**d**

BÁO CÁO ĐỒ ÁN THỰC HÀNH

MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

GVHD: ……….

# **THÔNG TIN NHÓM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ tên** | **Công việc** | **% Hoàn thành** |
| **1** |  |  |  |  |
| **2** |  |  |  |  |
| **3** |  |  |  |  |
| **4** |  |  |  |  |
| **5** |  |  |  |  |

# CÀI ĐẶT TÌNH HUỐNG TRANH CHẤP

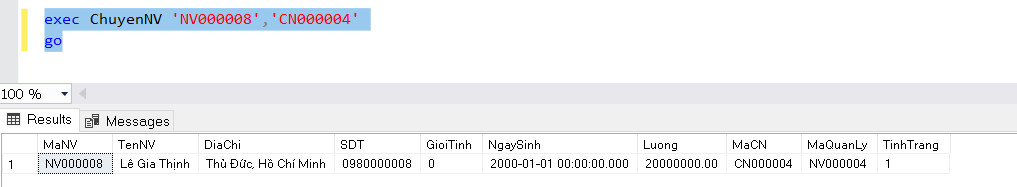
# Sinh viên thực hiện:…..

## 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 trường hợp: CEO chuyển công tác cho nhân viên với mã là NV000008, update Mã chi nhánh mới CN000004 và dựa trên Mã chi nhánh mới mà update Mã nhân viên quản lý theo chi nhánh đó. Tại thời điểm đó, QLCN cũng đang thống kê nhân viên ở các chi nhánh.  Khắc phục: vì uncommited sẽ gây ra Diry read nên chỉnh lại Isolation level của T2 thành Commited để không đọc phải dữ liệu rác và ta sẽ có kết quả đúng | | | |
| **sp\_ChuyenNV\_uncommited** | **Khóa** | **sp\_XemNV** | **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 COMMITTED |
| BEGIN TRAN |  |  |  |
| B1: Cập nhập chi nhánh mới  update NhanVien  set MaCN=@mcn  where MaNV=@manv | **X(Nhân viên)**  Xin khóa trên nhân viên |  |  |
| B2: Cập nhập nhân viên quản lý mới  update NhanVien  set MaQuanLy=(select distinct MaNV from ChiNhanh where MaCN=@mcn) where MaNV=@manv |  |  |  |
| B3: Thống kê Nhân viên  select \* from NhanVien where MaNV=@manv | R(Nhân viên) |  |  |
| 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 |  |  |  |
| COMMIT | Mở khóa ở bảng nhân viên |  |  |
|  |  | Begin Tran |  |
|  |  | B1: Thống kê nhân viên  select \* from NhanVien | *R(Nhân Viên)*  *Xia khóa ở bảng nhân viên* |
|  |  | Commit |  |

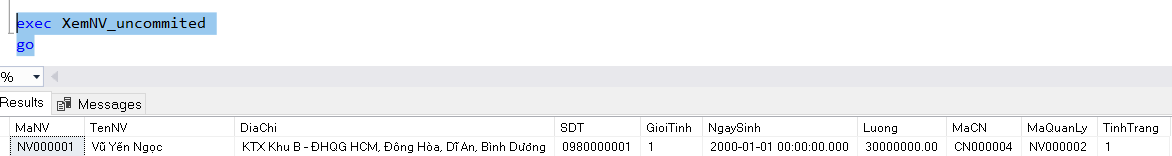
Kết quả: Dựa vào kết quả bên dưới ta thấy được đáng lẽ QLCN phải thấy được cả hai yếu tố update của CEO nhưng lần này sẽ thấy được mã chi nhánh lẫn mã nhân viên được cập nhập theo => Fix thành công lỗi Dirty read

User: CEO

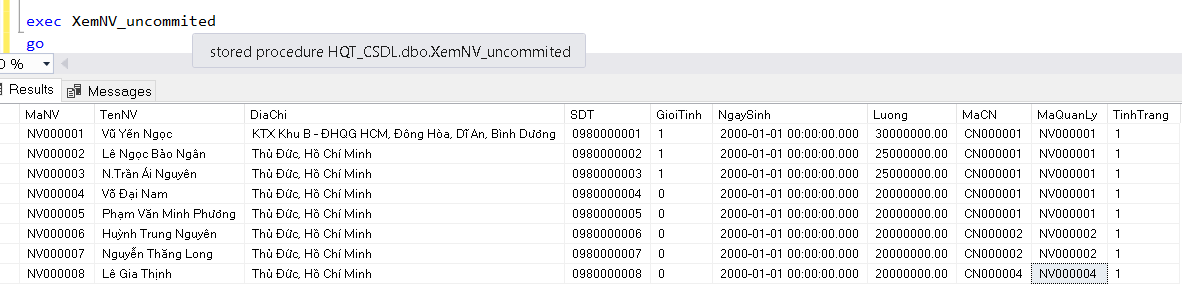


User: QLCN

Trước khi sửa:



Sau khi Sửa:

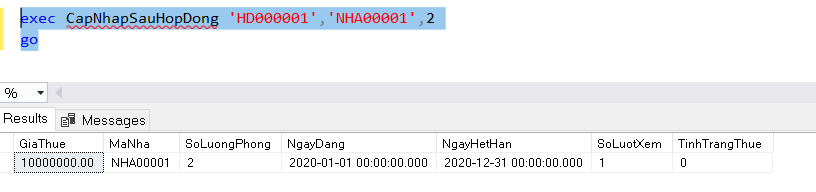


## 2. Tình huống 2: Lost Update

|  |  |  |  |
| --- | --- | --- | --- |
| **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 trường hợp: Sau khi làm hợp đồng cho thuê phòng xong của căn nhà có mã là NHA00001, nhân viên sẻ cập nhập lại số phòng=4 – số phòng đã cho thuê theo hợp đồng (ở đây là 2 phòng). Tại thời điểm đó chủ nhà bổng dư phòng nên chủ nhà cập nhập thêm 4 phòng của nhà mình.  Khắc phục: Sử dụng Isolation level Repeatable Read để giải quyết Lost data  Sử dụng Uplock để giải quyết Deadlock ở T2 khi T2 đang chuẩn bị Update | | | |
| **sp\_CapNhapSau** | **Khóa** | **sp\_CapNhap** | **Khóa** |
| ***Input*** @mhd, @manha, @sphd  ***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 with(Uplock) where MaNha=@manha | **R(Nhà thuê)// Xin khóa đọc tại Nhà thuê** |  |  |
| B2: Cập nhập số phòng mới  update NhaThue  set SoLuongPhong=@temp-@sphd  where MaNha=@manha | X(Nhà thuê)//Xin khóa ghi |  |  |
| B3: Thống kê lại số phòng  select \* from NhaThue where MaNha=@manha | R(Nhà thuê) |  |  |
| if(not exists(select \* from HopDong where MaHD=@mhd) or not exists(select\* from NhaThue where MaNha=@manha))  begin  RAISERROR('Cap Nhap khong thanh cong',1,1)  ROLLBACK TRAN sp\_CapNhapSau  End |  |  |  |
| Commit tran | //Trả khóa lại |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Đọc dữ liệu tại bảng nhà thuê  declare @temp int  select @temp=SoLuongPhong from NhaThue with Uplock where MaNha=@manha | **R(Nhà thuê)**  //Xin khóa đọc |
|  |  | B2: Cập nhập số phòng mới  update NhaThue  set SoLuongPhong=@temp+@spt  where MaNha=@manha | X(Nhà thuê)// Xin khóa ghi |
|  |  | if(not exists(select\* from NhaThue where MaNha=@manha))  begin  RAISERROR('Cap Nhap khong thanh cong',1,1)  ROLLBACK TRAN sp\_CapNhap  End |  |
|  |  | COMMIT |  |

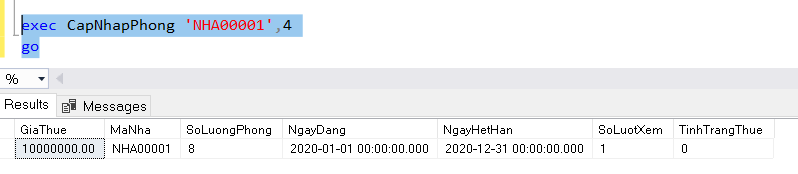
Kết quả: Dựa vào hình ở dưới ta thấy ở nhân viên số phòng là 2, ở chủ nhà số phòng là 8, điều này vô lý bởi vì số phòng đáng lẽ là 4+4-2=6 mà ở đây là 2 và 6 => Khắc phục Lost Update

User: Nhân viên



User: Chủ nhà

Trước:



Sau:

