Câu 2:  
Phương án 1: Sử dụng API trực tiếp và viết ETL code và sử dụng cron hoặc airflow để lập lịch cho task chạy hằng giờ

1. Workload:

+ Phương án này đòi hỏi một lượng công việc lớn để viết và duy trì ETL code.

+ Cần phải thường xuyên cập nhật mã nguồn để đảm bảo tính ổn định và hiệu suất của hệ thống. Bất kỳ thay đổi nào trong API của Keepa hoặc trong cấu trúc dữ liệu có thể yêu cầu sự can thiệp và điều chỉnh từ phía nhóm phát triển.

1. Chi phí:

+ Phương án này tốn chi phí nếu sử dụng các dịch vụ đám mây như EC2 (AWS), Google Compute Engine hoặc Azure để chạy các tác vụ ETL và lập lịch. Chi phí phát triển ban đầu cũng cao do yêu cầu phải xây dựng một hệ thống kết nối API độc lập.

+ Chi phí sử dụng dịch vụ Keepa là 99$ mỗi tháng cho gói 60 token mỗi phút. Mỗi request cho một sản phẩm sẽ tốn 1 token, vì vậy chi phí sử dụng Keepa cũng phải được tính toán.

1. Độ chính xác:

+ Phương án này cung cấp dữ liệu trực tiếp từ nguồn, giúp đảm bảo độ chính xác cao của dữ liệu. Tuy nhiên, độ chính xác cuối cùng cũng phụ thuộc vào cách thức triển khai ETL và xử lý dữ liệu.

Phương án 2: Sử dụng Airbyte và dbt:

1. Workload:

+ Sử dụng Airbyte và dbt giảm thiểu công việc phát triển và duy trì mã nguồn. Airbyte là một nền tảng tích hợp dữ liệu mã nguồn mở với khả năng sử dụng no-code, low-code để định nghĩa các luồng dữ liệu nhanh chóng.

+ Dbt là một công cụ transform dữ liệu mã nguồn mở, giúp xây dựng quy trình transform dữ liệu trực tiếp trên database, và những tính năng để test chất lượng của dữ liệu.

1. Chi phí:

+ Tương tự như phương án trước, phương án này cũng tốn chi phí cho việc sử dụng các dịch vụ đám mây và dịch vụ Keepa để request dữ liệu.

+ Chi phí có thể thấp hơn so với phương án 1 ít phải chi trả cho việc triển khai và duy trì mã nguồn ETL.

1. Độ chính xác:

+ Lấy dữ liệu trực tiếp từ Api có nhiều tính năng để test độ chất lượng của dữ liệu

Câu 3: Đây là schema mà em nghĩ sẽ dùng hiệu quả cho việc phân tích hiệu quả công việc của nhân viên

A black screen with white squares

Description automatically generated