TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

======\*\*\*======

BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

**XÂY DỰNG DATA LAKEHOUSE CHO NỀN TẢNG TƯƠNG TÁC KHÁCH HÀNG CHATWOOT**

|  |  |
| --- | --- |
| CBHD: | Ts. Đặng Trọng Hợp |
| Sinh viên: | Trần Minh Phương |
| Mã sinh viên: | 2021602643 |

Hà Nội, Năm 2024

LỜI CẢM ƠN

Trong thời gian làm đồ án tốt nghiệp, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè đã giúp em hoàn thiện đồ án tốt nghiệp.

Trước tiên em xin gửi tới các thầy cô khoa Công nghệ thông tin trường Đại học Công Nghiệp Hà Nội lời chúc sức khỏe và lời cảm ơn chân thành. Với sự quan tâm, chỉ bảo của thầy cô đã tạo điều kiện cho em trong suốt quá trình học tập và hoàn thành đồ án tốt nghiệp lần này.

Đặc biệt em xin gửi lời cảm ơn chân thành nhất đến thầy Đặng Trọng Hợp đã trực tiếp tận tình hướng dẫn cũng như nhận xét và giúp đỡ em trong suốt quá trình hoàn thiện đồ án.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, đồ án này không tránh khỏi những thiếu sót. Em rất mong nhận được sự giúp đỡ của thầy cô và các bạn để hoàn thiện hơn đồ án của mình.

Em xin chân thành cảm ơn!

|  |  |
| --- | --- |
|  | Hà Nội, ngày tháng năm 2024  Sinh viên thực hiện |

MỤC LỤC

[LỜI CẢM ƠN ii](#_Toc183785563)

[MỤC LỤC iii](#_Toc183785564)

[DANH MỤC HÌNH ẢNH v](#_Toc183785565)

[DANH MỤC BẢNG BIỂU v](#_Toc183785566)

[DANH MỤC CÁC CHỮ VIẾT TẮT v](#_Toc183785567)

[MỞ ĐẦU vi](#_Toc183785568)

[1. Đặt vấn đề vi](#_Toc183785569)

[2. Lý do chọn đề tài vi](#_Toc183785570)

[CHƯƠNG 1 TỔNG QUAN 1](#_Toc183785571)

[1.1. Mục tiêu của đồ án 1](#_Toc183785572)

[1.2. Giới hạn và phạm vi của đồ án 1](#_Toc183785573)

[1.3. Kết quả dự kiến đạt được 1](#_Toc183785574)

[CHƯƠNG 2 Cơ sở lý thuyết 2](#_Toc183785575)

[2.1. Tổng quan về Chatwoot 2](#_Toc183785576)

[2.2. Tổng quan về Data Lakehouse 2](#_Toc183785577)

[2.3. Điểm khác biệt giữa Data Lakehouse và Database 3](#_Toc183785578)

[2.4. Kiến trúc Data Lakehouse 5](#_Toc183785579)

[2.5. Tổng quan về kiến trúc Medallion 6](#_Toc183785580)

[2.6. Mô hình logic của Data Warehouse 6](#_Toc183785581)

[2.7. Quy trình ETL/ELT dữ liệu trong hệ thống Data Lakehouse 10](#_Toc183785582)

[2.8. Cơ chế lưu dữ liệu trong Data Lakehouse 11](#_Toc183785583)

[2.9. Giới thiệu các công nghệ sử dụng trong đồ án 11](#_Toc183785584)

[2.9.1. Python 11](#_Toc183785585)

[2.9.2. Airflow 11](#_Toc183785586)

[2.9.3. Trino 12](#_Toc183785587)

[2.9.4. Dbt 12](#_Toc183785588)

[2.9.5. Iceberg 12](#_Toc183785589)

[2.9.6. Hive Metastore 12](#_Toc183785590)

[2.9.7. MinIO 12](#_Toc183785591)

[2.9.8. Clickhouse 12](#_Toc183785592)

[2.9.9. Supserset 12](#_Toc183785593)

[CHƯƠNG 3 Thiết kế và xây dựng mô hình hệ thống Data Lakehouse 13](#_Toc183785594)

[3.1. Yêu cầu đề tài 13](#_Toc183785595)

[3.2. Thiết kế mô hình Data Lakehouse 13](#_Toc183785596)

[3.2.1. Luồng dữ liệu vào Bronze 14](#_Toc183785597)

[3.2.2. Luồng dữ liệu vào Silver 15](#_Toc183785598)

[3.2.3. Luồng dữ liệu vào Gold 16](#_Toc183785599)

[3.3. Xây dựng cơ sở dữ liệu cho Data Lakehouse 17](#_Toc183785600)

[3.3.1. Tầng Bronze 17](#_Toc183785601)

[3.3.2. Tầng Silver 17](#_Toc183785602)

[CHƯƠNG 4 Triển Khai mô hình Data Lakehouse 28](#_Toc183785603)

[4.1. Ứng dụng ETL bộ dữ liệu chăm sóc khách hàng lên Data Lakehouse 28](#_Toc183785604)

[4.2. Khai thác báo cáo dữ liệu 28](#_Toc183785605)

[4.3. Kết luận chương 28](#_Toc183785606)

[CHƯƠNG 5 Kết Luận 29](#_Toc183785607)

[TÀI LIỆU THAM KHẢO 30](#_Toc183785608)

DANH MỤC HÌNH ẢNH

[Hình 1. 1 3](#_Toc167216057)

[Hình 2. 1 8](#_Toc167216040)

[Hình 2. 2 9](#_Toc167216041)

[Hình 2. 3 9](#_Toc167216042)

Hình 4‑1

DANH MỤC BẢNG BIỂU

DANH MỤC CÁC CHỮ VIẾT TẮT

MỞ ĐẦU

## Đặt vấn đề

Trong thời đại số hóa như hiện nay, việc quản lý dữ liệu là một trong những yếu tố quan trọng nhất đối với các công ty, đặc biệt là trong lĩnh vực Thương mại điện tử. Các hệ thống xử lí thông tin và dữ liệu nên được tích hợp và quản lý một cách chặt chẽ để đảm bảo hiệu quả và độ chính xác của các thông tin báo cáo. Điều này càng trở nên quan trọng hơn đối với các hệ thống báo cáo tự động, vì hệ thống báo cáo này phải được tạo ra đúng thời điểm và đảm bảo tính chính xác về mặt dữ liệu.

Tại một công ty trong lĩnh vực thương mại điện tử, việc thu thập và phân tích dữ liệu là một hoạt động cực kỳ quan trọng. Tuy nhiên, các công ty đang phải đối mặt với một lượng lớn dữ liệu được thu thập từ nhiều nguồn khác nhau dẫn đến việc truy vấn dữ liệu mất nhiều thời gian. Điều này làm cho việc quản lý, truy xuất và thực hiện phân tích báo cáo trở nên khó khăn. Hơn nữa, dữ liệu từ các nguồn này thường không được chuẩn hóa, có định dạng khác nhau và chứa các giá trị bị thiếu hoặc không chính xác. Điều này làm cho việc phân tích dựa trên dữ liệu trở nên mơ hồ và không tin cậy, gây rủi ro cho việc đưa ra quyết định kinh doanh.

