**Tài liệu dự án tuyển cộng tác viên**

**Câu hỏi 1:** Mày muốn gì? Dự án này mày muốn làm nó với mục đích gì?

* TL: Tao thật sự không biết nữa! Chán quá! Không code thật chán! Tao chỉ muốn có một dự án để biết thứ giỏi nhất của mình không phế!

**Câu hỏi 2:** Vậy thì code đi. Nói tao nghe: dự án này mày muốn tập trung vào gì?

* TL: Tao muốn tập trung vào blog, vào chia sẻ, vào tìm kiếm.

Có nghĩ là mày muốn dự án tuyển cộng tác viên như 1 nơi chia sẻ niềm vui từ thiện, các dự án tốt, hay, được hỗ trợ bằng một hệ thống tìm kiếm theo **tag** chuẩn xác và nhanh chóng.

Mày có biết **nội dung** của một Blog: không chỉ có **chuỗi String** mà còn có **hình ảnh**, biểu tượng cảm xúc (sĩ nhiên là cần cái này, nếu không blog chỉ có chữ và chữ khá khô khan), bảng. Và còn có thể có biểu thức toán học và hình học.

* Dự án web tuyển cộng tác viên không tập trung vào sơ đồ dữ liệu, sơ đồ mối quan hệ trong database mà tập trung vào cấu trúc dữ liệu (cụ thể là cấu trúc nội dung của blog trong bảng Blog).
* Ta khuyên mày nên sử dụng elastichsearch để xây dựng công cụ tìm kiếm.

(Ý tưởng ngày: 09/11/2023 tạm thời là vậy đã)

11/11/2023 – làm chức năng của người dùng thông thường được trình bày trong file ‘v1\_database’.

# Chức năng thông báo:

## Giai đoạn 1 (24/11/2023): chức năng thông báo khi có bình luận trong bài viết

**Ý tưởng:** Mọi thông tin đều phải lưu trữ trong database kể cả số lượng thông báo chưa đọc. Redis sẽ lưu cache thông báo chưa đọc để tăng tốc độ truy vấn:  
Vậy ta sẽ có sơ đồ như sau:

Database

BE

Redis

request

Lấy số lượng thông báo

Gửi trả số lượng thông báo chưa đọc

Lưu xuống redis

**Câu hỏi 1:** Vậy Service nào của BE sẽ đóng vai trò cập nhật số lượng thông báo chưa đọc?

**Câu hỏi 2:** Cấu trúc bảng Comments hiện tại như sau:

Id | user\_id | blog\_id | commentLevel| comment| comment\_parent\_id

Làm sao mà ta biết được comment nào chưa đọc? và comment như nào thì thông báo? Comment trả lời 1 user cụ thể và comment mới xuất hiện trong bài viết cho tác giả à?

**Trả lời 1:** Để trả lời câu hỏi này thì rất khó. Ta có 2 kiểu cập nhật số lượng thông báo chưa đọc:

1. Khi sự kiện tạo thông báo xảy ra: ta + 1 trực tiếp vào số lượng thông báo chưa đọc.
2. Khi sự kiện tạo thông báo xảy ra: chèn thông báo vào bảng Notifications. Sau đó, ta lấy ra số lượng thông báo chưa đọc ở bảng Notifications và cập nhật lại số lượng thông báo chưa đọc.

Cách 1: cho phép giảm tải áp lực lên hệ thống vì không phải lấy nhiều dữ liệu, nhưng độ chính xác không cao và mức độ tập trung code kém do sự kiện dẫn đến tạo thông báo có thể phát sinh ở nhiều service.

Cách 2: thay vì để cho nhiều service cùng làm việc cập nhật thông báo chưa đọc, ta tập trung code ở 1 service **lắng nghe** các thay đổi trên các bảng trong database.

**Trả lời 2:** bảng Comment thì có liên quan gì. Ta cần tạo một bảng mới là Notification để ghi chép các thông báo và bảng Comment cũng **không cần** bổ sung thêm trường ‘**đọc nội dung hay chưa?**’vì chỉ cần người dùng bấm xem danh sách thông báo là **auto đánh dấu** phần thông báo câu trả lời đã được đọc như **youtube**.

Vậy thì làm sao thằng service lắng nghe các thay đổi trên các bảng trong database biết được người dùng đã lấy danh sách thông báo. Không lẽ lại làm thêm cho chức năng này phần cập nhật thông báo 🡪 có 2 nơi cập nhật thông báo đã đọc. Thế không ổn đâu.

Đọc binlog thì bó tay nhận biết được phần này, mà không đọc binlog thì phải chia phần cập nhật thông báo trên nhiều service 🡪 khó quản lý lắm!

Mà nghĩ lại thì binlog chỉ có thể cập nhật được thông báo chưa đọc nếu các bảng không có trường gì để nhận biết thông báo liên quan đã được xem.

Vậy là nhiều servie cập nhật **xem rồi**, binlog cập nhật **chưa xem** à? Đéo đâu, bảng Notifications có hẳn một trường để nhận biết thông báo đã đọc chưa mà! Còn cập nhật số lượng thông báo chưa đọc là việc của binlog. 🡺 nhiều service cập nhật trường “**xem rồi**” của Notification, binlog thì cập nhật **số lượng**.