ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

-ꕥ-

BÁO CÁO HỌC PHẦN MÔN HỌC
NHẬP MÔN KHOA HỌC DỮ LIỆU

Nghiên cứu xây dựng nền tảng tích hợp dữ liệu và trục liên thông dữ liệu thời gian thực
(6.4.2)

ĐỀ TÀI: "Nghiên cứu xây dựng module trích xuất – chuyển đổi – nạp dữ liệu ETL"

Trưởng nhóm	Sinh viên thực hiện
Nguyễn Huy Hoàng	Mai Minh Khôi Trần Đức Kiên Phạm Đức Lưu

Năm 2025

DANH MỤC CÁC CÔNG VIỆC

Nghiên cứu xây dựng module trích xuất – chuyển đổi – nạp dữ liệu ETL

STT	Tên nội dung công việc	Người thực hiện
1	6.4.2. Báo cáo kết quả nghiên cứu xây dựng module trích xuất – chuyển đổi – nạp dữ liệu ETL	Nguyễn Huy Hoàng Mai Minh Khôi Trần Đức Kiên Phạm Đức Lưu

MỤC LỤC

DANH MỤC HÌNH ẢNH **4**
DANH MỤC BẢNG BIỂU **5**
MỞ ĐẦU **6**
CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG ETL **8**
1.1. Định nghĩa 8
1.2. Lịch sử ra đời và phát triển 9
1.3. Ứng dụng của quy trình ETL 10
1.4. Ưu, nhược điểm của ETL 11
CHƯƠNG 2: KIẾN TRÚC VÀ CÁC THÀNH PHẦN CỦA ETL **13**
2.1. Kiến trúc tổng quan của quy trình ETL. 13
2.2. Extract – Trích xuất dữ liệu. 13
2.3. Transform – Chuyển đổi dữ liệu 16
2.4. Load – Tải dữ liệu 19
CHƯƠNG 3: DEMO XÂY DỰNG MODULE ETL **22**
3.1 Tìm hiểu tập dataset. 22
3.2 Thiết kế hệ thống ETL. 23
3.3 Xây dựng hệ thống 23
KẾT LUẬN **32**
TÀI LIỆU THAM KHẢO **33**

DANH MỤC HÌNH ẢNH

Hình 1: Quy trình ETL.....13
Hình 2: Định nghĩa các field của dataset.....22
Hình 3: Tổng quan hệ thống ETL.....23
Hình 4: Code DAG Bronze Layer.....24
Hình 5: Data Bronze Layer.....24
Hình 6: Code DAG Silver Layer.....25
Hình 7: Data Silver Layer.....26
Hình 8: Lược đồ Star Schema của Data Warehouse.....27
Hình 9: Code DAG tạo bảng dim_date.....28
Hình 10: Code DAG tạo bảng fact.....29
Hình 11: Kết quả chạy các DAG.....29
Hình 12: Data Gold Layer của bảng fact.....30
Hình 13: Looker Studio Report.....31

DANH MỤC BẢNG BIỂU

Bảng 1: So sánh các kỹ thuật trích xuất dữ liệu.....15

MỞ ĐẦU

Trong thời đại dữ liệu bùng nổ, các tổ chức và doanh nghiệp không chỉ đối mặt với khối lượng dữ liệu khổng lồ mà còn phải xử lý thông tin từ nhiều nguồn khác nhau với định dạng đa dạng. Để khai thác tối đa giá trị của dữ liệu, các hệ thống quản lý dữ liệu hiện đại cần có khả năng thu thập, tổng hợp, làm sạch và chuyển đổi dữ liệu một cách hiệu quả trước khi đưa vào kho lưu trữ phục vụ phân tích và ra quyết định. Đây chính là vai trò quan trọng của **hệ thống ETL (Extract - Transform - Load)**.

ETL là một quy trình quan trọng trong lĩnh vực xử lý dữ liệu, giúp trích xuất dữ liệu từ nhiều nguồn khác nhau, chuyển đổi dữ liệu theo yêu cầu nghiệp vụ và nạp vào hệ thống lưu trữ trung tâm như Data Warehouse hay Data Lake. Một hệ thống ETL được thiết kế tốt không chỉ đảm bảo tính chính xác, toàn vẹn của dữ liệu mà còn tối ưu hiệu suất xử lý, hỗ trợ doanh nghiệp khai thác dữ liệu một cách nhanh chóng và hiệu quả.

Báo cáo này sẽ tập trung vào nghiên cứu quá trình xây dựng một hệ thống ETL hoàn chỉnh, từ việc lựa chọn công nghệ phù hợp, thiết kế quy trình xử lý, đến triển khai thực tế trên một nền tảng cụ thể. Ngoài ra, báo cáo cũng đánh giá các thách thức thường gặp trong quá trình phát triển hệ thống ETL như xử lý dữ liệu lớn (big data), tối ưu hiệu suất, đảm bảo tính chính xác và bảo mật dữ liệu. Thông qua nghiên cứu này, chúng tôi mong muốn đưa ra các giải pháp tối ưu nhằm nâng cao hiệu quả của hệ thống ETL, giúp doanh nghiệp khai thác dữ liệu một cách có hệ thống và chính xác, từ đó hỗ trợ quá trình ra quyết định chiến lược.

Báo cáo sẽ bao gồm các nội dung sau:

- Chương 1: Tổng quan về module trích xuất – chuyển đổi – nạp dữ liệu ETL. Chương này sẽ giới thiệu chung những lý thuyết cơ bản về ETL, bao gồm: các khái niệm, lịch sử ra đời và phát triển, chức năng, ứng dụng của quy trình ETL.

- Chương 2: Kiến trúc và các thành phần của một hệ thống ETL. Trong chương này, sẽ tập trung vào tìm hiểu chi tiết các thành phần của ETL, các bước xây dựng hệ thống ETL, những vấn đề gặp phải với mỗi thành phần cụ thể trong một hệ thống ETL.

- Chương 3: Demo xây dựng module trích xuất – chuyển đổi – nạp dữ liệu ETL. Chương cuối của báo cáo trình bày kết quả thử nghiệm xây dựng module trích xuất – chuyển đổi – nạp dữ liệu ETL của nhóm thực hiện sau khi áp dụng phần lý thuyết đã nêu trong hai chương trước.

Phần kết luận và tài liệu tham khảo.

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG ETL

1.1. Định nghĩa

Quy trình ETL viết tắt của Extract, Transform, và Load (Trích xuất - biến đổi - tải) là một quy trình tiêu chuẩn giúp xử lý dữ liệu từ nhiều nguồn khác nhau, chuyển đổi chúng thành định dạng phù hợp và tải vào hệ thống lưu trữ trung tâm như Data Warehouse. Quy trình này không chỉ giúp hợp nhất dữ liệu từ nhiều hệ thống mà còn đảm bảo dữ liệu được làm sạch, chuẩn hóa và tối ưu hóa trước khi phục vụ mục đích phân tích.

ETL bao gồm ba bước chính, bao gồm:

a) Trích xuất (Extract): Đây là bước đầu tiên trong quy trình ETL, nơi dữ liệu được thu thập từ nhiều nguồn khác nhau, chẳng hạn như hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS), ứng dụng doanh nghiệp, dịch vụ web hoặc thậm chí là tệp dữ liệu phi cấu trúc. Quá trình trích xuất cần đảm bảo dữ liệu được lấy đầy đủ, chính xác và không gây ảnh hưởng đến hiệu suất của hệ thống nguồn.

b) Biến đổi (Transform): Sau khi dữ liệu được trích xuất, nó cần được chuyển đổi để phù hợp với mô hình dữ liệu của hệ thống đích. Giai đoạn này có thể bao gồm nhiều hoạt động như: chuẩn hóa lại dữ liệu, tính toán lại giá trị, làm sạch dữ liệu, …

c) Tải dữ liệu (Load): Sau khi dữ liệu đã được biến đổi, nó được ghi vào cơ sở dữ liệu đích, chuẩn bị cho việc truy vấn và phân tích. Quá trình này thường là quá trình cuối cùng trong quy trình ETL và có thể thực hiện thông qua việc sao chép dữ liệu trực tiếp vào cơ sở dữ liệu đích trước khi thực hiện bất kỳ biến đổi nào.

Nói chung, ETL là quá trình quan trọng để di chuyển dữ liệu từ nguồn đến đích và đảm bảo rằng dữ liệu đó đã được xử lý và chuẩn bị tốt trước khi sử dụng. Bằng cách tự động hóa quy trình ETL, doanh nghiệp có thể tối ưu hóa luồng dữ liệu, giảm thời gian xử lý thủ công và nâng cao hiệu quả trong phân tích dữ liệu. Ngoài ra, sự phát

triển của các công nghệ ETL hiện đại còn giúp mở rộng khả năng xử lý dữ liệu lớn (Big Data) và tích hợp với các hệ thống phân tán, giúp doanh nghiệp khai thác tối đa giá trị của dữ liệu trong thời đại kỹ thuật số.

1.2. Lịch sử ra đời và phát triển

+ Những năm 1970 và 1980: ETL xuất hiện trong bối cảnh sự phát triển của các hệ thống quản lý cơ sở dữ liệu (DBMS) và hệ thống thông tin doanh nghiệp (Enterprise Information Systems). Trong giai đoạn này, ETL chủ yếu được sử dụng để chuyển dữ liệu từ hệ thống cơ sở dữ liệu truyền thống sang hệ thống Data Warehouse mới nổi.

+ Những năm 1990: Vào thập kỷ này, sự phát triển của công nghệ và xu hướng sử dụng Data Warehouse để hỗ trợ quyết định doanh nghiệp đã thúc đẩy sự phát triển của ETL. Các công ty đã bắt đầu nhận ra giá trị của việc thu thập, biến đổi và tải dữ liệu từ nhiều nguồn khác nhau để phân tích và báo cáo.

+ Những năm 2000 và sau này: Trong thế kỷ này, ETL đã trở thành một phần quan trọng của ngành công nghiệp dữ liệu và phân tích. Các công ty và tổ chức đã sử dụng các công cụ và nền tảng ETL mạnh mẽ để xử lý và quản lý dữ liệu lớn từ nhiều nguồn, bao gồm dữ liệu từ các ứng dụng doanh nghiệp, cơ sở dữ liệu trực tuyến, tệp văn bản, và nhiều nguồn dữ liệu khác.

+ Những năm 2010 đến nay: Sự phát triển của Big Data và công nghệ điện toán đám mây đã thúc đẩy sự phát triển của ETL để xử lý dữ liệu trên quy mô lớn hơn. Các công cụ ETL ngày càng trở nên linh hoạt và có khả năng tích hợp với các nền tảng dữ liệu và công nghệ mới.

ETL vẫn là một phần quan trọng của quy trình xử lý dữ liệu và phân tích dữ liệu. Trong tương lai, ETL có thể liên quan chặt chẽ đến các công nghệ mới như trí tuệ

nhân tạo (AI) và học máy để tạo ra các quy trình xử lý dữ liệu tự động và thông minh hơn.

1.3. Ứng dụng của quy trình ETL

ETL được sử dụng rộng rãi trong nhiều lĩnh vực và hệ thống:

a) Thương mại điện tử (E-commerce):

+ **Phân tích hành vi khách hàng**: ETL trích xuất dữ liệu từ website, ứng dụng di động, CRM để theo dõi hành vi người mua, từ đó cá nhân hóa đề xuất sản phẩm.

+ **Dự báo hàng tồn kho**: Kết hợp dữ liệu bán hàng và kho hàng để dự báo nhu cầu, tối ưu hóa tồn kho.

+ **Phát hiện gian lận**: ETL giúp xử lý dữ liệu giao dịch theo thời gian thực để phát hiện các hành vi gian lận trong thanh toán online.

b) Ngân hàng, tài chính

+ **Phân tích rủi ro và gian lận**: ETL kết hợp dữ liệu từ nhiều nguồn (giao dịch, lịch sử tín dụng, dữ liệu khách hàng) để phát hiện giao dịch bất thường.

+ **Hệ thống báo cáo tài chính**: Dữ liệu từ các chi nhánh, ATM, giao dịch online được tổng hợp để tạo báo cáo tài chính chính xác.

+ **Tùy chỉnh sản phẩm tài chính**: Ngân hàng có thể phân tích hành vi giao dịch của khách hàng để gợi ý các khoản vay hoặc chương trình tiết kiệm phù hợp.

c) Giao thông, Logistics

+ **Tối ưu hóa tuyến đường giao hàng**: ETL thu thập dữ liệu GPS, đơn hàng, thời gian vận chuyển để tối ưu hóa tuyến đường.

+ **Dự đoán thời gian giao hàng**: Xử lý dữ liệu thời tiết, lưu lượng giao thông để cải thiện độ chính xác của thời gian giao hàng.

+ **Quản lý chuỗi cung ứng**: Hợp nhất dữ liệu từ các nhà cung cấp, kho hàng để dự đoán nhu cầu và tối ưu chuỗi cung ứng.

d) Y tế, chăm sóc sức khỏe

+ Hồ sơ bệnh án điện tử (EHR): ETL giúp hợp nhất dữ liệu từ nhiều bệnh viện, phòng khám để xây dựng hồ sơ sức khỏe điện tử cho bệnh nhân.

+ Dự đoán bệnh tật: ETL trích xuất và làm sạch dữ liệu từ xét nghiệm, đơn thuốc, cảm biến y tế để hỗ trợ chẩn đoán bệnh.

+ **Tối ưu hóa quản lý bệnh viện**: Tổng hợp dữ liệu về bệnh nhân, lịch trình bác sĩ, phòng bệnh để tối ưu hóa vận hành bệnh viện.

e) Giải trí , truyền thông

+ Cá nhân hóa nội dung: Netflix, YouTube sử dụng ETL để phân tích lịch sử xem và đề xuất nội dung phù hợp.

+ **Đo lường hiệu suất quảng cáo**: ETL giúp tổng hợp dữ liệu từ nhiều kênh quảng cáo để đánh giá hiệu quả chiến dịch.

+ **Phân tích xu hướng người dùng**: Xử lý dữ liệu mạng xã hội, lượt xem để dự đoán xu hướng nội dung hot.

1.4. Ưu, nhược điểm của ETL

a) Ưu điểm

+ Kiểm soát chất lượng tốt hơn: Chuyển đổi dữ liệu trước khi tải lên hệ thống, đảm bảo chất lượng dữ liệu từ đầu.

+ Bảo mật cao: Phù hợp với các hệ thống yêu cầu bảo mật cao.

+ Tăng tốc độ truy vấn và báo cáo: Dữ liệu sau ETL được lưu trữ dưới dạng có tổ chức, tối ưu cho việc phân tích, giúp tạo báo cáo nhanh hơn và chính xác hơn.

+ Tự động hóa và giảm công sức xử lý thủ công: ETL có thể tự động hóa với các công cụ như Airflow, AWS Glue, GCP Dataflow, giúp doanh nghiệp tiết kiệm thời gian xử lý dữ liệu.

+ Hỗ trợ xử lý dữ liệu lớn (Big Data)

b) Nhược điểm

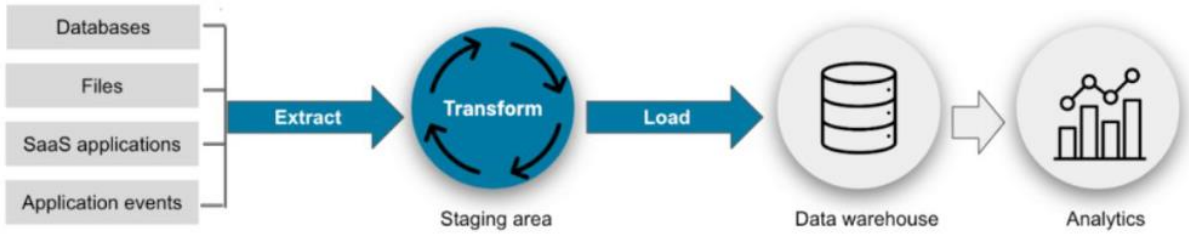
+ Thời gian xử lý lâu: Chuyển đổi dữ liệu trước khi tải kéo dài thời gian xử lý.

+ Chi phí cao: Yêu cầu đầu tư lớn cho hạ tầng và tài nguyên.

CHƯƠNG 2: KIẾN TRÚC VÀ CÁC THÀNH PHẦN CỦA ETL

2.1. Kiến trúc tổng quan của quy trình ETL.

Như định nghĩa đã trình bày ở trên, quy trình ETL bao gồm ba giai đoạn chính: Extract/Trích xuất, Transform/Chuyển đổi, Load/Tải dữ liệu, giúp đưa dữ liệu từ nhiều nguồn khác nhau vào hệ thống lưu trữ tập trung như Data Warehouse hay Data Lake.



Hình 1: Quy trình ETL

2.2. Extract – Trích xuất dữ liệu.

Đây là giai đoạn đầu tiên trong ETL, tập trung vào việc trích xuất dữ liệu từ các hệ thống nguồn. Hầu hết doanh nghiệp không chỉ làm việc với một loại dữ liệu hoặc một hệ thống duy nhất. Thay vào đó, họ quản lý thông tin từ nhiều nguồn và sử dụng các công cụ phân tích để tối ưu hóa quá trình quản trị. Để dữ liệu được chuyển đến một đích mới, trước tiên nó phải được trích xuất từ các nguồn.

Ở bước đầu tiên này, dữ liệu trích xuất được sẽ được nhập và hợp nhất vào một kho lưu trữ duy nhất. Dữ liệu thô có thể được trích xuất từ nhiều nguồn khác nhau, ví dụ như:

+ Cơ sở dữ liệu hiện có
+ Dữ liệu từ file – CSV, JSON, XML, Excel
+ Ứng dụng bán hàng, tiếp thị
+ Hệ thống quản lý khách hàng CRM
+ Kho dữ liệu

+ Công cụ phân tích

…

Dữ liệu có thể được xử lý thủ công, nhưng việc tự viết mã để trích xuất không chỉ mất nhiều thời gian mà còn tiềm ẩn nguy cơ sai sót. Các công cụ ETL giúp tự động hóa quá trình này, đảm bảo quy trình làm việc diễn ra nhanh chóng, chính xác và đáng tin cậy hơn.

Tần suất truyền dữ liệu từ nguồn đến kho lưu trữ phụ thuộc vào phương thức thu thập và cập nhật thay đổi. Quá trình trích xuất dữ liệu thường được thực hiện theo ba cách chính sau:

a) Trích xuất hoàn toàn (Full extraction)

Trích xuất toàn bộ dữ liệu từ nguồn mỗi lần chạy ETL

- Ưu điểm:

  + Dễ triển khai, không cần theo dõi thay đổi.

  + Phù hợp với dữ liệu nhỏ hoặc hệ thống không hỗ trợ incremental.

- Nhược điểm:

  + Tốn tài nguyên, thời gian và băng thông.

  + Không phù hợp với dữ liệu lớn.

b) Trích xuất gia tăng (Incremental extraction)

Chỉ trích xuất dữ liệu mới hoặc thay đổi kể từ lần trích xuất trước.

- Ưu điểm:

  + Giảm tải cho hệ thống, tiết kiệm tài nguyên.

  + Xử lý dữ liệu nhanh hơn so với Full Extraction.

- Nhược điểm:

  + Cần có cơ chế theo dõi thay đổi.

  + Nếu dữ liệu bị thiếu hoặc lỗi trong lần trích xuất trước, có thể gây mất dữ liệu.

c) Trích xuất dựa trên phát hiện thay đổi (Change Data Capture - CDC)

Phát hiện và trích xuất chỉ những thay đổi trong dữ liệu nguồn (chèn, sửa, xóa)

- Ưu điểm:

  + Hiệu suất cao nhất, không cần quét toàn bộ dữ liệu.

  + Hỗ trợ xử lý dữ liệu theo thời gian thực.

- Nhược điểm:

  + Phụ thuộc vào khả năng hỗ trợ của hệ thống nguồn.

  + Cấu hình phức tạp

So sánh các kỹ thuật trích xuất dữ liệu:

Bảng 1: So sánh các kỹ thuật trích xuất dữ liệu

Kỹ thuật	Dữ liệu trích xuất	Tài nguyên sử dụng	Độ phức tạp	Ứng dụng
Full extraction	Toàn bộ dữ liệu	Cao	Thấp	Khi tải dữ liệu lần đầu vào hệ thống hoặc khi dữ liệu nguồn thay đổi không thể theo dõi được.
Incremental extraction	Dữ liệu mới/thay đổi	Trung bình	Trung bình	Hệ thống xử lý dữ liệu theo batch, cập nhật dữ liệu định kỳ (hàng ngày, hàng giờ)
CDC (Change Data Capture)	Chỉ dữ liệu thay đổi (Insert,Update,Delete)	Thấp	Cao	Khi hệ thống yêu cầu xử lý theo thời gian thực.

2.3. Transform – Chuyển đổi dữ liệu

Giai đoạn **Transform (Chuyển đổi dữ liệu)** là một phần quan trọng của quy trình ETL, nơi dữ liệu được làm sạch, chuẩn hóa, tích hợp và biến đổi theo các quy tắc nghiệp vụ trước khi nạp vào hệ thống đích. Quá trình này đảm bảo chất lượng dữ liệu, giúp dữ liệu nhất quán, dễ dàng phân tích và khai thác.

Trong giai đoạn này, các quy tắc xử lý dữ liệu sẽ được áp dụng để:

- Đảm bảo tính chính xác và toàn vẹn của dữ liệu.
- Chuẩn hóa định dạng dữ liệu từ nhiều nguồn khác nhau.
- Tích hợp dữ liệu từ nhiều hệ thống với cấu trúc khác nhau.
- Biến đổi dữ liệu theo yêu cầu của hệ thống đích.

a) Chuyển đổi dữ liệu cơ bản

Đây là bước đầu tiên trong quá trình Transform, giúp cải thiện chất lượng dữ liệu bằng cách loại bỏ lỗi, xử lý các giá trị trống và đơn giản hóa dữ liệu.

Làm sạch dữ liệu

- Loại bỏ dữ liệu không hợp lệ, sai định dạng hoặc không đầy đủ.
- Xử lý các giá trị bị thiếu bằng cách thay thế bằng giá trị trung bình, giá trị mặc định hoặc loại bỏ hàng dữ liệu.
- Sửa lỗi chính tả, lỗi nhập liệu.
- Ví dụ:

  + Ánh xạ các trường dữ liệu trống thành số 0 hoặc “Unknown”.

Loại bỏ trùng lặp

- Xác định và loại bỏ các bản ghi trùng lặp dựa trên khóa chính hoặc các thuộc tính quan trọng.
- Áp dụng thuật toán so sánh dữ liệu (fuzzy matching) để phát hiện các bản ghi tương tự nhưng không hoàn toàn giống nhau.

<u>Chuẩn hóa định dạng dữ liệu</u>

- Chuyển đổi dữ liệu về một định dạng chung, nhất quán.
- Đồng nhất cách ghi chú, viết hoa/thường trong dữ liệu.
- Ví dụ:

  + Chuyển đổi các giá trị ngày tháng từ `MM/DD/YYYY` sang `YYYY-MM-DD`.

  + Một công ty thực phẩm có thể có dữ liệu nguyên liệu sử dụng cả kilogram và pound, ETL sẽ chuẩn hóa tất cả về pound.

b) Chuyển đổi dữ liệu nâng cao

Quá trình này tập trung vào **áp dụng quy tắc kinh doanh**, tối ưu hóa dữ liệu để phục vụ phân tích dễ dàng hơn.

<u>Dẫn xuất</u>

- Là việc áp dụng các quy tắc kinh doanh vào dữ liệu của bạn để tính toán các giá trị mới dựa trên các giá trị hiện có.
- Ví dụ:

  + Chuyển đổi **doanh thu** thành **lợi nhuận** bằng cách trừ đi chi phí.

  + Tính **tổng chi phí mua hàng** bằng cách nhân giá đơn vị với số lượng đặt hàng.

Gộp ghép

- Quá trình này liên kết **các dữ liệu tương đồng** từ nhiều nguồn khác nhau để tạo ra một bản ghi duy nhất, giúp dữ liệu nhất quán và tránh phân tán
- Ví dụ: Tổng hợp dữ liệu mua hàng từ nhiều nhà cung cấp và lưu lại tổng giá trị thay vì từng giao dịch riêng lẻ.

Chia tách dữ liệu

- Một cột hoặc một trường dữ liệu có thể được **tách thành nhiều cột nhỏ hơn** để cải thiện khả năng sử dụng dữ liệu.
- Ví dụ:

Tổng hợp dữ liệu

- Tổng hợp giúp **giảm số lượng dữ liệu**, tạo ra thông tin giá trị hơn từ dữ liệu gốc bằng cách nhóm và tính toán.
- Ví dụ: Dữ liệu hóa đơn của khách hàng chứa nhiều khoản thanh toán nhỏ lẻ, có thể tổng hợp thành **chỉ số giá trị lâu dài của khách hàng (CLV)** theo từng tháng/quý/năm.

Mã hóa dữ liệu

- Dữ liệu nhạy cảm có thể được **mã hóa hoặc ẩn danh** để tuân thủ quy định bảo mật và bảo vệ quyền riêng tư trước khi lưu trữ hoặc truyền tải.
- Ví dụ: Mã hóa số thẻ tín dụng của khách hàng thành dạng `****-****-****-1234` trước khi đưa vào cơ sở dữ liệu.

2.4. Load – Tải dữ liệu

Sau khi dữ liệu đã được trích xuất từ nguồn và chuyển đổi theo các quy tắc kinh doanh, bước cuối cùng của quy trình ETL (Extract, Transform, Load) là tải dữ liệu vào hệ thống đích, chẳng hạn như Data Warehouse, Data Lake, hoặc hệ thống cơ sở dữ liệu phân tích.

