**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC PHENIKAA**

Logo, company name

Description automatically generated

**PHẠM THÀNH TRUNG -20010931**

**NGUYỄN ĐỨC MẠNH -20010783**

**DƯƠNG MINH HIẾU -20010861**

**ĐỒ ÁN CƠ SỞ**

**NGÀNH : CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI : “BUILD & MANAGE YOUR DATA LAKE WITH DELTA LAKE”**

**Người hướng dẫn: TS.Trần Đăng Hoan**

**HÀ NỘI - 2022**

**LỜI CẢM ƠN**

Lời đầu tiên chúng em xin gửi lời cảm ơn sâu sắc đến Thầy – Tiến sĩ Trần Đăng Hoan, người đã tận tình chỉ bảo, hướng dẫn chúng em trong suốt quá trình thực hiện đề tài, để chúng em có thể hoàn thành nhiệm vụ đặt ra trong đề tài **“Build & Manage Your Data Lake With Delta Lake”.**

Chúng em cũng xin gửi lời cảm ơn đến các thầy cô trong Khoa Công nghệ thông tin, Trường Đại học Phenikaa đã nhiệt tình giảng dạy và tạo mọi điều kiện giúp đỡ chúng em trong quá trình học tập, thực hiện đề tài.

Hà Nội, Ngày 29 tháng 12 năm 2022

Nguyến Đức Mạnh

Phạm Thành Trung

Dương Minh Hiếu

**LỜI CAM ĐOAN**

Chúng em xin cam đoan tất cả nội dung, kết quả trong đề tài đồ án cơ sở **“Build & Manage Your Data Lake With Delta Lake”** là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Chúng em xin chịu mọi trách nhiệm với đề tài của nhóm mình.

**MỤC LỤC**

[LỜI NÓI ĐẦU 5](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_3znysh7)

[CHƯƠNG 1. TỔNG QUAN 6](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_2et92p0)

[1.1. Giới thiệu đề tài 6](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_tyjcwt)

[1.1.1. Tên đề tài 6](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_3dy6vkm)

[1.1.2. Mục đích, yêu cầu đề tài 6](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_1t3h5sf)

[1.1.3. Bảng phân công nhiệm vụ 6](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_4d34og8)

[1.2. Khảo sát hệ thống 6](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_17dp8vu)

[1.2.1. Phạm vi và đối tượng 6](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_3rdcrjn)

[1.2.2. Phương pháp nghiên cứu 6](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_26in1rg)

[1.3. Giới thiệu DataLake 7](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_26in1rg)

[1.4. Thiết kế kiến trúc dữ liệu trong Data Lake 12](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_26in1rg)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 17](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_2jxsxqh)

[2.1. Giới thiệu về ngôn ngữ Python 17](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_z337ya)

[2.2. Apache Spark 18](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_3j2qqm3)

[2.3. Delta Lake 19](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_1y810tw)

[2.4. Azure Databricks 19](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_1y810tw)

[CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 20](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_2xcytpi)

[3.1. Delta Architecture 20](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_1ci93xb)

[3.2. Thiết kế cơ sở dữ liệu 22](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_3whwml4)

[CHƯƠNG 4: XÂY DỰNG CHƯƠNG TRÌNH 23](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_279ka65)

[4.1 Xây dựng chương trình 23](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_meukdy)

[KẾT LUẬN VÀ KIẾN NGHỊ 38](file:///C:\Users\asus\Downloads\Bao_Cao_TN_TuyenSinhTHPT_Viet_Thuy_Trang%20(1).docx#_338fx5o)

**LỜI NÓI ĐẦU**

Ngày nay, với sự phát triển mạnh mẽ của khoa học công nghệ, đặc biệt là sự phát triển nhanh chóng của lĩnh vực công nghệ thông tin, công nghệ thông tin được ứng dụng trong mọi lĩnh vực của đời sống giúp cho mọi công việc đơn giản, chính xác, dễ dàng hơn.

**CHƯƠNG 1. TỔNG QUAN**

**1.1 Giới thiệu đề tài**

*1.1.1 Tên đề tài*

**Build & Manage Your Data Lake With Delta Lake**

*1.1.2 Mục đích, yêu cầu đề tài*

Mục đích nhóm em hướng đến đó là tìm hiểu và nghiên cứu về Data-Lake, hiểu rõ xem Data-Lake là gì, tìm hiểu các khía cạnh, đặc điểm quan trọng của Data-Lake, khi nào thì chúng ta cần sử dụng Data-Lake và sử dụng trong trường hợp nào. Tìm hiểu về cách tiếp cận của Data-Lake.

*1.1.3 Bảng phân công nhiệm vụ*

*Bảng 1. 1. Bảng phân công nhiệm vụ*

|  |  |
| --- | --- |
| Phạm Thành Trung | Tìm hiểu Data Lake, Quá trình xây dựng Data Lake, Làm báo cáo. |
| Nguyễn Đức Mạnh | Xây dựng chương trình, Thiết kế Database , Làm báo cáo |
| Dương Minh Hiếu | Tìm hiểu các công nghệ sử dụng, Làm báo cáo |

**1.2 Khảo sát hệ thống**

*1.2.1 Phạm vi và đối tượng*

* Tìm hiểu các ngôn ngữ lập trình và hệ cơ sở dữ liệu phù hợp để xây dựng hệ thống.
* Tìm hiểu về quá trình xây dựng một hệ thống Data-Lake và xác định những công việc cần thực hiện.

*1.2.2 Phương pháp nghiên cứu*

* Từ những gì đã thu thập được tiến hành so sánh ưu nhược điểm của những hệ thống cũ và sử dụng những hệ thống mới.
* Xác định đầy đủ các chức năng để tiến hành xây dựng một Data-Lake hoàn chỉnh theo yêu cầu đề ra.

**1.3. Giới thiệu về Data Lake***.*

**DATA LAKE LÀ GÌ?**

Hồ dữ liệu là một kho lưu trữ tập trung cho phép bạn lưu trữ tất cả dữ liệu có cấu trúc và phi cấu trúc của mình ở bất kỳ quy mô nào. Bạn có thể lưu trữ dữ liệu của mình nguyên trạng mà không cần phải cấu trúc dữ liệu trước và chạy các loại phân tích khác nhau — từ bảng điều khiển và hình ảnh hóa đến xử lý dữ liệu lớn, phân tích thời gian thực và học máy để đưa ra các quyết định tốt hơn.

