TRƯỜNG ĐẠI HỌC KINH TẾ

KHOA THỐNG KÊ – TIN HỌC

BÁO CÁO THỰC TẬP NGHỀ NGHIỆP

NGÀNH HỆ THỐNG THÔNG TIN QUẢN LÝ

CHUYÊN NGÀNH TIN HỌC QUẢN LÝ

XÂY DỰNG PINELINES TRÊN CLOUD

Sinh viên thực hiện	: **Đỗ Lê Khanh**
Lớp	: **47K14**
Đơn vị thực tập	: **TMA Solutions Bình Định**
Cán bộ hướng dẫn	: **Tăng Thị Thúy Vân**
Giảng viên hướng dẫn	: **Th.S Nguyễn Thành Thủy**

Đà Nẵng, 8/2024

CÔNG TY TNHH GIẢI PHÁP PHẦN MỀM TƯỜNG MINH BÌNH ĐỊNH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP

Họ và tên sinh viên: Đỗ Lê Khanh Lớp: 47K14

Khoa Thống kê – Tin học, Trường Đại học Kinh tế, Đại học Đà Nẵng

Thời gian thực tập: Từ ngày 13/5/2024 đến ngày 02/8/2024

Tên đơn vị thực tập: Công ty TNHH Giải pháp Phần mềm Tường Minh Bình Định (TMA Solutions Bình Định)

Địa chỉ: 12 Đại lộ Khoa học, KV 2, phường Ghềnh Ráng, thành phố Quy Nhơn

Số điện thoại liên hệ: (0256) 389 8979 – Máy lẻ: 7222 | Hotline: 0977 465 083

Họ tên cán bộ hướng dẫn: Tăng Thị Thúy Vân

Sau quá trình thực tập của sinh viên tại đơn vị, chúng tôi có một số đánh giá như sau:

STT	Nội dung đánh giá	Rất không tốt	Không tốt	Bình thường	Tốt	Rất tốt
1	Về thái độ, ý thức, đạo đức và việc tuân thủ các quy định và văn hóa đơn vị thực tập	☐	☐	☐	☐	☒
2	Kiến thức chuyên môn	☐	☐	☐	☒	☐
3	Khả năng hòa nhập, thích nghi và tác phong nghề nghiệp	☐	☐	☐	☒	☐
4	Trách nhiệm, sáng tạo trong công việc	☐	☐	☐	☒	☐

(Anh/chị vui lòng đánh dấu X vào ô tương ứng với năng lực của sinh viên)

Điểm thực tập: *(Vui lòng ghi rõ bằng số và bằng chữ - theo thang điểm 10)*

8.8	Tám chấm tám

Ý kiến nhận xét và đề xuất (*Nhằm nâng cao chất lượng đào tạo, Nhà trường rất mong muốn nhận thêm những ý kiến khác từ quý doanh nghiệp*):

❖ **Nhận xét:**

- Siêng năng hoàn thành tốt công việc được giao
- Làm việc nhóm tích cực
- Có tinh thần học hỏi
- Cần trau dồi thêm kỹ năng tiếng Anh và tự tin khi thuyết trình

❖ **Đề xuất:**

- Không

Xác nhận của đơn vị thực tập

TP. Hành chính và Đào tạo

TMA SOLUTIONS BINH DINH
INDUSTRY INTERNSHIP
12 Science Avenue, Ghenh Rang, Quy Nhon, Binh Dinh
Tel: 0256.3899879- Ext: 7222
Email: intern-binhdinh@tma.com.vn

Lâm Thị Thanh Thảo

Quy Nhơn, ngày 02 tháng 8 năm 2024

Người hướng dẫn

Tăng Thị Thúy Vân

LỜI CẢM ƠN

Trong suốt quá trình thực tập và hoàn thành báo cáo thực tập này, em đã nhận được sự hỗ trợ và giúp đỡ quý báu từ nhiều cá nhân và tổ chức. Em xin bày tỏ lòng biết ơn chân thành đến tất cả những người đã giúp đỡ và đóng góp cho sự thành công của dự án này.

Trước hết, em xin gửi lời cảm ơn sâu sắc đến các thầy cô trong Khoa Thống kê – Tin học, Trường Đại học Kinh tế, đặc biệt là thầy Nguyễn Thành Thủy, người đã hướng dẫn và chỉ dẫn tận tình trong suốt quá trình thực tập. Sự hướng dẫn chi tiết và những góp ý quý báu của thầy đã giúp em hoàn thiện kiến thức và kỹ năng cần thiết để thực hiện dự án này.

Em cũng xin gửi lời cảm ơn chân thành đến công ty TMA Solutions Bình Định, nơi em đã có cơ hội thực tập và học hỏi trong môi trường làm việc chuyên nghiệp. Cảm ơn sự hỗ trợ nhiệt tình từ ban lãnh đạo và các anh chị đồng nghiệp, đặc biệt là chị Tăng Thị Thúy Vân, người đã luôn sẵn sàng giúp đỡ và chia sẻ kiến thức, kinh nghiệm quý báu trong lĩnh vực Data Engineering.

Vì kiến thức bản thân còn nhiều hạn chế nên trong thời gian thực tập cũng như làm báo cáo khó tránh tránh khỏi những thiếu sót, em kính mong sự góp ý từ thầy cô và quý công ty để em có thể rút kinh nghiệm hoàn thành tốt đề tài của mình.

Em xin chân thành cảm ơn!

Đỗ Lê Khanh

LỜI CAM ĐOAN

Em xin cam đoan rằng báo cáo thực tập nghề nghiệp ngành Data Engineer với đề tài "Xây Dựng Pipelines Trên Cloud" này là công trình nghiên cứu và làm việc của riêng em dưới sự hướng dẫn của thầy Nguyễn Thành Thủy.

Tất cả các dữ liệu và kết quả trình bày trong báo cáo này là trung thực và là sản phẩm mà em đã nỗ lực nghiên cứu trong suốt quá trình thực tập.

Em xin chịu hoàn toàn trách nhiệm về tính chính xác và trung thực của nội dung trong báo cáo này.

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC CÁC TỪ VIẾT TẮT

IaaS : Infrastructure as a Service

PaaS : Platform as a Service

SaaS : Software as a Service

ADF : Azure Data Factory

IDE : Integrated Development Environment

SQL : Structured Query Language

VS Code : Visual Studio Code

PK : Primary Key

FK : Foreign Key

ERD : Entity Relationship Model

OVS : Oversize

KPI : Key Performance Indicators

LỜI MỞ ĐẦU

1. Mục tiêu của đề tài

- Khám phá và hiểu rõ quy trình xây dựng pipeline dữ liệu trên nền tảng điện toán đám mây, cụ thể là Microsoft Azure.
- Thực hành triển khai pipeline để xử lý và phân tích dữ liệu trong thực tế, sử dụng các công cụ và dịch vụ của Azure.
- Xây dựng kỹ năng và kiến thức về Data Engineering, đặc biệt là trong việc xử lý dữ liệu lớn và tự động hóa quy trình xử lý dữ liệu.

2. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Các khái niệm cơ bản và nâng cao về Data Engineering.
- Các dịch vụ và công cụ trên Microsoft Azure phục vụ cho việc xây dựng và triển khai pipelines dữ liệu.

Phạm vi nghiên cứu:

- Thực hành xây dựng một pipeline dữ liệu từ đầu đến cuối, bao gồm các bước: ingestion (nhập dữ liệu), processing (xử lý dữ liệu), storing (lưu trữ dữ liệu), và analysis (phân tích dữ liệu).

3. Kết cấu của đề tài

Đề tài được tổ chức gồm phần mở đầu, 5 chương nội dung và phần kết luận.

- **Mở đầu**
- **Chương 1**: Tổng quan về doanh nghiệp và cơ sở lý thuyết.
- **Chương 2**: Giới thiệu dữ liệu và thiết kế Pipelines.
- **Chương 3:** Triển khai Pipelines trên Azure.
- **Kết luận và đánh giá**

CHƯƠNG 1. TỔNG QUAN VỀ DOANH NGHIỆP VÀ CƠ SỞ LÝ THUYẾT

1.1. Giới thiệu tổng quát về doanh nghiệp thực tập

1.1.1. *Giới thiệu về TMA Solutions Bình Định.*

Hình 1.1 Logo TMA Solutions Bình Định

Được thành lập vào năm 1997, TMA trở thành tập đoàn công nghệ hàng đầu tại Việt Nam, phát triển vững mạnh cùng đội ngũ gần 4,000 kỹ sư tài năng. TMA tự hào khi hợp tác với các khách hàng, tập đoàn công nghệ hàng đầu từ hơn 30 quốc gia khác nhau trên toàn thế giới.

Hiện nay, TMA trải rộng thị trường với 7 chi nhánh tại Việt Nam, trong đó 6 chi nhánh tại thành phố Hồ Chí Minh và 1 chi nhánh tại thành phố Quy Nhơn, Bình Định. Ngoài ra, TMA còn có mạng lưới quốc tế với 6 chi nhánh tại các quốc gia khác nhau bao gồm Mỹ, Úc, Canada, Đức, Nhật Bản và Singapore.

Tháng 6 năm 2018, TMA đã mở chi nhánh tại Địa chỉ: 12 Đại lộ Khoa học, Khu vực 2, P. Ghềnh Ráng, TP Quy Nhơn, Bình Định. Sau 4 năm, công ty TMA Bình Định có tên đầy đủ là Công ty TNHH Giải pháp Phần mền Tường Minh Bình Định, TMA Bình Định đã phát triển nhanh chóng với hơn 400 kỹ sư, trong đó có nhiều kỹ sư đang làm việc tại TP.HCM đã trở về làm việc tại quê hương.

Tháng 8 năm 2018, TMA Bình Định khởi công xây dựng Công viên sáng tạo phần mềm TMA (TMA Innovation Park - TIP) tại Thung lũng Sáng tạo Quy Nhơn (Quy Nhon Innovation Park – QNIVY) với vốn đầu tư hàng trăm tỷ đồng.

Vinh dự là trung tâm sáng tạo phần mềm đầu tiên tại Thung lũng Sáng tạo tại Quy Nhơn, Bình Định, Công viên Sáng tạo TMA mang trong mình sứ mệnh trở thành trung tâm phát triển phần mềm và công nghệ cao hàng đầu tại miền Trung, góp phần quan trọng đưa

Thung lũng Sáng tạo Quy Nhơn thành một trong những điểm sáng khi nhắc đến nền công nghệ 5.0 tại Việt Nam.

Công viên Sáng tạo TMA bao gồm Trung tâm Phát triển Phần Mềm, Xưởng Phần mền, Trung tâm Sáng tạo, Trung tâm Dữ liệu, Trung tâm R&D, Học viện Công Nghệ.

1.1.2. *Tầm nhìn và sứ mệnh*

Trở thành một trong những công ty phần mềm hàng đầu thế giới, cung cấp các giải pháp và dịch vụ chất lượng cao cho khách hàng.

Với sứ mệnh "Technology for People & Business" cùng khẩu hiệu "Yes! We Can", Tập đoàn Công nghệ TMA sẽ tiếp tục đổi mới, sáng tạo, phát triển vững mạnh và đem lại những giá trị đích thực, vững bền cho khách hàng, đội ngũ nhân viên và cộng đồng.

1.1.3. *Giá trị cốt lõi*

- Sự tôn trọng
- Sự trung thực
- Sự cam kết

1.2. Tổng quan về vị trí Data Engineer

1.2.1. *Mô tả công việc.*

Data Engineer hay kỹ sư chuyên về dữ liệu thường làm các công việc như phân tích nguồn dữ liệu, tích hợp thông tin giữa các hệ thống với nhau, chuyển đổi và đồng bộ các dữ liệu trên nhiều hệ thống riêng biệt.

Data Engineer là người đề xuất các phương án và phụ trách việc cải thiện chất lượng các nguồn dữ liệu.

1.2.2. *Yêu cầu về kiến thức và kỹ năng.*

- Kỹ năng lập trình : Biết cơ bản về SQL, Python, Oracle. Phải nắm rõ các khái niệm giá trị đằng sau các công thức hiển thị ở màn hình.
- Kỹ năng phân tích logic : Cần sự chính xác và có tính liên kết với nhau, Data Engineer phải biết cách phân tích và tìm được ý nghĩa của những con số. Từ đó nhìn ra hướng giải quyết phù hợp
- Kỹ năng thiết kế và trình bày báo cáo.