## Lý do chọn đề tài

Để đảm bảo được việc thu thập, tích hợp dữ liệu đạt được hiệu quả cao nhất, Data Lakehouse là một trong những yếu tố không thể thiếu. Data Lakehouse giúp tổng hợp, tích hợp và phân tích các nguồn dữ liệu khác nhau trên cùng một nền tảng, đồng thời cung cấp các báo cáo định kì và tự động cho người dùng. Để thực hiện mục tiêu đó, đề tài "Xây dựng và triển khai mô hình Data Lakehouse cho công ty tài chính" được đưa ra để tối ưu hoá quy trình thu thập, xử lý và phân tích dữ liệu, từ đó công ty có thể đưa ra được các quyết định một cách chính xác và nhanh chóng hơn. Cụ thể, những vấn đề cần giải giải quyết bao gồm:

* Xác định nguồn dữ liệu.
* Xây dựng mô hình dữ liệu cho Data Lakehouse.
* Xây dựng luồng dữ liệu phân chia thành các khu vực rõ ràng
* Xây dựng cơ sở dữ liệu chuyên dụng cho báo cáo phân tích
* Lấy ra báo cáo phân tích số liệu.

Để thực hiện được các công việc được liệt kê phía trên, em xin đề xuất mô hình Data Lakehouse xử lý dữ liệu được thể hiện dưới hình bên dưới như sau:

A diagram of a stage

Description automatically generated

**Hình 1.1. Mô hình Data Lakehouse đề xuất**

Và để có thể thử nghiệm triển khai được mô hình này, đầu vào là bộ dữ liệu thử nghiệm được trích xuất từ các hệ thống tác nghiệp tại một công ty trong lĩnh vực thương mại điện tử, cụ thể là bộ dữ liệu về nền tảng chăm sóc khách hàng. Để xây dựng một luồng dữ liều như hình 1.1 bắt đầu từ việc xử lý dữ liệu nguồn, xây dựng một mô hình ETL dữ liệu và cuối cùng là trích xuất dữ liệu để đưa ra các báo cáo về các chỉ số hiệu quả chăm sóc khách hàng.

# TỔNG QUAN

*Bằng việc sử dụng những kiến thức đã tiếp thu được trong quá trình tìm hiểu và học hỏi trên ghế nhà trường cùng với thời gian nghiên cứu tài liệu tham khảo cho đến thời điểm hiện tại, em đã tổng hợp được kiến thức nền tảng quan trọng được em trình bày trong chương này sử dụng để làm cơ sở xây dựng đồ án này.*

## Mục tiêu của đồ án

* Tìm hiểu kiến trúc của hệ thống Data Lakehouse.
* Tìm hiểu các công nghệ đã được quy định trong kiến trúc Data Lakehouse.
* Triển khai các thành phần kiến trúc hệ thống.
* Xây dựng data model cho đồ án.
* Triển khai luồng dữ liệu phục vụ cho việc ETL.
* Tất cả các báo cáo được lấy từ 1 nguồn duy nhất
* Xây dựng báo cáo phân tích hiệu quả kinh doanh, hành vi người dùng, hoạt động của nhân viên định kỳ.

## Giới hạn và phạm vi của đồ án

Trong phạm vi đề tài này em sẽ nghiên cứu các vấn đề:

* Nghiên cứu, khảo sát, thu thập dữ liệu về quy trình trong chăm sóc khách hàng của công ty, đi sâu vào nghiên cứu và phân tích một hệ thống thông tin để xây dựng Data Lakehouse.
* Hệ thống được xây dựng bằng sử dụng các công nghệ xử lý dữ liệu, lưu trữ dữ liệu và báo cáo.
* Kiểm thử và phát triển hệ thống.

## Kết quả dự kiến đạt được

Hệ thống Data Lakehouse khi hoàn thành dự kiến đạt được các kết quả sau:

* Làm rõ được mục tiêu cần đạt được
* Triển khai hạ tầng hệ thống đạt được sự ổn định
* Xây dựng data model cho dự án theo Ralph Kimball (star schema, snowflake schema) để phân tích hiệu quả nhất theo các chiều dữ liệu.
* Xem xét, nghiên cứu các data strategy phù hợp cho data model và ETL pipeline
* Tính toán và xem xét khả năng lưu trữ, backup, backfill dữ liệu.
* Xây dựng báo cáo để có thể nhìn được hiệu quả kinh doanh, hành vi người dùng từ đó đưa ra chiến lược kinh doanh phù hợp.
* Hoàn thiện chi tiết báo cáo đồ án tốt nghiệp.

# Cơ sở lý thuyết

## Giới thiệu về Chatwoot

Chatwoot là một nền tảng quản lý hội thoại khách hàng đa kênh, giúp doanh nghiệp hợp nhất tất cả các kênh giao tiếp như email, live chat, mạng xã hội (Facebook, Twitter, Instagram), WhatsApp, Telegram và nhiều ứng dụng nhắn tin khác vào một hệ thống duy nhất. Điều này giúp doanh nghiệp dễ dàng theo dõi, quản lý và tương tác với khách hàng từ mọi kênh trong một giao diện tập trung, thay vì phải chuyển đổi qua lại giữa nhiều nền tảng khác nhau.

Một trong những công dụng nổi bật của Chatwoot là khả năng tự động hóa quy trình quản lý hội thoại. Nền tảng này cho phép tự động phân loại hội thoại dựa trên trạng thái (Đang xử lý, Đã giải quyết, Đã đóng) và gán chúng cho các nhóm hoặc cá nhân dựa trên các quy tắc định sẵn. Điều này giúp tối ưu hóa quy trình làm việc, giảm thời gian phản hồi và nâng cao hiệu suất hỗ trợ khách hàng.

Bên cạnh đó, Chatwoot còn hỗ trợ tính năng cộng tác nội bộ, cho phép các thành viên trong nhóm dễ dàng phối hợp xử lý các vấn đề phức tạp. Nhân viên có thể ghi chú nội bộ, chia sẻ thông tin hoặc tag đồng nghiệp để thảo luận trực tiếp trong từng cuộc hội thoại, giúp tăng cường sự chính xác và hiệu quả khi giải quyết các yêu cầu của khách hàng.

Tính năng bảo mật của Chatwoot cũng rất đáng chú ý, với việc cung cấp quyền truy cập theo vai trò và hỗ trợ xác thực hai yếu tố (2FA), giúp bảo vệ dữ liệu khách hàng một cách an toàn. Điều này đặc biệt quan trọng đối với các doanh nghiệp yêu cầu cao về bảo mật và quyền riêng tư.

Tóm lại, Chatwoot là một công cụ mạnh mẽ giúp doanh nghiệp cải thiện trải nghiệm khách hàng, tối ưu hóa quy trình hỗ trợ và tăng cường hiệu suất làm việc thông qua việc quản lý hội thoại khách hàng từ nhiều kênh trong một nền tảng duy nhất.

