lượng đơn hàng có sự gia tăng rõ rệt về số lượng đơn hàng từ tháng 2 đến tháng 3, sau đó giảm nhẹ vào tháng 4.
- Tổng giá trị đơn hàng tăng mạnh từ tháng 2 đến tháng 3, sau đó giảm vào tháng 4 nhưng vẫn cao hơn so với tháng 1 và tháng 2.
- Độ lệch chuẩn cao nhất vào tháng 3 (5,347.2) và thấp nhất vào tháng 1 (114.54), cho thấy sự biến động lớn về giá trị đơn hàng trong tháng 3.
- Giá trị 25%, 50%, và 75% có xu hướng tăng từ tháng 1 đến tháng 3 và giảm nhẹ vào tháng 4.
- Giá trị lớn nhất đạt đỉnh vào tháng 4 với 100,000.11 USD, cho thấy có một đơn hàng có giá trị rất cao trong tháng này.

=> Doanh nghiệp đã có những tháng tăng trưởng tích cực nhưng cần tiếp tục theo dõi và phân tích để duy trì và cải thiện kết quả này.

m. Phí và chi phí vận chuyển theo tháng

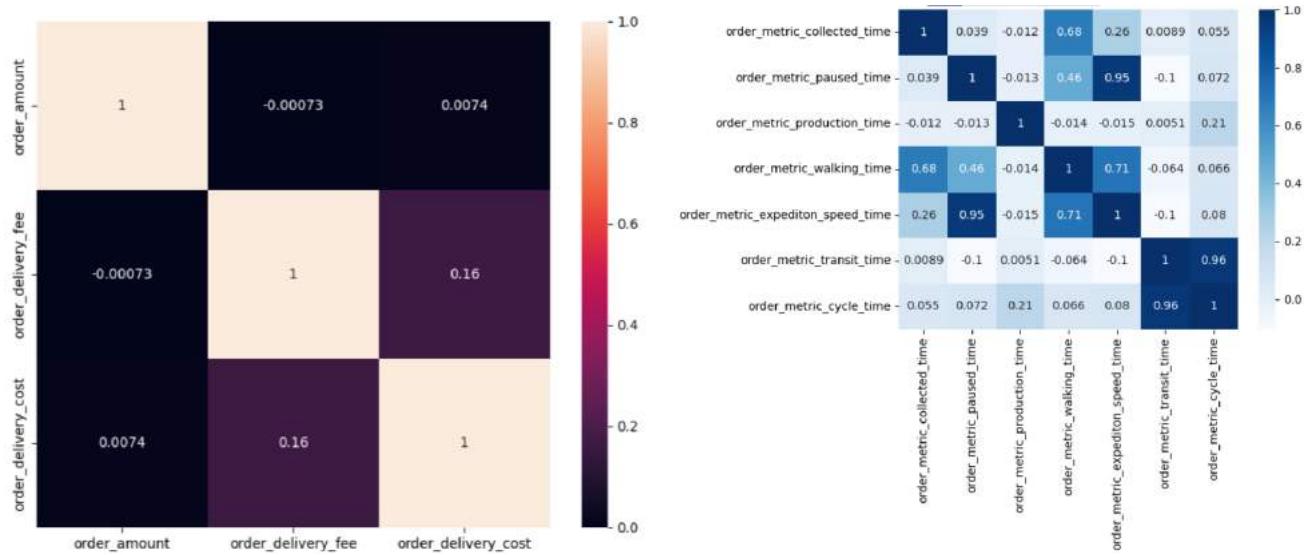


	Order delivery fee			
	2021-1	2021-2	2021-3	2021-4
count	75037.0	72915.0	112223.0	108834.0
mean	89.72	92.87	96.77	94.78
std	93.14	93.36	93.85	93.41
min	0.0	0.0	0.0	0.0
25%	12.0	12.0	12.0	12.0
50%	42.0	52.0	62.0	58.0
75%	164.0	169.0	174.0	170.0
max	300.0	373.0	670.0	990.0

	Order delivery fee			
	2021-1	2021-2	2021-3	2021-4
count	75037.0	72915.0	112223.0	108834.0
mean	89.72	92.87	96.77	94.78
std	93.14	93.36	93.85	93.41
min	0.0	0.0	0.0	0.0
25%	12.0	12.0	12.0	12.0
50%	42.0	52.0	62.0	58.0
75%	164.0	169.0	174.0	170.0
max	300.0	373.0	670.0	990.0

- Phí vận chuyển đều tăng mạnh vào tháng 3 và giảm nhẹ vào tháng 4.
- Chi phí vận chuyển trung bình giảm đến tháng 3 và tăng lại vào tháng 4.
- Chi phí vận chuyển tỷ lệ thuận với phí vận chuyển nhưng biến đổi khá ít (trong khoảng từ 0.6 đến 0.8 triệu R\$).

n. Tương quan giữa các dimension và fact



Ma trận tương quan fact - fact:

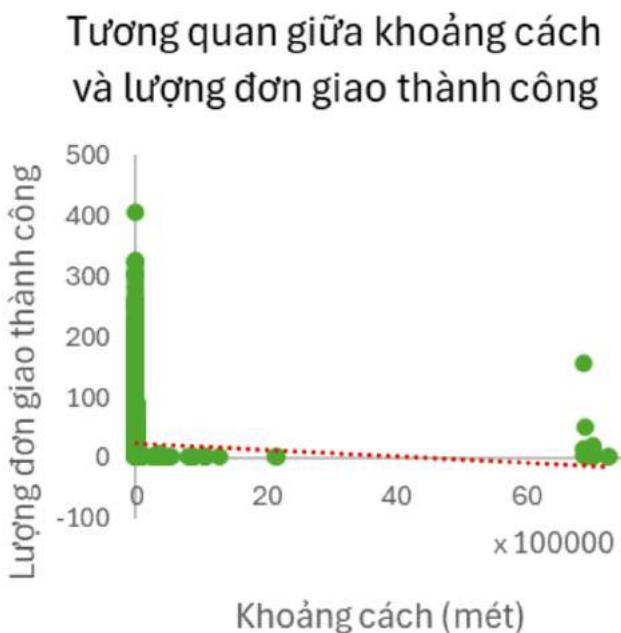
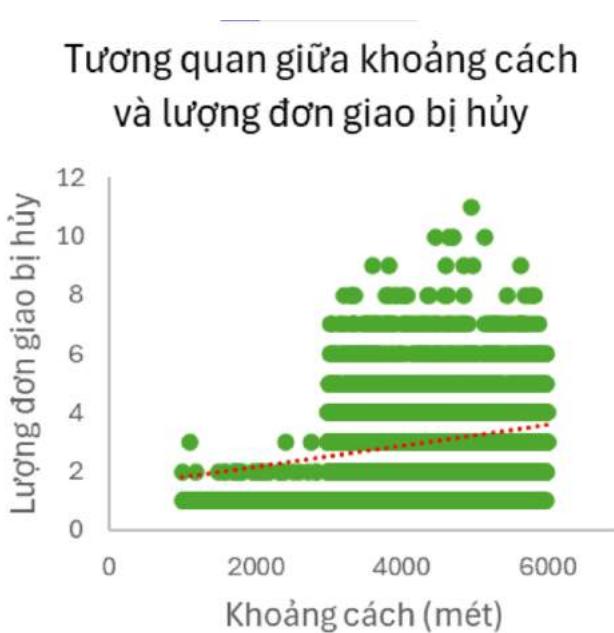
- Các tương quan rất yếu, gần như bằng 0.
- order_delivery_fee và order_delivery_cost: tương quan dương nhẹ (0.16).

Ma trận tương quan dim - dim:

- order_metric_walking_time có tương quan cao với order_metric_collected_time (0.68) và order_metric_cycle_time (0.71).
- order_metric_expedition_speed_time có tương quan rất cao với order_metric_collected_time (0.96).

=> Chí giao hàng và chi phí giao hàng có một chút liên quan, nhưng không đủ để xác định một mô hình rõ ràng. Các yếu tố liên quan đến thời gian thu thập, thời gian di chuyển và tốc độ giao hàng có mối liên hệ chặt chẽ, cho thấy việc cải thiện một yếu tố có thể ảnh hưởng tích cực đến các yếu tố còn lại.

- o.** Tương quan giữa khoảng cách và lượng đơn hàng theo trạng thái



- Khoảng cách vận chuyển ngắn: Tỷ lệ thành công cao hơn và ít đơn hàng bị hủy. Điều này có thể do quá trình vận chuyển dễ dàng hơn, ít rủi ro và chi phí thấp hơn.
- Khoảng cách vận chuyển dài: Tỷ lệ thành công giảm và số lượng đơn hàng bị hủy tăng. Điều này có thể do các khó khăn và rủi ro gia tăng khi vận chuyển xa, làm cho doanh nghiệp cần xem xét các biện pháp cải thiện hiệu quả vận chuyển cho các khoảng cách xa.

⇒ Doanh nghiệp có thể hiểu rõ hơn về ảnh hưởng của khoảng cách vận chuyển đến tỷ lệ thành công và hủy đơn, từ đó có thể điều chỉnh chiến lược vận hành và cải thiện dịch vụ.

- p.** Phân bố tình trạng đơn hàng theo tháng



Số lượng đơn hoàn thành :

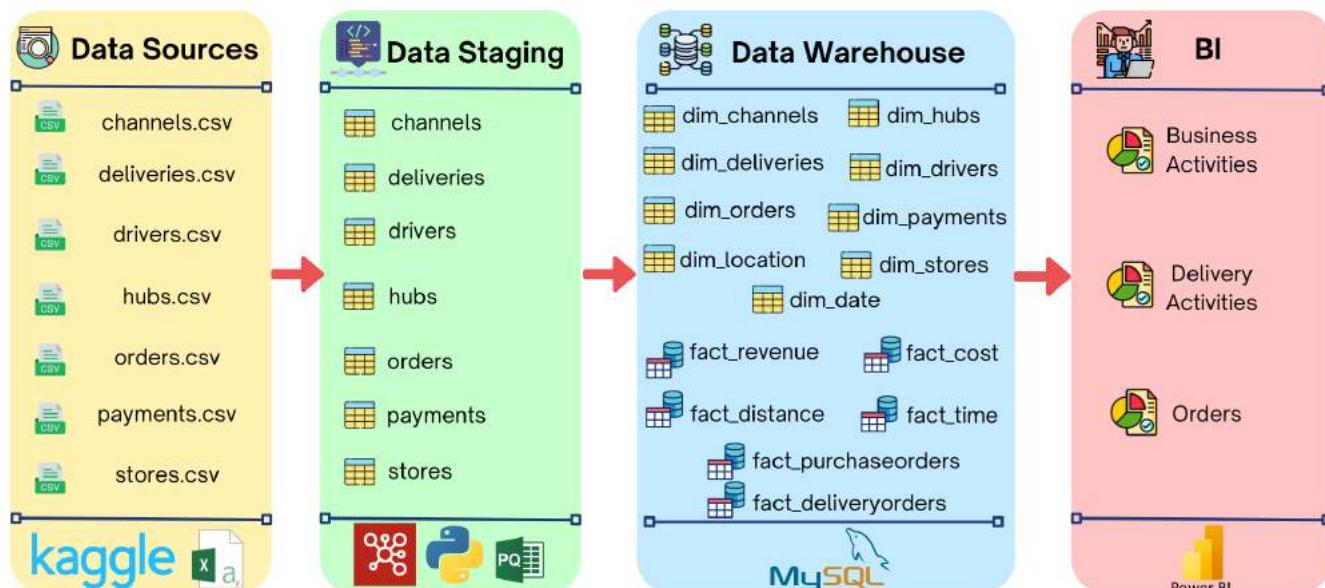
- Có sự giảm nhẹ từ tháng 1 (71,773 đơn) sang tháng 2 (69,653 đơn).
- Tăng mạnh từ tháng 2 (69,653 đơn) lên tháng 3 (107,232 đơn).
- Giảm nhẹ từ tháng 3 (107,232 đơn) sang tháng 4 (103,362 đơn)

Số lượng đơn hủy :

- Giữ ổn định từ tháng 1 (3,264 đơn) sang tháng 2 (3,262 đơn).
- Tăng từ tháng 2 (3,262 đơn) lên tháng 3 (4,991 đơn).
- Tăng tiếp từ tháng 3 (4,991 đơn) sang tháng 4 (5,472 đơn).

⇒ Tháng 3 có sự gia tăng đáng kể về số lượng đơn hàng hoàn thành, có thể do yếu tố mùa vụ hoặc chiến dịch bán hàng hiệu quả. Số lượng đơn hàng bị hủy tăng dần theo thời gian, điều này có thể phản ánh những vấn đề về chất lượng dịch vụ hoặc thay đổi nhu cầu khách hàng. Do đó, doanh nghiệp cần phân tích kỹ lưỡng nguyên nhân của việc hủy đơn để cải thiện dịch vụ và giảm tỷ lệ hủy đơn trong tương lai.

3.2.2 Kiến trúc Data Warehouse

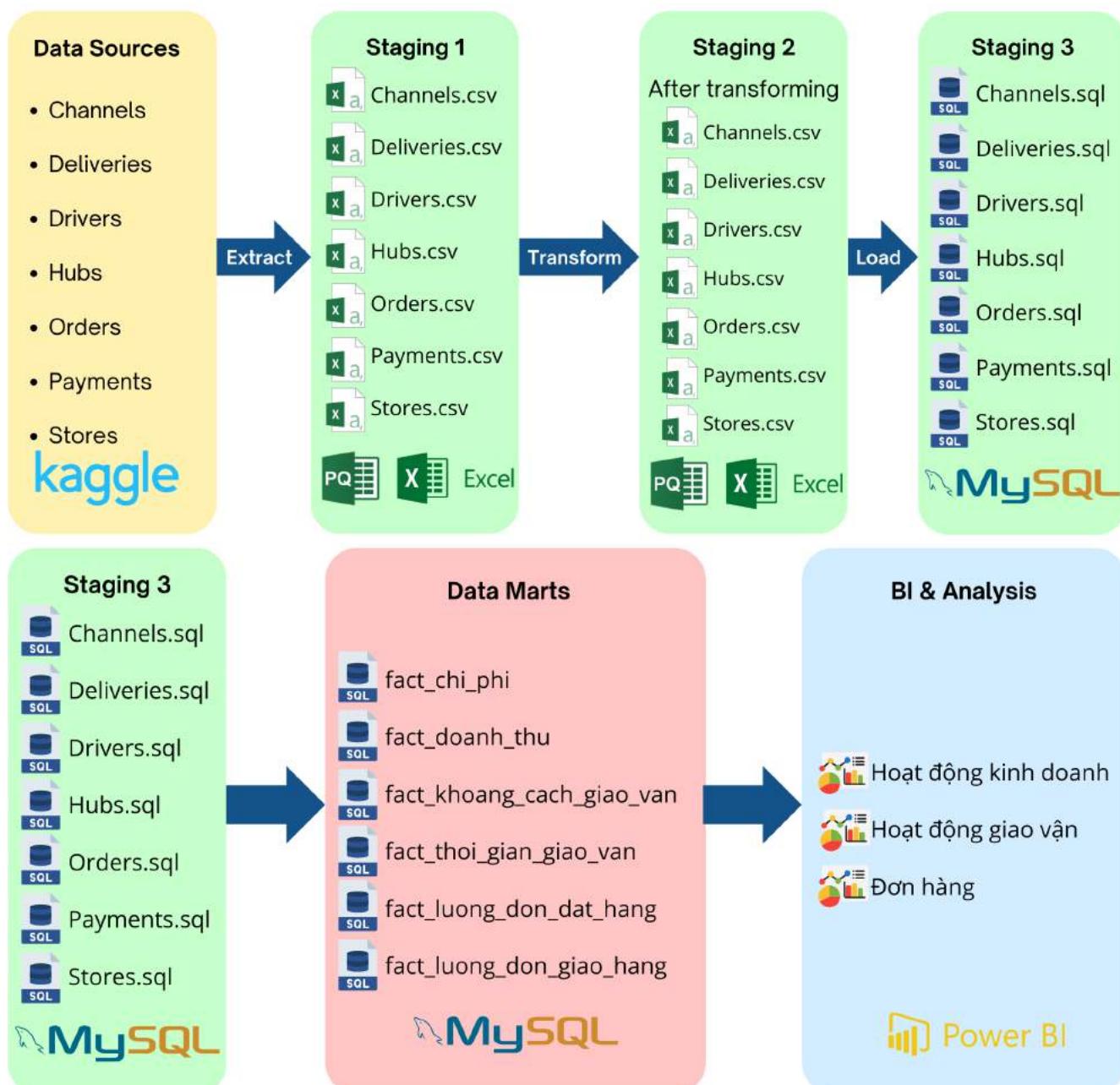


- Data Source: Đây là nơi dữ liệu được thu thập từ một hoặc nhiều nguồn khác nhau. Cụ thể ở đây là trên nền tảng Kaggle, bao gồm có 7 bảng là: channels.csv, deliveries.csv, drivers.csv, hubs.csv, orders.csv, payments.csv, stores.csv.
- Data Staging: Đây là vùng trung chuyển dữ liệu, là giai đoạn mà dữ liệu được làm sạch, chuẩn hóa và biến đổi để chuẩn bị cho việc thiết lập các bảng dim, fact trong Data Warehouse. Quá trình này thường được thực hiện thông qua các bước ETL (Extract, Transform, Load). Công nghệ sử dụng trong quá trình này là PowerQuery của Excel và ngôn ngữ lập trình Python, bên cạnh đó là ETL Tools Pentaho nếu xử lý khối lượng dữ liệu lớn.

- Data Warehouse: Đây là nơi dữ liệu đã được làm sạch và biến đổi được lưu trữ dưới dạng cơ sở dữ liệu lớn, tổ chức theo các chủ đề(fact) và khía cạnh(dim) khác nhau. Cụ thể ở đây trong Data Warehouse có 6 fact(fact_revenue, fact_cost, fact_distance, fact_time, fact_purchaseorders, fact_deliveryorders) và 9 dim(dim_channels, dim_hubs, dim_deliveries, dim_drivers, dim_orders, dim_payments, dim_location, dim_stores, dim_date). Công nghệ sử dụng ở đây là ngôn ngữ MySQL.
- BI: Đây là lớp cuối cùng trong kiến trúc Data Warehouse, nơi dữ liệu được trình bày cho người dùng cuối thông qua các công cụ BI (Business Intelligence) và phân tích. Cụ thể ở đây là phân tích đơn hàng, phân tích hoạt động giao vận, phân tích hoạt động kinh doanh. Công nghệ được sử dụng ở đây là Power BI.

3.2.3 Nội dung ETL

1. Data Pipeline



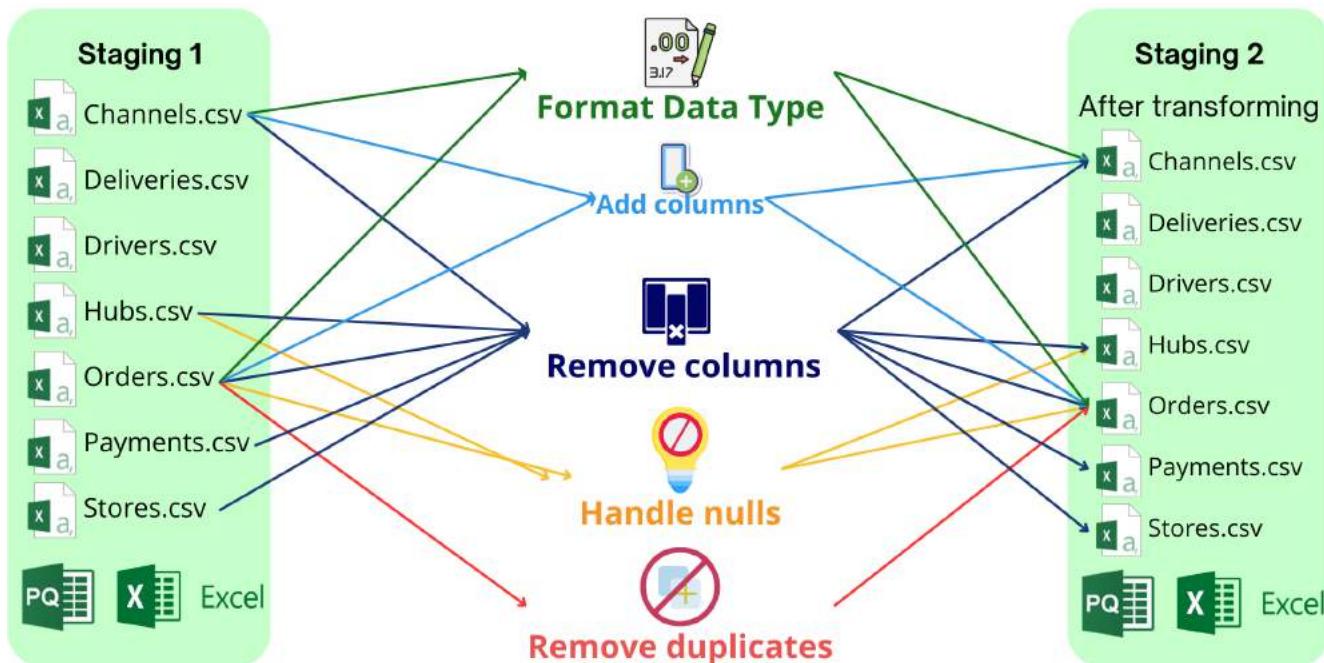
Quy trình xử lý dữ liệu được thể hiện rất rõ ở trong Data Pipeline. Cụ thể các công đoạn xử lý dữ liệu như sau:

- Data Sources: Nguồn dữ liệu bao gồm các tập tin dữ liệu từ Kaggle với các tên: channels, deliveries, drivers, hubs, orders, payments, stores.
- Staging 1 (Giai đoạn trung chuyển 1) Extract (Trích xuất dữ liệu): Các tập tin dữ liệu nguồn (.csv) từ Kaggle được trích xuất và lưu trữ trong Staging 1. Dữ liệu từ các nguồn này được chuẩn bị để qua giai đoạn tiếp theo.
- Staging 2 (Giai đoạn trung chuyển 2) Transform (Biến đổi dữ liệu): Dữ liệu từ Staging 1 được làm sạch, chuẩn hóa, và biến đổi để chuẩn bị cho việc tải vào Data Warehouse. Sau khi biến đổi, dữ liệu được lưu trữ trong Staging 2, vẫn dưới dạng các tập tin .csv.
- Staging 3 (Giai đoạn trung chuyển 3) Load (Tải dữ liệu): Dữ liệu từ Staging 2 sau khi biến đổi được tải vào hệ quản trị cơ sở dữ liệu MySQL. Dữ liệu được lưu trữ dưới dạng các bảng trong cơ sở dữ liệu MySQL (Channels.sql, Deliveries.sql, Drivers.sql, Hubs.sql, Orders.sql, Payments.sql, Stores.sql).
- Data Marts: Các bảng dữ liệu từ Staging 3 được tổ chức và tổng hợp thành các Data Mart. Data Mart là các cơ sở dữ liệu chuyên biệt chứa dữ liệu tổng hợp, tập trung vào các chủ đề hoặc lĩnh vực cụ thể. Các Data Mart ở đây bao gồm: fact_chi_phi, fact_doanh_thu, fact_khoang_cach_giao_van, fact_thoi_gian_giao_van, fact_luong_don_dat_hang, fact_luong_don_giao_hang.
- BI và Analysis: Dữ liệu từ các Data Mart được sử dụng để tạo các báo cáo và biểu đồ phân tích. Các công cụ BI như Power BI được sử dụng để trực quan hóa dữ liệu và cung cấp thông tin hỗ trợ ra quyết định. Các lĩnh vực phân tích được chỉ ra bao gồm:
 - Hoạt động kinh doanh.
 - Hoạt động giao vận.
 - Đơn hàng.