- Kỹ năng giao tiếp.

1.2.3. *Cơ hội và thách thức việc làm.*

Doanh nghiệp hoạt động kinh doanh hiện nay không chỉ quan tâm đến quản lý nguồn dữ liệu mà còn mong tìm ra hướng giải quyết mở rộng tại nguyên để lưu trữ và kiểm soát nguồn dữ liệu. Để làm được cần có Data Engineer, vì thế đây là ngành nghề có xu hướng tuyển dụng tăng trong các năm tiếp theo.

Cơ hội phát triển nghề nghiệp trong tương lại đối với kỹ sư dữ liệu đang rất rộng mở. Vị trí này thường được các doanh nghiệp ưu ái và nắm giữ vai trò quan trọng trong bộ phận công ty.

1.3. Cơ sở lý thuyết

1.3.1. *Tổng quan về Cloud Computing*

Cloud Computing là truyền tải những dịch vụ khác nhau thông qua nền tảng internet, bao gồm những công cụ và ứng dụng như lưu trữ dữ liệu, máy chủ, cơ sở dữ liệu, hệ thống mạng và phần mềm.

- Những loại Cloud Computing:

a. Infrastructure as a Service (IaaS)

IaaS cung cấp cơ sở hạ tầng IT cơ bản dưới dạng dịch vụ. Người dùng có thể thuê các tài nguyên tính toán, lưu trữ, mạng và các tài nguyên cơ bản khác từ nhà cung cấp dịch vụ đám mây.

Đặc điểm:

- Tài nguyên tính toán, lưu trữ, mạng và các thành phần cơ sở hạ tầng khác được cung cấp trên nền tảng đám mây.
- Người dùng có toàn quyền kiểm soát hệ điều hành và ứng dụng chạy trên cơ sở hạ tầng.
- Thanh toán theo mức sử dụng (pay-as-you-go).

b. Platform as a Service (PaaS)

PaaS cung cấp nền tảng và môi trường để phát triển, kiểm thử và triển khai ứng dụng. Nhà cung cấp dịch vụ đám mây quản lý cơ sở hạ tầng, trong khi người dùng chỉ cần tập trung vào phát triển và quản lý ứng dụng.

Đặc điểm:

- Cung cấp môi trường phát triển tích hợp với các công cụ hỗ trợ phát triển, kiểm thử và triển khai ứng dụng.
- Nhà cung cấp dịch vụ chịu trách nhiệm về cơ sở hạ tầng.
- Hỗ trợ nhiều ngôn ngữ lập trình và framework.

c. Software as a Service (SaaS)

SaaS cung cấp ứng dụng phần mềm hoàn chỉnh dưới dạng dịch vụ. Người dùng có thể truy cập và sử dụng ứng dụng qua internet mà không cần cài đặt hoặc quản lý phần mềm.

Đặc điểm:

- Ứng dụng hoàn chỉnh được cung cấp dưới dạng dịch vụ.
- Người dùng không cần quản lý cơ sở hạ tầng hoặc nền tảng.
- Thanh toán theo mức sử dụng hoặc theo thuê bao.

1.3.2. *Các mô hình Cloud Computing*

a. Public Cloud:

Public Cloud là mô hình đám mây công cộng, nơi các tài nguyên máy tính được chia sẻ và quản lý bởi một nhà cung cấp dịch vụ đám mây.

Ưu điểm:

- Chi phí ban đầu thấp.
- Linh hoạt và mở rộng dễ dàng.
- Dịch vụ quản lý và bảo mật bởi nhà cung cấp.

Nhược điểm:

- Vấn đề bảo mật cần được chú trọng hơn do sử dụng chung.
- Hiệu suất có thể bị ảnh hưởng bởi hoạt động của người dùng khác.

b. Private Cloud:

Private Cloud là một mô hình đám mây riêng tư, thường được xây dựng và quản lý bởi một tổ chức hoặc doanh nghiệp. Đảm bảo rằng tài nguyên máy tính chỉ dành riêng cho sử dụng nội bộ và được bảo mật cao. Private Cloud thích hợp cho các tổ chức có nhu cầu kiểm soát toàn bộ môi trường đám mây của họ.

Ưu điểm:

- Độ tin cậy cao và kiểm soát toàn bộ.
- Bảo mật được quả lý nội bộ.
- Tùy chỉnh linh hoạt theo nhu cầu của người dùng.

Nhược điểm:

- Chi phí bàn đầu và chi phí vận hành cao.
- Khả năng mở rộng hạn chế so với Public Cloud.

c. Hybrid Cloud:

Hybrid Cloud là sự kết hợp giữa Public Cloud và Private Cloud, cho phép tổ chức kết hợp các tài nguyên công cộng và riêng tư. Lợi thế từ 2 hình thức giúp tối ưu hiệu suất và tiết kiệm chi phí. Các ứng dụng và dữ liệu có thể di chuyển giữa 2 mô hình một cách linh hoạt.

Ưu điểm:

- Linh hoạt trong việc kết hợp tài nguyên công cộng và riêng tư.
- Cung cấp tính toàn diện và hiệu quả về chi phí.
- Đáp ứng được các nhu cầu đặc biệt của từng ứng dụng.

Nhược điểm:

- Quản lý phức tạp hơn do phải làm việc với hai môi trường.
- Đòi hỏi sự tích hợp tốt giữa các hệ thống.

1.3.3. *Giới thiệu Microsoft Azure*

Microsoft Azure là một nền tảng điện toán đám mây toàn diện được cung cấp bởi Microsoft. Azure cung cấp một loạt các dịch vụ đám mây, bao gồm tính toán, phân tích, lưu trữ và mạng, giúp doanh nghiệp phát triển, triển khai và quản lý các ứng dụng trên đám

mây. Azure hỗ trợ nhiều ngôn ngữ lập trình, công cụ và framework khác nhau, giúp các nhà phát triển có thể xây dựng và triển khai ứng dụng một cách linh hoạt và hiệu quả.

a. Azure Blob Storage

Azure Blob Storage là dịch vụ lưu trữ đối tượng của Microsoft Azure, được thiết kế để lưu trữ và quản lý một lượng lớn dữ liệu không cấu trúc (văn bản hoặc dữ liệu nhị phân).

Các tính năng chính:

- Lưu trữ đối tượng không cấu trúc: Bao gồm văn bản, hình ảnh, video,...
- Mở rộng linh hoạt: Khả năng mở rộng tự động, cho phép lưu trữ hàng petabyte dữ liệu mà không cần quản lý phức tạp.
- Bảo mật và tuân thủ: Hỗ trợ mã hóa dữ liệu khi lưu trữ và trong quá trình truyền tải, cung cấp các tùy chọn kiểm soát truy cập chi tiết.
- Tích hợp với các dịch vụ Azure: Dễ dàng tích hợp với các dịch vụ khác của Azure như Azure Data Factory, Azure Databricks và Azure Machine Learning.

b. Azure Data Factory

Azure Data Factory (ADF) là một dịch vụ tích hợp dữ liệu dựa trên đám mây, cho phép tạo các pipelines để di chuyển và biến đổi dữ liệu. ADF hỗ trợ nhiều nguồn dữ liệu khác nhau, bao gồm các cơ sở dữ liệu quan hệ, dịch vụ đám mây và các kho dữ liệu lớn.

Các tính năng chính:

- Tích hợp dữ liệu: Hỗ trợ kết nối và tích hợp với hàng trăm nguồn dữ liệu khác nhau, cả trong và ngoài Azure.
- Chuyển đổi dữ liệu: Cung cấp các công cụ mạnh mẽ để làm sạch, biến đổi và tích hợp dữ liệu từ nhiều nguồn khác nhau.
- Quản lý quy trình: Cho phép xây dựng và quản lý các quy trình dữ liệu phức tạp với giao diện người dùng trực quan và các công cụ lập lịch tự động.
- Theo dõi và giám sát: Cung cấp các công cụ để theo dõi và giám sát các quy trình dữ liệu, đảm bảo dữ liệu được xử lý chính xác và kịp thời.

c. Azure Databricks

Azure Databricks là một dịch vụ phân tích dữ liệu lớn được xây dựng trên nền tảng Apache Spark, cung cấp một môi trường phát triển tích hợp (IDE) cho việc phát triển, kiểm thử và triển khai các ứng dụng phân tích dữ liệu và học máy.

Các tính năng chính:

- Hiệu suất cao: Tận dụng khả năng xử lý dữ liệu phân tán của Apache Spark để xử lý dữ liệu lớn một cách nhanh chóng và hiệu quả.
- Hỗ trợ nhiều ngôn ngữ lập trình: Hỗ trợ các ngôn ngữ phổ biến như Python, SQL... giúp dễ dàng xây dựng và triển khai các mô hình phân tích dữ liệu.
- Tích hợp với các dịch vụ Azure: Dễ dàng tích hợp với các dịch vụ lưu trữ và quản lý dữ liệu của Azure.
- Hợp tác và chia sẻ: Cung cấp các công cụ để các nhóm làm việc có thể hợp tác và chia sẻ các notebook, bảng điều khiển và mô hình học máy.

1.3.4. *Các ngôn ngữ và công cụ sử dụng.*

Ngôn ngữ

- **Python:** Python là một ngôn ngữ lập trình bậc cao, phổ biến trong lĩnh vực khoa học dữ liệu và phân tích dữ liệu. Python có nhiều thư viện hỗ trợ như Pandas, NumPy, Faker..., giúp việc xử lý và phân tích dữ liệu trở nên dễ dàng và hiệu quả.
- **PySpark:** PySpark là một API của Apache Spark, cho phép viết mã Spark bằng Python. PySpark được sử dụng để xử lý dữ liệu lớn (big data) trên các cụm máy tính (clusters) với tốc độ cao và hiệu quả. Trong dự án này, PySpark được sử dụng để xây dựng và triển khai các quy trình xử lý dữ liệu lớn.

Công cụ

- **Power BI:** Power BI là một công cụ phân tích kinh doanh của Microsoft, cho phép người dùng kết nối, phân tích và trực quan hóa dữ liệu.
- **Visual Studio Code (VS Code):** VS Code là một trình soạn thảo mã nguồn mạnh mẽ và linh hoạt của Microsoft, hỗ trợ nhiều ngôn ngữ lập trình và có nhiều tiện ích mở rộng. VS Code được sử dụng để viết mã nguồn fake dữ liệu trong dự án này.
- **SQL Server:** SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ của Microsoft, được sử dụng để lưu trữ và quản lý dữ liệu.

1.3.5. *Cấu Trúc của Medallion Architecture:*

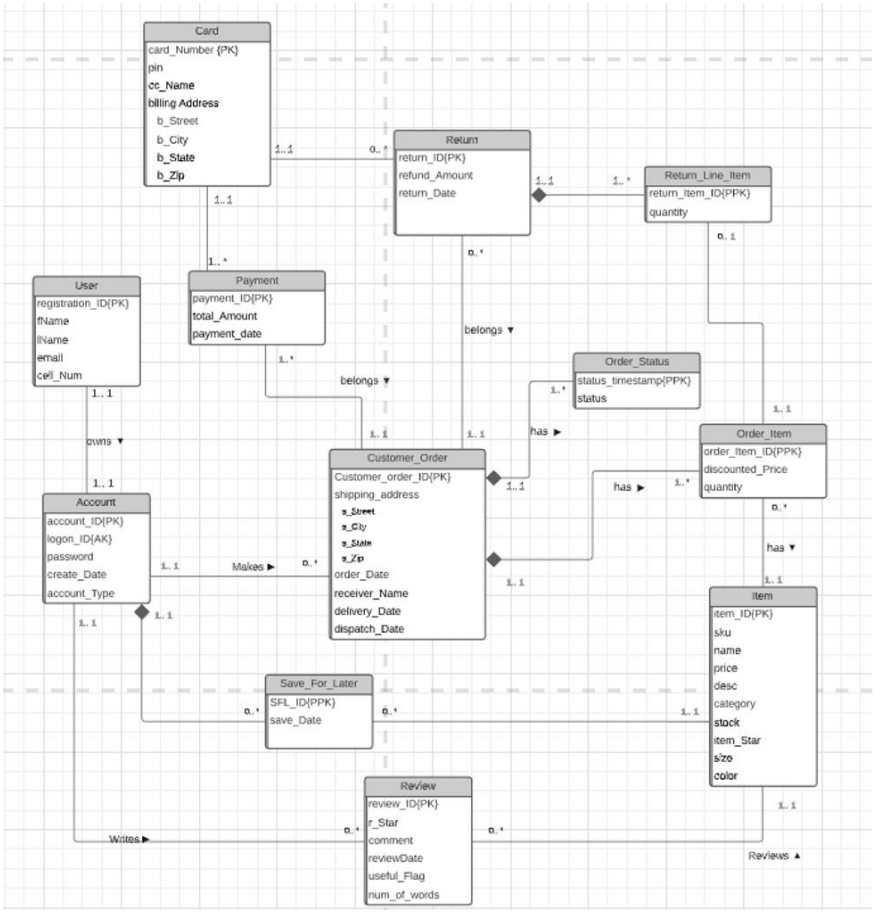
Medallion Architecture là một mô hình thiết kế dữ liệu trong hệ thống Data Lakehouse, nhằm tổ chức dữ liệu theo các cấp độ hoặc lớp khác nhau để tối ưu hóa quy trình lưu trữ, xử lý, và phân tích dữ liệu.

