

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN



BÁO CÁO ĐỒ ÁN

XỬ LÝ DỮ LIỆU MẠNG XÃ HỘI DỰA TRÊN CÔNG NGHỆ GOOGLE CLOUD PLATFORM

Sinh viên thực hiện:

Âu Trường Giang - 21522019

Trần Lê Khánh Hân - 21522039

Nguyễn Nguyễn Thành An - 21521806

Giảng viên:

ThS. Nguyễn Thị Anh Thư

Thành phố Hồ Chí Minh, tháng 12 năm 2023

MỤC LỤC

Chương 1: GIỚI THIỆU	2
I. Các công nghệ triển khai dữ liệu lớn	2
II. Nội dung tìm hiểu	2
Chương 2: ĐÁNH GIÁ	3
I. Ưu nhược điểm	3
1. Ưu điểm	3
2. Nhược điểm	3
II. Ứng dụng	4
Chương 3: CÁCH THỨC TRIỂN KHAI	5
I. Data House	5
1. Lakehouse	5
a. Lựa chọn Dịch vụ GCP phù hợp	5
b. Xử lý và Biến đổi Dữ liệu	5
c. Xây dựng Lakehouse	5
d. Phân tích và Trực quan hóa Dữ liệu	6
2. Kiến trúc component	6
II. Các components	8
1. Data sources	8
1. Các loại dữ liệu	8
2. Thu thập dữ liệu thô	8
3. Ingest	10
2. Process	13
a. Xử lý dữ liệu	13
b. Thống kê dữ liệu	19
c. Trực quan hóa dữ liệu	23
c. Enrich	26
Chương 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	31
TÀI LIỆU THAM KHẢO	32

Chương 1: GIỚI THIỆU

I. Các công nghệ triển khai dữ liệu lớn

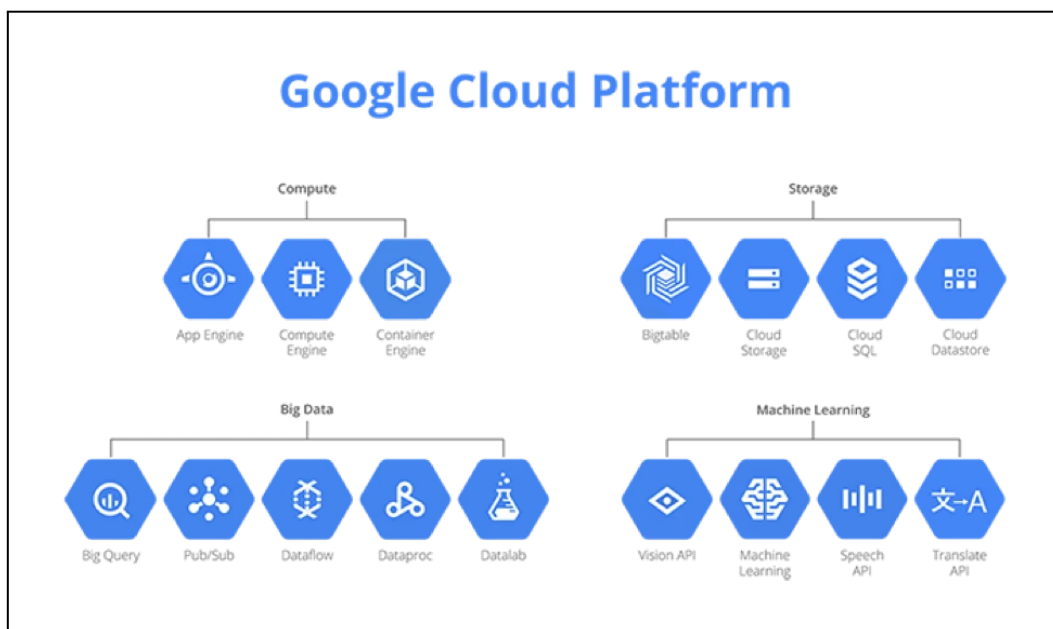
Hiện nay, trên thế giới có các công nghệ triển khai dữ liệu lớn: Apache Hadoop, Presto, Apache Spark, Kafka, Tableau,....

Trong đó, Google Cloud Platform là một trong những sự lựa chọn cho việc xử lý dữ liệu lớn mạng xã hội.

II. Nội dung tìm hiểu

Google Cloud Platform, viết tắt là GCP, là nền tảng điện toán đám mây do Google cung cấp. Nền tảng này bao gồm một loạt các dịch vụ được lưu trữ để tính toán, lưu trữ và phát triển ứng dụng chạy trên phần cứng của Google.

Google Cloud cung cấp các dịch vụ về máy tính, mạng, lưu trữ, big data, machine learning và IoT, cũng như các công cụ khác dành cho nhà phát triển bảo mật và quản lý Cloud.



Hình 1: Các dịch vụ phổ biến của Google Cloud Platform

Chương 2: ĐÁNH GIÁ

I. Ưu nhược điểm

1. Ưu điểm

- Tính linh hoạt và mở rộng: Cung cấp một loạt các dịch vụ và giải pháp để đáp ứng nhu cầu cụ thể của doanh nghiệp.
- Hiệu suất cao: Có hạ tầng toàn cầu với các trung tâm dữ liệu ở nhiều quốc gia, giúp cải thiện hiệu suất và giảm độ trễ đối với người dùng.
- Dịch vụ trí tuệ nhân tạo và Machine Learning: GCP cung cấp các công cụ và API mạnh mẽ cho việc phát triển các ứng dụng trí tuệ nhân tạo, Machine Learning và xử lý ngôn ngữ tự nhiên.
- Tính bảo mật: Tuân thủ các tiêu chuẩn bảo mật quốc tế, cung cấp các công cụ bảo mật như quản lý quyền truy cập và mã hóa dữ liệu.
- Hỗ trợ đa nền tảng: Hỗ trợ nhiều hệ điều hành và ngôn ngữ lập trình, bao gồm Linux, Windows, Java, Python, Node.js và nhiều ngôn ngữ khác.

2. Nhược điểm

- Chi phí: GCP có thể tốn kém hơn các nền tảng điện toán đám mây khác, chẳng hạn như Amazon Web Services (AWS) và Microsoft Azure.
- Tính phức tạp: GCP có thể phức tạp để sử dụng, đặc biệt là đối với người mới bắt đầu.
- Tương thích: GCP có thể không tương thích với tất cả các ứng dụng và dịch vụ.
- Khả năng di chuyển dữ liệu từ các hạ tầng on-premises hoặc từ nhà cung cấp đám mây khác gặp thách thức, đặc biệt là đối với các ứng dụng đòi hỏi thời gian hoạt động liên tục.

II. Ứng dụng

- Phân tích dữ liệu và kho dữ liệu lớn: BigQuery cho phép phân tích dữ liệu lớn một cách nhanh chóng và hiệu quả, Google Cloud Dataflow cung cấp khả năng xử lý dữ liệu dòng và dữ liệu tĩnh.
- Trí tuệ nhân tạo và Machine Learning: Google Cloud AI và Google Cloud Machine Learning Engine để phát triển và triển khai ứng dụng trí tuệ nhân tạo và machine learning.
- An ninh: cung cấp một loạt các dịch vụ bảo mật, chẳng hạn như xác thực, mã hóa và giám sát. Các dịch vụ này giúp bảo vệ dữ liệu và ứng dụng của bạn khỏi bị tấn công.
- Internet of Things (IoT): GCP có các dịch vụ như Google Cloud IoT Core cho phép kết nối, quản lý và thu thập dữ liệu từ các thiết bị IoT.

Chương 3: CÁCH THỨC TRIỂN KHAI

I. Data House

Đối với công nghệ Google Cloud, nhóm chúng em đã chọn mô hình Lakehouse để triển khai các thành phần theo kiến trúc dữ liệu lớn.

1. Lakehouse

Lakehouse là một mô hình kết hợp tính năng của data lake và data warehouse. Trong GCP, có thể triển khai lakehouse bằng cách kết hợp GCS và BigQuery.

a. Lựa chọn Dịch vụ GCP phù hợp

- Cloud Storage: lưu trữ dữ liệu gốc. Chúng ta có thể sử dụng các bucket để tổ chức dữ liệu của mình theo cách có cấu trúc.
- BigQuery: nơi để truy vấn và phân tích dữ liệu. BigQuery hỗ trợ truy vấn dữ liệu trực tiếp từ Cloud Storage và có tích hợp tính năng làm việc với dữ liệu lake house.

b. Xử lý và Biến đổi Dữ liệu

Cloud Dataflow: xử lý và biến đổi dữ liệu trước khi nạp vào BigQuery hoặc lưu trữ trong Cloud Storage, bao gồm việc lọc, sắp xếp, chuyển đổi định dạng, và làm sạch dữ liệu.

c. Xây dựng Lakehouse

Tạo cấu trúc thư mục hợp lý trong Cloud Storage để lưu trữ dữ liệu gốc và dữ liệu đã được tiền xử lý. Cấu trúc thư mục nên được thiết kế để phản ánh mô hình dữ liệu của bạn.

d. Phân tích và Trực quan hóa Dữ liệu

Sử dụng các công cụ như BigQuery, Data Studio để thực hiện phân tích dữ liệu và tạo báo cáo trực quan.

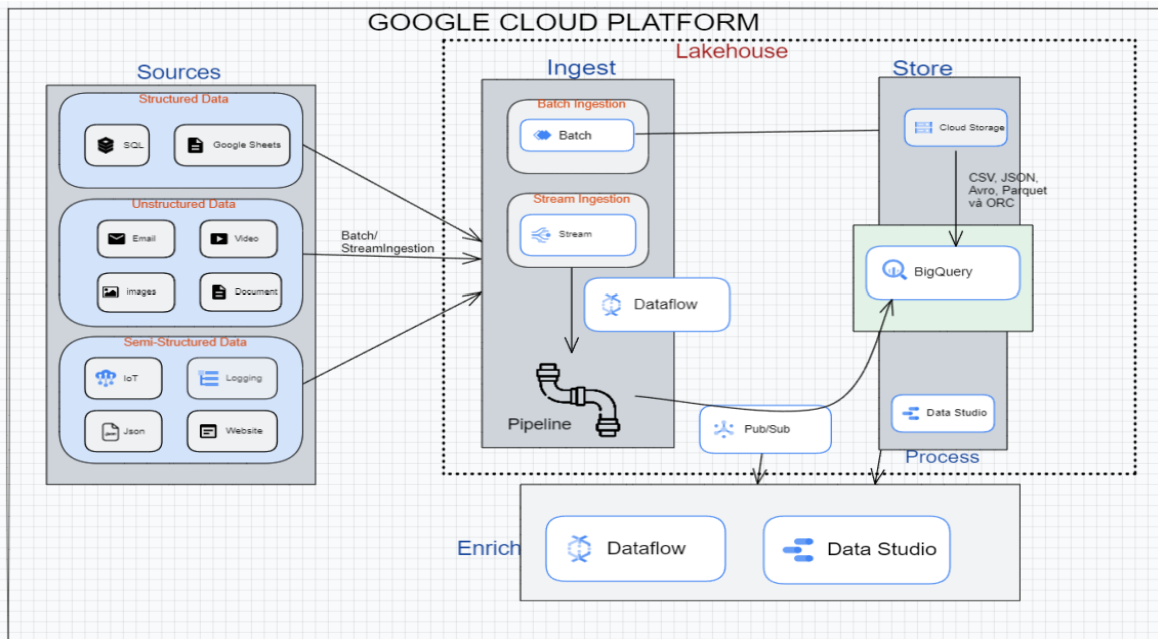
Triển khai kiến trúc lakehouse có thể phức tạp hơn và đòi hỏi kế hoạch và quản lý dữ liệu kỹ thuật hơn so với data warehouse hoặc data lake đơn lẻ. Tuy nhiên, nó cung cấp tính linh hoạt cho việc lưu trữ và xử lý dữ liệu lớn và không có cấu trúc.

2. Kiến trúc component

Kiến trúc component của công nghệ GCP trong việc khai thác, xử lý dữ liệu mạng xã hội có thể được chia thành ba lớp chính:

- Lớp thu thập dữ liệu: Dữ liệu được thu thập từ các nền tảng mạng xã hội. GCP cung cấp một số dịch vụ và công cụ để thu thập dữ liệu, bao gồm:
 - Google Cloud Dataproc: cung cấp môi trường để chạy các tác vụ xử lý dữ liệu hàng loạt, bao gồm thu thập dữ liệu từ các nền tảng mạng xã hội.
 - Google Cloud Natural Language API: cung cấp các khả năng xử lý ngôn ngữ tự nhiên để phân tích dữ liệu văn bản từ các nền tảng mạng xã hội.
 - Google Cloud Vision API: cung cấp các khả năng xử lý hình ảnh để phân tích dữ liệu hình ảnh từ các nền tảng mạng xã hội.
- Lớp lưu trữ dữ liệu: Chịu trách nhiệm lưu trữ dữ liệu đã thu thập. GCP cung cấp một số dịch vụ lưu trữ dữ liệu để lưu trữ dữ liệu mạng xã hội, bao gồm:
 - Google Cloud Storage: lưu trữ dữ liệu object.
 - Google Cloud Bigtable: lưu trữ dữ liệu NoSQL.
 - Google Cloud Spanner: lưu trữ dữ liệu SQL phân tán.
- Lớp xử lý dữ liệu: Xử lý các dữ liệu đã thu thập để trích xuất thông tin hữu ích. GCP cung cấp một số dịch vụ và công cụ gồm:
 - Google Cloud AI Platform: cung cấp các công cụ và dịch vụ để phát triển và triển khai các ứng dụng AI.

- Google Cloud AutoML: tự động hóa quá trình xây dựng các mô hình AI.
- Google Cloud Dataflow: cung cấp một nền tảng để chạy các tác vụ xử lý dữ liệu thời gian thực và hàng loạt.



Hình 2: Kiến trúc components

II. Các components

1. Data sources

1. Các loại dữ liệu

- Dữ liệu có cấu trúc (Structured Data)
 - + Bảng tính (Microsoft Excel, Google Trang tính).
 - + Cơ sở dữ liệu quan hệ.
 - + Mô hình dữ liệu được xác định trước.
- Dữ liệu phi cấu trúc (Unstructured Data)
 - + Tập văn bản (.doc, .txt).
 - + Email.
 - + Tập video (MP4, AVI hoặc MOV).
 - + Ảnh (jpg, tiff).
 - + Tập âm thanh (MP3, WAV hoặc FLAC).
- Dữ liệu bán cấu trúc (Semi-Structured Data)
 - + Các trang web (chứa cả dữ liệu có cấu trúc và không có cấu trúc. Thẻ HTML được coi là không có cấu trúc),
 - + Các bài đăng trên mạng xã hội (Twitter, Facebook hoặc Instagram, thường được lưu trữ dạng JSON hoặc XML).
 - + Tập nhật ký được tạo bởi hệ thống, ứng dụng hoặc thiết bị.

2. Thu thập dữ liệu thô

Raw data là dữ liệu thô hoặc không xử lý ban đầu thu thập từ các nguồn khác nhau. Đây là dữ liệu chưa qua bất kỳ quá trình biến đổi, làm sạch hay phân tích nào. Raw data thường được thu thập một cách tự động từ các hệ thống, thiết bị, hoặc nguồn dữ liệu khác.