**DÂN CHỦ HÓA DỮ LIỆU LÀ GÌ?**

Dân chủ hóa dữ liệu có nghĩa là mọi người đều có quyền truy cập vào dữ liệu và không có sự ngăn cản tạo ra nút thắt cổ chai ở cổng dữ liệu. Nó yêu cầu chúng ta cho phép truy cập dữ liệu một cách dễ dàng để mọi người hiểu được dữ liệu và có thể sử dụng dữ liệu để đẩy nhanh quá trình ra quyết định và tìm ra các cơ hội cho một tổ chức. Mục tiêu dân chủ hóa dữ liệu là để mọi người sử dụng dữ liệu bất kỳ lúc nào để đưa ra quyết định mà không có rào cản đối với việc tiếp cận.

**TẠI SAO PHẢI DÂN CHỦ HÓA DỮ LIỆU?**

Càng có nhiều người có chuyên môn đa dạng có khả năng truy cập dữ liệu dễ dàng và nhanh chóng sẽ giúp tổ chức của bạn xác định và thực hiện hành động đối với những thông tin chi tiết quan trọng về doanh nghiệp. Khi bạn cho phép truy cập dữ liệu vào bất kỳ cấp nào trong công ty của mình, điều đó sẽ trao quyền cho các cá nhân ở tất cả các cấp quyền sở hữu và trách nhiệm sử dụng dữ liệu trong quá trình ra quyết định của họ.

**CÁC KHÁI NIỆM CƠ BẢN TRONG DATA LAKE**

Chart, bubble chart

Description automatically generatedHình dưới là các khái niệm chính mà mọi người cần hiểu để nắm và triển khai được một hệ thống Data Lake:

**Data Ingestion**: Cung cấp và triển khai các công cụ đồng bộ và tiền xử lý dữ liệu để đưa vào Data Lake

**Data Storgare**: Việc lưu trữ dữ liệu trên Data Lake đòi hỏi phải có tính mở rộng, chi phí thấp và cho phép truy cập nhanh tới dữ liệu cần khai phá và đặc biệt hỗ trợ đa định dạng.

**Data** [**Governance**](https://www.facebook.com/notes/c%E1%BB%99ng-%C4%91%E1%BB%93ng-big-data-vi%E1%BB%87t-nam/data-governance-overview-l%C3%A0m-ch%E1%BB%A7-t%C3%A0i-nguy%C3%AAn-d%E1%BB%AF-li%E1%BB%87u/489320268613495/): Quản trị dữ liệu là một quá trình quản lý tính khả dụng, khả năng tương tác, bảo mật, tri thức nghiệp vụ và tính toán vẹn của dữ liệu trong tổ chức.

**Security**: Bảo mật và An toàn thông tin cần được thực hiện trong mọi lớp của hồ dữ liệu. Nó bắt đầu với việc lưu trữ, xử lý và khai thác. Đơn giản là việc cấm truy cập các tầng với những người không được cho phép. Nó nên hỗ trợ nhiều công cụ truy cập dữ liệu thông quan giao diện hoặc các màn hình quản lý.

**Data Quality**: Chất lượng dữ liệu là một thành phần thiết yếu của hệ thống Dữ liệu, đặc biệt với Data Lake. Dữ liệu khi sử dụng phải đảm bảo tính chính xác, toàn vẹn và kịp thời như vậy sẽ đem lại giá trị cho kinh doanh. Rủi ro về dữ liệu không chính xác, thiếu sẽ dẫn tới các quyết định sai lầm.

**Data Discovery**: Khai phá dữ liệu là một bước quan trọng trước khi bạn có thể bắt đầu phân tích chúng. Trong giai đoạn này, các kỹ thuật làm sạch và gán cấu trúc được sử dụng để gia tăng giá trị cho dữ liệu, giúp chúng được tổ chức và diễn giải dễ hiểu.

**Data Auditing**: Kiếm soát tác động với dữ liệu giúp theo dõi các tác động, thay đổi, rủi ro và tính tuân thủ đối với những người sử dụng.

**Data Lineage**: Dòng đời dữ liệu cho phép người dùng nắm được nguồn gốc của thông tin mình đang sử dụng, nơi nó đến, di chuyển qua và những gì được tác động lên nó. Nó giúp giảm các lỗi trong bước phân tích dữ liệu từ gốc tới đích.

**Data Exploration**: Không giống như Data Discovery, giai đoạn này người sử dụng cần phân tích dữ liệu, họ cần lẩy mẫu và thử nghiệm với các thông tin thu thập. Các thông tin cần được phối hợp với nhau để dễ dàng phát triển và xây dựng các bài toán phù hợp.

**TẠI SAO MỖI CÔNG TY VỀ CNTT PHẢI CẦN CÓ DATA LAKE?**

Các tổ chức tạo ra giá trị kinh doanh thành công từ dữ liệu của họ sẽ hoạt động tốt hơn các tổ chức khác. Một cuộc khảo sát của Aberdeen cho thấy các tổ chức sử dụng Data Lake vượt trội hơn khoảng 9% doanh thu với các công ty không sử dụng nó. Những nhà lãnh đạo này đã có thể thực hiện các loại phân tích mới như máy học qua các nguồn mới như tệp nhật ký, dữ liệu từ các luồng nhấp chuột, phương tiện truyền thông xã hội và các thiết bị kết nối internet được lưu trữ trong hồ dữ liệu.

Điều này đã giúp họ xác định và hành động theo các cơ hội phát triển kinh doanh nhanh hơn bằng cách thu hút và giữ chân khách hàng, thúc đẩy năng suất, chủ động bảo trì thiết bị và đưa ra quyết định sáng suốt.

**SỰ KHÁC BIỆT GIỮA DATA LAKE VỚI CÁC CÔNG NGHỆ ĐI TRƯỚC?**

Data lake chứa tất cả dữ liệu thô và dữ liệu chưa qua xử lý và cho phép phân tích sâu hơn.

Data lake lưu trữ nhiều dữ liệu phi truyền thống hơn, khó định lượng và khó đo lường hơn.