1. **Bronze Layer (Raw Data):**
    - Mô Tả: Lớp đầu tiên là nơi lưu trữ dữ liệu thô và không được xử lý..
2. **Silver Layer (Cleansed Data):**
    - Mô Tả: Lớp thứ hai là nơi dữ liệu được làm sạch và biến đổi có thể kết hợp dữ liệu từ nhiều nguồn khác nhau.
    - Dữ Liệu: Dữ liệu có thể bao gồm các bảng dữ liệu đã được chuẩn hóa, dữ liệu đã được phân loại, hoặc dữ liệu đã được kết hợp từ nhiều nguồn.
3. **Gold Layer (Aggregated Data):**
    - Mô Tả: Lớp cuối cùng là nơi lưu trữ dữ liệu đã được tổng hợp, phân tích, và có thể được tổ chức theo các dạng phù hợp cho báo cáo và phân tích nâng cao.
    - Dữ Liệu: Dữ liệu có thể bao gồm các bảng tổng hợp, các mô hình dữ liệu đã được tối ưu hóa, hoặc các chỉ số và KPI (Key Performance Indicators).

CHƯƠNG 2. GIỚI THIỆU DỮ LIỆU VÀ THIẾT KẾ PIPELINES

2.1. Mô tả bộ dữ liệu Ecommerce

2.1.1. *Cấu trúc database*



Hình 2.1 ERD bộ dữ liệu Ecommerce

2.1.2. *Các thành phần chi tiết của database.*

- **Bảng Account** : có 330 dòng

	Column Name	Data Type	Allow Nulls
🔑	account_ID	varchar(10)	☐
	registration_ID	varchar(10)	☐
	fName	varchar(20)	☐
	lName	varchar(40)	☐
	email	varchar(320)	☐
	cell_Num	char(12)	☑
	logon_ID	varchar(20)	☐
	password	varchar(32)	☐
	create_Date	date	☐
	account_Type	char(1)	☐

Hình 2.2 Bảng Account

account_Type: Loại tài khoản chỉ có thể là 'P' (cá nhân) hoặc 'B' (doanh nghiệp).

- **Bảng Card** : có 330 dòng

	Column Name	Data Type	Allow Nulls
🔑	card_number	varchar(16)	☐
	cc_name	varchar(40)	☐
	pin	char(4)	☐
	b_street	varchar(40)	☑
	b_city	varchar(20)	☑
	b_state	char(2)	☑
	b_zip	char(10)	☑

Hình 2.3 Bảng Card

- **Bảng Item** : có 330 dòng

	Column Name	Data Type	Allow Nulls
🔑	item_ID	varchar(10)	☐
	sku	char(8)	☐
	name	varchar(40)	☐
	price	decimal(12, 2)	☐
	item_desc	varchar(100)	☑
	category	varchar(10)	☑
	stock	int	☐
	item_star	int	☑
	item_sizes	varchar(5)	☑
	color	varchar(20)	☑

Hình 2.4 Bảng Item

- **Bảng Customer_order** : có 330 dòng

	Column Name	Data Type	Allow Nulls
🔑	order_ID	varchar(10)	☐
	account_ID	varchar(10)	☐
	s_street	varchar(40)	☑
	s_City	varchar(20)	☑
	s_State	char(2)	☐
	s_Zip	char(10)	☑
	order_date	date	☑
	receiver_name	varchar(40)	☐
	delivery_date	date	☑
	dispatch_date	date	☑

Hình 2.5 Bảng Customer_order

- **Bảng Order_Item** : có 328 dòng

Column Name	Data Type	Allow Nulls
order_ID	varchar(10)	☐
order_item_ID	varchar(10)	☐
item_ID	varchar(10)	☐
discounted_price	decimal(12, 2)	☐
quantity	int	☐

Hình 2.6 Bảng Order_Item

order_ID (FK) : order_ID tham chiếu đến Customer_Order(order_ID)

item_ID (FK) : item_ID tham chiếu đến Item(item_ID).

- **Bảng Order_status** : có 1000 dòng

Column Name	Data Type	Allow Nulls
status_Order_ID	varchar(10)	☐
status	varchar(40)	☑
status_Timestamp	varchar(20)	☐

Hình 2.7 Bảng Order_status

status_Order_ID (FK) : tham chiếu đến Customer_Order(order_ID).

- **Bảng Payment** : có 231 dòng

Column Name	Data Type	Allow Nulls
payment_id	varchar(10)	☐
order_id	varchar(10)	☐
payment_date	date	☑
total_amount	decimal(12, 2)	☑
card_n	varchar(16)	☑

Hình 2.8 Bảng Payment

order_id (FK) : tham chiếu đến Customer_Order(order_ID).

card_n (FK) : tham chiếu đến Card(card_number)

- **Bảng Return** : có 323 dòng

Column Name	Data Type	Allow Nulls
returns_ID	varchar(10)	☐
refund_amount	decimal(12, 2)	☐
returns_date	date	☐
order_ID	varchar(10)	☐
card_n	varchar(16)	☐

Hình 2.9 Bảng Return

order_ID (FK) : tham chiếu đến Customer_Order(order_ID)

card_n (FK) : tham chiếu đến Card(card_number).

- **Bảng Return_line_item** : có 330 dòng

Column Name	Data Type	Allow Nulls
return_ID	varchar(10)	☐
return_item_ID	varchar(10)	☐
quantity	int	☐
order_item	varchar(10)	☐
order_ID	varchar(10)	☐

Hình 2.10 Bảng Return_line_item

order_item và order_ID (FK) : tham chiếu đến order_ID và order_item_ID của bảng Order_Item.

- **Bảng Review** : có 330 dòng

Column Name	Data Type	Allow Nulls
review_ID	varchar(10)	☐
r_star	decimal(12, 2)	☐
comments	varchar(255)	☑
reviewDate	date	☐
useful_flag	int	☑
num_of_words	int	☑
review_account	varchar(10)	☐
review_item	varchar(10)	☐

Hình 2.11 Bảng Review

review_account (FK) : tham chiếu với account_ID của bảng Account.

review_item (FK) : liên kết review_item với item_ID của bảng Item.

- **Bảng Save_for_late** : có 330 dòng

Column Name	Data Type	Allow Nulls
sfl_account_ID	varchar(10)	☐
sfl_ID	varchar(10)	☐
sfl_item_ID	varchar(10)	☐
save_Date	date	☐

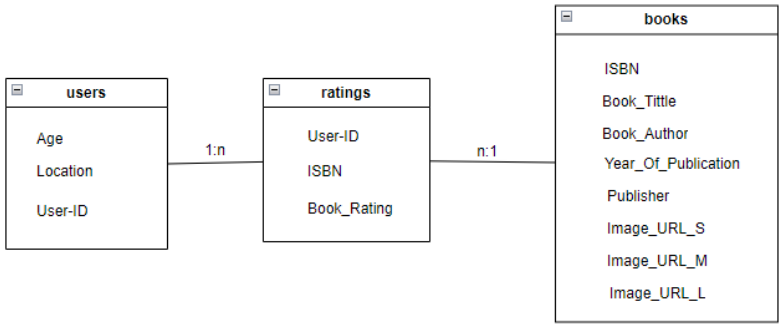
Hình 2.12 Bảng Save_for_late

sfl_account_ID (FK) : liên kết với account_ID của bảng Account.

2.2. Mô tả bộ dữ liệu Books

2.2.1. *Cấu trúc database*

- Dữ liệu trực tuyến cho sách từ Amazon cùng với xếp hạng của người dùng và người dùng đã mua chúng.



Hình 2.13 ERD của bộ dữ liệu Books

- Mô tả mối quan hệ của các bảng:

  - Bảng users liên kết với bảng ratings thông qua User-ID với mối quan hệ 1-nhiều (1 người đọc có thể đánh giá nhiều sách).
  - Bảng books liên kết với bảng ratings thông qua ISBN với mối quan hệ 1-nhiều (1 quyển sách có thể được đánh giá nhiều lần).

2.2.2. *Các thành phần chi tiết của database.*

- **Bảng Books :** có 271379 dòng

Column Name	Data Type	Allow Nulls
ISBN	nvarchar(10)	☐
Book_Title	nvarchar(300)	☐
Book_Author	nvarchar(300)	☐
Year_Of_Publication	smallint	☑
Publisher	nvarchar(300)	☐
Image_URL_S	nvarchar(100)	☑
Image_URL_M	nvarchar(100)	☑
Image_URL_L	nvarchar(100)	☑

Hình 2.14 Bảng Books

- **Bảng Users** : có 1149780 dòng

Column Name	Data Type	Allow Nulls
[User-ID]	nvarchar(50)	☐
Location	nvarchar(300)	☑
Age	int	☑

Hình 2.15 Bảng Users

- **Bảng Ratings :** có 278859 dòng

Column Name	Data Type	Allow Nulls
User_ID	nvarchar(50)	☐
ISBN	nvarchar(10)	☐
Book_Rating	tinyint	☐

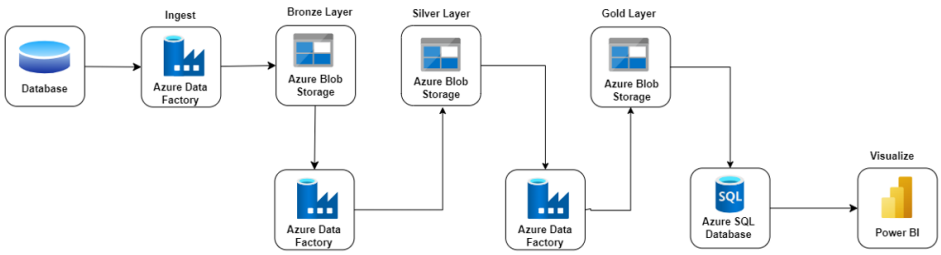
Hình 2.16 Bảng Ratings

User-ID (FK) : tham chiếu đến User-ID của bảng Users, kiểu string.

ISBN (FK) : tham chiếu đến ISBN của bảng Books, kiểu string.

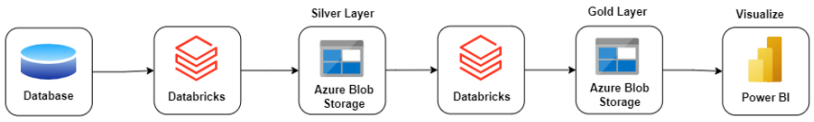
2.3. Xây dựng pipelines

- Pipeline 1 (sử dụng Azure Data Factory)



Hình 2.17 Pipeline sử dụng Azure Data Factory

- Pipeline 2 (sử dụng Azure Databricks)



Hình 2.18 Pipeline sử dụng Azure Databricks

2.4. Các thành phần chính

2.4.1. *Nhập dữ liệu (Ingestion)*

Nguồn dữ liệu: Mô tả các nguồn dữ liệu đầu vào. Theo dự án thì lấy dữ liệu từ máy chủ SQL và file (.csv).

- Pipeline 1: Sử dụng dữ liệu Ecommerce từ nguồn (SQL Server) truyền vào Azure Data Factory. Sau khi Azure Data Factory lấy dữ liệu thì cần được lưu trữ trong Azure Data Blob dưới dạng dữ liệu thô (Bronze Layer).
- Pipeline 2: Dữ liệu books (.csv) được truyền trực tiếp đến Azure Databricks để xử lý mà không cần thông qua lớp Bronze Layer.

