

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA HỆ THỐNG THÔNG TIN**



**BÁO CÁO BÀI TẬP QUÁ TRÌNH**  
**MÔN HỌC: ĐIỆN TOÁN Đám MÂY**

**ĐỀ TÀI**

**Sử dụng Azure Synapse Analytics để truy vấn Data Lake**  
**(Using Azure Synapse Analytics to Query Data Lake)**

**Lớp:** IS402.O21.HTCL

**GVHD:** ThS. Hà Lê Hoài Trung

**Nhóm sinh viên thực hiện:**

Phan Chí Cường                      21520673

Nguyễn Minh Duy                    21522005

**TP. HỒ CHÍ MINH, 2024**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA HỆ THỐNG THÔNG TIN**



**BÁO CÁO BÀI TẬP QUÁ TRÌNH**  
**MÔN HỌC: ĐIỆN TOÁN Đám MÂY**

**ĐỀ TÀI**

**Sử dụng Azure Synapse Analytics để truy vấn Data Lake**  
**(Using Azure Synapse Analytics to Query Data Lake)**

**Lớp:** IS402.O21.HTCL

**GVHD:** ThS. Hà Lê Hoài Trung

**Nhóm sinh viên thực hiện:**

Phan Chí Cường                      21520673

Nguyễn Minh Duy                    21522005

**TP. HỒ CHÍ MINH, 2024**

## MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU BÀI TOÁN .....	5
1.1. Giới thiệu bài toán.....	5
1.2. Giới thiệu loại dữ liệu .....	5
1.3. Kích thước dữ liệu – dung lượng bộ nhớ cho xử lý .....	6
1.4. So sánh trong trường hợp máy chủ truyền thống và hệ thống cloud.....	9
1.4.1. So sánh về tốc độ xử lý dữ liệu lớn.....	9
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT .....	10
2.1. Tổ chức lưu trữ dữ liệu .....	10
2.1.1. Định dạng dữ liệu .....	10
2.1.2. Kích thước dữ liệu và tối ưu hoá tệp dữ liệu .....	10
2.1.3. Cấu trúc dữ liệu.....	11
2.1.4. Cách tổ chức dữ liệu.....	11
2.2. Thuật toán xử lý, luồng xử lý và truy xuất dữ liệu .....	11
2.2.1. Thuật toán xử lý .....	11
2.2.2. Luồng xử lý dữ liệu .....	12
2.2.3. Truy xuất dữ liệu .....	12
2.3. Các dịch vụ Cloud đã sử dụng .....	13
2.3.1. Azure Synapse Analytics.....	13
2.3.2. Azure Data Lake Storage (ADLS) .....	13
2.3.3. Azure Data Factory .....	14
2.3.4. Azure Blob Storage.....	14
2.4. Đọc tài liệu khoa học.....	14
2.4.1. Ý tưởng .....	14

2.4.2.	Điểm mạnh.....	15
2.4.3.	Điểm yếu .....	16
CHƯƠNG 3.	MÔ HÌNH DỮ LIỆU .....	17
3.1.	Tốc độ đọc/ghi dữ liệu .....	17
3.2.	Thiết lập luồng xử lý dữ liệu tự động ETL.....	19
3.3.	Tối ưu hóa lưu trữ và cải thiện tốc độ đọc/ghi .....	21
CHƯƠNG 4.	XÂY DỰNG ETL PIPELINE TRÊN AZURE DATA FACTORY .....	23
4.1.	Trường hợp 1 .....	23
4.2.	Trường hợp 2 .....	27
TÀI LIỆU THAM KHẢO .....		36

# CHƯƠNG 1. GIỚI THIỆU BÀI TOÁN

## 1.1. Giới thiệu bài toán

Bài toán thuộc nhóm **xử lý và phân tích dữ liệu lớn (Big Data Analytics)** trên nền tảng **điện toán đám mây (Cloud Computing)**, với mục tiêu khai thác, truy vấn và phân tích dữ liệu có khối lượng lớn một cách **hiệu quả, nhanh chóng và tối ưu chi phí**.

Trong thời đại số hóa, dữ liệu từ các hệ thống thương mại điện tử, IoT, log web, mạng xã hội... liên tục được tạo ra với quy mô lớn và có cấu trúc đa dạng. Việc xử lý loại dữ liệu này đòi hỏi một nền tảng mạnh mẽ, linh hoạt và có khả năng mở rộng. Để đáp ứng yêu cầu đó, **Azure Synapse Analytics** được lựa chọn là giải pháp trung tâm.

Azure Synapse là nền tảng phân tích hợp nhất được Microsoft phát triển, cho phép truy vấn và phân tích trực tiếp dữ liệu lưu trữ trên **Azure Data Lake Storage (ADLS)** mà không cần di chuyển dữ liệu. Nền tảng này hỗ trợ ba công cụ phân tích mạnh mẽ:

- **Serverless SQL Pools:** Cho phép truy vấn dữ liệu linh hoạt mà không cần cấu hình trước tài nguyên tính toán, chỉ tính phí theo lượng dữ liệu đã quét.
- **Dedicated SQL Pools:** Cung cấp hiệu năng cao khi làm việc với tập dữ liệu lớn, cho phép tối ưu hóa thông qua các bảng phân tán và chỉ mục cột (columnstore).
- **Apache Spark Pools:** Hỗ trợ xử lý dữ liệu phi cấu trúc và tích hợp học máy (Machine Learning), phù hợp với log web, dữ liệu IoT, và các mô hình AI.

Azure Synapse giúp doanh nghiệp chuyển đổi từ hệ thống xử lý truyền thống sang mô hình phân tích hiện đại, hỗ trợ ra quyết định **theo thời gian thực (real-time)**, mang lại lợi thế cạnh tranh rõ rệt.

## 1.2. Giới thiệu loại dữ liệu

Dữ liệu được lưu trữ trên **Azure Data Lake Storage (ADLS)**, hỗ trợ nhiều định dạng hiện đại như:

- **Parquet:** Định dạng cột tối ưu về dung lượng và tốc độ truy vấn.
- **JSON, CSV, Avro:** Phù hợp với các ứng dụng phổ thông, dễ tích hợp với các hệ thống khác.

#### **Nguồn dữ liệu tiêu biểu:**

- **Log truy cập web:** Dữ liệu về hành vi người dùng, session, pageview, clickstream.
- **Giao dịch thương mại điện tử:** Bao gồm sản phẩm, giá, thời gian, trạng thái đơn hàng.
- **Dữ liệu cảm biến IoT:** Từ hàng nghìn thiết bị gửi về liên tục, bao gồm nhiệt độ, vị trí, trạng thái.
- **Dữ liệu phi cấu trúc/bán cấu trúc:** Văn bản, chuỗi JSON lồng nhau, ảnh, video metadata...

#### **Đặc điểm dữ liệu:**

- **Khối lượng lớn (Volume):** Từ vài TB đến hàng PB.
- **Tốc độ cao (Velocity):** Liên tục cập nhật theo thời gian thực.
- **Đa dạng định dạng (Variety):** Cấu trúc, bán cấu trúc, phi cấu trúc.

Tính chất này biến bài toán thành một **thách thức lý tưởng** cho các hệ thống xử lý dữ liệu lớn như Azure Synapse Analytics.

### **1.3. Kích thước dữ liệu – dung lượng bộ nhớ cho xử lý**

Trong các hệ thống phân tích dữ liệu lớn, đặc biệt là khi tích hợp với các ứng dụng web, lượng dữ liệu cần xử lý có thể rất lớn. Ví dụ, một nền tảng thương mại điện tử có thể thu thập hàng tỷ bản ghi về hành vi người dùng, lịch sử giao dịch và phản hồi từ khách hàng. Khi lượng dữ liệu tăng nhanh theo thời gian, việc truy vấn và xử lý dữ liệu từ **Data Lake** trở thành một thách thức lớn, đòi hỏi một giải pháp có khả năng mở rộng tốt.

#### **Ví dụ về khối lượng dữ liệu trong hệ thống thực tế**

Trong một doanh nghiệp lớn, dữ liệu trong **Data Lake** có thể đạt kích thước từ vài **terabyte (TB)** đến hàng **petabyte (PB)**. Các tập dữ liệu này bao gồm:

- **Log truy cập web** từ hàng triệu người dùng mỗi ngày, với mỗi sự kiện chiếm vài KB dữ liệu.
- **Giao dịch mua bán** với thông tin chi tiết về sản phẩm, giá cả và thời gian mua hàng, với ~1 KB/giao dịch
- **Dữ liệu cảm biến IoT**, thu thập liên tục từ hàng nghìn thiết bị, có thể tạo ra hàng tỷ bản ghi mỗi ngày.

Khi thực hiện các tác vụ phân tích nâng cao như phân cụm khách hàng, dự đoán hành vi, hay phân tích thời gian thực, dữ liệu cần xử lý mỗi ngày có thể lên đến **hàng chục terabyte**, đòi hỏi hệ thống có khả năng:

- **Mở rộng theo nhu cầu (Elastic scalability)**
- **Phân tích tức thì (Low latency)**
- **Tối ưu chi phí (Cost efficiency)**

### **Xử lý dữ liệu với Azure Synapse Analytics**

Để đảm bảo hiệu suất khi truy vấn lượng dữ liệu lớn này, **Azure Synapse Analytics** cung cấp các cơ chế sau:

- **Serverless SQL Pool**
  - Cho phép chạy các truy vấn SQL trực tiếp trên dữ liệu lưu trữ trong **Azure Data Lake**, không cần di chuyển dữ liệu.
  - Tự động điều chỉnh tài nguyên, chỉ tính phí theo lượng dữ liệu đã quét, giúp tối ưu chi phí.
  - Hỗ trợ **Parquet, JSON, CSV**, giúp giảm kích thước lưu trữ và tăng tốc độ truy vấn.

- **Dedicated SQL Pool**

- Cho phép phân tán dữ liệu thành nhiều **distributed tables**, giúp tối ưu hiệu suất khi xử lý tập dữ liệu lớn.
- Hỗ trợ **Columnstore Indexing**, giúp giảm dung lượng lưu trữ và tăng tốc độ truy vấn gấp nhiều lần.

- **Apache Spark for Synapse**

- Hỗ trợ xử lý dữ liệu song song (MPP), giúp tăng tốc độ phân tích dữ liệu phi cấu trúc như log web, dữ liệu IoT.
- Tích hợp với **ML & AI**, giúp xây dựng các mô hình dự đoán ngay trên dữ liệu lưu trữ trong Data Lake.

**Ưu điểm của kiến trúc MPP(Massively Parallel Processing):**

- Chia truy vấn thành nhiều phần nhỏ.
- Thực thi song song trên nhiều node tính toán.
- Tối ưu hóa hiệu suất xử lý so với kiến trúc tuần tự truyền thống.