2. Quy trình ETL

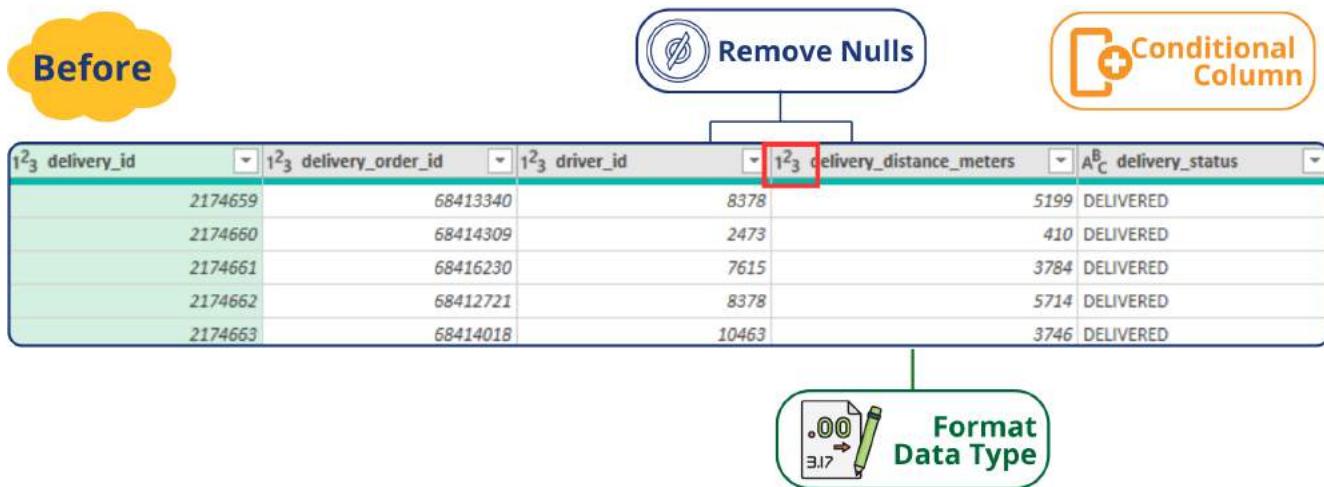
Từ Data Source sang Data Staging	Từ Data Staging sang Data Warehouse
<ul style="list-style-type: none"> • Loại bỏ trường dữ liệu không cần thiết • Xóa dữ liệu trùng lặp • Xử lý null • Định dạng lại kiểu dữ liệu • Thêm trường dữ liệu • Phân cụm dữ liệu • Chính sửa dữ liệu đảm bảo tính nhất quán • Dò trả về đúng giá trị 	<ul style="list-style-type: none"> • Tạo bảng tổng hợp chứa đầy đủ thông tin của các bảng trong database • Dùng các lệnh truy vấn tạo ra các fact, các dim 

3. ETL dữ liệu từ Data Source sang Data Staging



Bây giờ ta sẽ đi ETL các bảng bằng cách định dạng lại dữ liệu, thêm cột, xóa cột, xử lý nulls và xóa các giá trị trùng lặp.

a. Bảng deliveries



Ở bảng deliveries, ta cần định dạng lại dữ liệu của cột delivery_distance_meters từ số nguyên thành số thập phân.

Sau đó ta cần xử lý các giá trị nulls trong cột driver_id và delivery_distance_meters bằng cách sử dụng Filter Rows để loại bỏ các ô trống.

Cuối cùng, ta sử dụng tính năng Add Conditional Column để thêm cột phân loại nhóm khoảng cách dựa trên giá trị của cột delivery_distance_meters như sau

- Nếu deliver_distance_meters < 5000m thì giá trị sẽ là "Small".
- Nếu $5000m \leq \text{deliver_distance_meters} < 10000m$ thì trả về giá trị "Medium".
- Nếu $10000m \leq \text{deliver_distance_meters} < 15000m$ thì giá trị là "Large".

- Còn lại nếu $> 15000m$ sẽ là giá trị "Very Large".

Transform

i ² _3 delivery_id	i ² _3 delivery_order_id	i ² _3 driver_id	1.2 delivery_distance_meters	A ^B delivery_status	i ² delivery_range
2174659	68413340	8378	5199	DELIVERED	Medium
2174660	68414309	2473	410	DELIVERED	Small
2174661	68416230	7615	3784	DELIVERED	Small
2174662	68412721	8378	5714	DELIVERED	Medium
2174663	68414018	10463	3746	DELIVERED	Small

After

Kết quả bảng deliveries sau khi ETL đã định dạng lại được dữ liệu và thêm được cột phân loại khoảng cách.

b. Bảng hubs

Before

hub_id	hub_name	hub_city	hub_state	hub_latitude	hub_longitude
4	RED SHOPPING	PORTO ALEGRE	RS	-30.0219	-51.2084
5	FUNK SHOPPING	RIO DE JANEIRO	RJ	-23.0007	-43.3183
8	GOLDEN SHOPPING	RIO DE JANEIRO	RJ	-22.9215	-43.2348
13	HIP HOP SHOPPING	RIO DE JANEIRO	RJ	-22.8858	-43.2792
16	PEOPLE SHOPPING	RIO DE JANEIRO	RJ	-23.0175	-43.4799
17	SMALL SHOPPING	SÃO PAULO	SP	-23.592	-46.6365

Ta thấy ở bảng hubs, cột hub_state chứa các giá trị viết tắt của tên bang. Điều này khiến cho việc đọc dữ liệu trở nên khó khăn vì có thể người phân tích sẽ không biết. Vì vậy ở đây ta dùng hàm VLOOKUP trong Excel để dò tìm và trả về giá trị đầy đủ của tên bang.

Cột hub_city chứa giá trị SÃO PAULO, không đúng với tên thành phố ở Brazil nên ta cần thay đổi giá trị thành giá trị đúng là SÃO PAULO. Bằng cách sử dụng chức năng Remove Columns, ta có thể dễ dàng thực hiện được điều này.

Vì đã có thông tin về thành phố và bang của hub nên hai cột hub_latitude và hub_longitude trở nên dư thừa, ta cần xóa bỏ hai cột này bằng chức năng Remove Columns.

Transform

hub_id	hub_name	hub_city	hub_state	hub_latitude	hub_longitude	hub_state	hub_state
2	BLUE SHOPPING	PORTO ALEGRE	RS	-30.0474148	-51.21351	RIO GRANDE DO SUL	RIO GRANDE DO SUL
3	GREEN SHOPPING	PORTO ALEGRE	RS	-30.0374149	-51.20352	RIO GRANDE DO SUL	RIO DE JANEIRO
4	RED SHOPPING	PORTO ALEGRE	RS	-30.0219481	-51.2083816	RIO GRANDE DO SUL	SP
5	FUNK SHOPPING	RIO DE JANEIRO	RJ	-23.0007498	-43.318282	RIO DE JANEIRO	PARANÁ
8	GOLDEN SHOPPING	RIO DE JANEIRO	RJ	-22.921475	-43.234774	RIO DE JANEIRO	
13	HIP HOP SHOPPING	RIO DE JANEIRO	RJ	-22.8858199	-43.2792183	RIO DE JANEIRO	
16	PEOPLE SHOPPING	RIO DE JANEIRO	RJ	-23.0174723	-43.4799389	RIO DE JANEIRO	=VLOOKUP(D7,\$J\$1:\$K\$5,2, FALSE)

Replace Values

Replace one value with another in the selected columns.

Value To Find

SÃO PAULO

Replace With

SÃO PAULO

Advanced options

APPLIED STEPS

Source

Changed Type

Replaced Value



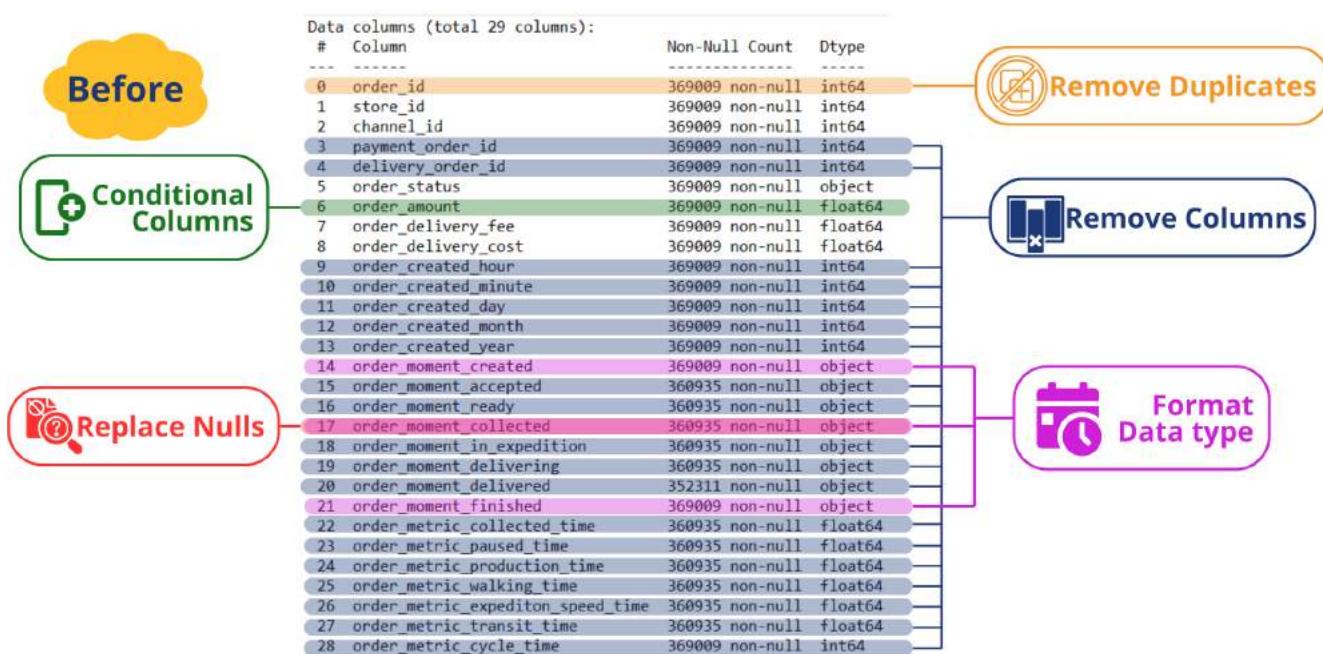
Removed Columns

Kết quả sau khi ETL, các giá trị lỗi đã được thay bằng giá trị đúng và tên bang đã được đổi trả về đúng giá trị đầy đủ.

After

hub_id	hub_name	hub_city	hub_state
16	PEOPLE SHOPPING	RIO DE JANEIRO	RIO DE JANEIRO
17	SMALL SHOPPING	SÃO PAULO	SÃO PAULO
18	STAR SHOPPING	RIO DE JANEIRO	RIO DE JANEIRO
20	PURPLE SHOPPING	RIO DE JANEIRO	RIO DE JANEIRO
21	WOLF SHOPPING	SÃO PAULO	SÃO PAULO
22	COLOR SHOPPING	RIO DE JANEIRO	RIO DE JANEIRO
25	AVENUE SHOPPING	SÃO PAULO	SÃO PAULO
26	SQL SHOPPING	SÃO PAULO	SÃO PAULO

c. Bảng orders



Replace Values

Replace one value with another in the selected columns.

Value To Find:

Replace With:

APPLIED STEPS

- Source
- Change Type
- Promoted Headers
- Removed Other Columns
- Removed Duplicates
- Changed Type
- Replaced Value
- Added Conditional Column**

Transform

Add Conditional Column

Add a conditional column that is computed from the other columns or values.

New column name:

Column Name	Operator	Value	Output
If <input type="text" value="order_amount"/>	<input type="text" value="is less than"/>	<input type="text" value="ABC 123"/> <input type="text" value="100"/>	Then <input type="text" value="ABC 123"/> <input type="text" value="Low"/>
Else If <input type="text" value="order_amount"/>	<input type="text" value="is less than"/>	<input type="text" value="ABC 123"/> <input type="text" value="1000"/>	Then <input type="text" value="ABC 123"/> <input type="text" value="Medium"/>
Else If <input type="text" value="order_amount"/>	<input type="text" value="is less than"/>	<input type="text" value="ABC 123"/> <input type="text" value="5000"/>	Then <input type="text" value="ABC 123"/> <input type="text" value="High"/> ...

Add Clause

Else

Bảng orders có rất nhiều cột dư thừa (là các cột tính toán thời gian giữa các công đoạn). Ta chỉ cần giữ lại các cột có thông tin quan trọng như order_id, store_id, channel_id, order_status, order_amount, order_delivery_fee, order_delivery_cost, hoặc thậm chí là có thể suy ra được thông tin của những cột dư thừa như order_moment_created, order_moment_collected và order_moment_finished.

Tiếp theo, ta cần xóa những đơn hàng có id bị trùng lặp bằng công cụ Remove Duplicates trong Power Query.

Trong cột order_moment_collected có một số thời gian bị bỏ trống (do đơn hàng bị hủy nên không có thời gian lấy hàng), ta thay các giá trị rỗng này bằng giá trị khác biệt hoàn toàn là 1/1/1900 0:0:0 để nhận diện và tiện xử lý sau này.

Đồng thời ta cần phải định dạng lại dữ liệu Datetime của các cột ngày đặt, ngày lấy hàng và ngày hoàn thiện đơn.

Cuối cùng, ta thêm một cột phân loại giá đơn hàng theo dữ liệu của cột order_amount như sau

- Nếu $order_amount < 100\$R$ thì được xếp vào loại "Low".
- Nếu $100\$R \leq order_amount < 1000\R thì được xếp vào loại "Medium".
- Nếu $1000\$R \leq order_amount < 2000\R thì được xếp vào loại "High".
- Còn lại sẽ là các đơn hàng với loại giá "Very high".

After

1.2 order_amount	1.2 order_delivery_fee	1.2 order_delivery_cost	order_moment_created	order_moment_collected	order_moment_finished	ABC 123 order_amount_value
62.7	130	20	1/1/2021 12:01:36 AM	1/1/1900 12:00:00 AM	1/1/2021 12:06:36 AM	Low
62.7	178	20	1/1/2021 12:04:26 AM	1/1/1900 12:00:00 AM	1/1/2021 12:09:26 AM	Low
115.5	83	20	1/1/2021 12:13:07 AM	1/1/1900 12:00:00 AM	1/1/2021 12:18:07 AM	Medium
55.9	177	20	1/1/2021 12:19:15 AM	1/1/1900 12:00:00 AM	1/1/2021 12:24:15 AM	Low
37.9	201	20	1/1/2021 12:26:25 AM	1/1/1900 12:00:00 AM	1/1/2021 12:29:25 AM	Low
80	109	20	1/1/2021 12:56:19 AM	1/1/1900 12:00:00 AM	1/1/2021 12:57:19 AM	Low

d. Bảng payments

Trong bài báo cáo này, chúng em không cần 2 cột payment_amount và payment_fee nên sẽ loại bỏ bằng công cụ Remove Columns.

Before



1 ² 3 payment_id	1 ² 3 payment_order_id	1.2 payment_amount	1.2 payment_fee	A ^B payment_method	A ^B payment_status
4427918	68410055	281.9	167.6	VOUCHER	PAID
4427919	68410055	281.9	111	ONLINE	PAID
4427920	68412721	207.1	49.1	ONLINE	PAID
4427921	68413340	58.9	106.4	ONLINE	PAID
4427922	68414018	124.8	184.8	ONLINE	PAID

After

1 ² 3 payment_id	1 ² 3 payment_order_id	A ^B payment_method	A ^B payment_status
4427918	68410055	VOUCHER	PAID
4427919	68410055	ONLINE	PAID
4427920	68412721	ONLINE	PAID
4427921	68413340	ONLINE	PAID
4427922	68414018	ONLINE	PAID

e. Bảng stores

Trong bài báo cáo này, chúng em không cần cột store_plan_price nên sẽ loại bỏ bằng công cụ Remove Columns.

Before

After

store_id	hub_id	store_name	store_segment	store_plan_price	store_latitude	store_longitude
3	2	CUMIURI	FOOD	0.0	-30.0474148	-51.21351
6	3	PIMGUCIS DA VIVA	FOOD	0.0	-30.0374149	-51.20352
8	3	RASMUR S	FOOD	0.0	-30.0374149	-51.20352
53	8	PAPA SUCIS	FOOD	0.0	-22.921475	-43.234822
54	8	VUZPI PAZZIS	FOOD	0.0	-22.921475	-43.234822

f. Tự động hóa bằng Python

Import CSV to MySQL

```

1 import os
2 import csv
3 import mysql.connector
4 from datetime import datetime
5 # Ket noi den co so du lieu
6 database_name = 'data_python'
7 # Ket noi den MySQL server
8 mydb = mysql.connector.connect(
9     host="localhost",
10    user="root",
11    password="dotrungquan183@",
12    database=database_name
13 )

```

Đoạn code này kết nối tới một cơ sở dữ liệu MySQL có tên là 'data_python' trên máy localhost, sử dụng người dùng 'root' và mật khẩu 'dotrungquan183@'. Thư viện mysql.connector được sử dụng để thiết lập và quản lý kết nối này. Sau khi kết nối thành công, biến mydb sẽ chứa đối tượng kết nối, từ đó cho phép ta thực hiện các truy vấn SQL và tương tác với cơ sở dữ liệu từ chương trình Python.

```

1 # Tao con tro de thao tac voi co so du lieu
2 mycursor = mydb.cursor()
3
4 # Duong dan toi muc chua cac file csv
5 folder_path = 'C:/Users/ASUS/Desktop/OLTP'
6
7 # Kiem tra xem thu muc co ton tai khong
8 if not os.path.exists(folder_path):
9     print(f"Thu muc '{folder_path}' khong ton tai.")
10    exit(1)
11
12 # Liet ke tat ca cac file csv trong thu muc
13 csv_files = [f for f in os.listdir(folder_path) if f.endswith('.csv')]

```

Ta bắt đầu bằng việc tạo một con trỏ (cursor) để thao tác với cơ sở dữ liệu MySQL đã kết nối (mydb.cursor()). Sau đó, ta sẽ xác định đường dẫn tới thư mục chứa các file CSV (folder_path). Tiếp theo, ta kiểm tra xem thư mục này có tồn tại không và hiển thị thông báo lỗi nếu không tồn tại. Cuối cùng, đoạn code sẽ liệt kê tất cả các file trong thư mục có đuôi .csv và lưu danh sách các tên file này vào biến csv_files, chuẩn bị cho việc nhập dữ liệu từ các file CSV vào cơ sở dữ liệu MySQL.

```

1   for csv_file in csv_files:
2       file_path = os.path.join(folder_path, csv_file)
3
4       # Doc file CSV de xac dinh cau truc bang
5       with open(file_path, 'r') as csvfile:
6           csv_reader = csv.reader(csvfile)
7           # Lay dong tieu de dau tien
8           header = next(csv_reader)
9
10      # Tim dong khong chua gia tri NULL de xac dinh kieu du lieu
11      for row in csv_reader:
12          # Kiem tra xem dong co chua gia tri NULL khong
13          if '' not in row:
14              first_row = row
15              break
16
17      # Xac dinh ten bang tu ten file CSV (bo phan mo rong)
18      table_name = os.path.splitext(csv_file)[0]

```

Đoạn code trên là một vòng lặp lặp dùng để xử lý nhiều file CSV trong một thư mục. Mỗi file CSV được mở và đọc để xác định cấu trúc của bảng dữ liệu. Đầu tiên, từ mỗi file CSV, đoạn mã xác định tiêu đề của bảng từ dòng đầu tiên của file CSV. Sau đó, nó duyệt qua từng dòng trong file để tìm dòng đầu tiên không chứa giá trị NULL, từ đó xác định được kiểu dữ liệu của các cột trong bảng. Cuối cùng, tên của bảng được xác định từ tên của file CSV bằng cách loại bỏ phần mở rộng của tên file.

```

1  # Xac dinh kieu du lieu cua tung cot dua tren gia tri dong dau tien
2  # Mac dinh la VARCHAR(255) cho tat ca cac cot
3  column_types = ['VARCHAR(255)' for _ in first_row]
4  for i, value in enumerate(first_row):
5      try:
6          int(value)
7          column_types[i] = 'BIGINT'
8      except ValueError:
9          try:
10             float(value)
11             column_types[i] = 'FLOAT'
12         except ValueError:
13             try:
14                 datetime.strptime(value, '%m/%d/%Y %H:%M:%S')
15                 column_types[i] = 'DATETIME'
16             except ValueError:
17                 pass
18
19 columns = ', '.join([f'{col} {col_type}' for col, col_type in zip(
20     header, column_types)])

```

```

21 # Tao bang neu chua ton tai
22 mycursor.execute(f """
23     CREATE TABLE IF NOT EXISTS {table_name} (
24         {columns}
25     )
26 """)

```

Ta xác định kiểu dữ liệu của từng cột trong bảng dữ liệu dựa trên giá trị của dòng đầu tiên trong file CSV. Đầu tiên, ta duyệt qua từng giá trị trong first_row, thử chuyển đổi thành int, float, và datetime để xác định kiểu dữ liệu phù hợp cho mỗi cột. Sau đó, ta sử dụng câu lệnh SQL CREATE TABLE để tạo bảng trong cơ sở dữ liệu nếu bảng chưa tồn tại, sử dụng tên bảng và cấu trúc cột đã xác định từ các bước trước đó.

```

1 # Dua con tro tro lai dau file de doc du lieu lan thu hai
2 csvfile.seek(0)
3 next(csv_reader) # Bo qua dong tieu de
4
5 # Chen du lieu vao bang
6 for row in csv_reader:
7     for i, value in enumerate(row):
8         if value == '':
9             row[i] = None
10    elif column_types[i] == 'DATETIME':
11        try:
12            # Chuyen doi dang dang ngay/gio tu MM/DD/YYYY HH:MM
13            #:SS sang YYYY-MM-DD HH:MM:SS
14            row[i] = datetime.strptime(value, '%m/%d/%Y %H:%M:%S')
15            S').strftime('%Y-%m-%d %H:%M:%S')
16        except ValueError:
17            pass
18    elif column_types[i] == 'FLOAT':
19        try:
20            row[i] = float(value)
21        except ValueError:
22            row[i] = None
23    elif column_types[i] == 'BIGINT':
24        try:
25            row[i] = int(value)
26        except ValueError:
27            row[i] = None

```

Đoạn mã trên có chức năng quan trọng là chèn dữ liệu từ các file CSV vào bảng trong cơ sở dữ liệu. Sau khi xác định cấu trúc bảng và kiểu dữ liệu của từng cột từ dòng đầu tiên của file CSV, nó sử dụng hai vòng lặp lồng nhau để duyệt qua từng dòng và từng giá trị trong file CSV. Đầu tiên, nó đặt lại con trỏ đọc file về đầu để bắt đầu đọc lại từ đầu file sau khi đã bỏ qua dòng tiêu đề. Dòng lặp ngoài duyệt qua từng dòng dữ liệu trong file CSV để chèn vào bảng, trong khi dòng lặp trong duyệt qua từng giá trị trong dòng hiện tại để xử lý từng cột tương ứng. Nếu giá trị của cột là rỗng (""), nó thay thế bằng None để phù hợp với quy ước cơ sở dữ liệu. Nếu cột có kiểu dữ liệu là 'DATETIME', nó cố gắng chuyển đổi định dạng ngày/giờ từ MM/DD/YYYY HH:MM sang YYYY-MM-DD HH:MM để lưu trữ đúng trong cơ sở dữ liệu. Các giá trị không phù hợp với kiểu dữ liệu 'DATETIME', 'FLOAT', hoặc 'BIGINT' sẽ được gán bằng None để tránh lỗi khi chèn dữ liệu vào bảng. Đoạn mã này đảm bảo rằng dữ liệu từ file CSV được chuẩn bị và chèn vào cơ sở dữ liệu một cách an toàn và

đúng đắn theo cấu trúc bảng đã xác định trước đó.

```

1     placeholders = ', '.join(['%s' for _ in row])
2 sql = f"INSERT INTO {table_name} ({', '.join(header)}) VALUES ({placeholder})
3 try:
4     mycursor.execute(sql, row)
5 except mysql.connector.errors.DatabaseError as e:
6     if "Data truncated for column" in str(e):
7         # Lay ten cot tu thong bao loi
8         error_message = str(e)
9         start_index = error_message.find("')") + 1
10        end_index = error_message.find("'", start_index)
11        column_name = error_message[start_index:end_index]
12        # Thay doi kieu du lieu cua cot thanh VARCHAR(255)
13        alter_sql = f"ALTER TABLE {table_name} MODIFY COLUMN {column_name} VARCHAR(255)"
14    try:
15        mycursor.execute(alter_sql)
16        mycursor.execute(sql, row) # Thu chen lai du lieu sau
17        # khi thay doi kieu du lieu
18    except mysql.connector.errors.DatabaseError as alter_e:
19        print(f"Loi khi thay doi kieu du lieu cua cot {column_name}: {alter_e}")
20    else:
21        print(f"Loi khi chen du lieu vao bang {table_name}: {e}")
22        print(f"Dong du lieu gay loi: {row}")
23
24 # Commit cac thay doi
25 mydb.commit()
26 # Dong con tro va ket noi
27 mycursor.close()
28 mydb.close()
29
30 print("Du lieu da duoc nhap thanh cong vao co so du lieu MySQL.")

```

Đoạn code trên được sử dụng để chèn dữ liệu từ các file CSV vào bảng trong cơ sở dữ liệu MySQL và xử lý các vấn đề phát sinh trong quá trình này. Đầu tiên, nó tạo chuỗi placeholders để định nghĩa các tham số trong câu lệnh SQL INSERT, đảm bảo rằng mỗi giá trị từ row sẽ được thay thế bằng %s. Sau đó, câu lệnh INSERT được tạo ra với tên bảng và danh sách các cột (header), cùng với các giá trị tương ứng. Nếu việc chèn dữ liệu gặp lỗi, đoạn mã kiểm tra xem lỗi có phải do "Data truncated for column" hay không. Trong trường hợp này, nó sẽ điều chỉnh kiểu dữ liệu của cột gây lỗi thành VARCHAR(255) để giải quyết vấn đề và thử chèn lại dữ liệu. Nếu không phải lỗi này, nó in ra thông báo lỗi chung và thông tin về dòng dữ liệu gây ra vấn đề. Sau khi hoàn thành quá trình chèn dữ liệu và xử lý lỗi, các thay đổi được commit vào cơ sở dữ liệu để lưu trữ và đảm bảo tính nhất quán. Cuối cùng, đoạn mã đóng các tài nguyên như con trỏ và kết nối đến cơ sở dữ liệu, và thông báo khi quá trình nhập dữ liệu hoàn tất thành công.