Quá trình tải dữ liệu có thể thực hiện theo hai phương pháp chính, tùy thuộc vào yêu cầu kinh doanh, khả năng xử lý của hệ thống, và khối lượng dữ liệu cần lưu trữ:

a) Tải đầy đủ (Full load)

Tải đầy đủ là quá trình chuyển toàn bộ dữ liệu từ nguồn vào hệ thống đích, thường được sử dụng trong các trường hợp sau:

- Khi lần đầu tiên triển khai hệ thống Data Warehouse/Data Lake.
- Khi thay đổi cấu trúc hoặc thiết kế dữ liệu, khiến dữ liệu cũ không còn phù hợp và cần làm mới hoàn toàn.
- Khi dữ liệu nguồn có kích thước nhỏ hoặc hệ thống đích có khả năng xử lý cao, giúp đảm bảo tính toàn vẹn dữ liệu mà không cần phải theo dõi các thay đổi.

Quy trình tải đầy đủ:

- Xóa toàn bộ dữ liệu cũ trong hệ thống đích (nếu có).
- Tải toàn bộ dữ liệu mới từ nguồn sau khi đã chuyển đổi.
- Kiểm tra và xác nhận tính toàn vẹn của dữ liệu sau khi tải.

Hạn chế:

- Tốn nhiều tài nguyên: Việc tải toàn bộ dữ liệu có thể tiêu tốn nhiều CPU, bộ nhớ và băng thông.
- Thời gian tải lâu: Với dữ liệu lớn, quá trình này có thể mất nhiều thời gian, ảnh hưởng đến hiệu suất hệ thống.
- Gián đoạn hoạt động: Nếu hệ thống đích đang hoạt động, việc tải đầy đủ có thể gây downtime hoặc làm chậm các truy vấn dữ liệu.

b) Tải tăng dần (Incremental load)

Tải tăng dần là phương pháp chỉ tải những bản ghi mới hoặc bị thay đổi từ hệ thống nguồn vào hệ thống đích. Đây là cách tiếp cận phổ biến trong các hệ thống dữ liệu lớn vì giúp giảm tải cho hệ thống và tối ưu hóa hiệu suất. Công cụ ETL sẽ theo dõi dấu thời gian của lần trích xuất cuối cùng, đảm bảo rằng chỉ các bản ghi mới hoặc đã thay đổi sau thời điểm đó mới được tải vào hệ thống đích.

Có hai cách thực hiện tải tăng dần:

Tải tăng dần theo luồng (Streaming Incremental Load)

Phương pháp này **truyền dữ liệu liên tục theo thời gian thực** hoặc gần thời gian thực vào hệ thống đích. Phù hợp với:

- Dữ liệu có khối lượng nhỏ hoặc trung bình nhưng yêu cầu cập nhật nhanh chóng.
- Hệ thống đích là Data Warehouse thời gian thực hoặc Data Lake hỗ trợ xử lý streaming**.**
- Ví dụ:

+ Hệ thống quản lý giao dịch ngân hàng cần cập nhật dữ liệu giao dịch khách hàng theo thời gian thực.

+ Các nền tảng thương mại điện tử cần đồng bộ số lượng hàng tồn kho ngay khi có giao dịch.

Tải tăng dần theo lô (Batch Incremental Load)

Thay vì truyền dữ liệu liên tục, phương pháp này thu thập dữ liệu thay đổi theo khoảng thời gian nhất định (ví dụ: hàng giờ, hàng ngày) và tải vào hệ thống đích theo từng đợt (batch). Phù hợp với:

- Dữ liệu có khối lượng lớn và không yêu cầu cập nhật liên tục.
- Hệ thống đích là Data Warehouse truyền thống, nơi dữ liệu cần được tổng hợp theo từng giai đoạn.
- Khi tải dữ liệu từ hệ thống giao dịch (OLTP) sang hệ thống phân tích (OLAP), không cần cập nhật tức thì.
- Ví dụ:

  + Một công ty bán lẻ cập nhật dữ liệu bán hàng theo từng ngày để phân tích doanh thu.

  + Hệ thống phân tích dữ liệu khách hàng cập nhật hành vi người dùng mỗi giờ để tối ưu quảng cáo.

CHƯƠNG 3: DEMO XÂY DỰNG MODULE ETL

Chương cuối này sẽ trình bày kết quả thử nghiệm xây dựng module ETL thu thập, xử lý và phân tích dữ liệu về những chuyến đi của một công ty taxi ở New York, Mỹ của nhóm sau khi áp dụng lý thuyết đã tìm hiểu được

3.1 Tìm hiểu tập dataset.

+ Tập dataset được lấy từ trang TLC Trip Record (https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page).

Đây dữ liệu được công khai bởi **New York City Taxi and Limousine Commission (TLC)**. Tập dữ liệu này chứa thông tin chi tiết về các chuyến đi của các dịch vụ taxi (như Yellow Taxi, Green Taxi) và dịch vụ chia sẻ xe như Uber hoạt động tại thành phố New York.

+ Tổng quan về dataset và ý nghĩa của từng trường (được định nghĩa trong file data dictionary)

Data Dictionary – Yellow Taxi Trip Records May 11, 2022 Page 1 of 2

This data dictionary describes yellow taxi trip data. For a dictionary describing green taxi data, or a map of the TLC Taxi Zones, please visit http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.

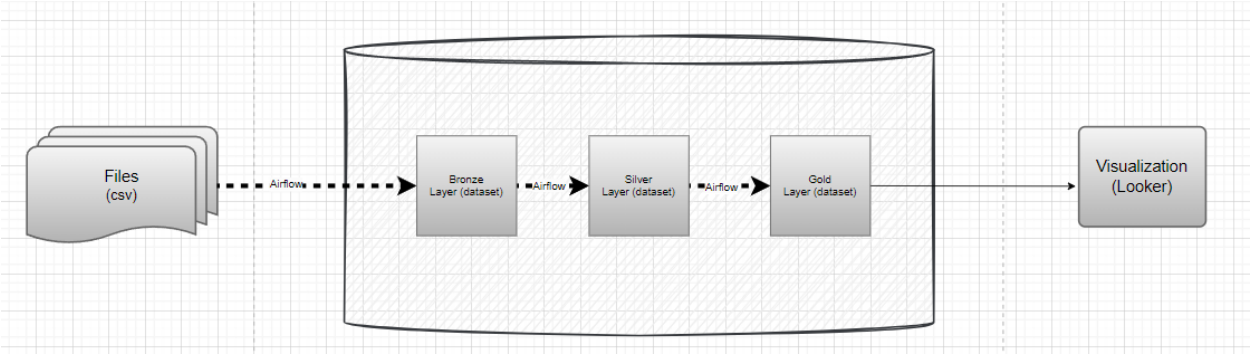
Field Name	Description
VendorID	A code indicating the TPEP provider that provided the record. **1= Creative Mobile Technologies, LLC; 2= VeriFone Inc.**
tpep_pickup_datetime	The date and time when the meter was engaged.
tpep_dropoff_datetime	The date and time when the meter was disengaged.
Passenger_count	The number of passengers in the vehicle. This is a driver-entered value.
Trip_distance	The elapsed trip distance in miles reported by the taximeter.
PULocationID	TLC Taxi Zone in which the taximeter was engaged
DOLocationID	TLC Taxi Zone in which the taximeter was disengaged
RateCodeID	The final rate code in effect at the end of the trip. **1= Standard rate** **2=JFK** **3=Newark** **4=Nassau or Westchester** **5=Negotiated fare** **6=Group ride**
Store_and_fwd_flag	This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka "store and forward," because the vehicle did not have a connection to the server. **Y= store and forward trip** **N= not a store and forward trip**
Payment_type	A numeric code signifying how the passenger paid for the trip. **1= Credit card** **2= Cash** **3= No charge** **4= Dispute** **5= Unknown** **6= Voided trip**
Fare_amount	The time-and-distance fare calculated by the meter.
Extra	Miscellaneous extras and surcharges. Currently, this only includes the $0.50 and $1 rush hour and overnight charges.
MTA_tax	$0.50 MTA tax that is automatically triggered based on the metered rate in use.
Improvement_surcharge	$0.30 improvement surcharge assessed trips at the flag drop. The improvement surcharge began being levied in 2015.
Tip_amount	Tip amount – This field is automatically populated for credit card tips. Cash tips are not included.
Tolls_amount	Total amount of all tolls paid in trip.
Total_amount	The total amount charged to passengers. Does not include cash tips.
Congestion_Surcharge	Total amount collected in trip for NYS congestion surcharge.
Airport_fee	$1.25 for pick up only at LaGuardia and John F. Kennedy Airports

Hình 2: Định nghĩa các field của dataset

3.2 Thiết kế hệ thống ETL.

+ Công nghệ sử dụng: Google Cloud Platform, Apache Airflow, Looker, Docker

+ Tổng quan hệ thống:



Hình 3: Tổng quan hệ thống ETL

- Đầu tiên, dữ liệu thô được extract từ file csv, thu thập từ trang TLC Trip Record
- Tiếp theo, tải dữ liệu thô vào Data Warehouse (gồm 3 layer Bronze -> Silver -> Gold để thực hiện transform dữ liệu)