## Tổng quan về Data Lakehouse

**Data Lakehouse:** là một tập hợp các dữ liệu được tổ chức và ổn định, thường được cập nhật thường xuyên để phục vụ cho mục đích phân tích và tạo báo cáo, giúp hỗ trợ trong quá trình ra quyết định định hướng cho doanh nghiệp. Các đặc trưng của Data Lakehouse như sau:

* **Theo chủ đề:** Data Lakehouse theo chủ đề được thiết kế để phục vụ nhu cầu phân tích và báo cáo trong một lĩnh vực cụ thể. Nó chứa các tập hợp dữ liệu được liên kết và liên quan đến chủ đề chính, đáp ứng nhu cầu thông tin của một nhóm người dùng. Mục tiêu của Data Lakehouse theo chủ đề là cung cấp một nguồn thông tin tập trung và tối ưu hóa cho việc phân tích dữ liệu và tạo báo cáo trong một lĩnh vực cụ thể. Bằng cách tập trung vào một chủ đề riêng biệt, Data Lakehouse theo chủ đề giúp cải thiện hiệu suất truy xuất và xử lý dữ liệu.
* **Dữ liệu tích hợp:** dữ liệu đầu vào Data Lakehouse được tập hợp từ nhiều nguồn khác nhau có các cơ chế lưu trữ khác nhau: cơ sở dữ liệu quan hệ, excel file, CSV file, flat file, v.v… điều này sẽ dẫn đến việc quá trình tập hợp phải thực hiện việc làm sạch, sắp xếp, rút gọn dữ liệu nhằm đảm bảo tính nhất quán dữ liệu.
* **Dữ liệu cố định:** khi một bản ghi hoàn chỉnh, dữ liệu không thể tạo thêm hay sửa. Dữ liệu được sẽ được lưu trữ trong một thời gian dài. Sau quá trình trích xuất, chuyển đổi làm sạch và nạp vào Data Lakehouse thì thao tác cập nhật hay xóa dữ liệu sẽ không xảy ra. Dữ liệu trong Data Lakehouse chỉ có thao tác là thêm mới và truy xuất, không thể sửa hoặc xóa đi.
* **Tính riêng biệt:** các dữ liệu truy suất không bị ảnh hưởng bởi các dữ liệu khác hoặc tác động lên nhau. Thời gian lưu trữ dài hơn so với hệ thống tác nghiệp. Nếu như đối với hệ thống tác nghiệp chỉ lưu giá trị hiện tại nhưng với dữ liệu trong Data Lakehouse cung cấp thông tin lịch sử lâu dài hơn. Biến thời gian cũng là một thuộc tính khóa để đảm bảo tính duy nhất của dữ liệu.

## Điểm khác biệt giữa Data Lakehouse và Database

**Data Lakehouse:** là nơi lưu trữ dữ liệu của một tổ chức, doanh nghiệp. Data Lakehouse chỉ có nhân viên trong tổ chức, doanh nghiệp được phép sử dụng để phục vụ cho việc phân tích dữ liệu và báo cáo. Dữ liệu đầu vào của Data Lakehouse có thể là một hoặc nhiều cơ sở dữ liệu quan hệ, các file excel, CSV, text, … lưu trữ thông tin dữ liệu cần phân tích. Tất cả được tổng hợp lại làm đầu vào cho Data Lakehouse tại tầng dưới cùng và sau một quá trình ETL phức tạp, các dữ liệu đó sẽ được chuyển đổi, làm sạch để có thể dễ dàng trích xuất lên báo cáo phân tích.

**Database:** là nơi tập hợp các dữ liệu có cấu trúc, có tổ chức và mối liên quan với nhau, được dùng để ghi và truy vấn dữ liệu. Cơ sở dữ liệu thường được lưu trữ và truy cập từ một hệ thống máy tính, được nhiều người sử dụng, tương tác với cơ sở dữ liệu và tổ chức theo mô hình. Cơ sở dữ liệu được sử dụng bằng hình thức xử lý trực tuyến, có sự chuẩn hóa (đối với mô hình cơ sở dữ liệu quan hệ) để giảm thiểu dữ liệu dư thừa, tối ưu hóa dung lượng lưu trữ.

Bảng 2.1. Điểm khác biệt giữa Data Lakehouse và Database

|  |  |  |
| --- | --- | --- |
|  | **Database** | **Data Lakehouse** |
| **Mục đích** | Giúp người dùng thao tác và truy vấn dữ liệu | Giúp người dùng xử lý, tích hợp và phân tích dữ liệu |
| **Chức năng** | Hỗ trợ các hoạt động hàng ngày cho các người dùng tương tác với cơ sở dữ liệu | Hỗ trợ những quyết định chiến lược cho doanh nghiệp, được nhân viên, lãnh đạo của doanh nghiệp sử dụng cho phân tích và báo cáo |
| **Mô hình sử dụng** | Mô hình quan hệ - thực thể | Mô hình dữ liệu đa chiều |
| **Phương pháp xử lý** | Giao dịch trực tuyến | Phân tích trực tuyến |
| **Các bảng và liên kết** | Có độ phức tạp cao vì chúng được chuẩn hóa (cho RDMS) theo các chuẩn như 1NF, 2NF, 3NF để giảm dữ liệu thừa, tối ưu hóa dung lượng lưu trữ | Các bảng được ghép nối rất dễ dàng trong Data Lakehouse để phục vụ cho các câu truy vấn |
| **Tính chất dữ liệu** | Chi tiết, được cập nhật thường xuyên | Có tính lưu trữ lịch sử và thống kê. Dữ liệu chỉ có thể được thêm mới chứ không thể xóa, cập nhật |
| **Lưu trữ dữ liệu** | Phương pháp tiếp cận theo quan hệ phẳng, nhiều dữ liệu khác nhau được tích hợp vào một nguồn | Phương pháp tiếp cận đa chiều và chuẩn hóa, được tích hợp từ nhiều nguồn khác nhau và định dạng lại |
| **Sử dụng** | Thường xuyên | Trong những trường hợp cụ thể khi cần phân tích |
| **Định hướng** | Định hướng ứng dụng | Định hướng chủ đề |
| **Truy vấn** | Truy vấn đơn giản | Truy vấn phức tạp |
| **Hiệu suất truy vấn** | Thấp | Cao |

Tóm lại, cơ sở dữ liệu thường là một ứng dụng, chương trình hoặc hệ thống để chứa các thông tin trong một nguồn. Data Lakehouse là tập hơn các nguồn, hệ thống thông tin khác nhau để sắp xếp, phân tích và xuất báo cáo theo truy vấn người dùng. Điều quan trọng cần lưu ý là Data Lakehouse có thể được lấy từ một hoặc nhiều cơ sở dữ liệu khác nhau hoặc nhiều nguồn dữ liệu khác nhau.

## Kiến trúc Data Lakehouse

Data Lakehouse là một mô hình kết hợp giữa data lake và data warehouse, nhằm tận dụng ưu điểm của cả hai loại hệ thống này. Nhằm tối ưu chi phí lưu trữ, cũng như xử lý dữ liệu so với kiến trúc Data Warehouse truyền thống. Nó bao gồm 4 thành phần cơ bản sau:

* Dữ liệu nguồn: dữ liệu từ các hệ thống khác nhau (bao gồm dữ liệu có cấu trúc và phi cấu trúc) như các CSDL, file excel, CSV, …
* Data Lakehouse: là nơi lữu trữ cả dữ liệu thô và dữ liệu đã qua xử lý.
* Presentation stage (vùng trình bày): là nơi lưu trữ dữ liệu cuối cùng đã được tổng hợp hay còn gọi là các cube.
* Người dùng cuối: là người dùng khai thác thông tin từ kho dữ liệu để phân tích, làm báo cáo