2.4.2. *Xử lý dữ liệu (Processing)*

- **Azure Databricks**: Sử dụng Azure Databricks để xử lý ETL và làm sạch dữ liệu. Azure Databricks cung cấp môi trường Spark mạnh mẽ cho việc xử lý dữ liệu lớn.
- **Azure Data Factory**: Sử dụng ADF để xử lý ETL và làm sạch dữ liệu.

2.4.3. *Lưu trữ dữ liệu (Storing)*

- **Azure Blob Storage**: Dữ liệu đã qua xử lý có thể được lưu trữ trong Azure Blob Storage để dễ dàng truy cập và quản lý.

2.4.4. *Phân tích dữ liệu (Analysis)*

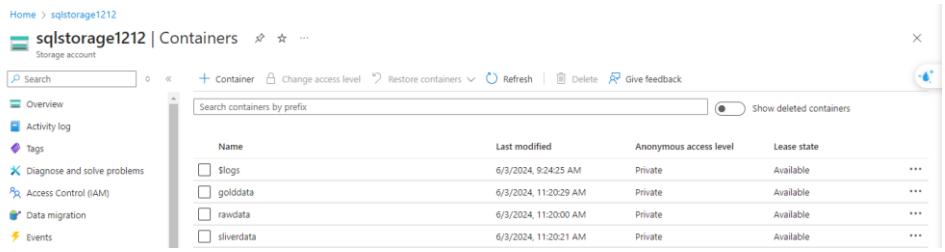
- **Power BI**: Sử dụng Power BI để tạo báo cáo và dashboard trực quan hóa dữ liệu, cung cấp thông tin chi tiết và hỗ trợ ra quyết định.

CHƯƠNG 3. TRIỂN KHAI PIPELINES TRÊN CLOUD

3.1. Pipeline 1 (Sử Dụng Azure Data Factory)

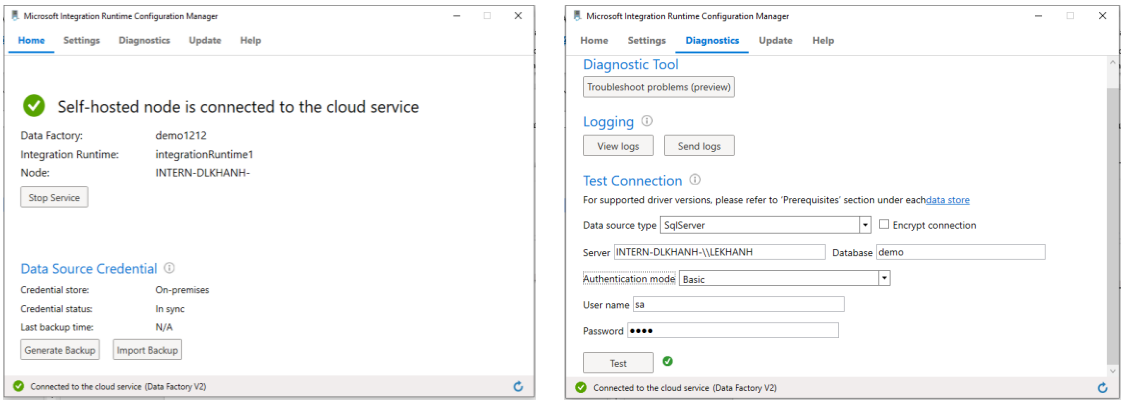
3.1.1. *Bước chuẩn bị*

- Tạo Storage Container (rawdata, sliverdata, golddata) để chứa dữ liệu trong mỗi lần xử lý.



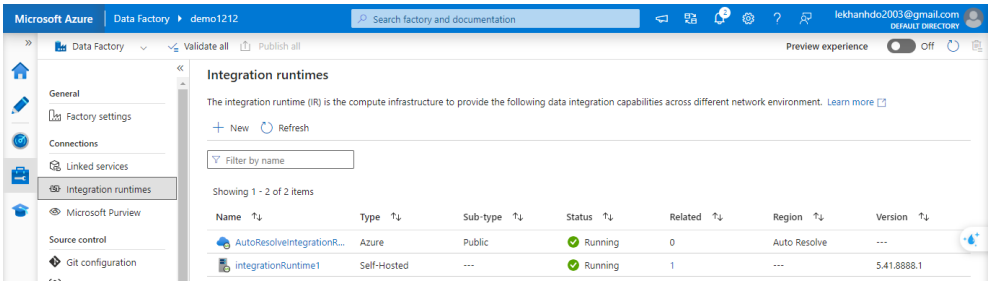
Hình 3.1 Tạo thư mục Storage Container

- Tạo kết nối từ SQL Server đến Azure Data Factory.



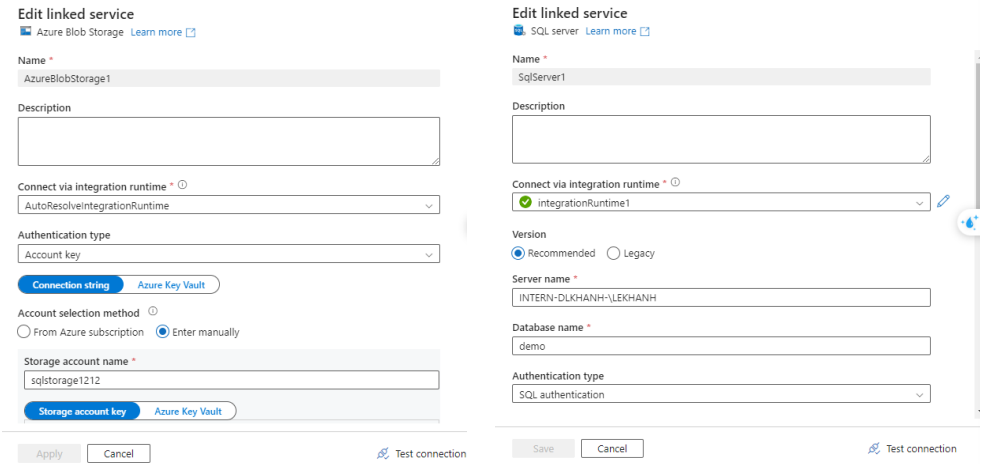
Hình 3.2 Kết nối đến Azure Data Factory

- Integration runtimes sử dụng để liên kết ADF với SQL Server giúp di chuyển và chuyển đổi dữ liệu thô thành định dạng có thể sử dụng cho kho dữ liệu.



Hình 3.3 Liên kết ADF với SQL Server

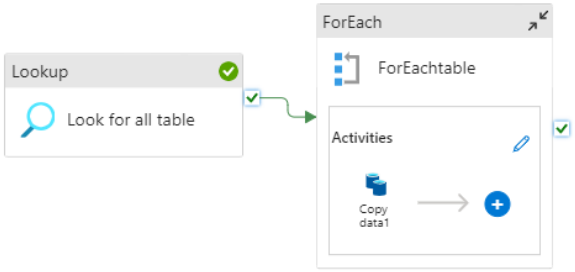
- Tạo liên kết kết nối tới máy chủ.



Hình 3.4 Tạo linked service

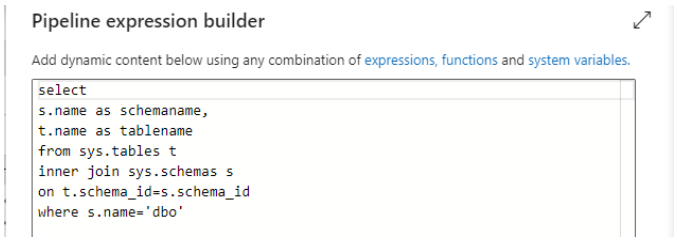
3.1.2. *Nhập dữ liệu*

- Tạo pipeline với "Look up" cho phép truy xuất dữ liệu từ các nguồn bên ngoài (SQL Server) và tích hợp nó.



Hình 3.5 Tạo pipelines

- Sử dụng Lookup để lọc tất cả tên bảng có trong database của SQL Server.

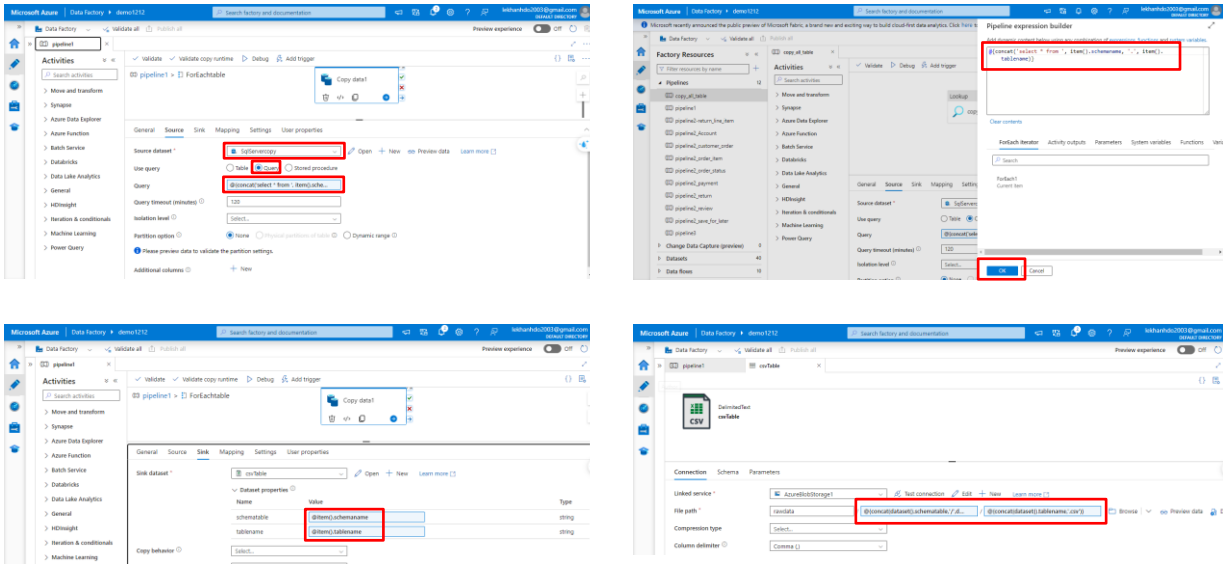


Hình 3.6 Lọc tên bảng

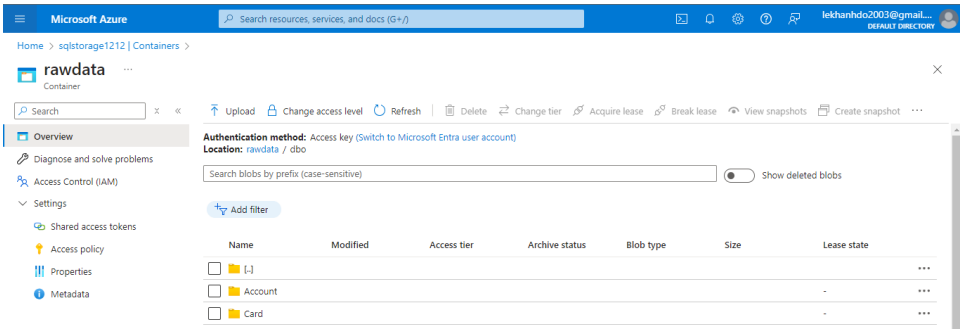
- Tạo “ForEach”. Trong "ForEach", tạo hoạt động "Copy data" đóng vai trò là khối xây dựng để thực hiện di chuyển hoặc sao chép dữ liệu giữa các kho dữ liệu khác nhau. Nó sao chép dữ liệu từ tập dữ liệu nguồn (SQL Server) sang tập dữ liệu đích lưu vào

Azure Blob Storage, mỗi lần lặp lại một lần, dựa trên các mục trong vòng lặp "ForEach".

- Cấu hình Source và Sink vào thư mục Rawdata trong Copy data.