Data lake không phân biệt dữ liệu mà nó lưu trữ mọi mức độ liên quan của dữ liệu.

Data lake duy trì dữ liệu ở trạng thái thô và chưa qua xử lý, cho phép mọi người có quyền truy cập bất kể có bất kỳ sự thay đổi nào.

Data lake có thời gian lưu trữ dữ liệu không giới hạn. Vì chúng chỉ lưu trữ dữ liệu thô.

**Data Lake và Data Warehouse: đâu là lựa chọn phù hợp?**

Cả hai mô hình đều cần thiết đối với tổ chức, doanh nghiệp. Data Lake được phát triển từ nhu cầu khai thác dữ liệu lớn và hưởng lợi từ dữ liệu thô, phục vụ cho học máy. Trong khi đó, Data Warehouse hướng đến người dùng là doanh nghiệp. Xu hướng ứng dụng Data Lake và Data Warehouse trong các lĩnh vực cụ thể được nêu ra dưới dây:

##### ***Y tế***

Data Warehouse đã được sử dụng phổ biến trong y học nhiều năm nay, song chưa gặt hái được thành công hoàn toàn. Do tính phi cấu trúc của phần lớn dữ liệu y tế (ghi chú của bác sĩ, dữ liệu lâm sàng,…) và nhu cầu hiểu biết sâu sắc theo thời gian thực, Data Warehouse rõ ràng chưa phải mô hình lý tưởng. Vì vậy, Data Lake, với khả năng cho phép kết hợp dữ liệu có cấu trúc và phi cấu trúc, có xu hướng phù hợp hơn với các đơn vị chăm sóc sức khỏe.

##### ***Giáo dục***

Trong những năm gần đây, giá trị của dữ liệu lớn đối với cải cách giáo dục ngày một trở nên rõ rệt. Dữ liệu lớn đang trở thành công cụ giúp cá nhân hóa và hợp lý hóa ngành giáo dục. Thực tế, phần lớn lượng dữ liệu khổng lồ này là dữ liệu thô, do đó, các tổ chức giáo dục sẽ được hưởng lợi tốt nhất từ ​​tính linh hoạt của Data Lake.

##### ***Tài chính***

Trong lĩnh vực tài chính, cũng như các môi trường kinh doanh khác, Data Warehouse thường là mô hình lưu trữ tối ưu nhất, bởi nó phù hợp với năng lực đọc dữ liệu của tất cả thành viên trong doanh nghiệp.

##### ***Giao thông vận tải***

Trong ngành giao thông vận tải, đặc biệt là quản lý chuỗi cung ứng, khả năng dự đoán đến từ dữ liệu linh hoạt trong Data Lake có thể mang lại lợi ích to lớn, cụ thể là cắt giảm chi phí bằng cách kiểm tra dữ liệu từ các biểu mẫu trong quy trình vận chuyển.

Những khác biệt chính về cấu trúc, quy trình, người dùng và khả năng truy cập khiến mỗi mô hình trở nên độc đáo. Tùy thuộc vào nhu cầu cụ thể, việc phát triển Data Lake hoặc Data Warehouse phù hợp sẽ là công cụ đắc lực giúp doanh nghiệp tăng trưởng.

**LỘ TRÌNH THÀNH CÔNG CỦA DATA LAKE**

Bây giờ chúng ta đã biết những gì cần thiết để một hồ dữ liệu thành công và những cạm bẫy đối với

tìm ra cho, làm thế nào để chúng ta đi về việc xây dựng một? Thông thường, các công ty làm theo điều này

tiến trình:

1. Xây dựng cơ sở hạ tầng (thiết lập và chạy cụm Hadoop).

2. Tổ chức hồ dữ liệu (tạo các vùng để sử dụng bởi các cộng đồng người dùng khác nhau và

nhập dữ liệu).

3. Thiết lập hồ dữ liệu để tự phục vụ (tạo danh mục nội dung dữ liệu, cấp phép thiết lập

và cung cấp các công cụ cho các nhà phân tích sử dụng).

4. Mở hồ dữ liệu cho người dùng.

**1.4. Thiết kế kiến trúc dữ liệu trong Data Lake**

Data Lake là một cách tiếp cận hoàn toàn mới giữa sự kết hợp sức mạnh của Big Data và khả năng Self-service. Nhiều doanh nghiệp hiện nay đã phát triển hoặc triển khai hệ thống này trong hoạt động điều hành, sản xuất kinh doanh.

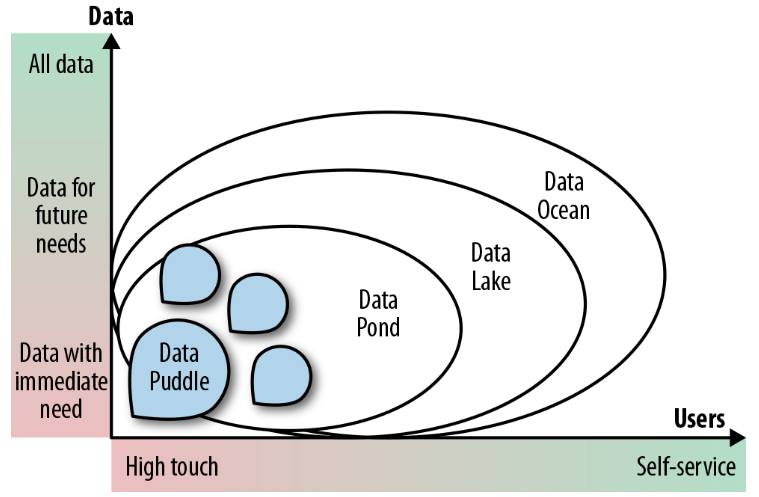
Vậy cách thiết kế hệ thống và tổ chức dữ liệu trong Data Lake như thế nào?