  + Bronze: Load thẳng data raw từ file csv

  + Silver: Làm sạch, xử lý chuyển đổi dữ liệu

  + Gold: Dữ liệu đã được xử lý, phục vụ cho báo cáo (dữ liệu từ layer gold sẽ được kết nối đến Looker Studio để tạo báo cáo)

3.3 Xây dựng hệ thống

a) Bronze Layer

+ Thực hiện load file dữ liệu csv lên GCS (Bucket), rồi từ Bucket đẩy sang BigQuery

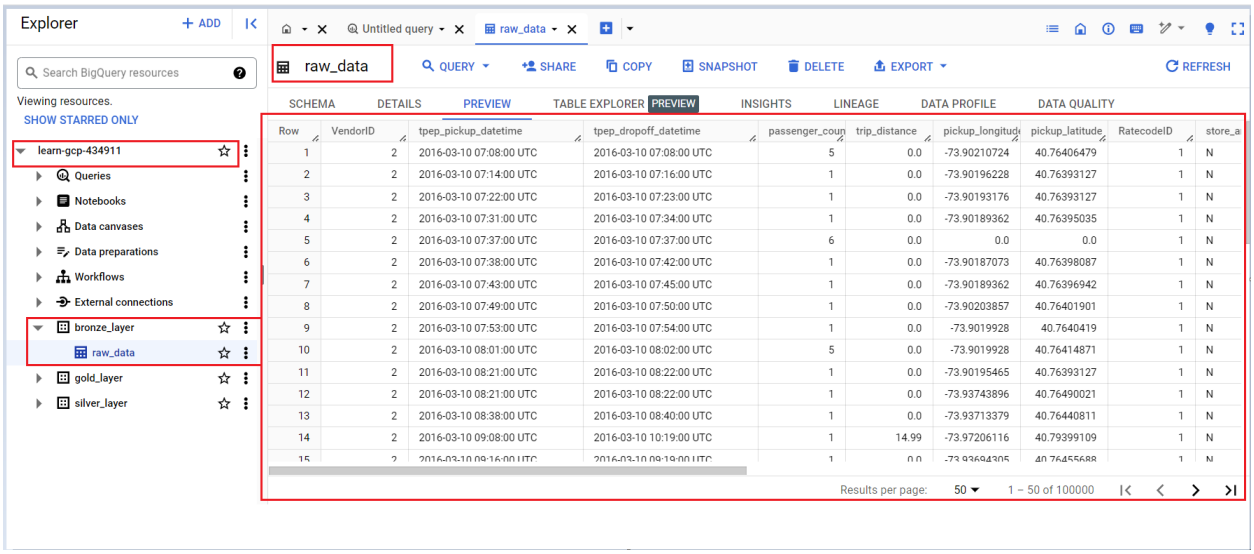
+ Code DAG thực hiện:

```python
gr2_load_to_bigquery = GCSToBigQueryOperator(
    task_id='load_to_bigquery',
    bucket='bkt-sales-data-hoang',
    source_objects=['uber_data.csv'],
    destination_project_dataset_table='learn-gcp-434911.bronze_layer.raw_data',
    schema_fields=[
        {'name': 'VendorID', 'type': 'INTEGER', 'mode': 'NULLABLE'},
        {'name': 'tpep_pickup_datetime', 'type': 'TIMESTAMP', 'mode': 'NULLABLE'},
        {'name': 'tpep_dropoff_datetime', 'type': 'TIMESTAMP', 'mode': 'NULLABLE'},
        {'name': 'passenger_count', 'type': 'INTEGER', 'mode': 'NULLABLE'},
        {'name': 'trip_distance', 'type': 'FLOAT64', 'mode': 'NULLABLE'},
        {'name': 'pickup_longitude', 'type': 'FLOAT64', 'mode': 'NULLABLE'},
        {'name': 'pickup_latitude', 'type': 'FLOAT64', 'mode': 'NULLABLE'},
        {'name': 'RatecodeID', 'type': 'INTEGER', 'mode': 'NULLABLE'},
        {'name': 'store_and_fwd_flag', 'type': 'STRING', 'mode': 'NULLABLE'},
        {'name': 'dropoff_longitude', 'type': 'FLOAT64', 'mode': 'NULLABLE'},
        {'name': 'dropoff_latitude', 'type': 'FLOAT64', 'mode': 'NULLABLE'},
        {'name': 'payment_type', 'type': 'INTEGER', 'mode': 'NULLABLE'},
        {'name': 'fare_amount', 'type': 'FLOAT64', 'mode': 'NULLABLE'},
        {'name': 'extra', 'type': 'FLOAT64', 'mode': 'NULLABLE'},
        {'name': 'mta_tax', 'type': 'FLOAT64', 'mode': 'NULLABLE'},
        {'name': 'tip_amount', 'type': 'FLOAT64', 'mode': 'NULLABLE'},
        {'name': 'tolls_amount', 'type': 'FLOAT64', 'mode': 'NULLABLE'},
        {'name': 'improvement_surcharge', 'type': 'FLOAT64', 'mode': 'NULLABLE'},
        {'name': 'total_amount', 'type': 'FLOAT64', 'mode': 'NULLABLE'},

    ],
    source_format = 'CSV',
    skip_leading_rows = 1,
    write_disposition ='WRITE_TRUNCATE',
    gcp_conn_id = 'google_cloud_default'
)
```

```python
gr2_upload_file_to_gcs = LocalFilesystemToGCSOperator(
    task_id='upload_file_to_gcs',
    src='/opt/airflow/uber_data.csv',
    dst='uber_data.csv',
    bucket='bkt-sales-data-hoang',
)
```

Hình 4: Code DAG Bronze Layer

Kết quả



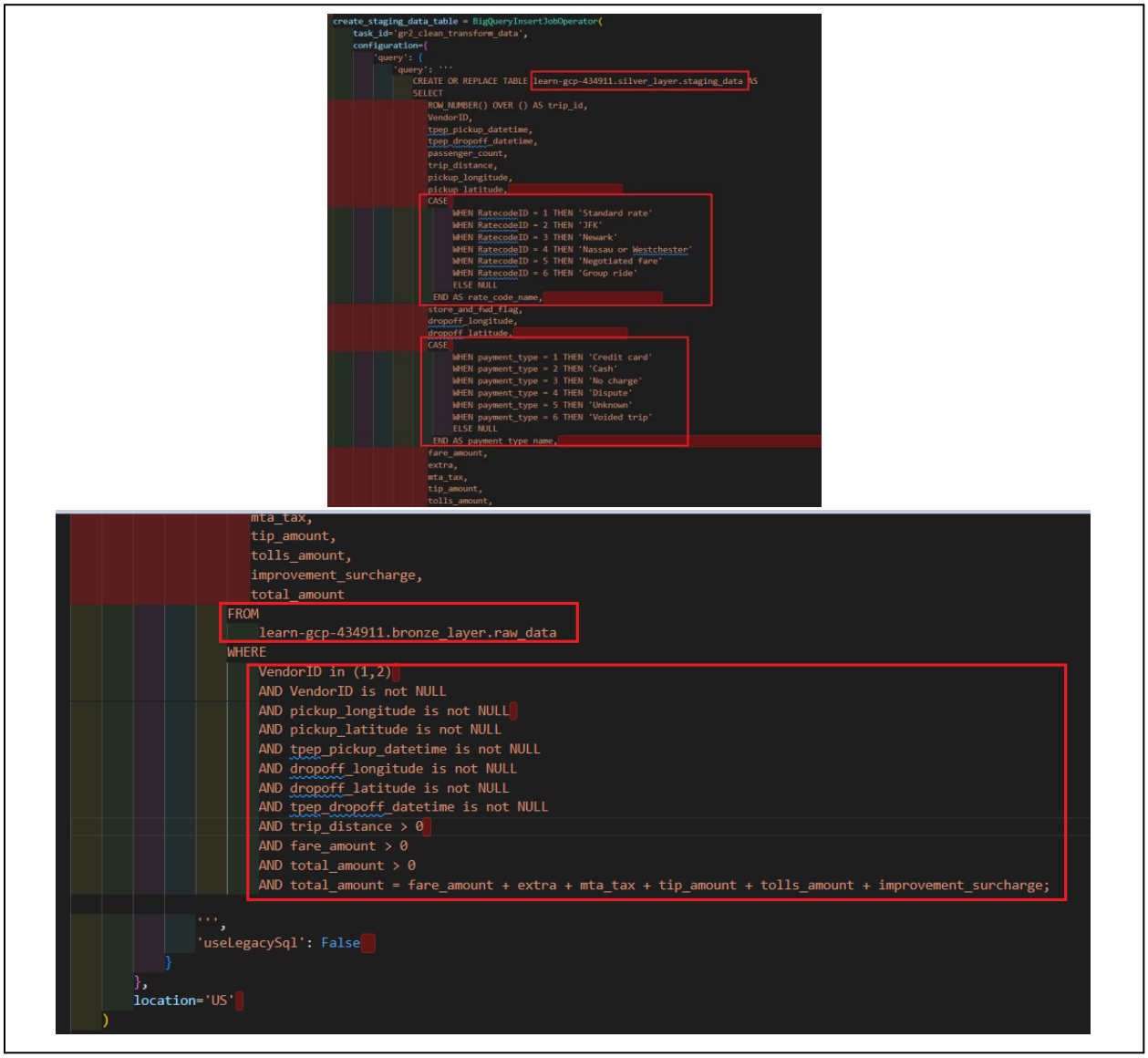
Hình 5: Data Bronze Layer

b) Silver Layer

+ Xử lý làm sạch, chuyển đổi dữ liệu từ tầng bronze, rồi đẩy dữ liệu đã được xử lý