Remove Columns

Before**Table: hubs****Columns:**

hub_id
hub_name
hub_city
hub_state
hub_latitude
hub_longitude

```
Data columns (total 29 columns):
 #  Column
 --- -----
 0  order_id
 1  store_id
 2  channel_id
 3  payment_order_id
 4  delivery_order_id
 5  order_status
 6  order_amount
 7  order_delivery_fee
 8  order_delivery_cost
 9  order_created_hour
10  order_created_minute
11  order_created_day
12  order_created_month
13  order_created_year
14  order_moment_created
15  order_moment_accepted
16  order_moment_ready
17  order_moment_collected
18  order_moment_in_expedition
19  order_moment_delivering
20  order_moment_delivered
21  order_moment_finished
22  order_metric_collected_time
23  order_metric_paused_time
24  order_metric_production_time
25  order_metric_walking_time
26  order_metric_expedition_speed_time
27  order_metric_transit_time
28  order_metric_cycle_time
```

**Table: payments****Columns:**

payment_id
payment_order_id
payment_amount
payment_fee
payment_method
payment_status

```

1  # Các câu lệnh SQL để xóa các cột
2  drop_columns_queries = [
3      "ALTER TABLE hubs DROP COLUMN hub_latitude;",
4      "ALTER TABLE hubs DROP COLUMN hub_longitude;",

5      "ALTER TABLE orders DROP COLUMN payment_order_id;",
6      "ALTER TABLE orders DROP COLUMN delivery_order_id;",
7      "ALTER TABLE orders DROP COLUMN order_created_hour;",
8      "ALTER TABLE orders DROP COLUMN order_created_minute;",
9      "ALTER TABLE orders DROP COLUMN order_created_day;",
10     "ALTER TABLE orders DROP COLUMN order_created_month;",
11     "ALTER TABLE orders DROP COLUMN order_created_year;",
12     "ALTER TABLE orders DROP COLUMN order_moment_accepted;",
13     "ALTER TABLE orders DROP COLUMN order_moment_ready;",
14     "ALTER TABLE orders DROP COLUMN order_moment_in_expedition;",
15     "ALTER TABLE orders DROP COLUMN order_moment_delivering;",
16     "ALTER TABLE orders DROP COLUMN order_moment_delivered;",
17     "ALTER TABLE orders DROP COLUMN order_metric_collected_time;",
18     "ALTER TABLE orders DROP COLUMN order_metric_paused_time;",
19     "ALTER TABLE orders DROP COLUMN order_metric_production_time;",
20     "ALTER TABLE orders DROP COLUMN order_metric_walking_time;",
21     "ALTER TABLE orders DROP COLUMN
22         order_metric_expedition_speed_time;",
23     "ALTER TABLE orders DROP COLUMN order_metric_transit_time;",
24     "ALTER TABLE orders DROP COLUMN order_metric_cycle_time;",

25
26     "ALTER TABLE payments DROP COLUMN payment_amount;",
27     "ALTER TABLE payments DROP COLUMN payment_fee;",
28     "ALTER TABLE stores DROP COLUMN store_plan_price;"

29 ]
```

```

30
31 # Thuc thi tung cau lenh SQL
32 for query in drop_columns_queries:
33     try:
34         cursor.execute(query)
35     except mysql.connector.Error as err:
36         print(f"Da gap loi khi thuc thi: {query}")
37         print(err)

```

Dây là một tập hợp các câu lệnh SQL được sử dụng để xóa các cột từ các bảng trong cơ sở dữ liệu MySQL. Mỗi câu lệnh ALTER TABLE ... DROP COLUMN ... được dùng để loại bỏ các cột không cần thiết từ các bảng hubs, orders, payments, và stores. Đoạn mã sau đó lặp qua từng câu lệnh trong danh sách drop_columns_queries và thực thi chúng bằng cách sử dụng cursor.execute(query). Trong trường hợp có lỗi trong quá trình thực thi, lỗi sẽ được bắt và in ra để thông báo về câu lệnh SQL gây ra vấn đề và thông tin chi tiết về lỗi. Điều này giúp quản trị cơ sở dữ liệu có thể dễ dàng điều chỉnh cấu trúc bảng bằng cách loại bỏ các cột không cần thiết một cách an toàn và hiệu quả.

Và dưới đây là kết quả sau khi đoạn lệnh này được thực thi:

After

Table: hubs

Columns:

- hub_id
- hub_name
- hub_city
- hub_state

Table: orders

Columns:

- order_id
- store_id
- channel_id
- order_status
- order_amount
- order_delivery_fee
- order_delivery_cost
- order_moment_created
- order_moment_collected
- order_moment_finished

 Remove Columns

Table: payments

Columns:

- payment_id
- payment_order_id
- payment_method
- payment_status

Remove Duplicates

Before

order_id	store_id	channel_id	order_status	order_amount	order_delivery_fee	order_delivery_cost	order_moment_created	order_moment_collected	order_moment_finished
68405206	3512	5	CANCELED	115.50	83.00	20.00	2021-01-01 00:13:07	NULL	2021-01-01 00:18:07
68412123	674	5	CANCELED	74.50	6.00	20.00	2021-01-01 13:57:31	NULL	2021-01-01 14:00:31
68412123	674	5	CANCELED	74.50	6.00	20.00	2021-01-01 13:57:31	NULL	2021-01-01 14:00:32
68412124	674	5	CANCELED	74.50	6.00	20.00	2021-01-01 13:57:31	NULL	2021-01-01 14:00:33
68412123	674	5	CANCELED	74.50	6.00	20.00	2021-01-01 13:57:31	NULL	2021-01-01 14:00:33
68412124	674	5	CANCELED	74.50	6.00	20.00	2021-01-01 13:57:31	NULL	2021-01-01 14:00:34
68405206	3512	5	CANCELED	115.50	83.00	20.00	2021-01-01 00:13:07	NULL	2021-01-01 00:18:08
68405206	3512	5	CANCELED	115.50	83.00	20.00	2021-01-01 00:13:07	NULL	2021-01-01 00:18:09

After

The screenshot shows a MySQL Workbench interface. In the SQL editor, there is a query to identify rows with duplicate order_ids:

```

1 • use data_python;
2 • SELECT * FROM orders
3 • WHERE order_id IN (
4     SELECT order_id
5         FROM orders
6         GROUP BY order_id
7         HAVING COUNT(*) > 1
8 );

```

The Result Grid shows the original data with 10 rows. Below it, a new Result Grid shows the data after duplicates have been removed, resulting in 8 unique rows.

```

1 for table in tables:
2     table_name = table[0]
3
4     # Lay ten cua cot dau tien trong bang
5     cursor.execute(f"SHOW COLUMNS FROM {table_name};")
6     # Doc ket qua tra ve
7     columns = cursor.fetchall()
8     # Lay ten cot dau tien
9     first_column = columns[0][0]
10
11    # Them mot cot moi
12    add_column_query = f """
13        ALTER TABLE {table_name}
14        ADD COLUMN {first_column}_sequence_number INT;
15        """
16
17    cursor.execute(add_column_query)

```

Đoạn mã trên được sử dụng để thêm một cột mới vào mỗi bảng trong danh sách tables của cơ sở dữ liệu MySQL. Đầu tiên, vòng lặp for duyệt qua từng bảng trong danh sách tables. Sau đó, với mỗi bảng, câu lệnh SHOW COLUMNS FROM {table_name}; được sử dụng để lấy danh sách các cột hiện tại của bảng từ cơ sở dữ liệu. Kết quả này được lấy bằng phương thức cursor.fetchall() và lưu vào biến columns.

Biến first_column lưu trữ tên của cột đầu tiên trong danh sách các cột của bảng. Cột đầu tiên thường được sử dụng để làm mẫu cho việc thêm cột mới, do đó tên của nó được sử dụng để đặt tên cho cột mới sẽ được thêm vào, với tên là {first_column}_sequence_number và kiểu dữ liệu là INT.

Câu lệnh ALTER TABLE ... ADD COLUMN ... được tạo ra dưới dạng chuỗi trong biến add_column_query, với tên bảng và tên cột mới được thay thế bằng giá trị thích hợp từ vòng lặp. Cuối cùng, câu lệnh này được thực thi bằng cursor.execute(add_column_query), từ đó thêm cột mới vào bảng trong cơ sở dữ liệu MySQL.

```

1   # Thiết lập biến chuỗi và cập nhật cột moi
2 set_sequence_query = """
3 SET @seq = 0;
4 """
5 update_sequence_query = f"""
6 UPDATE {table_name}
7 SET {first_column}_sequence_number = (@seq := @seq + 1)
8 ORDER BY {first_column};
9 """
10 cursor.execute(set_sequence_query)
11 cursor.execute(update_sequence_query)
12
13 # Xóa các dòng trùng lắp
14 delete_duplicates_query = f"""
15 DELETE
16 FROM {table_name}
17 WHERE {first_column}_sequence_number IN (
18     SELECT {first_column}_sequence_number
19     FROM (
20         SELECT {first_column}_sequence_number,
21             ROW_NUMBER() OVER (
22                 PARTITION BY {first_column}
23                 ORDER BY {first_column}
24             ) AS row_num
25     FROM {table_name}
26     ) t
27     WHERE row_num > 1
28 );
29 """
30 cursor.execute(delete_duplicates_query)

```

Đoạn mã trên thực hiện hai công việc chính trên một bảng trong cơ sở dữ liệu MySQL. Đầu tiên, nó thiết lập biến `@seq` bằng 0 để sử dụng cho việc đánh số thứ tự cho các hàng. Tiếp theo, nó cập nhật cột `{first_column}_sequence_number` trong bảng `{table_name}` bằng cách tăng dần biến `@seq` cho mỗi hàng và sắp xếp các hàng theo giá trị của cột `{first_column}`. Cuối cùng, nó xóa các dòng trùng lắp bằng cách chỉ giữ lại dòng đầu tiên trong mỗi nhóm các giá trị `{first_column}` và xóa các dòng còn lại. Đoạn mã này được sử dụng để cập nhật số thứ tự cho các dòng dữ liệu và loại bỏ các bản ghi trùng lắp dựa trên một cột nhất định trong cơ sở dữ liệu MySQL.

```

1 # Xóa cột moi da them
2 drop_column_query = f"""
3 ALTER TABLE {table_name}
4 DROP COLUMN {first_column}_sequence_number;
5 """
6 cursor.execute(drop_column_query)
7
8 connection.commit()
9 print("Da xoa cot moi da them!")

```

Đoạn mã trên được sử dụng để xóa cột `{first_column}_sequence_number` đã được thêm vào bảng `{table_name}` trong cơ sở dữ liệu MySQL. Đầu tiên, nó tạo ra một câu lệnh SQL để thực hiện việc xóa cột, sau đó thực thi câu lệnh này bằng phương thức `cursor.execute(drop_column_query)`.

Sau khi thực hiện xong, lệnh `connection.commit()` được sử dụng để lưu các thay đổi vào cơ sở dữ liệu. Cuối cùng, nó in ra thông báo "Đã xóa cột mới đã thêm!" để xác nhận rằng quá trình xóa cột đã được hoàn thành thành công.

Xử lý Nulls

delivery_id	delivery_order_id	driver_id	delivery_distance_meters	delivery_status
2565087	68405119	HULL	906.0	CANCELLED
2565098	68405123	HULL	867.0	CANCELLED
2565099	68405206	HULL	614.0	CANCELLED
2565100	68405465	HULL	471.0	CANCELLED
2565101	68406064	HULL	512.0	CANCELLED
2565102	68408108	HULL	145.0	CANCELLED
2565103	68408109	HULL	188.0	CANCELLED
2565104	68409030	HULL	509.0	CANCELLED
2565105	68410055	HULL	983.0	CANCELLED
2565106	68412121	HULL	383.0	CANCELLED
2565107	68412122	HULL	966.0	CANCELLED
2565108	68412123	HULL	584.0	CANCELLED
2565109	68412131	HULL	522.0	CANCELLED
2565110	68412134	HULL	556.0	CANCELLED
2565111	68412148	HULL	213.0	CANCELLED

order_id	store_id	channel_id	order_status	order_amount	order_delivery_fee	order_delivery_cost	order_moment_created	order_moment_collected	order_moment_finished
68405119	3512	5	CANCELED	62.70	130.00	20.00	2021-01-01 00:01:36	HULL	2021-01-01 00:06:36
68405123	3512	5	CANCELED	62.70	178.00	20.00	2021-01-01 00:04:26	HULL	2021-01-01 00:09:26
68405206	3512	5	CANCELED	115.30	83.00	20.00	2021-01-01 00:13:07	HULL	2021-01-01 00:18:07
68405465	3401	5	CANCELED	55.90	177.00	20.00	2021-01-01 00:19:15	HULL	2021-01-01 00:24:15
68406064	3401	5	CANCELED	37.90	201.00	20.00	2021-01-01 00:26:23	HULL	2021-01-01 00:29:25
68408108	786	5	CANCELED	80.00	109.00	20.00	2021-01-01 00:56:19	HULL	2021-01-01 00:57:19
68408109	1125	5	CANCELED	71.00	182.00	20.00	2021-01-01 00:56:49	HULL	2021-01-01 00:57:49
68412121	1152	5	CANCELED	25.50	166.00	20.00	2021-01-01 10:57:24	HULL	2021-01-01 10:59:21
68412122	490	5	CANCELED	30.00	99.00	20.00	2021-01-01 12:57:21	HULL	2021-01-01 13:58:21
68412123	674	5	CANCELED	74.50	6.00	20.00	2021-01-01 12:57:31	HULL	2021-01-01 14:00:31
68412131	490	5	CANCELED	48.00	57.00	20.00	2021-01-01 13:58:01	HULL	2021-01-01 14:01:01
68412134	679	5	CANCELED	34.90	10.00	20.00	2021-01-01 13:58:32	HULL	2021-01-01 14:00:32
68412148	294	5	CANCELED	34.00	220.00	20.00	2021-01-01 14:02:01	HULL	2021-01-01 14:06:01
68412322	294	5	CANCELED	39.30	131.00	20.00	2021-01-01 14:06:31	HULL	2021-01-01 14:10:31
68412444	294	5	CANCELED	41.50	132.00	20.00	2021-01-01 14:08:11	HULL	2021-01-01 14:12:11

```

1   def replace_null_values(connection, tables_to_replace):
2       cursor = connection.cursor()
3
4       for table in tables_to_replace:
5           cursor.execute(
6               f"SELECT COLUMN_NAME, DATA_TYPE FROM INFORMATION_SCHEMA
7                   .COLUMNS WHERE TABLE_NAME =
8                       '{table}' AND TABLE_SCHEMA = 'data_python'")
9           columns = cursor.fetchall()
10
11           for column in columns:
12               column_name = column[0]
13               data_type = column[1]
14
15               update_value = ""
16               if data_type.startswith("int") or data_type.startswith(
17                   "float") or data_type.startswith("decimal"):
18                   update_value = "-1"
19               elif data_type.startswith("date"):
20                   update_value = "'1900-01-01'"
21               elif data_type.startswith("datetime"):
22                   update_value = "'1900-01-01 00:00:00'"
23
24               if update_value:
25                   update_query = f"""
26                         UPDATE `{table}`
```

```
25         SET `'{column_name}` = {update_value}
26         WHERE `'{column_name}` IS NULL
27         """
28     cursor.execute(update_query)
29     connection.commit()
30     print(
31         f"Da thay the gia tri NULL trong cot `'{column_name}` cua bang `'{table}` bang gia tri
32             {update_value}")
33
34 cursor.close()
```

Đoạn mã trên là một hàm Python có tên `replace_null_values`, được thiết kế để thay thế các giá trị NULL trong các cột của các bảng trong cơ sở dữ liệu MySQL. Hàm này nhận hai đối số: `connection`, là đối tượng kết nối đến cơ sở dữ liệu MySQL, và `tables_to_replace`, là danh sách các bảng cần thực hiện thay thế giá trị NULL.

Dầu tiên, trong vòng lặp `for table in tables_to_replace`, hàm thực hiện truy vấn SQL để lấy tên cột và kiểu dữ liệu của từng cột trong bảng hiện tại từ `INFORMATION_SCHEMA.COLUMNS`. Truy vấn này chỉ lấy các cột trong cơ sở dữ liệu `data_python`.

Tiếp theo, với mỗi cột, hàm kiểm tra kiểu dữ liệu của cột để xác định giá trị mặc định nào sẽ được sử dụng để thay thế các giá trị NULL. Nếu cột có kiểu dữ liệu là int, float, hoặc decimal, giá trị thay thế sẽ là '-1'. Đối với các cột có kiểu dữ liệu là date, giá trị sẽ là '1900-01-01', và đối với các cột có kiểu dữ liệu là datetime, giá trị sẽ là '1900-01-01 00:00:00'.

Sau khi xác định được giá trị thay thế, hàm sẽ thực hiện câu lệnh SQL UPDATE để cập nhật các giá trị NULL trong cột tương ứng của bảng đó thành giá trị mới. Việc cập nhật được thực hiện với điều kiện là cột đó có giá trị NULL (`WHERE {column name} IS NULL`).

Mỗi khi thực hiện câu lệnh UPDATE, hàm gọi `connection.commit()` để lưu các thay đổi vào cơ sở dữ liệu. Cuối cùng, hàm in ra một thông báo xác nhận cho biết rằng đã thực hiện thay thế giá trị NULL thành công trong cột cu_thể của bảng tương ứng.

```
1 def delete_null_values(connection, tables_to_delete):
2     cursor = connection.cursor()
3
4     for table in tables_to_delete:
5         cursor.execute(
6             f"SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS
7                 WHERE TABLE_NAME "
8             f"= '{table}' AND TABLE_SCHEMA = 'data_python'")
9         columns = cursor.fetchall()
10        columns = [column[0] for column in columns]
11
12        conditions = " OR ".join([f"`{column}` IS NULL" for column
13            in columns])
14        if conditions:
15            delete_query = f"DELETE FROM `{table}` WHERE {
16                conditions}"
17            cursor.execute(delete_query)
18            connection.commit()
19            print(f"Da xoa cac dong co gia tri NULL trong bang `{table}`")
```

18 cursor.close()

Đoạn code trên định nghĩa hàm `delete_null_values` trong Python, được sử dụng để xóa các dòng có giá trị NULL từ các bảng trong cơ sở dữ liệu MySQL. Hàm này nhận hai đối số: `connection`, là đối tượng kết nối đến cơ sở dữ liệu MySQL, và `tables_to_delete`, là một danh sách các tên bảng cần xử lý.

Dầu tiên, với mỗi bảng trong `tables_to_delete`, hàm truy vấn cơ sở dữ liệu để lấy danh sách tên các cột trong bảng đó từ `INFORMATION_SCHEMA.COLUMNS`. Sau đó, hàm xây dựng một chuỗi điều kiện để xóa các dòng có giá trị NULL từng cột bằng cách sử dụng điều kiện '`column`' `IS NULL`. Nếu có dòng nào thỏa điều kiện này, hàm thực thi câu lệnh `DELETE` tương ứng và gọi `connection.commit()` để lưu thay đổi vào cơ sở dữ liệu.

Cuối cùng, hàm in ra thông báo xác nhận cho biết đã xóa các dòng có giá trị NULL thành công từ bảng cụ thể. Hàm đóng con trỏ `cursor` sau khi hoàn thành công việc để giải phóng tài nguyên.

The screenshot shows two separate MySQL queries in the SQL editor:

Deliveries:

```

1 *  SELECT * FROM deliveries
2 WHERE delivery_id IS NULL
3     OR delivery_order_id IS NULL
4     OR driver_id IS NULL
5     OR delivery_distance_meters IS NULL
6     OR delivery_status IS NULL;
7

```

Orders:

```

1 *  SELECT * FROM orders
2 WHERE order_id IS NULL
3     OR store_id IS NULL
4     OR channel_id IS NULL
5     OR order_status IS NULL
6     OR order_amount IS NULL
7     OR order_delivery_fee IS NULL
8     OR order_delivery_cost IS NULL
9     OR order_moment_created IS NULL
10    OR order_moment_collected IS NULL
11    OR order_moment_finished IS NULL;

```

Both queries use the same WHERE clause structure to filter for rows where all columns are NULL. The results are shown in the Result Grid below each query.

Data Binning

delivery_id	delivery_order_id	driver_id	delivery_distance_meters	delivery_status
2174659	68413340	8378	5199.0	DELIVERED
2174660	68414309	2473	410.0	DELIVERED
2174661	68416230	7615	3784.0	DELIVERED
2174662	68412721	8378	5714.0	DELIVERED
2174663	68414018	10463	3746.0	DELIVERED
2174664	68415103	16430	3924.0	DELIVERED
2174665	68416643	14513	2489.0	DELIVERED
2174667	68415457	9996	340.0	DELIVERED
2174668	68414563	23092	1081.0	DELIVERED
2174669	68415140	9996	2880.0	DELIVERED
2174670	68416059	9996	1450.0	DELIVERED

Deliveries



Before

Orders

order_id	store_id	channel_id	order_status	order_amount	order_delivery_fee	order_delivery_cost	order_moment_created	order_moment_collected	order_moment_finished
68405119	3512	5	CANCELED	62.70	130.00	20.00	2021-01-01 00:01:36	1900-01-01 00:00:00	2021-01-01 00:06:36
68405123	3512	5	CANCELED	62.70	178.00	20.00	2021-01-01 00:04:26	1900-01-01 00:00:00	2021-01-01 00:09:26
68405206	3512	5	CANCELED	115.50	83.00	20.00	2021-01-01 00:13:07	1900-01-01 00:00:00	2021-01-01 00:18:07
68405465	3401	5	CANCELED	55.90	177.00	20.00	2021-01-01 00:19:15	1900-01-01 00:00:00	2021-01-01 00:24:15
68406064	3401	5	CANCELED	37.90	201.00	20.00	2021-01-01 00:26:25	1900-01-01 00:00:00	2021-01-01 00:29:25
68408108	786	5	CANCELED	80.00	109.00	20.00	2021-01-01 00:56:19	1900-01-01 00:00:00	2021-01-01 00:57:19
68408109	1125	5	CANCELED	71.00	182.00	20.00	2021-01-01 00:56:49	1900-01-01 00:00:00	2021-01-01 00:57:49
68409030	1054	35	CANCELED	214.80	69.00	0.00	2021-01-01 01:56:42	2021-01-01 02:53:42	2021-01-01 06:08:42

```

1      delivery_range_sql_1 = """
2          ALTER TABLE deliveries
3              ADD COLUMN delivery_range VARCHAR(50);
4      """
5      cursor.execute(delivery_range_sql_1)
6
7      delivery_range_sql_2 = """
8          UPDATE deliveries
9              SET delivery_range = CASE
10                  WHEN delivery_distance_meters < 5000 THEN 'Small'
11                  WHEN delivery_distance_meters < 10000 THEN 'Medium'
12                  WHEN delivery_distance_meters < 15000 THEN 'Large'
13                  ELSE 'Very Large'
14          END;
15      """
16      cursor.execute(delivery_range_sql_2)
17
18      delivery_range_sql_3 = """
19          ALTER TABLE deliveries
20              MODIFY COLUMN delivery_range VARCHAR(50) AFTER
21                  delivery_distance_meters;
22      """
23      cursor.execute(delivery_range_sql_3)
print("Data binning for 'delivery_distance_meters' in "
      "deliveries' table done.")

```

Đoạn mã này thực hiện phân loại dữ liệu (data binning) cho cột `delivery_distance_meters` trong bảng `deliveries` của cơ sở dữ liệu. Đầu tiên, nó thêm một cột mới có tên là `delivery_range` với kiểu dữ liệu `VARCHAR(50)`. Sau đó, nó cập nhật giá trị của cột `delivery_range` dựa trên khoảng giá trị của cột `delivery_distance_meters`: 'Small' cho các giá trị dưới 5000 mét, 'Medium' cho các giá trị từ 5000 đến dưới 10000 mét, 'Large' cho các giá trị từ 10000 đến dưới 15000 mét, và 'Very Large' cho các giá trị lớn hơn hoặc bằng 15000 mét. Cuối cùng, nó điều chỉnh vị trí của cột `delivery_range` để đặt sau cột `delivery_distance_meters` và in ra thông báo xác nhận khi hoàn thành.

```

1     order_amount_sql_1 = """
2         ALTER TABLE orders
3             ADD COLUMN order_amount_value VARCHAR(50);
4         """
5     cursor.execute(order_amount_sql_1)
6
7     order_amount_sql_2 = """
8         UPDATE orders
9             SET order_amount_value = CASE
10                 WHEN order_amount < 100 THEN 'Low'
11                 WHEN order_amount >= 100 AND order_amount < 1000 THEN 'Medium'
12                 WHEN order_amount >= 1000 AND order_amount < 5000 THEN 'High'
13                 ELSE 'Very High'
14             END;
15         """
16     cursor.execute(order_amount_sql_2)
17
18     order_amount_sql_3 = """
19         ALTER TABLE orders
20             MODIFY COLUMN order_amount_value VARCHAR(50) AFTER
21                 order_amount;
22         """
23     cursor.execute(order_amount_sql_3)
24     print("Data binning for 'order_amount' in 'orders' table
done.")