*1.4.1. Khái niệm cơ bản*

Data Lake là một khái niệm tương đối mới, vì vậy để xác định được các kiến trúc xây dựng bạn có thể tham khảo một số định nghĩa sau đây:

* D*ata Puddle* là một vùng dữ liệu, cơ bản như một Data Mart với một mục đích hoặc phục vụ cho một dự án. Áp dụng trong những bước đầu tiên áp dụng công nghệ Big Data.
* D*ata Pond* là một tập hợp các vùng dữ liệu, có thể coi như một kho dữ liệu nhưng được thiết kế chưa tối ưu, giúp giảm tải kho dữ liệu truyền thống hiện có. Tuy có chi phí công nghệ thấp hơn, khả năng mở rộng tốt nhưng lại đòi hỏi chi phí CNTT lớn, kèm theo việc hạn chế trong tính khả dụng của dữ liệu nên nó không thực sự giúp tối ưu khả năng Self-service và Data-driven hỗ trợ ra quyết định cho người dùng doanh nghiệp.
* D*ata Lake* khác với Data Pond ở 2 đặc điểm quan trọng: đầu tiên, nó hỗ trợ khả năng Self-Service, nơi mà người dùng có thể tìm và sử dụng các dữ liệu mà họ muốn mà không cần nhờ tới sự trợ giúp của bộ phận CNTT. Hai là nó nhằm mục đích chứa các dữ liệu mà ngay hiện tại doanh nghiệp hay các cá nhân cũng chưa có nhu cầu sử dụng.
* *Data Ocean* mở rộng khả năng self-service dữ liệu và data-driven hỗ trợ ra quyết định dữ liệu, bất cứ nơi nào có thể, bất kể nó có có tải vào hệ thống Data Lake hay không.

*Hình 1:* *Minh hoạ sự khác biệt giữa các khái niệm trên. Khi mở rộng từ Puddle tới Pond, Lake và Ocean, lượng dữ liệu và số người cũng tăng theo kèm theo đó là mở rộng khả năng Self-service*



Vậy cần làm gì để xây dựng thành công một Data Lake? Tương tự như bất kỳ dự án nào, bắt buộc phải có việc liên kết nó với chiến lược của công ty kèm theo việc đầu tư và điều hành xuyên suốt. Ngoài ra, cần xác định 3 điều kiện chính trước khi bắt đầu:

· Nền tảng phù hợp: đánh giá và lựa chọn giữa Hadoop, Amazon Web Service, Microsoft Azure,..

· Dữ liệu : mục đích lưu càng nhiều dữ liệu càng tốt với định dạng gốc

· Giao diện tương tác: Khả năng cung cấp Self-service ở mức độ đơn giản cho người dùng, đảm bảo người dùng có thể tự tìm kiếm và khai thác

*1.4.2. Thiết kế kiến trúc hạ tầng*

Roap map: Chúng ta đã có các điều kiện cần, vậy các bước chính cần thực hiện là gì?

1. Triển khai cơ sở hạ tầng cho lưu trữ (Hadoop là một lựa chọn không tồi)

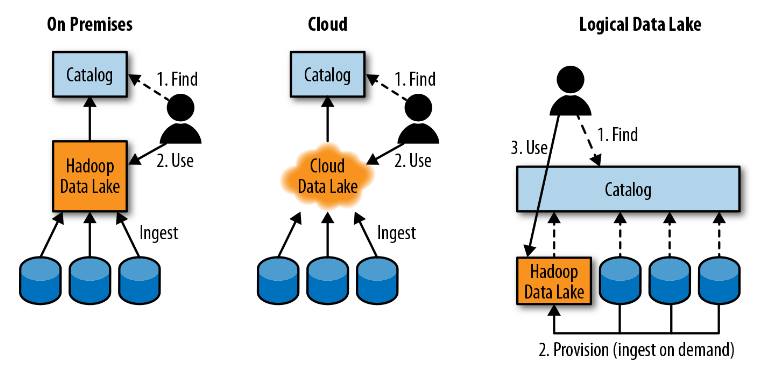
2. Tổ chức Data Lake (tạo các Zone để phân vùng cho các người dùng, dữ liệu khác nhau).

3. Thiết lập Self-service (tạo các danh mục quản lý dữ liệu, thiết lập queyefn và cung cấp các công cụ khai thác, phân tích dữ liệu).

4. Vận hành và cung cấp Data Lake cho người dùng

Vào năm 2015, hầu hết các doanh nghiệp xây dựng Data Lake trên hạ tầng On-Premises với Hadoop bản nguồn mở hoặc thương mai. Nhưng từ 2018, ít nhất một nửa số doanh nghiệp đã chuyển sang Cloud hoặc dạng Hybrid sử dụng cả On-Premises và Cloud. Nhiều công ty đã xây dựng nhanh chóng các Data Lake, sự đa dạng trong kiến trúc khiến các công ty cần phải xem xét lại các hướng tiếp cận phù hợp.

*Hình 2: So sánh 3 cách tiếp cận khác nhau đều với một mục đích cung cấp danh mục và khả năng khai thác dữ liệu cho người dùng*



*1.4.3. Thiết kế lưu trữ*

- Hầu hết các Data Lake được tổ chức theo cùng một cách, với các Zone khác nhau:

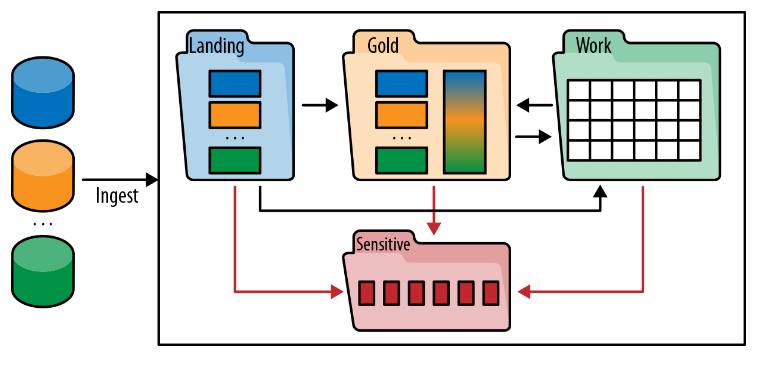
· **Raw hay Landing Zone** là nơi dữ liệu được đưa vào và xử lý, làm chuẩn với mục tiêu giống với hiện trạng ban đầu tối đa nhất.

· **Gold hay Production Zone** là nơi lưu trữ dữ liệu đã được xử lý, tổng hợp sạch sẽ.

