

BÁO CÁO SAU BUỔI THỰC HÀNH

Môn học: Cơ sở dữ liệu

Tên buổi thực hành: Thực hành Mini Project Session08

Thời gian thực hành: 180p

Họ và tên sinh viên: Đoàn Mạnh Duy

Mã sinh viên: PTIT-HN-043

Lớp: K24-CNTT4

Nhóm: 3

I. NỘI DUNG ĐÃ THỰC HÀNH

1. Trình bày các giải pháp:

Câu hỏi: SỰ KHÁC NHAU GIỮA VIEW VÀ STORED PROCEDURE LÀ GÌ?

Câu trả lời: Bản chất:

- + View: là Bảng ảo
- + Stored Procedure: Là Chương trình (Hành động).

Chức năng:

- + View: Chỉ dùng để XEM (SELECT).
- + Stored Procedure: Dùng để LÀM (Thêm, Sửa, Xóa, Tính toán).

Tham số:

- + View: KHÔNG nhận tham số.
- + Stored Procedure: CÓ nhận tham số (IN/OUT/INOUT).

Câu hỏi :.Nếu người dùng SELECT từ View thì dữ liệu có được lưu riêng không?

Trả lời: Không

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

- 1. Stored Procedure là gì? lợi ích của nó là gì?
- Trả lời: Là tập hợp các câu lệnh SQL được biên dịch sẵn và lưu trong Database (như một hàm/function).
- lợi ích:
 - - Nhanh: Chạy ngay không cần dịch lại.
 - - Bảo mật: Chống hack (SQL Injection), ẩn cấu trúc bảng.
 - - Tập trung: Sửa logic ở 1 chỗ, tất cả ứng dụng tự cập nhật.
-

- 2. SỰ KHÁC NHAU GIỮA VIEW VÀ STORED PROCEDURE LÀ GÌ?
- Bản chất:
 - + View: là Bảng ảo
 - + Stored Procedure: Là Chương trình (Hành động).
- Chức năng:
 - + View: Chỉ dùng để XEM (SELECT).
 - + Stored Procedure: Dùng để LÀM (Thêm, Sửa, Xóa, Tính toán).
- Tham số:
 - + View: KHÔNG nhận tham số.
 - + Stored Procedure: CÓ nhận tham số (IN/OUT).
- Cách dùng:
 - + View: SELECT * FROM ...
 - + Stored Procedure: CALL ...

II. CÔNG VIỆC CÁC EM ĐÃ LÀM

1. Công việc cá nhân:

Câu 1: Tạo View View_StudentBasic hiển thị: StudentID, FullName , DeptName. Sau đó truy vấn toàn bộ View_StudentBasic;

Câu 2: Tạo Regular Index cho cột FullName của bảng Student.

Câu 3: Viết Stored Procedure GetStudentsIT.

Câu 4:

a) Tạo View View_StudentCountByDept hiển thị: DeptName, TotalStudents (số sinh viên mỗi khoa).

b) Từ View trên, viết truy vấn hiển thị khoa có nhiều sinh viên nhất.

Câu 5: Viết Stored Procedure GetTopScoreStudent

2. Công việc nhóm:

- Vẽ ERD
- Bài 11. Thống kê tương tác nâng cao
- BÀI 12. QUẢN LÝ BÌNH LUẬN
- BÀI 13. QUẢN LÝ LƯỢT THÍCH
- Bài 14. TÌM KIẾM NGƯỜI DÙNG & BÀI VIẾT

III. KẾT QUẢ CÁC EM ĐẠT ĐƯỢC

Sau buổi thực hành, em đã:

Link mini-project: [link](#)

- Khi làm tính năng Thống kê (Bài 7, 11), em thấy Stored Procedure giúp **giảm tải băng thông mạng**. Thay vì kéo hàng nghìn dòng dữ liệu về ứng dụng để cộng trừ nhân chia, em để SQL Server tính toán tại chỗ và chỉ trả về một con số kết quả duy nhất.
- Với bài quản lý Bình luận và Like (Bài 12, 13), em hiểu Procedure là **chốt chặn cuối cùng**. Em nhúng logic kiểm tra (IF/ELSE) ngay trong Procedure. Ví dụ: Nếu người dùng Like rồi thì không được Like nữa. Việc này đảm bảo dữ liệu luôn đúng, bất kể code bên ngoài (Web/App) có bị lỗi hay không.

IV. KHÓ KHĂN VÀ VẤN ĐỀ CÁC EM GẶP PHẢI

1. Khó khăn:

- Tư duy xử lý tập hợp: Khi viết Stored Procedure (như câu 3 GetStudentsIT hay bài 11 thống kê), em gặp khó khăn trong việc chuyển đổi tư duy từ lập trình tuần tự (duyệt vòng lặp `for` như trong C++/Java) sang tư duy tập hợp (Set-based) của SQL. Việc hình dung dữ liệu trả về dưới dạng bảng đôi khi bị rối.
- Xử lý Logic nghiệp vụ phức tạp: Với các bài toán nhóm (Bài 12, 13), việc lồng ghép các điều kiện kiểm tra (IF EXISTS) và giao dịch (TRANSACTION) để đảm bảo tính toàn vẹn dữ liệu (ví dụ: kiểm tra người dùng đã like chưa trước khi tăng biến đếm) phức tạp hơn nhiều so với các câu truy vấn đơn lẻ.

2. Các lỗi thường gặp:

Lỗi tham số: Khai báo sai kiểu dữ liệu của tham số đầu vào (ví dụ: **VARCHAR** quá ngắn) khiến dữ liệu bị cắt bớt hoặc lỗi khi truyền vào Procedure.

Lỗi trùng tên (Ambiguity): Đặt tên tham số đầu vào của Procedure (ví dụ: **StudentID**) trùng với tên cột trong bảng, dẫn đến việc câu lệnh **WHERE** luôn đúng (cột tự bằng chính nó), gây ra việc Update/Delete nhầm toàn bộ dữ liệu.

3. Cách khắc phục:

Quy ước đặt tên: Thống nhất thêm tiền tố cho tham số (ví dụ: dùng `p_StudentID` hoặc `@_StudentID`) để phân biệt rõ ràng với cột `StudentID` trong bảng.

Chia nhỏ bài toán: Viết và chạy thử câu lệnh `SELECT` thuần bên ngoài cho đúng kết quả trước, sau đó mới bao đóng (wrap) nó vào trong cấu trúc `CREATE VIEW` hoặc `CREATE PROCEDURE`.

Kiểm tra kỹ Group By: Luôn rà soát lại mệnh đề `GROUP BY` khi sử dụng các hàm tập hợp như `COUNT`, `SUM`, `AVG`.

V. KINH NGHIỆM RÚT RA

1. Phân định trách nhiệm rõ ràng:

- Sử dụng `VIEW` cho các tác vụ ĐỌC và HIỂN THỊ (như xem danh sách sinh viên, bảng thống kê). Giúp query ngắn gọn và che giấu cấu trúc bảng thật.
 - Sử dụng `STORED PROCEDURE` cho các tác vụ GHI và XỬ LÝ (như thêm sinh viên, chấm điểm, like/comment). Giúp tập trung logic và bảo mật.
2. Tầm quan trọng của Index: Qua câu 2 (Tạo Index cho `FullName`), em nhận thấy việc đánh chỉ mục là bắt buộc đối với các trường thông tin thường xuyên được dùng để tìm kiếm hoặc lọc, giúp tăng tốc độ truy vấn đáng kể.
3. Validation tại nguồn: Không nên chỉ tin tưởng vào kiểm tra ở phía giao diện (Front-end). `Stored Procedure` đóng vai trò là chốt chặn cuối cùng (Last line of defense) để kiểm tra tính hợp lệ của dữ liệu trước khi ghi vào ổ cứng.

VI. ĐỀ XUẤT / KIẾN NGHỊ

Về nội dung: Mong muốn có thêm bài tập về **Trigger** để xử lý các hành động tự động (ví dụ: tự động cập nhật `TotalStudents` trong bảng `Khoa` khi có sinh viên mới thêm vào) nhằm hoàn thiện kiến thức về tự động hóa trong DB.

VII. KẾT LUẬN

Buổi thực hành Mini Project Session 08 đã giúp em củng cố vững chắc kiến thức về Cơ sở dữ liệu nâng cao. Em đã hoàn thành việc xây dựng các thành phần cốt lõi của một hệ thống quản lý và mạng xã hội thu nhỏ, từ việc thiết kế ERD, tối ưu hóa tìm kiếm bằng Index, đến việc phân tách lớp xử lý dữ liệu bằng View và Stored Procedure.