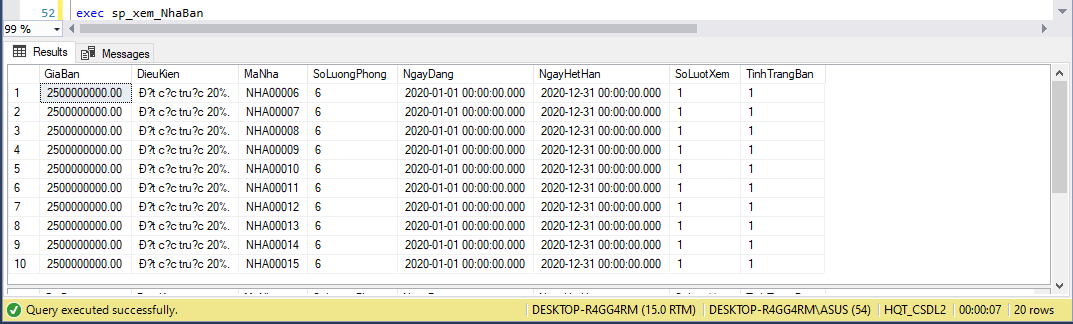
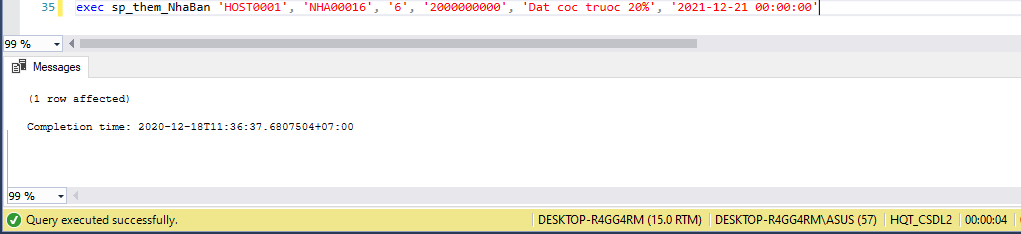
|  |  |  |  |
| --- | --- | --- | --- |
| **ERR??:  *Phantom***  T1 (User = NguoiThue): Thực hiện xem danh sách Nhà  T2 (User = ChuNha): Thực hiện thêm NhaBan  VD: @maChuNha = ‘HOST0001’, @maNha = ‘NHA00016’, @soLuongPhong = ‘6’, @giaBan = ‘2000000000’, @dieuKien=’Dat coc truoc 20%’, @ngayHetHan = ‘2021-12-21 00:00:00’  Ở VD này, nếu lỗi Phantom xảy ra thì 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). Thiết đặt Transaction Isolation Level là Serializable tại T1(sp\_xem\_NhaBan) để ngăn lỗi. | | | |
| 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 SERIALIZABLE** | SET TRANSACTION ISOLATION  LEVEL **READ COMMITED** |
| BEGIN TRAN |  |  |  |
| B1: Đọc dữ liệu bảng NhaBan  SELECT \* FROM NhaBan | **R(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 | **R(NhaBan)** |  |  |
| COMMIT TRAN |  |  |  |

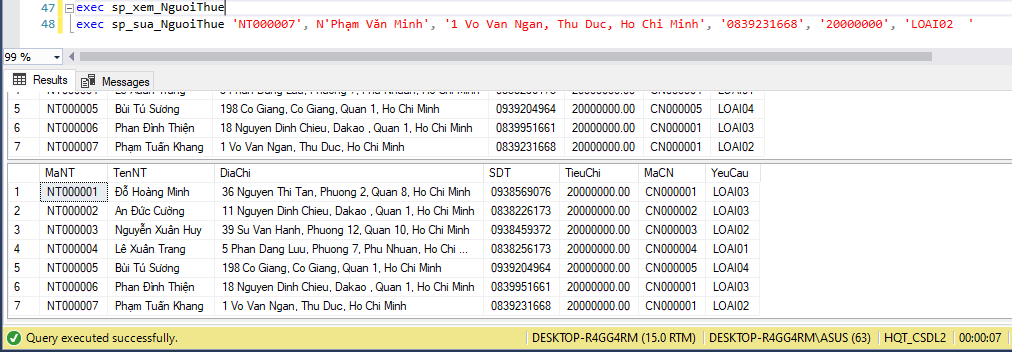
Kết quả: T2 phải chờ cho đến khi T1 commit thì mới được phép thêm dữ liệu trên bảng NhaBan

T1:

T2: Thời gian thực hiện dài hơn vì phải chờ T1.

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR??:  *Unrepeatable read***  T1 (User = NhanVien): Thực hiện xem danh sách NguoiThue  T2 (User = NguoiThue): Thực hiện cập nhật thông tin NguoiThue  VD: @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 nếu lỗi Unrepaeatable read xảy ra thì 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. Thiết đặt Transaction Isolation Level thành Repeatable Read để ngăn lỗi. | | | |
| 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 **REPEATABLE READ** | SET TRANSACTION ISOLATION  LEVEL **READ COMMITED** |
| BEGIN TRAN |  |  |  |
| B1: Đọc dữ liệu bảng NguoiThue  SELECT \* FROM NguoiThue | **S(NguoiThue)** |  |  |
| 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 | **S(NguoiThue)** |  |  |
| COMMIT TRAN |  |  |  |

Kết quả: Hai lần đọc của T1 cho ra 2 kết quả giống nhau tại dòng có MaNT = ‘NT000007’. T2 phải chờ T1 đọc xong mới được phép sửa dữ liệu trong bảng người thuê.

T1: 

T2: 