# Xây dựng hệ thống

## Quy trình xây dựng

Quy trình xây dựng gồm 5 bước chính là tạo các Storage account để lưu dữ liệu, tạo key vault để quản lý các key, xử lý dữ liệu trên Databrick Azure, pipeline bằng Data Factory, cuối cùng tạo Sql server Azure để tải dữ liệu vào Database và liên kết với PowerBi trên DeskTop để trực quan hóa dữ liệu. Toàn bộ quy trình được thực hiện trên dịch vụ cloud của Microsoft Azure.

A screenshot of a computer

Description automatically generated

Hình 4.1 Tổng quát về resource group trên Azure

### Tạo stogare account

Tạo các Storage Account theo từng cấp độ tương ứng, trong mỗi storage account sẽ tạo những Containers để chứa dữ liệu với từ tệp dữ liệu. Tại Storage-Level01 sẽ tạo 2 ccontainers data-raw và data\_log cho mỗi file dữ liệu tương ứng (customers, sale).

Data raw: sẽ chứa các dữ liệu thô ban đầu. Dữ liệu tại đây là được dùng cho lần đầu đọc và xử lý dữ liệu đầu tiên của hệ thống

DataLog: là dữ liệu trong quá trình thu thập ghi lại nhật ký thay đổi của dữ liệu. Chẳng hạn như kích thước dữ liệu, sửa đổi, thêm hoặc có thể là cập nhật. Dữ liệu tại đây sẽ được đọc và xử lý cho các lần xử lý tiếp theo

A screenshot of a computer

Description automatically generated

Hình 4.2 Storage-Level01

Trong StogareLevel-2 tạo các containers tương ứng để lưu các lớp dữ liệu sau xử lý gồm lớp Bronze và lớp Silver

A screenshot of a computer

Description automatically generated

Hình 4.3 Stogare-Level-2

### Tạo Key Vault

Tạo key vault để lưu và mã hóa các key của từng Stograge. Đảm bảo tính bảo mật khi dùng để truy cập và load dữ liệu ở DataBricks (không sử dụng key trực tiếp trên code). Trong mỗi KeyVault sẽ chứa các key của các Storage Account và được mã hóa bên trong.

A screenshot of a computer

Description automatically generated

Hình 4.4 Tạo Key vault

### Xử lý bằng Databricks

#### Lớp Bronze- Processing

**Bước 1:** Tạo scope

Tạo Scope có đường dẫn của KeyVault đã tạo để tạo vùng truy cập đến chứa dữ liệu

A screenshot of a computer

Description automatically generated

Hình 4.5 Tạo scope trên databricks

**Bước 2**: Truy cập vào các StorageLevel\_01

Sử dụng scope đã tạo cùng keyvault để cấp quyền truy cập data trong Storage

A screenshot of a computer

Description automatically generated

Hình 4.6 Sử dụng Scope truy cập vào các Storage

**Bước 3:** Tạo tham số để nhập tên file dữ liệu từ tên file mới nhất được tải lên để kích hoạt trigger. File kích mới nhất kích hoạt trigger sẽ được sử dụng để xử lý chứ không phải các file cũ đã tải lên trước đó trong storage\_level01

A close-up of a website

Description automatically generated

Hình 4.7 Tạo tham số đầu vào cho file dữ liệu

Trong lần đầu đọc dữ liệu của hệ thống thì data sẽ được lấy từ container “Raw Data”

A computer code on a white background

Description automatically generated

Hình 4.8 Đọc dữ liệu lần đầu

Trong các lần đọc và xử lý dữ liệu tiếp theo sẽ đọc dữ liệu log từ container “Data-log”

A close-up of a computer code

Description automatically generated

Hình 4.9 Đọc dữ liệu sau lần đầu tiên

**Bước 4: Quy trình xử lý khi lưu dữ liệu vào Bronze**

Trong lần đầu xử lý thì Delta Table với đường dẫn để lưu lớp Bronze của dữ liệu sẽ trống và không ở dạng delta. Vì vậy nếu DelatTable với đường dẫn Bronze trả về Flash thì dữ liệu mới đầu tiên sẽ được tải vào lớp Bronze và sẽ kích hoạt Change Data Feed để có thể sử dụng các tính năng truy cập các phiên bản history .