· **Dev hay Work Zone** là nơi có nhiều nhân sự phát triển, phân tích, khai phá làm việc và được tổ chức theo nhu cầu của người dùng, theo dự án hoặc theo chủ đề, khi hoàn thành sản phẩm triển khai, dữ liệu sẽ được chuyển lên Gold Zone.

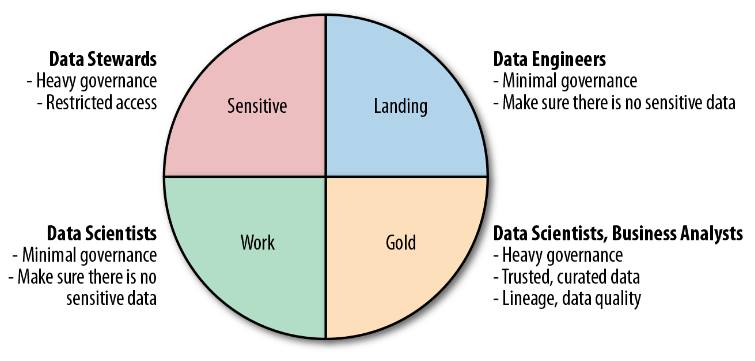
· **Sensitive Zone** là nơi chưa dữ liệu nhạy cảm, dữ liệu mã hóa phục vụ để trao đổi với các hệ thống ngoài Data Lake.

*Hình 3 minh họa cho tổ chức này*



Trong những năm gần đây, các nhà phân tích dữ liệu từ Gartner đã thúc đẩy khái niệm Multi-Modal IT (CNTT đa phương thức) , về cơ bản ý tưởng này là việc quản trị dữ liệu sẽ phản ánh việc sử dụng và yêu cầu của cộng đồng người dùng. Với các Zone khác nhau, sẽ có cách chính sách quản lý và phân quyền phù hợp. Ví dụ, dữ liệu trong Gold Zone thường được tổ chức chặt chẽ, đảm bảo chất lượng và độ chính xác. Những người dùng khác nhau sẽ có nhu cầu với từng vùng, những người phân tích kinh doanh sẽ sử dụng chủ yếu Gold Zone để khai thác, các nhân viên phát triển, vận hành hệ thống sẽ đưa và xử lý dữ liệu vào Raw Zone sau đó chuyển sang Gold Zone, những người thử nghiệm hoặc xây dựng bài toán học máy sẽ sử dụng Work Zone để làm việc trước khi triển khai sản phẩm.

*Hình 4 minh họa các cấp quản trị khác nhau cho các Zone khác nhau*



**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

**2.1. Giới thiệu về ngôn ngữ Python** Python là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh" \o "Ngôn ngữ lập trình) bậc cao cho các mục đích lập trình đa năng, do [Guido van Rossum](https://vi.wikipedia.org/w/index.php?title=Guido_van_Rossum&action=edit&redlink=1) tạo ra và lần đầu ra mắt vào năm [1991](https://vi.wikipedia.org/wiki/1990). Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển [trí tuệ nhân tạo](https://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o" \o "Trí tuệ nhân tạo). Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Python hoàn toàn [tạo kiểu động](https://vi.wikipedia.org/w/index.php?title=T%E1%BA%A1o_ki%E1%BB%83u_%C4%91%E1%BB%99ng&action=edit&redlink=1" \o "Tạo kiểu động (trang không tồn tại)) và dùng cơ chế [cấp phát bộ nhớ tự động](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3n_l%C3%BD_b%E1%BB%99_nh%E1%BB%9B" \o "Quản lý bộ nhớ); do vậy nó tương tự như [Perl](https://vi.wikipedia.org/wiki/Perl), [Ruby](https://vi.wikipedia.org/wiki/Ruby_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Scheme](https://vi.wikipedia.org/wiki/Scheme), [Smalltalk](https://vi.wikipedia.org/wiki/Smalltalk), và [Tcl](https://vi.wikipedia.org/wiki/Tcl" \o "Tcl). Ban đầu, Python được phát triển để chạy trên nền [Unix](https://vi.wikipedia.org/wiki/Unix). Nhưng rồi theo thời gian, Python dần mở rộng sang mọi [hệ điều hành](https://vi.wikipedia.org/wiki/H%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh" \o "Hệ điều hành) từ [MS-DOS](https://vi.wikipedia.org/wiki/MS-DOS) đến [Mac OS](https://vi.wikipedia.org/wiki/Mac_OS), [OS/2](https://vi.wikipedia.org/wiki/OS/2), [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux) và [các hệ điều hành khác thuộc họ Unix](https://vi.wikipedia.org/wiki/T%C6%B0%C6%A1ng_t%E1%BB%B1_Unix" \o "Tương tự Unix).

* Đặc trưng Python:

+ Python là một [ngôn ngữ lập trình đa mẫu hình](https://vi.wikipedia.org/wiki/M%E1%BA%ABu_h%C3%ACnh_l%E1%BA%ADp_tr%C3%ACnh" \o "Mẫu hình lập trình). [Lập trình hướng đối tượng](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng" \o "Lập trình hướng đối tượng) và [lập trình cấu trúc](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_c%E1%BA%A5u_tr%C3%BAc" \o "Lập trình cấu trúc) được hỗ trợ hoàn toàn, và nhiều tính năng của nó cũng hỗ trợ lập trình hàm và [lập trình hướng khía cạnh](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_kh%C3%ADa_c%E1%BA%A1nh" \o "Lập trình hướng khía cạnh) (bao gồm [siêu lập trình](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_meta" \o "Lập trình meta)và [siêu đối tượng](https://vi.wikipedia.org/w/index.php?title=Si%C3%AAu_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng&action=edit&redlink=1" \o "Siêu đối tượng (trang không tồn tại)) (phương thức thần kì)). Các mẫu hình khác cũng được hỗ trợ thông qua các phần mở rộng, bao gồm [thiết kế theo hợp đồng](https://vi.wikipedia.org/w/index.php?title=Thi%E1%BA%BFt_k%E1%BA%BF_theo_h%E1%BB%A3p_%C4%91%E1%BB%93ng&action=edit&redlink=1" \o "Thiết kế theo hợp đồng (trang không tồn tại)) và [lập trình logic](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_logic" \o "Lập trình logic).

+ Thiết kế của Python cung cấp một số tính năng cho lập trình hàm giống như trong ngôn ngữ [Lisp](https://vi.wikipedia.org/wiki/Lisp)

**2.2. Apache Spark**

[**Apache Spark**](https://bizflycloud.vn/tin-tuc/apache-spark-20210726115949898.htm) là một framework xử lý dữ liệu mã nguồn mở trên quy mô lớn. Spark cung cấp một giao diện để lập trình các cụm tính toán song song với khả năng chịu lỗi. Khả năng tính toán phân tán của Apache Spark khiến nó rất phù hợp với big data và machine learning, vốn đòi sức mạnh tính toán khổng lồ để làm việc trên các kho dữ liệu lớn. Spark cũng giúp loại bỏ một số gánh nặng lập trình khỏi vai của các nhà phát triển với một API dễ sử dụng đảm nhiệm phần lớn công việc khó khăn của tính toán phân tán và xử lý dữ liệu lớn. Spark có các API dễ sử dụng để làm việc trên các tập dữ liệu lớn, bao gồm hơn 100 toán tử để chuyển đổi dữ liệu và các API dataframe quen thuộc để xử lý dữ liệu bán cấu trúc.

## **Kiến trúc của Apache Spark**

Apache Spark bao gồm hai thành phần chính: trình điều khiển (driver) và trình thực thi (executors). Trình điều khiển dùng để chuyển đổi mã của người dùng thành nhiều tác vụ (tasks) có thể được phân phối trên các nút xử lý (worker nodes).

Trình thực thi chạy trên các nút xử lý và thực hiện các nhiệm vụ được giao cho chúng. Spark cũng có thể chạy ở chế độ cụm độc lập chỉ yêu cầu khung Apache Spark và JVM trên mỗi máy trong cụm. Tuy nhiên, sử dụng các công cụ quản lý cụm như trung gian giữa hai thành phần giúp tận dụng tài nguyên tốt hơn và cho phép phân bổ theo yêu cầu. Trong doanh nghiệp, Apache Spark có thể chạy trên Apache Mesos, Kubernetes và Docker Swarm.  
 Apache Spark xây dựng các lệnh xử lý dữ liệu của người dùng thành Đồ thị vòng có hướng hoặc DAG. DAG là lớp lập lịch của Apache Spark; nó xác định những tác vụ nào được thực thi trên những nút nào và theo trình tự nào.

**2.3. Delta Lake**

**Delta Lake** là lớp lưu trữ được tối ưu hóa cung cấp nền tảng để lưu trữ dữ liệu và bảng trong Nền tảng Databricks Lakehouse. Delta Lake là [phần mềm nguồn mở](https://delta.io/) mở rộng các tệp dữ liệu Parquet với nhật ký giao dịch dựa trên tệp cho [các giao dịch ACID](https://learn.microsoft.com/en-us/azure/databricks/lakehouse/acid) và xử lý siêu dữ liệu có thể mở rộng. Delta Lake hoàn toàn tương thích với API Apache Spark và được phát triển để tích hợp chặt chẽ với Truyền có cấu trúc, cho phép bạn dễ dàng sử dụng một bản sao dữ liệu cho cả hoạt động hàng loạt và truyền trực tuyến, đồng thời cung cấp quy trình xử lý gia tăng theo quy mô.

Delta Lake là định dạng lưu trữ mặc định cho mọi hoạt động trên Azure Databricks. Trừ khi có quy định khác, tất cả các bảng trên Azure Databricks đều là bảng Delta.

**2.4. Azure Databricks**

**Azure Databricks** là giải pháp phần mềm Big Data Analytics Software với chức năng và chi phí phù hợp cho các doanh nghiệp từ nhỏ và vừa (SMEs) tới các doanh nghiệp lớn. **[Phần mềm Azure Databricks](https://sotaydoanhtri.com/softwares/azure-databricks-8717/)** được đánh giá cao bởi cả người dùng lẫn chuyên gia trong lĩnh vực Database Software.SQL.

Databricks, dịch vụ phân tích dữ liệu lớn được thành lập bởi các nhà phát triển ban đầu của Apache Spark, hôm nay tuyên bố rằng họ sẽ khởi động dự án nguồn mở Delta Lake của mình để xây dựng các hồ dữ liệu cho Quỹ Linux và dưới sự quản trị mở Mô hình cung cấp. Công ty đã công bố ra mắt Delta Lake vào đầu năm nay. Mặc dù đây là một dự án tương đối mới, nhưng nó đã được nhiều tổ chức áp dụng và hỗ trợ bởi các công ty như Intel, Alibaba và Booz Allen Hamilton.

Theo mô hình này, các công ty nói rằng họ lấy tất cả dữ liệu của họ và lưu trữ nó trong các hồ dữ liệu, sau đó thực hiện một số thứ với dữ liệu đó, với máy học và khoa học dữ liệu là những thứ hiển nhiên. Nhưng họ cũng làm những việc truyền thống hơn với kho dữ liệu, chẳng hạn như thông tin kinh doanh và báo cáo. Thuật ngữ Ghodsi sử dụng cho loại sử dụng này là "Lake House". Databricks đang ngày càng thấy Spark được sử dụng cho mục đích này, không chỉ để thay thế Hadoop và thực hiện ETL (Trích xuất, Chuyển đổi, Tải). "Kiểu mẫu Lake House này đang tăng lên và chúng tôi muốn nhân đôi nó."

**CHƯƠNG 3 : THIẾT KẾ HỆ THỐNG**

## **3.1. Delta Architecture**

A picture containing graphical user interface

Description automatically generated

Diagram

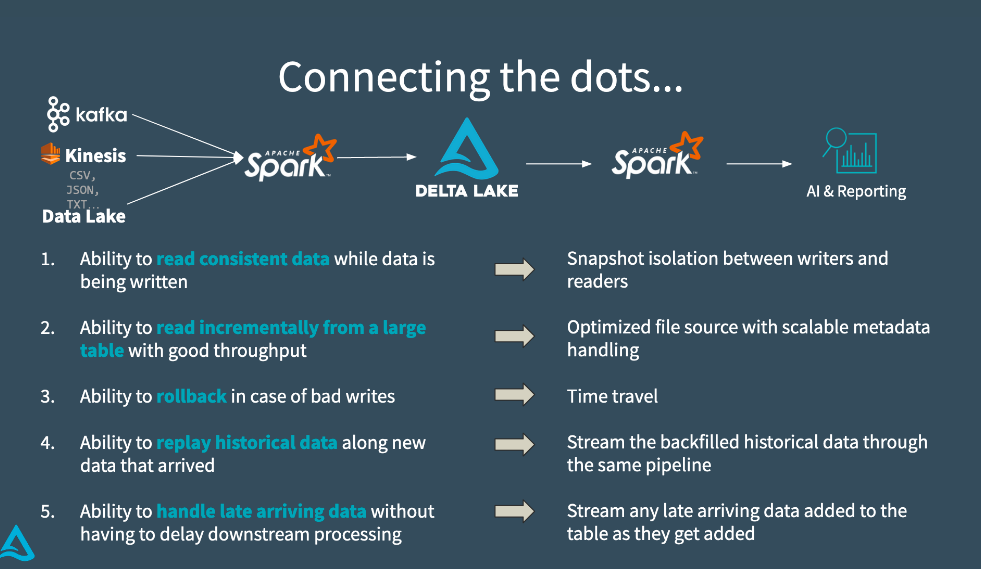
Description automatically generated

Text

Description automatically generated

A picture containing text

Description automatically generated



**3.2. Thiết kế cơ sở dữ liệu**

Table

Description automatically generated

Graphical user interface, table

Description automatically generated

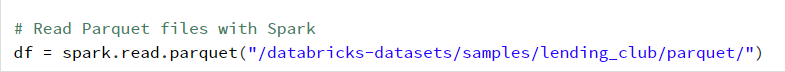
**CHƯƠNG 4 : XÂY DỰNG CHƯƠNG TRÌNH**

**4.1. Quá trình thực hiện và demo**

Đối với bản demo này, chúng tôi sẽ sử dụng bộ dữ liệu công khai về các khoản vay từ Lending Club. Nó bao gồm tất cả các khoản vay được tài trợ từ năm 2012 đến 2017. Mỗi khoản vay bao gồm thông tin nhân khẩu học do người nộp đơn cung cấp, cũng như tình trạng khoản vay hiện tại (Hiện tại, Trễ hạn, Đã thanh toán đầy đủ, v.v.) và thông tin thanh toán mới nhất. Để có cái nhìn đầy đủ về dữ liệu, vui lòng xem từ điển dữ liệu có sẵn tại đường dẫn dưới.

<https://www.kaggle.com/wendykan/lending-club-loan-data>

## **Import raw data and save it into a Delta Lake table**



Graphical user interface, text, application

Description automatically generated

Kết quả demo :

Table

Description automatically generated

**ETL - Filter Data and Fix Schema**

Text

Description automatically generated

## **Save our cleaned and conformed table as a Silver table in Delta Lake**

Graphical user interface, text, application

Description automatically generated

Kết quả demo :

Table

Description automatically generated

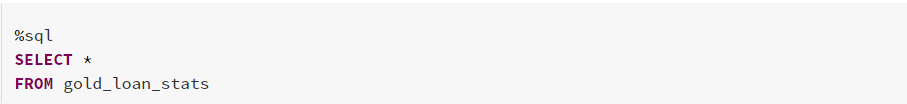
### **Create Gold table**

Hiện tại, bảng Silver của chúng ta đã được làm sạch, đồng thời chúng ta đã phát triển lược đồ, bước tiếp theo là tạo bảng Gold. Bảng Gold thường được tạo để cung cấp dữ liệu rõ ràng, đáng tin cậy cho một đơn vị kinh doanh hoặc trường hợp sử dụng cụ thể.

Trong trường hợp của chúng em, chúng em sẽ tạo một bảng Gold chỉ bao gồm 2 cột - addr\_state và count - để cung cấp chế độ xem tổng hợp cho dữ liệu của chúng em. Đối với mục đích của chúng em, bảng này sẽ cho phép chúng em hiển thị những gì Delta Lake có thể làm, nhưng trên thực tế, một bảng như thế này có thể được sử dụng để cung cấp báo cáo xuôi dòng hoặc công cụ BI cần dữ liệu được định dạng theo một cách rất cụ thể. Các bảng Silver thường cung cấp nhiều bảng Gold xuôi dòng.

Graphical user interface, text, application

Description automatically generated

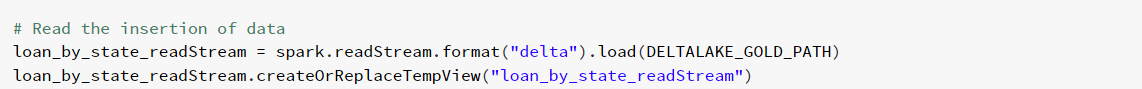


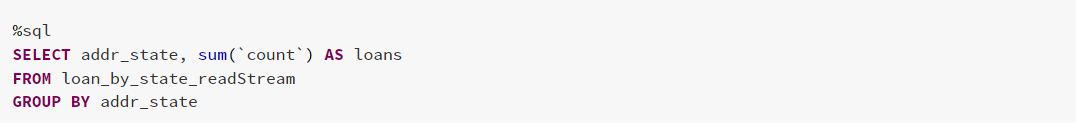
Kết quả demo :

Graphical user interface, application

Description automatically generated

## **Unified Batch and Streaming Source and Sink**





Kết quả demo :

Map

Description automatically generated

Text

Description automatically generated with low confidence

A picture containing graphical user interface

Description automatically generated

Kết quả demo :

A picture containing chart

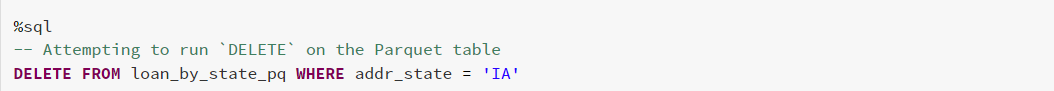
Description automatically generated

## **Full DML Support**

Graphical user interface, text, application

Description automatically generated

### **DELETE Support**





Kết quả demo :

A picture containing application

Description automatically generated

A picture containing chart

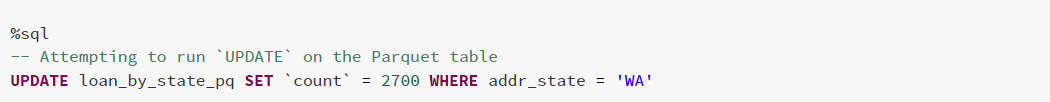
Description automatically generated

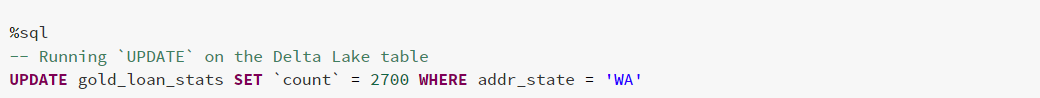
Kết quả demo :

Map

Description automatically generated

### **UPDATE Support**





Kết quả demo :

Background pattern

Description automatically generated

A picture containing logo

Description automatically generated

Kết quả demo :

Map

Description automatically generated

### **MERGE INTO Support**

#### *INSERT or UPDATE with Parquet: 7-step process*

Với đường dẫn dữ liệu cũ, để chèn hoặc cập nhật bảng, bạn phải:

1. Xác định các hàng mới được chèn
2. Xác định các hàng sẽ được thay thế (tức là đã cập nhật)
3. Xác định tất cả các hàng không bị ảnh hưởng bởi thao tác chèn hoặc cập nhật
4. Tạo một tạm thời mới dựa trên cả ba câu lệnh chèn
5. Xóa bảng gốc (và tất cả các tệp được liên kết đó)
6. "Đổi tên" bảng tạm thời trở lại tên bảng gốc
7. Thả bảng tạm thời

Shape, square

Description automatically generatedShape

Description automatically generated

A picture containing shape

Description automatically generated

Graphical user interface, application

Description automatically generated with medium confidenceDiagram

Description automatically generated

#### *INSERT or UPDATE with Delta Lake: 2-step process*

Graphical user interface, application

Description automatically generated

Kết quả demo :

Graphical user interface, text, application

Description automatically generated

A picture containing background pattern

Description automatically generated

Kết quả demo :

Graphical user interface, application, Word

Description automatically generated

A picture containing text

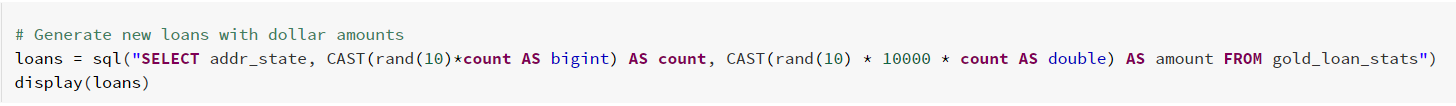
Description automatically generated

Kết quả demo :

Map

Description automatically generated

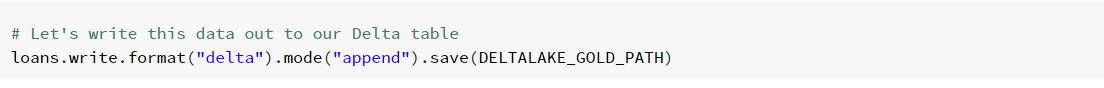
## **Schema Evolution**

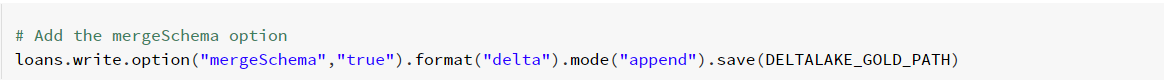


Kết quả demo :

Graphical user interface, application

Description automatically generated

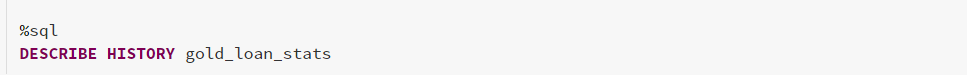




Graphical user interface, text

Description automatically generated with medium confidence

### **Review Delta Lake Table History**

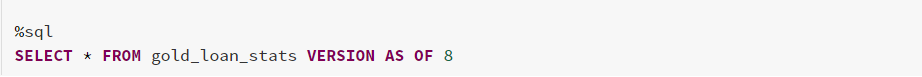


Kết quả demo :

Table

Description automatically generated

### **Time Travel via Version Number**



Kết quả demo :

Map

Description automatically generated



Kết quả demo :

Map

Description automatically generated

**KẾT LUẬN VÀ KIẾN NGHỊ**

Trong suốt quá trình hoàn thành đồ án cơ sở, nhóm chúng em đã rút ra được rất nhiều kinh nghiệm và bài học về quá trình thực hiện một dự án thực tế, được áp dụng những kiến thức đã học trong suốt những năm qua để có thể hoàn thành đồ án một cách tốt nhất. Trong quá trình thực hiện, nhóm chúng em không thể tránh khỏi những sai sót mong các thầy cô có thể thông cảm và giúp đỡ cho nhóm em.

Cuối cùng, nhóm em gồm Nguyễn Đức Mạnh, Dương Minh Hiếu, Phạm Thành Trung xin trân thành cảm ơn đến thầy giáo Tiến sĩ Trần Đăng Hoan là người đã trực tiếp hướng dẫn để nhóm em có thể hoàn thành được đề tài này. Chúng em cũng xin trân thành cảm ơn các thầy cô giáo khoa Công nghệ thông tin, trường Đại học Phenikaa đã tận tình giảng dạy những bài học, truyền đạt nhiều kinh nghiệm quý giá để chúng em có thể học hỏi và vận dụng vào cuộc sống.

***Chúng em xin trân thành cảm ơn!***

**TÀI LIỆU THAM KHẢO**

1. The Enterprise Big Data Lake Delivering the Promise of Big Data and Data Science
2. <https://aita.gov.vn/data-lake-xu-huong-cong-nghe-cho-du-lieu-lon->
3. <https://www.osam.io/post/data-lake-ho-du-lieu-la-gi>

**(Link Code :** <https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/5930262858117312/2479940380502877/2465630684816107/latest.html> **)**