```

Đoạn mã trên được sử dụng để thực hiện phân loại dữ liệu (data binning) cho cột `order_amount` trong bảng `orders` của cơ sở dữ liệu. Đầu tiên, nó thêm một cột mới có tên là `order_amount_value` với kiểu dữ liệu `VARCHAR(50)`. Tiếp theo, thông qua câu lệnh `UPDATE`, nó cập nhật giá trị của cột `order_amount_value` dựa trên giá trị của cột `order_amount` như sau: 'Low' cho các giá trị dưới 100, 'Medium' cho các giá trị từ 100 đến dưới 1000, 'High' cho các giá trị từ 1000 đến dưới 5000, và 'Very High' cho các giá trị lớn hơn hoặc bằng 5000. Cuối cùng, nó điều chỉnh vị trí của cột `order_amount_value` để đặt sau cột `order_amount` và in ra thông báo xác nhận khi hoàn thành quá trình phân loại dữ liệu.

delivery_id	delivery_order_id	driver_id	delivery_distance_meters	delivery_range	delivery_status
2174659	68413340	8378	5199.0	Medium	DELIVERED
2174660	68414309	2473	410.0	Small	DELIVERED
2174661	68416230	7615	3784.0	Small	DELIVERED
2174662	68412721	8378	5714.0	Medium	DELIVERED
2174663	68414018	10463	3746.0	Small	DELIVERED
2174664	68415103	16430	3924.0	Small	DELIVERED
2174665	68416643	14513	2489.0	Small	DELIVERED
2174667	68415457	9996	340.0	Small	DELIVERED
2174668	68414563	23092	1081.0	Small	DELIVERED
2174669	68415140	9996	2880.0	Small	DELIVERED
2174670	68416059	9996	1450.0	Small	DELIVERED



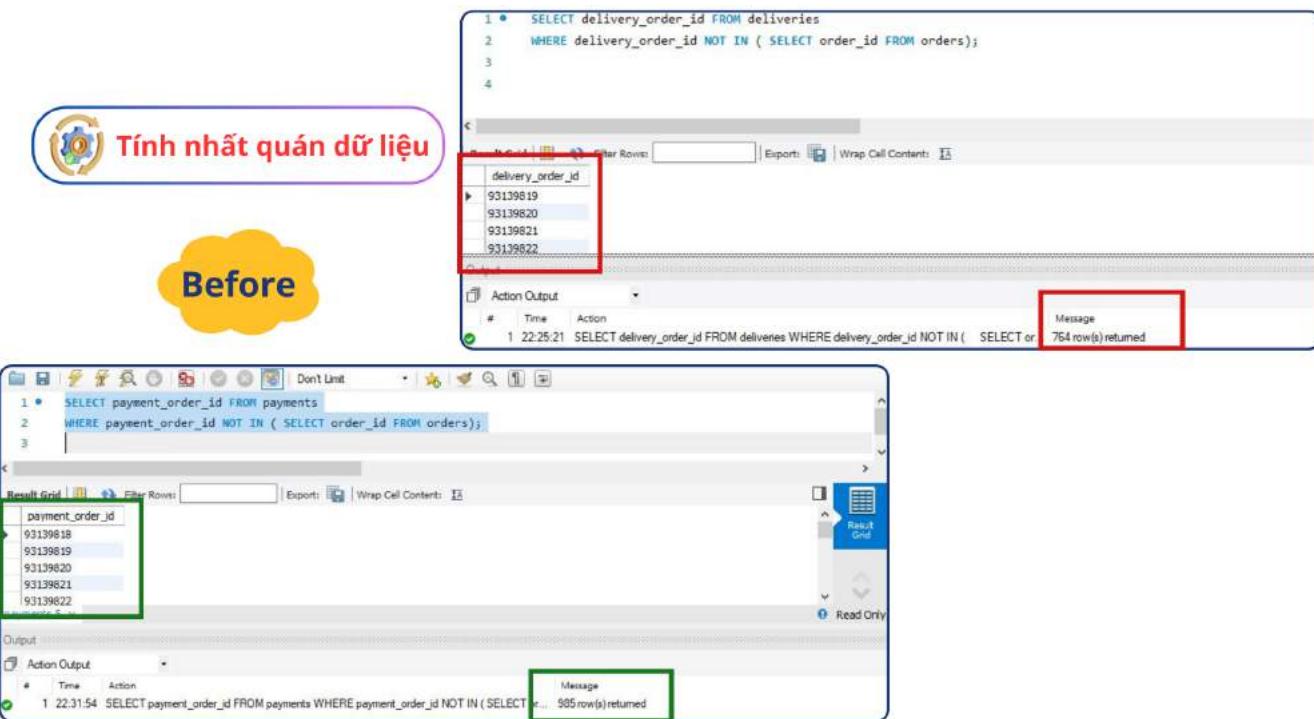
Deliveries

After

Orders

order_id	store_id	channel_id	order_status	order_amount	order_amount_value	order_delivery_fee	order_delivery_cost	order_moment_created	order_moment_collected	order_moment_finished
68405119	3512	5	CANCELED	62.70	Low	130.00	20.00	2021-01-01 00:01:36	1900-01-01 00:00:00	2021-01-01 00:06:36
68405123	3512	5	CANCELED	62.70	Low	178.00	20.00	2021-01-01 00:04:26	1900-01-01 00:00:00	2021-01-01 00:09:26
68405206	3512	5	CANCELED	115.50	Medium	33.00	20.00	2021-01-01 00:13:07	1900-01-01 00:00:00	2021-01-01 00:18:07
68405465	3401	5	CANCELED	55.90	Low	177.00	20.00	2021-01-01 00:19:15	1900-01-01 00:00:00	2021-01-01 00:24:15
68406064	3401	5	CANCELED	37.90	Low	201.00	20.00	2021-01-01 00:26:25	1900-01-01 00:00:00	2021-01-01 00:29:25
68408108	786	5	CANCELED	80.00	Low	109.00	20.00	2021-01-01 00:56:19	1900-01-01 00:00:00	2021-01-01 00:57:19
68408109	1125	5	CANCELED	71.00	Low	182.00	20.00	2021-01-01 00:56:49	1900-01-01 00:00:00	2021-01-01 00:57:49
68412121	1152	5	CANCELED	25.50	Low	166.00	20.00	2021-01-01 13:57:21	1900-01-01 00:00:00	2021-01-01 13:59:21
68412122	490	5	CANCELED	30.00	Low	99.00	20.00	2021-01-01 13:57:21	1900-01-01 00:00:00	2021-01-01 13:58:21

Tính nhất quán dữ liệu



```
1 def delete_deliveries_excess_data():
2     cnx = connect_to_database()
3     cursor = cnx.cursor()
4
5     delete_query = """
6         DELETE FROM deliveries
7         WHERE delivery_order_id NOT IN (SELECT order_id FROM orders
8             )
9     """
10    cursor.execute(delete_query)
11    cnx.commit()
12
13    cursor.close()
14    cnx.close()
15
16 def delete_payments_excess_data():
17     cnx = connect_to_database()
18     cursor = cnx.cursor()
19
20     delete_query = """
21         DELETE FROM payments
22         WHERE payment_order_id NOT IN (SELECT order_id FROM orders)
23     """
24    cursor.execute(delete_query)
25    cnx.commit()
26
27    cursor.close()
28    cnx.close()
```

Đoạn code trên định nghĩa hai hàm `delete_deliveries_excess_data()` và

`delete_payments_excess_data()` để xóa dữ liệu dư thừa từ các bảng `deliveries` và `payments` trong cơ sở dữ liệu. Mỗi hàm sử dụng một câu lệnh `DELETE` để loại bỏ các hàng trong bảng tương ứng (`deliveries` và `payments`) nơi mà giá trị của cột `delivery_order_id` và `payment_order_id` không có trong cột `order_id` của bảng `orders`.

Cụ thể, hàm `delete_deliveries_excess_data()` xử lý việc xóa các bản ghi trong bảng `deliveries` mà không có `order_id` tương ứng trong bảng `orders`. Tương tự, hàm `delete_payments_excess_data()` xóa các bản ghi trong bảng `payments` mà không có `order_id` tương ứng trong bảng `orders`.

Sau khi thực hiện câu lệnh `DELETE`, mỗi hàm gọi phương thức `commit()` để lưu các thay đổi vào cơ sở dữ liệu. Cuối cùng, các đối tượng `cursor` và `cnx` (kết nối đến cơ sở dữ liệu) được đóng lại để giải phóng tài nguyên.



Tính nhất quán dữ liệu

After

```

1 •   SELECT delivery_order_id FROM deliveries
2     WHERE delivery_order_id NOT IN ( SELECT order_id FROM orders);
3
4

```

Result Grid | Filter Rows: [] Export: [] Wrap Cell Content: []

delivery_order_id

Output

Action Output		
#	Time	Action
1	22:35:27	SELECT delivery_order_id FROM deliveries WHERE delivery_order_id NOT IN (SELECT order_id FROM orders);

Message: 0 row(s) returned

```

1 •   SELECT payment_order_id FROM payments
2     WHERE payment_order_id NOT IN ( SELECT order_id FROM orders);
3
4

```

Result Grid | Filter Rows: [] Export: [] Wrap Cell Content: []

payment_order_id

Output

Action Output		
#	Time	Action
2	22:36:52	SELECT payment_order_id FROM payments WHERE payment_order_id NOT IN (SELECT order_id FROM orders);

Message: 0 row(s) returned

Thu thập kinh độ vĩ độ và dữ liệu địa lý từ API



Before



Phân cụm kinh vĩ độ và thu thập dữ liệu địa lý từ API

store_id	hub_id	store_name	store_segment	store_latitude	store_longitude
3	2	CUMIURI	FOOD	-30.0474148	-51.21351
6	3	PIMGUCIS DA VIVA	FOOD	-30.0374149	-51.20352
8	3	RASMUR S	FOOD	-30.0374149	-51.20352
53	8	PAPA SUCIS	FOOD	-22.921475	-43.234822
54	8	VUZPI PAZZIS	FOOD	-22.921475	-43.234822
56	8	SUPSIDO	FOOD	-22.921475	-43.234822
58	8	PIAMUARIS	FOOD	-22.921475	-43.234822
82	8	LUCITA	FOOD	-22.921475	-43.234822
83	8	PRARIZZAI	FOOD	-22.921475	-43.234822
84	8	PALLO MZU GRALA	FOOD	-22.921475	-43.234822

Bảng stores có các thông tin về kinh độ, vĩ độ như trong hình vẽ, ta sẽ tiến hành chuyển đổi tọa độ của mỗi store thành địa điểm cụ thể để thuận tiện cho quá trình phân tích.

Để làm được điều này, đầu tiên ta sẽ thực hiện phân cụm các tọa độ của các store để từ đó tìm ra tọa độ trung tâm của mỗi cụm bằng thuật toán phân cụm K-Means, rồi từ tọa độ trung tâm của mỗi cụm, ta sử dụng thư viện Nominatim trong Python để thu thập dữ liệu địa lý từ API, từ đó ta có thể xác định được địa điểm cụ thể dựa trên tọa độ trung tâm của mỗi cụm

Thuật toán phân cụm K-means có các bước chính như sau:

- Chọn số lượng cụm k mà ta muốn tạo.
- Khởi tạo các điểm ngẫu nhiên làm các điểm trung tâm ban đầu cho các cụm.
- Lặp lại các bước sau cho đến khi không có sự thay đổi đột phá nào:
 - Gán từng điểm dữ liệu vào cụm gần nhất bằng cách tính khoảng cách Euclid từ mỗi điểm đến các điểm trung tâm của các cụm.
 - Cập nhật lại vị trí của các điểm trung tâm của các cụm bằng cách lấy trung bình của tất cả các điểm dữ liệu thuộc cụm đó.
- Kết thúc khi không có sự thay đổi đáng kể nào trong việc gán các điểm vào các cụm hoặc khi đạt được số lần lặp tối đa đã định trước.

Để có thể đánh giá được hiệu quả của thuật toán phân cụm cũng như xác định được nên phân bao nhiêu cụm, ta thường sử dụng chỉ số Silhouette và phương pháp Elbow. Cụ thể:

- Phương pháp Elbow được sử dụng để xác định số lượng cụm tối ưu trong thuật toán phân cụm K-means. Quá trình đánh giá số lượng cụm bao gồm các bước sau:

- Thực hiện thuật toán K-means với các giá trị khác nhau của số lượng cụm k .
- Đo lường và tính toán tổng bình phương khoảng cách từ các điểm dữ liệu đến các trung tâm cụm (SSE - Sum of Squared Errors). Tổng bình phương sai số (SSE) đo lường tổng

của các bình phương khoảng cách giữa mỗi điểm dữ liệu và trọng tâm của cụm mà nó thuộc về. Công thức tính SSE như sau:

$$SSE = \sum_{i=1}^n \sum_{x \in C_i} \|x - \mu_i\|^2$$

trong đó:

- n là số lượng điểm dữ liệu,
- C_i đại diện cho cụm mà điểm i thuộc về,
- μ_i là trọng tâm (trung bình) của cụm C_i ,
- $\|x - \mu_i\|^2$ là bình phương khoảng cách Euclide giữa điểm dữ liệu x và trọng tâm μ_i .
- Vẽ biểu đồ SSE theo số lượng cụm k .
- Chọn số lượng cụm là nơi mà đường cong SSE bắt đầu giảm chậm (tạo hình dạng như "khuỷu tay" - elbow), cho rằng đây là số lượng cụm tối ưu.
- Chỉ số Silhouette là một phương pháp đo lường chất lượng phân cụm dựa trên khoảng cách giữa các điểm dữ liệu trong cùng một cụm và giữa các cụm khác nhau. Quá trình đánh giá sử dụng chỉ số Silhouette bao gồm:
 - Tính toán chỉ số Silhouette cho từng điểm dữ liệu trong tập dữ liệu.
 - Chỉ số Silhouette cho một điểm dữ liệu i được tính bằng công thức sau:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

trong đó:

- $s(i)$ là chỉ số Silhouette cho điểm dữ liệu i ,
- $a(i)$ là khoảng cách trung bình giữa điểm i và tất cả các điểm trong cùng cụm (khoảng cách nội-cụm),
- $b(i)$ là khoảng cách trung bình giữa điểm i và tất cả các điểm trong cụm gần nhất (khoảng cách liên-cụm).
- Chỉ số Silhouette của từng điểm dữ liệu nằm trong khoảng $[-1, 1]$, với giá trị càng gần 1 cho thấy điểm dữ liệu đó được phân cụm tốt, còn giá trị càng gần -1 cho thấy điểm dữ liệu nên được phân cụm vào cụm khác hơn.

```

1 # Lay du lieu tu bang stores
2 query = "SELECT store_id, store_latitude, store_longitude FROM
3 stores"
4 cursor = db_connection.cursor()
5 cursor.execute(query)
6 data = cursor.fetchall()
7
8 # Chuyen du lieu thanh numpy array
9 coordinates = np.array([(row[1], row[2]) for row in data])
10 store_ids = [row[0] for row in data]
11
12 # Tim so luong cum toi uu su dung phuong phap Elbow va Silhouette
13 wcss = []
14 silhouette_scores = []

```

```

15 K_range = range(2, 10)
16 for k in K_range:
17     kmeans = KMeans(n_clusters=k, random_state=42)
18     kmeans.fit(coordinates)
19     wcss.append(kmeans.inertia_)
20     silhouette_scores.append(silhouette_score(coordinates, kmeans.
21         labels_))

22 # Ve bieu do Elbow va Silhouette
23 plt.figure(figsize=(12, 5))

24
25 plt.subplot(1, 2, 1)
26 plt.plot(K_range, wcss, marker='o')
27 plt.xlabel('So luong cum (K)')
28 plt.ylabel('Tong binh phuong khoang cach')
29 plt.title('Phuong phap Elbow')

30
31 plt.subplot(1, 2, 2)
32 plt.plot(K_range, silhouette_scores, marker='o')
33 plt.xlabel('So luong cum (K)')
34 plt.ylabel('Chi so Silhouette')
35 plt.title('Chi so Silhouette')

36
37 plt.show()

38
39 # Chon so cum toi uu
40 optimal_k = K_range[np.argmax(silhouette_scores)]
41 print(f'So cum toi uu dua tren chi so Silhouette: {optimal_k}')

42
43 # Ap dung K-means clustering voi so cum toi uu
44 kmeans = KMeans(n_clusters=optimal_k, random_state=42)
45 kmeans.fit(coordinates)

46
47 # Lay ket qua phan cum va toa do trung tam
48 labels = kmeans.labels_
49 centroids = kmeans.cluster_centers_

```

Đoạn code trên thực hiện các bước để phân cụm các cửa hàng dựa trên tọa độ địa lý (latitude và longitude) của từng cửa hàng từ một bảng dữ liệu trong cơ sở dữ liệu. Đầu tiên, nó thực hiện truy vấn để lấy thông tin về mã cửa hàng, vĩ độ và kinh độ của từng cửa hàng từ bảng "stores". Dữ liệu sau đó được chuyển đổi thành một mảng numpy, trong đó mỗi phần tử là một cặp tọa độ (latitude, longitude) của từng cửa hàng.

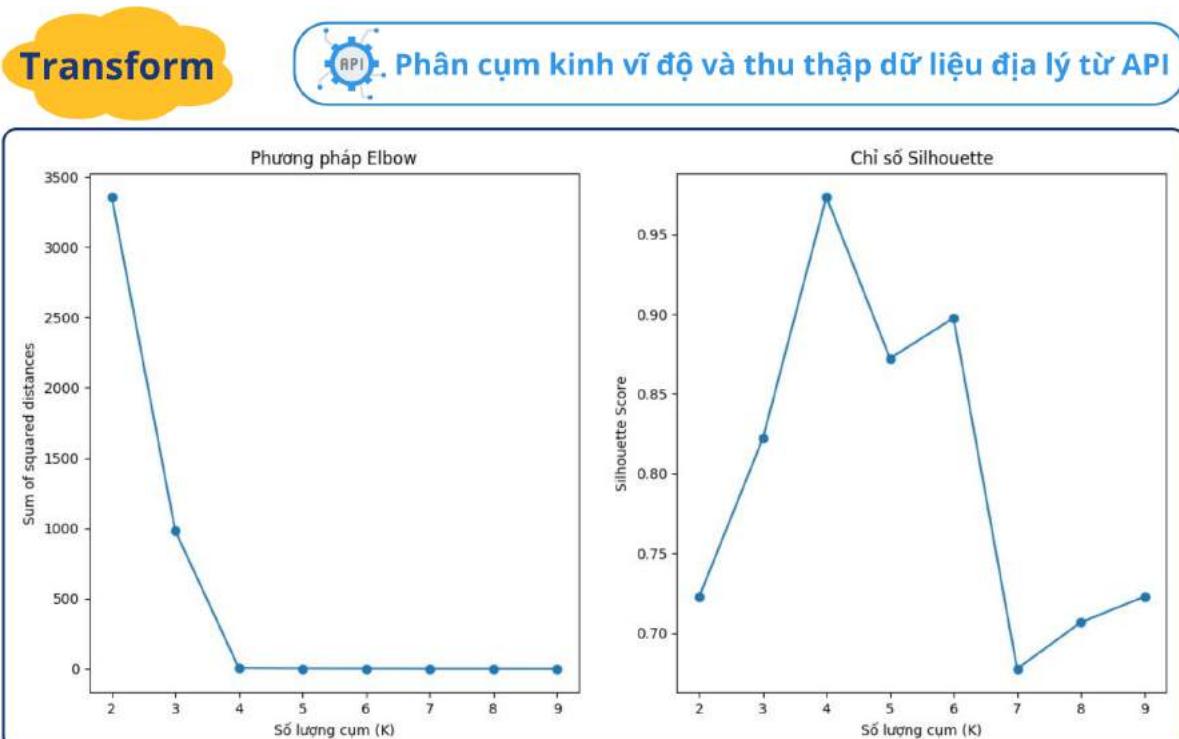
Tiếp theo, đoạn mã sử dụng phương pháp Elbow và chỉ số Silhouette để xác định số lượng cụm tối ưu. Phương pháp Elbow được áp dụng để tìm ra giá trị của K mà ở đó tổng bình phương khoảng cách từ các điểm dữ liệu đến trung tâm cụm gần nhất không tăng đáng kể nữa. Chỉ số Silhouette đo độ tương đồng của các điểm dữ liệu trong cùng một cụm và độ khác biệt giữa các cụm khác nhau.

Sau khi thu được các giá trị của Elbow và Silhouette cho các giá trị K từ 2 đến 9, đoạn mã vẽ biểu đồ để hỗ trợ việc lựa chọn số lượng cụm tối ưu. Bằng cách so sánh các giá trị này, nó xác định số lượng cụm tối ưu dựa trên chỉ số Silhouette và lưu giữ kết quả này vào biến optimal_k.

Cuối cùng, K-means clustering được áp dụng với số lượng cụm tối ưu để phân loại các cửa

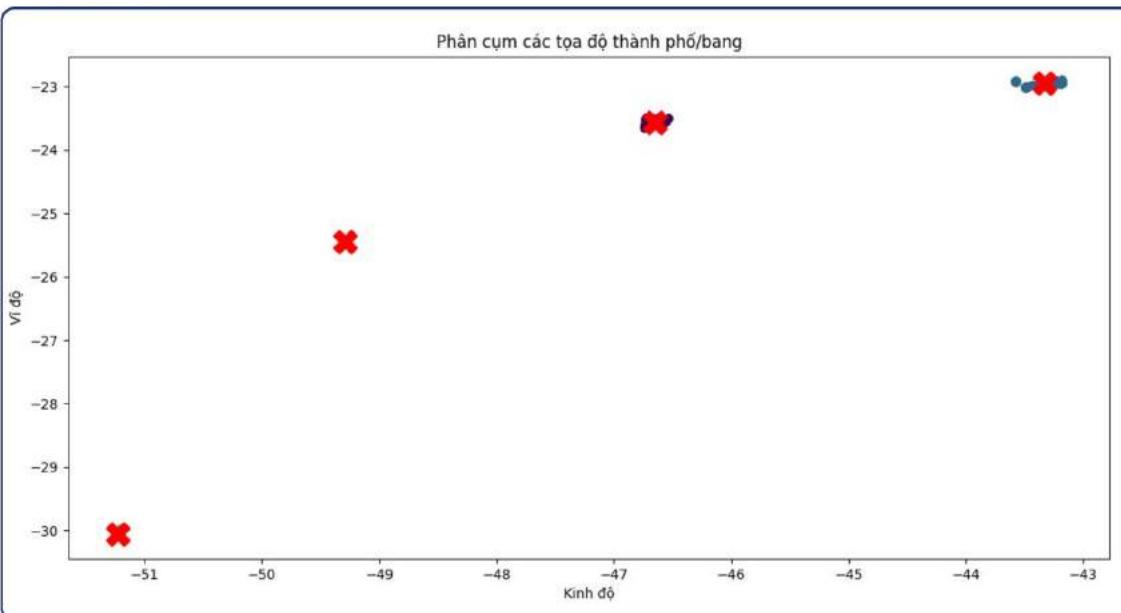
hàng vào từng nhóm. Kết quả cuối cùng bao gồm nhãn của từng cụm mà mỗi cửa hàng thuộc về và tọa độ của các trung tâm cụm, được lưu vào biến labels và centroids tương ứng. Và đây là 2 biểu đồ kết quả của phương pháp Elbow và chỉ số Silhouette, ta có thể thấy rằng với số lượng cụm là $k = 4$ thì chỉ số SSE là thấp và chỉ số Silhouette cao nhất (> 0.95), do đó thuật toán phân cụm đạt hiệu quả cao nhất tại $k = 4$. Ta sẽ phân tọa độ của các cửa hàng vào 4 cụm.

