##### **ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

##### **TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

****

**BÁO CÁO ĐỒ ÁN THỰC HÀNH**

**MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**GVHD: Hồ Thị Hoàng Vy**

### Nhóm 07:

### 18120213 Võ Đại Nam

### 18120214 Lê Ngọc Bảo Ngân

### 18120215 Vũ Yến Ngọc

### 18120217 Nguyễn Trần Ái Nguyên

### 18120227 Phạm Văn Minh Phương

# THÔNG TIN NHÓM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ tên** | **Công việc** | **% Hoàn thành** |
| **1** | 18120213 | Võ Đại Nam | * Cài đặt tình huống tranh chấp: Dirty Read * Cài đặt tình huống tranh chấp: Lost Update | 100% |
| **2** | 18120214 | Lê Ngọc Bảo Ngân | * Cài đặt tình huống tranh chấp: Cycle DeadLock * Cài đặt tình huống tranh chấp: Conversion DeadLock | 100% |
| **3** | 18120215 | Vũ Yến Ngọc | * Cài đặt tình huống tranh chấp: Dirty Read * Cài đặt tình huống tranh chấp: Lost Update | 100% |
| **4** | 18120217 | Nguyễn Trần Ái Nguyên | * Cài đặt tình huống tranh chấp: Unrepeatable Read * Cài đặt tình huống tranh chấp: Phantom | 100% |
| **5** | 18120227 | Phạm Văn Minh Phương | * Cài đặt tình huống tranh chấp: Unrepeatable Read * Cài đặt tình huống tranh chấp: Phantom | 100% |

# CÀI ĐẶT TÌNH HUỐNG TRANH CHẤP

# Sinh viên thực hiện: 18120213

## Tình huống 1: Dirty Read

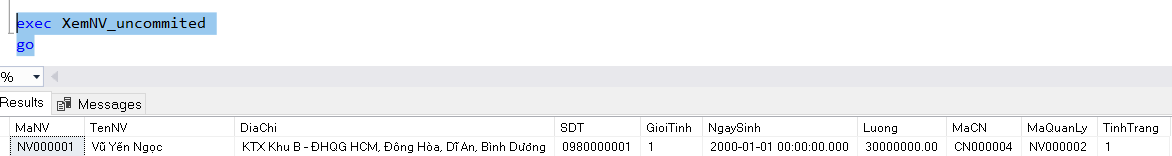
|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Dirty Read***  T1 (User = CEO): thực hiện update chi nhánh và nhân viên quản lý cho 1 nhân viên  T2 (User = QLCN): thực hiện thống kê nhân viên tại các chi nhánh  Ví dụ: CEO chuyển công tác cho nhân viên với mã là NV000008, update Mã chi nhánh mới CN000004, update Mã nhân viên quản lý mới NV000002. Tại thời điểm đó, QLCN cũng đang thống kê nhân viên ở các chi nhánh. | | | |
| **sp\_ChuyenNV\_uncommited** | **Khóa** | **sp\_ChuyenNV\_uncommited** | **Khóa** |
| ***Input:*** @manv, @mcn, @mql  ***Output***: Cập nhập chi nhánh và nhân viên quản lý mới cho nhân viên đã chọn | ***Input***:  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL COMMITTED. | SET TRANSACTION ISOLATION  LEVEL UNCOMMITTED |
| BEGIN TRAN |  |  |  |
| B1: Cập nhập chi nhánh mới  update NhanVien  set MaCN=@mcn  where MaNV=@manv | **X**  **(Nhân viên)** |  |  |
| waitfor delay '00:00:10' |  | BEGIN TRAN |  |
|  |  | B1: Thống kê Nhân viên  select \* from NhanVien | **R**  **(Nhân Viên)**  Không cần xin khoá |
|  |  | COMMIT |  |
| B2: Cập nhập nhân viên quản lý mới  update NhanVien  set MaQuanLy=@mql  where MaNV=@manv | **X**  **(Nhân viên)** |  |  |
| B3: Thống kê Nhân viên  select \* from NhanVien where MaNV=@manv | **R**  **(Nhân viên)** Xin khóa tại bảng Nhân Viên |  |  |
| COMMIT |  |  |  |
| if(not exists(select \* from ChiNhanh where MaCN=@mcn) or not exists(select \* from NhanVien where MaNV=@manv))  begin  RAISERROR('Chuyen nhan vien khong thanh cong',1,1)  ROLLBACK TRAN sp\_ChuyenNV\_uncommited |  |  |  |

Kết quả: Dựa vào kết quả bên dưới ta thấy được đáng lẽ QLCN phải thấy được cả hai yếu tố update của CEO nhưng lại chỉ thấy được mã chi nhánh được cập nhập 🡺 Xuất hiện lỗi Dirty Read