vào dataset mới (silver_layer) trên BigQuery

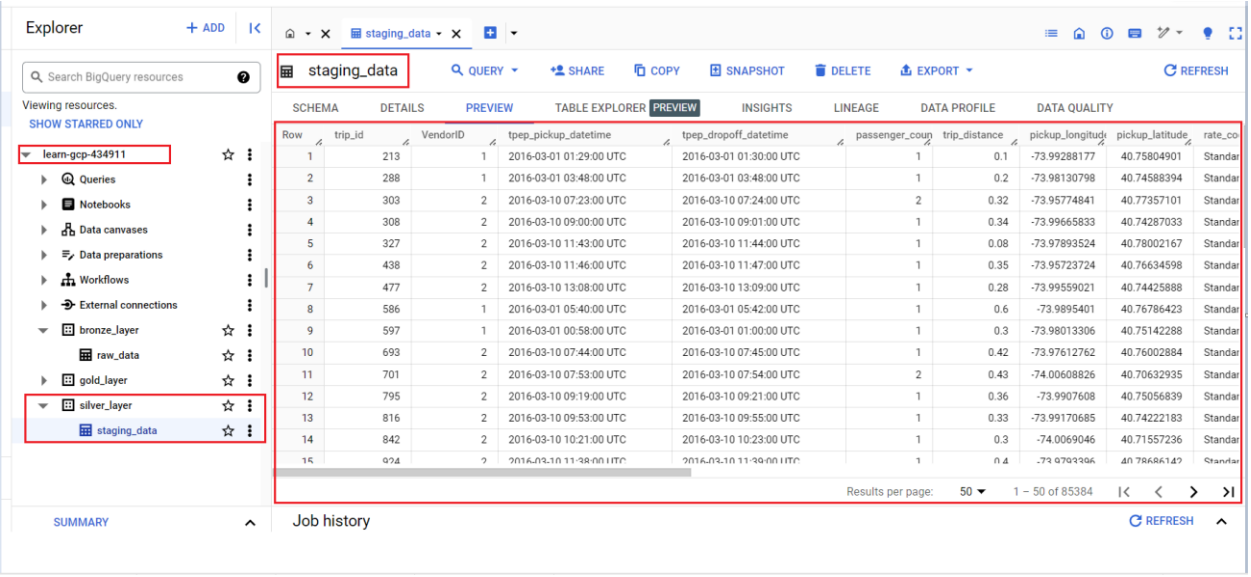
+ Code DAG thực hiện:



```
create_staging_data_table = BigQueryInsertJobOperator(
    task_id='gr2_clean_transform_data',
    configuration={
        'query': {
            'query': '''
                CREATE OR REPLACE TABLE learn-gcp-434911.silver_layer.staging_data AS
                SELECT
                    ROW_NUMBER() OVER () AS trip_id,
                    VendorID,
                    tpep_pickup_datetime,
                    tpep_dropoff_datetime,
                    passenger_count,
                    trip_distance,
                    pickup_longitude,
                    pickup_latitude,
                    CASE
                        WHEN RatecodeID = 1 THEN 'Standard rate'
                        WHEN RatecodeID = 2 THEN 'JFK'
                        WHEN RatecodeID = 3 THEN 'Newark'
                        WHEN RatecodeID = 4 THEN 'Nassau or Westchester'
                        WHEN RatecodeID = 5 THEN 'Negotiated fare'
                        WHEN RatecodeID = 6 THEN 'Group ride'
                        ELSE NULL
                    END AS rate_code_name,
                    store_and_fwd_flag,
                    dropoff_longitude,
                    dropoff_latitude,
                    CASE
                        WHEN payment_type = 1 THEN 'Credit card'
                        WHEN payment_type = 2 THEN 'Cash'
                        WHEN payment_type = 3 THEN 'No charge'
                        WHEN payment_type = 4 THEN 'Dispute'
                        WHEN payment_type = 5 THEN 'Unknown'
                        WHEN payment_type = 6 THEN 'Voided trip'
                        ELSE NULL
                    END AS payment_type_name,
                    fare_amount,
                    extra,
                    mta_tax,
                    tip_amount,
                    tolls_amount,
                    improvement_surcharge,
                    total_amount
                FROM
                    learn-gcp-434911.bronze_layer.raw_data
                WHERE
                    VendorID in (1,2)
                    AND VendorID is not NULL
                    AND pickup_longitude is not NULL
                    AND pickup_latitude is not NULL
                    AND tpep_pickup_datetime is not NULL
                    AND dropoff_longitude is not NULL
                    AND dropoff_latitude is not NULL
                    AND tpep_dropoff_datetime is not NULL
                    AND trip_distance > 0
                    AND fare_amount > 0
                    AND total_amount > 0
                    AND total_amount = fare_amount + extra + mta_tax + tip_amount + tolls_amount + improvement_surcharge;
            ''',
            'useLegacySql': False
        }
    },
    location='US'
)
```

Hình 6: Code DAG Silver Layer

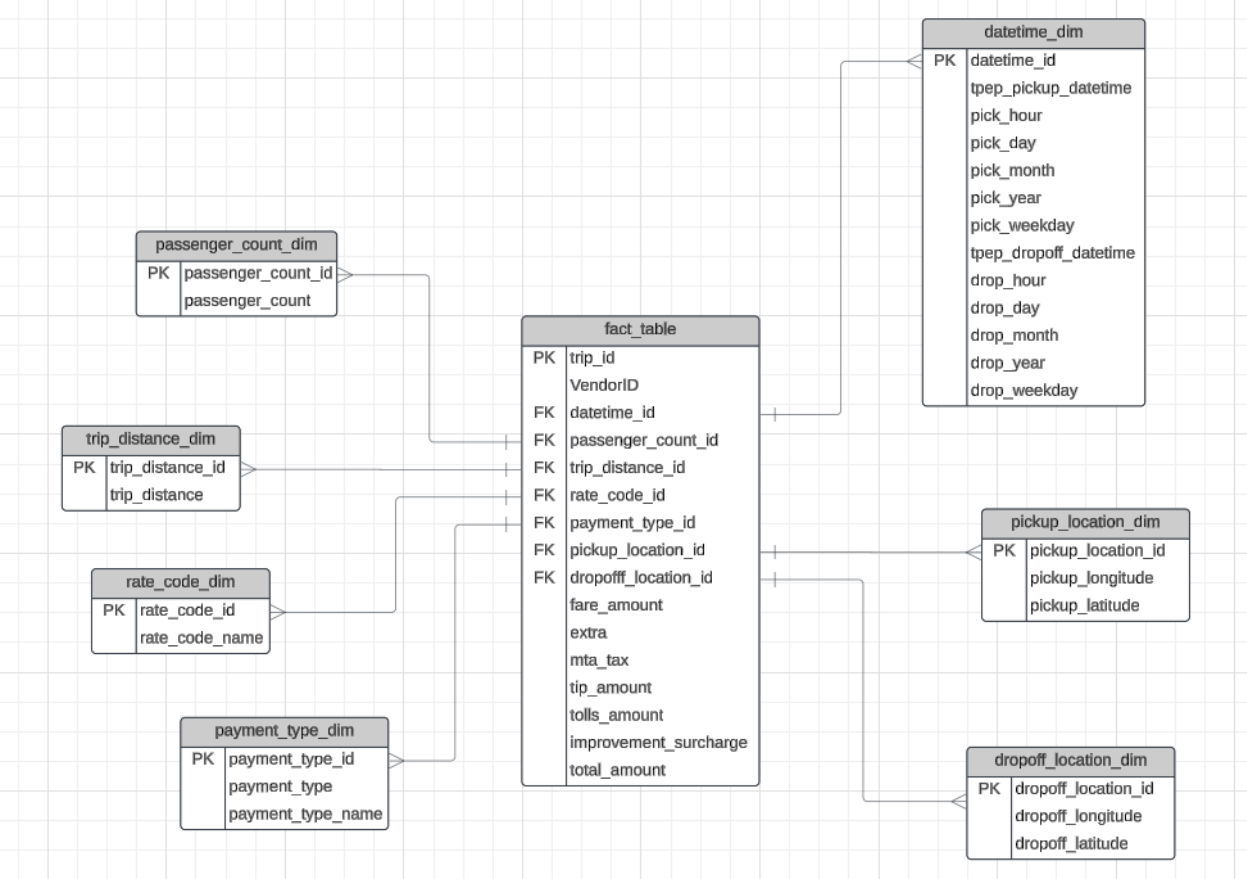
Kết quả:



Hình 7: Data Silver Layer

b) Gold Layer

+ Thiết kế Data Warehouse theo mô hình Star Schema như sau:



Hình 8: Lược đồ Star Schema của Data Warehouse

+ Từ data của tầng Silver, sẽ xử lý để tạo ra các bảng dim và fact như trên, rồi đẩy các bảng đó vào dataset mới (gold_layer) trên BigQuery

+ Ví dụ code DAG tạo bảng dim date và bảng fact:

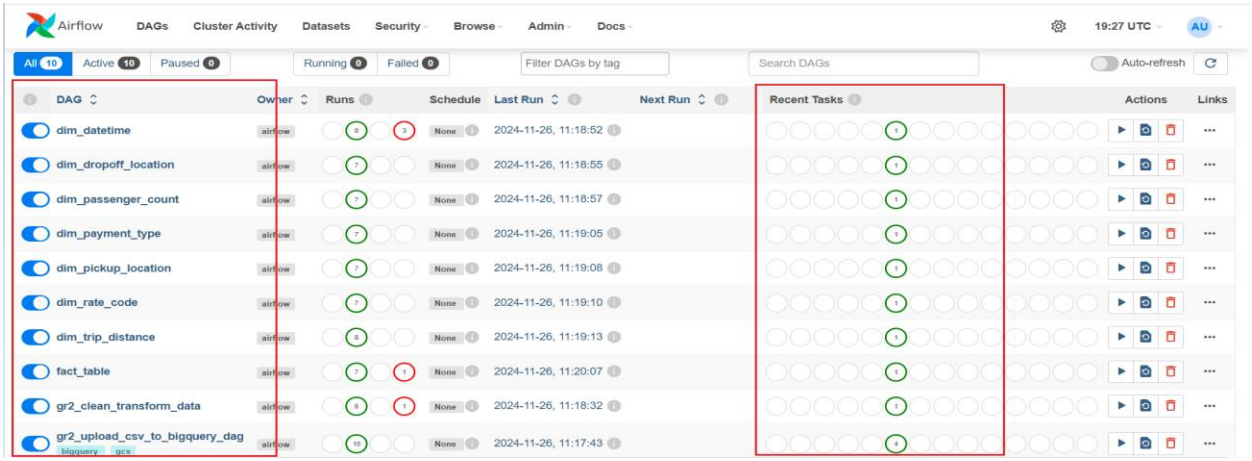
```
with DAG(
    'dim_datetime',
    default_args={'start_date': days_ago(1)},
    schedule_interval=None,
    catchup=False
) as dag:

    create_datetime_dim = BigQueryInsertJobOperator(
        task_id='create_datetime_dim',
        configuration={
            'query': {
                'query': '''
                    CREATE OR REPLACE TABLE learn-gcp-434911.gold_layer.dim_datetime AS
                    SELECT
                        ROW_NUMBER() OVER () AS datetime_id,

                        tpep_pickup_datetime,
                        EXTRACT(HOUR FROM tpep_pickup_datetime) AS pick_hour,
                        EXTRACT(DAY FROM tpep_pickup_datetime) AS pick_day,
                        EXTRACT(MONTH FROM tpep_pickup_datetime) AS pick_month,
                        EXTRACT(YEAR FROM tpep_pickup_datetime) AS pick_year,
                        EXTRACT(DAYOFWEEK FROM tpep_pickup_datetime) AS pick_weekday,

                        tpep_dropoff_datetime,
                        EXTRACT(HOUR FROM tpep_dropoff_datetime) AS drop_hour,
                        EXTRACT(DAY FROM tpep_dropoff_datetime) AS drop_day,
                        EXTRACT(MONTH FROM tpep_dropoff_datetime) AS drop_month,
                        EXTRACT(YEAR FROM tpep_dropoff_datetime) AS drop_year,
                        EXTRACT(DAYOFWEEK FROM tpep_dropoff_datetime) AS drop_weekday
                    FROM
                        learn-gcp-434911.silver_layer.staging_data;
                ''',
                'useLegacySql': False
```

Hình 9: Code DAG tạo bảng dim_date

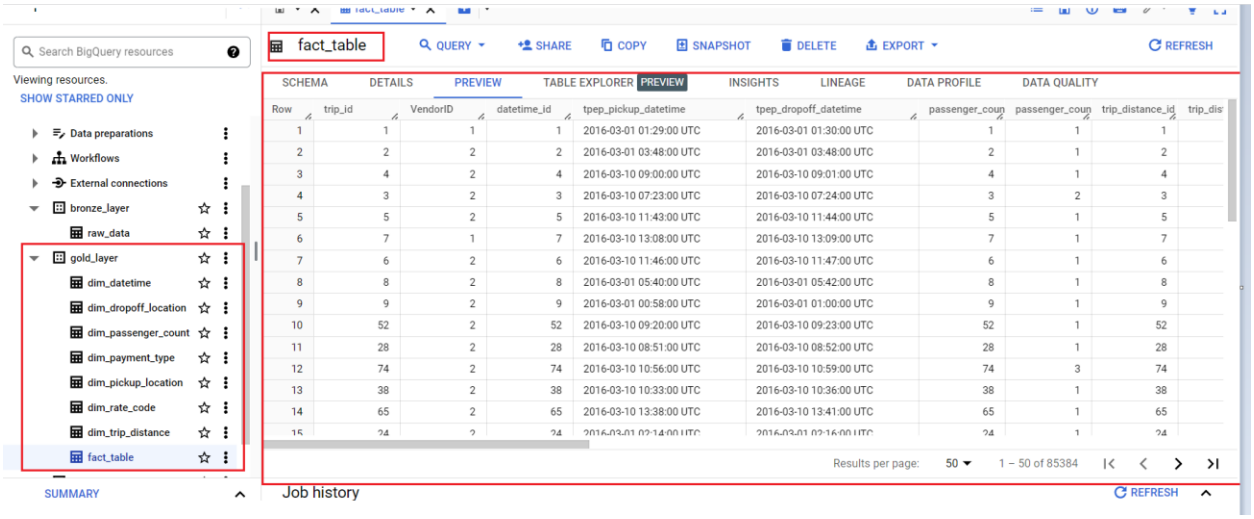
```
task_id='create_fact_table',
configuration={
    'query': {
        'query': '''
            CREATE OR REPLACE TABLE learn-gcp-434911.gold_layer.fact_table AS
            SELECT
                s.trip_id,
                s.VendorID,
                dt.datetime_id,
                dt.tpep_pickup_datetime,
                dt.tpep_dropoff_datetime,
                pc.passenger_count_id,
                pc.passenger_count,
                td.trip_distance_id,
                td.trip_distance,
                rc.rate_code_id,
                rc.rate_code_name,
                s.store_and_fwd_flag,
                pl.pickup_location_id,
                pl.pickup_latitude,
                pl.pickup_longitude,
                dl.dropoff_location_id,
                dl.dropoff_latitude,
                dl.dropoff_longitude,
                pt.payment_type_id,
                pt.payment_type_name,
                s.fare_amount,
                s.extra,
                s.mta_tax,
                s.tip_amount,
                s.tolls_amount,
                s.improvement_surcharge,
                s.total_amount
            FROM
            learn-gcp-434911.silver_layer.staging_data s
            JOIN learn-gcp-434911.gold_layer.dim_datetime dt on s.trip_id = dt.datetime_id
            JOIN learn-gcp-434911.gold_layer.dim_dropoff_location dl on s.trip_id = dl.dropoff_location_id
            JOIN learn-gcp-434911.gold_layer.dim_passenger_count pc on s.trip_id = pc.passenger_count_id
            JOIN learn-gcp-434911.gold_layer.dim_payment_type pt on s.trip_id = pt.payment_type_id
            JOIN learn-gcp-434911.gold_layer.dim_pickup_location pl on s.trip_id = pl.pickup_location_id
            JOIN learn-gcp-434911.gold_layer.dim_rate_code rc on s.trip_id = rc.rate_code_id
            JOIN learn-gcp-434911.gold_layer.dim_trip_distance td on s.trip_id = td.trip_distance_id;
```

Hình 10: Code DAG tạo bảng fact

Kết quả



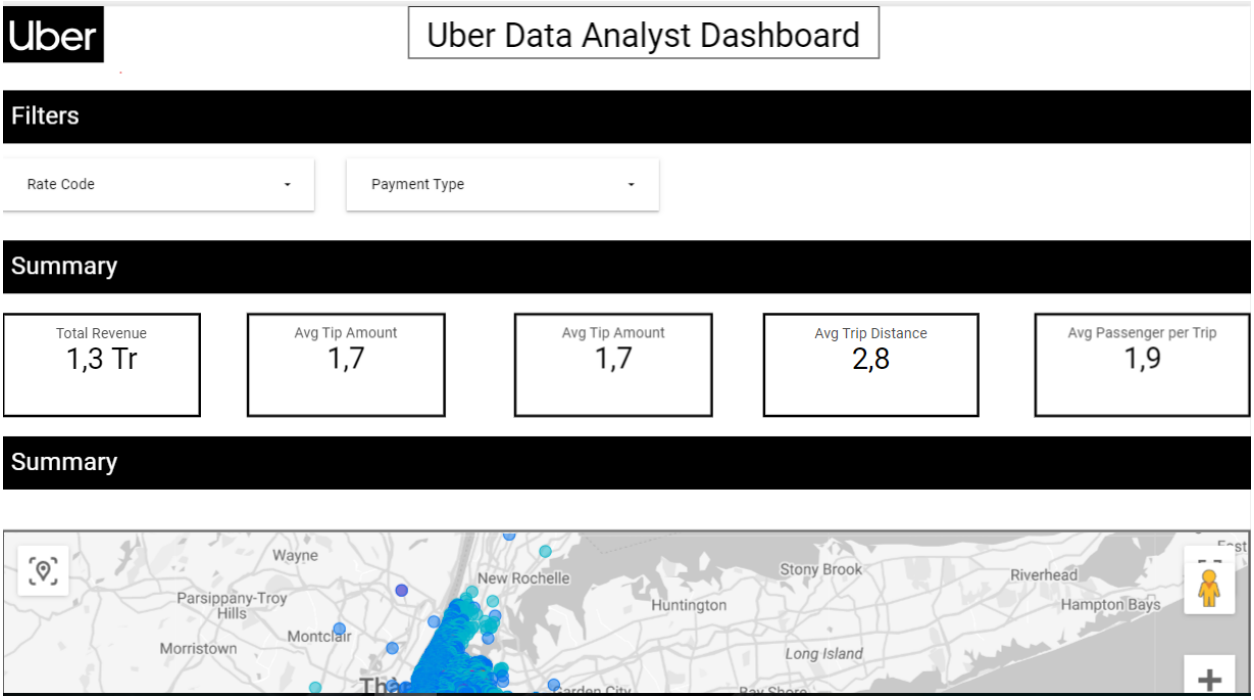
Hình 11: Kết quả chạy các DAG

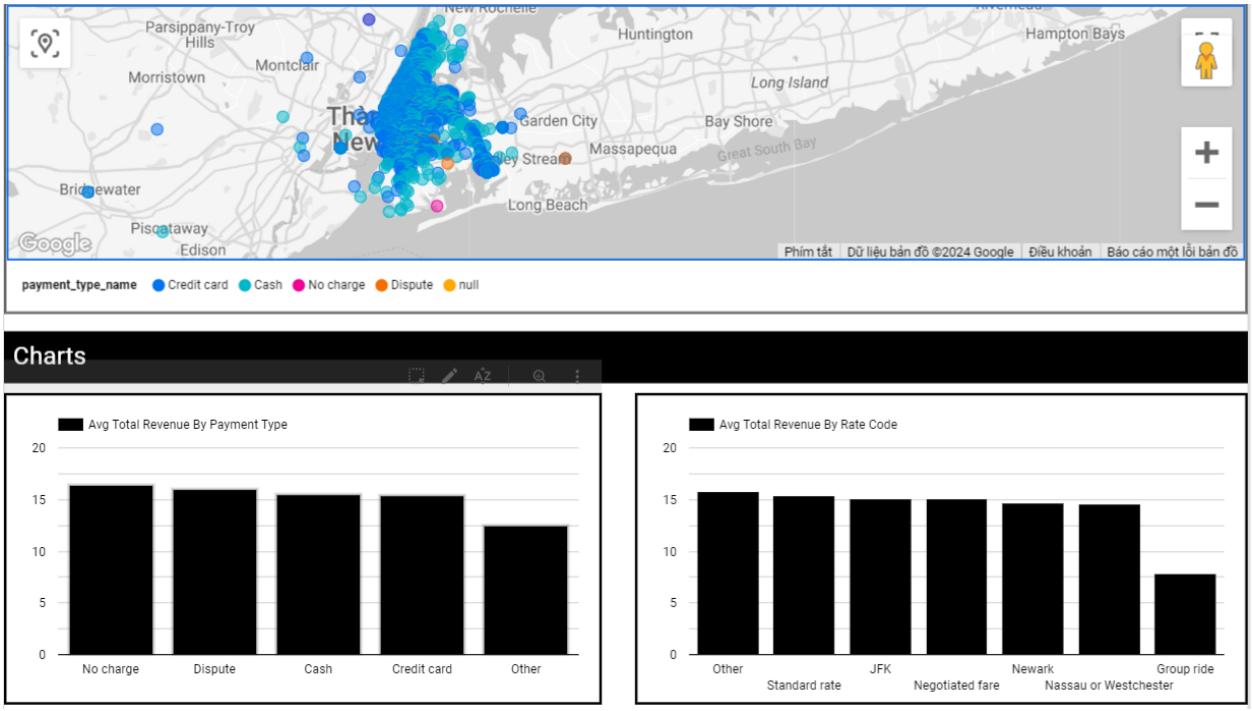


Hình 12: Data Gold Layer của bảng fact

d) Data Visualization

Data từ gold_layer của BigQuery sẽ được kết nối đến Looker Studio để tạo báo cáo trực quan với 1 số thông tin như: Tổng doanh thu, Trung bình khoảng cách/số khách mỗi chuyến đi, Trung bình doanh thu theo mỗi phương thức thanh toán,…





Hình 13: Looker Studio Report

KẾT LUẬN

Thiết kế hệ thống ETL đóng vai trò quan trọng trong việc đảm bảo chất lượng và tính sẵn sàng của dữ liệu phục vụ phân tích và ra quyết định. Mục tiêu chính của thiết kế ETL là xây dựng một quy trình xử lý dữ liệu hiệu quả, giúp trích xuất, biến đổi và tải dữ liệu từ nhiều nguồn khác nhau vào hệ thống lưu trữ tập trung. Điều này đảm bảo dữ liệu được tổ chức chặt chẽ, nhất quán và đáng tin cậy cho các ứng dụng phân tích.

Trong quá trình thiết kế hệ thống ETL, các yếu tố quan trọng như hiệu suất xử lý, khả năng mở rộng, tính toàn vẹn dữ liệu và khả năng giám sát đã được xem xét kỹ lưỡng. Kết quả triển khai cho thấy hệ thống có thể xử lý dữ liệu với độ chính xác cao, tốc độ tối ưu và khả năng tích hợp linh hoạt với nhiều nền tảng lưu trữ khác nhau. Điều này không chỉ giúp đảm bảo dòng chảy dữ liệu ổn định mà còn hỗ trợ tốt hơn cho việc khai thác thông tin và tối ưu hóa quy trình kinh doanh.

Tuy nhiên, để duy trì và nâng cao hiệu quả của hệ thống ETL, việc liên tục giám sát, kiểm tra và cải tiến là rất cần thiết. Điều này bao gồm việc tối ưu hóa hiệu suất xử lý, cải thiện khả năng mở rộng và tích hợp các công nghệ mới như xử lý dữ liệu theo thời gian thực. Thông qua việc điều chỉnh và nâng cấp định kỳ, hệ thống ETL sẽ ngày càng hoàn thiện, đáp ứng tốt hơn các nhu cầu phân tích dữ liệu ngày càng phức tạp của doanh nghiệp.

TÀI LIỆU THAM KHẢO

[1] Cem Kaner, «Testing Computer Software,» Wiley, 1999.

[2] «Bài giảng kiểm thử phần mềm,» Trung tâm đào tạo TesterTOP, 2022.

[3] Ron Patton, «Software Testing 2nd Edition,» Sams Publishing, 2005.

[4] Trần Anh Tuấn, «ĐỒ ÁN TỐT NGHIỆP "Ứng dụng tập mờ bức tranh trong quản lý nuôi trồng thủy hải sản và rừng ngập mặn dựa vào ảnh hưởng các yếu tố tự nhiên",» Đại học Bách khoa Hà Nội, 2022.

[5] Phạm Quang Huy, «Giáo trình thực hành Kiểm thử phần mềm,» Nhà xuất bản Thanh Niên, 2020.

[6] Ian Sommerville, «Software Engineering: Seventh,» Pearson Education, 2004.

[7] Roger S. Pressman, «Software Engineering: A Practitioner's Approach 6th edition,» McGraw-Hill, 2004.

[8] John D. Musa, «Software Reliability Engineering,» McGraw-Hill, 1998.

[9] Nguyễn Thanh Hùng, «Bài giảng Kiểm thử phần mềm,» Đại học Bách khoa Hà Nội.