Và dưới đây là kết quả sau khi phân cụm:



Transform

Phân cụm kinh vĩ độ và thu thập dữ liệu địa lý từ API



```

1 from geopy.geocoders import Nominatim
2

```

```

3 # Su dung geopy de xac dinh bang va thanh pho cua cac toa do trung
4 # tam
5
6 store_states_cities = []
7 for centroid in centroids:
8     location = geolocator.reverse((centroid[0], centroid[1]),
9         language='en', timeout=10)
10    address = location.raw['address']
11    state = address.get('state', '')
12    city = address.get('city', '')
13    store_states_cities.append((state, city))
14
15 # Cap nhat bang stores voi state va city
16 cursor = db_connection.cursor()
17
18 # Them cot store_state va store_city neu chua co
19 cursor.execute("ALTER TABLE stores ADD COLUMN store_state VARCHAR
20 (255)")
21 cursor.execute("ALTER TABLE stores ADD COLUMN store_city VARCHAR
22 (255)")
23
24 # Tao dictionary de tra cuu state va city theo label
25 state_city_lookup = {i: store_states_cities[i] for i in range(
26 optimal_k)}
27
28 # Cap nhat cac hang trong bang stores voi state va city moi
29 for i, store_id in enumerate(store_ids):
30     state, city = state_city_lookup[labels[i]]
31     cursor.execute(
32         "UPDATE stores SET store_state = %s, store_city = %s WHERE
33             store_id = %s",
34         (state, city, store_id)
35     )
36
37 # Xoa cot store_latitude va store_longitude
38 cursor.execute("ALTER TABLE stores DROP COLUMN store_latitude")
39 cursor.execute("ALTER TABLE stores DROP COLUMN store_longitude")
40
41 # Luu thay doi va dong ket noi
42 db_connection.commit()
43 cursor.close()
44 db_connection.close()

```

Đoạn code trên được thiết kế để tự động xác định và cập nhật thông tin về bang và thành phố của các cửa hàng dựa trên tọa độ địa lý của chúng. Đầu tiên, thư viện geopy được sử dụng để tìm kiếm thông tin chi tiết từ tọa độ với sự hỗ trợ của dịch vụ Nominatim, bao gồm cả thông tin về bang (*state*) và thành phố (*city*). Sau khi các tọa độ trung tâm cụm (*centroids*) đã được xác định, đoạn mã lặp lại từng tọa độ này để truy vấn và lấy thông tin địa lý từ geopy.

Tiếp theo, thông tin về bang và thành phố của từng tọa độ được lưu trữ vào một danh sách *store_states_cities*. Sau khi thu thập đủ thông tin, đoạn mã sử dụng một kết nối cơ sở dữ liệu để cập nhật bảng "*stores*". Trước tiên, nó thêm hai cột mới là *store_state* và

`store_city` vào bảng nếu chúng chưa tồn tại, đảm bảo rằng bảng có đầy đủ thông tin để lưu trữ.

Một dictionary `state_city_lookup` được sử dụng để ánh xạ từ nhãn cụm (*cluster labels*) đến thông tin về bang và thành phố tương ứng đã được xác định từ geopy. Với mỗi cửa hàng trong danh sách `store_ids`, đoạn mã trích xuất `state` và `city` từ `state_city_lookup` dựa trên nhãn tương ứng được phân loại bởi thuật toán phân cụm K-means. Nó sau đó thực hiện các lệnh SQL để cập nhật bảng "stores" với thông tin mới này.

Cuối cùng, các cột `store_latitude` và `store_longitude` không còn cần thiết nữa và sẽ được xóa khỏi bảng để gọn dẹp dữ liệu. Quá trình này giúp tổ chức cơ sở dữ liệu hiệu quả hơn, cung cấp thông tin chi tiết về vị trí địa lý của từng cửa hàng và cải thiện khả năng phân tích và quản lý trong các hoạt động kinh doanh và điều hành.

Và đây là bảng stores sau khi đã chuyển tọa độ địa lý thành khu vực cụ thể (thành phố, bang):

After

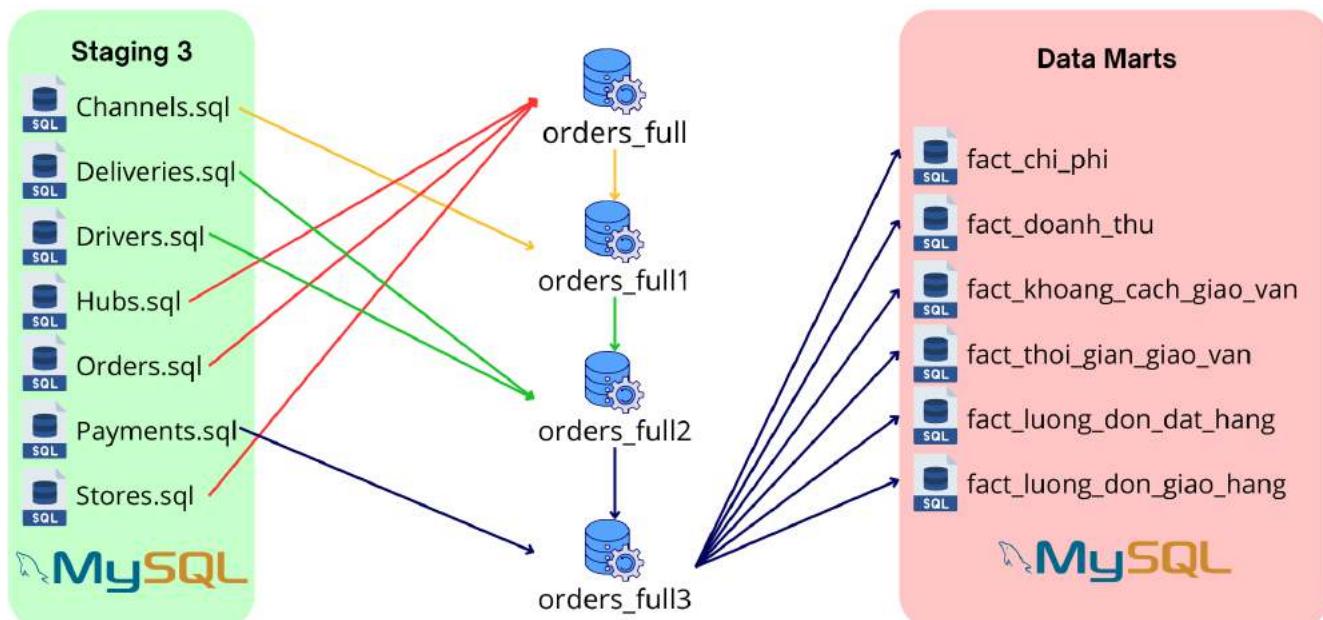
Phân cụm kinh vĩ độ và thu thập dữ liệu địa lý từ API

	store_id	hub_id	store_name	store_segment	store_plan_price	store_state	store_city
▶	3	2	CUMIURI	FOOD	0.0	Rio Grande do Sul	Porto Alegre
	6	3	PIMGUCIS DA VIVA	FOOD	0.0	Rio Grande do Sul	Porto Alegre
	8	3	RASMUR S	FOOD	0.0	Rio Grande do Sul	Porto Alegre
	53	8	PAPA SUCIS	FOOD	0.0	Rio de Janeiro	Rio de Janeiro
	54	8	VUZPI PAZZIS	FOOD	0.0	Rio de Janeiro	Rio de Janeiro
	56	8	SUPSIO	FOOD	49.0	Rio de Janeiro	Rio de Janeiro
	58	8	PIAMUARIS	FOOD	49.0	Rio de Janeiro	Rio de Janeiro
	82	8	LUCITA	FOOD	0.0	Rio de Janeiro	Rio de Janeiro
	83	8	PRARIZZAI	FOOD	0.0	Rio de Janeiro	Rio de Janeiro
	84	8	PALLO MZU GRALA	FOOD	49.0	Rio de Janeiro	Rio de Janeiro
	85	8	PRISMAURAI	FOOD	0.0	Rio de Janeiro	Rio de Janeiro
	88	8	EUGUSMI	GOOD	0.0	Rio de Janeiro	Rio de Janeiro
	89	8	LIS URPIMIOLUS	FOOD	49.0	Rio de Janeiro	Rio de Janeiro

Sau quá trình ETL, ta thu được mô hình dữ liệu OLTP như sau:



4. ETL dữ liệu từ Data Staging sang Data Warehouse



Ở trên là một phần của data pipeline từ Staging 3 sang Data Marts. Sau khi dữ liệu đã được xử lý, làm sạch, chuẩn hóa thông qua quá trình ETL thì ta sẽ thực hiện JOIN lần lượt

các bảng với nhau, tạo thành một bảng tổng hợp (orders_full3) và thực hiện các lệnh truy vấn trên bảng tổng hợp này để tạo ra các bảng fact trong Data Marts.

Để tạo ra các bảng dim, ta sẽ sử dụng câu lệnh SELECT DISTINCT từ các cột chứa thông tin trong các bảng:

```
CREATE TABLE dim_date AS
SELECT DISTINCT
    DATE(order_moment_created) AS date,
    MONTH(order_moment_created) AS month,
    YEAR(order_moment_created) AS year
FROM orders;
```

date	month	year
2021-01-01	1	2021
2021-01-02	1	2021
2021-01-03	1	2021
2021-01-04	1	2021
2021-01-05	1	2021
2021-01-06	1	2021
2021-01-07	1	2021
2021-01-08	1	2021
2021-01-09	1	2021
2021-01-10	1	2021
2021-01-11	1	2021

Hình 7: dim_date

```
CREATE TABLE location AS
SELECT DISTINCT
    hub_city AS city,
    hub_state AS state
FROM hubs;
```

city	state
CURITIBA	PR
PORTO ALEGRE	RS
RIO DE JANEIRO	RJ
SÃO PAULO	SP

Hình 8: dim_location

```
CREATE TABLE dim_hub AS
SELECT distinct hub_id, hub_name from hubs;
```

hub_id	hub_name
2	BLUE SHOPPING
3	GREEN SHOPPING
4	RED SHOPPING
5	FUNK SHOPPING
8	GOLDEN SHOPPING
13	HIP HOP SHOPPING
16	PEOPLE SHOPPING
17	SMALL SHOPPING
18	STAR SHOPPING
20	PURPLE SHOPPING

Hình 9: dim_hub

```
CREATE TABLE dim_channel AS
SELECT * FROM channels
```

channel_id	channel_name	channel_type
1	OTHER PLACE	OWN CHANNEL
2	PHONE PLACE	OWN CHANNEL
3	WHATS PLACE	OWN CHANNEL
4	FACE PLACE	OWN CHANNEL
5	FOOD PLACE	MARKETPLACE
6	STORE PLACE	OWN CHANNEL
7	BERLIN PLACE	OWN CHANNEL
8	MADRID PLACE	OWN CHANNEL
9	THINK PLACE	OWN CHANNEL
10	LISBON PLACE	OWN CHANNEL

Hình 10: dim_channel

```
CREATE TABLE dim_driver as
SELECT * FROM drivers
```

driver_id	driver_modal	driver_type
133	MOTOBOT	LOGISTIC OPERATOR
134	MOTOBOT	LOGISTIC OPERATOR
138	MOTOBOT	FREELANCE
140	MOTOBOT	FREELANCE
143	BIKER	FREELANCE
148	MOTOBOT	FREELANCE
165	MOTOBOT	FREELANCE
172	MOTOBOT	FREELANCE
174	BIKER	FREELANCE
187	BIKER	FREELANCE
196	BIKER	FREELANCE
202	BIKER	FREELANCE
210	MOTOBOT	FREELANCE
217	MOTOBOT	LOGISTIC OPERATOR

Hình 11: dim_driver

```
CREATE TABLE dim_payment_method AS
SELECT DISTINCT payment_method FROM payments;
INSERT INTO dim_payment_method (payment_method)
VALUES ('NONE');
```

payment_method
BANK_TRANSFER_DC
CREDIT
CREDIT_STORE
DEBIT
DEBIT_STORE
INSTALLMENT_CREDIT_STORE
MEAL_BENEFIT
MONEY
NONE
ONLINE
PAYMENT_LINK
STORE_DIRECT_PAYMENT
VOUCHER
VOUCHER_DC
VOUCHER_DL
VOUCHER_STORE

Hình 12: dim_payment_method

```
CREATE TABLE dim_store AS
SELECT store_id, store_name, store_segment AS product_type
FROM stores
```

store_id	store_name	product_type
3	CUMIURI	FOOD
6	PIMGUCIS DA VIVA	FOOD
8	RASMUR S	FOOD
53	PAPA SUCIS	FOOD
54	VUZPI PAZZIS	FOOD
56	SUPSIO	FOOD
58	PIAMUARIS	FOOD
82	LUCITA	FOOD
83	PRARIZZAI	FOOD
84	PALLO MZU GRALA	FOOD
85	PRISMAURAI	FOOD
88	EUGUSMI	GOOD

Hình 13: dim_store

```
CREATE TABLE dim_order_status
SELECT DISTINCT order_status FROM orders
```

order_status
CANCELED
FINISHED

Hình 14: dim_order_status

```
CREATE TABLE dim_order_status
SELECT DISTINCT delivery_status FROM deliveries
```

delivery_status
CANCELLED
DELIVERED

Hình 15: dim_delivery_status

```
CREATE TABLE dim_payment_status AS
SELECT DISTINCT payment_status FROM payments;
INSERT INTO dim_payment_status (payment_status)
VALUES ('NONE');
```

payment_status
CHARGEBACK
NONE
PAID

Hình 16: dim_payment_status

```
CREATE TABLE dim_delivery_range AS
SELECT distinct delivery_range FROM deliveries;
```

delivery_range
Medium
Small
Very Large
Large

Hình 17: dim_delivery_range

```
CREATE TABLE dim_order_value AS
SELECT distinct order_amount_value from orders
```

order_amount_value
Low
Medium
High
Very High

Hình 18: dim_order_value

Tiếp theo, sử dụng các câu lệnh JOIN để nối dàn các bảng trong cơ sở dữ liệu với nhau, tạo thành bảng orders_full3.

```
CREATE TABLE orders_full AS
SELECT
    o.*,
    s.hub_id,
    h.hub_city,
    h.hub_state,
    h.hub_name,
    s.store_name, s.store_segment
FROM orders o
INNER JOIN stores s ON o.store_id = s.store_id
INNER JOIN hubs h ON s.hub_id = h.hub_id;
```

```
CREATE TABLE orders_full2 AS
SELECT o.*, d.driver_modal, d.driver_type, del.delivery_status, d.driver_id,
del.delivery_distance_meters, del.delivery_range
FROM orders_full1 o
LEFT JOIN deliveries del ON o.order_id = del.delivery_order_id
LEFT JOIN drivers d ON del.driver_id = d.driver_id;
```

```
CREATE TABLE orders_full1 AS
SELECT o.* , c.channel_type, c.channel_name
FROM orders_full o
JOIN channels c ON o.channel_id = c.channel_id;
```

```
CREATE TABLE orders_full3 AS
SELECT o.* , p.payment_method, p.payment_status
FROM orders_full2 o
LEFT JOIN payments as p on o.order_id = p.payment_order_id
```

order_id	store_id	channel_id	order_status	order_amount	order_amount_value	order_delivery_fee	order_delivery_cost	order_moment_created	order_moment_collected	order_moment_finished	hub_id	hub_city
68410055	2181	35	FINISHED	394.80	Medium	169.00	6.00	2021-01-01 02:32:51	2021-01-01 03:04:51	2021-01-01 07:46:51	13	RIO DE JANEIRO
68410055	2181	35	FINISHED	394.80	Medium	169.00	6.00	2021-01-01 02:32:51	2021-01-01 03:04:51	2021-01-01 07:46:51	13	RIO DE JANEIRO
68412721	631	5	FINISHED	195.05	Medium	12.00	11.00	2021-01-01 14:12:11	2021-01-01 15:38:11	2021-01-01 22:44:11	28	SÃO PAULO
68413340	631	5	FINISHED	46.90	Low	12.00	11.00	2021-01-01 14:14:51	2021-01-01 15:14:51	2021-01-01 16:24:51	28	SÃO PAULO
68414018	3265	5	FINISHED	45.80	Low	79.00	10.00	2021-01-01 14:17:31	2021-01-01 15:15:31	2021-01-01 20:34:31	37	SÃO PAULO
68414309	236	5	FINISHED	94.90	Low	12.00	6.00	2021-01-01 14:21:02	2021-01-01 15:53:02	2021-01-01 23:25:02	13	RIO DE JANEIRO
68414512	631	5	FINISHED	45.90	Low	225.00	12.00	2021-01-01 14:24:01	2021-01-01 15:57:01	2021-01-01 20:49:01	28	SÃO PAULO
68414563	955	5	FINISHED	26.90	Low	281.00	6.00	2021-01-01 14:24:26	2021-01-01 15:21:26	2021-01-01 21:23:26	51	RIO DE JANEIRO
68415103	631	5	FINISHED	103.60	Medium	12.00	6.00	2021-01-01 14:31:33	2021-01-01 16:02:33	2021-01-01 20:06:33	28	SÃO PAULO
68415140	631	5	FINISHED	187.40	Medium	12.00	4.00	2021-01-01 14:31:52	2021-01-01 15:12:52	2021-01-01 22:06:52	28	SÃO PAULO
68415344	631	5	FINISHED	83.40	Low	12.00	11.00	2021-01-01 14:33:32	2021-01-01 16:03:32	2021-01-01 20:59:32	28	SÃO PAULO

Hình 19: Bảng order_full3

Cuối cùng, sử dụng các câu lệnh truy vấn trên bảng tổng hợp orders_full3 để tạo ra các bảng fact:

```

CREATE TABLE doanh_thu AS
SELECT
    DATE(order_moment_created) AS date,
    hub_state AS state,
    hub_city AS city,
    channel_id,
    channel_name,
    channel_type,
    store_id,
    store_name,
    store_segment AS product_type,
    SUM(CASE WHEN order_status = 'FINISHED' THEN order_amount + order_delivery_fee ELSE 0 END) AS order_revenue,
    SUM(order_delivery_fee - order_delivery_cost) AS delivery_revenue
FROM orders_full3
WHERE delivery_status = 'DELIVERED'
GROUP BY date, state, city, channel_id, channel_name, channel_type, store_id, store_name, product_type;

```

date	state	city	channel_id	channel_name	channel_type	store_id	store_name	product_type	order_revenue	delivery_revenue
2021-01-01	PR	CURITIBA	5	FOOD PLACE	MARKETPLACE	2948	GUIRGUES	FOOD	2761.0	2018
2021-01-01	PR	CURITIBA	15	EATS PLACE	MARKETPLACE	2060	FIPRACI EA MPECILIMU	FOOD	466.0	382
2021-01-01	PR	CURITIBA	15	EATS PLACE	MARKETPLACE	2061	FIPRACI EA MPECILIMU	FOOD	158.5	92
2021-01-01	RS	PORTO ALEGRE	5	FOOD PLACE	MARKETPLACE	8	RASMUR S	FOOD	2602.5	1624
2021-01-01	RS	PORTO ALEGRE	5	FOOD PLACE	MARKETPLACE	656	IPALUPAES	FOOD	3285.6	1259
2021-01-01	RS	PORTO ALEGRE	5	FOOD PLACE	MARKETPLACE	1016	PIPS	FOOD	4242.2	2924
2021-01-01	RS	PORTO ALEGRE	5	FOOD PLACE	MARKETPLACE	1057	IUMPICA	FOOD	5974.2	1554
2021-01-01	RS	PORTO ALEGRE	5	FOOD PLACE	MARKETPLACE	1804	RIRAI CISAMZI	FOOD	415.6	344
2021-01-01	RS	PORTO ALEGRE	10	LISBON PLACE	OWN CHANNEL	1018	ICI EU SIVEMI	FOOD	948.5	437
2021-01-01	RS	PORTO ALEGRE	13	VELOCITY PLACE	MARKETPLACE	8	RASMUR S	FOOD	703.5	435
2021-01-01	RS	PORTO ALEGRE	13	VELOCITY PLACE	MARKETPLACE	563	MZU PLICA PAF	FOOD	364.0	272
2021-01-01	RS	PORTO ALEGRE	15	EATS PLACE	MARKETPLACE	8	RASMUR S	FOOD	818.0	652
2021-01-01	RS	PORTO ALEGRE	15	EATS PLACE	MARKETPLACE	1324	PAZZI PIRPACO	FOOD	1854.0	1243
2021-01-01	RS	PORTO ALEGRE	15	EATS PLACE	MARKETPLACE	1804	RIRAI CISAMZI	FOOD	187.7	151
2021-01-01	RJ	RIO DE JANEIRO	5	FOOD PLACE	MARKETPLACE	58	PIAMUARIS	FOOD	516.4	241

Hình 20: Bảng doanh thu

```

CREATE TABLE chi_phi AS
SELECT
    DATE(order_moment_created) AS date,
    hub_state AS state,
    hub_city AS city,
    hub_id,
    hub_name,
    driver_id,
    driver_modal,
    driver_type, delivery_range,
    store_segment as product_type,
    SUM(order_delivery_cost) AS delivery_cost
FROM orders_full3
WHERE delivery_status = 'DELIVERED'
GROUP BY date, state, city, hub_id, hub_name, driver_id, driver_modal, driver_type, delivery_range, product_type;

```

date	state	city	hub_id	hub_name	driver_id	driver_modal	driver_type	product_type	delivery_range	delivery_cost
2021-01-01	PR	CURITIBA	42	PHP SHOPPING	10616	MOTOBOY	LOGISTIC OPERATOR	FOOD	Small	8
2021-01-01	PR	CURITIBA	42	PHP SHOPPING	11615	MOTOBOY	FREELANCE	FOOD	Small	9
2021-01-01	PR	CURITIBA	42	PHP SHOPPING	21923	MOTOBOY	LOGISTIC OPERATOR	FOOD	Small	26
2021-01-01	PR	CURITIBA	42	PHP SHOPPING	22198	MOTOBOY	LOGISTIC OPERATOR	FOOD	Small	18
2021-01-01	PR	CURITIBA	42	PHP SHOPPING	23784	MOTOBOY	LOGISTIC OPERATOR	FOOD	Small	9
2021-01-01	PR	CURITIBA	42	PHP SHOPPING	24584	MOTOBOY	FREELANCE	FOOD	Small	32
2021-01-01	PR	CURITIBA	42	PHP SHOPPING	26319	MOTOBOY	LOGISTIC OPERATOR	FOOD	Small	19
2021-01-01	PR	CURITIBA	42	PHP SHOPPING	29271	MOTOBOY	LOGISTIC OPERATOR	FOOD	Small	9
2021-01-01	PR	CURITIBA	43	HOTMILK SHOPPING	29341	MOTOBOY	FREELANCE	FOOD	Small	16
2021-01-01	RS	PORTO ALEGRE	3	GREEN SHOPPING	378	MOTOBOY	FREELANCE	FOOD	Medium	11
2021-01-01	RS	PORTO ALEGRE	3	GREEN SHOPPING	378	MOTOBOY	FREELANCE	FOOD	Small	56
2021-01-01	RS	PORTO ALEGRE	3	GREEN SHOPPING	526	MOTOBOY	LOGISTIC OPERATOR	FOOD	Medium	7

Hình 21: Bảng chi phí

```
CREATE TABLE don_dat AS
```

```

SELECT DATE(order_moment_created) AS date,
    DAYNAME(order_moment_created) AS day_of_week, HOUR(order_moment_created) AS hour,
    hub_state AS state, hub_city AS city,
    channel_id, channel_name, channel_type,
    order_amount_value, orders_full3.order_status AS order_status,
    store_id, store_name, store_segment AS product_type,
    COALESCE(payment_method, 'NONE') AS payment_method,
    COALESCE(payment_status, 'NONE') AS payment_status,
    COUNT(order_id) AS quantity_of_orders
FROM orders_full3
GROUP BY date, day_of_week, hour, hub_state, hub_city, channel_id, channel_name, channel_type,
order_amount_value, order_status, store_id, store_name, product_type, payment_method, payment_status;

```

date	day_of_week	hour	state	city	channel_id	channel_name	channel_type	order_amount_value	order_status	store_id	store_name	product_type	payment_method	payment_status	quantity
2021-01-01	Friday	2	RJ	RIO DE JANEIRO	35	BRAZIL PLACE	MARKETPLACE	Medium	FINISHED	2181	LILI CISRUMAC'S	GOOD	ONLINE	PAID	1
2021-01-01	Friday	2	RJ	RIO DE JANEIRO	35	BRAZIL PLACE	MARKETPLACE	Medium	FINISHED	2181	LILI CISRUMAC'S	GOOD	VOUCHER	PAID	1
2021-01-01	Friday	14	RS	PORTO ALEGRE	5	FOOD PLACE	MARKETPLACE	Low	FINISHED	1016	PIPS	FOOD	ONLINE	PAID	1
2021-01-01	Friday	14	RJ	RIO DE JANEIRO	5	FOOD PLACE	MARKETPLACE	Low	FINISHED	236	ILIMPICA	FOOD	ONLINE	PAID	1
2021-01-01	Friday	14	RJ	RIO DE JANEIRO	5	FOOD PLACE	MARKETPLACE	Low	FINISHED	399	PIARO	FOOD	DEBIT_STORE	PAID	1
2021-01-01	Friday	14	RJ	RIO DE JANEIRO	5	FOOD PLACE	MARKETPLACE	Low	FINISHED	955	CIRICUIU CAI	FOOD	DEBIT	PAID	1
2021-01-01	Friday	14	RJ	RIO DE JANEIRO	5	FOOD PLACE	MARKETPLACE	Low	FINISHED	955	CIRICUIU CAI	FOOD	ONLINE	PAID	3
2021-01-01	Friday	14	RJ	RIO DE JANEIRO	5	FOOD PLACE	MARKETPLACE	Low	FINISHED	3528	ZIM M NUMEUR	FOOD	ONLINE	PAID	1
2021-01-01	Friday	14	RJ	RIO DE JANEIRO	5	FOOD PLACE	MARKETPLACE	Low	FINISHED	3528	ZIM M NUMEUR	FOOD	VOUCHER	PAID	1
2021-01-01	Friday	14	SP	SÃO PAULO	5	FOOD PLACE	MARKETPLACE	Low	FINISHED	631	ILIMPICA	FOOD	ONLINE	PAID	3
2021-01-01	Friday	14	SP	SÃO PAULO	5	FOOD PLACE	MARKETPLACE	Low	FINISHED	3265	EAVALI FIGOI	FOOD	ONLINE	PAID	7
2021-01-01	Friday	14	SP	SÃO PAULO	5	FOOD PLACE	MARKETPLACE	Medium	FINISHED	631	ILIMPICA	FOOD	ONLINE	PAID	7
2021-01-01	Friday	15	PR	CURITIBA	5	FOOD PLACE	MARKETPLACE	Low	FINISHED	2948	GLURIGLUES	FOOD	ONLINE	PAID	2
2021-01-01	Friday	15	RS	PORTO ALEGRE	5	FOOD PLACE	MARKETPLACE	Low	FINISHED	1016	PIPS	FOOD	ONLINE	PAID	1

Hình 22: Bảng đơn đặt

```

CREATE TABLE don_giao AS
SELECT
    DATE(order_moment_created) AS date,
    DAYNAME(order_moment_created) AS day_of_week, HOUR(order_moment_created) AS hour,
    hub_state AS state, hub_city AS city, hub_id, hub_name,
    driver_id, driver_modal, driver_type,
    order_amount_value,
    store_segment AS product_type,
    delivery_status,
    COUNT(order_id) AS quantity_of_orders
FROM orders_full3
WHERE delivery_status IS NOT NULL
GROUP BY date, day_of_week, hour, state, city, hub_id, hub_name,
driver_id, driver_modal, driver_type, order_amount_value, product_type, delivery_status;

```

date	day_of_week	hour	state	city	hub_id	hub_name	driver_id	driver_modal	driver_type	order_amount_value	product_type	delivery_status	quantity_of_orders
2021-01-01	Friday	1	RJ	RIO DE JANEIRO	8	GOLDEN SHOPPING	8598	MOTOBOT	LOGISTIC OPERATOR	Medium	GOOD	CANCELLED	1
2021-01-01	Friday	2	RJ	RIO DE JANEIRO	13	HIP HOP SHOPPING	10239	BIKER	FREELANCE	Medium	GOOD	DELIVERED	2
2021-01-01	Friday	14	RS	PORTO ALEGRE	36	BLACK SHOPPING	7615	MOTOBOT	LOGISTIC OPERATOR	Low	FOOD	DELIVERED	1
2021-01-01	Friday	14	RJ	RIO DE JANEIRO	13	HIP HOP SHOPPING	2473	BIKER	FREELANCE	Low	FOOD	DELIVERED	1
2021-01-01	Friday	14	RJ	RIO DE JANEIRO	13	HIP HOP SHOPPING	8456	MOTOBOT	LOGISTIC OPERATOR	Low	FOOD	DELIVERED	1
2021-01-01	Friday	14	RJ	RIO DE JANEIRO	51	RUBY SHOPPING	14511	MOTOBOT	FREELANCE	Low	FOOD	DELIVERED	2
2021-01-01	Friday	14	RJ	RIO DE JANEIRO	51	RUBY SHOPPING	14513	MOTOBOT	FREELANCE	Low	FOOD	DELIVERED	1
2021-01-01	Friday	14	RJ	RIO DE JANEIRO	51	RUBY SHOPPING	23092	BIKER	FREELANCE	Low	FOOD	DELIVERED	1
2021-01-01	Friday	14	RJ	RIO DE JANEIRO	51	RUBY SHOPPING	25575	MOTOBOT	LOGISTIC OPERATOR	Low	FOOD	DELIVERED	1
2021-01-01	Friday	14	RJ	RIO DE JANEIRO	51	RUBY SHOPPING	29669	MOTOBOT	FREELANCE	Low	FOOD	DELIVERED	1
2021-01-01	Friday	14	SP	SÃO PAULO	28	RAP SHOPPING	8378	MOTOBOT	FREELANCE	Low	FOOD	DELIVERED	1
2021-01-01	Friday	14	SP	SÃO PAULO	28	RAP SHOPPING	8378	MOTOBOT	FREELANCE	Medium	FOOD	DELIVERED	1
2021-01-01	Friday	14	SP	SÃO PAULO	28	RAP SHOPPING	9996	BIKER	FREELANCE	Medium	FOOD	DELIVERED	4
2021-01-01	Friday	14	SP	SÃO PAULO	28	RAP SHOPPING	16430	BIKER	FREELANCE	Medium	FOOD	DELIVERED	1

Hình 23: Bảng đơn giao

```

CREATE TABLE thoi_gian_giao_van AS
SELECT
    DATE(order_moment_created) AS date,
    hub_state AS state, hub_city AS city, hub_id,
    driver_id, driver_modal, driver_type,
    delivery_range,
    store_segment AS product_type,
    CEIL(SUM(TIMESTAMPDIFF(SECOND, order_moment_created, order_moment_collected) / 60)) AS Thoi_Gian_Chuan_Bi_Hang,
    CEIL(SUM(TIMESTAMPDIFF(SECOND, order_moment_collected, order_moment_finished) / 60)) AS Thoi_Gian_Giao_Hang,
    CEIL(SUM(TIMESTAMPDIFF(SECOND, order_moment_created, order_moment_collected) / 60)) +
    CEIL(SUM(TIMESTAMPDIFF(SECOND, order_moment_collected, order_moment_finished) / 60)) AS Thoi_Gian_Xu_Ly_Don_Hang
FROM orders_full3
WHERE
    delivery_status = 'DELIVERED'
GROUP BY
    date, state, city, hub_id, driver_id, driver_modal, driver_type, delivery_range, product_type;

```

date	state	city	hub_id	driver_id	driver_modal	driver_type	delivery_range	product_type	order_preparation_time	delivery_time	handling_time
2021-01-01	PR	CURITIBA	42	10616	MOTOBOY	LOGISTIC OPERATOR	Small	FOOD	47	339	386
2021-01-01	PR	CURITIBA	42	11615	MOTOBOY	FREELANCE	Small	FOOD	52	286	338
2021-01-01	PR	CURITIBA	42	21923	MOTOBOY	LOGISTIC OPERATOR	Small	FOOD	259	844	1103
2021-01-01	PR	CURITIBA	42	22198	MOTOBOY	LOGISTIC OPERATOR	Small	FOOD	162	506	668
2021-01-01	PR	CURITIBA	42	23784	MOTOBOY	LOGISTIC OPERATOR	Small	FOOD	57	319	376
2021-01-01	PR	CURITIBA	42	24584	MOTOBOY	FREELANCE	Small	FOOD	324	716	1040
2021-01-01	PR	CURITIBA	42	26319	MOTOBOY	LOGISTIC OPERATOR	Small	FOOD	104	645	749
2021-01-01	PR	CURITIBA	42	29271	MOTOBOY	LOGISTIC OPERATOR	Small	FOOD	100	343	443
2021-01-01	PR	CURITIBA	43	29341	MOTOBOY	FREELANCE	Small	FOOD	147	737	884
2021-01-01	RS	PORTO ALEGRE	3	378	MOTOBOY	FREELANCE	Medium	FOOD	71	206	277
2021-01-01	RS	PORTO ALEGRE	3	378	MOTOBOY	FREELANCE	Small	FOOD	416	1725	2141

Hình 24: Bảng thời gian giao vận

```

CREATE TABLE khoang_cach_giao_van AS
SELECT
    DATE(order_moment_created) AS date,
    hub_state AS state,
    hub_city AS city,
    driver_id,
    driver_modal,
    driver_type,
    delivery_range,
    SUM(delivery_distance_meters) as distance
FROM orders_full3
WHERE delivery_status = 'DELIVERED'
GROUP BY date, state, city, driver_id, driver_modal, driver_type, delivery_range;

```

date	state	city	driver_id	driver_modal	driver_type	delivery_range	distance
2021-01-01	PR	CURITIBA	10616	MOTOBOY	LOGISTIC OPERATOR	Small	408
2021-01-01	PR	CURITIBA	11615	MOTOBOY	FREELANCE	Small	3426
2021-01-01	PR	CURITIBA	21923	MOTOBOY	LOGISTIC OPERATOR	Small	8935
2021-01-01	PR	CURITIBA	22198	MOTOBOY	LOGISTIC OPERATOR	Small	5638
2021-01-01	PR	CURITIBA	23784	MOTOBOY	LOGISTIC OPERATOR	Small	3198
2021-01-01	PR	CURITIBA	24584	MOTOBOY	FREELANCE	Small	6613
2021-01-01	PR	CURITIBA	26319	MOTOBOY	LOGISTIC OPERATOR	Small	8345
2021-01-01	PR	CURITIBA	29271	MOTOBOY	LOGISTIC OPERATOR	Small	3480
2021-01-01	PR	CURITIBA	29341	MOTOBOY	FREELANCE	Small	4989
2021-01-01	RS	PORTO ALEGRE	134	MOTOBOY	LOGISTIC OPERATOR	Medium	7318
2021-01-01	RS	PORTO ALEGRE	134	MOTOBOY	LOGISTIC OPERATOR	Small	4477

Hình 25: Bảng khoảng cách giao vận

Cuối cùng, để hoàn thành quy trình tự động hóa, ta tạo 1 file python để điều hành các file python khác làm nhiệm vụ ETL và tạo dim, fact:

```

1 import subprocess
2 import sys
3
4 def run_files_in_sequence(files):
5     for file in files:
6         try:
7             subprocess.run([sys.executable, file], check=True)
8             print(f"{file} da chay xong.")
9         except subprocess.CalledProcessError as e:
10            print(f'Da xay ra loi khi chay {file}: {e}')
11 files_to_run = ["ETL_1_Delete_Column.py", "ETL_2_Remove_Duplicate.py",
12                 'ETL_3_Handle_Null.py', 'ETL_4_Data_Binning.py',
13                 'ETL_5_Unique.py', 'ETL_6_stores_by_Clustering.py', 'Fact', 'Dim']
14
15 run_files_in_sequence(files_to_run)
16 print("Chuong trinh da chay xong!")

```

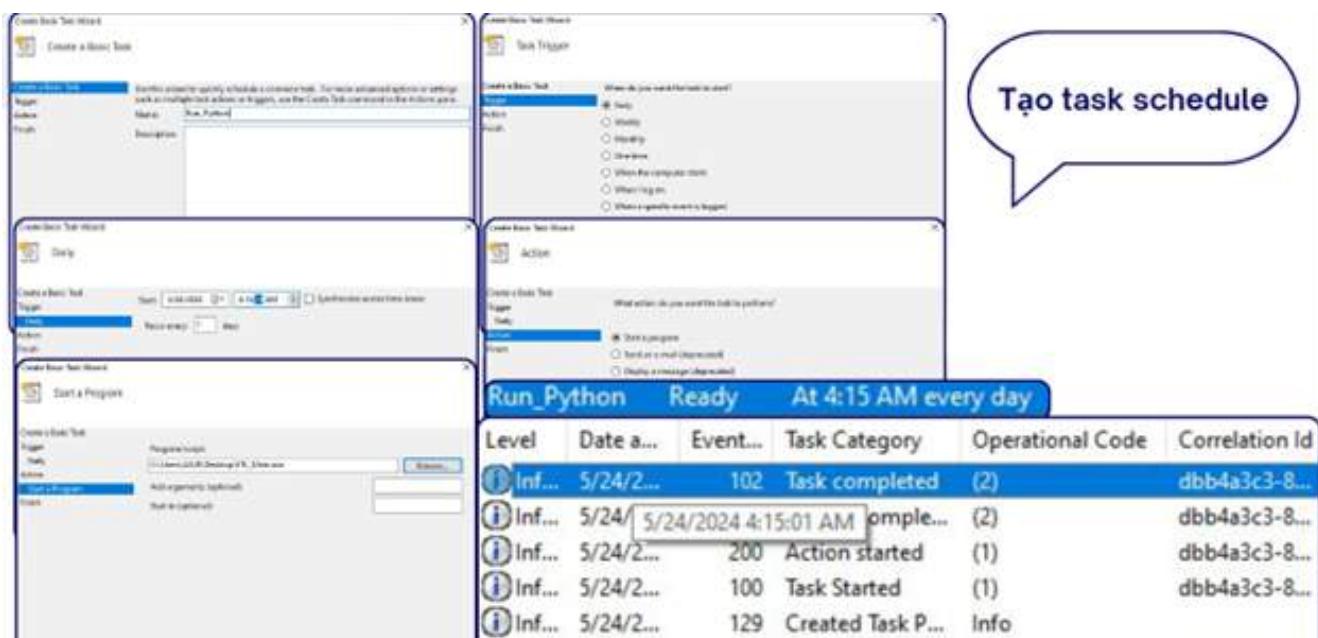
Ta sử dụng 1 file cpp viết bằng C++ để chạy file điều hành python trên. Cụ thể:

```

1 #include <iostream>
2 using namespace std;
3 int main() {
4     const char* command = "C:\\\\Users\\\\ASUS\\\\Desktop\\\\CODE\\\\ETL\\\\ETL_Main
5         .py";
6     cout << "Running command: " << command << endl;
7     system(command);
8     return 0;
}

```

Sau đó, ta thực hiện lấy đường dẫn của file thực thi viết bằng C++ (có định dạng .exe) vào công cụ task schedule để lập lịch trình chạy tự động. Chi tiết các bước thực hiện được hiển thị trong hình dưới đây:



Hình 26: Task Schedule

3.2.4 Hệ thống chiều khái niệm (Voi Dimension)

dim_channel

channel_name	channel_type
OTHER PLACE	OWN CHANNEL
PHONE PLACE	MARKETPLACE
WHATS PLACE	2 Values
FACE PLACE	
FOOD PLACE	
STORE PLACE	
...	

40 Values

dim_delivery

delivery_status	delivery_range
CANCELLED	Large
DELIVERED	Medium
2 Values	Small
	Very Large
	4 Values

dim_driver

driver_modal	driver_type
MOTOBOY	LOGISTIC OPERATOR
BIKER	FREELANCE
2 Values	2 Values

dim_product		dim_order		dim_date				
store_name	product_type	order_status	order_amount_value	year	month	day	dayofweek	hour
CUMIURI	FOOD	CANCELED	High	2021	1	1	Sunday	0
PIMGUCIS DA VIVA	GOOD	FINISHED	Low	1 Value	2	2	Monday	1
RASMURS	2 Values	2 Values	Medium		3	3	Tuesday	2
PAPA SUCIS			Very High		4	4	Wednesday	3
VUZPI PAZZIS		4 Values		4 Values	5	Thursday	4	
SUPSIOS					6	Friday	5	
...					7	Saturday	6	
					...	7 Values	...	
480 Values				31 Values				24 Values

dim_payment	
payment_method	payment_status
BANK_TRANSFER_DC	CHARGEBACK
CREDIT	NONE
CREDIT_STORE	PAID
DEBIT	3 Values
DEBIT_STORE	
INSTALLMENT_CREDIT_STORE	
...	

16 Values

dim_hub
hub_name
BLUE SHOPPING
GREEN SHOPPING
RED SHOPPING
FUNK SHOPPING
GOLDEN SHOPPING
HIP HOP SHOPPING
PEOPLE SHOPPING
...

dim_location	
city	state
CURITIBA	RIO GRANDE DO SUL
PORTO ALEGRE	RIO DE JANEIRO
RIO DE JANEIRO	SÃO PAULO
SÃO PAULO	PARANÁ

Sau quá trình ETL, từ mô hình dữ liệu OLTP ban đầu, ta phân tích được 9 bảng dim là các Dimension dữ liệu sẽ sử dụng trong quá trình phân tích.

1. dim channel: Chứa các thông tin về các kênh bán hàng (Channels)

- ***channel_name***: Gồm 40 giá trị, là tên của các Channel. (Ví dụ: OTHER PLACE, PHONE PLACE,...)
 - ***channel_type***: Gồm 2 giá trị, là kiểu kênh Channel. Cụ thể ở bộ dữ liệu này Channel có 2 loại là OWN CHANNEL và MARKETPLACE.

2. dim delivery: Chứa các thông tin về giao vận của các đơn hàng (Deliveries).

- ***delivery_status***: Gồm 2 giá trị là CANCELLED (Đơn hàng bị hủy) và DELIVERED (Giao thành công), là trạng thái vận chuyển của đơn hàng.
 - ***delivery_range***: Gồm 4 giá trị, là khoảng cách vận chuyển của đơn hàng. Ở bộ dữ liệu này, khoảng cách được chia thành 4 giá trị là Very Large (Rất lớn), Large (Lớn), Medium (Trung bình) và Small (Nhỏ).

3. dim driver: Thông tin về các tài xế giao hàng (Drivers).

- ***driver_modal***: Gồm 2 giá trị là MOTOBOY (Xe có thùng hàng) và BIKER (Xe máy không có thùng hàng), là các loại phương tiện của Driver
 - ***driver_type***: Gồm 2 giá trị, là loại tài xế, được chia thành 2 giá trị là LOGISTIC OPERATOR (Nhà vận hành logistics), FREELANCE (Tài xế tự do).

4. dim_product: Thông tin về các sản phẩm được bán trong bộ dữ liệu (Products) và cửa hàng phân phối các sản phẩm (Stores).

- ***store_name***: Gồm 480 giá trị, là tên của các cửa hàng cung cấp sản phẩm. (Ví dụ: CUMIURI, RASMUR S,...)
- ***product_type***: Gồm 2 giá trị, là loại sản phẩm, được chia thành 2 loại là FOOD (Thực phẩm), GOOD (Hàng hóa).

5. dim_order: Thông tin về các đơn hàng (Orders).

- ***order_status***: Gồm 2 giá trị là CANCELED (Bị hủy) và FINISHED (Đã hoàn thành), là trạng thái của đơn hàng.
- ***order_amount_value***: Gồm 4 giá trị, biểu thị giá trị đơn hàng. Các giá trị đơn hàng được chia thành các mức: Very High (Rất cao), High (Cao), Medium (Trung bình), Low (Thấp).

6. dim_date: Thông tin về thời gian (Dates).

- ***year***: Gồm 1 giá trị là năm 2021.
- ***month***: Gồm 4 giá trị. Dữ liệu trong khoảng từ tháng 1 tới hết tháng 4.
- ***day***: Gồm 31 giá trị, là các ngày trong tháng. (Ví dụ: 1, 2, ..., 31)
- ***dayofweek***: Gồm 7 giá trị, là các ngày trong tuần. (Ví dụ: Sunday, Monday,..., Saturday)
- ***hour***: Gồm 24 giá trị, là đơn vị chỉ giờ trong 1 ngày. (Ví dụ: 0, 1, ..., 23)

7. dim_payment: Thông tin về phương thức thanh toán (Payments).

- ***payment_method***: Gồm 16 giá trị, chỉ các phương thức thanh toán khách hàng thực hiện đối với đơn hàng. (Ví dụ: BANK_TRANSFER_DC (Chuyển khoản ngân hàng), CREDIT (Tín dụng), CREDIT_STORE (Tín dụng cửa hàng), DEBIT (Ghi nợ))
- ***payment_status***: Gồm 3 giá trị, biểu thị trạng thái thanh toán. Các trạng thái bao gồm CHARGEBACK (Hoàn trả), PAID (Đã thanh toán), NONE (Không).

8. dim_hub: Thông tin về các trung tâm phân phối hàng hóa (Hubs).

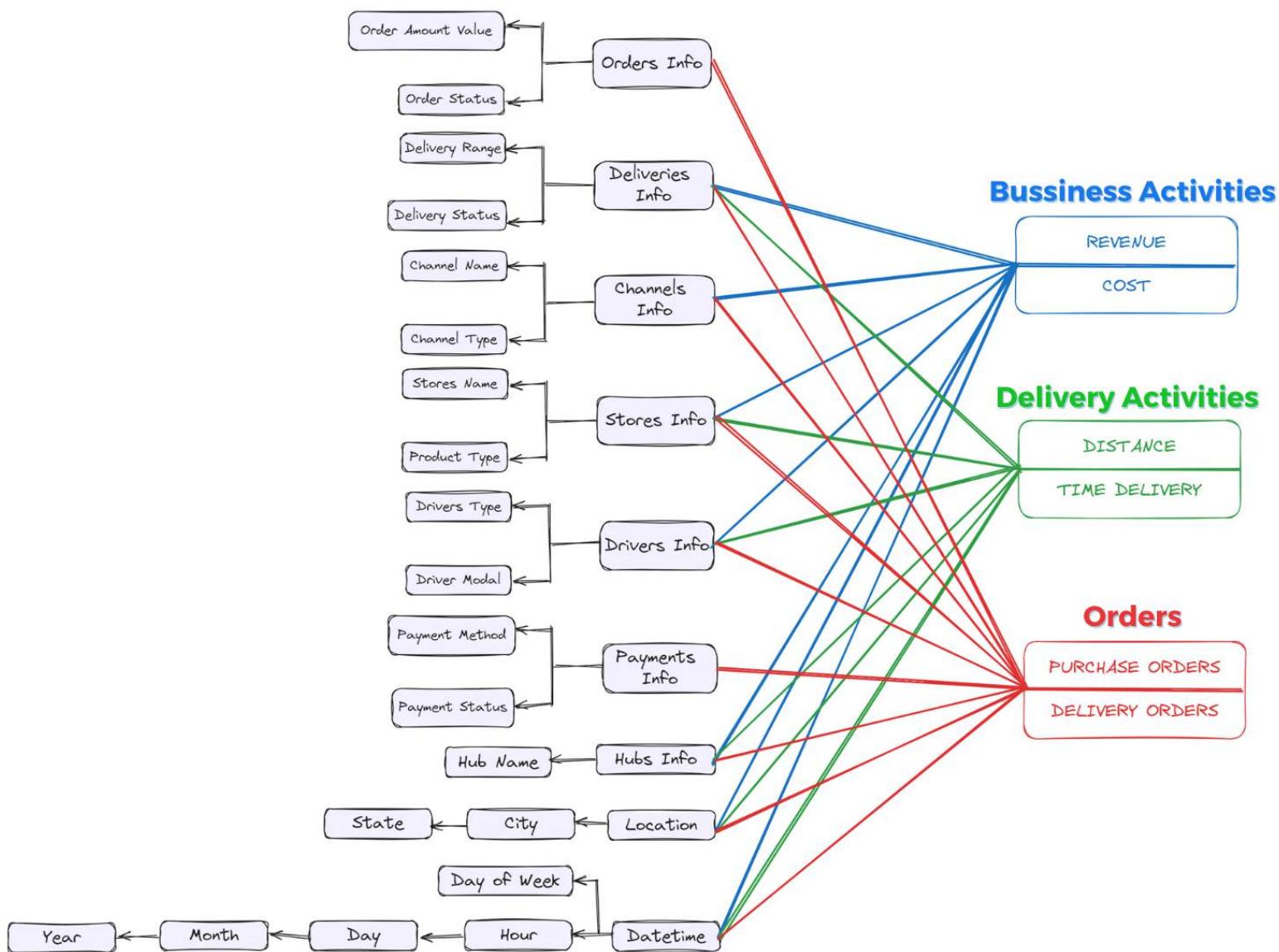
- ***hub_name***: Gồm 32 giá trị, là tên của từng trung tâm phân phối. (Ví dụ: BLUE SHOPPING, HIP HOP SHOPPING,...)

9. dim_location: Thông tin về địa điểm (Locations).

- ***city***: Gồm 4 giá trị, là các thành phố có trong tập dữ liệu. Ở đây gồm các thành phố: CURITIBA, PORTO ALEGRE, RIO DE JANEIRO, SÃO PAULO.
- ***state***: Gồm 4 giá trị, là các bang có trong tập dữ liệu. Ở đây gồm các bang: RIO GRANDE DO SUL, RIO DE JANEIRO, SÃO PAULO, PARANÁ.

3.2.5 Data Model

1. Logical Data Model



Mô hình Logical Data Logic: Ta sẽ phân tích dữ liệu theo 3 chủ điểm chính là Hoạt động kinh doanh (Business Activities), Hoạt động giao vận (Delivery Activities) và Đơn hàng (Orders). Các chủ điểm được phân tích theo các thông tin là các Dim dữ liệu đã xác định thông qua Hệ thống chiều khái niệm.

Mỗi chủ điểm phân tích chia nhỏ làm 2 chủ điểm nhỏ, về sau sẽ trở thành các fact nhỏ để phân tích.