Người dùng có thể tìm thấy raw data ở nhiều nguồn khác nhau, ví dụ:

- Hệ thống và ứng dụng nội bộ: Thu thập từ các hệ thống và ứng dụng trong tổ chức, như hệ thống quản lý khách hàng, hệ thống giám sát, máy chủ web, thiết bị IoT, và ứng dụng di động.
- Mạng xã hội và trang web công cộng: Cung cấp raw data đối với việc phân tích, theo dõi xu hướng, nghiên cứu thị trường và phân tích tương tác người dùng.
- Cảm biến và thiết bị IoT: Thu thập dữ liệu từ môi trường và các hoạt động. Ví dụ, cảm biến nhiệt độ, áp suất, độ ẩm, GPS, camera, v.v. gửi dữ liệu thô cho phân tích và lưu trữ.
- Dịch vụ bên ngoài: Các dịch vụ, nhà cung cấp dữ liệu và API có thể cung cấp raw data cho việc phân tích và ứng dụng khác. Ví dụ, dữ liệu thời tiết, dữ liệu tài chính, dữ liệu vận chuyển, v.v.
- Nhật ký và hệ thống ghi lại: cung cấp thông tin về hoạt động và sự kiện trong hệ thống, như đăng nhập, hoạt động mạng, hoạt động hệ điều hành, v.v.

Google Cloud Storage là dịch vụ lưu trữ đám mây của Google, cho phép lưu trữ và quản lý raw data một cách linh hoạt và bảo mật. Dưới đây là chi tiết cách GCS thu thập raw data và lưu trữ:

- Tạo bucket: Đầu tiên, ta cần tạo một bucket trong GCS để lưu trữ raw data. Mỗi bucket có thể chứa nhiều tệp tin hoặc đối tượng.
- Upload tệp tin: Sau khi tạo bucket, có thể tải lên các tệp tin raw data vào bucket này. Có nhiều cách để thực hiện việc tải lên, ví dụ như sử dụng giao diện người dùng trực tuyến của GCS, sử dụng công cụ dòng lệnh (gsutil) hoặc sử dụng API để tương tác với GCS.
 - + Giao diện người dùng trực tuyến: Dễ dàng nhấp chọn các thư mục và kéo thả tệp từ máy tính cục bộ
 - + gsutil: Đây là một chương trình Python cho phép ta truy cập GCS bằng các dòng lệnh. Ví dụ dòng lệnh tải dữ liệu từ máy tính vào GCS

- + JSON, XML: sử dụng các APIs cho phép khởi động HTTP POST để tải dữ liệu vào bucket hay folder
- + Cloud Storage Transfer Appliance: Một dịch vụ hỗ trợ khi bạn cần tải một lượng lớn dữ liệu.
- Quản lý phiên bản: Tự động duy trì các phiên bản trước đó của các tệp tin đã được tải lên, giúp theo dõi và phục hồi các phiên bản trước đó của dữ liệu nếu cần thiết.
- Định cấu trúc dữ liệu: Tổ chức dữ liệu trong bucket thành các thư mục và tệp tin con tùy ý. Điều này giúp quản lý và tìm kiếm dữ liệu một cách hiệu quả.
- Bảo mật: Cung cấp các công cụ bảo mật để đảm bảo an toàn cho dữ liệu. Có thể sử dụng chính sách truy cập, quyền hạn người dùng, mã hóa dữ liệu và tính năng xác minh danh tính để bảo vệ raw data khỏi truy cập trái phép.
- Quản lý dữ liệu: Có các tính năng quản lý dữ liệu như sao lưu, di chuyển, xóa và chia sẻ dữ liệu. Có thể lên kế hoạch và tổ chức quản lý dữ liệu theo yêu cầu cụ thể của dự án.
- Tích hợp với dịch vụ khác: Kết hợp tốt với các dịch vụ khác trên GCP như BigQuery, Cloud Dataflow, và AI Platform. Có thể truy cập raw data để tiếp tục xử lý, phân tích hoặc train mô hình máy học.

3. Ingest

- Batch Ingestion:

Nhập dữ liệu theo loạt (Batch Ingestion) liên quan đến việc nạp các tập dữ liệu lớn, có giới hạn, không cần xử lý trong thời gian thực.

Dữ liệu đã được nhận vào sau đó được truy vấn để tạo báo cáo hoặc kết hợp với các nguồn khác bao gồm cả dữ liệu thời gian thực.

Các batch load job của BigQuery là miễn phí, chỉ cần phải trả phí cho việc lưu trữ và truy vấn dữ liệu mà không phải trả phí cho việc nạp dữ liệu.

GCS được khuyến nghị là nơi lý tưởng để lưu trữ dữ liệu đầu vào. Nạp dữ liệu từ Cloud Storage vào BigQuery hỗ trợ nhiều định dạng tệp - CSV, JSON, Avro, Parquet và ORC.

- Stream Ingestion:

Nhập dữ liệu theo luồng (Streaming Ingestion) dùng cho yêu cầu phân tích lượng dữ liệu liên tục. Theo dõi sự kiện trên ứng dụng di động là một ví dụ cho mô hình này.

Thông qua Cloud Pub/Sub, ta có thể ghi lại tương tác của người dùng vào hệ và truyền chúng vào BigQuery bằng cách sử dụng công cụ như Cloud Dataflow để xử lý dữ liệu trên pipeline

Sau đó, phân tích dữ liệu này để xác định xu hướng tổng quan, như các khu vực có tương tác cao hoặc các vấn đề và theo dõi điều kiện lỗi trong thời gian thực.

BigQuery cho phép nhập dữ liệu theo luồng, truyền dữ liệu vào từng bản ghi một bằng phương thức `tabledata.insertAll`. API cho phép truyền dữ liệu không đồng bộ từ nhiều nhà sản xuất khác nhau.

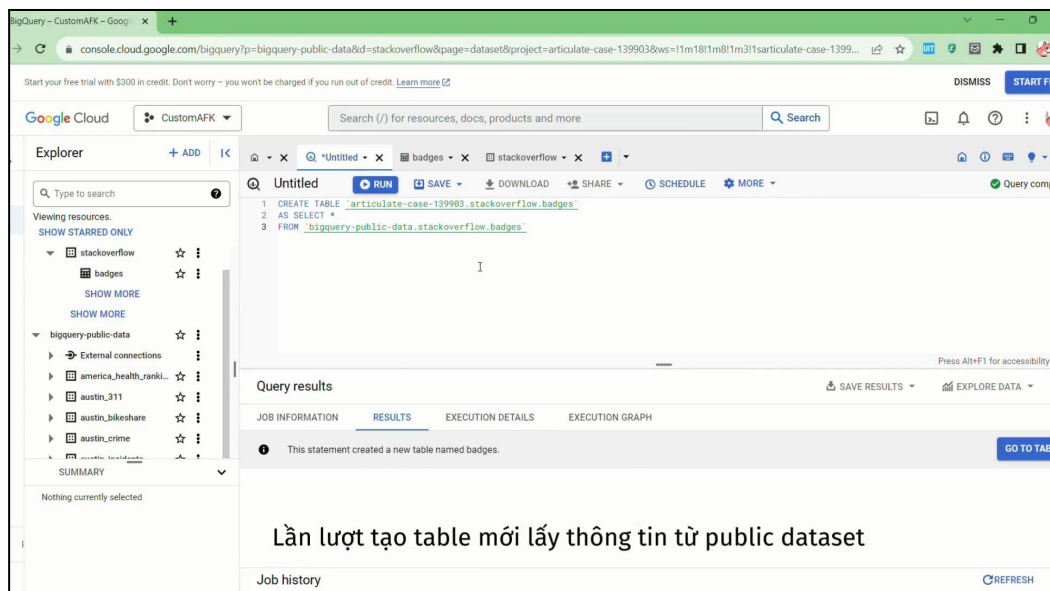
Một mô hình để nhập dữ liệu thời gian thực trên GCP là sử dụng pipeline Cloud Dataflow chạy ở chế độ theo luồng và ghi vào các bảng BigQuery sau khi quá trình xử lý cần thiết hoàn thành.

Ưu điểm pipeline Cloud Dataflow là tái sử dụng mã cùng cho cả hai hình thức stream và batch. Google sẽ quản lý công việc bắt đầu, thực thi và dừng các tài nguyên tính toán để xử lý pipeline song song.

Dưới đây là ví dụ về Ingest một bộ dữ liệu mà nhóm đã lựa chọn triển khai. Tổng cộng có 16 bảng để tạo mới, nhóm sẽ minh họa bằng bảng badges:

```
CREATE TABLE `articulate-case-139903.stackoverflow.badges`  
  
AS SELECT *  
  
FROM `bigquery-public-data.stackoverflow.badges`
```

Kết quả thu được như hình vẽ:



Hình 3: Minh họa Ingest

2. Process

a. Xử lý dữ liệu

Xử lý dữ liệu là quá trình thu thập, làm sạch, chuyển đổi dữ liệu. Xử lý dữ liệu là một phần quan trọng của quá trình phân tích dữ liệu, giúp chuẩn hóa dữ liệu để đưa vào phân tích, giúp doanh nghiệp đưa ra các quyết định kinh doanh sáng suốt dựa trên dữ liệu.

Để chuyển hóa và xử lý dữ liệu trên GCP thì ta có thể dùng Google BigQuery.

- **Xử lý dữ liệu trùng lặp:**

Để xử lý data trùng lặp GCP sử dụng function `DISTINCT()` trong BigQuery: Function này được sử dụng để trả về một tập hợp các giá trị duy nhất từ một bảng. Ví dụ:

```
SELECT DISTINCT name  
  
FROM customers;
```

Câu lệnh sẽ trả về một tập hợp các giá trị duy nhất cho cột name trong bảng customers.

- **Xử lý dữ liệu bị thiếu:**

Để xử lý data bị thiếu sử dụng function `ISNULL()` trong BigQuery: Function này được sử dụng để kiểm tra xem một giá trị có phải là NULL hay không. Ví dụ:

```
SELECT name,  
  
       age,  
  
       IFNULL(age, 0) AS age_filled  
  
FROM customers;
```

Câu lệnh sẽ trả về một tập dữ liệu bao gồm cột age_filled được điền bằng 0 cho các bản ghi có giá trị age là NULL.

- **Chuẩn hóa dữ liệu:**

- Để chuẩn hóa data sử dụng function CAST() trong BigQuery: Function này được sử dụng để chuyển đổi một giá trị từ một kiểu dữ liệu sang kiểu dữ liệu khác. Ví dụ:

```
SELECT name,  
       CAST(age AS INT64) AS age_int  
FROM customers;
```

Câu lệnh trả về một tập dữ liệu bao gồm cột age_int được chuyển đổi thành kiểu dữ liệu INT64.

- Để chuẩn hóa data sử dụng function TO_LOWER() trong BigQuery: Function này được sử dụng để chuyển đổi một giá trị thành chữ thường. Ví dụ:

```
SELECT name,  
       TO_LOWER(name) AS name_lower  
FROM customers;
```

Câu lệnh trả về một tập dữ liệu bao gồm cột name_lower được chuyển đổi thành chữ thường.

- Để chuẩn hóa data sử dụng function TRIM() trong BigQuery: Function này được sử dụng để loại bỏ các khoảng trắng thừa khỏi một giá trị. Ví dụ:

```
SELECT name,  
       TRIM(name) AS name_trimmed  
FROM customers;
```

Câu lệnh trả về một tập dữ liệu bao gồm cột name_trimmed được loại bỏ các khoảng trắng thừa.

- Để chuẩn hóa data sử dụng function TO_UPPER() trong BigQuery: Function này được sử dụng để chuyển đổi một giá trị thành chữ hoa. Ví dụ:

```
UPDATE customers  
SET name = TO_UPPER(name);
```

Câu lệnh này sẽ chuyển đổi tất cả các giá trị trong cột name thành chữ hoa.

- Để chuẩn hóa data sử dụng function TO_UPPER() trong BigQuery: Function này được sử dụng để chuyển đổi một giá trị thành chữ hoa. Ví dụ:

```
UPDATE customers  
SET name = TO_UPPER(name);
```

Câu lệnh này sẽ chuyển đổi tất cả các giá trị trong cột name thành chữ hoa.

- Để chuẩn hóa data sử dụng function REGEXP_EXTRACT() trong BigQuery: Function này được sử dụng để trích xuất một phần của một giá trị dựa trên một biểu thức chính quy. Ví dụ:

```
SELECT REGEXP_EXTRACT(email, r'@(.*)') AS domain  
FROM customers;
```


Câu lệnh sẽ trích xuất tên miền từ tất cả các địa chỉ email trong cột email.

- Để chuẩn hóa data sử dụng function REGEXP_REPLACE() trong BigQuery:
Function này được sử dụng để thay thế một phần của một giá trị bằng một giá trị khác dựa trên một biểu thức chính quy. Ví dụ:

```
UPDATE customers  
  
SET    email    =    REGEXP_REPLACE(email,    r'@gmail.com',  
    '@yahoo.com');
```

Câu lệnh này sẽ thay thế tất cả các địa chỉ email có miền gmail.com bằng miền yahoo.com.

Với dataset về Stack Overflow mà nhóm đã triển khai,

```
SELECT
```

```
    id,
```

```
    REGEXP REPLACE(text, r'\s+', '') AS text_without_extra_spaces
```

```
FROM `articulate-case-139903.stackoverflow.comments`
```

```
SELECT
```

```
    id,
```

```
    REGEXP REPLACE(text, r'^\p{L}\p{N}\p{P}\p{M}\p{S}\p{Z}+', '') AS  
text_without_non_unicode
```

```
FROM `articulate-case-139903.stackoverflow.comments`
```

+ Lưu vào bảng mới hoặc cập nhật lại bảng cũ:

```
CREATE TABLE 'articulate-case-139903.stackoverflow.new.comments' AS

SELECT

    id,

    creation_date,

    post_id,

    user_id.

    user_display_name.

    score.

    REGEXP REPLACE (text, r'^\p{L}\p{N}\p{P}\p{M}\p{S}\p{Z}+', '') AS
text_without_non_unicode

FROM

    `articulate-case-139903.stackoverflow.comments`:
```

