Dân chủ hóa dữ liệu cho quyết định kịp thời với text-to-SQL tại Parcel Perform

**📖 Bài viết gốc:** https://aws.amazon.com/vi/blogs/machine-learning/democratize-data-for-timely-decisions-with-text-to-sql-at-parcel-perform/  
**👤 Tác giả:** Yudho Ahmad Diponegoro, Le Vy, và Jun Kai Loke  
**📅 Ngày xuất bản:** 09 tháng 7 năm 2025  
**🌐 Nguồn:** AWS Blog  
**👨‍💻 Người dịch:** Nguyễn Quang Bình  
**📅 Ngày dịch:** 08/07/2025  
**⏱️ Thời gian đọc:** 15 phút

──────────────────────────────────────────────────

## 📋 Tóm tắt

Bài viết này được đồng tác giả với Le Vy từ Parcel Perform.

**🎯 Đối tượng đọc:** Developers, Solution Architects, DevOps Engineers  
**📊 Độ khó:** Intermediate  
**🏷️ Tags:** AWS, Cloud Computing, Architecture

──────────────────────────────────────────────────

## 📚 Mục lục

• Giới thiệu  
• Kiến trúc giải pháp  
• Triển khai  
• Kết luận  
• Glossary - Thuật ngữ  
• Tài liệu tham khảo

──────────────────────────────────────────────────

Bài viết này được đồng tác giả với Le Vy từ Parcel Perform.

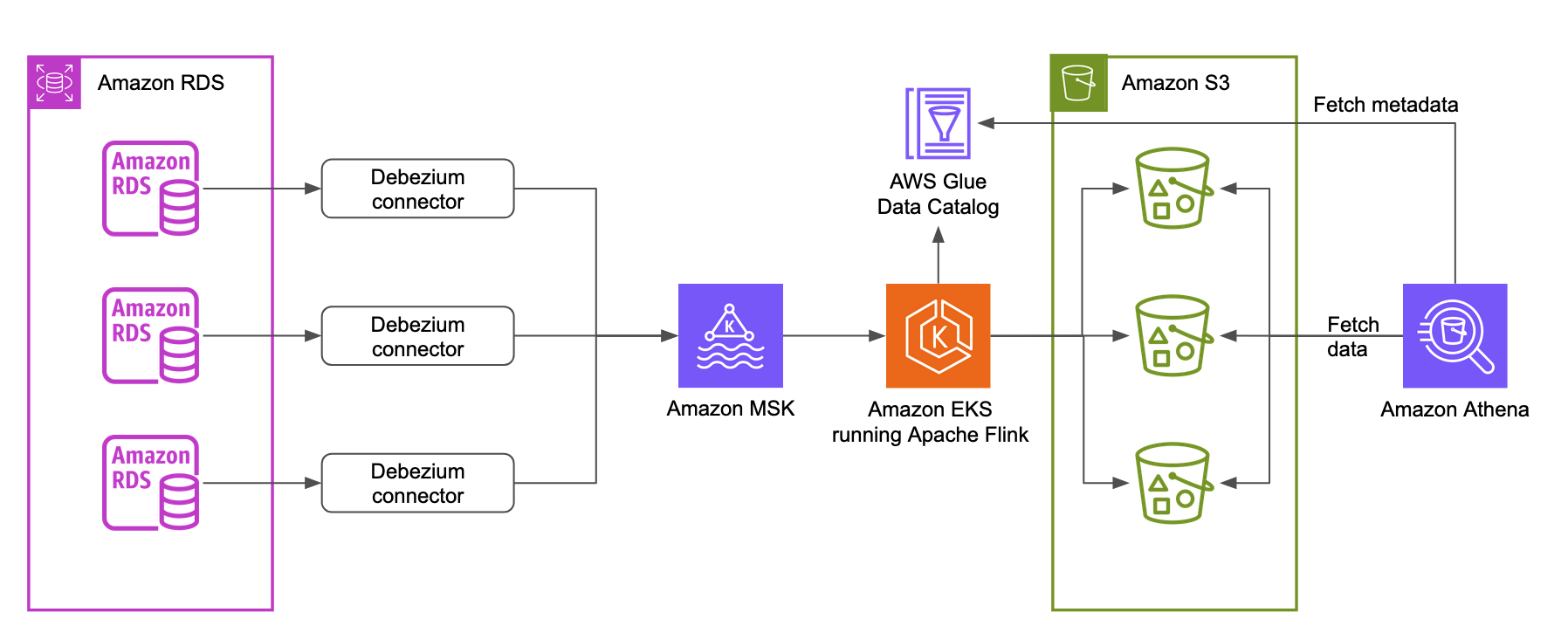
Quyền truy cập vào dữ liệu chính xác thường là yếu tố phân biệt thực sự của các quyết định xuất sắc và kịp thời. Điều này thậm chí còn quan trọng hơn đối với các quyết định và hành động hướng tới khách hàng. Một AI tiên tiến được triển khai đúng cách có thể giúp tổ chức của bạn đơn giản hóa quyền truy cập vào dữ liệu để ra quyết định chính xác và kịp thời cho đội ngũ kinh doanh hướng tới khách hàng, đồng thời giảm công việc nặng nhọc không có sự khác biệt được thực hiện bởi đội dữ liệu của bạn. Trong bài viết này, chúng tôi chia sẻ cách Parcel Perform, một Nền tảng Trải nghiệm Giao hàng AI hàng đầu cho các doanh nghiệp thương mại điện tử trên toàn thế giới, đã triển khai một giải pháp như vậy.