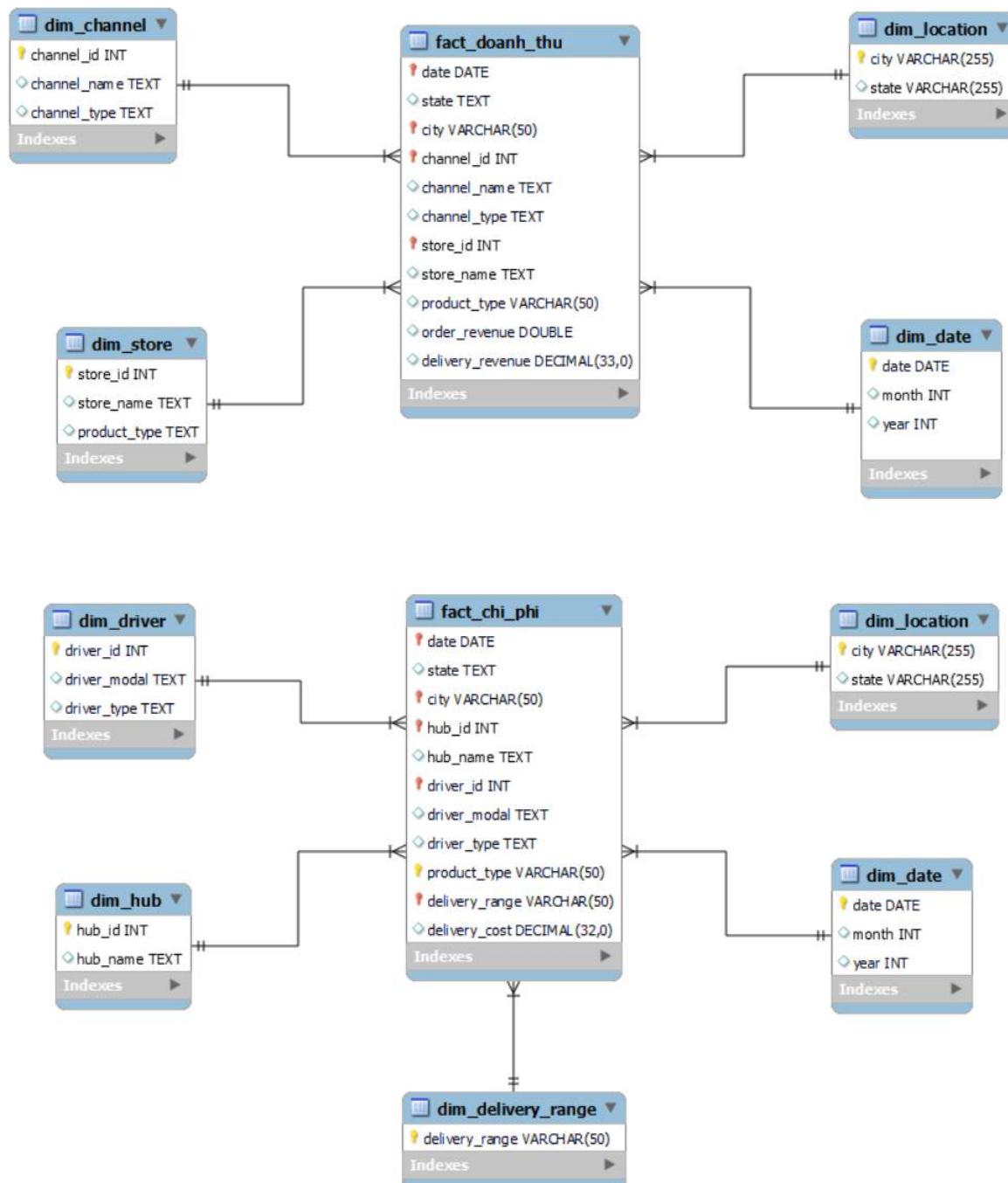
- Đối với chủ điểm Business Activities, phân tích Revenue (Doanh thu) và Cost (Chi phí).
- Đối với chủ điểm Delivery Activities, phân tích Distance (Khoảng cách giao vận) và Time Delivery (Thời gian giao hàng).
- Đối với chủ điểm Orders, phân tích Purchase Orders (Đơn đặt hàng) và Delivery Orders (Đơn giao hàng).

2. Physical Data Model

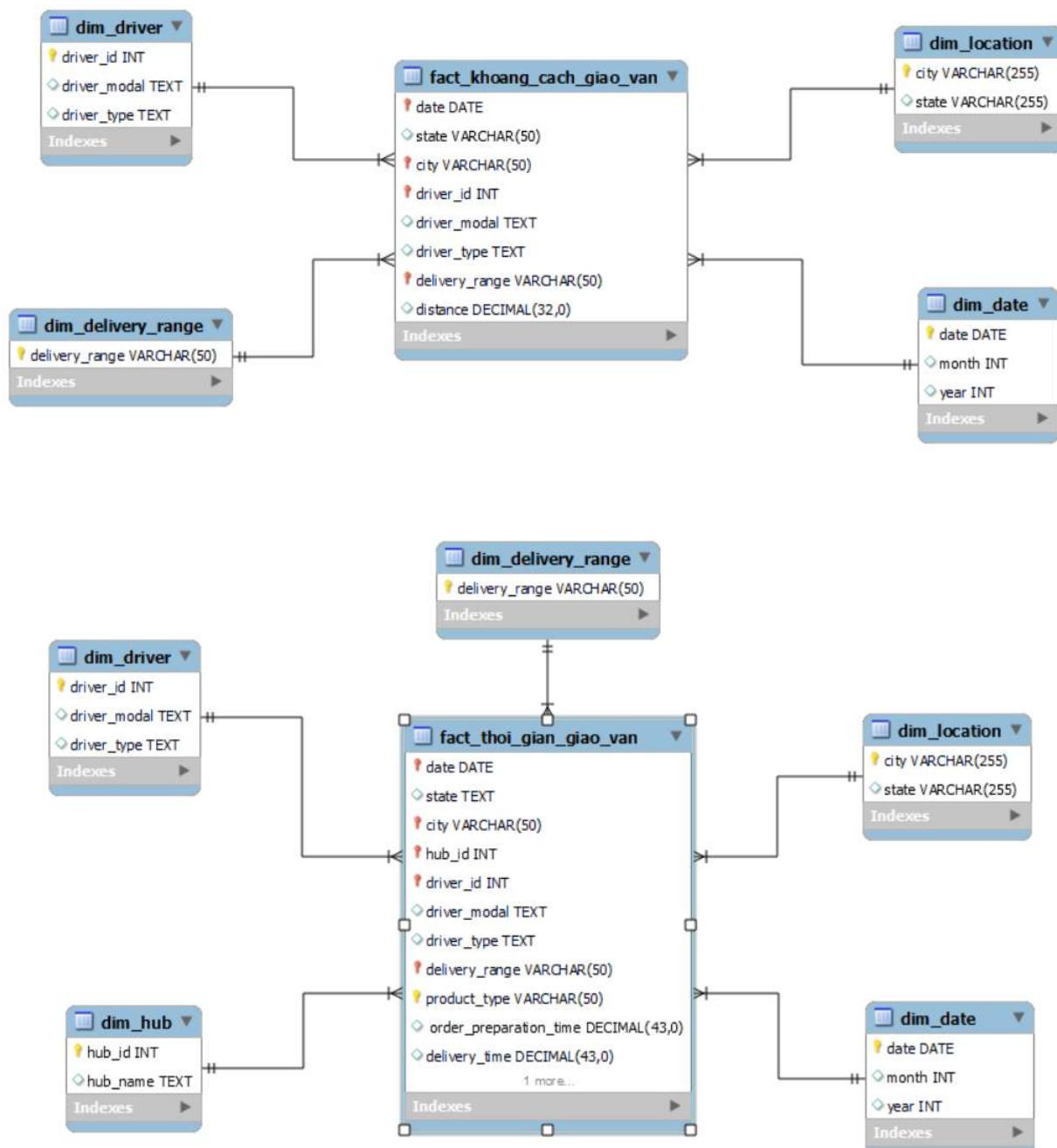
Từ mô hình logic, ta đưa dữ liệu về mô hình vật lý để lưu trữ trong cơ sở dữ liệu và sử dụng để phân tích.

Mô hình Physical Data Model như sau:

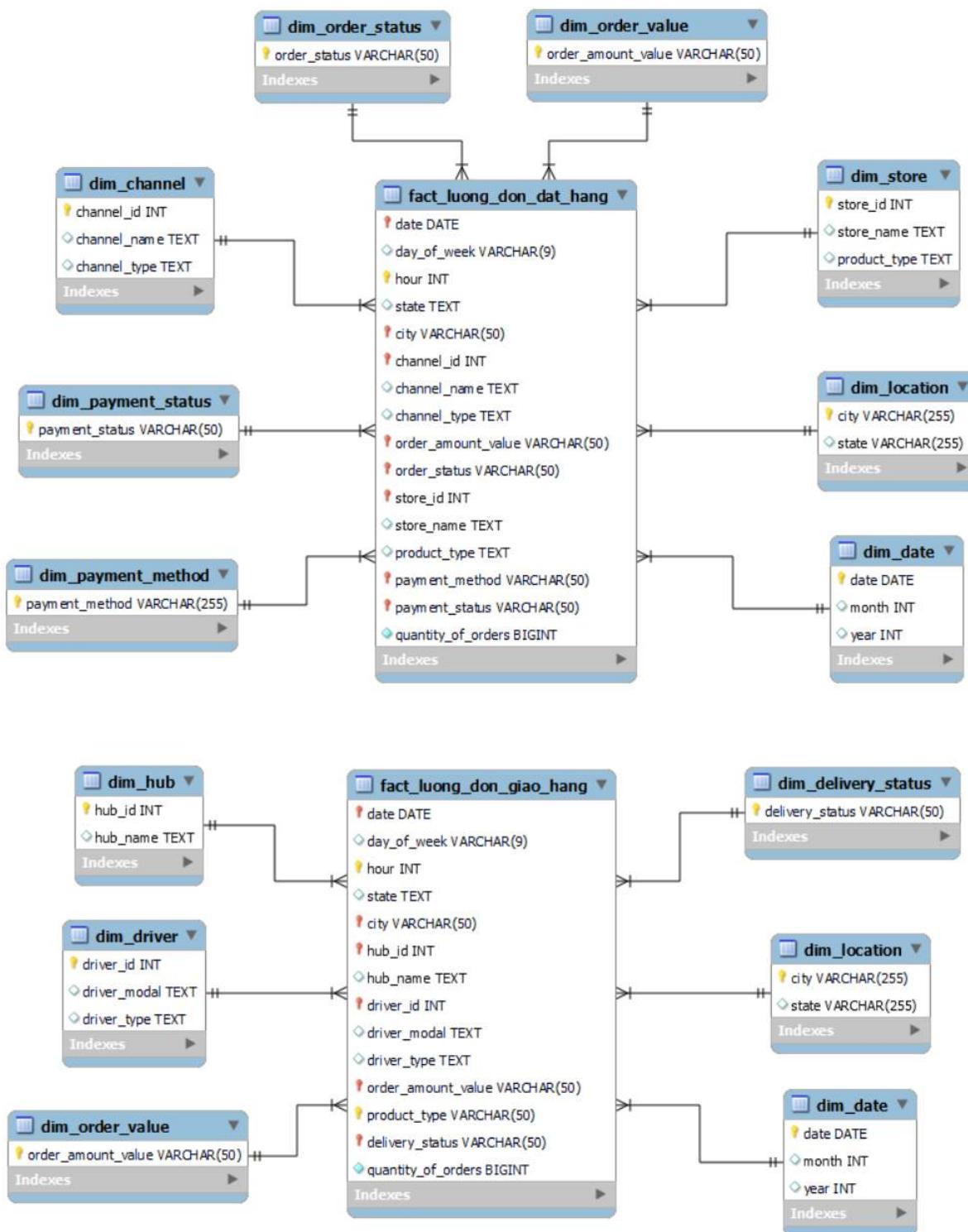
a. Business Activities - Hoạt động kinh doanh



b. Delivery Activities - Hoạt động giao vận



c. Orders - Đơn hàng



Mô hình dữ liệu gồm 9 bảng dim và 6 bảng fact. 9 bảng dim trong mô hình tương ứng với 9 dim đã nêu ở phần hệ thống chiềut khái niệm. 6 bảng fact gồm có:

- **fact _doanh _thu:** Chứa các thông tin về các nguồn thu của công ty, bao gồm dữ liệu của các channel, cửa hàng, cùng với dữ liệu về địa điểm và thời gian.
- **fact _chi _phi:** Chứa các thông tin về các đối tượng tạo ra chi phí của công ty, bao gồm dữ liệu về tài xế, hub, phạm vi giao vận, địa điểm và thời gian.

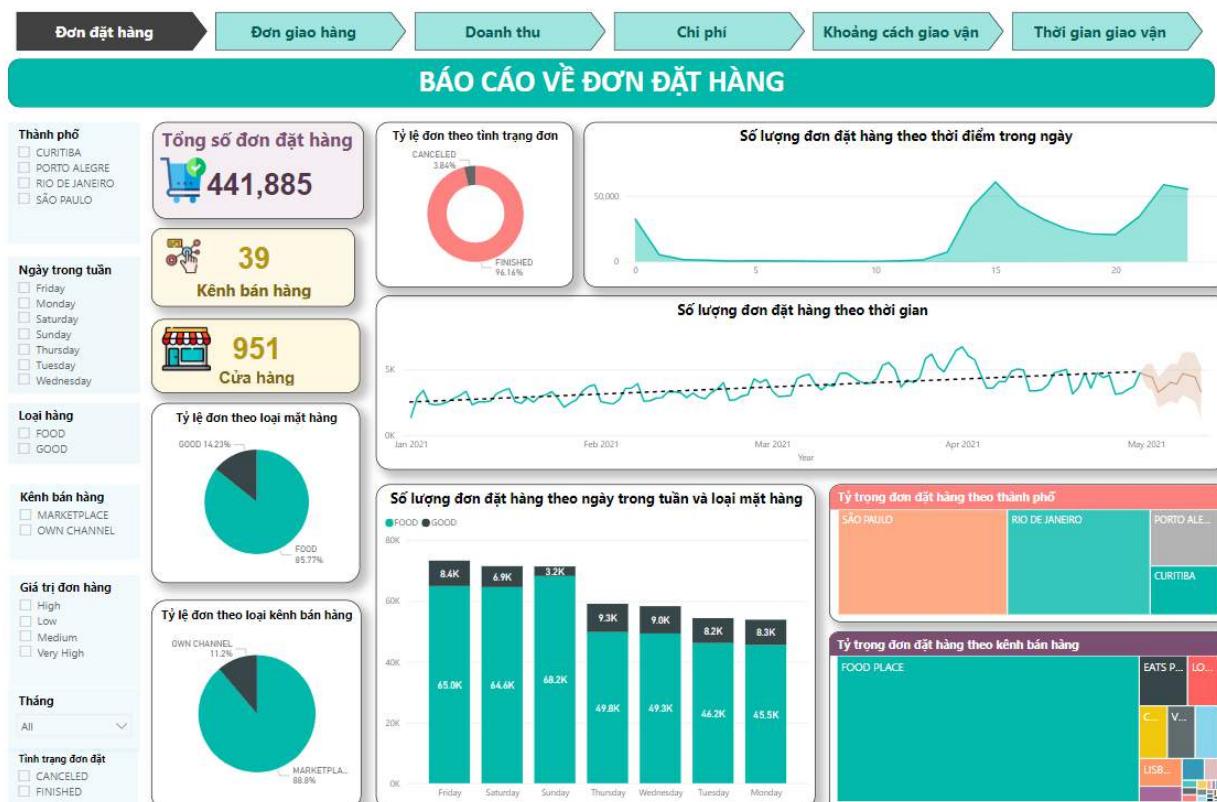
- **fact_khoang_cach_giao_van:** Chứa các thông tin liên quan tới khoảng cách giao vận của các đơn hàng, gồm dữ liệu về tài xế, phạm vi giao vận, địa điểm và thời gian.
- **fact_thoi_gian_giao_van:** Chứa các thông tin liên quan tới thời gian giao vận của các đơn hàng, gồm dữ liệu về tài xế, hub, phạm vi giao vận, địa điểm và thời gian.
- **fact_luong_don_dat_hang:** Chứa các thông tin liên quan tới các đơn đặt hàng đã thống kê được thông qua dữ liệu được thu thập, bao gồm dữ liệu về channel, cửa hàng, giá trị đơn hàng, trạng thái đơn hàng, phương thức thanh toán, trạng thái thanh toán, địa điểm và thời gian.
- **fact_luong_don_giao_hang:** Chứa các thông tin liên quan tới các đơn hàng đã giao và được thống kê thông qua dữ liệu thu thập được, bao gồm dữ liệu về các hub, tài xế, giá trị đơn hàng, trạng thái giao vận, địa điểm và thời gian.

3.3 Xây dựng báo cáo trực quan

Dựa trên nhu cầu của doanh nghiệp và quy mô bộ dữ liệu, chúng ta xây dựng được báo cáo trực quan về 5 chủ đề sau đây:

1. Báo cáo về đơn đặt hàng
2. Báo cáo về đơn giao hàng
3. Báo cáo về doanh thu
4. Báo cáo về chi phí
5. Báo cáo về khoảng cách giao vận
6. Báo cáo về thời gian giao vận

3.3.1 Báo cáo về đơn đặt hàng



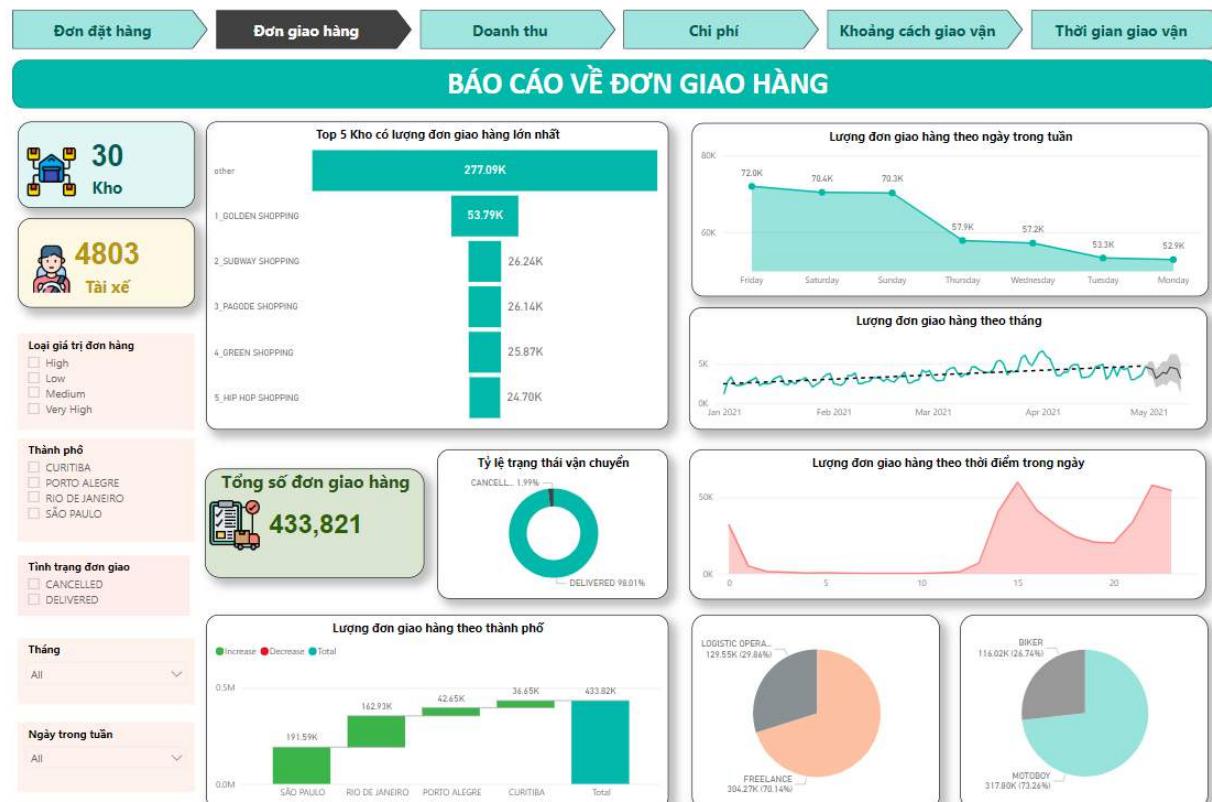
Hình 27: Báo cáo về đơn đặt hàng

- Tổng số đơn đặt hàng là 441,885 đơn và chủ yếu đến từ 2 thành phố São Paulo và Rio De Janeiro. Có thể giải thích cho sự chênh lệch này đó vì:
 - São Paulo là thành phố lớn nhất và trung tâm kinh tế quan trọng nhất của Brazil. Thành phố này đóng góp một phần lớn vào GDP của quốc gia, và là trung tâm thương mại, tài chính và công nghiệp quan trọng.
 - Vùng đô thị Rio de Janeiro có dân số lên đến 11.620.000 dân. Thành phố này nổi tiếng với phong cảnh tự nhiên, các lễ hội carnival và nhạc samba và các loại hình âm nhạc khác, các bãi biển.

2 thành phố có dân cư đông đúc, đời sống vật chất cao, nhu cầu mua sắm cũng vượt trội so với 2 thành phố còn lại.

- Thời điểm đặt hàng chủ yếu rơi vào khoảng thời gian buổi chiều từ khoảng 3h đến 6h chiều, thời điểm đi làm về (đồ thi đi lên, đạt đỉnh lúc 3h chiều và giảm dần đến gần 8 tối và tiếp tục tăng, chạm đỉnh vào 10h sáng sau đó giảm dần). Lượng đặt đơn từ 2h sáng đến 12h trưa gần như không đáng kể).
- Số lượng đơn về FOOD chủ yếu là 3 ngày cuối tuần, các ngày trong tuần ngang nhau và ít hơn hẳn so vs 3 ngày cuối tuần. GOOD thì ngược lại: Chủ nhật lại ít đơn hơn hẳn trong khi các ngày khác thì số lượng đơn không có nhiều sự khác biệt
- Tỉ trọng đơn hàng FOOD chiếm phần đa (85.77%), đồng thời ta cũng thấy kênh bán hàng liên quan đến thực phẩm luôn duy trì một tỷ trọng cao (FOOD PLACE), vượt trội hơn hẳn 38 kênh còn lại.
- Số lượng đơn tháng 1,2 khá ngang bằng nhau. Tháng 3 có sự tăng trưởng rõ rệt khi tăng thêm hơn 40k đơn. Tháng 4 bị chững lại và giảm đi 1 chút. Tuy nhiên dựa vào đường xu hướng, có thể thấy nhìn chung số lượng đơn đặt là tăng nhẹ và sẽ ổn định trong thời gian tới.
- Tỷ trọng các đơn hàng được đặt bởi Market Place chiếm đa số với hơn 88% và chủ yếu là bán thực phẩm. Tuy nhiên với mặt hàng là GOOD thì tỉ lệ được bán bởi Own Place lại tăng lên rõ rệt và tăng lên gần 50%

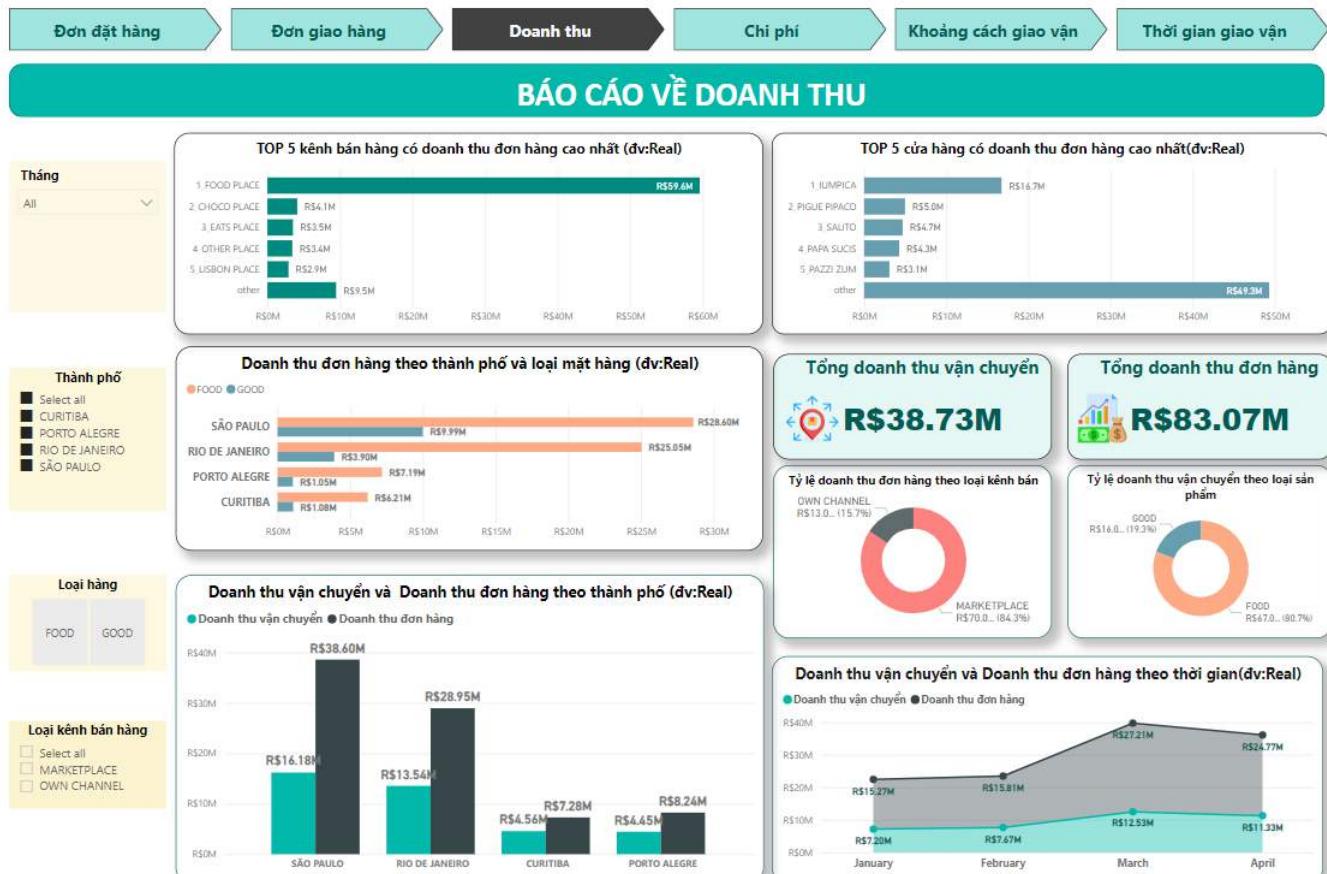
3.3.2 Báo cáo về đơn giao hàng



Hình 28: Báo cáo về đơn giao hàng

- São Paulo dẫn đầu về số lượng đơn hàng giao: Vì là thành phố lớn nhất và trung tâm kinh tế của Brazil, do đó lượng đơn hàng nhận cao hơn hẳn các thành phố khác. Điều này phản ánh nhu cầu tiêu thụ lớn và hoạt động kinh tế sôi động tại đây.
- Ngày cuối tuần có lượng đơn hàng cao: 3 ngày thứ 6-7-CN có lượng đơn hàng giao cao nhất, điều này có thể do lượng đơn đặt hàng đã phân tích ở trên tăng cao vào cuối tuần và tài xế tự do sẽ tận dụng thời gian này để giao nhiều hơn so với ngày trong tuần.
- Tỷ lệ giao thành công đạt 98.01%, cho thấy hiệu quả hoạt động giao hàng tốt và quản lý logistics hiệu quả.
- Phần lớn các đơn hàng được giao bởi các tài xế freelance (70.14%), điều này cho thấy sự phổ biến của mô hình làm việc tự do trong ngành giao hàng.
- Motoboy chiếm phần lớn (73.26%) các đơn giao hàng, cho thấy sự phổ biến và hiệu quả của xe máy có giỏ hàng trong giao vận tại các thành phố lớn. Biker là loại xe không có giỏ hàng và tài xế đều làm việc tự do (thuộc Freelance), do đó giao được ít đơn hơn.
- Trong top 5 Kho có lượng đơn giao hàng nhiều nhất, Golden Shopping có số lượng đơn vượt trội so với 4 kho còn lại. Có thể giải thích kho này nằm tại Rio de Janeiro, một vùng đô thị có nhu cầu mua sắm cao như đã phân tích ở trên.