Trong các lần xử lý tiếp theo, Delta Table của Bronze đã được kích hoạt và trả về True thì dữ liệu sẽ được lấy từ data-log và tiến hành merge dữ liệu mới vào bảng dữ liệu của bronze theo các điều kiện được tạo sẵn trong data-log. Nếu trong ‘Operation’ của nguồn dữ liệu mới là delete thì sẽ tiến hành xóa dòng đó. Khi update sẽ cập nhật lại. Còn các trường hợp còn lại như insert thì sẽ thêm các hàng mới vào trong dữ liệu cũ.

***Giải thích về cột Operation:*** Trong các file mới được update sẽ có thêm cột (Operation có thể là delete hoặc insert) để làm điều kiện merge gọi là file log, đây là file được thu thập trong quá trình hoạt động mà khách hàng thu được mà muốn cập nhật lên lại hệ thống.

**A screenshot of a computer code

Description automatically generated**

Hình 4.10 Lưu dữ liệu vào lớp Bronze

#### Silver- Processing

Bước 1: Tương tự ở Bronze\_Processing, cũng sẽ tạo Scope để có thể truy cập vào lớp dữ liệu được lưu ở lớp Bronze.

A screenshot of a computer

Description automatically generated

Hình 4.11 Tạo Scope

**Bước 2:** Truy cập và tải dữ liệu theo đường dẫn từ vùng truy cập scope và keyvault vừa tạo. Khi sử dụng keyvault sẽ đảm bảo được tính bảo mật vì không sử dụng key trực tiếp trên code, key đã được mã hóa trong keyvault



Hình 4.12 Sử dụng scope và keyvault truy cập vào Storage

**Bước 3:** Tạo hàm làm sạch dữ liệu

Tạo hàm gồm tất cả các quy trình làm sạch xử liệu thô bao gồm như xử lý dữ liệu null, giá trị trùng lặp, schema hay các lỗi có trong tập dữ liệu. Hàm làm sạch dữ liệu sẽ sử dụng để xử lý dữ liệu trước khi đưa dữ liệu vào trong lớp Silver

**Bước 4**: Xây dựng quy trình tải dữ liệu thô từ Bronze sau đó đưa qua hàm làm sạch và lưu vào lớp Silver

Ở lần đầu tải dữ liệu lên lớp đường dẫn lưu trữ của lớp silver vẫn chưa phải dạng Delta table, thì hệ thống sẽ đọc file dữ liệu thô từ lớp Bronze sau đó đưa qua hàm xử lý để làm sạch. Tiếp theo đó là lưu dữ liệu đã làm sạch dưới dạng Delta vào đường dẫn đã tạo và kích hoạt tính năng Change Data Feed.

A computer code with many colorful text

Description automatically generated with medium confidence

Hình 4.13 Lưu dữ liệu vào silver ở lần đầu

Trong những lần tiếp theo sẽ lấy dữ liệu mới nhất vừa được tải lên lớp bronze bằng tính năng history của change data feed mà t đã kích hoạt ở các lớp từ lần đầu xử lý. Sau đó dữ liệu sẽ đưa qua hàm xử lý để làm sạch và cuối cùng sẽ merge với bảng dữ liệu trong Silver.

A screen shot of a computer code

Description automatically generated

Hình 4.14 Xây dựng luồng dữ liệu trên Data Factory

### Tạo SQL Server và database trên azure

### Sử Data Factory xây dựng các đường ống xử lý dữ liệu theo quy trình

Xây dựng luồng dữ liệu trên data Factory: Dữ liệu sau khi kích hoạt trigger sẽ đưa qua 2 quy trình xử lý của databricks là Bronze\_Processing và Silver\_Processing. Sau đó dữ liệu từ Silver\_Processing sẽ được sao chép vào DataBase đã được liên kết. Từ DataBase chúng ta có thể đưa dữ liệu vào PowerBi để trực quan hóa dữ liệu.