A diagram of a data flow

Description automatically generated

*Hình ảnh mô tả kiến trúc tổng thể của Data Lakehouse*

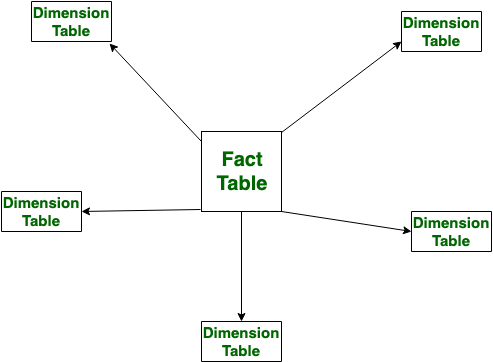
## Tổng quan về kiến trúc Medallion

Kiến trúc medallion: là một cách tiếp cận của kiến trúc Data Lakehouse, phân tầng dữ liệu thành các cấp độ khác nhau nhằm tối ưu hóa quy trình xử lý và tổ chức dữ liệu từ dạng thô đến dữ liệu sẵn sàng cho phân tích và báo cáo. Mỗi cấp độ dữ liệu có một vai trò cụ thể, giúp đảm bảo chất lượng dữ liệu, hiệu suất xử lý, và tính linh hoạt trong việc khai thác dữ liệu.Có 3 tầng chính trong kiến trúc: Bronze (Đồng), Silver (Bạc), và Gold (Vàng).

* Tầng Bronze: Chứa dữ liệu thô (chưa qua xử lý), không hoặc rất ít chuẩn hóa.
* Tầng Silver: chứa dữ liệu đã làm sạch (đã qua xử lý), những bảng chiều (dim table), bảng sự kiện giao dịch (transaction fact table), bảng tổng hợp sự kiện (fact aggreate table).
* Tầng Gold: chứa dữ liệu đã qua tổng hợp, sẵn sàng cho việc phân tích, báo cáo,... gồm những khối dữ liệu (cube)

## Mô hình logic của Data Warehouse

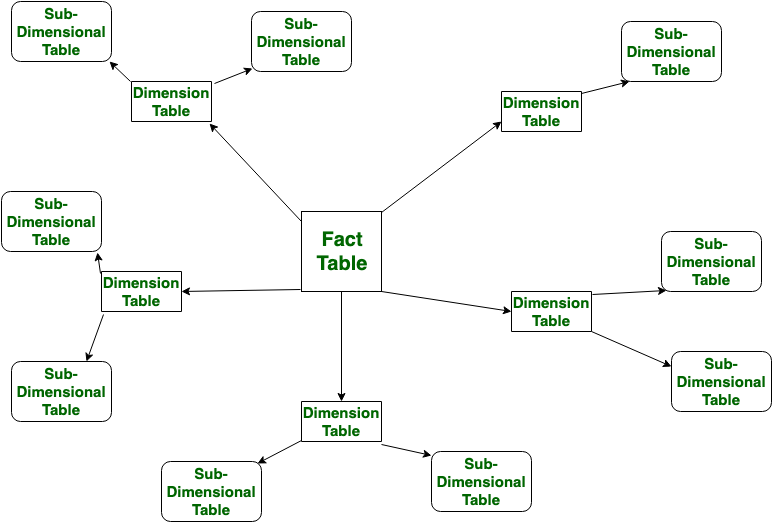
**Mô hình ngôi sao (Star schema):** một bảng Fact (sự kiện) ở trung tâm được kết nối với một tập các bảng Dimension (chiều) khác nhau



*Hình ảnh mô hình ngôi sao (star schema)*

* **Ưu điểm:**
* **Dễ hiểu và dễ xây dựng:** Đơn giản, các bảng sự kiện và bảng chi tiết có cấu trúc rõ ràng, dễ nắm bắt.
* **Hiệu quả cho truy vấn:** Phù hợp với các truy vấn phân tích tốc độ cao.
* **Tối ưu cho công cụ BI:** Hỗ trợ các công cụ BI và trực quan hóa dữ liệu rất tốt do cấu trúc đơn giản.
* **Nhược điểm:**
* **Không tiết kiệm không gian lưu trữ:** Dữ liệu trong các bảng chi tiết có thể bị trùng lặp.
* **Không phù hợp cho dữ liệu phức tạp:** Không hiệu quả khi cần quản lý các mối quan hệ phức tạp giữa dữ liệu.

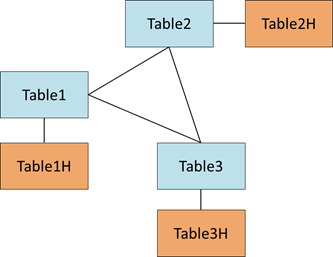
**Mô hình bông tuyết (Snowflake Schema):** Là một biến thể của mô hình ngôi sao, được chuẩn hóa để loại bỏ sự trùng lặp dữ liệu. Các bảng fact có thể liên kết với các bảng phụ (sub-dimension tables), giúp giảm kích thước lưu trữ và cải thiện quản lý dữ liệu.



*Hình ảnh mô hình Snowflake*

* **Ưu điểm:**
* **Tiết kiệm không gian lưu trữ:** Dữ liệu chuẩn hóa giúp giảm trùng lặp trong các bảng chi tiết.
* **Quản lý mối quan hệ phức tạp:** Hỗ trợ cấu trúc dữ liệu phức tạp hơn mô hình ngôi sao.
* **Nhược điểm:**
* **Truy vấn phức tạp hơn:** Các truy vấn có thể phức tạp do cần nhiều phép nối (join) hơn giữa các bảng.
* **Khó xây dựng và quản lý:** Cần thiết kế và bảo trì chi tiết hơn, dễ gây nhầm lẫn trong quản lý dữ liệu.

**Mô hình 3 chiều (3NF Schema):** Dữ liệu được chuẩn hóa theo chuẩn hóa bậc ba (3NF - Third Normal Form)



*Hình ảnh mô hình 3 chiều*

* **Ưu điểm:**
* **Đảm bảo toàn vẹn dữ liệu:** Chuẩn hóa bậc ba giúp giảm trùng lặp và đảm bảo dữ liệu nhất quán.
* **Phù hợp cho dữ liệu giao dịch:** Lý tưởng cho dữ liệu giao dịch cần độ chính xác cao.
* **Nhược điểm:**
* **Không tối ưu cho phân tích:** Do cần nhiều phép nối, truy vấn phức tạp, nên không phù hợp với các yêu cầu phân tích và báo cáo tốc độ cao.
* **Khó sử dụng với công cụ BI:** Nhiều công cụ BI hoạt động kém với mô hình dữ liệu phức tạp này.

## Quy trình ETL/ELT dữ liệu trong hệ thống Data Lakehouse

A diagram of a diagram

Description automatically generated

*Hình ảnh chi tiết kiến trúc hệ thống Data Lakehouse*

**Thu thập nguồn dữ liệu:** Thu thập dữ liệu từ các nguồn khác nhau, bao gồm cả dữ liệu cấu trúc và phi cấu trúc. Sử dụng Python, Trino để thực hiện EL (Extract, Load) cho quá trình này.

**Lưu trữ dữ liệu:** Lưu trữ dữ liệu thô, dữ liệu sẽ được lưu vào Minio, Iceberg có vai là tầng Bronze của hệ thống Lakehouse.

**Xây dựng Data Warehouse:** Thiết kế các mô hình dữ liệu cho hệ thống Data Warehouse, bao gồm cả các bảng kích thước (dimension tables) và các bảng sự kiện (fact tables). Sử dụng mô hình dữ liệu Star Schema của Ralph Kimball để đảm bảo tính nhất quán và dễ dàng mở rộng. Sử dụng Python, Trino để transform dữ liệu, sau đó lưu kết quả (các bảng kích thước và bảng sự kiện) vào Minio, Iceberg có vai là tầng Silver của hệ thống Lakehouse.