* **User**: CEO



* **User**: QLCN

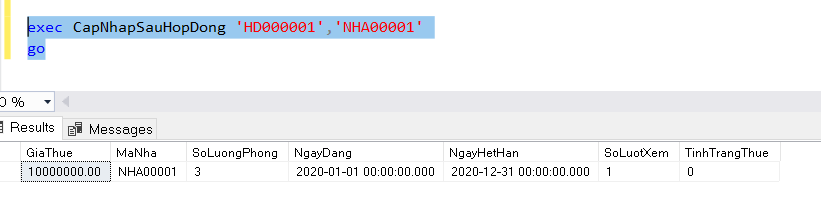


## Tình huống 2: Lost Update

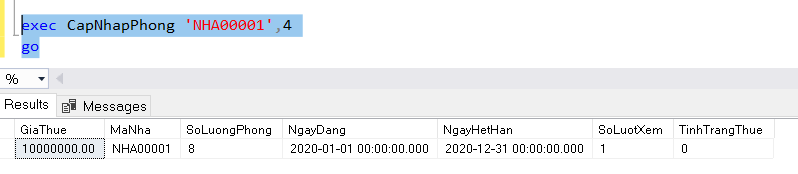
|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Lost Update***  T1 (User = Nhân viên): thực hiện update số phòng sau khi hợp đồng thành công  T2 (User = Chủ nhà): thực hiện update số phòng còn lại mới  Ví dụ: Sau khi làm hợp đồng cho thuê phòng xong, nhân viên sẽ cập nhập lại số phòng= số phòng -1. Tại thời điểm đó chủ nhà xây thêm phòng cho thuê nên chủ nhà cập nhập lại số phòng của nhà mình. | | | |
| **sp\_ChuyenNV\_uncommited** | **Khóa** | **sp\_CapNhat** | **Khóa** |
| ***Input*** @mhd, @manha  ***Output***: Cập nhập Số phòng | ***Input***: @manha, @spt  ***Output***: Cập nhập số phòng |
| SET TRANSACTION ISOLATION  LEVEL COMMITTED. | SET TRANSACTION ISOLATION  LEVEL UNCOMMITTED |
| BEGIN TRAN |  |  |  |
| B1: Đọc dữ liệu tại bảng Nhà thuê  declare @temp int  select @temp=SoLuongPhong from NhaThue where MaNha=@manha | **R**  **(Nhà thuê)**  Xin khóa đọc tại Nhà thuê |  |  |
| waitfor delay '00:00:10' |  | BEGIN TRAN |  |
|  |  | B1: Đọc dữ liệu tại bảng nhà thuê  declare @temp int  select @temp=SoLuongPhong from NhaThue where MaNha=@manha | **R**  **(Nhà thuê)**  Không cần xin khoá |
|  |  | B2: Cập nhập số phòng mới  update NhaThue  set SoLuongPhong=@temp+@spt  where MaNha=@manha | **X**  **(Nhà thuê)** |
|  |  | COMMIT |  |
| B2: Cập nhập số phòng mới  update NhaThue  set SoLuongPhong=@temp-1  where MaNha=@manha | **X**  **(Nhà thuê)** |  |  |
| B3: Thống kê lại số phòng  select \* from NhaThue where MaNha=@manha | **R**  **(Nhà thuê)** |  |  |
| COMMIT |  |  |  |
| if(not exists(select \* from HopDong where MaHD=@mhd) or not exists(select\* from NhaThue where MaNha=@manha))  begin  RAISERROR('Chuyen nhan vien khong thanh cong',1,1)  ROLLBACK TRAN sp\_CapNhapSau  end |  | if(not exists(select\* from NhaThue where MaNha=@manha))  begin  RAISERROR('Chuyen nhan vien khong thanh cong',1,1)  ROLLBACK TRAN sp\_CapNhap  end |  |

Kết quả: Dựa vào hình ở dưới ta thấy ở nhân viên số phòng là 3, ở chủ nhà số phòng là 8, điều này vô lý bởi vì số phòng đáng lẽ là 4+4-1=7 mà ở đây lại là 3 và 8 => Lost Update

* **User**: Nhân viên



* **User**: Chủ nhà

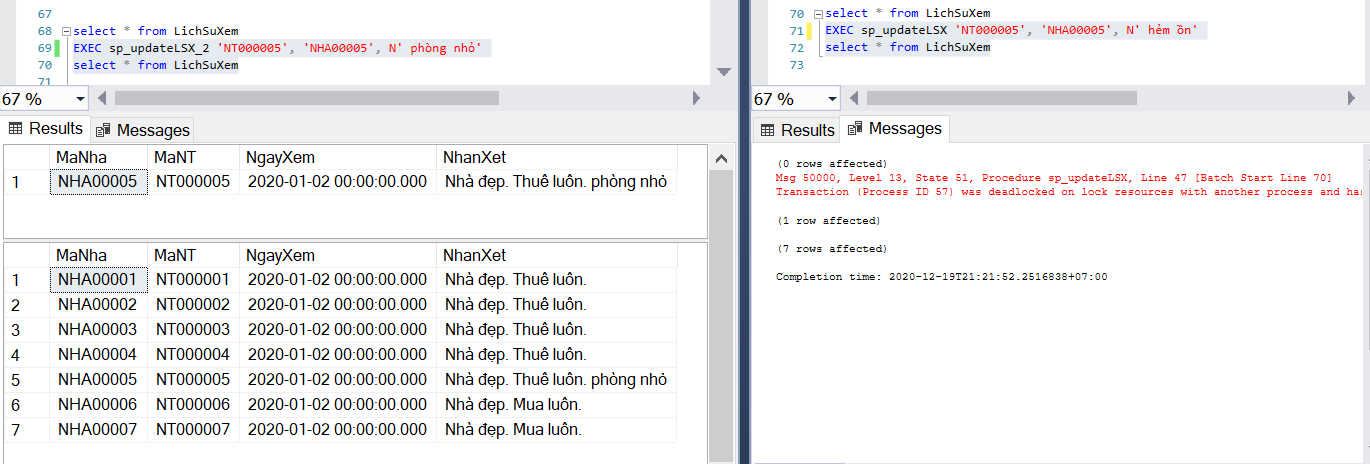


# Sinh viên thực hiện: 18120214

## Tình huống 3: Conversion DeadLock

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR02: *Conversion DeadLock***  T1 (User = NhanVien): cả 2 nhân viên đồng thời cập nhật nhận xét cùng 1 nhà trong LichSuXem  T2 (User = NhanVien): cả 2 nhân viên đồng thời cập nhật nhận xét cùng 1 nhà trong LichSuXem  Ví dụ trường hợp: Nhân viên 1 cập nhật nhật xét @NX = “hẻm ồn” cho @maNT = 'NT000006', @maNha ='NHA00006', đồng thời Nhân viên 2 cũng cập nhật cho @maNT = 'NT000006', @maNha ='NHA00006' nhưng @NX = ‘phòng nhỏ’ | | | |
| **sp\_updateLSX** | **Khóa** | **sp\_updateLSX\_2** | **Khóa** |
| ***Input:*** @maNT, @maNha, @NX  ***Output***: Cập nhật lịch sử xem | ***Input:*** @maNT, @maNha, @NX  ***Output***: Cập nhật lịch sử xem |
| SET TRAN ISOLATION LEVEL SERIALIZABLE | SET TRAN ISOLATION LEVEL SERIALIZABLE |
| BEGIN TRAN |  |  |  |
| B1: Đọc dữ liệu tại bảng LichSuXem  DECLARE @nxet NVARCHAR(50) = (SELECT NhanXet FROM LichSuXem WHERE MaNT = @maNT and MaNha = @maNha) | **R(LichSuXem)**  Dọc Dữ liệu Luong trên 1 dòng của bảng LichSuXem |  |  |
| waitfor delay '00:00:7' |  | BEGIN TRAN |  |
|  |  | B1: Đọc dữ liệu tại bảng LichSuXem  DECLARE @nxet NVARCHAR(50) = (SELECT NhanXet FROM LichSuXem WHERE MaNT = @maNT and MaNha = @maNha) | **R**  **(LichSuXem)**  Không cần xin khoá |
|  |  | B2: B2: Cập nhập nhận xét  SET @nxet = @nxet + @NX  UPDATE LichSuXem SET NhanXet = @nxet WHERE MaNha = @maNha and MaNT = @maNT | **X**  **(LichSuXem)** |
|  |  | COMMIT |  |
| B2: Cập nhập nhận xét  SET @nxet = @nxet + @NX  UPDATE LichSuXem SET NhanXet = @nxet WHERE MaNha = @maNha and MaNT = @maNT | **X**  **(LichSuXem)** |  |  |
| RAISERROR  IF @@TRANCOUNT > 0  ROLLBACK TRAN |  |  |  |
| COMMIT |  |  |  |

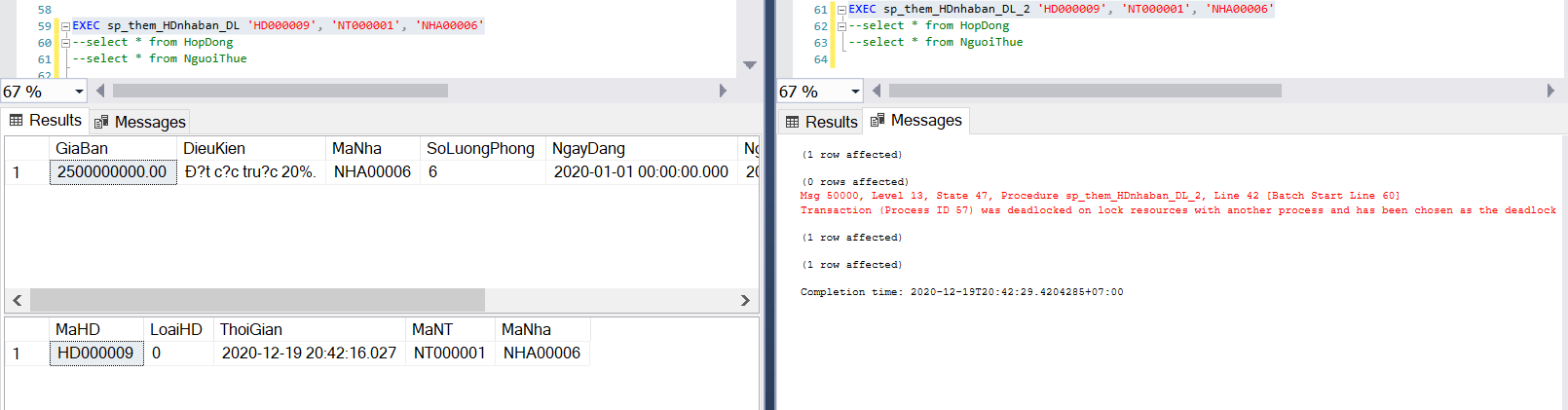
Kết quả: T1 bắt DeadLock, kết quả của bảng LichSuXem chỉ cập nhật nhận xét của T1



## Tình huống 4: Cycle DeadLock

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR04: *Cycle DeadLock***  T1 (User = NhanVien): Thêm 1 hợp đồng và cập nhật lại tình trạng của Nhà bán  T2 (User = NhanVien): Cập nhật tình trạng của Nhà bán và thêm 1 hợp đồng  Ví dụ trường hợp: Nhân viên 1 thêm 1 hợp đồng với @maHD= 'HD000009', @maNT= 'NT000001’ và @maNha='NHA00006' rồi sau đó tiến hành cập nhật tình trạng nhà đó thành đã bán, còn Nhân viên 2 thực hiện ngược lại | | | |
| **sp\_ChuyenNV\_uncommited** | **Khóa** | **sp\_CapNhap** | **Khóa** |
| ***Input*** @maNT, @maNha, @maHD  ***Output***: Cập nhật tình trạng nhà và thêm 1 hợp đồng | ***Input*** @maNT, @maNha, @maHD  ***Output***: Thêm 1 hợp đồng và cập nhật tình trạng nhà |
| BEGIN TRAN |  |  |  |
| B1: Cập nhật tình trạng nhà bán  DECLARE @tt BIT  SET @tt = '0'  UPDATE NhaBan SET TinhTrangBan = @tt WHERE MaNha = @maNha | **X**  **(Nhà bán)** |  |  |
| waitfor delay '00:00:07' |  | BEGIN TRAN |  |
|  |  | B1: Thêm 1 hợp đồng  INSERT INTO HopDong (MaHD, LoaiHD, ThoiGian, MaNT, MaNha) VALUES (@maHD, 0, GETDATE(), @maNT, @maNha) | **X(Hợp Đồng)** |
|  |  | B2: Cập nhập tình trạng nhà bán  DECLARE @tt BIT  SET @tt = '0'  UPDATE NhaBan SET TinhTrangBan = @tt WHERE MaNha = @maNha | **X**  **(Nhà bán)** |
|  |  | COMMIT |  |
| B2: Thêm 1 hợp đồng  INSERT INTO HopDong (MaHD, LoaiHD, ThoiGian, MaNT, MaNha) VALUES (@maHD, 0, GETDATE(), @maNT, @maNha) | **X(Hợp Đồng)** |  |  |
| RAISERROR  IF @@TRANCOUNT > 0  ROLLBACK TRAN |  |  |  |
| COMMIT |  |  |  |

Kết quả: T2 bắt DeadLock



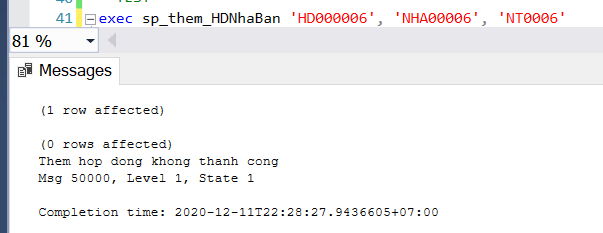
# Sinh viên thực hiện: 18120215

## Tình huống 5: Dirty Read

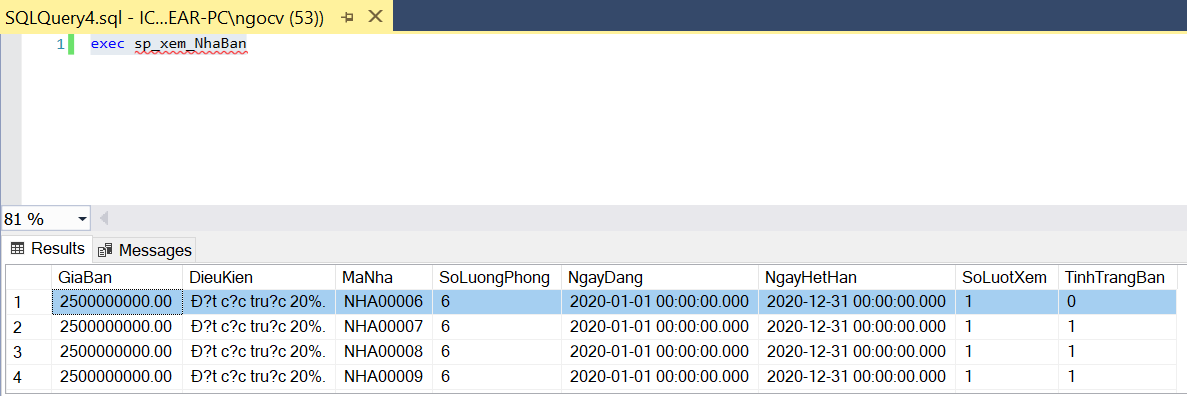
|  |  |  |  |
| --- | --- | --- | --- |
| **ERR05: *Dirty Read***  T1 (User = NhanVien): Thực hiện thêm một hợp đồng (NV có mã là ‘NV000006’)  T2 (User = NguoiThue): Thực hiện thống kê NhaBan  Ví dụ: @maHD=’HD000006’, @maNha=’NHA00006’, @maNT=’NT06’  Ở VD này mã Người Thuê không hợp lệ làm cho T1 bị rollback 🡪 T2 select bảng nhà sẽ đọc dữ liệu sai | | | |
| **sp\_them\_HDNhaBan** | **Khóa** | **sp\_xem\_NhaBan** | **Khóa** |
| ***Input:*** @m*aHD, @maNha và @maNT*  ***Output***: Cập nhật lại tình trạng của căn nhà được bán và thêm 1 hợp đồng mới | ***Input***:  ***Output***: Danh sách NhaBan |
| SET TRANSACTION ISOLATION  LEVEL **READ COMMITED** | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITED** |
| BEGIN TRAN |  |  |  |
| B1: Cập nhật tình trạng căn nhà được bán  update NhaBan set TinhTrangBan = 0 where MaNha=@maNha | **X(NhaBan)**  Xin khoá ghi trên bảng NhaBan |  |  |
| B2: Thêm 1 dòng mới vào bảng Hợp Đồng  insert into HopDong(MaHD,LoaiHD,ThoiGian,MaNT,MaNha) values (@maHD,0,GETDATE(),@maNT,@maNha) | **X(HopDong)**  Xin khoá ghi trên bảng HopDong |  |  |
| WAITFOR DELAY '00:00: 07' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Thống kê Nhân viên  select \* from NhaBan | **R(NhaBan)** |
|  |  | COMMIT |  |
| if not exists(select \* from NhaBan where MaNha=@maNha) or not exists(select \* from NguoiThue where MaNT=@maNT)  begin    RAISERROR('Them hop dong khong thanh cong',1,1)  ROLLBACK TRAN sp\_them\_HDNhaBan  end  else  COMMIT TRAN sp\_them\_HDNhaBan |  |  |  |

Kết quả: Hợp đồng không được thêm thành công tức ‘NHA00006’ chưa được bán (TinhTrangNha=1) nhưng khi T2 đọc dữ liệu thì tình trạng nhà của ‘NHA00006’ lại là 0 nghĩa là đã bán 🡪 Dirty Read

* **User**: Nhân viên



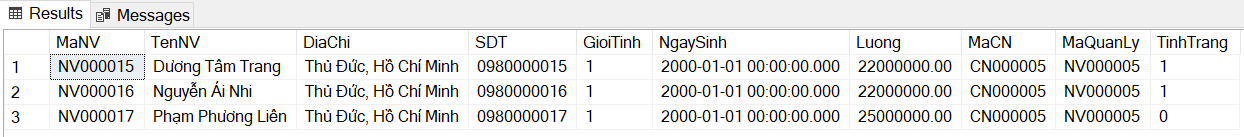
* **User**: Người thuê



## Tình huống 6: Lost Update

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR06: *Lost Update***  T1 (User = NhanVienQuanLy): Thực hiện tăng lương của 1 nhân viên trong chi nhánh lên 5tr  T2 (User = CEO): Thực hiện tăng hệ số lương của 1 chi nhánh  Ví dụ:  T1: NhanVienQuanLy chi nhánh có maCN=’CN000005’ thay đổi lương của nhân viên có maNV=’NV000017’ tăng lên 1 khoảng @Luong=5tr  T2: CEO thay đổi lương của toàn bộ chi nhánh có maCN=’CN000005’ lên 1.1 lần  Ở VD này khi NhanVienQuanLy cập nhật lương tăng 5tr tức Luong = 25tr cho nhân viên có MaNV=’NV000017’ sau đó CEO lại tăng lương thêm 1.1 lần thì lương sẽ là 27.5tr 🡪 nhưng ở đây do cả 2 đều ghi trên dữ liệu của nhân viên NV000015 nên mức lương của nó chỉ nhận cái commit sau cùng đó là 25tr và bị mất việc CEO update hệ số lên 1.1 | | | |
| **sp\_tang\_Luong\_nhanvien** | **Khóa** | **sp\_tang\_luong\_theochinhanh** | **Khóa** |
| ***Input:*** @m*aNV, @maCN và @Luong*  ***Output***: Cập nhật lại Luong của nhân viên đó là Luong= Luong+@Luong | ***Input***: @maCN, @heso  ***Output***: Luong của cả chi nhánh có mã là @maCN sẽ tăng lên @heso lần |
| SET TRANSACTION ISOLATION  LEVEL **READ COMMITED** | SET TRANSACTION ISOLATION  LEVEL **READ COMMITED** |
| BEGIN TRAN |  |  |  |
| B1: Đọc Luong của Nhân viên cần tăng  declare @setluong money  select @setluong=Luong from NhanVien where MaNV=@maNV | **R(NhanVien)**  Dọc Dữ liệu Luong trên 1 dòng của bảng NhanVien |  |  |
| WAITFOR DELAY '00:00: 07' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B2: Tăng lương của Nhân viên thuộc Chi nhánh theo hệ số  update NhanVien set Luong = Luong\*@hesoLuong where MaCN=@maCN | **X(NhanVien)** |
|  |  | COMMIT |  |
| B2: Tăng lương của Nhân viên thuộc Chi nhánh lên một lượng là @Luong  set @setluong=@setluong+@Luong  update NhanVien set Luong = @setluong where MaNV=@maNV | **X(NhanVien)** |  |  |
| COMMIT |  |  |  |

Kết quả:



# Sinh viên thực hiện: 18120217

## Tình huống 7: Unrepeatable Read

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR07: *Unrepeatable Read***  T1 (User = Nhân viên): thực hiện thống kê Người thuê nhà thuộc chi nhánh  T2 (User = người thuê): người thuê NT000001 thực hiện update thông tin của mình  Ví dụ: @MaNT=’NT000001’@MaCN=’CN000001’ 🡪 T1 không đọc được update dữ liệu | | | |
| **sp\_Xem\_NguoiThue** | **Khóa** | **sp\_Update\_NguoiThue** | **Khóa** |
| ***Input:*** @MaCN  ***Output***: Danh sách người thuê | ***Input***: @MaNT, @TenNT  ***Output***: Cập nhật lại tên mới của người thuê |
| SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITED** | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITED** |
| BEGIN TRAN |  |  |  |
| B1: thống kê danh sách người thuê  select \*  from NguoiThue  where MaCN=@MaCN | **R(NguoiThue)**  Xin khoá đọc trên bảng NguoiThue với đk MaCN=’ CN000001’ |  |  |
| WAITFOR DELAY '00:00:20' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | Update tên NguoiThue  update NguoiThue set TenNT = @TenNT where MaNT=@MaNT | **X(NguoiThue)**  Xin khóa ghi trên bảng NguoiThue |
|  |  | COMMIT |  |
| B2: thống kê danh sách người thuê  select \*  from NguoiThue  where MaCN=@MaCN  COMMIT |  |  |  |

Table

Description automatically generatedKết quả:

## Tình huống 8: Phantom

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Phantom***  T1 (User = ngườithue): thực hiện thống kê nhà thuê  T2 (User = chủ nhà): them 1 nha thuê  Vd: @MaNT=’NT000001’@MaCN=’ CN000001’ 🡪 T1 không đọc được update dữ liệu | | | |
| **sp\_Xem\_NhaThue** | **Khóa** | **Sp\_Them\_NhaThue** | **Khóa** |
| ***Input:***  ***Output***: Danh sách nhà thuê | ***Input***: @GiaThue,@MaNha,@SLphong,@Dang,@HetHan,  @HetHan,@SLXem,@TinhTrang  ***Output***: them 1 nhà thuê |
| SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITED** | SET TRANSACTION ISOLATION  LEVEL **READ COMMITED** |
| BEGIN TRAN |  |  |  |
| B1: thống kê danh sách nhà thuê  SELECT \*  FROM NhaThue | **R(NhaThue)**  Xin khoá đọc trên bảng NhaThue |  |  |
| WAITFOR DELAY '00:00:20' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Thêm 1 dòng mới vào bảng NhaThue  INSERT INTO NhaThue(GiaThue,MaNha,SoLuongPhong, NgayDang, NgayHetHan, SoLuotXem, TinhTrangThue) VALUES | **X(NhaThue)**  Xin khóa ghi trên bảng NhaThue |
|  |  | B2: Kiểm tra thông tin mã NhaThue có trong danh sách Nha và có đúng chủ nhà hay không  IF NOT EXISTS(SELECT \* FROM Nha WHERE MaNha=@maNha AND MaChuNha = @maChuNha)  BEGIN    RAISERROR('Them nha khong thanh cong',1,1)  ROLLBACK TRAN sp\_them\_NhaThue  END  ELSE  COMMIT TRAN sp\_them\_NhaThue | **X(NhaThue)**  Xin khóa ghi trên bảng NhaThue |
| B2: thống kê danh sách nhà thuê  SELECT \*  FROM NhaThue  COMMIT |  |  |  |

Table

Description automatically generatedKết quả:

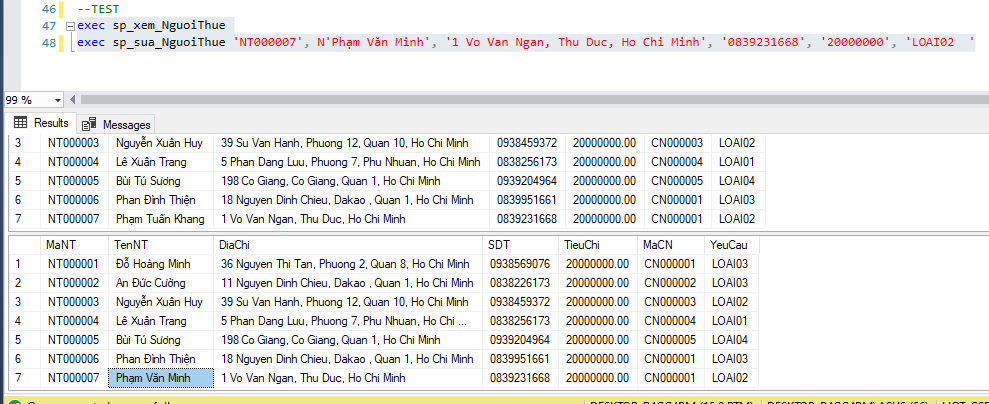
# Sinh viên thực hiện: 18120370

## Tình huống 9: Unrepeatable Read

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR09: *Unrepeatable Read***  T1 (User = NhanVien): Thực hiện xem danh sách NguoiThue  T2 (User = NguoiThue): Thực hiện cập nhật thông tin NguoiThue  Ví dụ: @maNT = ‘NT000001’ , @tenNT = N’Phạm Văn Minh’, @diaChi = ‘36 Nguyen Thi Tan, Phuong 2, Quan 8, Ho Chi Minh’, @sdt = ‘0938569076’, @tieuChi = ‘20000000’, @yeuCau = ‘LOAI03 ‘  Ở VD này một dòng dữ liệu trong NguoiThue được sửa khi đang được đọc bởi người dùng khiến lần đọc sau không còn thấy dữ liệu ban đầu nữa | | | |
| **sp\_xem\_NguoiThue** | **Khóa** | **sp\_sua\_NguoiThue** | **Khóa** |
| ***Input:***  ***Output***: Danh sách NguoiThue | ***Input***: @maNT, @tenNT, @diaChi, @sdt, @tieuChi, @yeuCau  ***Output***: Sửa 1 dòng trong bảng NguoiThue |
| SET TRANSACTION ISOLATION  LEVEL **READ COMMITED** | SET TRANSACTION ISOLATION  LEVEL **READ COMMITED** |
| BEGIN TRAN |  |  |  |
| B1: Đọc dữ liệu bảng NguoiThue  SELECT \* FROM NguoiThue | **R(NhaBan)** |  |  |
| WAITFOR DELAY '00:00: 07' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin mã NguoiThue có đúng không  IF NOT EXISTS(SELECT \* FROM NguoiThue WHERE MaNT = @maNT)  BEGIN  RAISERROR('Khong ton tai ma nguoi thue nay.',1,1)  ROLLBACK TRAN sp\_sua\_NguoiThue  END  ELSE |  |
|  |  | B2: Sửa dòng dữ liệu có MaNT = @maNT  UPDATE NguoiThue SET TenNT = @tenNT, DiaChi = @diaChi, SDT = @sdt, TieuChi = @tieuChi, YeuCau = @yeuCau WHERE MaNT = @maNT  COMMIT TRAN sp\_sua\_NguoiThue | **X(NguoiThue)** |
| B2: Đọc dữ liệu bảng NguoiThue  SELECT \* FROM NguoiThue | **R(NhaBan)** |  |  |
| COMMIT TRAN |  |  |  |
| ***Input:***  ***Output***: Danh sách NguoiThue |  | ***Input***: @maNT, @tenNT, @diaChi, @sdt, @tieuChi, @yeuCau  ***Output***: Sửa 1 dòng trong bảng NguoiThue |  |

Kết quả: Hai lần đọc của T1 cho ra 2 kết quả khác nhau tại dòng có MaNT = ‘NT000007’ với TenNT của lần 1 là Phạm Tuấn Khang, lần 2 là Phạm Văn Minh (mới được insert) à Unrepeatable Read

* **User**: Nhân viên

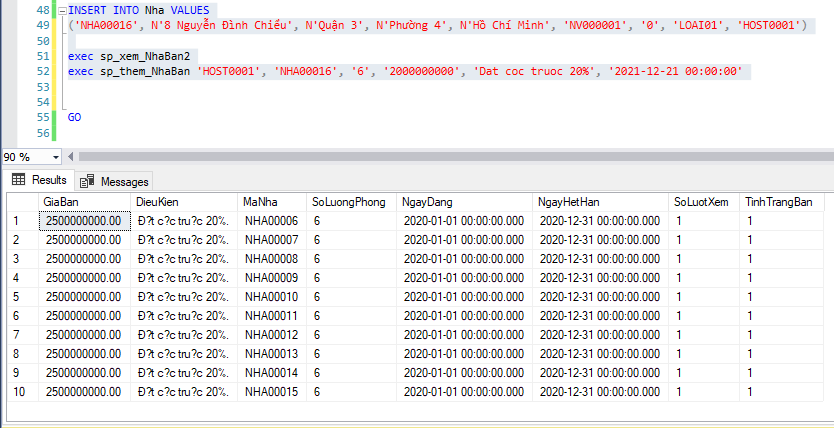


## Tình huống 10: Phantom

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR10: *Phantom***  T1 (User = NguoiThue): Thực hiện xem danh sách Nhà  T2 (User = ChuNha): Thực hiện thêm NhaBan  VD: @maChuNha = ‘HOST0001’, @maNha = ‘NHA00016’, @soLuongPhong = ‘6’, @giaBan = ‘2000000000’, @dieuKien=’Dat coc truoc 20%’, @ngayHetHan = ‘2021-12-21 00:00:00’  Ở VD này danh sách nhà bán được thêm 1 dòng trong khi đang được xem bởi người dùng khiến cho dữ liệu đọc lên bị sai (thiếu) | | | |
| **sp\_xem\_NhaBan** | **Khóa** | **sp\_them\_NhaBan** | **Khóa** |
| ***Input:***  ***Output***: Danh sách NhaBan | ***Input***: @maChuNha, @maNha, @soLuongPhong, @giaBan, @dieuKien, @ngayHetHan  ***Output***: Thêm 1 dòng trong danh sách NhaBan |
| SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITED** | SET TRANSACTION ISOLATION  LEVEL **READ COMMITED** |
| BEGIN TRAN |  |  |  |
| B1: Đọc dữ liệu bảng NhaBan  SELECT \* FROM NhaBan |  |  |  |
| WAITFOR DELAY '00:00: 07' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Thêm 1 dòng mới vào bảng NhaBan  INSERT INTO NhaBan(MaNha,SoLuongPhong,GiaBan,DieuKien,NgayDang, NgayHetHan, SoLuotXem, TinhTrangBan) VALUES (@maNha, @soLuongPhong ,@giaBan, @dieuKien, GETDATE(), @ngayHetHan, 0, 1) | **X(NhaBan)** |
|  |  | B2: Kiểm tra thông tin mã NhaBan có trong danh sách Nha và có đúng chủ nhà hay không  IF NOT EXISTS(SELECT \* FROM Nha WHERE MaNha=@maNha AND MaChuNha = @maChuNha)  BEGIN    RAISERROR('Them nha ban khong thanh cong',1,1)  ROLLBACK TRAN sp\_them\_NhaBan  END  ELSE  COMMIT TRAN sp\_them\_NhaBan |  |
| B2: Đọc dữ liệu bảng NhaBan  SELECT \* FROM NhaBan |  |  |  |
| COMMIT TRAN |  |  |  |
| ***Input:***  ***Output***: Danh sách NhaBan |  | ***Input***: @maChuNha, @maNha, @soLuongPhong, @giaBan, @dieuKien, @ngayHetHan  ***Output***: Thêm 1 dòng trong danh sách NhaBan |  |
| SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITED** |  | SET TRANSACTION ISOLATION  LEVEL **READ COMMITED** |  |

Kết quả: NhaBan được thêm 1 dòng dữ liệu (NHA00016) nhưng T1 lại không đọc được dòng dữ liệu mới thêm vào à Phantom

**User**: Người thuê



* **User**: Chủ nhà