A screenshot of a computer

Description automatically generated

Hình 4.15 Xây dựng luồng dữ liệu trên Data Factory

A screenshot of a computer

Description automatically generated

* Tạo các đường liên kết dịch vụ với Storage Account, DataBricks, Azure SQL trên Pipeline
* Tạo các tham số cần thiết để liên kết các file dữ liệu khi được tải lên để kích hoạt Trigger và sử dụng file đó để xử lý trên DataBricks
* Tạo Trigger tự động khi người dùng tải tệp dữ liệu mới lên sẽ kích hoạt hệ thống. File bắt đầu bằng Cust hoặc Sale tùy vào loại dữ liệu và kết thúc bằng .CVS. Sẽ có 2 trigger với cho raw-data và data-log
* Thêm Trigger đã tạo vào pipeline và Publish all cho hệ thống và hoàn thành

## Các bước trong hàm xử lý dữ liệu của DataBricks

### Bảng customer

**Bước 1:** Kiểm tra các giá trị null

* + Đếm các hàng có giá trị null của từng cột
  + Hiển thị các hàng có giá trị null
  + Xóa các hàng có giá trị null

**Bước 2:** Loại bỏ các ký tự lỗi ở các cột QueQuan, NamHocThu, ThoiGianSuDung

**Bước 3:** Lọc các hàng ở các cột đánh giá, có giá trị bé hơn 0 và lớn hơn 5

* Chỉnh các giá trị ngoại lai ở Cột MucDoHaiLong
  + Tính giá trị trung bình của các cột.....
  + Lấy phần nguyên của giá trị trung bình fill vào các hàng có giá trị không hợp lệ

**Bước 4:** Chuyển các giá trị ở cột GioiTinh về Nam và Nữ

* + Xem các giá trị duy nhất trong cột GioiTinh
  + Thay thế tất cả về Nam và Nữ

**Bước 5:** Chuyển các cột có giá trị số nguyên về kiểu dữ liệu integer

### Bảng Sale

**Bước 1**: Chuyển giá trị có ký tự $ thành tiền Việt

* + Lọc ra các hàng ở cột SalePrice có kí tự $
  + Xóa bỏ ký tự $ và nhân 23000 để chuyển thành tiền Việt

**Bước 2**: Thay thế các giá trị null

**Bước 3:** Xóa các hàng trùng lặp

**Bước 4:** Xử lý kiểu dữ liệu của cột sale Date

## Kiểm tra và chạy thử hoạt động của hệ thống sau khi hoàn thành gồm trigger, pipeline và databricks

### Lần chạy đầu tiên với rawdata

#### Hoạt động của trigger và pipeline

Sau khi đẩy dữ liệu thô vào container Raw-data tiến hành vào “Monitor” của “DataFactory” kiểm tra trigger cũng như hoạt động của pipeline có hoạt động hay không. Nếu trigger hoạt động thì kiểm tra pipeline chạy có lỗi hay không. Nếu không thì tiếp tục triển khai lần chạy tiếp theo với các tệp dữ liệu data-log

Trigger raw-customer chạy thành công

A screenshot of a computer

Description automatically generated

Hình 4.16 Hoạt động của trigger Raw

***Pipeline chạy thành công***

A screenshot of a computer

Description automatically generated

Hình 4.17 Pipeline chạy thành công

#### Kiểm tra kết quả trả về của data bricks

Xem chi tiết kết quả thông qua link Output trả về

A screenshot of a computer

Description automatically generated

Hình 4.18 Xem kết quả trả về ua DataBricks

Qua kết quả bên dưới có thể thấy được Parameters khớp với file dữ liệu raw-data đã tải lên để kích hoạt trigger chạy. Và số lượng dòng dữ liệu được ghi vào 2 lớp Bronze và Silver là 501 dòng

A screenshot of a computer

Description automatically generated

Hình 4.19 Kết quả Bronze\_Processing

A screenshot of a computer

Description automatically generated

Hình 4.20 Kết quả Silver\_Processing

### Lần chạy tiếp theo với các tệp data-log

Sau lần đầu tiên ta sẽ tải tệp các tệp dữ liệu data-log có thể gồm insert, delete, update. Trong Demo này sẽ tải một tệp data-log insert với số dòng là.