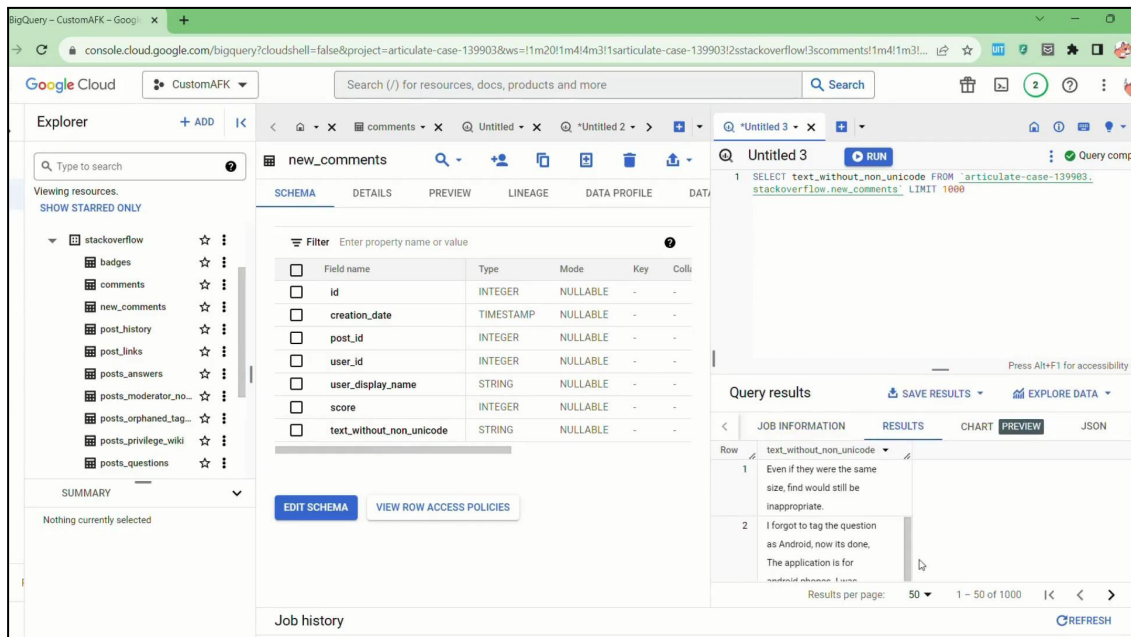
+ Kiểm tra:

```
SELECT text_without_non_unicode

FROM `articulate-case-139903.stackoverflow.hew_comments`

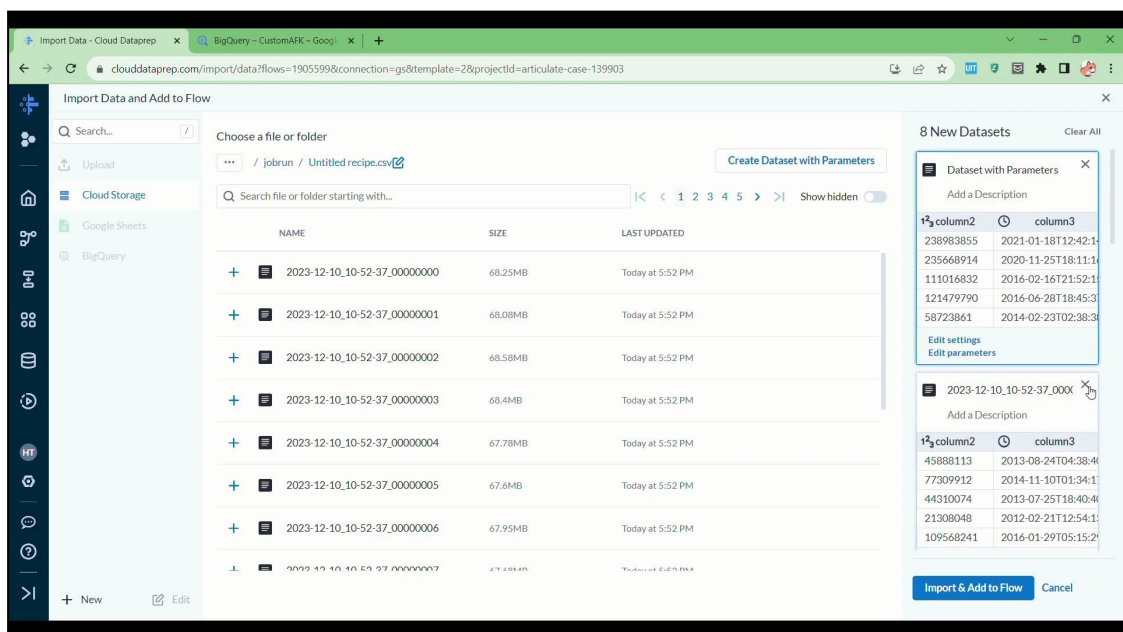
LIMIT 1000
```

Kết quả cho việc xử lý:



Hình 4: Minh họa cho chuẩn hóa dữ liệu ở comments

Ngoài ra, chúng em cũng thử dùng Dataprep để hỗ trợ cho việc Process (Hướng dẫn tại link video Youtube phần Process trong thư mục Technology).



Hình 5: Minh họa cho chuẩn hóa dữ liệu ở comments với Dataprep

b. Thống kê dữ liệu

Thống kê dữ liệu, là sử dụng các phương pháp để tạo ra thông tin hữu ích từ các dữ liệu. Với GCP, ta có thể sử dụng ngôn ngữ SQL trong dịch vụ BigQuery để truy vấn dữ liệu, từ đó vẽ biểu đồ và tạo báo cáo.

Để thống kê data sử dụng các function: COUNT(), SUM(), AVG(), MIN(), MAX(), STDDEV() và VAR() trong BigQuery.

- Sử dụng function COUNT() trong BigQuery: Để trả về số lượng bản ghi trong một tập kết quả. Ví dụ:

```
SELECT COUNT(*) AS total_customers  
FROM customers;
```

Câu lệnh này sẽ trả về tổng số bản ghi trong bảng customers.

- Sử dụng function SUM() trong BigQuery: Để trả về tổng của tất cả các giá trị trong một tập kết quả. Ví dụ:

```
SELECT SUM(age) AS total_age  
FROM customers;
```

Câu lệnh này sẽ trả về tổng số tuổi trong bảng customers.

- Sử dụng function AVG() trong BigQuery: Để trả về giá trị trung bình của tất cả các giá trị trong một tập kết quả. Ví dụ:

```
SELECT AVG(age) AS average_age  
FROM customers;
```

Câu lệnh này sẽ trả về tuổi trung bình trong bảng customers.

- Sử dụng function MIN() trong BigQuery: Để trả về giá trị nhỏ nhất trong một tập kết quả. Ví dụ:

```
SELECT MIN(age) AS min_age  
FROM customers;
```

Câu lệnh này sẽ trả về tuổi nhỏ nhất trong bảng customers.

- Sử dụng function MAX() trong BigQuery: Để trả về giá trị lớn nhất trong một tập kết quả. Ví dụ:

```
SELECT MAX(age) AS max_age  
FROM customers;
```

Câu lệnh này sẽ trả về tuổi lớn nhất trong bảng customers.

- Sử dụng function STDDEV() trong BigQuery: Để trả về độ lệch chuẩn của một tập kết quả. Ví dụ:

```
SELECT STDDEV(age) AS stddev_age  
FROM customers;
```

Câu lệnh này sẽ trả về độ lệch chuẩn của tuổi trong bảng customers.

- Sử dụng function VAR() trong BigQuery: Để trả về phương sai của một tập kết quả. Ví dụ:

```
SELECT VAR(age) AS variance_age  
FROM customers;
```

Câu lệnh này sẽ trả về phương sai của tuổi trong bảng customers.

Với thống kê về người dùng trong Stack Overflow, nhóm thực hiện code như sau:

- Lọc các giá trị null theo location:

```
select      location,      count(1)      as      Cnt      from
`bigquery-public-data.stackoverflow.users`

group by location

order by Cnt desc
```

- Trích xuất thông tin người dùng Stack Overflow:

```
select extract (year from last_access_date) as LastAccessedYear,

extract  (YEAR    FROM    CURRENT    DATE())-    EXTRACT(YEAR    FROM
last_access_date) as DornminantYears,

COUNT(1) As UserCnt

FROM `bigquery-public-data.stackoverflow.users`

group by 1, 2 order by 1 desc
```

```
select count(1), max(reputation), min(reputation), avg(reputation)

from `bigquery-public-data.stackoverflow.users`
```

```
select count(1) from      bigquery-public-data.stackoverflow.users`

where reputation <123
```

- Lọc ra reputation-group của users:

```
select bucket,  
  
format("%1-1, IFNULL(RANGES[SAFE_OFFSET(bucket-1)]+1,0) ranges SAFE  
DEESET (bucket) AS reputation_group)  
  
COUNT(*) AS COUNT  
  
FROM `bigquery-public-data.stackoverflow.users`,  
  
UNNEST([STRUCT([123, 200000, 400000, 500000, 600000, 700000, 800000,  
900000, 1000000, 1100000, 1200000 as ranges])),  
  
UNNEST([RANGE_BUCKET(reputation, ranges)])bucket  
  
group by 1,2  
  
order by bucket
```

Còn đối với xử lý các câu hỏi thường gặp trên Stack Overflow, nhóm đã thực hiện như sau:

- Xử lý theo County và TotalUser:

```
select  
  
case  
  
when location like '%United States%' or location LIKE 'USA' THEN  
'USA'  
  
when location like '%London%' or location LIKE 'United Kingdom' THEN  
'UK'  
  
when location like '%France%' THEN 'France'  
  
else location end as Country
```

```
count(1) as TotalUser FROM
`bigquery-public-data.stackoverflow.users`

where location is not null and reputation >200000 and reputation
<400000

group by Country

order by count(1) desc
```

- Thống kê ở posts_questions:

```
select category, count(*) as TagsTotal

from    bigquery-public-data.stackoverflow.posts_questions`

cross Join UNNEST (SPLIT(tags,'|')) as category

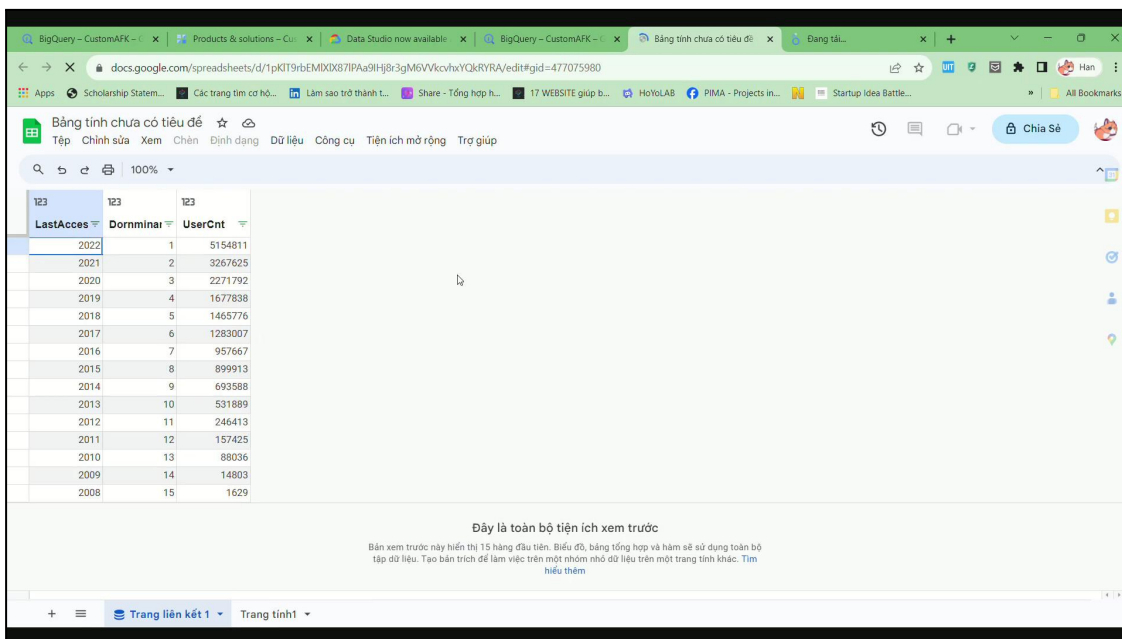
group by category

order by TagsTotal Desc
```

c. Trực quan hóa dữ liệu

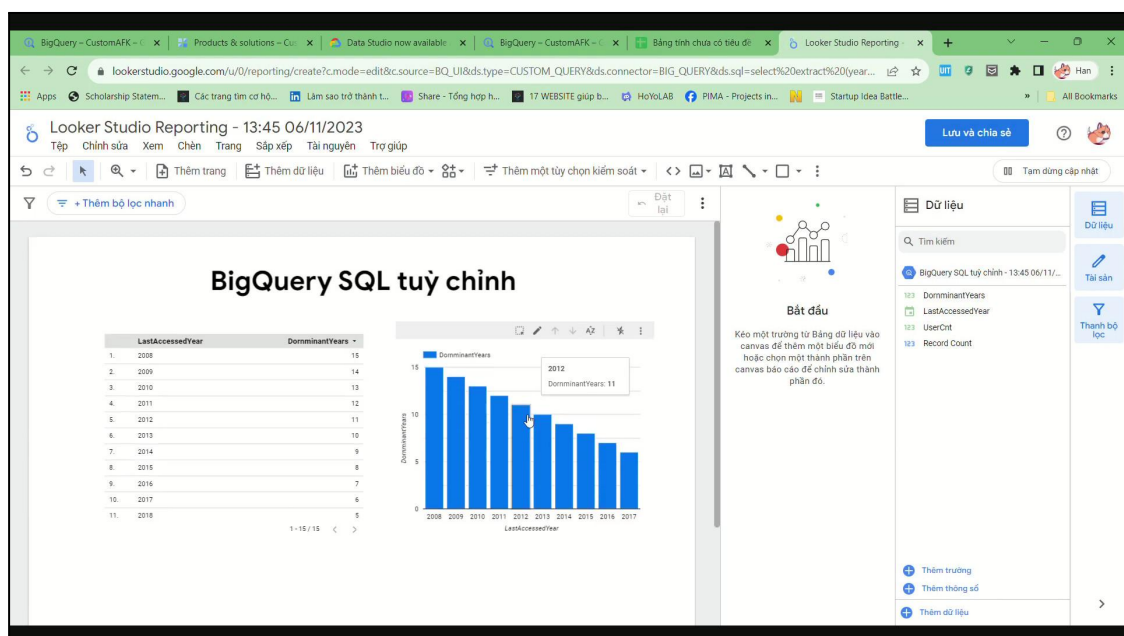
Nhằm dễ hiểu những dữ liệu phức tạp, phát hiện xu hướng và tạo biểu đồ,... ta cần trực quan hóa các dữ liệu đã được thu thập. Dịch vụ Google Data Studio cho phép tạo báo cáo và biểu đồ tương tác từ dữ liệu trong BigQuery và các nguồn khác trên GCP.

Ở Process đầu tiên về người dùng, đây là hình ảnh trực quan:



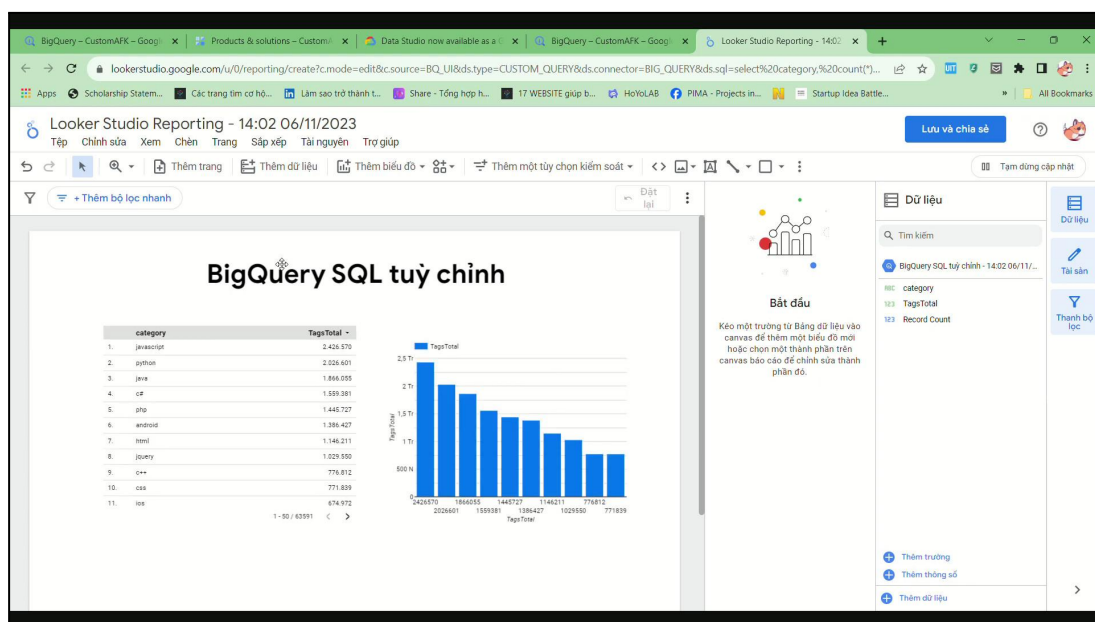
LastAccessedYear	DominantYears	UserCnt
2022	1	5154811
2021	2	3267625
2020	3	2271792
2019	4	1677838
2018	5	1465776
2017	6	1283007
2016	7	957667
2015	8	899913
2014	9	693588
2013	10	531889
2012	11	246413
2011	12	157425
2010	13	88036
2009	14	14803
2008	15	1629

Hình 6: Minh họa cho bảng xử lý thống kê users

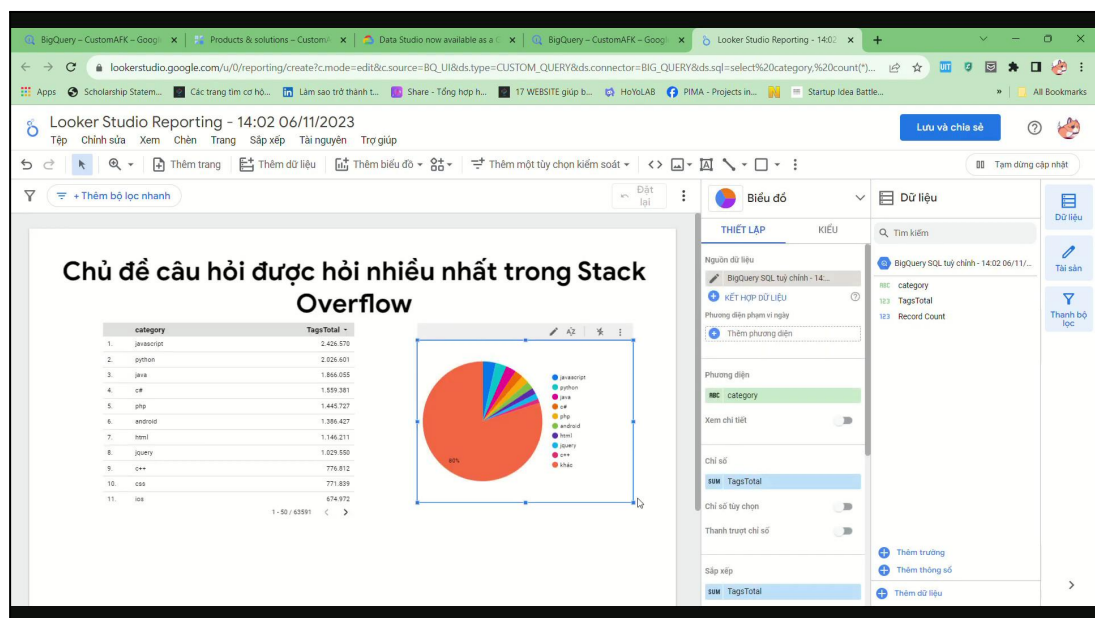


Hình 7: Minh họa cho trực quan hóa của xử lý thống kê users

Đối với xử lý các câu hỏi thường gặp, kết quả cũng được thu thập như sau:



Hình 8: Minh họa cho trực quan hóa dạng cột xử lý thống kê posts_questions

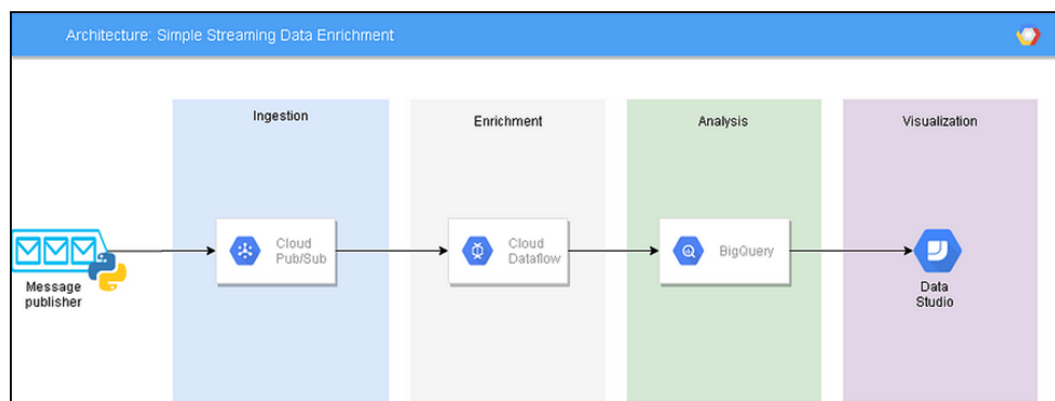


Hình 9: Minh họa cho trực quan hóa dạng tròn xử lý thống kê posts_questions

c. Enrich

Enrich data (hay còn gọi là làm giàu dữ liệu) là quá trình mở rộng dữ liệu bằng cách thêm thông tin hay chi tiết vào dữ liệu để làm dữ liệu cơ động và hữu ích hơn.

Quá trình này có thể bao gồm việc tập hợp thông tin từ nhiều nguồn khác nhau, làm sạch dữ liệu để loại bỏ dữ liệu trùng lặp hoặc không cần thiết, hoặc thêm dữ liệu bổ sung để làm cho dữ liệu ban đầu trở nên đa dạng hơn và cung cấp thông tin chi tiết hơn về đối tượng hoặc sự kiện mà dữ liệu đó đại diện.



Hình 10: Kiến trúc thành phần enrich

Với GCP, ta có thể sử dụng dịch vụ Dataflow để làm giàu dữ liệu, làm sạch các streaming data. Và thông qua sử dụng Dataflow SQL, ta có thể sử dụng cú pháp giống SQL để xác định và thực thi luồng làm việc xử lý dữ liệu. Tính năng này giúp đơn giản hóa quá trình tạo các pipeline và biến đổi dữ liệu mà không cần phải viết mã Python hoặc Java phức tạp.

Các bước enrich streaming data:

- Thiết lập các message publisher.
- Tạo pub/sub topic: Pub/Sub là một dịch vụ gửi thông điệp không đồng bộ, giúp lưu trữ thông điệp bền vững và gửi thông điệp theo thời gian thực. Có thể sử dụng GCP console để tạo.
- Thêm nguồn dữ liệu pub/sub và dựng schema cho nó.

- Tạo bảng BigQuery cùng dataset.
- Kết hợp dữ liệu từ BigQuery.
- Làm giàu dữ liệu bằng cách join các dữ liệu từ BigQuery đến dữ liệu streaming.
- Hợp thức hóa. Lúc này dữ liệu của bạn đã sẵn sàng cho các bước phân tích tiếp theo.
- Trực quan hóa. Có thể sử dụng Data Studio.