Hình 3.7 Cấu hình nơi lưu dữ liệu

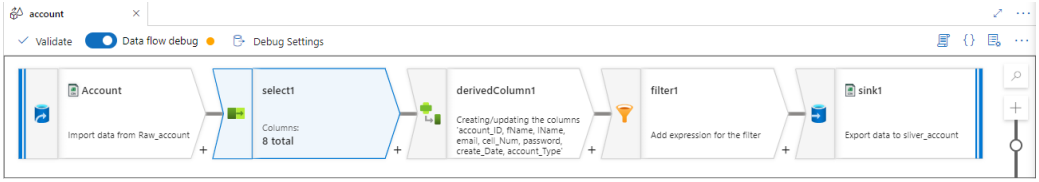


Hình 3.8 Dữ liệu được lưu ở Rawdata

3.1.3. *Bronze Layer đến Silver Layer*

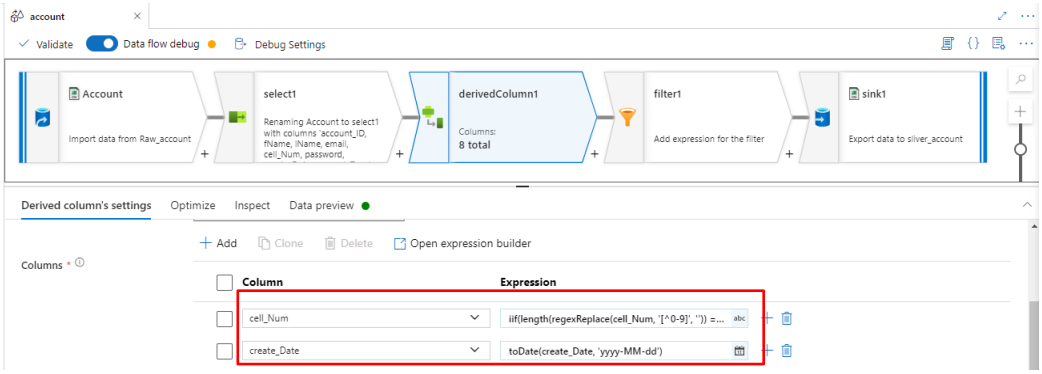
- **Bảng Account**

- Sử dụng Select loại bỏ những cột không cần thiết trong việc phân tích như cột registration_ID, logon_ID và account_Type.



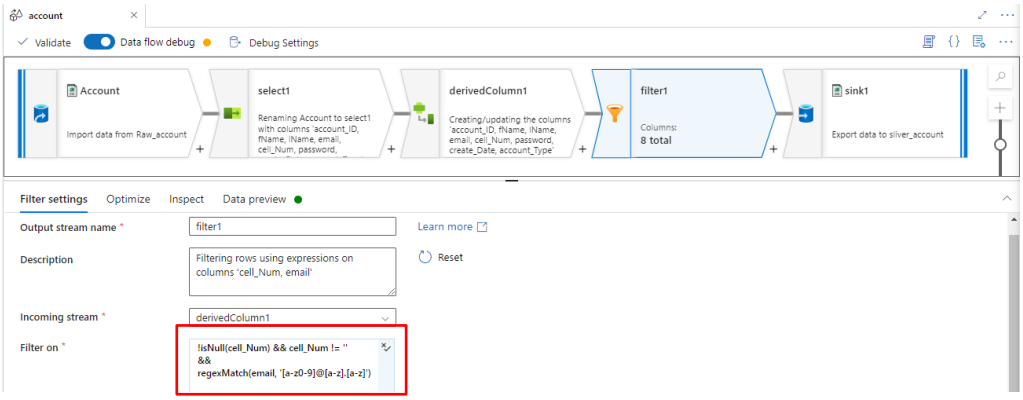
Hình 3.9 Pipeline bảng Account

- Dùng Aggregate chuyển đổi kiểu dữ liệu cho cột thời gian create_Date, cột cell_num về đúng dạng số điện thoại xxxx-xxx-xxx, cột (fname, lname) loại bỏ ký tự đặc biệt.



Hình 3.10 Aggregate trong pipeline Account

- Sử dụng 'filter' để loại bỏ các hàng trong đó cell_Num null hoặc trống và cột email không đúng định dạng.

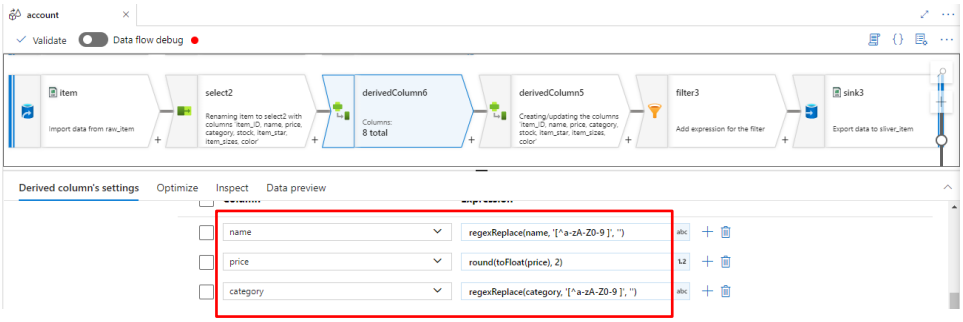


Hình 3.11 Filter trong pipeline Account

- Sau khi hoàn tất các bước xử lý, chuyển dữ liệu lên cloud trong thư mục silverdata để xử lý tiếp nếu dữ liệu không sạch.

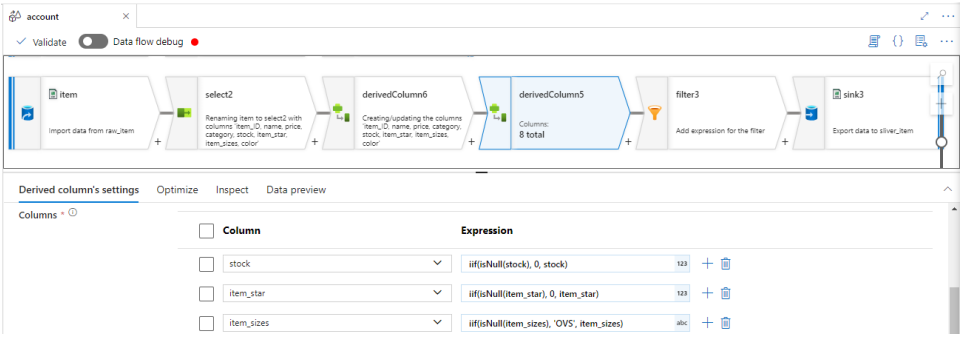
- **Bảng Item**

- Xử lý các kiểu dữ liệu trong cột Category, stock, item_star và loại bỏ các ký tự đặc biệt ở cột 'name'.



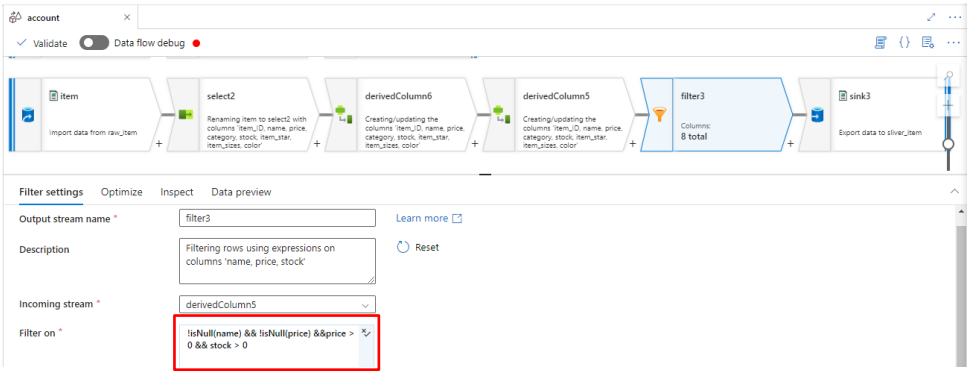
Hình 3.12 Aggregate trong pipeline Item

- Điền giá trị 0 vào cột stock, item_star nếu null, điền OVS khi cột item_sizes null, cột color null thì điền ‘Black’, cột category null thì điền ‘Unknown’.



Hình 3.13 Aggregate trong pipeline Item xử lý logic

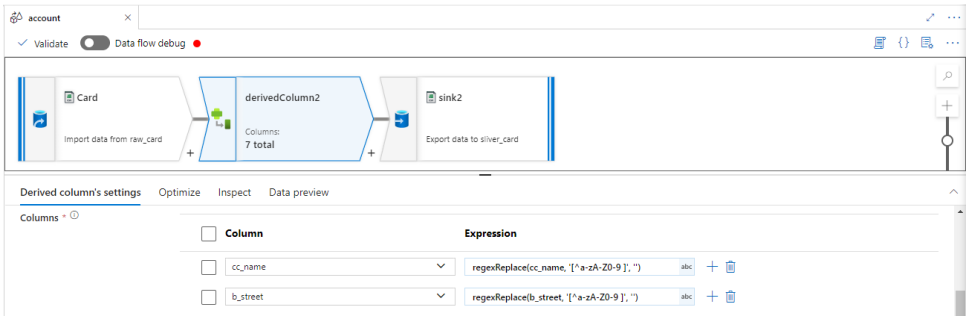
- Lọc khi name không null và price, stock >=0.



Hình 3.14 Filter trong pipeline Item

- **Bảng Card**

- Xử lý kiểu dữ liệu của các cột sao cho phù hợp và loại bỏ các kí tự đặc biệt trong các cột kiểu string.

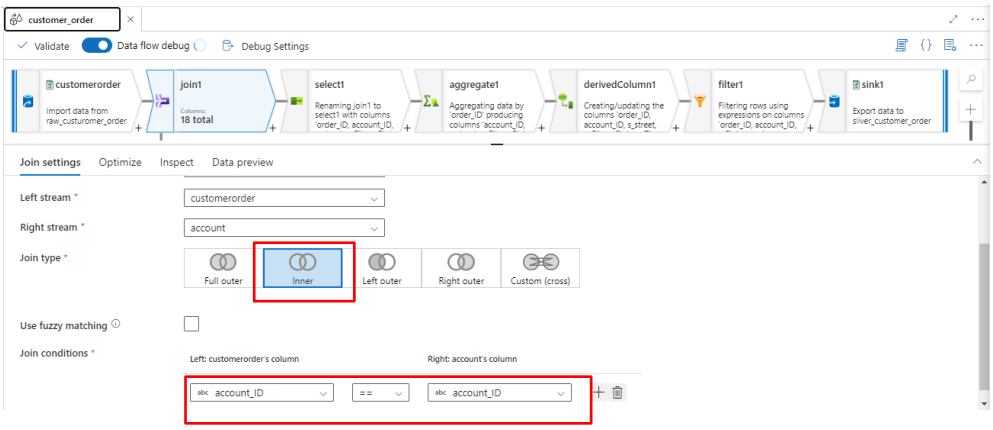


Hình 3.15 Pipeline của bảng Card

- **Bảng Customer_order**

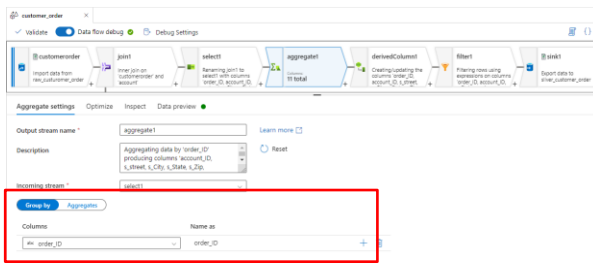
Hình 3.16 Pipeline của bảng Customer_order

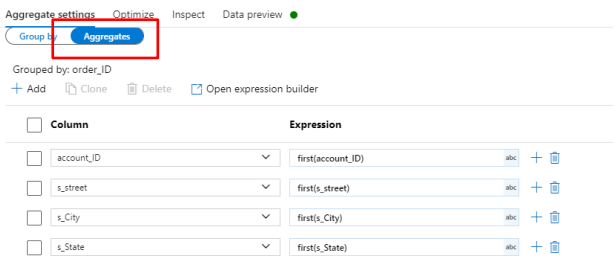
- Nối các bảng theo khóa ngoại tương ứng, sử dụng kiểu inner join để lấy các hàng chung của 2 bảng. Chọn các cột cần thiết trong bảng khi nối nhiều bảng lại với nhau.



Hình 3.17 Join trong pipeline Customer_order

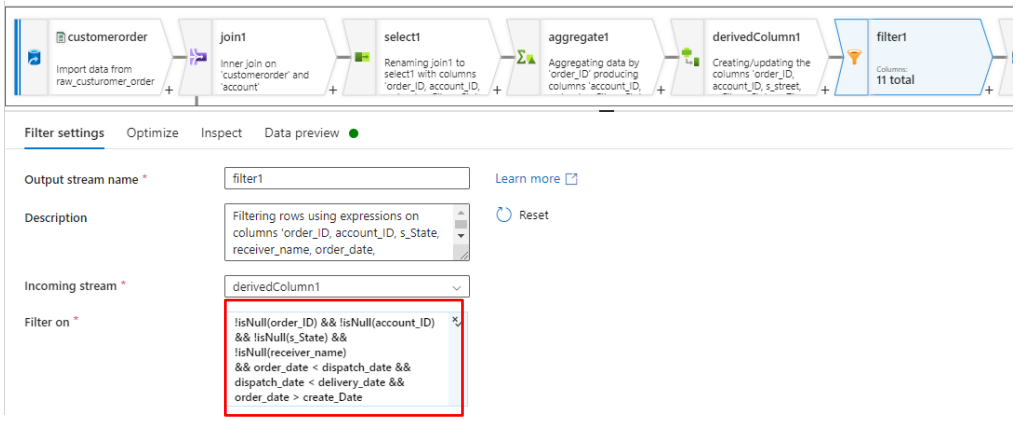
- Lọc các giá trị trùng lặp trong bảng theo order_ID và lấy hàng xuất hiện đầu tiên khi có dữ liệu trùng lặp.





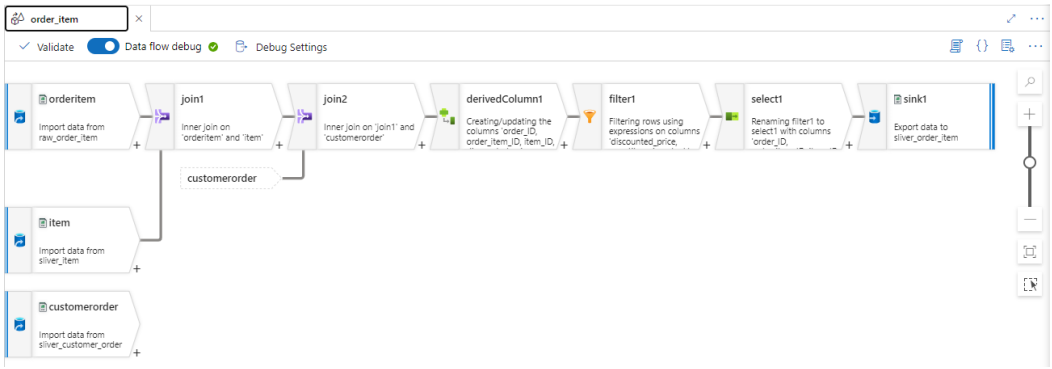
Hình 3.18 Lọc dữ liệu trùng

- Lọc các dòng với điều kiện các ô không rỗng và điều kiện logic (create_Date < order_date < dispatch_date < delivery_date).



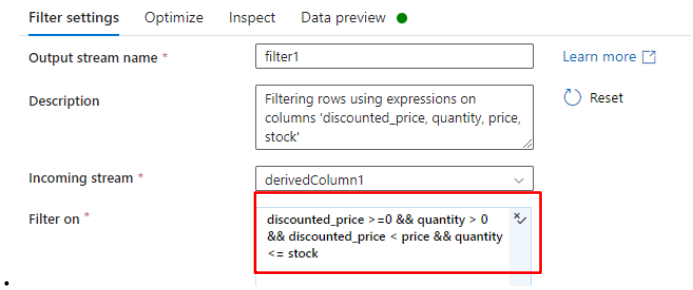
Hình 3.19 Lọc với điều kiện logic

- **Bảng Order_item**



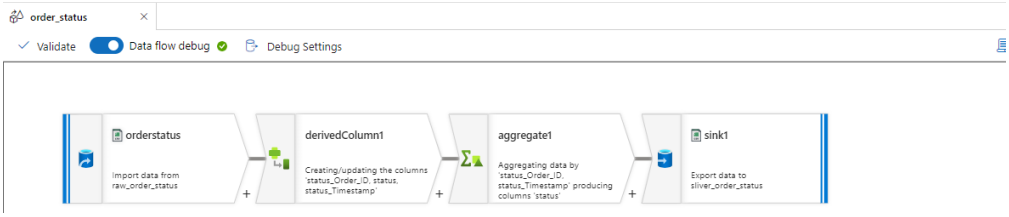
Hình 3.20 Pipeline bảng Order_item

- Chuyển đổi kiểu dữ liệu cho phù hợp tương tự như các bảng trên.
- Lọc các dòng thỏa điều kiện logic (0 <= discounted_price < price và 0 <= quantity <= stock).



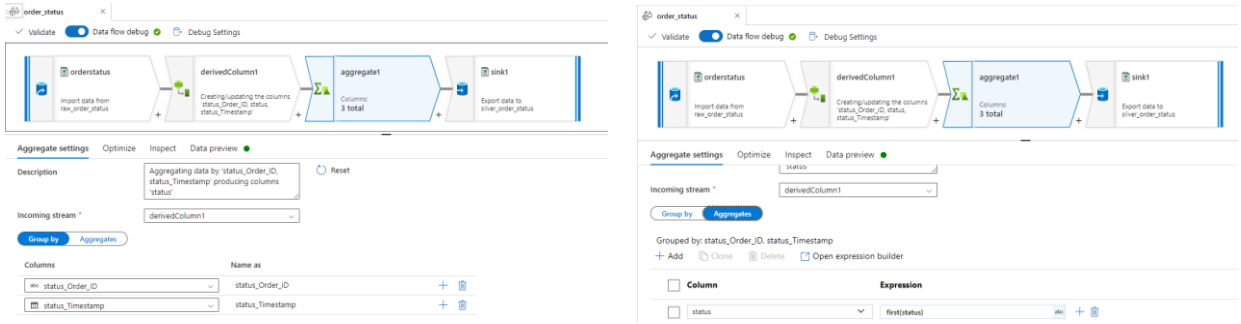
Hình 3.21 Lọc dữ liệu theo điều kiện logic

- **Bảng Order_status**



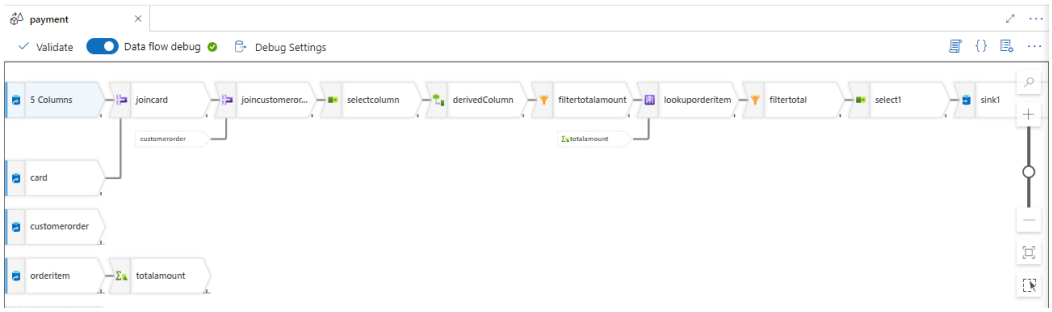
Hình 3.22 Pipeline bảng Order_status

- Chuyển đổi kiểu dữ liệu cho phù hợp.
- Lọc dữ liệu trùng lặp.



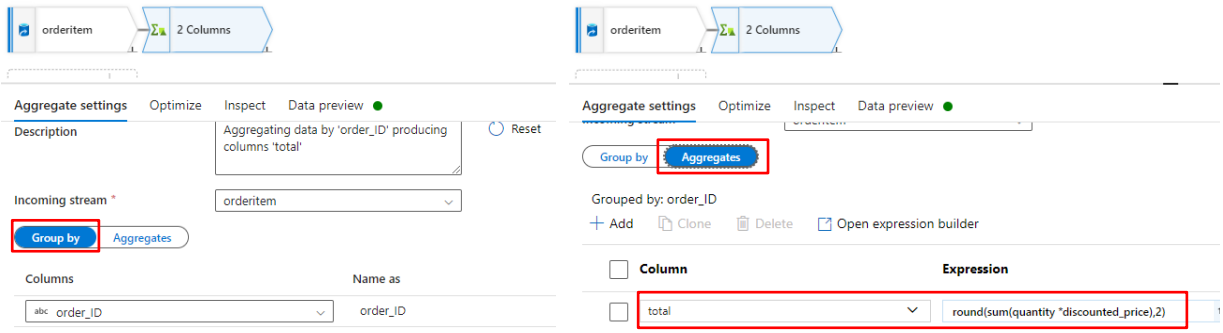
Hình 3.23 Lọc dữ liệu trùng lặp bảng Order_status

- **Bảng Payment**



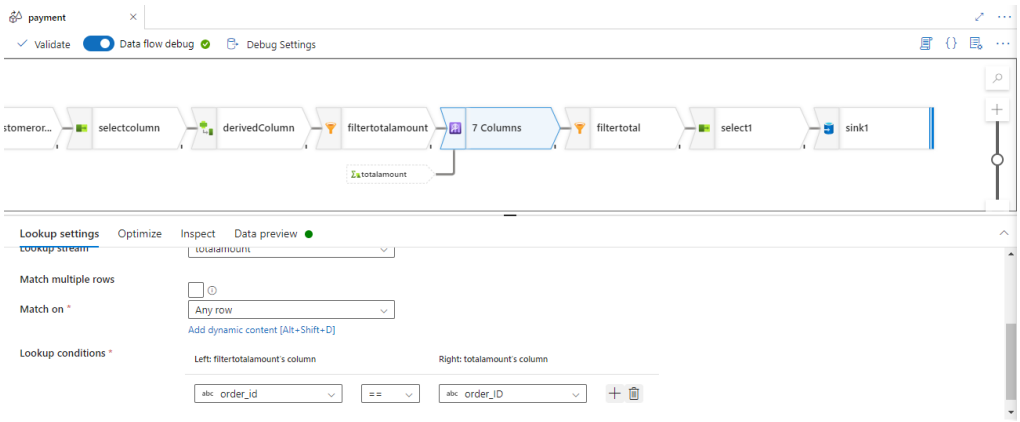
Hình 3.24 Pipeline bảng Payment

- Ở đây chúng ta xử lý bảng order_item, cộng cột total theo công thức sum(quantity* detect_price) theo order_ID.



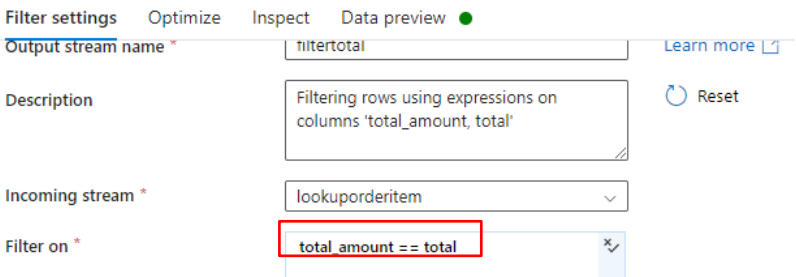
Hình 3.25 Tính cột Total theo công thức

- Thực hiện join giữa các bảng theo khóa ngoại tương ứng và chuyển đổi kiểu dữ liệu cột cho phù hợp.
- Dùng 'Look up' bảng order_item để tìm thông tin về total_amount.



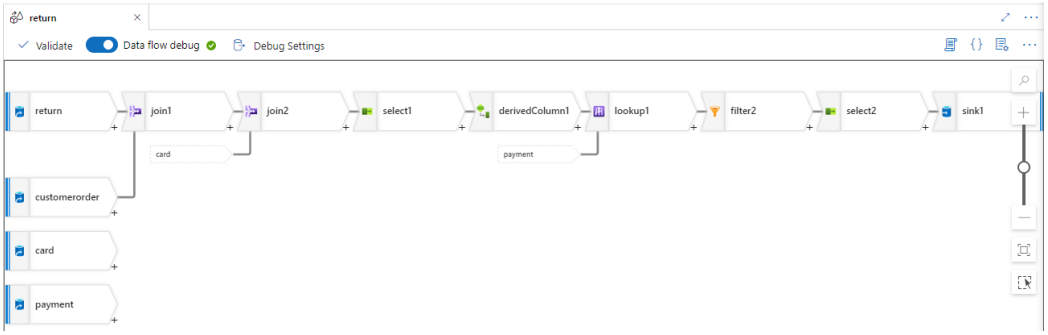
Hình 3.26 Tìm kiếm thông tin về total_amount

- Dùng filter để so sánh cột total_amount từ bảng order_item tính theo công thức với cột total_amount trong bronze data và chỉ lấy dữ liệu khi total = total_amount.