Trong lần này sẽ kiểm tra trigger-log có hoạt động hay không, và kết quả trả về của databrick có thay đổi đúng với số dòng insert của tệp data-log hay không

#### Hoạt động của trigger-log và pipeline

Trigger chạy thành công

A screenshot of a computer

Description automatically generated

Hình 4.21 Trigger-Log chạy thành công

A screenshot of a computer

Description automatically generatedPipeline hoạt động thành công

Hình 4.22 Pipeline do trigger-log kích hoạt chạy thành công

#### Kiểm tra kết quả trả về

Theo kết quả trả về của databricks Parameters là file insert-log và dữ liệu được tăng thêm lên 555 dòng so với ban đầu là 501 dòng ở cả 2 lớp Bronze và Silver

A screenshot of a computer

Description automatically generated

Hình 4.23 Xem kết quả ở Bronze\_Processing

A screenshot of a computer

Description automatically generated

Hình 4.24 Xem kết quả ở Silver\_Processing

### Cuối cùng kiểm tra dữ liệu sau khi được lưu vào cơ sở dữ liệu SQL server

Sau khi kiểm tra xong các chức năng của hệ thống sẽ kiểm tra dữ liệu cuối cùng được đưa vào trong Database để phân tích

A screenshot of a computer

Description automatically generated

Hình 4.25 Kiểm tra dữ liệu nạp vào Database

# TẠO DASHBOARD BẰNG POWER BI

## Tạo Dashboard bằng Power Bi

### Liên kết dữ liệu trên SQL Azure Database vào PowerBi Desktop

Lấy liên kết và thực hiện liên kết bằng DataBase trên server SQL Azure đã tạo. Sau đó chọn các bảng dữ liệu cần cần phân tích.

Đảm bảo tính bảo mật của sever tất cả các networking muốn kết nối truy cập vào server đều cần cung cấp địa chỉ IP.

A screenshot of a computer

Description automatically generated

Hình 5.1 Cung cấp địa chỉ IP của máy cho networking server

A screenshot of a computer

Description automatically generated

Hình 5.2 LIên kết dữ liệu DataBase với PowerBI

## Tạo Dashboard

Qua Dashboard trên sẽ có cái nhìn một cách trực quan hơn về bộ dữ liệu, tuy nhiên để phân tích sâu hơn về dữ liệu và đưa ra các kết luận chính xác thì cần phải xây dựng thêm các mô hình machine learning, các thuật toán phân cụm phân lớp. Các công việc đó sẽ do data analyst thực hiện

Với ngành Data Engineer công việc chính là xây dựng các đường ống dữ liệu, làm sạch và chuẩn bị nguồn dữ liệu cho data analyst thực hiện phân tích chuyên sâu.

Sau đây là 2 Dashboard về thông tin khách hàng và Dashboard về hành vi mua hàng của khách hàng. Chúng ta sẽ có được cái nhìn tổng quát về bộ dữ liệu để mọi người có thể hình dung được các thông tin một các trực quan và dễ hiểu nhất.

### Dashboard thông tin khách hàng

Ở Dashboard sẽ cho chúng ta thấy được một cách tổng quan nhất về các thông tin của khách hàng như chi tiêu hàng tháng, chuyên ngành, nghề nghiệp gia đình, quê quán, năm học của sinh viên.

A screenshot of a graph

Description automatically generated

Hình 5.3 Dashboard thông tin khách hàng

### Dashboard về hành vi khách hàng

Ta sẽ nhìn thấy được những thương hiệu sản phẩm, cửa hàng, kênh mua sắm cũng như phương thức thanh toán mà khách hàng lựa chọn nhiều nhất.

* Về thương hiệu Asus, Dell là 2 thương hiệu mà khách hàng chọn nhiều nhất, còn Microsof khá ít sinh viên lựa chọn
* Về cửa hàng: FPT shop và thế giới di động và 2 cửa hàng chiếm thị phần cao nhất
* Phương thức thanh toán: vì giá trị Laptop thường có giá trị cao nên đa phần mọi người sẽ lựa chọn thanh toán qua thẻ để đảm bảo an toàn và tiện lợi

A screenshot of a graph

Description automatically generated

Hình 5.4 Dashboard về hành vi khách hàng

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Đạt được

Trong suốt thời gian thực tập tại TMA Solution với vị trí Data Engineer, em đã có thêm rất nhiều kiến thức quý báu về nhiều khía cạnh của việc này. Những kỹ năng về lập trình, tư duy và phát triển hệ thống, xử lý và quản lý các đường ống dữ liệu dữ liệu.Trên những kiến thức được học em đã hiểu rõ hơn về những công việc và nhiệm vụ của một Data Engineer cần làm, và em đã áp dụng được những kỹ năng và kiến thức học được vào những nhiệm vụ mà mentor giao.

Ngoài ra không chỉ tiếp xúc với những kiến thức chuyên môn liên quan đến dữ liệu, mà em còn có cơ hội được phát triển những kỹ năng mềm cần thiết trong quá trình làm việc như thuyết trình hay trình bày một thông tin nào đó đến người nghe. Những kiến thức về bảo mật trong lĩnh vực công nghệ thông tin. Trong quá trình làm việc khi được tiếp xúc với nhiều anh chị em cũng đã cải thiện rất nhiều về kỹ năng giao tiếp trong công ty.

Tóm lại trong quá trình thực tập vừa qua em đã được học hỏi rất nhiều kiến thức về công việc và những kỹ năng cần thiết cho công việc từ các anh chị trong công ty. Em tin với những gì mình học được sẽ giúp ích cho sự nghiệp của bản thân rất nhiều trong thời gian sắp tới.

## Hạn chế

Ngoài những điều đạt được thì thời gian thực tập tại công ty là một hạn chế, thời gian hạn ngắn trong vòng 10 tuần để phải vừa hoàn thành các nhiệm vụ trên công ty cũng như hoàn thành bài báo cáo cho trường, em đã nỗ lực hết mình trong việc học hỏi tìm hiểu đề tài. Tuy nhiên vì thời gian ngắn nên em cũng chưa tận dụng hết kiến thức và kỹ năng học được vào các công việc thực tế nhiều. Việc làm quen với ngôn ngữ lập trình mới, các môi trường làm việc mới như Cloud Azure cần một thời gian đủ dài để tìm hiểu sử dụng được tối đa các chức năng cũng như các cách hoạt động của chúng thì mới tạo ra được những giá trị thực sự hữu ích cho các dự án thực tế. Và bài cáo cũng sẽ không tránh được những sai sót nhất định

## Hướng phát triển:

Với sự tích lũy trong quá trình học tập và thực tập tại doanh nghiệp, em đã có được cho mình hướng đi rõ hơn về công việc sắp tới của mình. Sau thời gian thực tập em sẽ tiếp tục trau dồi và rèn luyện tốt hơn những kỹ năng đã học được để củng cố kiến thức. Đồng thời em sẽ cố gắng tìm một công việc với mức độ học hỏi hơn về lĩnh data để có thể có thêm cho bản thân cơ hội được tiếp xúc và làm việc tại các doanh nghiệp

Ngoài định hướng về Data Engineer em cũng sẽ cố gắng tìm hiểu những công việc có liên quan như data analyst, data data science..... Không dừng ở mức độ học hỏi em rất muốn được áp dụng những kiến thức đã học vào các công việc thực tế để có thể rút ra những kinh nghiệm cần thiết một cách tốt nhất trước khi ra khỏi trường. Em tin với việc cố gắng không ngừng mở rộng kiến thức cũng như kinh nghiệm của bản thân sẽ giúp em đạt được mức độ thành thạo về lập trình cũng như xử lý và phân tích dữ liệu

# TÀI LIỆU THAM KHẢO

1. <https://spark.apache.org/>

2. [Microsoft Azure](https://vinsep.com/kien-thuc/microsoft-kien-thuc/azure/tim-hieu-ve-microsoft-azure-cloud-computing/)

3. [Data warehouse](https://viblo.asia/p/kien-thuc-co-ban-ve-kien-truc-dwh-data-warehouse-architecture-zXRJ8n8NVGq)

# PHỤ LỤC

**1. Code Bronze\_Processing**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**2. Code Silver\_Processing**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated