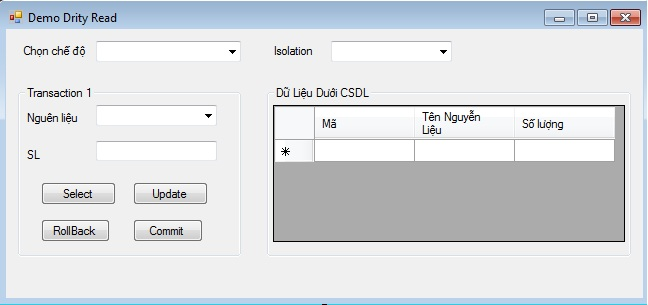
1. “1 chương trình A để thể hiện conflict xảy ra (có thể dùng waitfor để dễ thấy lỗi) và 1 chương trình B sử dụng isolation lock hoặc locking hints để giải quyết vấn đề trên”
2. Như vậy chương trình A sẽ bị gán isolation lock là Read Uncommited hay là chương trình A sử dụng isolation lock mặc định của SQL Server (Read commited)?
3. Giả sử chương trình B sử dụng isolation lock, thì mức cô lập là do nhóm em tự phân tích và lựa chọn cho phù hợp, hay là phải demo từng mức độ lock?
4. Nhóm tụi em xây dựng một chương trình duy nhất, sử dụng một biến flag, khi gán flag = 0 thì chức năng của chương trình sẽ có conflict, khi gán flag = 1 và chạy lại chương trình thì conflict đó đã được xử lý bởi cách thức tụi em đã chọn. Như vậy có được hay không?
5. Thay vì dùng waitfor hoặc thread, tụi em có thể tạo thêm nút commit và rollback trên các form có sử dụng transaction, để khi nào click commit(rollback) thì mới thực hiện commit(rollback) cho transaction đó hay không? (Em có gửi kèm theo hình ảnh)
6. Em nghĩ rằng khi dùng waitfor thì sẽ khó tạo ra những xung đột làm cho transaction phải rollback. Như vậy, tụi em sẽ phải tìm hoặc tạo ra những quá trình xử lý làm cho transaction bị lỗi rồi tự rollback; hay chỉ đơn giản là làm thêm nút rollback giống hình ảnh em gửi kèm?



Hi em,

1/ Chương trình A để ở mức cô lập nào đó mà nó sẽ bị phát sinh lỗi. Đôi khi ta sẽ dùng read uncommitted cho những chức năng cần đọc dữ liệu nhanh mà không cần dữ liệu phải chính xác tuyệt đối. Vì vậy, không phải lúc nào chúng ta để ở mức uncommitted. Ở đây, chương trình A phải được giữ ở mức cô lập nào đó phù hợp với thực tế và gặp lỗi phát sinh. Chương trình B mới thể hiện giải pháp của em như thế nào.

Mức cô lập hay locking hints do nhóm em tự chọn để giải quyết vấn đề. Tuy nhiên, tui có thể hỏi nếu đặt ở mức cô lập khác thì hiện tượng gì xảy ra.

2/ Em làm cách nào cũng được. Những ghi chú của tui chỉ là gợi ý, tuy nhiên, nếu dùng những phần tui gợi ý em sẽ làm được nhiều thứ hơn là đặt cờ, và nó cũng sẽ hợp lý hơn. Vì trong thực tế chẳng ai đặt cờ để cố tình phát sinh lỗi cả.

3/ Nếu thêm nút commit, rollback trên giao diện thì em phải xử lý transaction ở trên [ADO.NET](http://ado.net/) mà không giải quyết được ở mức cơ sở dữ liệu. Cần lưu ý là phải xử lý trong cơ sở dữ liệu, chỉ những phần nào không làm được mới đưa lên [ADO.NET](http://ado.net/). Xử lý ở đâu cũng cần phải hợp lý và có lý do đầy đủ thuyết phục được tui.

4/ Khi 2 transaction bị conflict, thường nó sẽ xảy ra ở 2 dòng nào đó trong transaction, khi đó, chúng ta có thể dùng waitfor cho dừng lại tại dòng đó để dễ thấy lỗi hơn. Còn thêm nút commit, rollback thì em đang làm ở mức ứng dụng rồi, ko còn là trong database nữa.

GVTH