**Xây dựng Presentation Stage:** Thiết kế các mô hình dữ liệu cho các khối dữ liệu (cube) được sử dụng để có thể trực quan hóa dữ liệu. Vùng này có tác dụng tăng tính sẵn sàng của dữ liệu trên báo cáo, không bị gián đoạn khi các công việc (job) chạy trước đó. Sử dụng Trino, Dbt để transform dữ liệu, sau đó lưu kết quả (các khối dữ liệu) vào Clickhouse có vai trò là tầng Gold của hệ thống Lakehouse.

**Xây dựng các Dashboard và Reports:** Sử dụng các công cụ BI như Tableau, PowerBI hoặc Superset để truy vấn và trực quan hóa dữ liệu trong Presentation Stage. Các bảng kích thước và các khối dữ liệu (cube) có thể được kết nối với công cụ BI để tạo ra các báo cáo và dashboard.

**Đồng bộ dữ liệu một cách tự động:** Cập nhật và đồng bộ hóa các báo cáo và dashboard với dữ liệu mới nhất, sử dụng các công cụ ETL như Airflow và lên lịch tự động để thực hiện quá trình này.

## Cơ chế lưu dữ liệu trong Data Lakehouse

SCD (Slowly Changing Dimension) là việc so sánh dữ liệu nguồn với dữ liệu bảng đích hiện có bằng cách sử dụng Khóa nghiệp vụ – Business key (Khóa duy nhất – Unique Key). Nếu không có bản ghi nào khớp thì sẽ coi là bản ghi mới hoặc nếu bản ghi khớp thì sẽ so sánh các thuộc tính với các thuộc tính đã thay đổi nếu dữ liệu có vẻ được cập nhật thì nó cập nhật bản ghi hoặc nếu không thì nó để nguyên như không thay đổi. Slowly Changing Dimension sẽ kiểm tra các thuộc tính cho ba trường hợp: Bản ghi mới, đã thay đổi hoặc chưa thay đổi.

Trong Datawarehouse, đôi khi việc theo dõi sự thay đổi kích thước theo thời gian là rất quan trọng. Điều này giúp chúng ta theo dõi dữ liệu tốt hơn và cũng tạo ra các sản phẩm hiệu quả tùy thuộc vào các trường hợp sử dụng. Và chỉ sử dụng thành phần này chỉ cho các bảng không được cập nhật thường xuyên.

Các loại SCD:

* Type – 0 Giữ lại dữ liệu gốc
* Type – 1 Ghi đè lên dữ liệu hiện có
* Type – 2 Thêm các bản ghi mới trên cùng một bảng
* Type – 3 Thêm cột mới trên cùng một bảng
* Type – 4 Sử dụng bảng lịch sử
* Type – 6 Phương pháp kết hợp (Loại 1 + Loại 2 + Loại 3)

## Giới thiệu các công nghệ sử dụng trong đồ án

### Python

Ngôn ngữ lập trình chính, được sử dụng trong việc phát triển các pipeline ETL, tích hợp dữ liệu, và viết các hàm xử lý logic phức tạp. Được kết hợp với các thư viện như Pandas, SQLAlchemy để tương tác với các hệ thống dữ liệu

### Airflow

Công cụ quản lý workflow và lập lịch (scheduler) cho các pipeline dữ liệu. Hỗ trợ xây dựng, theo dõi, và quản lý các job ETL phức tạp trong hệ thống.

### Trino

Query engine phân tán, hỗ trợ truy vấn dữ liệu trực tiếp từ nhiều nguồn (Data Lake, Data Warehouse). Được dùng trong việc chạy truy vấn SQL nhanh chóng trên hệ sinh thái Data Lakehouse.

### Dbt

Framework hỗ trợ tạo và quản lý các mô hình dữ liệu thông qua SQL và Jinja2. Hỗ trợ phiên bản hóa, kiểm tra, và tổ chức các mô hình dữ liệu.

### Iceberg

Hệ thống lưu trữ định dạng bảng trên Data Lake với khả năng hỗ trợ ACID. Dùng để lưu trữ và quản lý dữ liệu có cấu trúc, cho phép truy vấn hiệu suất cao và khả năng cập nhật/dọn dẹp dữ liệu.

### Hive Metastore

Metastore trung tâm để quản lý metadata của bảng và schema. Dùng để tích hợp với Iceberg và các công cụ khác như Trino.

### MinIO

Giải pháp lưu trữ đối tượng tương thích với S3, được sử dụng làm storage backend cho Data Lake. Lưu trữ dữ liệu thô và xử lý trước khi đưa vào các bảng trong Iceberg.

### Clickhouse

Hệ thống quản lý cơ sở dữ liệu dạng cột, tối ưu hóa cho các truy vấn phân tích nhanh. Dùng để lưu trữ dữ liệu đã được tổng hợp (aggregation) cho các dashboard hoặc báo cáo.

### Supserset

Nền tảng Business Intelligence mã nguồn mở, được dùng để trực quan hóa và tạo báo cáo từ dữ liệu trong ClickHouse, Trino, hoặc các nguồn khác. Hỗ trợ xây dựng dashboard để phân tích dữ liệu theo thời gian thực.

# Thiết kế và xây dựng mô hình hệ thống Data Lakehouse

*Trong chương này, có hai phần chính bao gồm thiết kế mô hình hệ thống và xây dựng cơ sở dữ liệu chuyên dụng cho từng thành phần của hệ thống*

## Yêu cầu đề tài

Hệ thống Data Lakehouse cần đáp ứng được những yêu cầu sau:

* Thu thập và lưu trữ dữ liệu.
* Quản lý và tổ chức dữ liệu
* Xử lý dữ liệu (ETL/ELT).
* Người dùng cuối có thể truy vấn dữ liệu.
* Bảo mật và kiểm soát truy cập.
* Tính linh hoạt và mở rộng:
* Monitoring và Logging

## Thiết kế mô hình Data Lakehouse

Mô hình kiến trúc của hệ thống Data Lakehouse

A diagram of a diagram

Description automatically generated

*Hình ảnh chi tiết kiến trúc hệ thống Data Lakehouse*

Dữ liệu nguồn đi qua các tầng Bronze, Silver, Gold để lấy được báo cáo số liệu, trong đó:

**Data Source:** vùng dữ liệu bao gồm các nguồn dữ liệu thử nghiệm công ty Hebela liên quan đến các thông tin về khách hàng và đội ngũ chăm sóc khách hàng của công ty.

**Bronze:** vùng tập hợp dữ liệu từ các nguồn khác nhau và dữ liệu đã được phân vùng để tối ưu cho việc lưu trữ và truy vấn.

**Silver:** vùng dữ liệu được xử lý, làm sạch, được xây dựng thành các bảng chiều (dim table) và bảng sự kiện (fact table).

**Gold:** vũng dữ liệu đã được làm sạch ở bước cuối cùng, được xây dựng thành các bảng sự kiện tổng hợp (fact aggreate) và các khối (cube) cung cấp data cuối cùng để xây dựng báo cáo.

**Reporting:** tầng này sẽ dựa vào các yêu cầu nghiệp vụ để đưa ra các báo cáo. Các báo cáo có thể dưới dạng bảng số liệu, hoặc các Dashboard.