**Lợi ích của Azure Synapse Analytics trong xử lý dữ liệu lớn**

- **Hiệu suất cao:** Nhờ khả năng mở rộng linh hoạt và xử lý song song, Azure Synapse giúp giảm thời gian truy vấn từ **vài giờ xuống chỉ còn vài phút hoặc giây**.
- **Tối ưu chi phí:** Mô hình tính phí dựa trên lượng dữ liệu quét giúp doanh nghiệp chỉ trả tiền cho tài nguyên thực sự sử dụng.
- **Dễ tích hợp với ứng dụng web:** Các kết quả truy vấn có thể được sử dụng ngay trong dashboard BI hoặc API web, giúp cung cấp thông tin theo thời gian thực.



#### 1.4. So sánh trong trường hợp máy chủ truyền thống và hệ thống cloud

Việc xử lý và phân tích dữ liệu lớn có thể được triển khai trên cả hệ thống **máy chủ truyền thống (on-premises)** và **điện toán đám mây (cloud)**. Tuy nhiên, hai mô hình này có sự khác biệt rõ rệt về khả năng mở rộng, hiệu suất, chi phí và bảo trì.

##### 1.4.1. So sánh về tốc độ xử lý dữ liệu lớn

Tiêu chí	Máy chủ truyền thống (On-premises)	Hệ thống cloud (Azure Synapse Analytics)
<b>Khả năng xử lý dữ liệu lớn</b>	Giới hạn bởi tài nguyên phần cứng, có thể chậm khi dữ liệu vượt quá khả năng xử lý của máy chủ.	Sử dụng kiến trúc <b>MPP (Massively Parallel Processing)</b> để xử lý nhiều tập dữ liệu cùng lúc, giúp tăng tốc đáng kể.
<b>Tốc độ truy vấn SQL</b>	Truy vấn SQL có thể chậm nếu dữ liệu lớn do giới hạn CPU và RAM.	<b>Dedicated SQL Pool</b> trong Azure Synapse cho phép chạy các truy vấn SQL trên dữ liệu lớn với tốc độ cao.
<b>Tích hợp caching</b>	Phải thiết lập thủ công cơ chế caching, dễ bị quá tải nếu truy vấn lặp lại nhiều lần.	Hỗ trợ <b>caching tự động</b> , giúp tăng tốc các truy vấn lặp lại trên cùng một tập dữ liệu.
<b>Độ trễ xử lý</b>	Độ trễ cao nếu số lượng người dùng tăng đột biến, vì tài nguyên cố định không thể tự động mở rộng.	Độ trễ thấp nhờ khả năng <b>scale-out</b> tự động khi có nhiều truy vấn đồng thời.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1. Tổ chức lưu trữ dữ liệu

#### 2.1.1. Định dạng dữ liệu

Trong hệ sinh thái Azure, đặc biệt là khi làm việc với Azure Data Lake và Azure Synapse Analytics, việc lựa chọn định dạng lưu trữ dữ liệu là yếu tố quan trọng nhằm tối ưu hóa hiệu suất truy vấn và dung lượng lưu trữ. Một số định dạng phổ biến bao gồm:

- **Parquet:** Định dạng cột, hỗ trợ nén mạnh và tối ưu cho phân tích dữ liệu lớn. Đây là định dạng được khuyến nghị khi làm việc với Azure Synapse do tốc độ truy vấn và khả năng nén cao.
- **CSV (Comma-Separated Values):** Dễ sử dụng và tương thích rộng rãi, nhưng chiếm nhiều dung lượng và không tối ưu cho khối lượng dữ liệu lớn.
- **JSON (JavaScript Object Notation):** Phù hợp với dữ liệu bán cấu trúc, hỗ trợ tốt cho việc lưu trữ các đối tượng phức tạp.
- **Avro:** Thường được sử dụng cho các luồng dữ liệu truyền trực tuyến (streaming) nhờ khả năng tuần tự hóa hiệu quả.

Khi sử dụng **Azure Synapse Analytics**, **Parquet** là lựa chọn tối ưu do khả năng nén và truy vấn nhanh hơn so với CSV hoặc JSON.

#### 2.1.2. Kích thước dữ liệu và tối ưu hoá tệp dữ liệu

Dữ liệu trong Azure Data Lake có thể dao động từ vài megabyte đến hàng terabyte, phụ thuộc vào ứng dụng cụ thể. Một số chiến lược tối ưu hóa bao gồm:

- **Chia nhỏ tệp:** Khuyến nghị chia dữ liệu thành các tệp có kích thước khoảng 100MB đến 1GB để tăng tốc độ truy vấn và phân tán xử lý.
- **Tránh tệp lớn hơn 10GB:** Các tệp quá lớn có thể gây tắc nghẽn I/O và ảnh hưởng đến hiệu suất hệ thống.

### 2.1.3. Cấu trúc dữ liệu

Dữ liệu trong Data Lake thường được chia thành ba loại chính:

- **Dữ liệu có cấu trúc (Structured):** Dữ liệu dạng bảng như trong SQL, Parquet.
- **Dữ liệu bán cấu trúc (Semi-structured):** Dữ liệu dưới dạng JSON, XML, Avro.
- **Dữ liệu phi cấu trúc (Unstructured):** Bao gồm hình ảnh, video, tệp văn bản hoặc log.

Azure Synapse Analytics chủ yếu xử lý dữ liệu có cấu trúc và bán cấu trúc thông qua các truy vấn SQL và Spark.

### 2.1.4. Cách tổ chức dữ liệu

Để đảm bảo hiệu suất và khả năng mở rộng, dữ liệu trong Data Lake thường được tổ chức theo các nguyên tắc:

- **Cấu trúc phân cấp thư mục (Folder Hierarchy):** Dữ liệu được sắp xếp theo cây thư mục dạng năm/tháng/ngày để dễ dàng truy cập.
- **Phân vùng (Partitioning):** Dữ liệu được chia theo các thuộc tính như thời gian, khu vực nhằm tối ưu truy vấn.
- **Quản lý siêu dữ liệu (Metadata Management):** Sử dụng Azure Data Catalog để lập danh mục, gán nhãn và quản lý dữ liệu giúp nâng cao khả năng tìm kiếm và phân tích.

## 2.2. Thuật toán xử lý, luồng xử lý và truy xuất dữ liệu

### 2.2.1. Thuật toán xử lý

Trong **Azure Synapse Analytics**, thuật toán xử lý dữ liệu trong **Data Lake** có thể chia thành hai nhóm chính:

- **Xử lý theo lô (Batch Processing):** Dữ liệu được xử lý theo từng khối lớn, phù hợp với các tác vụ như tổng hợp, phân tích lịch sử.
- **Xử lý thời gian thực (Stream Processing):** Dữ liệu được xử lý gần như theo thời gian thực, hữu ích cho các trường hợp như giám sát hệ thống hoặc phân tích dữ liệu liên tục.

Một số thuật toán và kỹ thuật phổ biến:

- **MapReduce:** Chia nhỏ dữ liệu thành các phần nhỏ hơn để xử lý song song, sau đó tổng hợp kết quả.
- **Tối ưu truy vấn (Query Optimization):** Tối ưu truy vấn bằng cách sử dụng **cost-based optimizer, caching, adaptive query execution**.
- **Pipeline học máy (ML Pipelines):** Sử dụng Spark ML hoặc Azure Machine Learning để phân tích dữ liệu.

### 2.2.2. Luồng xử lý dữ liệu

Quy trình xử lý dữ liệu tổng thể trong Azure Synapse thường gồm các bước:

- 1) **Ingest (Thu nạp dữ liệu):** Nhập dữ liệu từ Azure Data Lake Storage, Azure Blob Storage, hoặc các nguồn khác được đưa vào Synapse.
- 2) **Transform (Biến đổi dữ liệu):** Sử dụng SQL hoặc Spark để làm sạch, chuyển đổi dữ liệu.
- 3) **Store (Lưu trữ):** Dữ liệu sau xử lý được lưu vào Synapse SQL Pool hoặc lưu trữ lại Data Lake.
- 4) **Analyze (Phân tích):** Dữ liệu được truy vấn bằng SQL, Spark hoặc tích hợp với Power BI để phân tích và trực quan hóa.

### 2.2.3. Truy xuất dữ liệu

Azure Synapse cung cấp nhiều phương pháp để truy xuất dữ liệu từ Data Lake:

- **Serverless SQL Pool:** Truy vấn dữ liệu trực tiếp từ Data Lake mà không cần tải dữ liệu vào hệ thống.

- **Dedicated SQL Pool:** Lưu trữ dữ liệu có cấu trúc để truy vấn nhanh hơn, truy vấn dữ liệu đã được nạp vào với hiệu suất cao hơn.
- **Spark Pool:** Sử dụng Apache Spark để chạy các truy vấn phân tán trên khối lượng dữ liệu lớn.
- **Synapse Pipelines:** Tự động hóa quy trình ETL để tải và chuyển đổi dữ liệu, kết nối linh hoạt giữa các dịch vụ.

### 2.3. Các dịch vụ Cloud đã sử dụng

Trong quá trình sử dụng Azure Synapse Analytics để truy vấn Data Lake, chúng ta cần đến một số dịch vụ Cloud chính của Microsoft Azure, bao gồm:

#### 2.3.1. Azure Synapse Analytics

- **Vai trò:** Dịch vụ phân tích dữ liệu lớn, cho phép truy vấn dữ liệu trên Data Lake một cách linh hoạt bằng SQL hoặc Spark.
- **Tính năng chính:**
  - **Serverless SQL Pool:** Truy vấn dữ liệu trực tiếp từ Azure Data Lake Storage mà không cần nạp dữ liệu.
  - **Dedicated SQL Pool:** Hỗ trợ lưu trữ và truy vấn dữ liệu lớn với hiệu suất cao.
  - **Apache Spark Pool:** Chạy các tác vụ big data processing với Spark trên Data Lake.

#### 2.3.2. Azure Data Lake Storage (ADLS)

- **Vai trò:** Kho lưu trữ dữ liệu lớn, hỗ trợ lưu trữ dữ liệu có cấu trúc, phi cấu trúc và bán cấu trúc.
- **Tính năng chính:**
  - Hỗ trợ lưu trữ khối lượng dữ liệu lớn theo **dạng file-based (Parquet, CSV, JSON, ORC, Avro)**.
  - Kết hợp với **Synapse Analytics** để truy vấn dữ liệu mà không cần di chuyển nó.
  - Hỗ trợ **Hierarchical Namespace (HNS)** để tổ chức dữ liệu theo thư mục.

### 2.3.3. Azure Data Factory

- **Vai trò:** Công cụ tích hợp dữ liệu (ETL) để di chuyển và biến đổi dữ liệu từ nhiều nguồn vào Data Lake, công cụ mạnh mẽ giúp tích hợp và tự động hóa quy trình dữ liệu.
- **Tính năng chính:**
  - **Data Pipeline:** Xây dựng các quy trình thu thập, xử lý và di chuyển dữ liệu tự động.
  - **Data Flow:** Chuyển đổi dữ liệu bằng giao diện trực quan mà không cần viết code.
  - **Hỗ trợ nhiều nguồn dữ liệu** như SQL Server, Blob Storage, API, v.v.

### 2.3.4. Azure Blob Storage

- **Vai trò:** Lưu trữ dữ liệu phi cấu trúc như logs, hình ảnh, dữ liệu raw trước khi xử lý.
- **Tính năng chính:**
  - Chi phí thấp, tối ưu cho **cold storage** (lưu trữ lâu dài).
  - Hỗ trợ Blob **Lifecycle Management**, giúp tự động di chuyển dữ liệu giữa các lớp lưu trữ.
  - Tích hợp dễ dàng với Azure Synapse Analytics để truy vấn dữ liệu.

## 2.4. Đọc tài liệu khoa học

### [POLARIS: The Distributed SQL Engine in Azure Synapse](#)

#### 2.4.1. Ý tưởng

**Polaris** là một công cụ truy vấn SQL phân tán hiện đại được thiết kế riêng cho **Azure Synapse Analytics**, với mục tiêu cung cấp hiệu suất cao và khả năng mở rộng tối đa trong môi trường đám mây. Polaris được xây dựng theo kiến trúc **cloud-native**, cho phép tách biệt giữa các thành phần compute (tính toán) và stateful (quản lý trạng thái), hỗ trợ thực thi song song và phân tán truy vấn SQL trên dữ liệu lưu trữ tại chỗ trong Azure Data Lake mà không cần sao chép dữ liệu. Polaris hướng tới hỗ trợ đầy

đủ các nhu cầu phân tích dữ liệu hiện đại: từ truy vấn tức thời đến xử lý theo lô quy mô lớn, đồng thời tích hợp chặt chẽ với các dịch vụ bảo mật và tối ưu hóa truy vấn trong hệ sinh thái Azure.

Các đặc điểm cốt lõi của POLARIS:

- **Kiến trúc tách biệt compute và state:** POLARIS thiết kế lại mô hình truy vấn SQL truyền thống để tách biệt quá trình xử lý (compute) khỏi trạng thái dữ liệu (state), cho phép mở rộng linh hoạt và chịu lỗi tốt hơn trên nền tảng đám mây.
- **Lập lịch động & điều phối thông minh:** POLARIS sử dụng kiến trúc **control plane / data plane** để lên lịch truy vấn động, đảm bảo tính co giãn và tối ưu hiệu suất.
- **Hỗ trợ đa dạng định dạng dữ liệu và nguồn lưu trữ:** Hệ thống này có thể truy vấn trực tiếp dữ liệu từ Azure Data Lake Storage (ADLS), hỗ trợ các định dạng như Parquet, CSV, JSON, ORC.
- **Tối ưu hóa truy vấn theo thời gian thực:** POLARIS tích hợp với các hệ thống chỉ mục như **Hyperspace** để tăng tốc độ thực thi truy vấn.

#### 2.4.2. Điểm mạnh

- **Kiến trúc cloud-native linh hoạt:** Polaris tận dụng hạ tầng đám mây để tách biệt compute và stateful, cho phép tự động mở rộng quy mô xử lý mà không ảnh hưởng đến độ tin cậy hoặc hiệu suất.
- **Truy vấn dữ liệu tại chỗ (in-place):** Người dùng có thể truy vấn trực tiếp các tệp định dạng phổ biến như Parquet, CSV, JSON từ Azure Data Lake mà không cần sao chép dữ liệu, giúp tiết kiệm chi phí lưu trữ và tăng tốc độ triển khai.
- **Tối ưu hóa hiệu suất nâng cao:** Polaris hỗ trợ các kỹ thuật tối ưu hóa như **caching**, **adaptive query execution**, và **Hyperspace indexing** giúp cải thiện tốc độ truy vấn trên khối lượng dữ liệu lớn.

- **Bảo mật và kiểm soát truy cập chặt chẽ:** Hệ thống tích hợp với **Role-Based Access Control (RBAC)** và Azure Active Directory để kiểm soát quyền truy cập và đảm bảo an toàn dữ liệu.
- **Hỗ trợ đa dạng khối lượng công việc:** Polaris phù hợp cho cả truy vấn tương tác (interactive queries) và phân tích theo lô (batch processing), hỗ trợ nhu cầu phân tích đa dạng từ BI đến khoa học dữ liệu.

#### 2.4.3. Điểm yếu

- **Độ phức tạp cao trong triển khai và tối ưu:** Việc khai thác hiệu quả Polaris yêu cầu kiến thức chuyên sâu về hệ thống phân tán, tối ưu truy vấn, và quản lý tài nguyên Azure – điều này có thể làm tăng thời gian học và chi phí đào tạo.
- **Phụ thuộc vào hệ sinh thái Azure:** Polaris được tối ưu hóa cho Azure, dẫn đến hiện tượng **vendor lock-in**, làm giảm tính linh hoạt nếu doanh nghiệp muốn chuyển đổi sang nền tảng đám mây khác.
- **Độ trễ truy vấn lần đầu (cold start latency):** Với mô hình serverless, lần truy vấn đầu tiên có thể gặp độ trễ nhất định do phải khởi tạo tài nguyên động, ảnh hưởng đến trải nghiệm người dùng trong các trường hợp truy vấn nhanh hoặc không thường xuyên.
- **Khả năng phát sinh chi phí cao:** Mô hình tính phí dựa trên tài nguyên tiêu thụ (per TB scanned) yêu cầu người dùng tối ưu truy vấn cẩn thận. Nếu không có chiến lược kiểm soát hợp lý, chi phí vận hành có thể tăng nhanh chóng theo khối lượng dữ liệu và tần suất truy vấn.



## CHƯƠNG 3. MÔ HÌNH DỮ LIỆU

### 3.1. Tốc độ đọc/ghi dữ liệu

Để đánh giá hiệu suất của hệ thống Azure Synapse Analytics, việc sử dụng các phương pháp benchmark là rất cần thiết. Benchmark giúp đo lường khả năng xử lý dữ liệu, tốc độ phản hồi, và hiệu quả vận hành trong môi trường phân tích dữ liệu lớn. Trong Azure Data Lake và Synapse Analytics, các chỉ số hiệu năng thường được sử dụng bao gồm:

- **Thời gian phản hồi (Response Time):** Khoảng thời gian hệ thống cần để trả về kết quả truy vấn.
- **Tốc độ xử lý (Processing Speed):** Số lượng truy vấn có thể thực hiện trong một khoảng thời gian nhất định.
- **Độ ổn định (System Stability):** Hệ thống có duy trì hiệu năng khi khối lượng dữ liệu tăng đột biến hay không.

Trong **Azure Synapse Analytics**, hiệu suất đọc/ghi dữ liệu được tối ưu hóa dựa trên **Distributed Query Processing (Xử lý truy vấn phân tán)** và **cấu trúc lưu trữ dữ liệu**. Các chỉ số quan trọng:

- **Throughput (Lưu lượng dữ liệu):** Tốc độ đọc/ghi dữ liệu tính theo MB/s hoặc GB/s.
- **Latency (Độ trễ):** Thời gian trễ khi thực hiện thao tác đọc/ghi dữ liệu, tính bằng mili-giây hoặc giây.
- **Concurrency (Mức độ đồng thời):** Số lượng truy vấn hệ thống có thể xử lý cùng lúc.

Tốc độ truy xuất dữ liệu trong **Azure Synapse Analytics** bị ảnh hưởng bởi nhiều yếu tố:

- **Loại Pool được sử dụng:**

- **Serverless SQL Pool:** Phù hợp cho truy vấn dữ liệu chưa được nạp sẵn; tốc độ phụ thuộc vào định dạng file, số lượng file.
- **Dedicated SQL Pool:** Hiệu suất cao hơn khi dữ liệu đã được nạp vào bảng có **Columnstore Indexing**.
- **Định dạng tệp dữ liệu:**
  - **Parquet và ORC:** Hiệu suất cao nhờ cơ chế truy vấn theo cột.
  - **CSV và JSON:** Hiệu suất thấp hơn do xử lý dòng nhiều hơn cột.
- **Phân vùng dữ liệu (Partitioning):** Truy vấn hiệu quả hơn nếu dữ liệu được chia theo ngày, khu vực hoặc đặc trưng nghiệp vụ.

## 2. Đơn vị đo hiệu suất trong Azure Synapse Analytics

- **DWU (Data Warehouse Unit):** Là đơn vị đo hiệu suất của **Dedicated SQL Pool**, xác định khả năng xử lý dữ liệu theo quy mô cụm máy chủ.
- **Tự động mở rộng (Auto-scaling):** Hệ thống có thể tự động mở rộng tài nguyên khi tải truy vấn tăng cao, giúp duy trì hiệu suất ổn định.

### Ví dụ về tốc độ đọc ghi

- **Serverless SQL Pool:**
  - Truy vấn tệp **Parquet 1GB** từ **Azure Data Lake** có thể mất khoảng **2-5 giây**, tùy vào số lượng file.
- **Dedicated SQL Pool:**
  - Khi sử dụng **1000 DWU**, truy vấn trên bảng có 1 tỷ dòng có thể thực hiện dưới **1 giây** nếu đã được tối ưu bằng Columnstore Indexing và phân vùng hợp lý.

### 3.2. Thiết lập luồng xử lý dữ liệu tự động ETL

**ETL (Extract – Transform – Load)** là quy trình trọng yếu trong các hệ thống phân tích dữ liệu hiện đại, giúp di chuyển dữ liệu từ các nguồn khác nhau vào hệ thống phân tích trung tâm để xử lý và khai thác.

#### (a) Mô hình ETL trong Azure Synapse Analytics

- **Extract (Trích xuất dữ liệu):**
  - Dữ liệu được lấy từ **Azure Data Lake Storage (ADLS)**, **SQL Server**, **Blob Storage**, hoặc API bên ngoài.
  - Hỗ trợ nhiều định dạng dữ liệu như **CSV**, **Parquet**, **JSON**, **Avro**, **ORC**.
- **Transform (Chuyển đổi dữ liệu):**
  - Làm sạch dữ liệu, chuẩn hóa định dạng, loại bỏ dữ liệu trùng lặp.
  - Dùng **Apache Spark Pool** hoặc **Serverless SQL Pool** để thực hiện các phép toán biến đổi dữ liệu.
  - Hỗ trợ tích hợp **AI/ML** để xử lý dữ liệu nâng cao.
- **Load (Tải dữ liệu vào hệ thống đích):**
  - Sau khi xử lý, dữ liệu có thể được tải vào **Azure Synapse Dedicated SQL Pool** để truy vấn nhanh hơn.
  - Cũng có thể lưu trữ lại trên **Azure Data Lake** dưới dạng Parquet để tối ưu chi phí.

#### (b) Công cụ hỗ trợ ETL trong Azure Synapse Analytics

- **Azure Data Factory (ADF)**
  - Công cụ **ETL chính** trong hệ sinh thái Azure, hỗ trợ thu thập và biến đổi dữ liệu từ nhiều nguồn.
  - Hỗ trợ **Data Flow**, giúp chuyển đổi dữ liệu mà không cần viết code.
  - Tích hợp với **Azure Synapse Pipelines** để tự động hóa quy trình ETL.

- **Azure Synapse Pipelines**
  - Tương tự Azure Data Factory nhưng được tích hợp trực tiếp trong **Synapse Analytics**.
  - Cho phép tạo **workflow tự động** để chuyển đổi và tải dữ liệu định kỳ.
  - Hỗ trợ **Trigger Scheduling**, giúp chạy ETL theo thời gian thực hoặc theo lịch trình.
- **Apache Spark Pool trong Synapse**
  - Dùng cho **ETL nâng cao**, đặc biệt khi xử lý dữ liệu lớn hoặc phi cấu trúc.
  - Hỗ trợ **PySpark, Scala, SQL**, giúp biến đổi dữ liệu theo yêu cầu.
- **T-SQL trong Serverless & Dedicated SQL Pool**
  - Khi dữ liệu có cấu trúc, có thể sử dụng **T-SQL** để thực hiện ETL trực tiếp trong **Synapse Analytics**.
  - Hỗ trợ **MERGE INTO, CTEs, Window Functions** để xử lý dữ liệu hiệu quả.

#### Ví dụ quy trình ETL:

- **Extract (Trích xuất dữ liệu)**
  - Dữ liệu được lấy từ **Azure Data Lake, SQL Server, Blob Storage**, hoặc API bên ngoài.
  - Hỗ trợ nhiều định dạng như **CSV, Parquet, JSON, Avro**.
  - Công cụ sử dụng: **Azure Data Factory, Synapse Pipelines** sẽ kết nối với Data Lake để lấy dữ liệu.
- **Transform (Chuyển đổi dữ liệu)**
  - **Làm sạch dữ liệu**: Xóa trùng lặp, xử lý lỗi, chuẩn hóa định dạng.
  - **Tính toán & tổng hợp**: Tạo các cột mới, phân loại dữ liệu.

- Công cụ sử dụng: **Apache Spark Pool** (nếu dữ liệu lớn) **hoặc T-SQL trong Synapse SQL Pool** (để xử lý dữ liệu có cấu trúc).
- **Load (Tải dữ liệu vào hệ thống lưu trữ)**
  - Sau khi xử lý, dữ liệu được tải vào **Azure Synapse Dedicated SQL Pool** để tối ưu truy vấn.
  - Có thể lưu trữ lại **Data Lake dưới dạng Parquet** để giảm chi phí lưu trữ.
  - Công cụ sử dụng: **Azure Synapse Pipelines, Data Factory**.

### 3.3. Tối ưu hóa lưu trữ và cải thiện tốc độ đọc/ghi

Để đảm bảo hiệu năng cao và tiết kiệm chi phí khi vận hành hệ thống dữ liệu lớn trên nền tảng Azure, cần áp dụng các chiến lược tối ưu hóa sau:

#### Giải pháp tối ưu hóa lưu trữ dữ liệu trên Azure Data Lake

- **Chọn định dạng tệp hiệu quả**
  - **Parquet & ORC**: Định dạng cột, giúp truy vấn nhanh hơn **10-100 lần** so với CSV.
  - **JSON & Avro**: Phù hợp với dữ liệu bán cấu trúc, nhưng truy vấn chậm hơn Parquet.
  - **Giảm kích thước tệp** bằng **nén Gzip, Snappy** để tối ưu không gian lưu trữ.
- **Partitioning (Phân vùng dữ liệu)**
  - Chia dữ liệu theo **thời gian (year/month/day)** hoặc **khu vực địa lý** để giảm phạm vi quét dữ liệu.
  - Ví dụ: Truy vấn dữ liệu chỉ trong **tháng 1/2024** nhanh hơn so với tìm trong toàn bộ dataset.
- **Indexing & Metadata Management**
  - Sử dụng **Azure Data Lake Hierarchical Namespace (HNS)** để tổ chức dữ liệu theo cấu trúc thư mục giúp tìm kiếm nhanh hơn.

- Kết hợp **Azure Data Catalog** để quản lý metadata, hỗ trợ truy vấn hiệu quả.

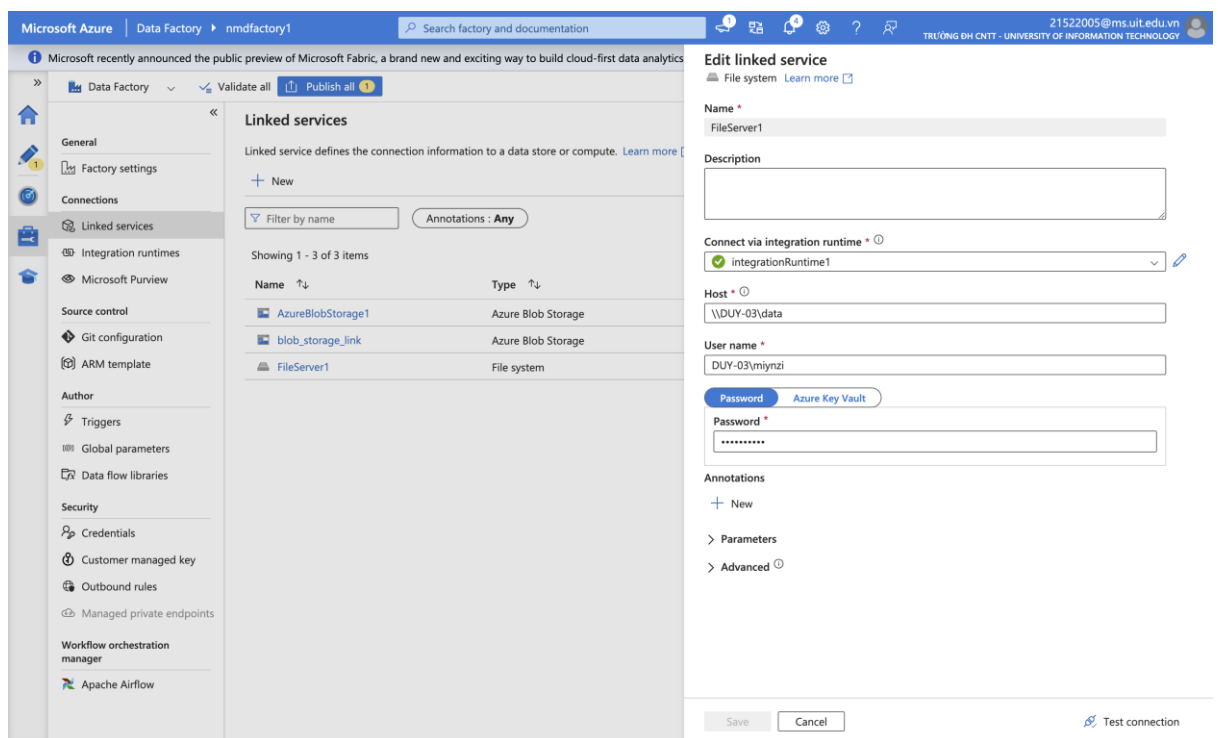
### **Giải pháp cải thiện tốc độ đọc ghi dữ liệu từ vùng lưu trữ sang ứng dụng**

- **Xử lý song song (Parallel Processing):**
  - **Serverless SQL Pool:** Xử lý dữ liệu trực tiếp trên Data Lake mà không cần di chuyển dữ liệu.
  - **Dedicated SQL Pool:** Hỗ trợ **Massively Parallel Processing (MPP)**, tăng tốc độ truy vấn dữ liệu lớn.
- **Sử dụng Caching và Materialized Views:**
  - **Materialized Views:** Lưu trữ kết quả truy vấn trước để giảm thời gian thực thi.
  - **Result-set caching:** Tăng tốc độ trả kết quả cho truy vấn lặp lại.
- **Quản lý tải công việc (Workload Management):**
  - Dùng **Query Performance Insights** trong Synapse để theo dõi truy vấn chậm.
  - **Scaling DWU (Data Warehouse Unit)** tự động mở rộng tài nguyên khi tải truy vấn tăng.

## CHƯƠNG 4. XÂY DỰNG ETL PIPELINE TRÊN AZURE DATA FACTORY




### 4.1. Trường hợp 1

Ví dụ 1 doanh nghiệp có 3 chi nhánh tại Hà Nội, HCM, Đà Nẵng. Dữ liệu bán hàng được lưu dưới định dạng CSV và doanh nghiệp cần phân tích dữ liệu trên cả 3 chi nhánh. Ở phần này, nhóm sẽ thực hiện sử dụng linked service là file system của máy chủ Windows, như vậy, dịch vụ Azure sẽ kết nối với thư mục làm việc đã được cấu hình trên máy Windows để lấy các dữ liệu từ thư mục đó



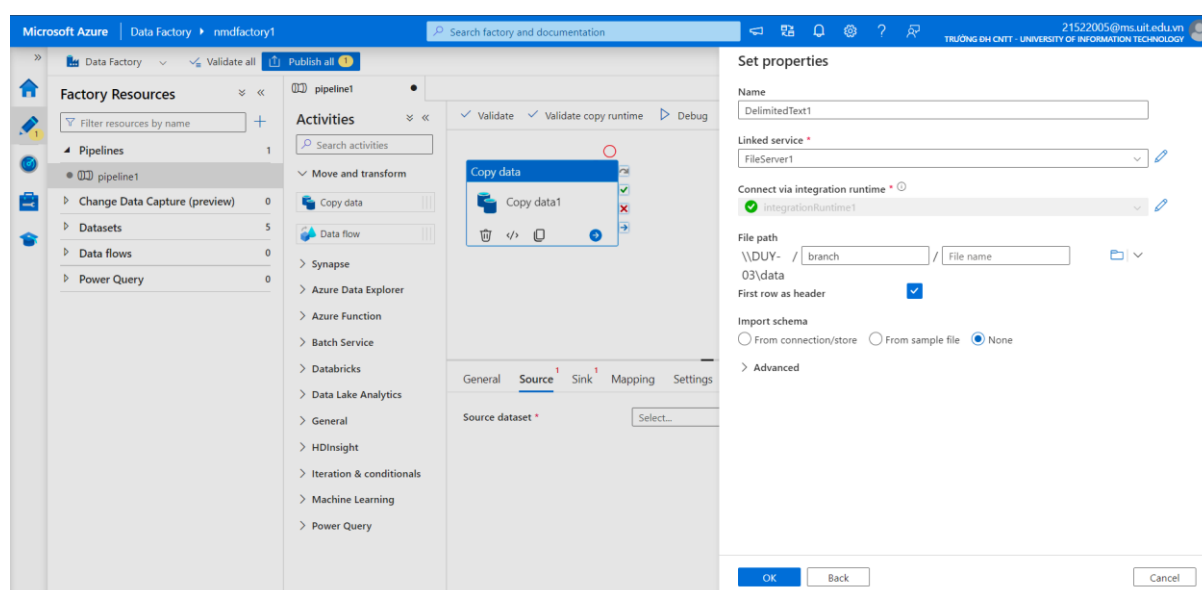
Hình 1. Cấu hình linked service với máy chủ Windows

Ở đây, ta cần gộp dữ liệu ở 3 chi nhánh (trên local) để tổng hợp

This PC > DATA (D:) > data > branch			
Name	Date modified	Type	Size
 sales_DN.csv	3/16/2024 12:03 AM	Microsoft Excel Co...	2 KB
 sales_HCM.csv	3/16/2024 12:03 AM	Microsoft Excel Co...	2 KB
 sales_HN.csv	3/16/2024 12:03 AM	Microsoft Excel Co...	2 KB

Hình 2. Các tệp trên máy chủ Windows

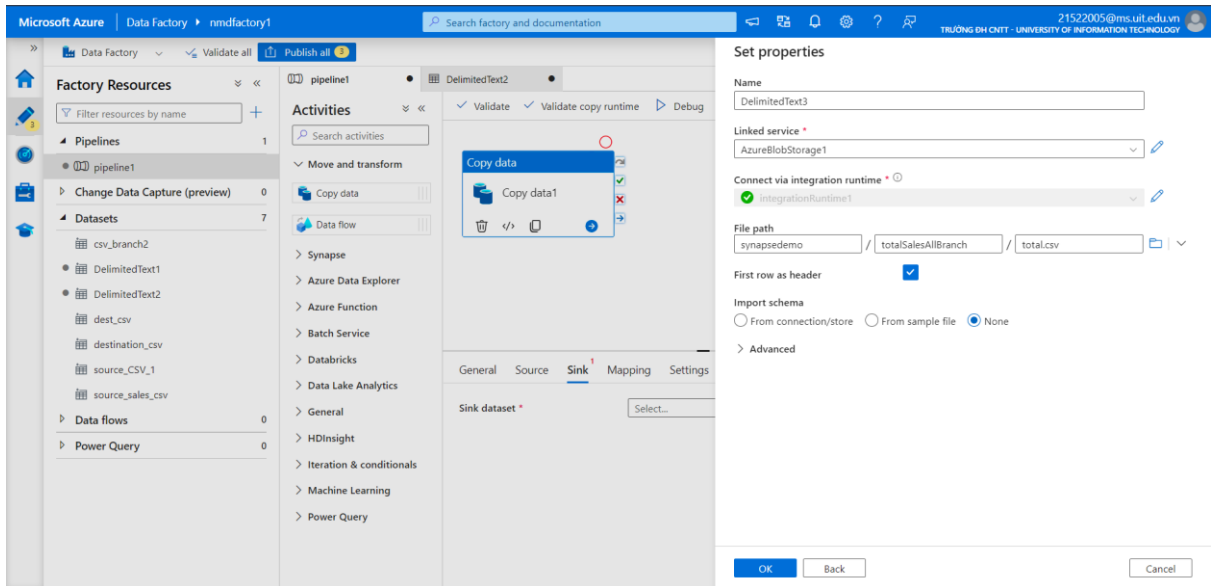
Tiến hành tạo Pipeline với Copy Data, với nguồn dữ liệu là cả 3 chi nhánh trên



Hình 3. Tạo copy data activity với source là folder "branch" nơi chứa các tệp cần merge

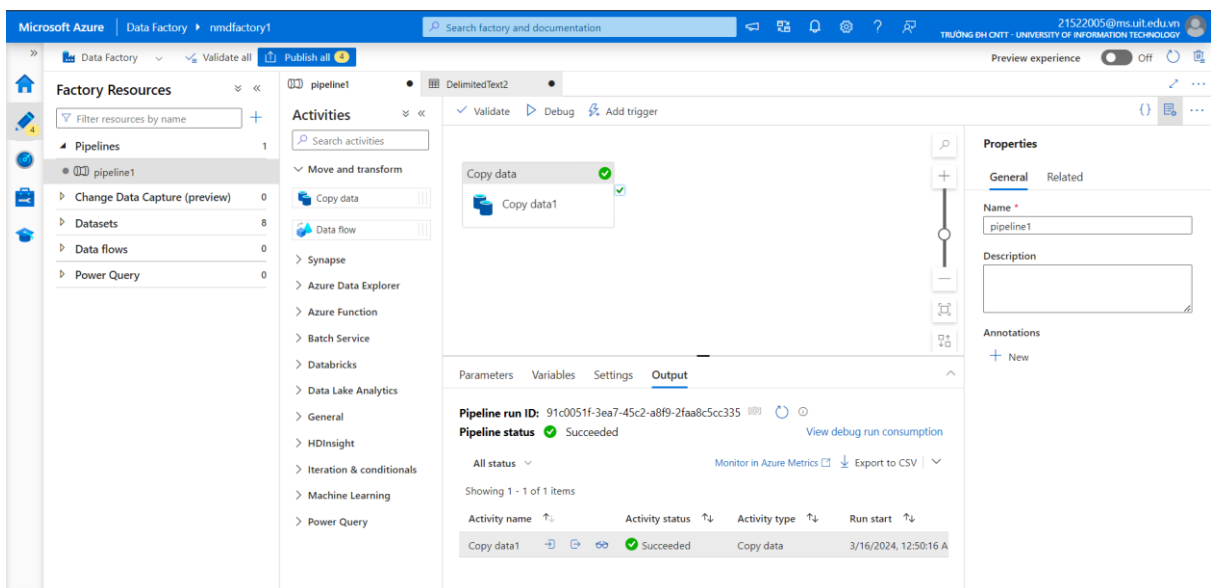
Tiếp đến trữ thư mục đến Container (Azure Blob Storage) để chứa file tổng hợp



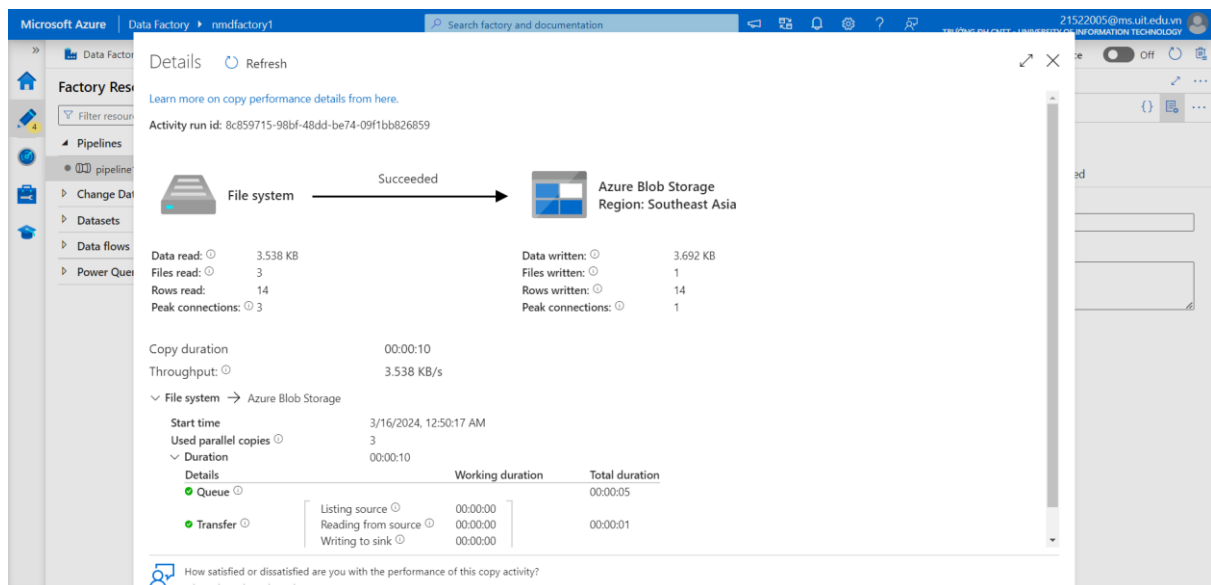


Hình 4. Tập tin đích sẽ được lưu trên container

Tiến hành Merges File lại với nhau, và kết quả là gộp thành công

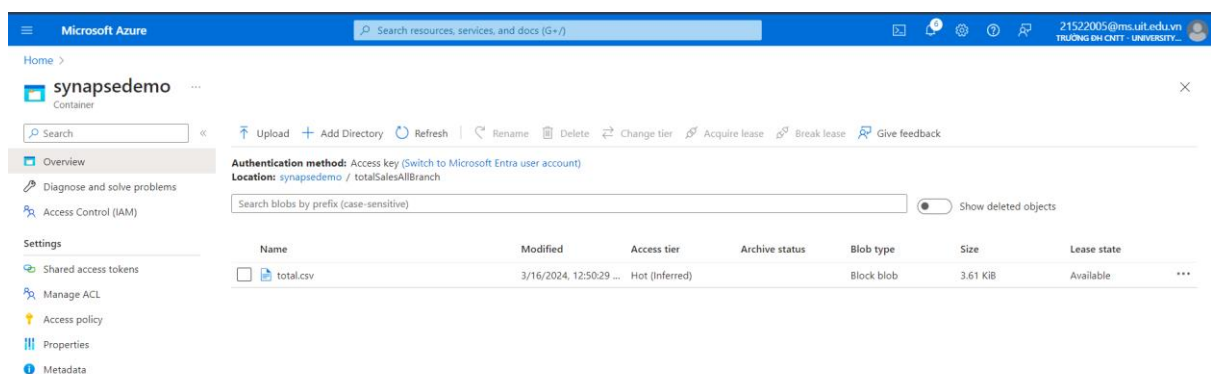


Hình 5. Chạy pipeline để merge các tập lại

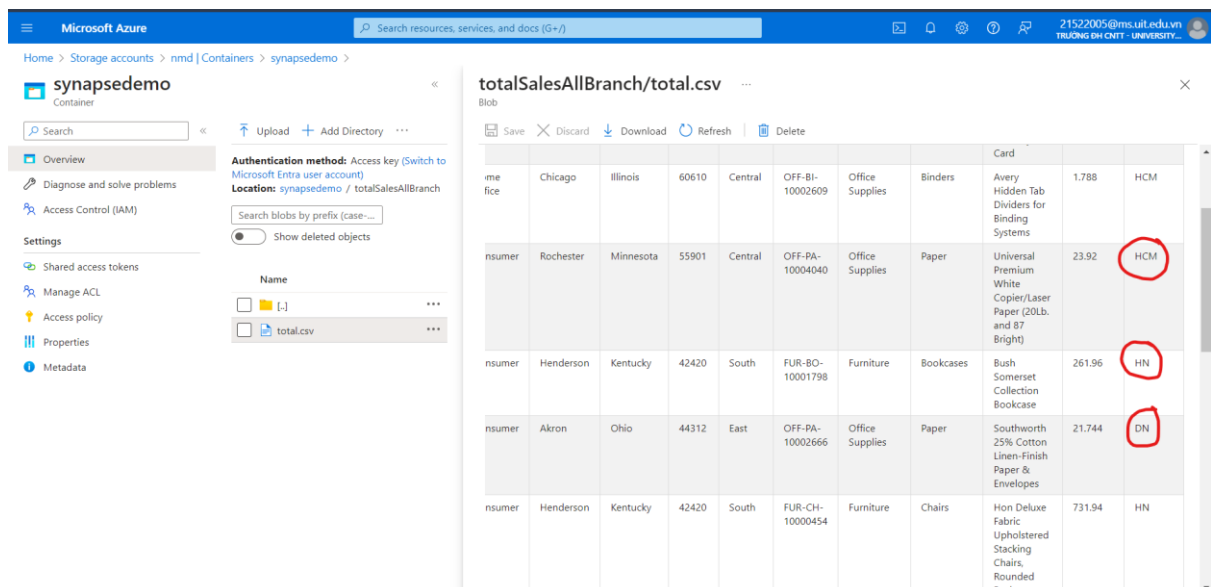


Hình 6. Chi tiết của lệnh chạy pipeline

Ta thu được file tổng hợp dữ liệu ở 3 chi nhánh



Hình 7. Tập tin đích sau khi gộp



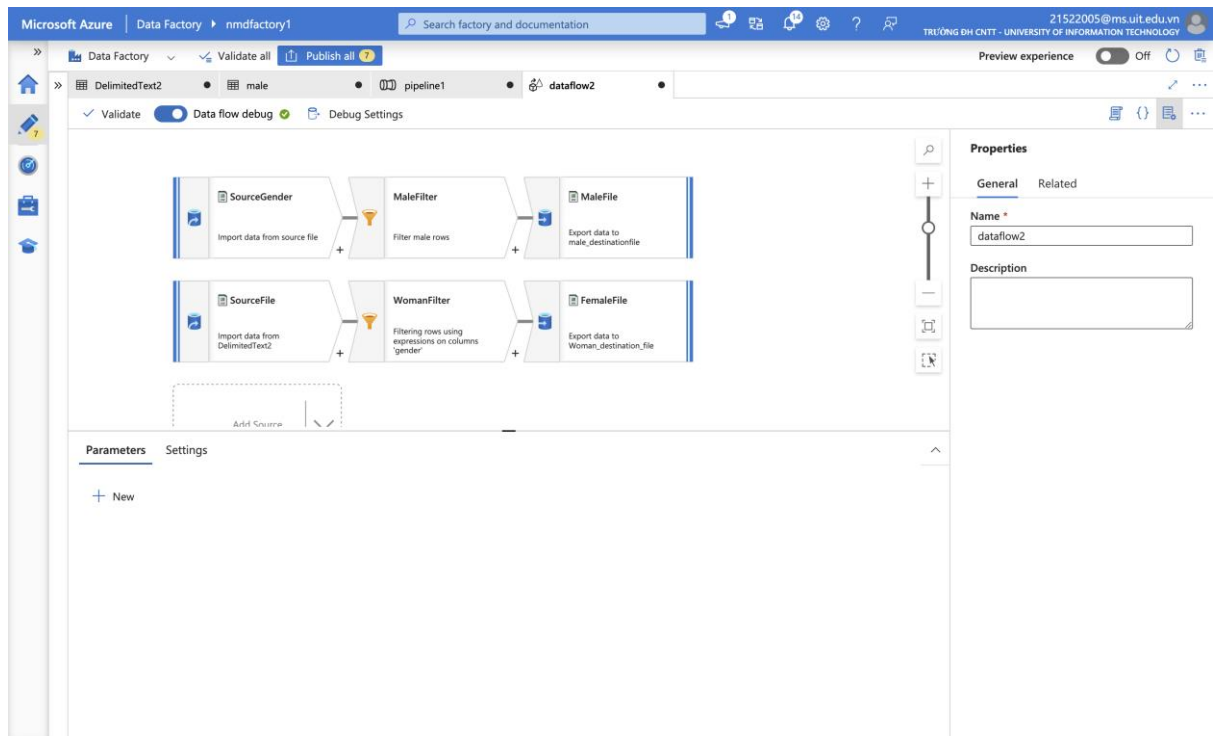
Office	Location	Zip	Region	Product ID	Category	Item	Quantity	Unit
Chicago	Illinois	60610	Central	OFF-BI-10002609	Office Supplies	Binders	1,788	HCM
Rochester	Minnesota	55901	Central	OFF-PA-10004040	Office Supplies	Paper	23.92	HCM
Henderson	Kentucky	42420	South	FUR-BO-10001798	Furniture	Bookcases	261.96	HN
Akron	Ohio	44312	East	OFF-PA-10002666	Office Supplies	Paper	21.744	DN
Henderson	Kentucky	42420	South	FUR-CH-10000454	Furniture	Chairs	731.94	HN

Hình 8. Chi tiết của tệp đích sau khi gộp dữ liệu

## 4.2. Trường hợp 2

Giả sử tại 1 chi nhánh nào đó, chúng ta có thông tin của các cá nhân với giới tính khác nhau và bây giờ chúng ta cần tách riêng thông tin của nam và nữ để thực hiện phân tích dữ liệu.

Ở đây, nhóm thực hiện tạo 1 data flow để filter các bộ dữ liệu với điều kiện là nam và nữ để tách ra 2 file csv mới.



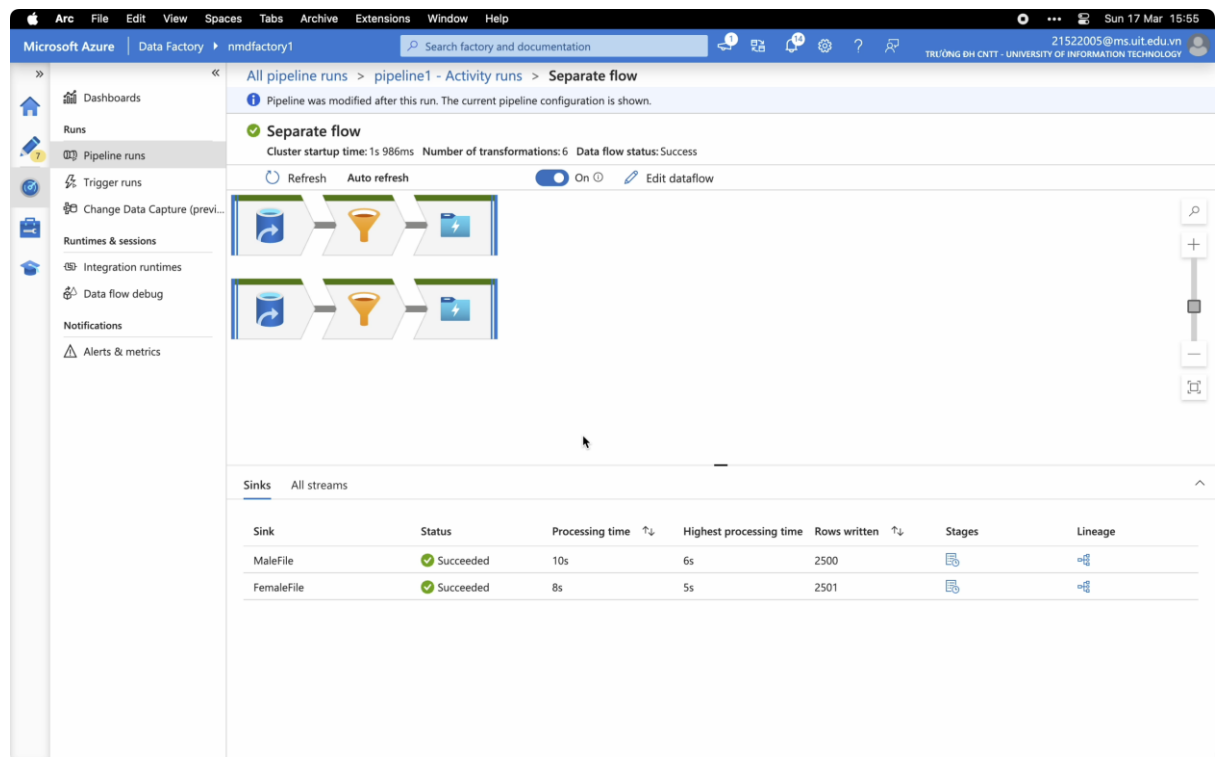
Hình 9. Pipeline để tách dữ liệu dựa trên thuộc tính cột

Thời gian thực thi cho data flow là 2 phút 10s

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties	Activity ID
Separate flow	Succeeded	Data flow	3/17/2024, 3:52:58 PM	2m 10s	debugpool-8Cores-Gei		8b30978

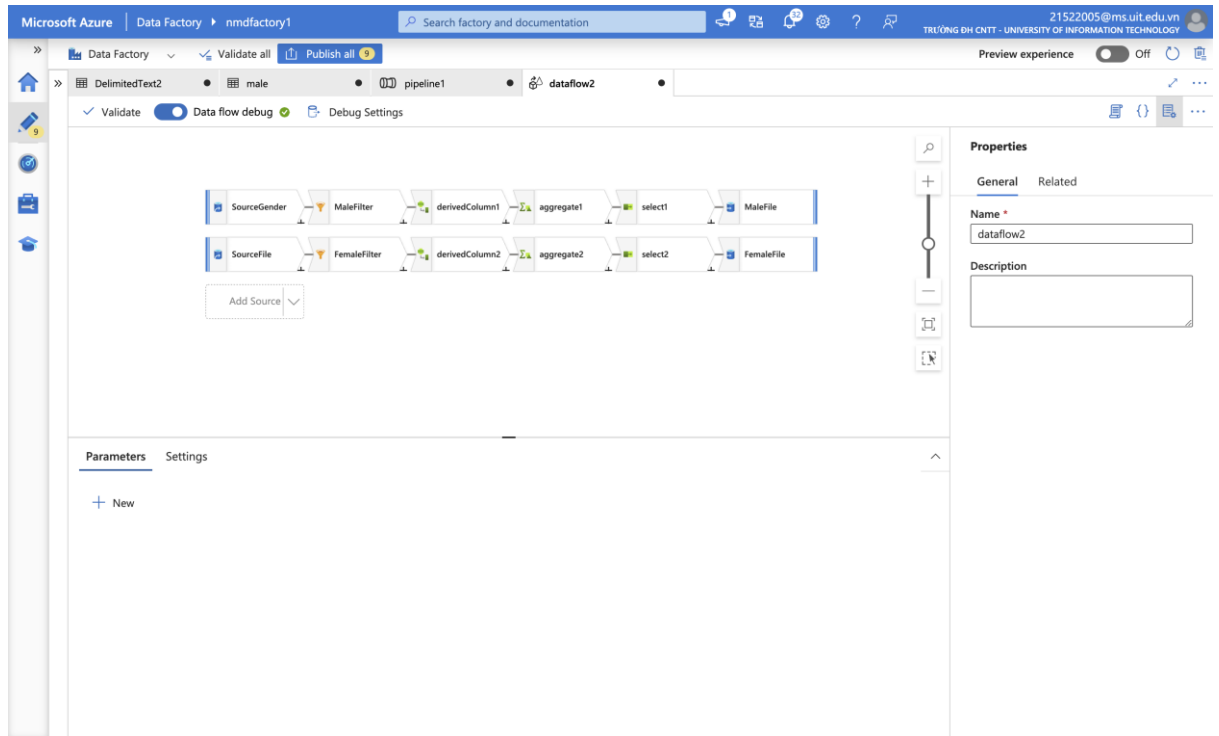
Hình 10. Thực thi luồng xử lý

Sau đây là kết quả thực thi pipeline



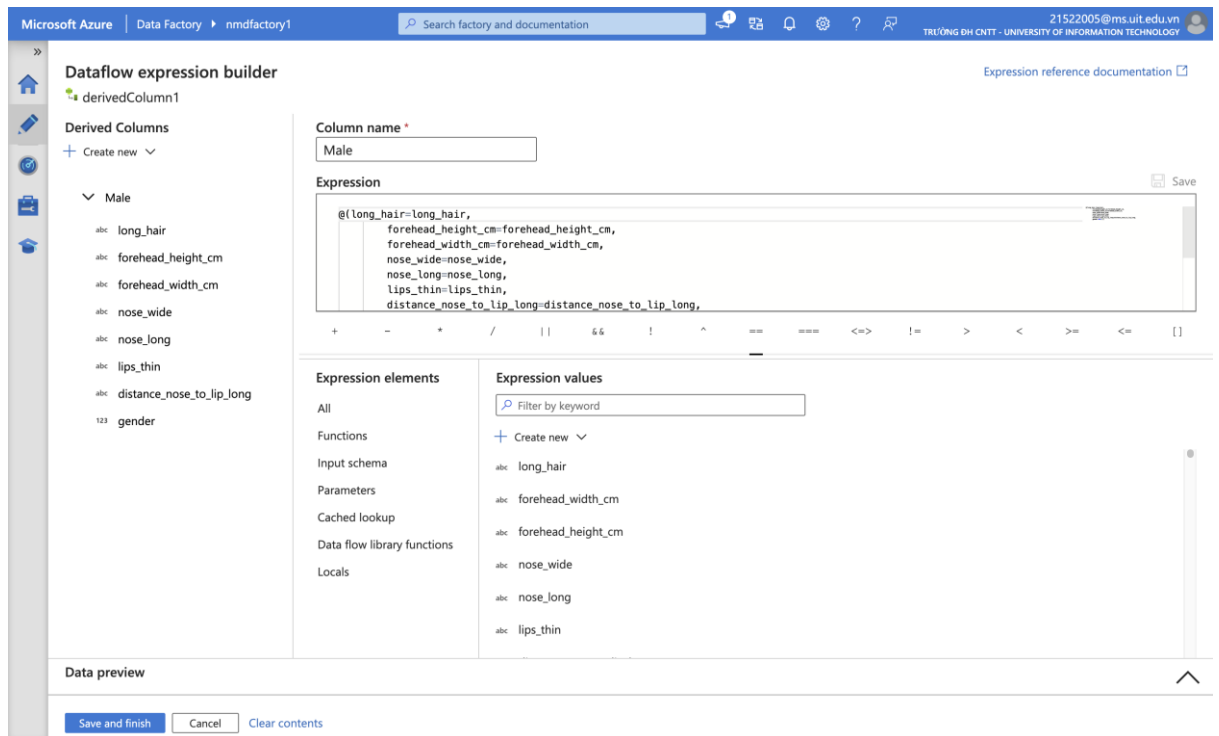
Hình 11. Kết quả chi tiết của luồng xử lý

Tiếp theo, nhóm sẽ thử phân tích các đặc điểm unique của mỗi giới tính và thử chuyển đổi bằng file CSV.



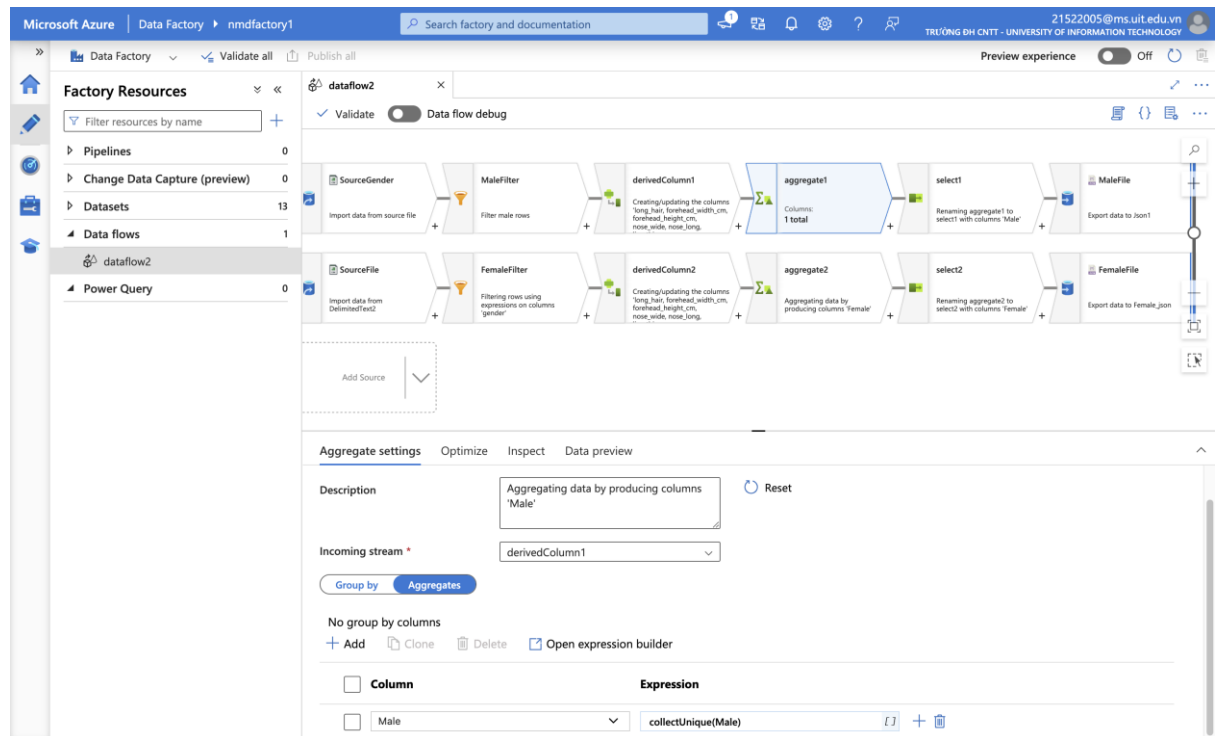
Hình 12. Luồng xử lý để tách và chuyển đổi tệp tin thành định dạng khác

Sau khi filter data source thì lúc này chúng ta có 2 đối tượng CSV nam và nữ. Vì vậy khi map sang file json, nhóm sẽ tạo 1 column mới tên là male và female để chứa các thuộc tính còn lại. Khi cột này chuyển sang định dạng JSON sẽ trở thành mảng Male và Female chứa các object tương ứng



Hình 13. Để tạo 1 mảng Male trong Json, chúng ta sẽ tạo 1 cột mới chứa object cũ

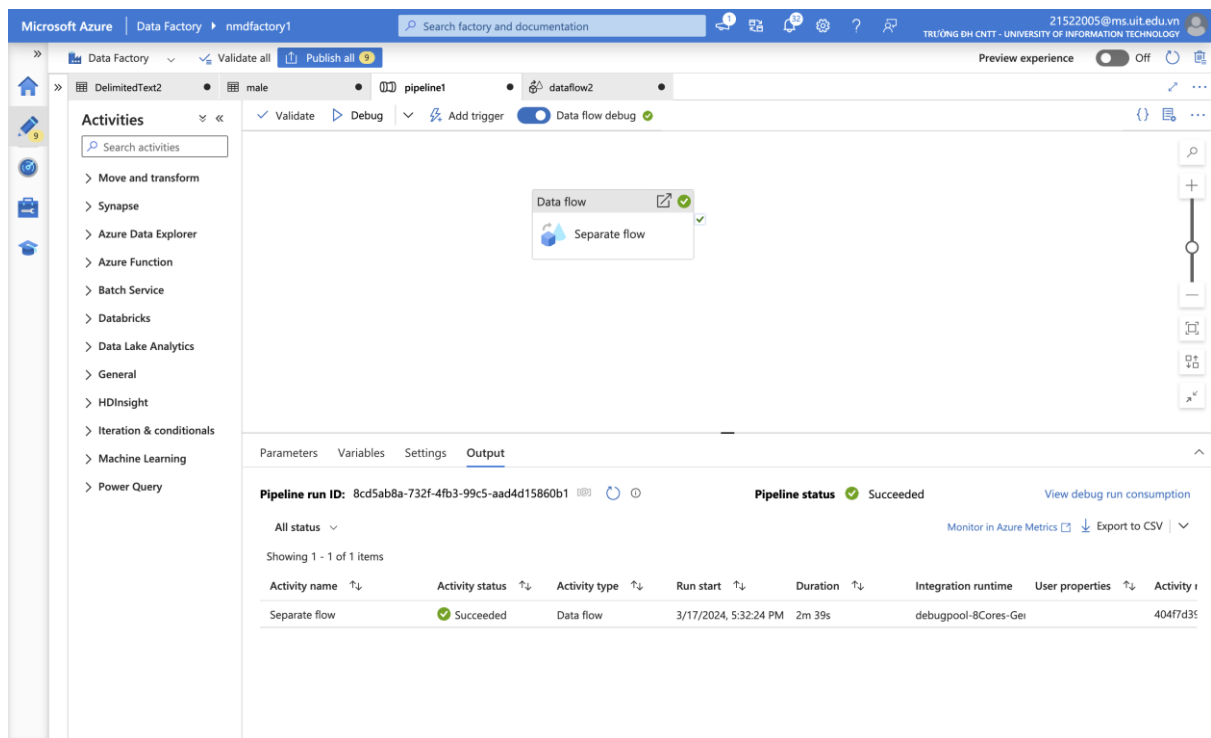
Khi tổng hợp dữ liệu (aggregate data), nhóm đã kết hợp lấy những object unique để lọc những nhóm ngoại hình trùng lặp



Hình 14. Lọc những đối tượng unique trong dataset

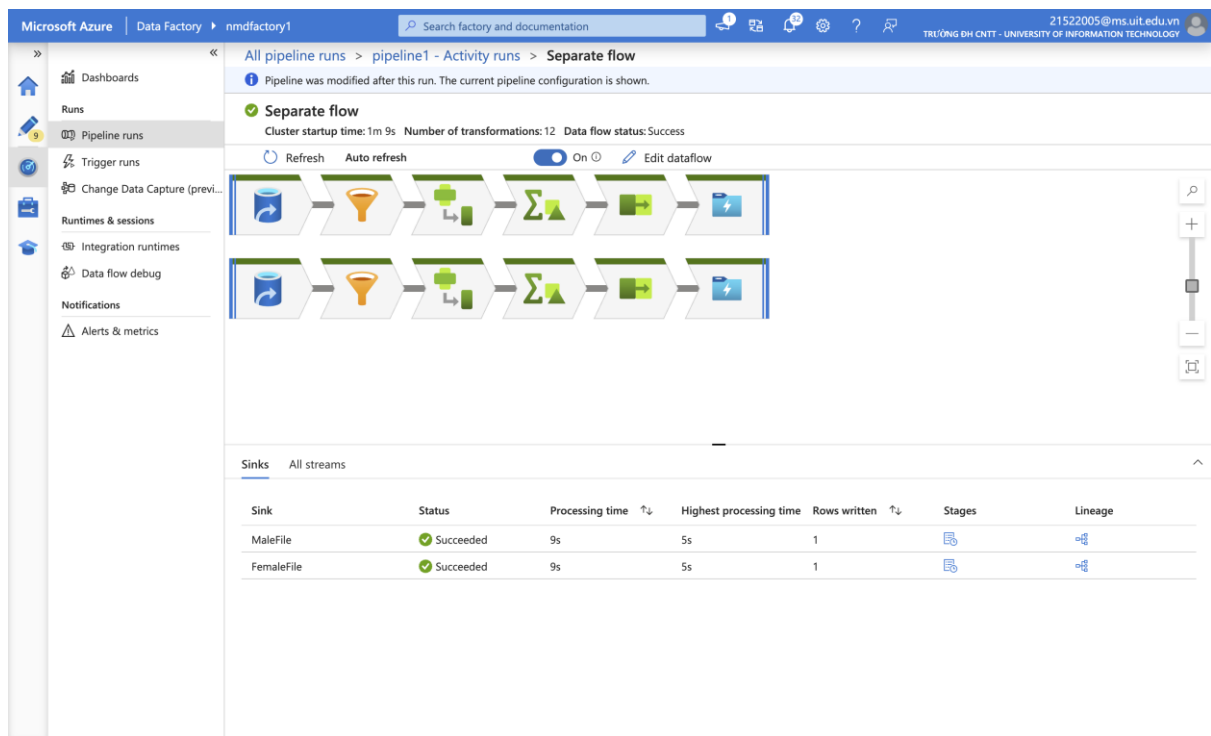
Kết quả chạy pipeline mất 2 phút 39 giây





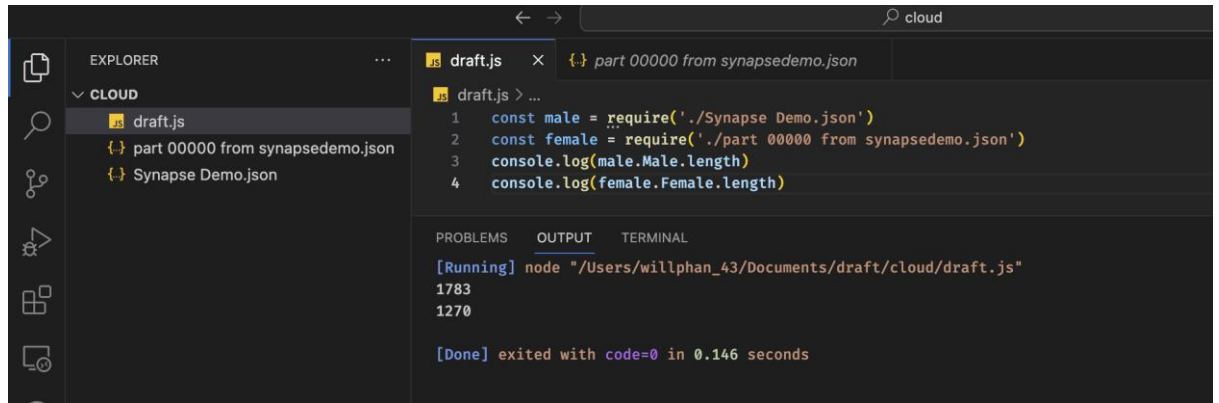
Hình 15 Kết quả thực thi luồng xử lý

Pipeline đã tạo ra 2 tệp đích Json cho male và female



Hình 16. Kết quả chi tiết của luồng xử lý

Bởi vì, pipeline chỉ collect những unique object nên so kết quả bên trên, ta thấy chiều dài của mảng đã giảm.



Hình 17. Chiều dài của mảng sau khi chỉ chọn những đối tượng unique

Bây giờ, ta đã có 1 tệp tin dữ liệu mảng Male như sau:

```

{
  Male: [
    {
      long_hair: '0',
      forehead_height_cm: '5.2',
      forehead_width_cm: '12.8',
      nose_wide: '1',
      nose_long: '0',
      lips_thin: '1',
      distance_nose_to_lip_long: '1',
      gender: 1
    },
    {
      long_hair: '0',
      forehead_height_cm: '6.9',
      forehead_width_cm: '12.7',
      nose_wide: '0',
      nose_long: '1',
      lips_thin: '1',
      distance_nose_to_lip_long: '1',
      gender: 1
    },
    {
      long_hair: '0',
      forehead_height_cm: '5.5',
      forehead_width_cm: '12.5',
      nose_wide: '1',
      nose_long: '0',
      lips_thin: '1',
      distance_nose_to_lip_long: '1',
      gender: 1
    },
    {
      long_hair: '0',
      forehead_height_cm: '6.1'
    }
  ]
}

```

Hình 18. Tập tin sau khi được chuyển đổi định dạng

## TÀI LIỆU THAM KHẢO

### Sách tham khảo chính:

[Shiyal, B. \(2021\). \*Beginning Azure Synapse Analytics: Transition from Data Warehouse to Data Lakehouse\*. Apress.](#)

### Tài liệu về Data Lake:

- Azure. (n.d.). *What is a data lake?*. Retrieved from <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-data-lake>
- AWS. (n.d.). *What is a data lake?*. Retrieved from <https://aws.amazon.com/vi/what-is/data-lake/>
- Google Cloud. (n.d.). *What is a data lake?*. Retrieved from <https://cloud.google.com/learn/what-is-a-data-lake>
- Databricks. (n.d.). *Discover data lakes*. Retrieved from <https://www.databricks.com/discover/data-lakes>
- Oracle. (n.d.). *What is a data lake?*. Retrieved from <https://www.oracle.com/th/big-data/data-lake/what-is-data-lake/>
- Microsoft Learn. (n.d.). *Data lake scenarios*. Retrieved from <https://learn.microsoft.com/en-us/azure/architecture/data-guide/scenarios/data-lake>
- Azure. (n.d.). *Solutions – Data lake*. Retrieved from <https://azure.microsoft.com/en-us/solutions/data-lake>

### Tài liệu về Azure Synapse Analytics và các thành phần:

- Microsoft Azure. (n.d.). *Azure Synapse Analytics*. Retrieved from <https://azure.microsoft.com/en-us/products/synapse-analytics>
- Microsoft Learn. (n.d.). *Azure Synapse Analytics documentation*. Retrieved from <https://learn.microsoft.com/en-us/azure/synapse-analytics/>

- K21Academy. (n.d.). *Azure Synapse Analytics Overview*. Retrieved from <https://k21academy.com/microsoft-azure/data-engineer/azure-synapse-analytics/>
- YouTube – Azure Synapse. (n.d.). Retrieved from <https://youtube.com/@AzureSynapse>
- Integrate.io. (n.d.). *What is Azure Synapse Analytics?*. Retrieved from <https://www.integrate.io/blog/what-is-azure-synapse-analytics/>
- Microsoft Learn. (n.d.). *Introduction to Azure Synapse Analytics*. Retrieved from <https://learn.microsoft.com/en-us/training/modules/introduction-azure-synapse-analytics/>
- Microsoft Learn. (n.d.). *Data platform end-to-end scenario*. Retrieved from <https://learn.microsoft.com/en-us/azure/architecture/example-scenario/dataplate2e/data-platform-end-to-end?tabs=portal>
- Microsoft Learn. (n.d.). *SQL architecture overview*. Retrieved from <https://learn.microsoft.com/en-us/azure/synapse-analytics/sql/overview-architecture>
- Microsoft Learn. (n.d.). *MPP architecture in Synapse SQL*. Retrieved from <https://learn.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/massively-parallel-processing-mpp-architecture>
- Microsoft Learn. (n.d.). *Synapse SQL features overview*. Retrieved from <https://learn.microsoft.com/en-us/azure/synapse-analytics/sql/overview-features>
- Microsoft Learn. (n.d.). *Apache Spark in Synapse*. Retrieved from <https://learn.microsoft.com/en-us/azure/synapse-analytics/spark/apache-spark-overview>

- Microsoft Learn. (n.d.). *Data Factory concepts – Pipelines & Activities*. Retrieved from <https://learn.microsoft.com/en-us/azure/data-factory/concepts-pipelines-activities?tabs=data-factory>
- Microsoft Learn. (n.d.). *Get started with Synapse Pipelines*. Retrieved from <https://learn.microsoft.com/en-us/azure/synapse-analytics/get-started-pipelines>
- Microsoft Learn. (n.d.). *SQL Synapse Link overview*. Retrieved from <https://learn.microsoft.com/en-us/azure/synapse-analytics/synapse-link/sql-synapse-link-overview>

Microsoft Learn. (n.d.). *SQL Server views*. Retrieved from <https://learn.microsoft.com/en-us/sql/relational-databases/views/views?view=sql-server-ver16>