Sau khi làm giàu dữ liệu, dữ liệu được lưu trữ trong data lakehouse sẽ có thể được sử dụng để phân tích, báo cáo...

Nhóm tiến hành thực hiện Enrich chủ yếu với những dòng code như sau:

```
CREATE TABLE articulate-case-139903.stackoverflow.enriched_users AS

SELECT

    u.*,

    COUNT(q.id) AS total_questions

FROM

    articulate-case-139903.stackoverflow.users AS u

LEFT JOIN

    articulate-case-139903.stackoverflow.posts_questions AS q

ON

    u.id = q.owner_user_id

GROUP BY

    u.id, u.display_name, u.about_me, u.age, u.creation_date,

    u.last_access_date, u.location, u.reputation, u.up_votes,
```

```
u.down_votes, u.views, u.profile_image_url, u.website_url;
```

```
CREATE TABLE articulate-case-139903.stackoverflow.new_comments AS
```

```
SELECT
```

```
    id,
```

```
    creation_date,
```

```
    post_id,
```

```
    user_id,
```

```
    user_display_name,
```

```
    score,
```

```
    REGEXP_REPLACE(text, r'^\p{L}\p{N}\p{P}\p{M}\p{S}\p{Z}']', '') AS  
text_without_non_unicode
```

```
FROM
```

```
    articulate-case-139903.stackoverflow.comments;
```

```
SELECT
```

```
    id,
```

```
    REGEXP_REPLACE(text, r'\s+', ' ') AS text_without_extra_spaces
```

```
FROM articulate-case-139903.stackoverflow.comments
```

```
CREATE TABLE articulate-case-139903.stackoverflow.enriched_users AS
```

```
SELECT
```

```
    u.*,
```

```

COUNT(q.id) AS total_questions

FROM

articulate-case-139903.stackoverflow.users AS u

LEFT JOIN

articulate-case-139903.stackoverflow.posts_questions AS q

ON

u.id = q.owner_user_id

GROUP BY

u.id, u.display_name, u.about_me, u.age, u.creation_date,

u.last_access_date, u.location, u.reputation, u.up_votes,

u.down_votes, u.views, u.profile_image_url, u.website_url;

```

Sau đó chúng ta có được các hình ảnh output:

Đã tạo xong table mới

Hình 11: Minh họa tạo bảng ở phần Enrich

Query results

Row	reputation	up_votes	down_votes	views	profile_image_url	website_url	total_questions
1	43934	1860	43	4569	https://stack.imgur.com/Yw9...	https://medium.com/@ViaCog...	1162
2	52314	1590	300	3662	https://www.gravatar.com/ava... tar/ea8f12a687ab83fec7cd25... 94ef2ce81c?	null	1192

Kiểm tra các user có hơn 1000 câu hỏi -> có ích cho việc phân tích sau này

Hình 12: Minh họa kiểm tra các user có hơn 1000 câu hỏi

Chương 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Qua quá trình thực hiện đề tài, nhóm đã tìm hiểu được những thành phần của công nghệ dữ liệu lớn và dịch vụ Google Cloud Platform. Nhóm đã biết được các thành phần theo kiến trúc dữ liệu lớn theo kiến trúc Lakehouse, từ data sources, ingest cho đến process và enrich để phục vụ cho việc xử lý dữ liệu lớn

Theo những kiến thức đã tích lũy được, nhóm đã áp dụng các bước triển khai, xử lý dữ liệu mạng xã hội. Chúng em đã được biết qua một số dịch vụ trên Google Cloud để tiến hành thực hiện xử lý dữ liệu, tiêu biểu như Storage, BigQuery, Dataflow,...

Tuy nhiên, do thời lượng hạn chế của môn học, nhóm chúng em phần lớn chỉ đạt được mức tìm hiểu tài liệu và triển khai theo những tài liệu mà công nghệ có hướng dẫn. Nhìn chung, nhóm đã cố gắng tìm hiểu và chạy thử các bước xử lý dữ liệu một cách tốt nhất có thể.

Từ việc demo dữ liệu, chúng em đã mô phỏng theo được một số kỹ thuật thao tác trên Google Cloud cơ bản. Trong tương lai, nếu có cơ hội chúng em đã nghiên cứu kỹ hơn để ứng dụng vào những bài toán dữ liệu lớn trong thực tiễn sau này.

TÀI LIỆU THAM KHẢO

- [1] Bap-Software, “Các công nghệ Big Data hàng đầu mà bạn cần biết” [Accessed 26 September 2023]. Available: <https://bap-software.net/vi/knowledge/top-big-data-technologies/>
- [2] Bkhost, “Google Cloud Platform là gì? GCP dùng để làm gì?” [Accessed 26 September 2023]. Available: <https://bkhost.vn/blog/gcp-google-cloud-platform/>
- [3] Pluralsight, “What is Google Cloud Platform (GCP)?” [Accessed 26 September 2023]. Available: <https://www.pluralsight.com/resources/blog/cloud/what-is-google-cloud-platform-gcp>
- [4] Cloudaz, “Google Cloud Platform là gì? Ưu, nhược và các trường hợp sử dụng của GCP” [Accessed 26 September 2023]. Available: <https://cloudaz.io/google-cloud-platform-la-gi-uu-nhuoc-va-cac-truong-hop-su-dung-cua-gcp/>
- [5] Google Cloud, “Google Cloud Architecture Framework” [Accessed 26 September 2023]. Available: <https://cloud.google.com/architecture/framework>
- [6] Google Cloud, “Google Cloud Architecture Framework: System design” [Accessed 26 September 2023]. Available: <https://cloud.google.com/architecture/framework/system-design>
- [7] Google Cloud, “Google Cloud Architecture Framework: Operational excellence” [Accessed 26 September 2023]. Available: <https://cloud.google.com/architecture/framework/operational-excellence>
- [8] Google Cloud, “Google Cloud Architecture Framework: Security, privacy, and compliance” [Accessed 26 September 2023]. Available: <https://cloud.google.com/architecture/framework/security>

[9] Google Cloud, “Google Cloud Architecture Framework: Reliability”

[Accessed 26 September 2023]. Available:

<https://cloud.google.com/architecture/framework/reliability>

[10] Google Cloud, “Google Cloud Architecture Framework: Cost optimization”

[Accessed 26 September 2023]. Available:

<https://cloud.google.com/architecture/framework/cost-optimization>

[11] Google Cloud, “Google Cloud Architecture Framework: Performance optimization” [Accessed 26 September 2023]. Available.

<https://cloud.google.com/architecture/framework/performance-optimization>

[12] Genz, “Google Cloud Storage là gì?” [Accessed 8 October 2023].

Available: <https://genz.edu.vn/google-storage-la-gi/>

[13] Google Cloud Blog, “How to load, import, or ingest data into BigQuery for analysis” [Accessed 9 October 2023]. Available:

<https://cloud.google.com/blog/topics/developers-practitioners/bigquery-explained-data-ingestion?hl=en>

[14] Cloud Ace, “Run PySpark trên Dataproc như thế nào?” [Accessed 15 October 2023]. Available:

<https://blog.cloud-ace.vn/run-pyspark-tren-dataproc-nhu-the-nao/>

[15] Google Cloud Blog, “5-ish ways to get your data into Cloud Storage”

[Accessed 16 October 2023]. Available:

<https://cloud.google.com/blog/topics/developers-practitioners/5-ish-ways-get-your-data-cloud-storage>

[16] Google Cloud, “Loading CSV data from Cloud Storage” [Accessed 16 October 2023]. Available:

<https://cloud.google.com/bigquery/docs/loading-data-cloud-storage-csv>