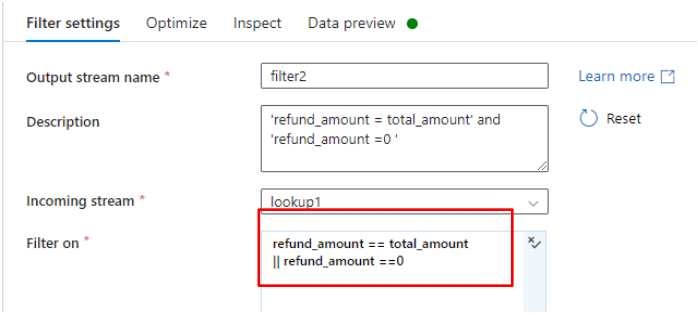
Hình 3.27 Lọc theo điều kiện logic để lấy cột total

- **Bảng Return**



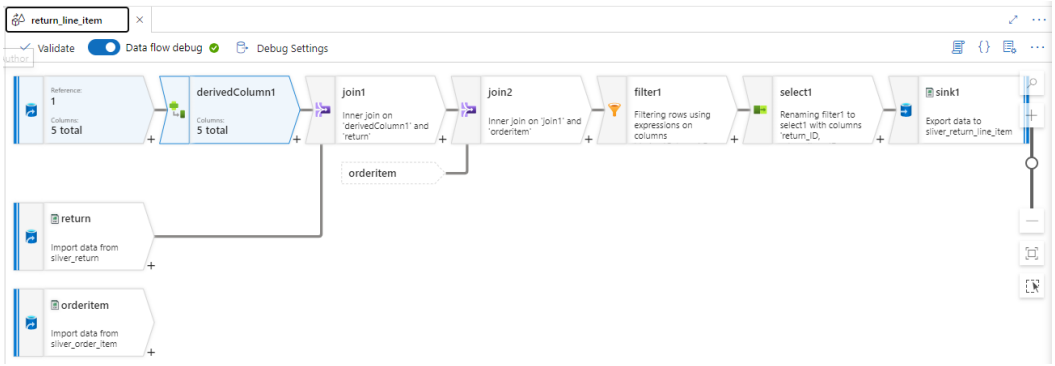
Hình 3.28 Pipeline của bảng Return

- Nối các bảng theo khóa ngoại và chọn các cột cần thiết để xử lý.
- Chuyển đổi các cột đã chọn sang kiểu dữ liệu thích hợp.
- Dùng Look up để tìm kiếm thông tin trong bảng thanh toán và lọc với điều kiện logic (0 <= return_amount = total_amount).



Hình 3.29 Tìm kiếm và lọc với điều kiện logic

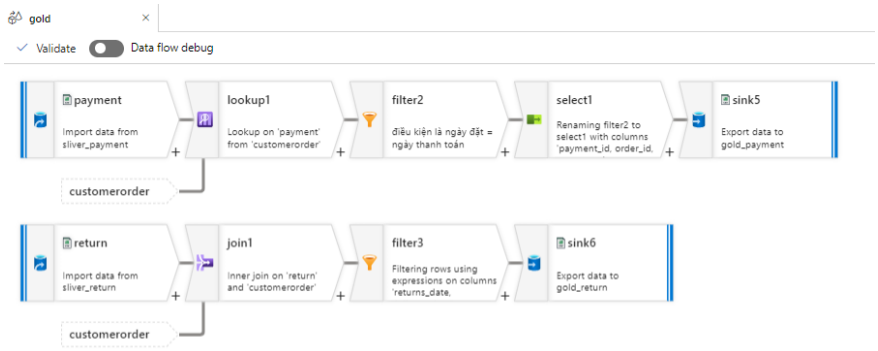
- **Bảng Return_line_item**



Hình 3.30 Pipeline bảng Return_line_item

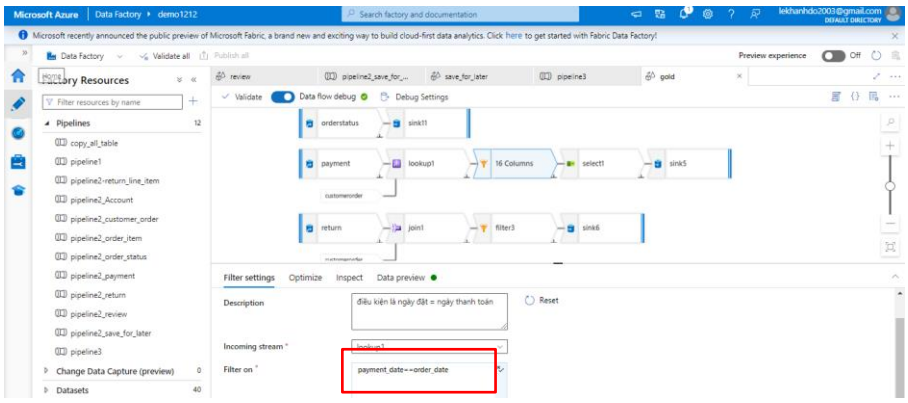
- Tương tự như các bảng trên, bảng này cũng được lọc theo các hoạt động để các bảng có liên kết logic và được xử lí làm sạch.

3.1.4. *Từ Silver Layer qua Gold Layer*

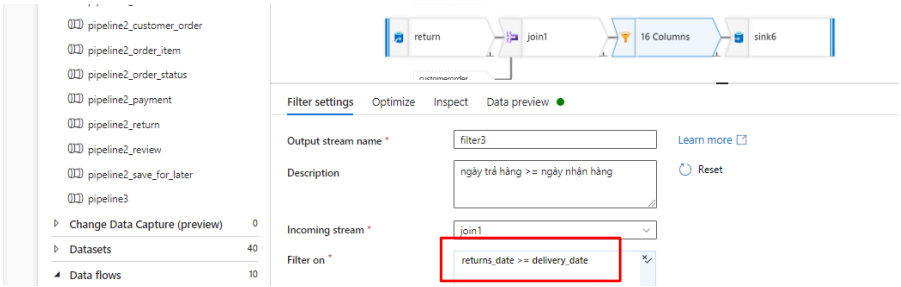


Hình 3.31 Xử lý từ Silver Layer qua Gold Layer

- Kiểm tra và lọc những dữ liệu thỏa: bảng Payment có ngày đặt = ngày thanh toán (payment date = order date) và bảng Return có ngày trả hàng >= ngày nhận hàng (return date >= delivery date).

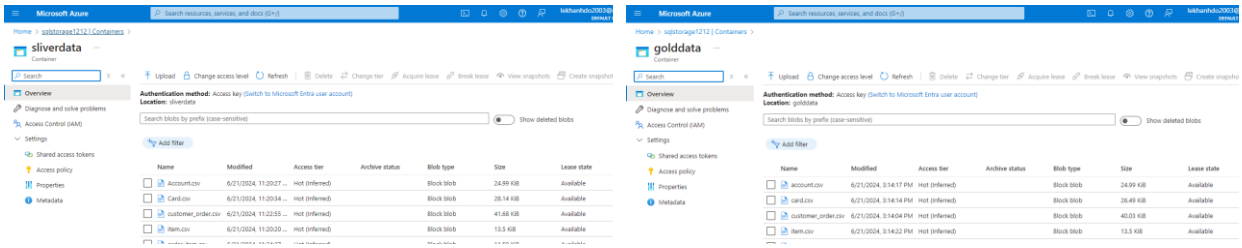


Hình 3.32 Lọc dữ liệu ở Bảng Payment (Gold Layer)



Hình 3.33 Lọc dữ liệu ở Bảng Return (Gold Layer)

- Dữ liệu sau khi thực hiện hết tất cả các bước xử lý thì được lưu vào thư mục tương ứng (silverdata và golddata).



Hình 3.34 Thư mục sliverdata và golddata

3.2. Pipeline 2 (Sử Dụng Azure Databricks)

- Đọc file (.csv) với các cột được phân cách bởi dấu ‘;’ và đọc bởi mã ‘iso-8859-1’.

```
20 hours ago (42s)
df_books= spark.read.csv('dbfs:/FileStore/end_of_term/books.csv', header=True, inferSchema=True, sep=';',encoding='iso-8859-1')
df_ratings= spark.read.csv('dbfs:/FileStore/end_of_term/ratings.csv', header=True, inferSchema=True, sep=';',encoding='iso-8859-1')
df_users= spark.read.csv('dbfs:/FileStore/end_of_term/users.csv', header=True, inferSchema=True, sep=';',encoding='iso-8859-1')
```

Hình 3.35 Đọc file (.csv)

- Kiểm tra sự duy nhất của khóa chính trong các bảng. Nếu có trùng lặp khóa chính thì xóa dòng trùng lặp.

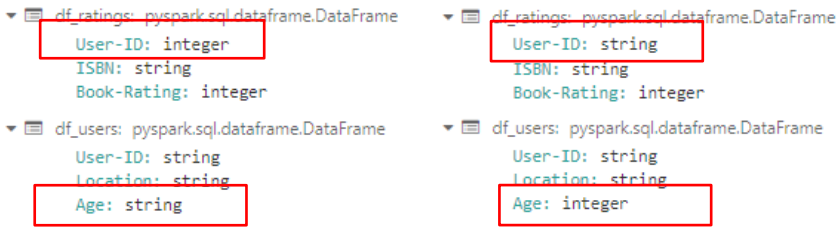
```
20 hours ago (<1s)
df_books = df_books.dropDuplicates(["ISBN","Book-Title"])
df_users= df_users.dropDuplicates(["User-ID"])
df_ratings = df_ratings.dropDuplicates(["ISBN", "User-ID"])
```

Hình 3.36 Xóa các dòng trùng lặp

- Xử lí cột Age trong bảng User (chuyển đổi kiểu dữ liệu sang integer).

```
2 minutes ago (<1s)
from pyspark.sql.functions import col
from pyspark.sql import functions as F
# cột Age chỉ chứa số
df_users = df_users.withColumn("Age", F.regexp_extract("Age", r"\d+", 0))
# Chuyển đổi kiểu dữ liệu
df_ratings = df_ratings.withColumn("User-ID", col("User-ID").cast("string"))
df_users = df_users.withColumn("Age", col("Age").cast("integer"))
```

Hình 3.37 Đổi kiểu dữ liệu



Hình 3.38 Trước và sau khi đổi kiểu dữ liệu

- Lọc các kí tự đặc biệt của tất cả các cột kiểu string trong tất cả các bảng trừ cột Location (không được xóa dấu ‘,’) và các cột URL (có chứa đường link ảnh).

```
20 hours ago (4s)                                                    8
from pyspark.sql.functions import col, regexp_replace
from pyspark.sql.types import StringType
def clean_all_string_columns_except_location(df):
    string_columns = [col_name for col_name, col_type in df.dtypes if col_type == "string" and col_name != "Location" and "URL" not in
    col_name]
    for column_name in string_columns:
        df = df.withColumn(column_name, regexp_replace(col(column_name), r"[^0-9a-zA-Z\s\.\(\)]", ""))
    return df
df_books = clean_all_string_columns_except_location(df_books)
df_users = clean_all_string_columns_except_location(df_users)
df_ratings = clean_all_string_columns_except_location(df_ratings)

#xóa chữ số trong các cột bảng Books
df_books = df_books.withColumn("Book-Author", regexp_replace(col("Book-Author"), "[^a-zA-Z\s\.\(\)]", ""))
df_books = df_books.withColumn("Publisher", regexp_replace(col("Publisher"), "[^a-zA-Z\s\.\(\)]", ""))
```

Hình 3.39 Lọc kí tự đặc biệt

ISBN	Book-Title
3257207522	Der K\u00f6?\u00f6nig in Gelb.
3257208626	Fahrenheit 451
3257208634	Die Mars- Chroniken. Roman in Erz\u00f6?\u00e4hlungen.
3257208669	Das B\u00f6?\u00dfse kommt auf leisen Sohlen.

ISBN	Book-Title
3257207522	Der Knig in Gelb
3257208626	Fahrenheit 451
3257208634	Die Mars Chroniken Roman in Erzhlungen
3257208669	Das Bse kommt auf leisen Sohlen

Hình 3.40 Trước và sau khi lọc kí tự đặc biệt

- Tách cột Location thành cột City và Country để tiện cho nhu cầu phân tích. Trước khi tách chúng ta xóa tất cả kí tự đặc biệt trừ dấu ‘,’ trong cột Location.

```
20 hours ago (<1s)                                                   9
from pyspark.sql.functions import col, regexp_replace
df_users = df_users.withColumn("Location", regexp_replace(col("Location"), "[^a-zA-Z\s,]", ""))
```

Hình 3.41 Xóa kí tự đặc biệt cột Location

```
20 hours ago (7s)                                                    10
from pyspark.sql.functions import col, split, reverse, expr
df_users = df_users.withColumn("Reversed_Location", expr("reverse(Location)")) \
                   .withColumn("Reversed_Split", split(col("Reversed_Location"), ",\s*")) \
                   .withColumn("Country", expr("reverse(Reversed_Split[0])")) \
                   .withColumn("City", expr("reverse(Reversed_Split[1])")) \
                   .drop("Reversed_Location", "Reversed_Split")
```

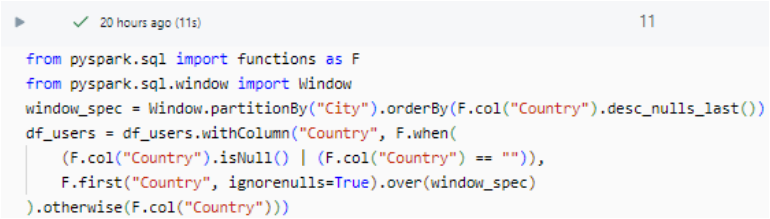
Hình 3.42 Tách cột

	User-ID	Location	Age
50	1305	edina, minnesota, usa	17
51	1314	monza, lombardia, italy	48
52	1318	le\u00f6n, castilla-le\u00f6n,	33

	User-ID	Location	Age	Country	City
50	1305	edina, minnesota, usa	17	usa	minnesota
51	1314	monza, lombardia, italy	48	italy	lombardia
52	1318	len, castillalen,	33		castillalen

Hình 3.43 Trước và sau xử lý tách cột

- Sau khi tách, có những ô không có giá trị thì ta sẽ điền giá trị ở cột Country dựa vào cột City thông qua những dòng mà có cả City, Country.

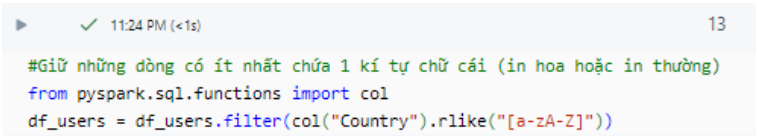


Hình 3.44 Xử lý cột Country dựa vào City

- Trong quá trình xử lý, dữ liệu cột Country có sai chính tả ví dụ như America với Americ thì ta sẽ xử lý bằng cách gọi API tới 1 trang web có chứa tên tất cả Country sau đó so sánh cột Country với API đó nếu đúng hơn 80% thì thay thế Country trong API.

Hình 3.45 Điền tên Country sao cho đúng chính tả

- Nếu cột Country chứa chuỗi rỗng hay null, thì xóa những dòng đó (vì cột Country khá quan trọng cho việc phân tích dữ liệu theo từng quốc gia hơn là cột City). Ta xử lý bằng cách lọc và chỉ giữ lại dòng ở cột Country mà tối thiểu phải có ít nhất 1 kí tự chữ trong chuỗi Country.



Hình 3.46 Xóa giá trị rỗng ở cột Country

- Xử lý giá trị null hoặc các giá trị không thỏa yêu cầu cho cột Age (từ 12 đến 80 tuổi) của bảng users và cột Year_of_publication (nhỏ hơn năm 2025) của bảng books.

- Nếu tổng bảng ghi null hoặc không hợp lệ lớn hơn 10% tổng số bản ghi thì thay giá trị không thỏa đó bằng giá trị trung bình.
- Nếu nhỏ hơn 10% thì thực hiện xóa các bảng ghi không thỏa đó.

```
11:24 PM (48s)                                                    15

from pyspark.sql import functions as F

def process_null_values(df, column_name, condition):
    total_count = df.count()
    null_count = df.filter(condition).count()
    null_percentage = null_count / total_count * 100

    if null_percentage > 10:
        # Thay thế giá trị null hoặc không hợp lệ bằng giá trị trung vị là số nguyên
        valid_df = df.filter(~condition)
        median_value = valid_df.approxQuantile(column_name, [0.5], 0.25)[0]  # Sử dụng median thay vì mean
        df = df.withColumn(column_name, F.when(condition, int(median_value)).otherwise(F.col(column_name)))
    else:
        # Loại bỏ các hàng chứa giá trị null hoặc không hợp lệ
        df = df.filter(~condition)
    return df

# Xử lý cột Age: null hoặc nhỏ hơn 12
age_condition = (F.col("Age").isNull()) | (F.col("Age") < 12) | (F.col("Age") > 80)
df_users = process_null_values(df_users, "Age", age_condition)

# Xử lý cột Year-Of-Publication: null hoặc bằng 0
year_condition = (F.col("Year-Of-Publication").isNull()) | (F.col("Year-Of-Publication") == 0) | (F.col("Year-Of-Publication") > 2024 )
df_books = process_null_values(df_books, "Year-Of-Publication", year_condition)
```

Hình 3.47 Xử lí những giá trị không hợp lệ

User-ID	Location	Age	Country	City
30590	cambridge, cambridgeshire, united kingdom	141	united kingdom	cambridgeshire

User-ID	Location	Age	Country	City
30590	cambridge, cambridgeshire, united kingdom	32	united kingdom	cambridgeshire

Hình 3.48 Trước và sau khi xử lý

- Kiểm tra tính ràng buộc giữa các bảng. Bảng rating tham chiếu đến bảng books thông qua ISBN và bảng users qua khóa ngoại User_ID. Đồng thời kiểm tra và xóa giá trị không hợp lệ của bảng ratings.

```
04:04 PM (26s)                                                    17                                                    Python

from pyspark.sql import functions as F

# Xóa các bản ghi không hợp lệ dựa trên ISBN từ df_ratings
df_ratings_valid_isbn = df_ratings.alias("ratings").join(df_books.alias("books"), F.col("ratings.ISBN") == F.col("books.ISBN"), how="inner")

# Xóa các bản ghi không hợp lệ dựa trên User-ID từ df_ratings
df_ratings_valid = df_ratings_valid_isbn.join(df_users.alias("users"), F.col("ratings.User-ID") == F.col("users.User-ID"), how="inner")

# Chọn các cột cần thiết từ df_ratings_cleaned và tránh lỗi "AMBIGUOUS_REFERENCE"
df_ratings_cleaned = df_ratings_valid.select(
    F.col("ratings.User-ID").alias("UserID"),
    F.col("ratings.ISBN"),
    F.col("ratings.Book-Rating")
)
```

Hình 3.49 Kiểm tra tính ràng buộc

- Thiết lập các thông tin cần thiết để kết nối Databricks với Azure Blob Storage. Và tạo đường dẫn lưu các file đã xử lý vào những thư mục đã tạo sẵn ở Azure Blob Storage tương ứng.

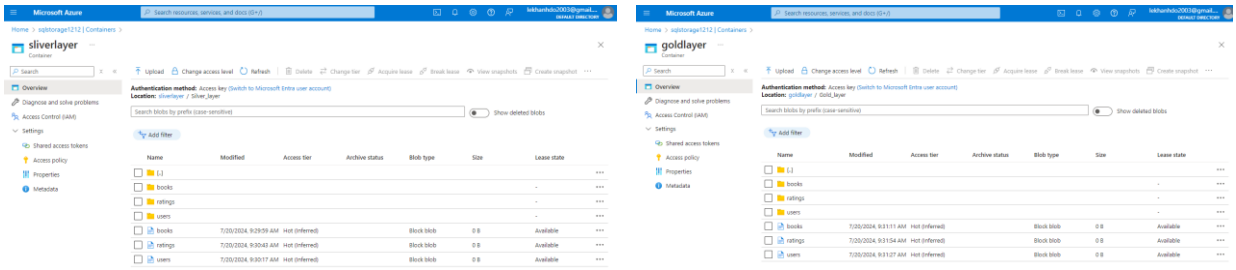
Hình 3.50 Lưu dữ liệu vào thư mục Silver Layer

- Thực hiện loại bỏ những cột hay những bảng ghi không cần thiết cho việc phân tích và lưu vào thư mục Gold Layer.

Hình 3.51 Lấy những cột dữ liệu cần thiết để phân tích

Hình 3.52 Lưu dữ liệu vào Gold Layer

- Sau khi thực hiện xử lý dữ liệu và lưu chúng vào thư mục tương ứng trong Azure Blob Storage, ta thu được kết quả như hình.



Hình 3.53 Thư mục sliverlayer và goldlayer sau xử lý

- Cấu hình và điền thông tin kết nối trực tiếp từ Power BI đến Azure Blob Storage như Storage Name , Access Key. Sau đó, chọn thư mục đã lưu trữ dữ liệu xử lý ở Gold Layer để thực hiện tạo biểu đồ hỗ trợ cho việc phân tích và báo cáo.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

- **Đạt được:**

- Đã được tìm hiểu các kiến thức về Data Engineer cũng như các công việc thực tế mà một Data Engineer làm và cách thực hiện các công việc đó trong thực tế.
- Nâng cao kỹ năng viết báo cáo, tham gia xử lý dữ liệu dưới sự hỗ trợ của mentor.
- Ngoài ra còn được học thêm một số công cụ dịch vụ hỗ trợ trong việc xử lý dữ liệu.
- Được làm trong một môi trường chuyên nghiệp và tiếp xúc với các anh chị đã có kinh nghiệm.
- Phát triển kĩ năng mềm như giao tiếp, khả năng thuyết trình, quản lí giờ giấc. Làm việc có trách nhiệm hơn.

- **Hạn chế:**

- Kiến thức chuyên môn còn nhiều hạn chế nền cần đầu tư thời gian để trau dồi.
- Khả năng tư duy viết code còn chưa được tối ưu.
- Vẫn còn yếu về khả năng đọc tài liệu bằng Tiếng Anh.

- **Hướng phát triển:**

- Nâng cao kiến thức và kinh nghiệm về xử lý dữ liệu bằng cách tìm hiểu và thực hành tiếp xúc nhiều với những bộ dữ liệu lớn và phức tạp được chia sẽ miễn phí trên các nền tảng.
- Tìm hiểu những yêu cầu cần cho công việc từ các bài đăng tuyển dụng. Từ đó tự trau dồi thêm kiến thức mới.
- Tự ứng tuyển học việc để được va chạm thực tế nhiều hơn.

TÀI LIỆU THAM KHẢO

1. https://www.tma-binhdinh.vn/kham-pha-tip
2. https://tuoitre.vn/data-engineer-la-gi-cong-viec-va-ky-nang-can-thiet-doi-voi-vi-tri-nay-20230720142335839.htm
3. https://innovativehub.com.vn/cloud-computing-la-gi-dinh-nghia-phan-loai-uu-va-nhuoc-diem/#:~:text=Th%C3%B4ng%20th%C6%B0%E1%BB%9Dng%2C%20m%E1%BB%99t%20h%E1%BB%87%20th%E1%BB%91ng,v%E1%BB%A5%20c%C6%A1%20s%E1%BB%9F%20h%E1%BA%A1%20t%E1%BA%A7ng.
4. https://onesme.vn/blog/san-pham/mo-hinh-dich-vu-dien-toan-dam-may.html
5. https://github.com/janampatel15/Ecommerce_Database/tree/main
6. https://www.kaggle.com/datasets/saurabhbagchi/books-dataset/code

CHECK LIST CỦA BÁO CÁO

STT	Nội dung công việc	Có	Không	Ghi chú
1	Báo cáo được trình bày (định dạng) đúng với yêu cầu.	x		
2	Báo cáo có số lượng trang đáp ứng đúng yêu cầu (30-50 trang)	x		
3	Báo cáo trình bày được phần mở đầu bao gồm: Mục tiêu, Phạm vi và đối tượng, kết cấu …	x		
4	Báo cáo trình bày về công ty, vị trí việc làm (công việc đó làm gì, kiến thức và kỹ năng cần thiết là gì, con đường phát triển sự nghiệp (career path)), cơ sở lý thuyết phù hợp với nội dung của đề tài (Tối đa 10-12 trang)	x		
5	Báo cáo có sản phẩm cụ thể phù hợp với mục tiêu đặt ra của đề tài	x		
6	Báo cáo có phần kết luận và hướng phát triển của đề tài	x		