### Luồng dữ liệu vào Bronze

Tại bước đầu của luồng dữ liệu tổng thể, dữ liệu từ nguồn cụ thể là database sẽ được trích xuất và được upsert đến tầng Bronze.

A diagram of a process

Description automatically generated

*Lưu đồ thuật toán luồng dữ liệu vào tầng Bronze*

### Luồng dữ liệu vào Silver

Sau khi dữ liệu nguồn được trích xuất và tổng hợp lại tại tầng Bronze, dữ liệu tiếp tục được xử lý để đưa vào Silver và tổ chức theo mô hình Star Schema qua các bảng Dimension và Fact.

A diagram of a flowchart

Description automatically generated

*Lưu đồ thuật toán luồng dữ liệu vào tầng Silver*

### Luồng dữ liệu vào Gold

Sau khi dữ liệu được xử lý và tổ chức theo mô hình Star Schema, dữ liệu sẽ được đưa tổ chức thành các khối (cube) để chuẩn bị cho báo cáo.

A diagram of a process

Description automatically generated

*Lưu đồ thuật toán luồng dữ liệu vào tầng Gold*

Sau khi dữ liệu được lưu trữ tại tầng Gold, công cụ xây dựng báo cáo sẽ kết nối tới để xây dựng báo cáo trên các bảng dim, fact và cube.

## Xây dựng cơ sở dữ liệu cho Data Lakehouse

### Tầng Bronze

Tầng Bronze là khu vực chứa dữ liệu được trích xuất từ các nguồn. Từ thông tin các bảng nguồn có được từ bộ dữ liệu chăm sóc khách hàng tại công ty. Vì giới hạn phạm vi cho phép của công ty, nên bộ dữ liệu chỉ bao gồm dữ liệu tại channel whatsapp. Tầng Bronze gồm các bảng như sau:

A screenshot of a computer screen

Description automatically generated

##### Bảng accounts

Là một bảng lưu trữ thông tin người dùng

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | Id |
| name | VARCHAR | Id tài khoản |
| locale | VARCHAR | Ngôn ngữ mặc định của tài khoản |
| doamin | VARCHAR | Tên miền |
| support\_email | VARCHAR | Email hỗ trợ khách hàng |
| custome\_attributes | JSON | Các thuộc tính tùy chỉnh |
| limits | JSON | Thông tin về các giới hạn của tài khoản |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày cập nhật |

##### Bảng users

Là một bảng lưu trữ thông tin người dùng

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | Id |
| name | VARCHAR | Id tài khoản |
| locale | VARCHAR | Ngôn ngữ mặc định của tài khoản |
| doamin | VARCHAR | Tên miền |
| support\_email | VARCHAR | Email hỗ trợ khách hàng |
| custome\_attributes | JSON | Các thuộc tính tùy chỉnh |
| limits | JSON | Thông tin về các giới hạn của tài khoản |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày cập nhật |

##### Bảng account\_users

Là một bảng liên quan đến mối quan hệ giữa tài khoản và người dùng

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | Id |
| account\_id | INTEGER | Id tài khoản |
| user\_id | INTEGER | Id người dùng |
| role | VARCHAR(255) | Vai trò người dùng trong tài khoản |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày cập nhật |

##### Bảng attachments

Là bảng lưu trữ thông tin về các tệp đính kèm mà người dùng tải lên hoặc gửi qua các cuộc hội thoại.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | Id |
| account\_id | INTEGER | Id tài khoản liên quan đến tệp đính kèm |
| user\_id | INTEGER | ID người dùng tải tệp lên |
| message\_id | INTEGER | ID tin nhắn |
| file\_type | VARCHAR | Id người dùng |
| file\_size | INTEGER | Kích thước của tệp được tính bằng byte |
| file\_url | VARCHAR | URL để truy cập hoặc tải xuống tệp |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày cập nhật |

##### Bảng campaigns

Là bảng chứa thông tin về các chiến dịch được sử dụng để quản lý và tổ chức các hoạt động tiếp thị hoặc hỗ trợ khách hàng.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | Id |
| account\_id | INTEGER | Id tài khoản sở hữu chiến dịch |
| title | VARCHAR | Tên hoặc tiêu đề của chiến dịch |
| message | VARCHAR | Nội dung tin nhắn |
| campaign\_type | VARCHAR | Loại chiến dịch |
| status | VARCHAR | Trạng thái của chiến dịch |
| scheduled\_at | TIMESTAMP | Thời gian dự kiến gửi chiến dịch |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày cập nhật |

##### Bảng contacts

Là bảng lưu trữ thông tin về các khách hàng hoặc người liên hệ

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | Id |
| account\_id | INTEGER | Id tài khoản mà người liên hệ thuộc về |
| name | VARCHAR | Tên của người liên hệ |
| email | VARCHAR | Email của người liên hệ |
| phone\_number | VARCHAR | Số điện thoại của người liên hệ |
| identifier | VARCHAR | Mã định danh duy nhất của người liên hệ nếu có |
| custom\_attributes | JSON | Các thuộc tính tùy chỉnh, chứa thông tin bổ như địa chỉ, tuổi,… |
| last\_activity\_at | TIMESTAMP | Thời điểm cuối cùng người liên hệ thực hiện hành động |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày cập nhật |

##### Bảng conversations

Là bảng lưu trữ thông tin về các cuộc hội thoại giữa người dùng và nhân viên hỗ trợ (agents) hoặc hệ thống.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | Id |
| account\_id | INTEGER | Id tài khoản liên quan đến cuộc hội thoại |
| contact\_id | INTEGER | ID của người liên hệ tham gia cuộc hội thoại |
| status | VARCHAR | Trạng thái của cuộc hội thoại |
| source | VARCHAR | Nguồn gốc của cuộc hội thoại |
| is\_automated | BOOLEAN | Tự động hay không |
| resolved\_at | TIMESTAMP | Thời điểm cuộc hội thoại được giải quyết |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày cập nhật |

##### Bảng inboxes

Là bảng lưu trữ thông tin về các hộp thư (inboxes) mà các cuộc hội thoại có thể được gắn vào.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | Id |
| account\_id | INTEGER | Id tài khoản sở hữu hộp thư |
| name | VARCHAR | Tên của hộp thư |
| channel | VARCHAR | Kênh của hộp thư |
| status | VARCHAR | Trạng thái của hộp thư |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày cập nhật |

##### Bảng messages

Là bảng lưu trữ thông tin về các tin nhắn giữa khách hàng và các nhân viên hỗ trợ (agents), hoặc giữa hệ thống và khách hàng trong các cuộc hội thoại.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | Id |
| conversation\_id | INTEGER | Id cuộc hội thoại mà tin nhắn thuộc về |
| account\_id | INTEGER | ID của tài khoản sở hữu tin nhắn |
| sender\_type | VARCHAR | Loại người gửi tin nhắn |
| content | VARCHAR | Nội dung tin nhắn |
| message\_type | VARCHAR | Loại tin nhắn |
| status | VARCHAR | Trạng thái tin nhắn |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày cập nhật |

##### Bảng channel\_whatsapp

Là bảng liên quan đến việc quản lý các cuộc hội thoại và tin nhắn qua kênh WhatsApp, lưu trữ các thông tin như cấu hình kênh, các tin nhắn WhatsApp được gửi/nhận và trạng thái của các tin nhắn.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | Id |
| account\_id | INTEGER | ID tài khoản sở hữu kênh |
| name | VARCHAR | Tên kênh WhatsApp |
| whatsapp\_number | VARCHAR | Số điện thoại WhatsApp được sử dụng cho kênh |
| status | VARCHAR | Trạng thái kênh |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày cập nhật |

### Tầng Silver

Tầng Silver là khu vực chứa dữ liệu sau khi đã chuẩn hóa, làm sạch và được chuyển đổi theo mô hình Star Schema thông qua các bảng dimension, fact và fact giao dịch. Tầng Silver gồm các bảng như sau:

ẢNH

#### Cấu trúc bảng Dimension

Bảng Dimension cung cấp các thông tin, ngữ cảnh cho bảng fact cũng như cung cấp tất cả số liệu thể hiện trong Data Warehouse.