Theo dõi giao hàng sau mua hàng chính xác có thể rất quan trọng đối với nhiều thương gia thương mại điện tử. Parcel Perform cung cấp một hệ thống phần mềm dưới dạng dịch vụ (SaaS) và trải nghiệm dữ liệu và giao hàng từ đầu đến cuối thông minh được điều khiển bởi AI cho các thương gia thương mại điện tử. Hệ thống sử dụng các dịch vụ AWS và AI tiên tiến để xử lý hàng trăm triệu dữ liệu chuyển động giao hàng bưu kiện hàng ngày và cung cấp khả năng theo dõi thống nhất trên các nhà vận chuyển cho các thương gia, với sự nhấn mạnh vào độ chính xác và sự đơn giản.

Đội ngũ kinh doanh tại Parcel Perform thường cần quyền truy cập vào dữ liệu để trả lời các câu hỏi liên quan đến việc giao hàng bưu kiện của thương gia, chẳng hạn như "Chúng ta có thấy sự gia tăng đột biến trong việc chậm trễ giao hàng tuần trước không? Nếu có, điều này được quan sát thấy ở những cơ sở vận chuyển nào, và nguyên nhân chính của vấn đề là gì?" Trước đây, đội dữ liệu phải thủ công tạo truy vấn và chạy nó để lấy dữ liệu. Với khả năng text-to-SQL được hỗ trợ bởi AI tạo sinh mới trong Parcel Perform, đội ngũ kinh doanh có thể tự phục vụ nhu cầu dữ liệu của họ bằng cách sử dụng giao diện trợ lý AI. Trong bài viết này, chúng tôi thảo luận về cách Parcel Perform đã kết hợp AI tạo sinh, lưu trữ dữ liệu và truy cập dữ liệu thông qua các dịch vụ AWS để đưa ra quyết định kịp thời.

**Kiến trúc phân tích dữ liệu**

Giải pháp bắt đầu với việc thu thập, lưu trữ và truy cập dữ liệu. Parcel Perform đã áp dụng kiến trúc phân tích dữ liệu được hiển thị trong sơ đồ sau.



Một loại dữ liệu chính trong ứng dụng giám sát bưu kiện Parcel Perform là dữ liệu sự kiện bưu kiện, có thể đạt tới hàng tỷ hàng. Điều này bao gồm thay đổi trạng thái vận chuyển của bưu kiện, thay đổi vị trí và nhiều hơn nữa. Dữ liệu hàng ngày này từ nhiều đơn vị kinh doanh được lưu trữ trong các cơ sở dữ liệu quan hệ được lưu trữ trên Amazon Relational Database Service (Amazon RDS).

Mặc dù các cơ sở dữ liệu quan hệ phù hợp cho việc thu thập và tiêu thụ dữ liệu nhanh chóng từ ứng dụng, một ngăn xếp phân tích riêng biệt là cần thiết để xử lý phân tích theo cách có thể mở rộng và hiệu suất mà không làm gián đoạn ứng dụng chính. Những nhu cầu phân tích này bao gồm trả lời các truy vấn tổng hợp từ các câu hỏi như "Có bao nhiêu bưu kiện bị chậm trễ tuần trước?"

Parcel Perform sử dụng Amazon Simple Storage Service (Amazon S3) với một công cụ truy vấn được cung cấp bởi Amazon Athena để đáp ứng nhu cầu phân tích của họ. Với cách tiếp cận này, Parcel Perform được hưởng lợi từ lưu trữ hiệu quả về chi phí trong khi vẫn có thể chạy các truy vấn SQL khi cần trên dữ liệu thông qua Athena, được định giá theo sử dụng.

Dữ liệu trong Amazon S3 được lưu trữ ở định dạng dữ liệu Apache Iceberg cho phép cập nhật dữ liệu, điều này hữu ích trong trường hợp này vì các sự kiện bưu kiện đôi khi được cập nhật. Nó cũng hỗ trợ phân vùng để có hiệu suất tốt hơn. Amazon S3 Tables, được ra mắt vào cuối năm 2024, là một tính năng bảng Iceberg được quản lý cũng có thể là một tùy chọn cho bạn.

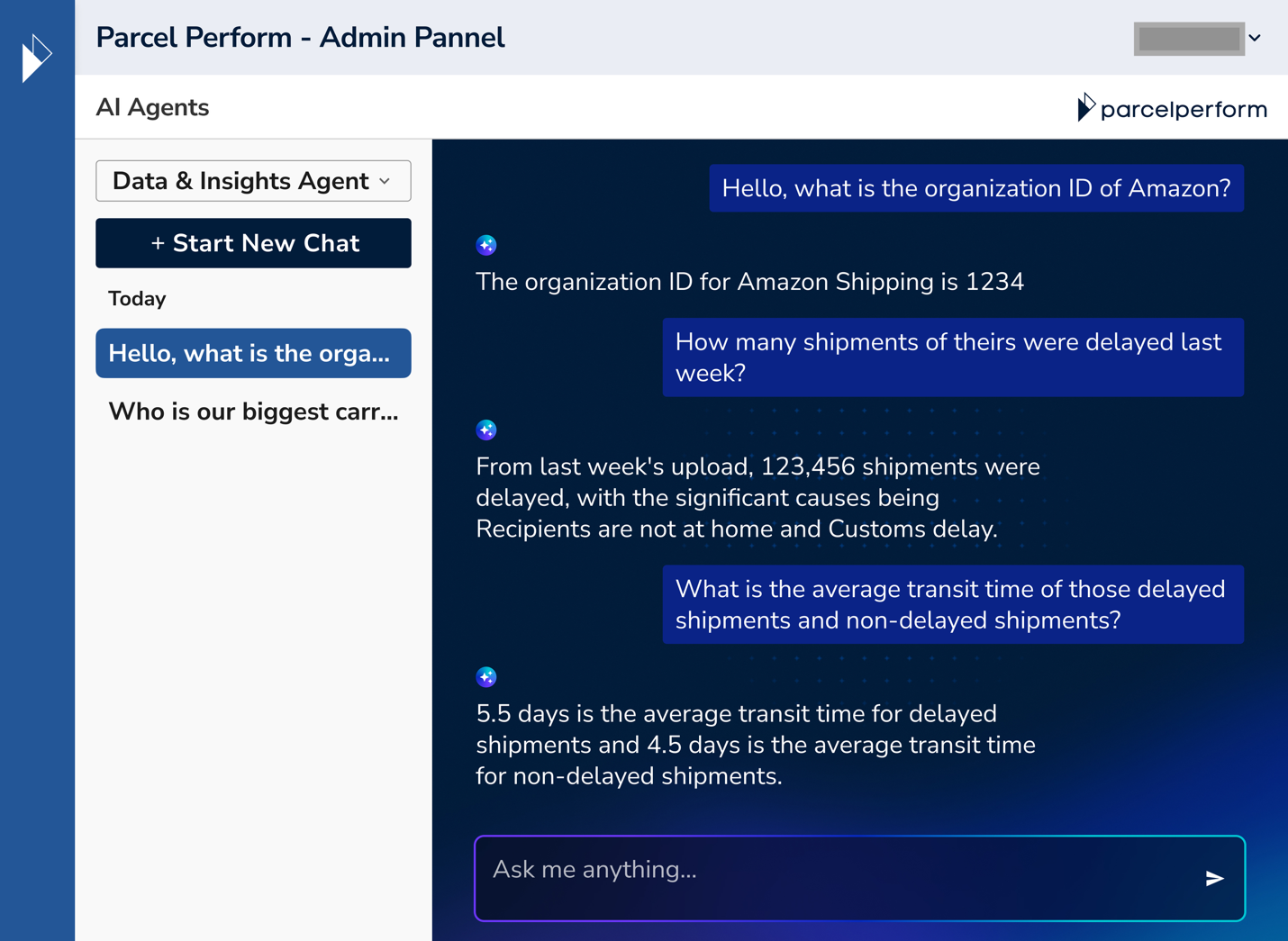
Parcel Perform sử dụng một cụm Apache Kafka được quản lý bởi Amazon Managed Streaming for Apache Kafka (Amazon MSK) như luồng để di chuyển dữ liệu từ nguồn đến bucket S3. Amazon MSK Connect với một connector Debezium truyền dữ liệu với change data capture (CDC) từ Amazon RDS đến Amazon MSK.

Apache Flink, chạy trên Amazon Elastic Kubernetes Service (Amazon EKS), xử lý các luồng dữ liệu từ Amazon MSK. Nó ghi dữ liệu này vào một bucket S3 theo định dạng Iceberg, và cập nhật lược đồ dữ liệu trong AWS Glue Data Catalog. Lược đồ dữ liệu cho phép Athena truy vấn đúng dữ liệu trong bucket S3.

Bây giờ bạn đã hiểu cách dữ liệu được thu thập và lưu trữ, hãy cho chúng tôi chỉ ra cách dữ liệu được tiêu thụ bằng cách sử dụng trợ lý phục vụ dữ liệu được hỗ trợ bởi AI tạo sinh cho các đội kinh doanh trong Parcel Perform.

**AI agent có thể truy vấn dữ liệu**

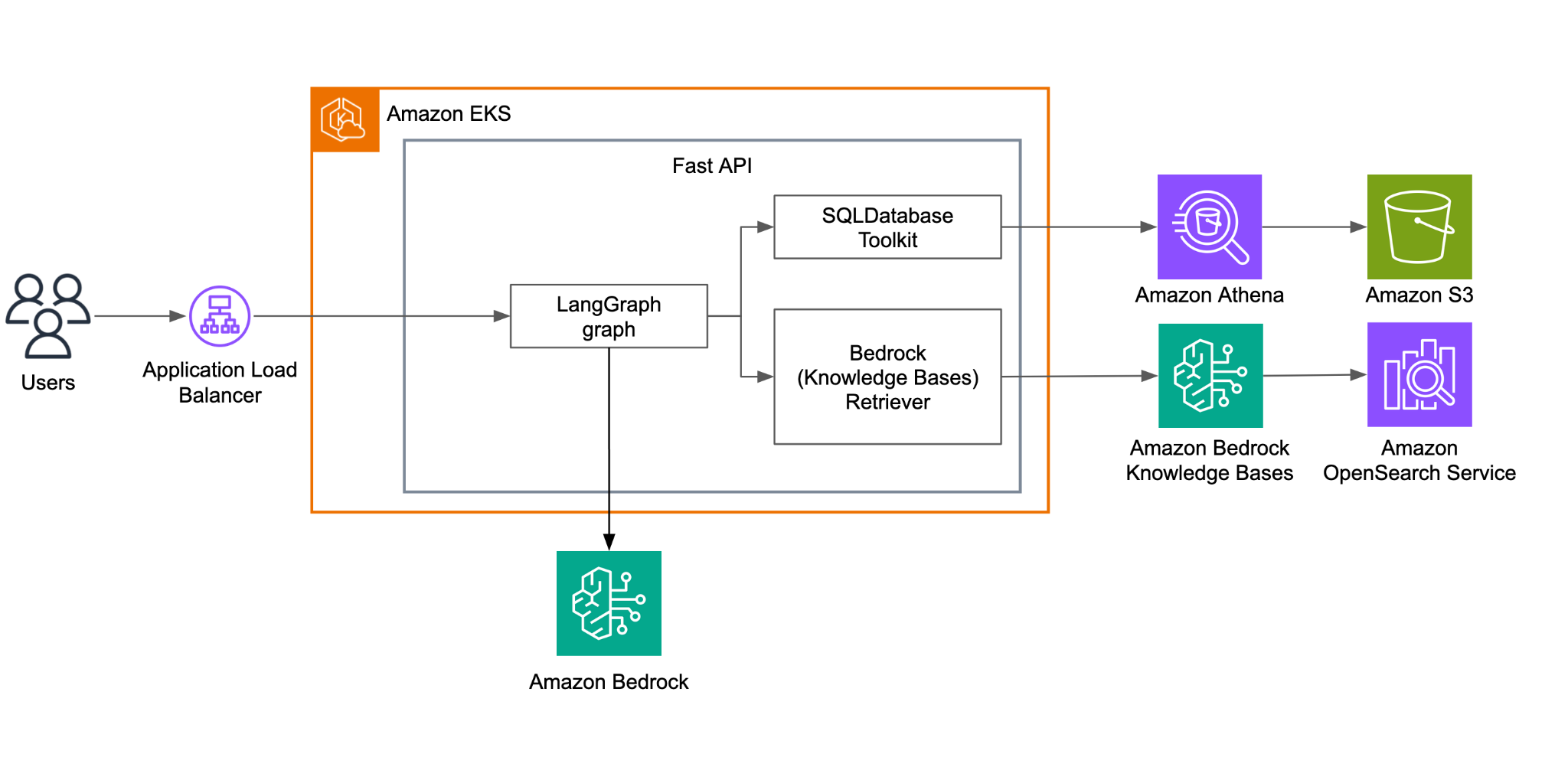
Người dùng của AI agent phục vụ dữ liệu trong Parcel Perform là các thành viên đội kinh doanh hướng tới khách hàng thường truy vấn dữ liệu sự kiện bưu kiện để trả lời các câu hỏi từ các thương gia thương mại điện tử về việc giao hàng bưu kiện và để hỗ trợ họ một cách chủ động. Ảnh chụp màn hình sau đây cho thấy trải nghiệm UI cho trợ lý AI agent, được hỗ trợ bởi text-to-SQL với AI tạo sinh.



Chức năng này đã giúp đội Parcel Perform và khách hàng của họ tiết kiệm thời gian, điều mà chúng tôi thảo luận sau trong bài viết này. Trong phần sau, chúng tôi trình bày kiến trúc hỗ trợ tính năng này.

**Kiến trúc AI agent Text-to-SQL**

Kiến trúc trợ lý AI phục vụ dữ liệu trong Parcel Perform được hiển thị trong sơ đồ sau.



UI trợ lý AI được hỗ trợ bởi một ứng dụng được xây dựng với khung Fast API được lưu trữ trên Amazon EKS. Nó cũng được đặt phía trước bởi một Application Load Balancer để cho phép khả năng mở rộng ngang tiềm năng.

Ứng dụng sử dụng LangGraph để điều phối quy trình làm việc của các lời gọi mô hình ngôn ngữ lớn (LLM), việc sử dụng công cụ và checkpoint bộ nhớ. Đồ thị sử dụng nhiều công cụ, bao gồm những công cụ từ SQLDatabase Toolkit, để tự động lấy lược đồ dữ liệu thông qua Athena. Đồ thị cũng sử dụng một retriever Amazon Bedrock Knowledge Bases để truy xuất thông tin kinh doanh từ một cơ sở kiến thức. Parcel Perform sử dụng các mô hình Claude của Anthropic trong Amazon Bedrock để tạo SQL.

