**d**

BÁO CÁO ĐỒ ÁN THỰC HÀNH

MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

GVHD: Tiết Gia Hồng

# **THÔNG TIN NHÓM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ tên** | **Công việc** | **% Hoàn thành** |
| **1** | **18120418** | **Phạm Minh Khoa** | **xử lý tranh chấp, viết báo cáo.** | **90%** |
| **2** |  |  |  |  |
| **3** |  |  |  |  |
| **4** |  |  |  |  |
| **5** |  |  |  |  |

# XỬ LÍ TÌNH HUỐNG TRANH CHẤP

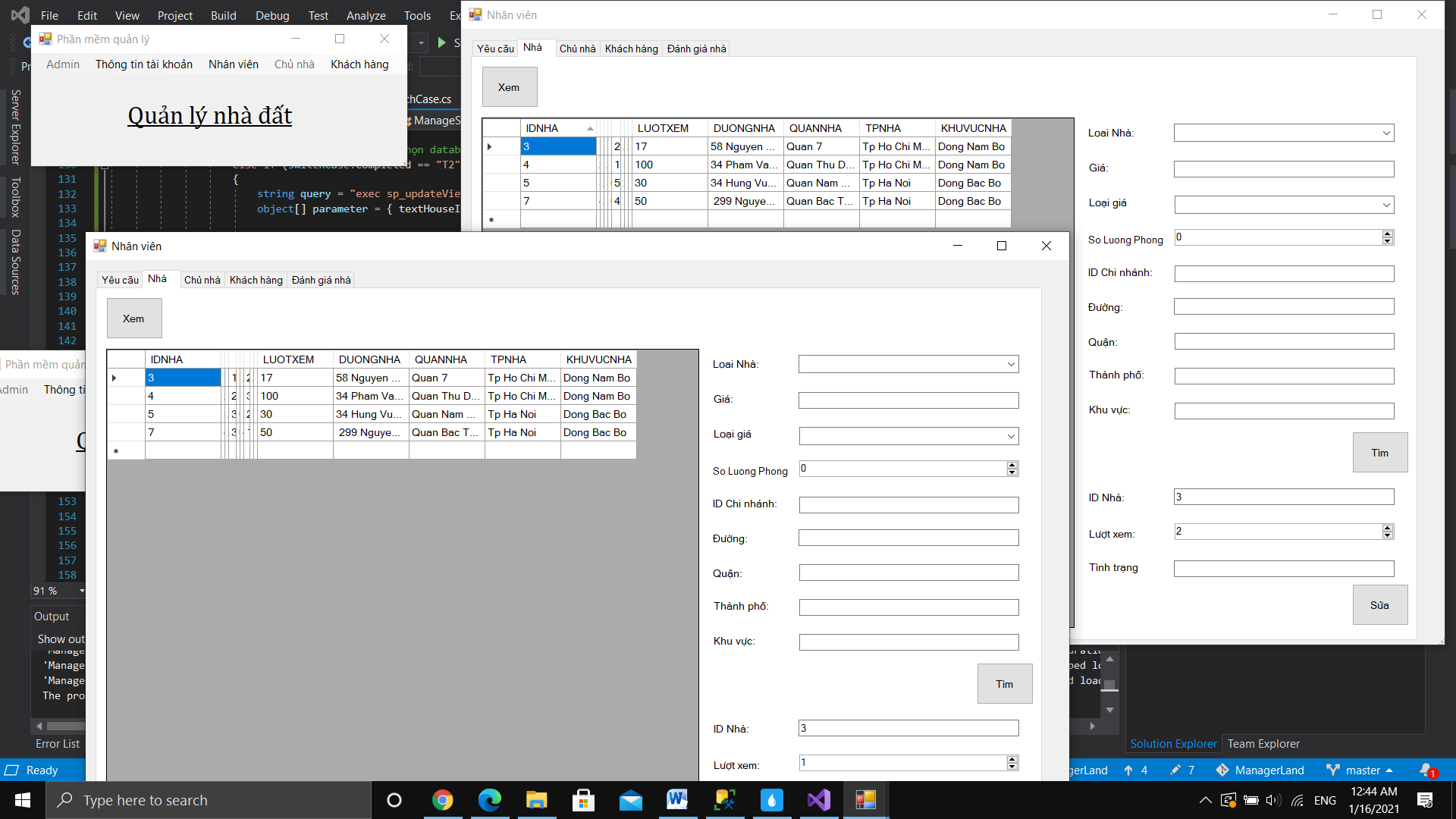
# Sinh viên thực hiện: Phạm Minh Khoa

## Tình huống 1: Hai nhân viên cùng cập nhật lượt xem của 1 nhà.

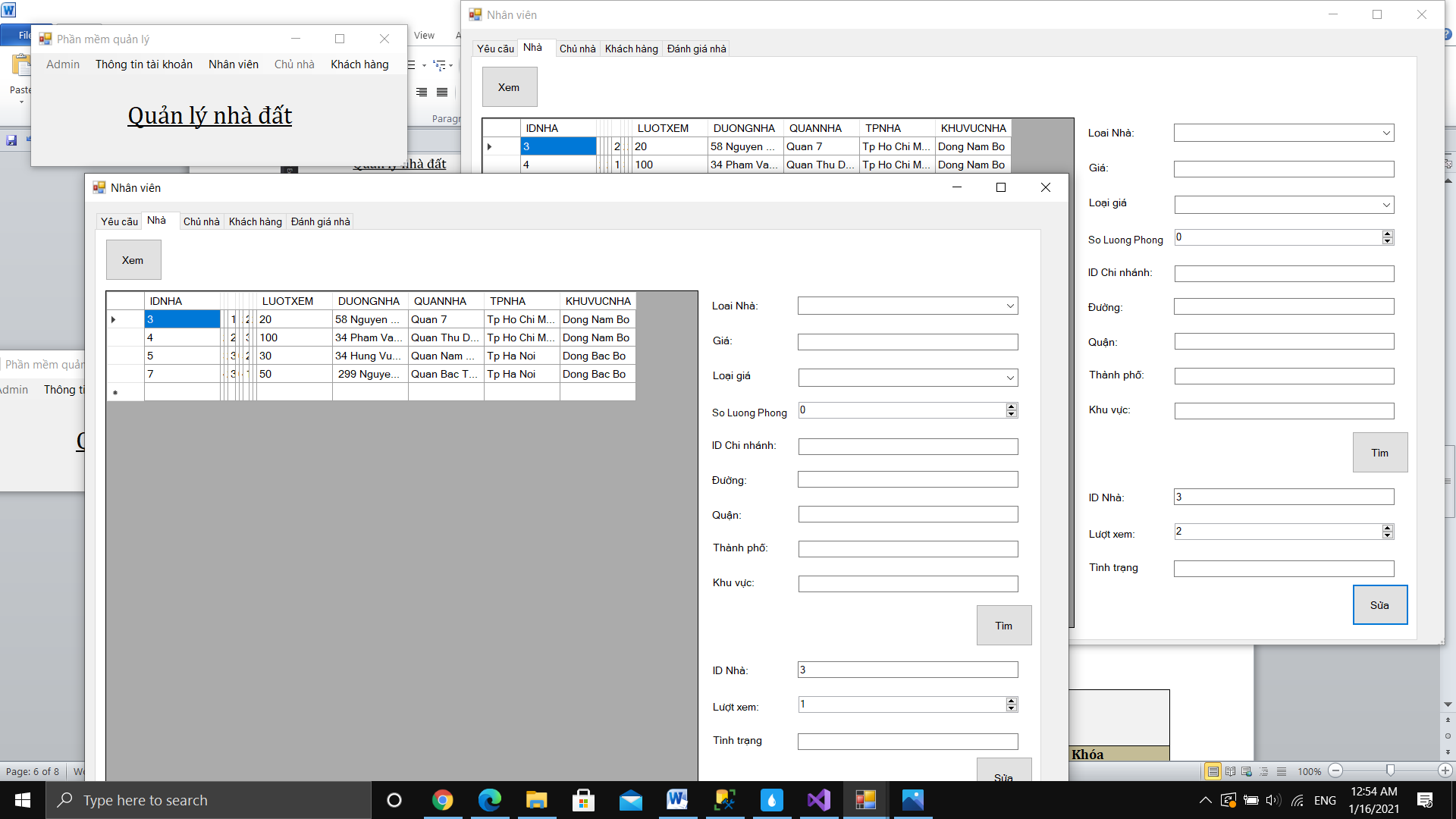
|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Lost Update***  T1 (User = Nhân viên): thực hiện cập nhật lượt xem của nhà A  T2 (User = Nhân viên): thực hiện cập nhật lượt xem của nhà A | | | |
| **sp\_updateView\_FIX\_T1** | **Khóa** | **sp\_updateView\_FIX\_T2** | **Khóa** |
| ***Input:*** *lượt xem nhà*  ***Output***: lượt xem mới | ***Input***: lươt xem nhà  ***Output***: lượt xem nhà mới |
|  |  |
| BEGIN TRAN |  |  |  |
| B1: Lấy lượt xem hiên tại của nhà A.  declare @currentViews int  set @currentViews = (select LUOTXEM from NHA n WITH(UPDLOCK, ROWLOCK) where n.IDNHA = @houseID) | **U(A)**  xin cấp khóa U trên A, hqt thấy thỏa bảng tương thích vì T2 chưa bắt đầu. Sử dụng thêm ROWLOCK để khóa trên một dòng thôi. Nếu không có ROWLOCK thì khóa sẽ phát cho những dòng đã đọc qua mà không liên quan tới dòng dữ liệu thao tác và gây trì trệ hệ thống. | BEGIN TRAN |  |
| waitfor delay '00:00:05' |  |  |  |
|  |  | B1: Đọc lượt xem hiện tại của nhà A.  declare @currentViews int  set @currentViews = (select LUOTXEM from NHA n WITH(UPDLOCK, ROWLOCK) where n.IDNHA = @houseID) | U(A)  xin khóa U, không thoa điều kiện vì T1 đang giữ khóa U trên A. WAIT… |
| B2: Cập nhật lượt xem mới vào nhà A  update NHA  set LUOTXEM = @currentViews + @views  where IDCNHA = @houseID | W(A)  xin khóa X, thỏa bảng tương thích vì T2 không giữ khóa gì cả. |  |  |
| COMMIT |  |  |  |
|  |  | Tiếp tục, B1: Đọc lượt xem hiện tại của nhà A.  declare @currentViews int  set @currentViews = (select LUOTXEM from NHA n WITH(UPDLOCK, ROWLOCK) where n.IDNHA = @houseID) | U(A)  xin khóa U, thỏa điều kiện vì T1 đã commit và trả khóa U. |
|  |  | B2: Cập nhật lượt xem mới vào nhà A  update NHA  set LUOTXEM = @currentViews + @views  where IDCNHA = @houseID | W(A)  xin khóa X, thỏa bảng tương thích.  Giữ cho đến khi commited. |
|  |  | COMMIT |  |

## Để xử lý lostupdate thì không dùng serialization hoặc repeatable read vì nó gây deadlock nên sử dụng updlock và thêm rowlock để khóa trên dòng thỏa điều kiện where. Nếu không có rowlock thì những dòng không có trong điều kiện where nếu có đọc qua chúng để duyệt tìm dòng phụ hợp thì chúng cũng bị lock lại. Khi bị lock lại những giao tác khác cần mà thao tác trên đó không được sẽ gây trì trệ hệ thống.

## Lúc đầu cả hai nhà có id =3 lượt view đều là 17:



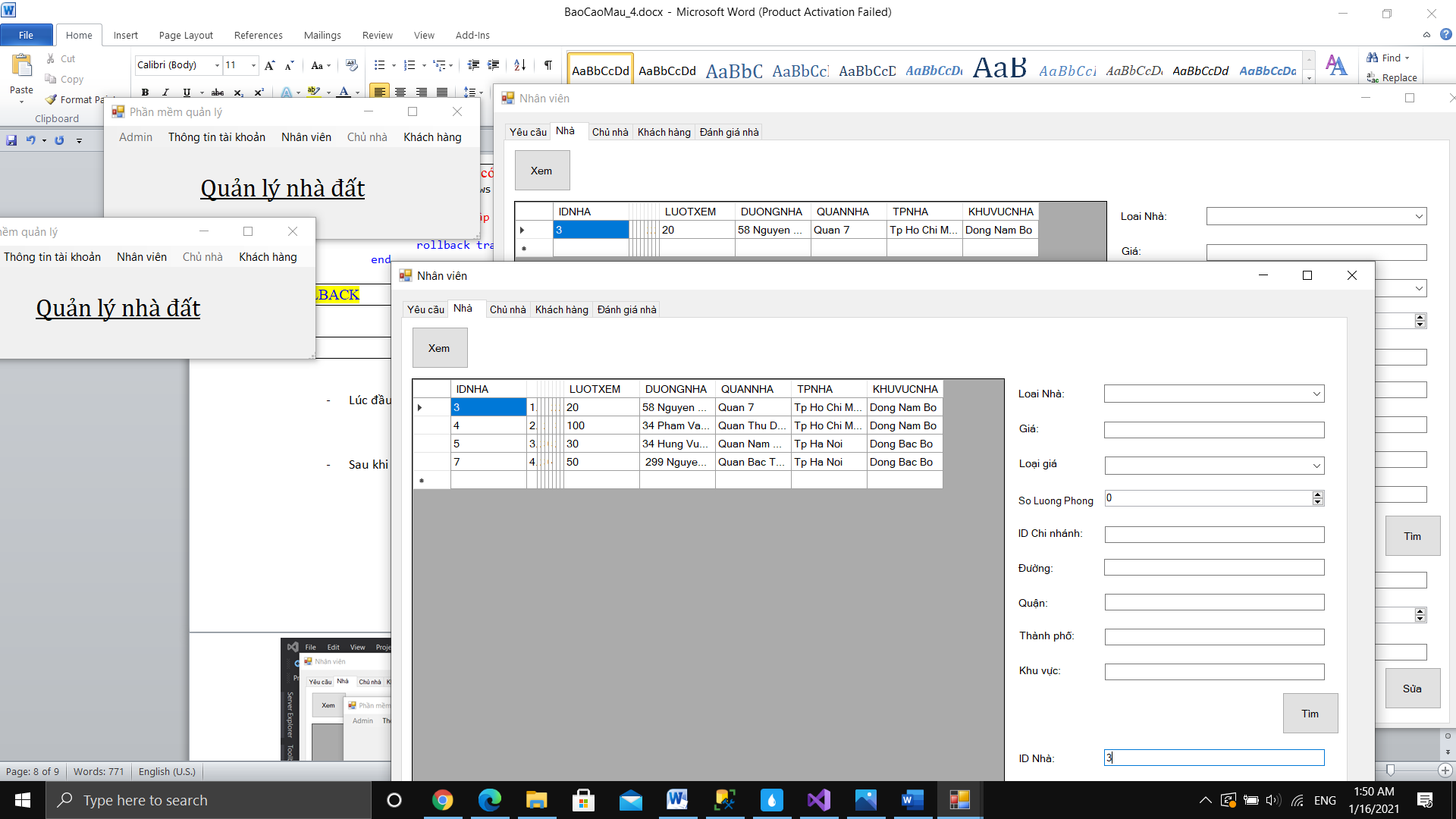
* Sau đó giao tác T1 tăng view lên 1, T2 tăng view lên 2. Thực hiện xong cả hai giao tác lượt view lúc này là 17 + 1 + 2 = 20



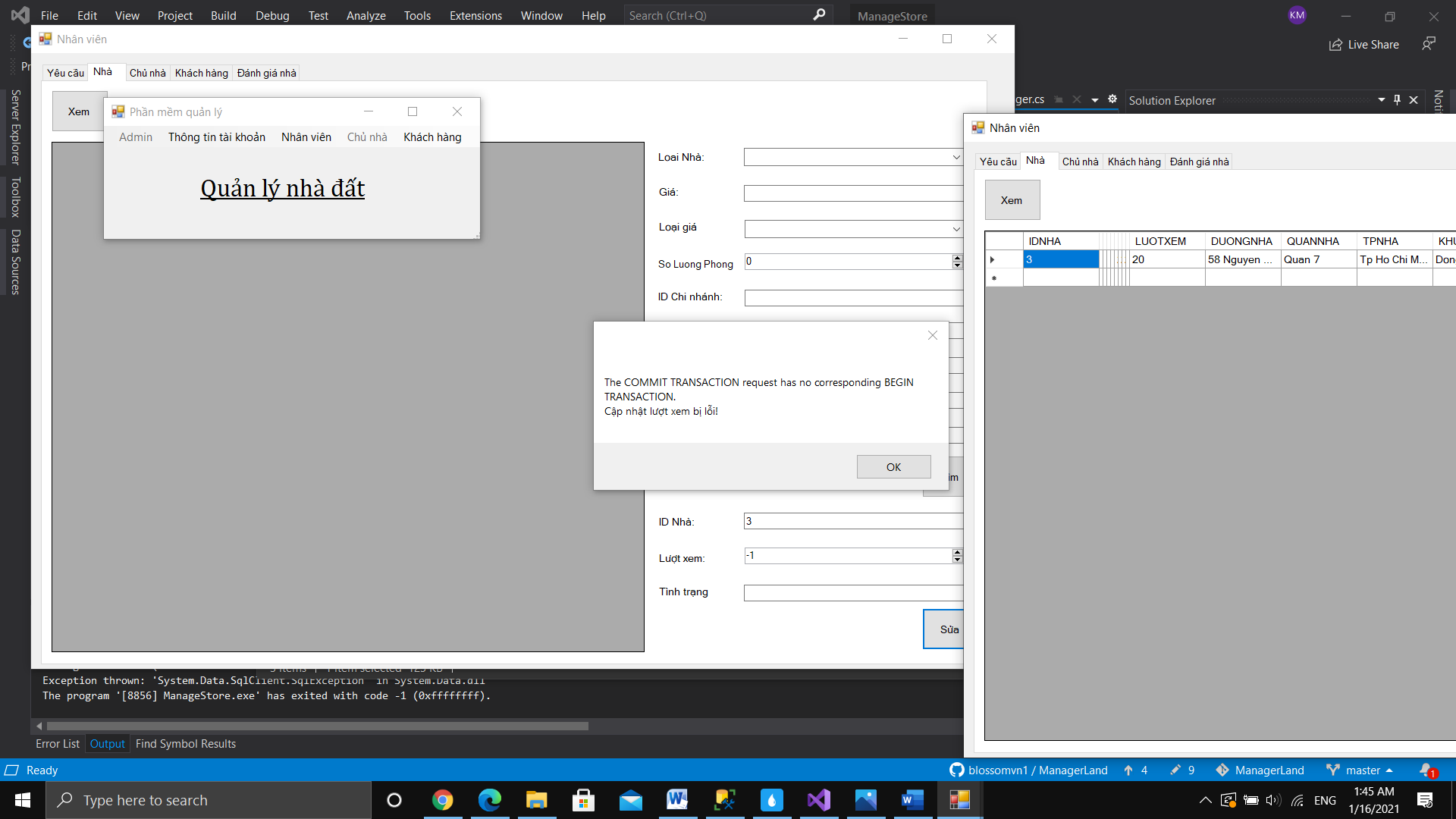
## Tình huống 2: Nhân viên cập nhật lượt xem và chủ nhà vào xem lượt xem.

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR02: Dirty Read**  T1 (User = Nhân viên): thực hiện cập nhật lượt xem của nhà A  T2 (User = Chủ nhà): thực hiện xem lượt xem của nhà A | | | |
| sp\_updateView\_2\_T1 | **Khóa** | **sp\_getView** | **Khóa** |
| ***Input:*** *lượt xem nhà*  ***Output***: rollback | ***Input***: IDNHA  ***Output***: lượt xem rác |
| SET TRANSACTION ISOLATION  LEVEL READ COMMITTED (mặc định) | SET TRANSACTION ISOLATION  LEVEL READ COMMITTED |
| BEGIN TRAN |  |  |  |
| B1: Lấy lượt xem hiên tại của nhà A.  declare @currentViews int  set @currentViews = (select LUOTXEM from NHA where IDNHA = @houseID) | **R(A)**  đọc lượt xem của nhà A trên bảng NHA  Hệ quản trị phát khóa S. Đọc xong trả lại S. |  |  |
| B2: Cập nhật lượt xem tại nhà A  update NHA  set LUOTXEM = @currentViews + @views  where IDCNHA = @houseID | W(A)  xin khóa X, thỏa hqt cấp cho khóa X.  Giữ tới cuối giao tác |  |  |
| waitfor delay '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B2: Đọc lượt xem  select \* from NHA where IDNHA = @houseID | **R(A)**  **Xin khóa S và không được cấp vì T1 đang giữ khóa X trên nhà A.**  **Chờ…** |
|  |  |  |  |
| B3: Kiểm tra điều kiện lượt xem có thỏa không  if(@currentViews > @currentViews + @views)  begin  print N'Cập nhật lượt xem bị lỗi!'  rollback tran  end |  |  |  |
| ROLLBACK |  |  |  |
|  |  | B2: Đọc lượt xem  select \* from NHA where IDNHA = @houseID | Tiếp tục |
|  |  | COMMIT TRAN |  |

* Để ngăn dirty read ta set lại mức cô lập của T2 từ read uncommmited htành read commited
* Lúc đầu nhà có id = 3 có view = 20



* Sau khi thực hiện T1 và T2 thì T1 rollback và T2 đọc lên vẫn 20



# II.Sinh viên thực hiện:Huỳnh Ngọc Hòa

## Tình huống 1: Chủ nhà thêm 1 nhà nhưng chưa commit thì nhân viên vào xem thông tin nhà

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Dirty Read***  T1 (User = Chủ nhà): thực hiện thêm 1 nhà  T2 (User = Nhân viên): thực hiện xem thông tin nhà | | | |
| **sp\_ThemNha** | **Khóa** | **sp\_XemThongTinNha** | **Khóa** |
| ***Input:*** *Thông tin nhà*  ***Output***: Rollback | ***Input***: không  ***Output***: thông tin tất cả nhà |
| SET TRANSACTION ISOLATION  LEVEL READ COMMITTED (mặc định) | SET TRANSACTION ISOLATION  LEVEL READ COMMITTED |
| BEGIN TRAN |  |  |  |
| B1: Thêm 1 nhà vào bảng NHA.  Insert into NHA values (Thông tin nhà)  waitfor delay '00:00:10' | **W(NHA)**  Chèn thông tin nhà mới vào bảng NHA  Hệ quản trị cấp khóa X và giữ đến hết giao tác | BEGIN TRAN |  |
|  |  | Đọc thông tin nhà.  select \*  from NHA | R(NHA)  **Xin khóa đọc trên bảng NHA nhưng không được vì T1 đang giữ khóa ghi trên nhà(phải chờ)** |
|  |  |  |  |
| B2: kiểm tra điều kiện  if(DATEDIFF(day,@NgayHetHan,@NgayDang)>0)  begin  print N'Ngày hết hạn không được nhỏ hơn ngày đăng'  rollback tran  end |  |  |  |
| ROLLBACK | Trả khóa đang giữ trên bảng NHA |  |  |
|  |  | Tiếp tục đọc thông tin nhà vì T1 đã trả khóa  select \*  from NHA | R(NHA)  Xin khóa đọc trên bảng nhà |
|  |  | COMMIT |  |

## Tình huống 2: Nhân viên đang đọc thông tin nhà thì chủ nhà cập nhập lại nhà khiến lần đọc sau đó của nhân viên không còn chính xác nữa

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Unrepeatable Read***  T1 (User = Nhân viên): Đọc thông tin nhà  T2 (User = Chủ nhà): Cập nhập lại địa chỉ nhà | | | |
| **sp\_XemThongTinNha** | **Khóa** | **sp\_CapNhapNha** | **Khóa** |
| ***Input:*** *không*  ***Output***: không | ***Input***: Thông tin mới của nhà cần cập nhập  ***Output***: không |
| SET TRANSACTION ISOLATION  LEVEL REPEATABLE READ | SET TRANSACTION ISOLATION  LEVEL REPEATABLE READ |
| BEGIN TRAN |  |  |  |
| B1: Đọc thông tin nhà.  select \*  from NHA  waitfor delay '00:00:10' | **R(NHA)**  Xin khóa S trên bảng NHA và giữ đến hết giao tác | BEGIN TRAN |  |
|  |  | Cập nhập thông tin nhà.  update NHA  set NHA.DUONGNHA =@DuongNha,  NHA.IDCNHA = @idChuNha,  NHA.IDCNHANH = @idChiNhanh,  NHA.IDLOAINHA = @idLoaiNha,  NHA.IDNV = @idNhanVien,  NHA.KHUVUCNHA = @KhuVucNha,  NHA.LUOTXEM = @LuotXem,  NHA.NGAYDANG=@NgayDang,  NHA.NGAYHETHANG = @NgayHetHan,  NHA.QUANNHA = @QuanNha,  NHA.SOLUONGNHA = @SoLuongNha,  NHA.TINHTRANG = @TinhTrang,  NHA.TPNHA = @TPNha  where NHA.IDNHA = @idNha | W(NHA)  **Xin khóa X trên bảng NHA nhưng không dược do T1 đang giữ khóa S trên NHA (phải chờ)** |
|  |  |  |  |
| B2: Đọc thông tin nhà.  select \*  from NHA | **R(NHA)**  Xin khóa S trên bảng NHA |  |  |
| COMMIT | **Trả khóa S đang giữ trên bảng NHA** |  |  |
|  |  | Tiếp tục cập nhập do khóa đã được trả  update NHA  set NHA.DUONGNHA =@DuongNha,  NHA.IDCNHA = @idChuNha,  NHA.IDCNHANH = @idChiNhanh,  NHA.IDLOAINHA = @idLoaiNha,  NHA.IDNV = @idNhanVien,  NHA.KHUVUCNHA = @KhuVucNha,  NHA.LUOTXEM = @LuotXem,  NHA.NGAYDANG=@NgayDang,  NHA.NGAYHETHANG = @NgayHetHan,  NHA.QUANNHA = @QuanNha,  NHA.SOLUONGNHA = @SoLuongNha,  NHA.TINHTRANG = @TinhTrang,  NHA.TPNHA = @TPNha  where NHA.IDNHA = @idNha  COMMIT | W(NHA)  Xin khóa X trên bảng NHA và giữ đến hết giao tác |