3.3.3 Báo cáo về doanh thu

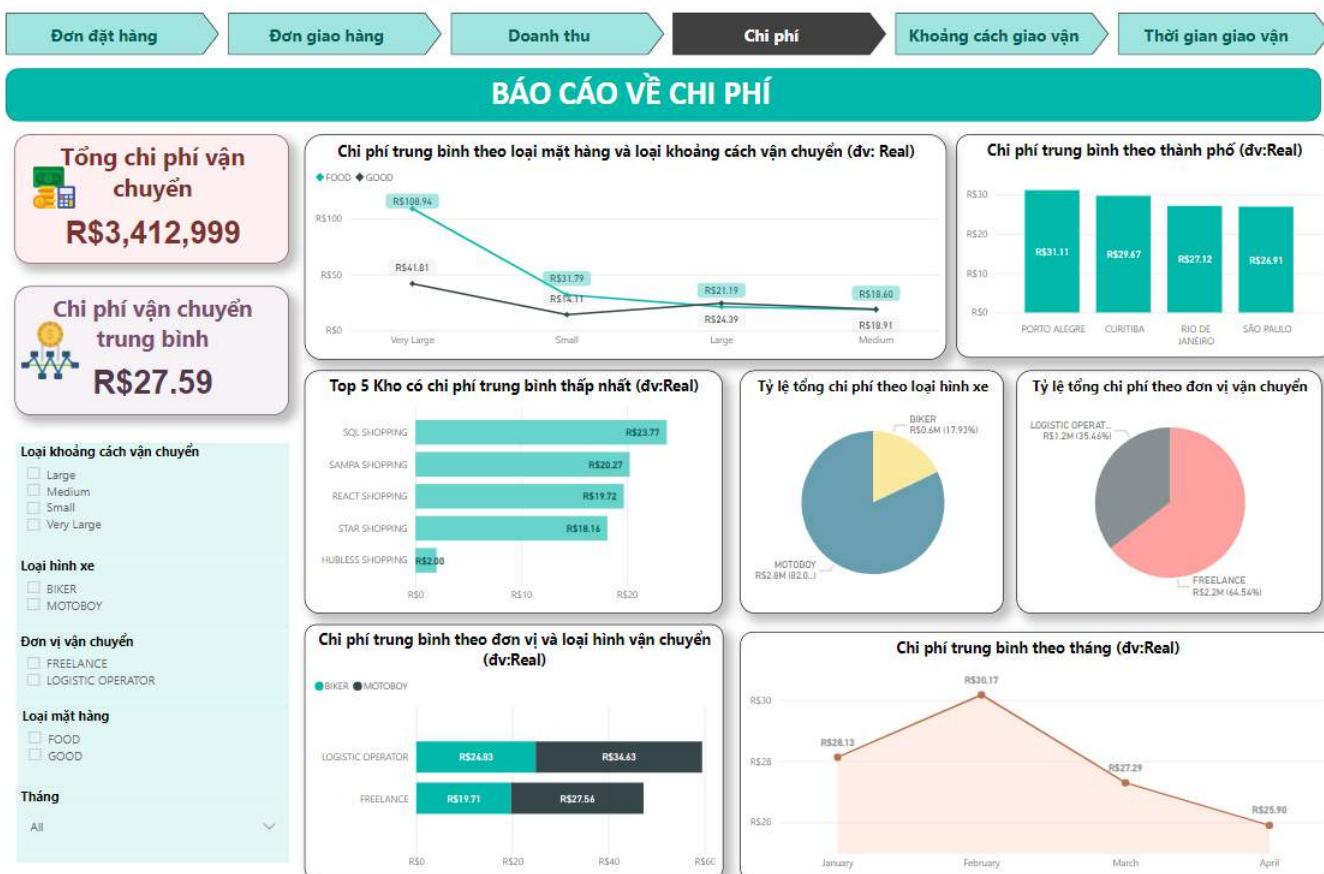


Hình 29: Báo cáo về doanh thu

- Tổng doanh thu vận chuyển là 38.73M (real) và doanh thu đơn hàng là 83.07M (real)

- Sử dụng hàm DAX để tính toán top 5 doanh thu và phần còn lại:
 - FOOD PLACE có doanh thu cao nhất, đứng top 1 kênh bán hàng và vượt trội rõ rệt, chứng tỏ sức hấp dẫn của thị trường thực phẩm cao hơn thị trường hàng hóa.
 - Cửa hàng top 1 là JUMPICA cũng có doanh thu lớn, vì đây là chuỗi cửa hàng phân bố chủ yếu ở 2 thành phố lớn São Paulo và Rio de Janeiro.
- Với dân số đông và hoạt động kinh tế mạnh, không có gì ngạc nhiên khi São Paulo có doanh thu cao nhất cả về giao hàng và đơn hàng. Đứng thứ 2 không có sự chênh lệch nhiều là Rio de Janeiro. Sự phát triển của kinh tế, văn hóa, du lịch giúp 2 thành phố này thu hút nhiều khách hàng.
- Các sản phẩm Food có tỷ lệ doanh thu vận chuyển cao hơn, cho thấy nhu cầu cao về giao thực phẩm, có thể là do đặc thù sản phẩm cần giao nhanh để đảm bảo chất lượng.
- Có sự tăng trưởng ổn định trong doanh thu vận chuyển và doanh thu đơn hàng qua các tháng từ tháng 1 đến tháng 4, đặc biệt tăng mạnh vào tháng 3 - thời điểm nhiều lễ hội, sự kiện của Brazil. Xu hướng trên có thể cho thấy sự tăng trưởng đáng kể trong nhu cầu tiêu dùng và sự cải thiện trong hệ thống logistics.
- Marketplace đóng góp phần lớn vào doanh thu, cho thấy người tiêu dùng ưa chuộng việc mua sắm qua các nền tảng này hơn so với các kênh bán hàng riêng lẻ (Own Channel).

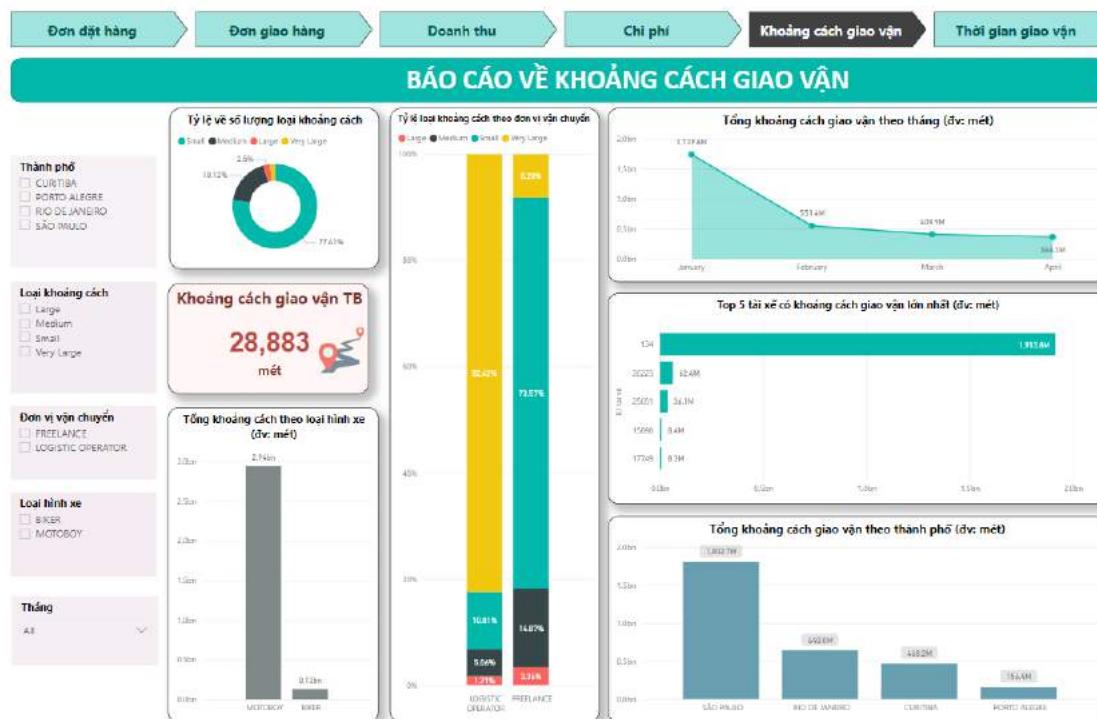
3.3.4 Báo cáo về chi phí



Hình 30: Báo cáo về chi phí

- Tổng chi phí vận chuyển là khoảng 3.4M (real) và chi phí vận chuyển trung bình là 27.59 (real)
- Trong 4 tháng đầu năm 2021 tại Brazil, tháng 2 có chi phí vận chuyển lớn nhất. Có thể giải thích bằng 1 số nguyên nhân thực tế:
 - Các cảng lớn như Santos (gần São Paulo) và Rio de Janeiro bị tắc nghẽn do lưu lượng hàng hóa tăng đột biến. Các biện pháp phòng chống COVID-19 tại các cảng cũng làm giảm hiệu quả hoạt động, dẫn đến tình trạng chậm trễ và tăng chi phí vận chuyển.
 - Giá dầu và nhiên liệu tăng mạnh trong giai đoạn này cũng là một yếu tố quan trọng dẫn đến việc tăng chi phí vận chuyển
 - Tháng 2 thường là mùa mưa ở nhiều khu vực của Brazil, đặc biệt là ở Rio de Janeiro và São Paulo. Điều kiện thời tiết xấu có thể gây ra các vấn đề trong vận chuyển hàng hóa, bao gồm việc chậm trễ và tăng chi phí bảo quản.
- Để tối ưu chi phí vận chuyển, tập trung nhiều hơn vào những kho có chi phí trung bình thấp nhất như SQL Shopping, Sampa Shopping,..
- Chi phí 2 loại hình vận chuyển Motoboy và Biker không chênh lệch nhiều. Motoboy là loại xe có giỏ hàng chuyên dụng nên có phí cao hơn Biker là xe máy bình thường.
- Chi phí giao hàng trung bình lớn nhất thuộc về thành phố Porto Alegre, nhỏ nhất là São Paulo nhưng không có sự cách biệt lớn. Trung tâm đã làm tốt việc ổn định chi phí vận chuyển tại các thành phố.
- Dối với loại khoảng cách Very Large thì chi phí vận chuyển cao, đặc biệt là với mặt hàng thực phẩm, cần vận chuyển nhanh để đảm bảo chất lượng.

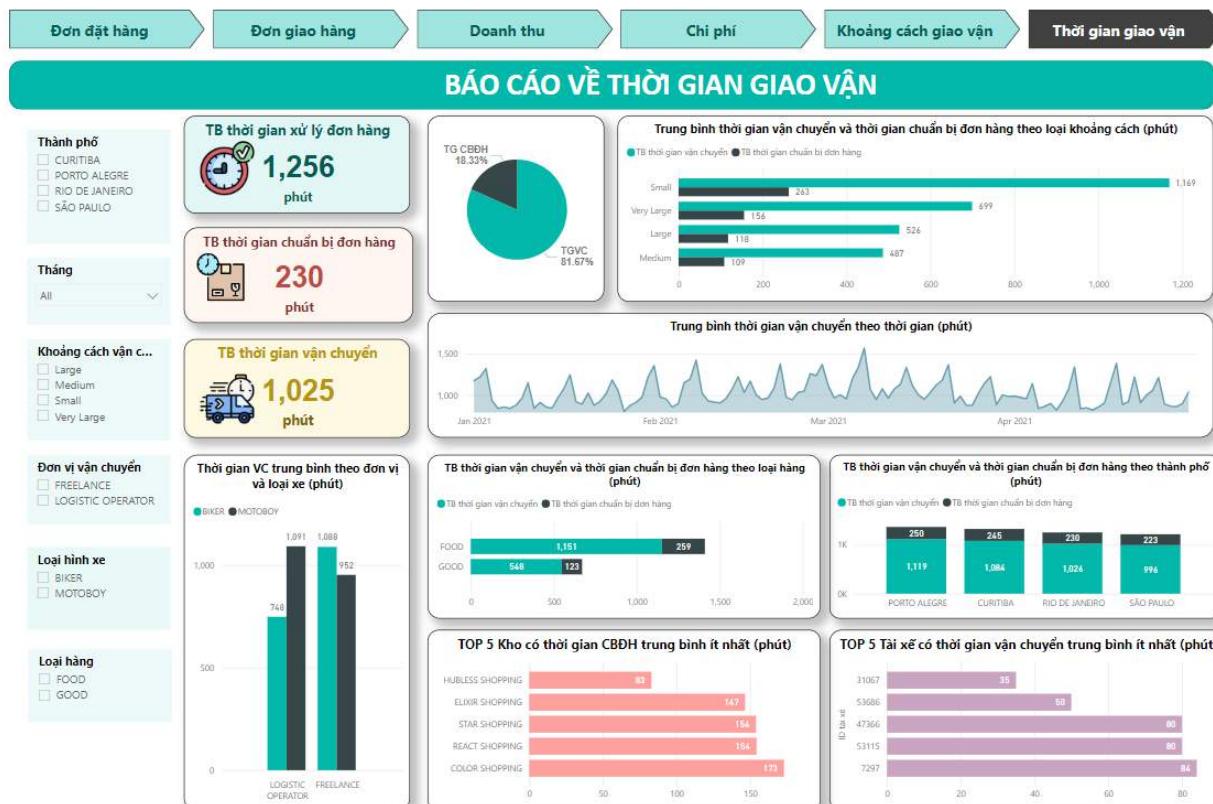
3.3.5 Báo cáo về khoảng cách giao vận



Hình 31: Báo cáo về khoảng cách giao vận

- Khoảng cách vận chuyển trung bình khoảng 29km, trong đó chủ yếu lượng đơn giao vận là loại khoảng cách ngắn (Small).
- Tổng khoảng cách vận chuyển đến São Paulo cao hơn hẳn so với các thành phố khác (Porto Alegre, Curitiba, và Rio de Janeiro) có thể được giải thích bởi một số lý do:
 - Nhu cầu vận chuyển hàng hóa đến thành phố lớn như São Paulo là rất lớn để phục vụ cho các hoạt động kinh tế đa dạng và phong phú.
 - São Paulo có hệ thống cơ sở hạ tầng và giao thông phát triển hơn so với nhiều thành phố khác ở Brazil. Điều này làm cho việc vận chuyển hàng hóa đến và đi từ São Paulo trở nên thuận lợi và hiệu quả hơn, thu hút nhiều hoạt động vận chuyển.
- Tài xế có ID 134 giao hàng nhiều nhất với tổng khoảng cách cao vượt trội, có thể giải thích từ biểu đồ vì người này thuộc đơn vị Logistic Operator và chủ yếu giao loại khoảng cách Very Large.
- Khoảng cách giao hàng được chia thành 4 loại, Small là khoảng 10km, với khoảng cách ngắn này thì việc giao hàng bằng moto hay là bike vẫn hợp lý, nhưng khi khoảng cách từ trung bình trở nên dài thì việc giao hàng bằng motorboy hay biker gần như là bất khả thi, công ty nên có thêm những phương thức giao hàng khác như là ô tô hoặc tàu hỏa, máy bay để giảm thời gian nhận hàng.
- Tháng 1 có tổng khoảng cách giao vận là lớn nhất, có thể vì các đơn hàng bị trì hoãn từ tháng 12 do quá tải mùa lễ có thể được giao vào tháng 1, làm tăng tổng khoảng cách giao vận trong tháng này.

3.3.6 Báo cáo về thời gian giao vận



Hình 32: Báo cáo về thời gian giao vận

- Thời gian xử lý đơn hàng trung bình là 1,256 phút, trong đó 18,33% là thời gian chuẩn bị đơn hàng và 81,67% là thời gian vận chuyển.
- Từ biểu đồ ta có thể thấy top 5 Kho có thời gian chuẩn bị đơn hàng trung bình thấp nhất và top 5 tài xế có thời gian giao vận trung bình thấp nhất. Trung tâm có thể đầu tư vào những đơn vị này để nâng cao sự hài lòng của khách hàng.
- Các đơn vị vận chuyển tự do (Freelance) có thể không được tổ chức và quản lý chặt chẽ như các công ty logistics, dẫn đến thời gian vận chuyển dài hơn với loại hình Biker. Khi được quản lý như trong công ty Logistic xe Motorboy có thể gấp nhiều hạn chế về tốc độ và khả năng vận chuyển so với Biker, nên có thời gian vận chuyển cao hơn.
- Thời gian vận chuyển trung bình theo thời gian từ tháng 1/2021 đến 4/2021 có sự biến động ổn định.
- Thời gian xử lý đơn hàng trung bình tại cả 4 thành phố không chênh lệch nhiều. Trung tâm cần duy trì và phát huy tình trạng này trong tương lai để đảm bảo sự hài lòng của khách hàng ở tất cả địa điểm.
- Hàng hóa thường không yêu cầu điều kiện vận chuyển đặc biệt như thực phẩm, do đó thời gian vận chuyển ngắn hơn. Hàng hóa có thể được lưu trữ và vận chuyển dễ dàng hơn, góp phần giảm thời gian vận chuyển.
- Thời gian chuẩn bị đơn hàng giảm dần khi khoảng cách vận chuyển tăng lên, điều này có thể do quy trình chuẩn bị được tối ưu hóa hơn cho các đơn hàng cần vận chuyển xa. Thời gian vận chuyển tăng dần theo khoảng cách, điều này hoàn toàn hợp lý do khoảng cách vận chuyển lớn hơn sẽ cần nhiều thời gian hơn. Tuy nhiên với loại khoảng cách Small lại có thời gian vận chuyển cao, có thể do các yếu tố như tắc nghẽn giao thông hoặc quy trình giao hàng tại các địa điểm gần nhau có thể bị phức tạp.

3.4 Tổng kết

3.4.1 Những nội dung đã thực hiện được

- Thực hiện Khảo sát về quy trình nghiệp vụ bài toán, mô hình kinh doanh, trình bày cây phân tích.
- Hệ thống, đánh giá quy mô bộ dữ liệu. Trình bày ERD OLTP, Data Flow.
- Khám phá dữ liệu, trình bày kiến trúc Data Warehouse, Data Pipeline ETL Job, thực hiện các nội dung ETL dữ liệu (sử dụng đầy đủ và đa dạng các công cụ), xây dựng hệ thống lập lịch ETL tự động.
- Trình bày chiều hệ thống khái niệm, xây dựng Logical Data Model, biến đổi OLTP -> OLAP, Physical Data Model.
- Xây dựng đa dạng Dashboard, đưa ra các phân tích dữ liệu.

3.4.2 Những hạn chế cần khắc phục

- Bộ dữ liệu cũ (từ năm 2021), dữ liệu chưa đa dạng và còn hạn chế, đặc biệt về mặt thời gian (chỉ có 4 tháng đầu năm 2021).
- Chưa có thông tin về khách hàng, khó phân tích được khuynh hướng đặt hàng và hiệu quả hoạt động.

- Phân tích tổng quan bài toán trung tâm giao vận, chưa đi sâu vào chi tiết hoạt động giao vận, vận tải.
- Mô hình Data chưa tối ưu với bài toán đưa ra.

3.4.3 Hướng phát triển và bài học rút ra

- Xây dựng mô hình Machine Learning và các thuật toán máy học dự báo các chỉ số liên quan tới hoạt động giao vận, hoạt động bán hàng, số lượng đơn hàng,...
- Cần tìm hiểu kỹ càng bộ dữ liệu, đánh giá sâu hơn về các chỉ số để tránh xảy ra sai sót và mất mát dữ liệu.
- Sử dụng đa dạng hơn các tính năng Visualization.
- Xây dựng mô hình cần đủ các chức năng, nhiệm vụ của hệ thống hơn.

Kết luận

Qua bài báo cáo, chúng em đã phần nào hiểu được tầm quan trọng của kho dữ liệu và kinh doanh thông minh trong môi trường kinh doanh ngày nay. Chúng em đã được trải nghiệm và áp dụng các công cụ, kỹ thuật để xây dựng và quản lý kho dữ liệu một cách hiệu quả, từ đó khai thác dữ liệu để đưa ra những quyết định chiến lược có tính toán và đáp ứng nhu cầu thực tiễn của doanh nghiệp.

Dưới sự hướng dẫn và hỗ trợ của thầy Nguyễn Danh Tú, chúng em đã không chỉ học hỏi được những kiến thức lý thuyết về kho dữ liệu và kinh doanh thông minh mà còn có cơ hội áp dụng chúng vào thực tiễn một cách có hiệu quả. Qua dự án với chủ đề "Transportation", chúng em đã áp dụng các công cụ và kỹ thuật phân tích dữ liệu để hiểu sâu hơn về hoạt động và thị trường trong ngành vận tải.

Cuối cùng, chúng em xin cảm ơn Thầy Nguyễn Danh Tú đã hỗ trợ và hướng dẫn chúng em suốt quá trình học tập và nghiên cứu về môn học "Kho dữ liệu và kinh doanh thông minh". Thầy không chỉ chia sẻ những kiến thức sâu rộng mà còn giúp đỡ chúng em áp dụng những kiến thức đó vào thực tiễn một cách hiệu quả nhất. Chúng em hy vọng sẽ tiếp tục được đồng hành với thầy trong những hành trình học tập và nghiên cứu trong tương lai.

Tài liệu

- [1] Thầy Nguyễn Danh Tú, (2022), Bài giảng "*Kho dữ liệu và kinh doanh thông minh*", Khoa Toán - Tin, Đại học Bách khoa Hà Nội.
- [2] Thầy Nguyễn Danh Tú, Kênh Youtube *Learn Excel*, BKIndex Group, Đại học Bách khoa Hà Nội.