Mặc dù chức năng của Athena như một công cụ truy vấn để truy vấn dữ liệu sự kiện bưu kiện trên Amazon S3 là rõ ràng, Parcel Perform vẫn cần một cơ sở kiến thức. Trong trường hợp sử dụng này, việc tạo SQL hoạt động tốt hơn khi LLM có nhiều thông tin bối cảnh kinh doanh hơn để giúp diễn giải các trường cơ sở dữ liệu và dịch thuật ngữ logistics thành các biểu diễn dữ liệu. Điều này được minh họa tốt hơn với hai ví dụ sau:

Các hoạt động data lake của Parcel Perform sử dụng các mã cụ thể: c cho create và u cho update. Khi phân tích dữ liệu, Parcel Perform đôi khi cần tập trung chỉ vào các bản ghi tạo ban đầu, nơi mã hoạt động bằng c. Vì logic kinh doanh này có thể không vốn có trong việc đào tạo LLM nói chung, Parcel Perform định nghĩa rõ ràng điều này trong bối cảnh kinh doanh của họ.

Trong thuật ngữ logistics, thời gian vận chuyển có các quy ước ngành cụ thể. Nó được đo bằng ngày, và việc giao hàng trong ngày được ghi lại là transit\_time = 0. Mặc dù điều này trực quan đối với các chuyên gia logistics, một LLM có thể diễn giải sai một yêu cầu như "Lấy cho tôi tất cả các lô hàng với giao hàng trong ngày" bằng cách sử dụng WHERE transit\_time = 1 thay vì WHERE transit\_time = 0 trong câu lệnh SQL được tạo.

Do đó, mỗi câu hỏi đến đi đến một quy trình làm việc Retrieval Augmented Generation (RAG) để tìm thông tin kinh doanh được lưu trữ có thể liên quan, để làm phong phú bối cảnh. Cơ chế này giúp cung cấp các quy tắc và diễn giải cụ thể mà ngay cả các LLM tiên tiến có thể không thể rút ra từ dữ liệu đào tạo chung.

Parcel Perform sử dụng Amazon Bedrock Knowledge Bases như một giải pháp được quản lý cho quy trình làm việc RAG. Họ thu thập thông tin bối cảnh kinh doanh bằng cách tải lên các tệp lên Amazon S3. Amazon Bedrock Knowledge Bases xử lý các tệp, chuyển đổi chúng thành các chunk, sử dụng các mô hình embedding để tạo vector, và lưu trữ các vector trong một cơ sở dữ liệu vector để làm cho chúng có thể tìm kiếm được. Các bước được quản lý hoàn toàn bởi Amazon Bedrock Knowledge Bases. Parcel Perform lưu trữ các vector trong Amazon OpenSearch Serverless như cơ sở dữ liệu vector lựa chọn để đơn giản hóa quản lý cơ sở hạ tầng.

Amazon Bedrock Knowledge Bases cung cấp Retrieve API, nhận một đầu vào (như một câu hỏi từ trợ lý AI), chuyển đổi nó thành một vector embedding, tìm kiếm các chunk thông tin bối cảnh kinh doanh liên quan trong cơ sở dữ liệu vector, và trả về các chunk tài liệu liên quan hàng đầu. Nó được tích hợp với LangChain Amazon Bedrock Knowledge Bases retriever bằng cách gọi phương thức invoke.

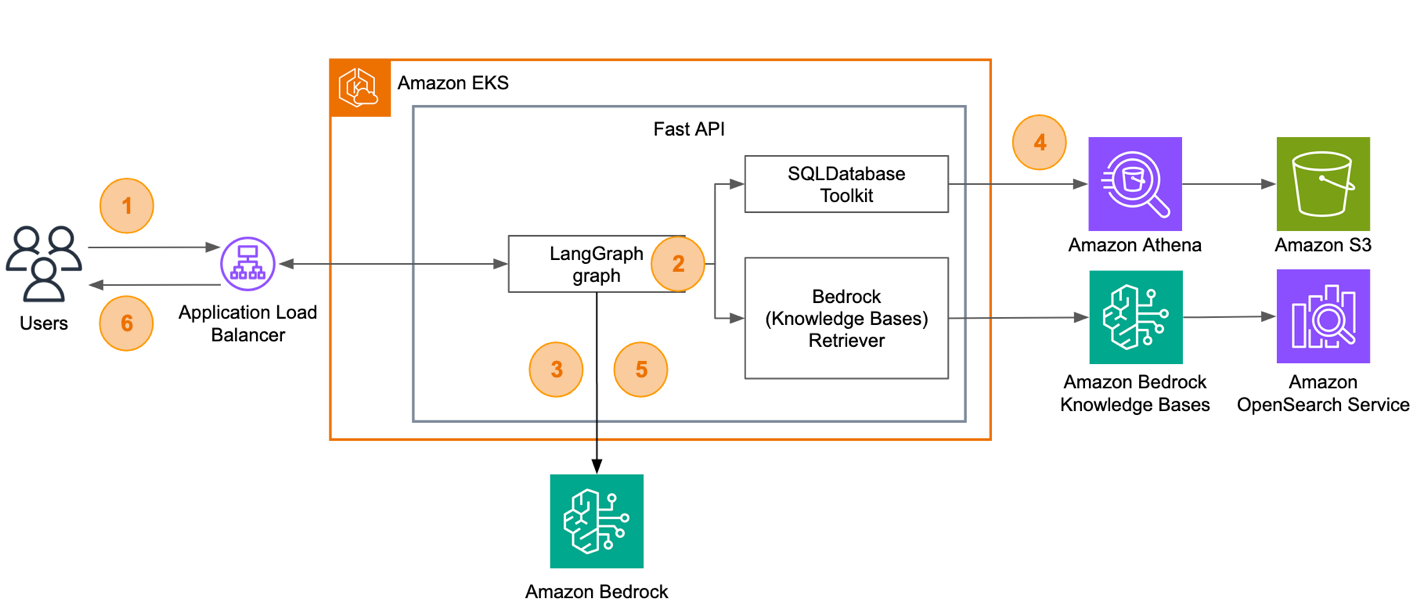
Bước tiếp theo liên quan đến việc gọi một AI agent với thông tin bối cảnh kinh doanh được cung cấp và prompt tạo SQL. Prompt được lấy cảm hứng từ một prompt trong LangChain Hub. Sau đây là một đoạn mã của prompt:

*Bạn là một agent được thiết kế để tương tác với một cơ sở dữ liệu SQL.  
Được đưa ra một câu hỏi đầu vào, tạo một truy vấn {dialect} đúng cú pháp để chạy, sau đó nhìn vào kết quả của truy vấn và trả về câu trả lời.  
Trừ khi người dùng chỉ định một số lượng cụ thể các ví dụ họ muốn có được, luôn giới hạn truy vấn của bạn tối đa {top\_k} kết quả.  
Bối cảnh liên quan:  
{rag\_context}  
Bạn có thể sắp xếp kết quả theo một cột liên quan để trả về các ví dụ thú vị nhất trong cơ sở dữ liệu.  
Không bao giờ truy vấn tất cả các cột từ một bảng cụ thể, chỉ yêu cầu các cột liên quan được đưa ra câu hỏi.  
Bạn có quyền truy cập vào các công cụ để tương tác với cơ sở dữ liệu.  
- Chỉ sử dụng các công cụ dưới đây. Chỉ sử dụng thông tin được trả về bởi các công cụ dưới đây để xây dựng câu trả lời cuối cùng của bạn.  
- KHÔNG thực hiện bất kỳ câu lệnh DML nào (INSERT, UPDATE, DELETE, DROP, v.v.) đối với cơ sở dữ liệu.  
- Để bắt đầu truy vấn cho câu trả lời cuối cùng, bạn nên LUÔN nhìn vào các bảng trong cơ sở dữ liệu để xem những gì bạn có thể truy vấn. KHÔNG bỏ qua bước này.  
- Sau đó bạn nên truy vấn lược đồ của các bảng liên quan nhất*

Mẫu prompt là một phần của hướng dẫn ban đầu cho agent. Lược đồ dữ liệu được tự động chèn bởi các công cụ từ SQLDatabase Toolkit ở một bước sau của quy trình làm việc agentic này. Các bước sau xảy ra sau khi người dùng nhập một câu hỏi trong UI trợ lý AI:

1. Câu hỏi kích hoạt một lần chạy của đồ thị LangGraph.  
2. Các quy trình sau xảy ra song song:  
 • Đồ thị lấy lược đồ cơ sở dữ liệu từ Athena thông qua SQLDatabase Toolkit.  
 • Đồ thị chuyển câu hỏi đến Amazon Bedrock Knowledge Bases retriever và nhận một danh sách thông tin kinh doanh liên quan về câu hỏi.  
3. Đồ thị gọi một LLM sử dụng Amazon Bedrock bằng cách chuyển câu hỏi, bối cảnh cuộc trò chuyện, lược đồ dữ liệu và thông tin bối cảnh kinh doanh. Kết quả là SQL được tạo.  
4. Đồ thị sử dụng SQLDatabase Toolkit một lần nữa để chạy SQL thông qua Athena và lấy đầu ra dữ liệu.  
5. Đầu ra dữ liệu được chuyển vào một LLM để tạo phản hồi cuối cùng dựa trên câu hỏi ban đầu được hỏi. Amazon Bedrock Guardrails được sử dụng như một biện pháp bảo vệ để tránh các đầu vào và phản hồi không phù hợp.  
6. Phản hồi cuối cùng được trả về cho người dùng thông qua UI trợ lý AI.

Sơ đồ sau minh họa các bước này.



Việc triển khai này chứng minh cách Parcel Perform chuyển đổi các câu hỏi thô thành dữ liệu có thể hành động để ra quyết định kịp thời. Bảo mật cũng được triển khai trong nhiều thành phần. Từ góc độ mạng, các pod EKS được đặt trong các subnet riêng tư trong Amazon Virtual Private Cloud (Amazon VPC) để cải thiện bảo mật mạng của ứng dụng trợ lý AI. AI agent này được đặt phía sau một lớp backend yêu cầu xác thực. Đối với bảo mật dữ liệu, dữ liệu nhạy cảm được che giấu khi nghỉ trong bucket S3. Parcel Perform cũng giới hạn quyền của vai trò AWS Identity and Access Management (IAM) được sử dụng để truy cập bucket S3 để nó chỉ có thể truy cập các bảng nhất định.

Trong các phần sau, chúng tôi thảo luận về cách tiếp cận của Parcel Perform để xây dựng giải pháp chuyển đổi dữ liệu này.

**Từ ý tưởng đến sản xuất**

Parcel Perform bắt đầu với ý tưởng giải phóng đội dữ liệu của họ khỏi việc phục vụ thủ công yêu cầu từ đội kinh doanh, đồng thời cải thiện tính kịp thời của tính khả dụng dữ liệu để hỗ trợ việc ra quyết định của đội kinh doanh.

Với sự giúp đỡ của đội AWS Solutions Architect, Parcel Perform đã hoàn thành một bằng chứng khái niệm sử dụng các dịch vụ AWS và một Jupyter notebook trong Amazon SageMaker Studio. Sau một thành công ban đầu, Parcel Perform đã tích hợp giải pháp với công cụ điều phối lựa chọn của họ, LangGraph.

Trước khi đưa vào sản xuất, Parcel Perform đã tiến hành thử nghiệm rộng rãi để xác minh kết quả nhất quán. Họ đã thêm LangSmith Tracing để ghi lại các bước và kết quả của AI agent để đánh giá hiệu suất của nó.

Đội Parcel Perform đã khám phá ra những thách thức trong hành trình của họ, mà chúng tôi thảo luận trong phần sau. Họ đã thực hiện kỹ thuật prompt để giải quyết những thách thức đó. Cuối cùng, AI agent đã được tích hợp vào sản xuất để được sử dụng bởi đội kinh doanh. Sau đó, Parcel Perform đã thu thập phản hồi người dùng nội bộ và giám sát nhật ký từ LangSmith Tracing để xác minh hiệu suất được duy trì.

**Những thách thức**

Hành trình này không tránh khỏi thách thức. Thứ nhất, một số thương gia thương mại điện tử có thể có một số bản ghi trong data lake dưới các tên khác nhau. Ví dụ, một thương gia có tên "ABC" có thể có nhiều bản ghi như "ABC Singapore Holdings Pte. Ltd.," "ABC Demo Account," "ABC Test Group," và vân vân. Đối với một câu hỏi như "Có bất kỳ chậm trễ vận chuyển bưu kiện nào bởi ABC tuần trước không?", SQL được tạo có yếu tố WHERE merchant\_name LIKE '%ABC%', có thể dẫn đến sự mơ hồ. Trong giai đoạn bằng chứng khái niệm, vấn đề này đã gây ra việc khớp không chính xác của kết quả.

Đối với thách thức này, Parcel Perform dựa vào kỹ thuật prompt cẩn thận để hướng dẫn LLM xác định khi tên có thể mơ hồ. AI agent sau đó gọi Athena một lần nữa để tìm kiếm các tên khớp. LLM quyết định tên thương gia nào để sử dụng dựa trên nhiều yếu tố, bao gồm tầm quan trọng trong đóng góp khối lượng dữ liệu và trạng thái tài khoản trong data lake. Trong tương lai, Parcel Perform dự định triển khai một kỹ thuật tinh vi hơn bằng cách nhắc người dùng giải quyết sự mơ hồ.

Thách thức thứ hai là về các câu hỏi không hạn chế có thể tạo ra các truy vấn đắt đỏ chạy trên lượng lớn dữ liệu và dẫn đến thời gian chờ truy vấn dài hơn. Một số câu hỏi này có thể không có mệnh đề LIMIT được áp đặt trong truy vấn. Để giải quyết điều này, Parcel Perform hướng dẫn LLM thêm mệnh đề LIMIT với một số lượng kết quả tối đa nhất định nếu người dùng không chỉ định số lượng kết quả dự định. Trong tương lai, Parcel Perform có kế hoạch sử dụng kết quả EXPLAIN truy vấn để xác định các truy vấn nặng.

Thách thức thứ ba liên quan đến việc theo dõi sử dụng và chi phí phát sinh của giải pháp cụ thể này. Đã bắt đầu nhiều dự án AI tạo sinh sử dụng Amazon Bedrock và đôi khi với cùng một LLM ID, Parcel Perform phải phân biệt sử dụng phát sinh bởi các dự án. Parcel Perform tạo một hồ sơ suy luận cho mỗi dự án, liên kết hồ sơ với các thẻ, và bao gồm hồ sơ đó trong mỗi lời gọi LLM cho dự án đó. Với thiết lập này, Parcel Perform có thể tách biệt chi phí dựa trên các dự án để cải thiện khả năng hiển thị và giám sát chi phí.

**Tác động**

Để trích xuất dữ liệu, đội kinh doanh làm rõ chi tiết với đội dữ liệu, đưa ra yêu cầu, kiểm tra tính khả thi và chờ đợi băng thông. Quy trình này kéo dài khi yêu cầu đến từ khách hàng hoặc các đội ở múi giờ khác nhau, với mỗi làm rõ thêm 12-24 giờ do giao tiếp không đồng bộ. Các yêu cầu đơn giản hơn được thực hiện sớm trong ngày làm việc có thể hoàn thành trong vòng 24 giờ, trong khi các yêu cầu phức tạp hơn hoặc những yêu cầu trong thời gian bận rộn có thể mất 3-5 ngày làm việc.

Với AI agent text-to-SQL, quy trình này được sắp xếp hợp lý một cách đáng kể—giảm thiểu giao tiếp qua lại để làm rõ yêu cầu, loại bỏ sự phụ thuộc vào băng thông đội dữ liệu, và tự động hóa diễn giải kết quả.

Các phép đo của Parcel Perform cho thấy AI agent text-to-SQL giảm thời gian trung bình để có hiểu biết 99%, từ 2,3 ngày xuống trung bình 10 phút, tiết kiệm khoảng 3.850 tổng giờ chờ đợi mỗi tháng trên các người yêu cầu trong khi duy trì độ chính xác dữ liệu.

Người dùng có thể trực tiếp truy vấn dữ liệu mà không cần trung gian, nhận kết quả trong vài phút thay vì vài ngày. Các đội trên các múi giờ giờ có thể truy cập hiểu biết bất kỳ lúc nào trong ngày, giảm bớt sự chậm trễ bực bội "chờ đến khi châu Á thức dậy" hoặc "bắt EMEA trước khi họ rời đi", dẫn đến khách hàng hạnh phúc hơn và giải quyết vấn đề nhanh hơn.

Sự chuyển đổi này đã tác động sâu sắc đến năng lực và trọng tâm của đội phân tích dữ liệu, giải phóng đội dữ liệu cho công việc chiến lược hơn và giúp mọi người đưa ra quyết định nhanh hơn, có thông tin hơn. Trước đây, các nhà phân tích dành khoảng 25% thời gian làm việc của họ để xử lý các yêu cầu trích xuất dữ liệu thường xuyên—tương đương với hơn 260 giờ hàng tháng trên toàn đội. Bây giờ, với các truy vấn cơ bản và trung cấp được tự động hóa, con số này đã giảm xuống chỉ 10%, giải phóng gần 160 giờ mỗi tháng cho công việc có tác động cao. Các nhà phân tích giờ tập trung vào phân tích dữ liệu phức tạp thay vì dành thời gian cho các nhiệm vụ truy xuất dữ liệu cơ bản.

**Kết luận**

Giải pháp của Parcel Perform chứng minh cách bạn có thể sử dụng AI tạo sinh để tăng cường năng suất và trải nghiệm khách hàng. Parcel Perform đã xây dựng một AI agent text-to-SQL chuyển đổi câu hỏi của đội kinh doanh thành SQL có thể lấy dữ liệu thực tế. Điều này cải thiện tính kịp thời của tính khả dụng dữ liệu cho việc ra quyết định liên quan đến khách hàng. Hơn nữa, đội dữ liệu có thể tránh công việc nặng nhọc không có sự khác biệt để tập trung vào các nhiệm vụ phân tích dữ liệu phức tạp.

Giải pháp này sử dụng nhiều dịch vụ AWS như Amazon Bedrock và các công cụ như LangGraph. Bạn có thể bắt đầu với một bằng chứng khái niệm và tham khảo AWS Solutions Architect của bạn hoặc tham gia với AWS Partners. Nếu bạn có câu hỏi, hãy đăng chúng trên AWS re:Post. Bạn cũng có thể làm cho việc phát triển trở nên đơn giản hơn với sự giúp đỡ của Amazon Q Developer. Khi bạn đối mặt với thách thức, bạn có thể lặp lại để tìm giải pháp, có thể bao gồm kỹ thuật prompt hoặc thêm các bước bổ sung vào quy trình làm việc của bạn.

──────────────────────────────────────────────────

## 📖 Glossary - Thuật ngữ

|  |  |  |
| --- | --- | --- |
| English | Tiếng Việt | Định nghĩa |
| Auto Scaling | Tự động mở rộng quy mô | Khả năng tự động tăng/giảm resources dựa trên demand |
| Load Balancer | Bộ cân bằng tải | Phân phối traffic đến multiple servers |
| Microservices | Kiến trúc microservices | Architectural pattern chia application thành small services |

## 🔗 Tài liệu tham khảo

**Tài liệu gốc**

• AWS Blog: https://aws.amazon.com/vi/blogs/machine-learning/democratize-data-for-timely-decisions-with-text-to-sql-at-parcel-perform/

• Related Articles: https://aws.amazon.com/vi/blogs/machine-learning/build-a-robust-text-to-sql-solution-generating-complex-queries-self-correcting-and-querying-diverse-data-sources/

──────────────────────────────────────────────────

## 💬 Ghi chú của người dịch

**Challenges trong quá trình dịch**

• Technical Terms: Giữ nguyên các thuật ngữ AWS để đảm bảo tính chính xác

• Cultural Context: Điều chỉnh ví dụ cho phù hợp với bối cảnh Việt Nam

• Complex Concepts: Giải thích rõ ràng các khái niệm phức tạp

**Insights gained**

• Technical Learning: Hiểu sâu hơn về kiến trúc AWS và best practices

• Language Skills: Cải thiện kỹ năng dịch thuật kỹ thuật

• Industry Knowledge: Nắm bắt xu hướng công nghệ mới

──────────────────────────────────────────────────

## 🤝 Đóng góp và Feedback

Bài dịch này được thực hiện trong khuôn khổ FCJ Internship Program.

**📧 Liên hệ:** ngqbinh456@gmail.com.  
**💬 Feedback:** Mọi góp ý để cải thiện chất lượng dịch thuật xin gửi về email trên  
**🔄 Updates:** Bài dịch sẽ được cập nhật dựa trên feedback từ cộng đồng

──────────────────────────────────────────────────

© 2024 - Bản dịch thuộc về Nguyễn Quang Bình. Vui lòng credit khi sử dụng.