# **BÁO CÁO SAU BUỔI THỰC HÀNH**

**Môn học:** Cơ Sở Dữ Liệu

**Tên buổi thực hành:** Thực hành mini\_project - Hệ thống mạng xã hội

**Thời gian thực hành:** 45p

**Họ và tên sinh viên:** Vũ Thu Trang

**Mã sinh viên:** PTIT-HN-198

**Lớp:** HN-KS24-CNTT4

**Nhóm:** 3

## **I. NỘI DUNG ĐÃ THỰC HÀNH**

### **1. Trình bày các giải pháp**

Nhóm đã thiết kế hệ thống CSDL mạng xã hội tập trung vào tính tự động hóa và toàn vẹn dữ liệu. Hệ thống sử dụng các thực thể chính: Users, Posts, Comments, Likes, và Friends.

**Các kỹ thuật nâng cao đã triển khai:**

* **TRIGGER:** Tự động hóa việc ghi nhật ký (log) khi người dùng đăng bài, thích bài viết hoặc gửi lời mời kết bạn. Đảm bảo tính nhất quán dữ liệu mà không cần can thiệp từ tầng ứng dụng.
* **TRANSACTION:** Xử lý các quy trình phức tạp đòi hỏi tính toàn vẹn tuyệt đối như: "Xóa tài khoản người dùng" (phải xóa đồng thời tất cả bài viết, bình luận, lượt thích) theo nguyên tắc All-or-Nothing.
* **STORED PROCEDURE:** Đóng gói các logic nghiệp vụ như quy trình xóa bài viết, gợi ý bạn bè và thống kê báo cáo.
* **INDEX & VIEW:** Tối ưu tốc độ tìm kiếm bài viết và bảo mật thông tin qua các bảng ảo.

### **2. Liệt kê các câu hỏi từ nhóm khác và câu hỏi phản biện**

**Câu hỏi từ nhóm khác:**

1. Nếu trigger tg\_before\_delete\_post bị lỗi khi insert log, điều gì xảy ra với procedure sp\_delete\_post?
2. Trong chức năng xóa người dùng, dùng câu lệnh gì để kiểm tra trường hợp người dùng không tồn tại?
3. Nếu cả Trigger và Procedure cùng ghi log cho hành động kết bạn, làm sao để tránh ghi log trùng hoặc sai ngữ cảnh?
4. Một Trigger có thể sinh dữ liệu (Insert/Update) cho nhiều bảng khác nhau cùng lúc được không?
5. Ràng buộc UNIQUE(post\_id, user\_id) trong bảng Likes có tác dụng cụ thể gì?

**Câu hỏi phản biện:**

1. Từ khóa DECLARE trong SQL dùng để làm gì và phạm vi hoạt động của nó?
2. Biến NEW và OLD trong Trigger được kích hoạt cụ thể khi nào (Insert/Update/Delete)?
3. Câu lệnh hoặc cơ chế nào giúp Trigger có thể chạy lặp lại nhiều lần?
4. Tại sao nên sử dụng kiểu dữ liệu ENUM thay vì VARCHAR cho cột status (trạng thái)?

### **3. Thực hành triển khai**

**a. Các lệnh SQL nâng cao đã sử dụng:**

* CREATE TRIGGER (AFTER INSERT, BEFORE DELETE).
* START TRANSACTION, COMMIT, ROLLBACK.
* DECLARE biến cục bộ, SIGNAL SQLSTATE để bắt lỗi.

**b. Thiết kế quan hệ:**

* **Users – Posts – Comments/Likes:** Quan hệ 1 – N (Cascade Delete).
* **Friends:** Quan hệ N – N tự thân với trạng thái pending/accepted.

## **II. CÔNG VIỆC ĐÃ LÀM**

### **1. Công việc cá nhân**

* Xây dựng hệ thống **Trigger** để tự động cập nhật số lượng tương tác khi có hành động Like/Unlike.
* Thiết lập **Transaction** cho tính năng xóa tài khoản, đảm bảo tính nhất quán giữa các bảng liên kết.
* Viết Stored Procedure xử lý logic kết bạn và kiểm tra điều kiện không được kết bạn với chính mình.

### **2. Công việc nhóm**

* Thống nhất sơ đồ thực thể (ERD) và các ràng buộc toàn vẹn dữ liệu.
* Phối hợp xử lý lỗi khi Transaction bị ngắt quãng giữa chừng.
* Chuẩn bị nội dung trả lời phản biện về sự khác biệt giữa xử lý logic tại DB và tại App.

## **III. KẾT QUẢ ĐẠT ĐƯỢC**

* Nắm vững cơ chế kích hoạt của **Trigger** và cách kiểm soát luồng dữ liệu tự động.
* Thành thạo kỹ thuật **Transaction** để bảo vệ dữ liệu khỏi các trạng thái không nhất quán.
* Hiểu rõ cách tối ưu hóa các ràng buộc (Unique, Check) để giảm thiểu lỗi logic từ phía người dùng.

## **IV. KHÓ KHĂN VÀ VẤN ĐỀ GẶP PHẢI**

* Lỗi cú pháp khi sử dụng DECLARE không đặt đúng vị trí đầu khối lệnh BEGIN...END.
* Khó khăn khi xử lý lỗi trong Trigger khiến toàn bộ Transaction bị ROLLBACK không mong muốn.
* Xung đột dữ liệu khi đánh chỉ mục UNIQUE trên các cặp khóa ngoại.

## **V. KINH NGHIỆM RÚT RA**

* Cần kiểm tra kỹ logic của Trigger để tránh gây chậm tiến trình khi thực hiện các lệnh Insert/Update hàng loạt.
* Sử dụng **Transaction** là bắt buộc đối với mọi thao tác xóa dữ liệu có quan hệ khóa ngoại phức tạp.
* Nên ưu tiên kiểu dữ liệu ENUM cho các trường trạng thái cố định để tối ưu bộ nhớ và hạn chế sai sót nhập liệu.

## **VI. ĐỀ XUẤT / KIẾN NGHỊ**

* Mong muốn tìm hiểu thêm về cách xử lý các Transaction lồng nhau (Nested Transactions) trong các hệ thống quy mô lớn hơn.

## **VII. KẾT LUẬN**

Em đã hoàn thành tốt các yêu cầu của mini-project, hiểu sâu về cách phối hợp giữa Procedure, Trigger và Transaction để tạo ra một hệ thống CSDL mạng xã hội hoạt động ổn định và an toàn.