A diagram of a data flow

Description automatically generated

Về cơ bản các bảng Dimension đều có cấu trúc như hình trên. Khóa chính của bảng Dimension là trường dữ liệu (thường là kiểu số) lưu những giá trị duy nhất, không có ý nghĩa, gọi là khóa thay thế (Surrogate Key). Khóa thay thế này thường được sinh ra bằng các luồng ETL xử lý dữ liệu, thường là các khóa tự sinh và chỉ được tạo ra duy nhất trong nội tại của Data Warehouse.

Một thành phần khác của bảng Dimension là khóa chính của dữ liệu trong dữ liệu nghiệp vụ, được gọi là khóa tự nhiên (Natural Key. Ví dụ bảng Dimension Customer sẽ có trường CST\_ID lưu mã khách hàng. Mặc dù trường CST\_ID cũng có thể làm khóa chính cho bảng Dimension Customer nhưng khi thiết kế vẫn phải cung cấp cho bảng Customer Dim một khóa thay thế là CST\_DIM\_ID (trong trường hợp phải nhập dữ liệu từ hai hệ thống nguồn khác nhau dẫn đến khóa tự nhiên có thể bị trùng nhau).

Trong hệ thống Data Lakehouse trong đồ án bao gồm các bảng dimension sau:

##### Bảng ngữ cảnh cho ngày (dim\_date)

Bảng dim\_date là bảng dimension thường được sử dụng trong mô hình Kimball hoặc Star Schema để quản lý và phân tích dữ liệu liên quan đến thời gian. Bảng này cung cấp các thuộc tính chi tiết về ngày tháng, giúp dễ dàng phân tích dữ liệu theo thời gian trong các báo cáo BI.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | Id ngày VD: 20100101 |
| day\_of\_week | INTEGER | Ngày trong tuần |
| day\_of\_month | INTEGER | Ngày trong tháng |
| day\_of\_year | INTEGER | Ngày trong năm |
| is\_last\_day\_of\_month | INTEGER | Ngày cuối tháng |
| is\_weekend | INTEGER | Ngày cuối tuần |
| week\_start\_id | INTEGER | Ngày đầu tuần |
| week\_end\_id | INTEGER | Ngày cuối tuần |
| week\_of\_year | INTEGER | Tuần trong năm |
| month | INTEGER | Tháng |
| quarter | VARCHAR | Quý |
| year | INTEGER | Năm |
| is\_holiday | INTEGER | Ngày nghỉ lễ |
| unix\_timestamp | INTEGER | UNIX timestamp |
| date | DATE | Ngày |

##### Bảng thông tin người dùng (dim\_user)

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | Mã người dùng |
| name | VARCHAR | Tên người dùng |
| display\_name | VARCHAR | Tên hiển thị của người dùng |
| email | VARCHAR | Email người dùng |
| type | VARCHAR | Xác định SuperAdmin |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày sửa |

##### Bảng thông tin tài khoản chi nhánh (dim\_account)

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | ID chi nhánh |
| hrm\_organization\_id | VARCHAR | ID hệ thống |
| name | VARCHAR | Tên chi nhánh |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày sửa |

##### Bảng cầu nối người dùng và tài khoản chi nhánh (dim\_account\_user)

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| account\_id | INTEGER | ID chi nhánh |
| user\_id | INTEGER | ID người dùng |
| role | INTEGER | Xác định nhân viên hay admin |
| active\_at | TIMESTAMP | Thời gian hoạt động |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày sửa |

##### Bảng thông tin liên hệ (dim\_contact)

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | ID liên hệ (khách hàng) |
| account\_id | INTEGER | ID chi nhánh |
| name | VARCHAR | Tên liên hệ |
| middle\_name | VARCHAR | Họ đệm |
| last\_name | VARCHAR | Tên |
| email | VARCHAR | Email |
| phone\_number | VARCHAR | Số điện thoại liên hệ |
| location | VARCHAR | Vị trí |
| country\_code | VARCHAR | Mã quốc gia |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày sửa |

##### Bảng thông tin whatsapp (dim\_channel\_whatsapp)

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | ID kênh whatsapp |
| account\_id | INTEGER | ID chi nhánh |
| phone\_number | VARCHAR | Số điện thoại |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày sửa |

##### Bảng thông tin hộp thư đến (dim\_inbox)

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | ID hộp thư |
| name | VARCHAR | Tên hộp thư |
| channel\_type | VARCHAR | Loại kênh |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày sửa |

##### Bảng thông tin chiến dịch (dim\_campaign)

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | ID chiến dịch |
| display\_id | INTEGER | ID hiển thị |
| title | VARCHAR | Tiêu đề chiến dịch |
| campaign\_status | VARCHAR | Trạng thái chiến dịch |
| campaign\_type | VARCHAR | Loại chiến dịch |
| scheduled\_at | TIMESTAMP | Thời gian lập lịch |
| created\_at | TIMESTAMP | Ngày tạo |
| updated\_at | TIMESTAMP | Ngày sửa |

##### Bảng thông tin template của whatsapp (dim\_ws\_template)

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| id | INTEGER | ID template |
| name | VARCHAR | Tên |
| status | VARCHAR | Trạng thái |
| category | VARCHAR | Loại (MARKETING, UTILITY) |
| language | VARCHAR | Ngôn ngữ |
| header\_type | VARCHAR | Loại (NULL, TEXT, IMAGE, VIDEO, DOCUMENT) |

##### Bảng lỗi whatshapp (dim\_ws\_error\_code)

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| code | INTEGER | Mã lỗi whatsapp |
| error\_type | VARCHAR | Loại lỗi |
| description | VARCHAR | Mô tả |

#### Cập nhật giá trị Dimension

Khi data warehouse nhận thấy có sự thay đổi giá trị trong một bản ghi của Dimension, cần phải được lập trình để có hành động tương thích với nó. Có 2 phương án xử lý được đánh số thứ tự là Kiểu thay đổi 1 và Kiểu thay đổi 2.

Kiểu thay đổi 1 (Type 1 Slowly Changing Dimension) đơn giản là gì đè các dữ liệu bị thay đổi vào bảng dimension. Người thiết kế Data Warehouse chọn kiểu 1 khi dữ liệu nguồn thay đổi khi phần cập nhật này không quan trọng, không làm thay đổi ý nghĩa của bảng Fact.

Kiểu thay đổi 2 (Type 2 Slowly Changing Dimension) cho phép theo dõi các thay đổi xảy ra trong bảng dimension và liên kết chính xác giữa bản ghi fact với bản ghi dimension đang có hiệu lực. Khi Data Warehouse nhận ra dữ liệu nguồn có cập nhật, thay vì ghi đè, hệ thống cập nhật trạng thái bản ghi cũ và sinh thêm một bản ghi mới vào bảng dimension. Bản ghi này được gán cho một khóa thay thế mới và từ lúc này, Data Warehouse sẽ dùng bản ghi mới này để liên kết với bản ghi fact được sinh ra. Các bản ghi fact sinh ra trong những ngày trước đó vẫn liên kết với bản ghi dimension cũ

A diagram of a flowchart

Description automatically generated

*Xử lý dữ liệu thay đổi Slowly Changing Dimension*

Kiểu thay đổi này thể hiện rõ nhất sự thay đổi của dữ liệu theo dòng thời gian vì mỗi sự thay đổi dù nhỏ nhất của thực thể trên dữ liệu nguồn đều được ghi nhận trong Data Warehouse. Hình dưới đây là một ví dụ cho phương án thay đổi giá trị dimension theo kiểu 2. Khách hàng Phong qua một khoảng thời gian nhận những công việc khác nhau, mỗi lần thay đổi công việc đều được Data Warehouse ghi nhận lại là một bản ghi mới trong Data Warehouse (mỗi bản ghi nhận một khóa thay thế tương ứng). A diagram of a block diagram

Description automatically generated

*Ví dụ cho SCD-2*

#### Cấu trúc bảng Fact

Mỗi bảng Fact được xác định bằng độ chi tiết của bảng dữ liệu. Các loại bảng fact thường gặp trong mô hình Data Lakehouse:

* **Fact giao dịch (Transaction Fact Table):** Mỗi hàng trong bảng fact giao dịch tương ứng với một sự kiện đo lường tại một thời điểm cụ thể. Bảng fact giao dịch nguyên tử là loại bảng fact chi tiết nhất và thể hiện rõ ràng nhất, cho phép phân tích dữ liệu giao dịch ở mức độ chi tiết nhất.

A screenshot of a phone

Description automatically generated

*Hình ảnh cấu trúc bảng fact giao dịch tin nhắn*

### Tầng Gold

Tầng Gold là khu vực chứa dữ liệu sau khi đã được xây dựng từ bảng dim, fact từ đó xây dựng nên các bảng fact tổng hợp, fact tích lũy, fact định kỳ và cube là những bảng cuối cùng phục vụ mục đích báo cáo.

#### Cấu trúc bảng Fact

##### **Bảng Fact tổng hợp (Aggregate Fact Table)**

* Là một loại bảng fact được tạo ra bằng cách tổng hợp dữ liệu từ các bảng fact chi tiết để cải thiện hiệu suất truy vấn, đặc biệt trong các trường hợp báo cáo hoặc phân tích cần xử lý lượng dữ liệu lớn.
* Bảng fact tổng hợp account

A screenshot of a computer

Description automatically generated

*Cấu trúc bảng fact tổng hợp hoạt động người dùng*

##### **Bảng Fact Tích Lũy (Accumulating Snapshot Fact Table)**

* Mỗi bản ghi trong bảng fact tích lũy tóm tắt các phép đo cho một quy trình hoặc quy trình công việc hoàn chỉnh. Bảng fact được cập nhật khi quy trình hoặc quy trình công việc tiến triển.

A screenshot of a phone

Description automatically generated

*Cấu trúc bảng fact tích lũy về hoạt động người dùng*

##### **Bảng Fact Ảnh Chụp Định Kỳ (Periodic Snapshot Fact Table)**

* Mỗi hàng trong bảng fact ảnh chụp định kỳ tóm tắt nhiều sự kiện đo lường xảy ra trong một khoảng thời gian tiêu chuẩn, chẳng hạn như một ngày, một tuần hoặc một tháng. Cấp độ chi tiết là khoảng thời gian, không phải là giao dịch riêng lẻ.

A screenshot of a computer

Description automatically generated

*Cấu trúc bảng fact định kỳ về hoạt động người dùng*

# Triển Khai mô hình Data Lakehouse

## Ứng dụng ETL bộ dữ liệu chăm sóc khách hàng lên Data Lakehouse

## Khai thác báo cáo dữ liệu

## Kết luận chương

# Kết Luận

Trong thời gian làm đề tài này đã giúp em tìm hiểu và nắm bắt được nhiều kiến thức bổ ích, đồng thời rút ra nhiều kinh nghiệm thực tế. Do hạn chế về thời gian cũng như trình độ nên chương trình không tránh khỏi nhiều sai sót. Em xin chân thành cảm ơn và mong nhận được sự đóng góp ý kiến từ các thầy cô và các bạn để đề tài được hoàn thiện hơn.

Đề tài ***“Xây dựng website bán thực phẩm sạch HFresh sử dụng ASP.NET”*** của em đã hoàn thiện và đạt được các kết quả sau:

Đã phân tích, khảo sát qua việc đọc tài liệu và các bài báo từ đó đã làm rõ được các quy nghiệp cũng như nghiệp vụ của hệ thống website bán hàng từ đó vẽ được biểu đồ use case tổng quát, phân rã các use case để xây dựng được các đặc tả yêu cầu của từng usecase, đưa ra được các biểu đồ hoạt động, trình tự, trạng thái và biểu đồ lớp, mô hình hoạt động của hệ thống để từ đó xây dựng được cơ sở dữ liệu hợp lý và tối ưu.

Đáp ứng được những yêu cầu mà người dùng đã đặt ra về tiêu chí, hoạt động được, dễ sử dụng, đầy đủ chức năng cần thiết như thêm sửa xóa được giao dịch, quản lý sản phẩm, ngoài ra còn có chức năng tìm kiếm bài viết hay cho ẩn hiện tin lên trang chủ website để từ đó dễ quản lý hơn, thống kê báo cáo, xây dựng và thiết kế trang chủ bắt mắt dễ nhìn, giao diện trang quản lý thân thiện, dễ sử dụng, trực quan cho người dùng.

Do hạn chế về mặt thời gian và kiến thức nên hệ thống của em chưa hoàn thiện được các chức năng như quên mật khẩu, thanh toán online,... Hệ thống mới chỉ chạy đúng, đầy đủ nghiệp vụ và thông luồng các chức năng chính ngoài ra những trường hợp ngoại lệ vẫn chưa được kiểm tra và kiểm thử sâu. Cấu trúc chương trình vẫn chưa tối ưu. Vẫn còn 1 số vấn đề tồn đọng chưa thể giải quyết dứt điểm.

Trong thời gian sắp tới, em mong nhận được sự giúp đỡ quý báu của thầy cô và bạn bè để em có thể hoàn thiện và bổ sung đề tài tốt hơn để xây dựng hoàn thiện các chức năng cụ thể như: Hoàn thiện và bổ sung thêm chức năng tìm kiếm nâng cao, chức năng quên mật khẩu cho người dùng. Xây dựng cơ chế bảo mật và an toàn dữ liệu tốt hơn. Triển khai cơ chế phân quyền mới có tính linh động hơn. Xây dựng chức năng chat trực tuyến để người bán và người mua có thể trao đổi dễ dàng hơn. Phát triển và kiểm thử hệ thống trên tất cả các trình duyệt và phiên bản để đánh giá được tốc độ xử lý cũng như khả năng truy cập.

TÀI LIỆU THAM KHẢO