

BÁO CÁO ĐỒ ÁN THỰC HÀNH
MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU
GVHD:Tiết Gia Hồng

THÔNG TIN NHÓM

XỬ LÝ TÌNH HUỐNG TRANH CHẤP

I. Sinh viên thực hiện: Trần Minh Bảo

1. Tình huống 1: Xem danh sách nhà ở session 1 – Xoá một nhà ở một session 2 với id x – Xem chi tiết nhà ở session 1 với id x

ERR01: Phantom T1 : thực hiện xem thông tin nhà. T2 : thực hiện xoá 1 nhà.			
House_GetAll	Khóa	sp_XoaMotNha	Khóa
Input: Output:		Input: @pId int Output:	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
B1: Xem danh sách nhà cho thuê SELECT * FROM HOUSE	R(House) //Xin khoá đọc trên bảng House		
		BEGIN TRAN	
		B1: Xoá một nhà T1 chuẩn bị xem DELETE FROM [HOUSE] WHERE [ID] = @id	X(House) //Xin khoá ghi trên bảng House
		COMMIT	
sp_XemThongTinNha			
Input: @pId int Output:			
SET TRANSACTION ISOLATION LEVEL READ COMMITTED			
B2: Xem thông tin nhà T2 vừa xoá			

<pre> SELECT HOU.[ID], HOU.[RoomNumber], HOU.[FeeMonth], HOU.[IDHouseholder], HHD.[Name] AS HouseholderName, HOU.[IDEmployee], EMP.[Name] AS EmployeeName, HOU.[IDHouseType], HUT.[Name] AS HouseTypeName, HOU.[IDLocation], LCT.[StreetName] + ', ' + LCT.[District] + ', ' + LCT.[City] AS [Address], LCT.[Region], HOU.[IsActive] FROM House HOU LEFT JOIN Employee EMP ON EMP.ID = HOU.IDEmployee LEFT JOIN HouseType HUT ON HUT.ID = HOU.IDHouseType LEFT JOIN Householder HHD ON HHD.ID = HOU.IDHouseholder LEFT JOIN Location LCT ON LCT.ID = HOU.IDLocation WHERE HOU.[IsActive] = 1 and HOU.[ID] = @id </pre>			
Không tìm thấy nhà			

- Đăng nhập ứng dụng với một session – Gọi là session 1
- Mở danh sách nhà với session 1
- Đăng nhập ứng dụng với một session – Gọi là session 2
- Xoá nhà với id = 4 với session 2
- Xem chi tiết nhà với id = 4 với session 1 => Không tìm thấy vì đã bị xoá

2. Tình huống 2: Cập nhật thông tin nhà giữa 2 session khác nhau (Tài khoản nhân viên hoặc admin)

ERR03: Lost Updated T1 : Chính sửa thông tin nhà T2 : Chính sửa thông tin nhà			
House_InsUpd	Khóa	House_InsUpd	Khóa
Input: @pRoomNumber int, @pFeeMonth bigint, @pIDHouseholder bigint, @pIDEmployee bigint, @pIDHouseType bigint, @pIDLocation bigint Output:		Input: @pRoomNumber int, @pFeeMonth bigint, @pIDHouseholder bigint, @pIDEmployee bigint, @pIDHouseType bigint, @pIDLocation bigint Output:	
SET TRANSACTION ISOLATION LEVEL READ COMMITED		SET TRANSACTION ISOLATION LEVEL READ COMMITED	
BEGIN TRAN			
B1: Sửa thông tin nhà UPDATE [dbo].[House] SET [RoomNumber] = @pRoomNumber, [FeeMonth] = @pFeeMonth, [IDHouseholder] = @pIDHouseholder, [IDEmployee] = @pIDEmployee, [IDHouseType] = @pIDHouseType, [IDLocation] = @pIDLocation WHERE [ID] = @pID	X(House) //Xin khoá ghi trên bảng House		
		BEGIN TRAN	
		B1: Sửa thông tin nhà UPDATE [dbo].[House] SET [RoomNumber] = @pRoomNumber, [FeeMonth] = @pFeeMonth, [IDHouseholder] = @pIDHouseholder,	X(House) //Xin khoá ghi trên bảng House

		[IDEmployee] = @pIDEmployee, [IDHouseType] = @pIDHouseType, [IDLocation] = @pIDLocation WHERE [ID] = @pID	
		COMMIT	
COMMIT			
		sp_XemThongTinNha	
		<u>Input:</u> @pID int <u>Output:</u>	
		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
		B2: Xem thông tin nhà T2 vừa cập nhật SELECT HOU.[ID], HOU.[RoomNumber], HOU.[FeeMonth], HOU.[IDHouseholder], HHD.[Name] AS HouseholderName, HOU.[IDEmployee], EMP.[Name] AS EmployeeName, HOU.[IDHouseType], HUT.[Name] AS HouseTypeName, HOU.[IDLocation], LCT.[StreetName] + ', ' + LCT.[District] + ', ' + LCT.[City] AS [Address], LCT.[Region], HOU.[IsActive] FROM House HOU LEFT JOIN Employee EMP ON EMP.ID = HOU.IDEmployee LEFT JOIN HouseType HUT ON HUT.ID = HOU.IDHouseType	

		LEFT JOIN Householder HHD ON HHD.ID = HOU.IDHouseholder LEFT JOIN Location LCT ON LCT.ID = HOU.IDLocation WHERE HOU.[IsActive] = 1 and HOU.[ID] = @id	
		Mất dữ liệu cập nhật tại T2	

- Đăng nhập ứng dụng với session 1
- Vào tab chỉnh sửa một nhà với id x ở session 1 nhưng chưa lưu lại
- Đăng nhập ứng dụng với session 2
- Vào tab chỉnh sửa một nhà với id x và lưu lại ở session 2
- Lưu dữ liệu vừa chỉnh sửa ở session 1
- Xem chi tiết nhà tại session 2 => Mất dữ liệu đã cập nhật

II. Sinh viên thực hiện: Vũ Mạnh Hùng

1. Tình huống 1: Cập nhật – Xem thông tin lương của nhân viên

ERR01: Dirty read T1: User thực hiện cập nhật thông tin nhà T2: User thực hiện xem thông tin nhà			
Employee_InsUpd Input: Danh sách các tham số @pID bigint = -1, @pName nvarchar (255) = '', @pAddress nvarchar (255) = '', @pPhone varchar (15) = '', @pGender bit = 1, @pBirthday datetime = NULL, @pSalary bigint = 0, @pIDBranchOffice bigint = NULL, @pPassword varchar (255) = '' Output: Kết quả Insert – Update	Khóa	Employee_Search Input: Danh sách tham số @pID bigint = '', @pName varchar (255) = '', @pAddress varchar (255) = '', @pPhone varchar (15) = '', @pGender bit = -1, @pBirthday varchar (10) = '', @pSalary bigint = -1, @pIDBranchOffice bigint = NULL Output: Danh sách thỏa điều kiện tìm kiếm	Khóa

		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRANSACTION			
B1: Khai báo biến lưu thông báo trả về DECLARE @vMessage nvarchar (max) SET @vMessage = ''			
B2: Kiểm tra tham số @pID truyền vào IF EXISTS (@pID = -1) BEGIN B3.1.1: Thêm dữ liệu vào bảng nhân viên INSERT INTO [dbo].[Employee]([Name], [Address], [Phone], [Gender], [Birthday], [Salary], [IDBranchOffice], [IsActive]) VALUES (@pName, @pAddress, @pPhone, @pGender, @pBirthday, @pSalary, @pIDBranchOffice, 1) IF @@Error <> 0 GOTO ABORT	X(Employee)		
B3.1.2: Thêm dữ liệu vào bảng người dùng INSERT INTO [dbo].[Users]([UserName], [Password], [IsAdmin], [IsActive]) VALUES (IDENT_CURRENT ('Employee'), @pPassword, 0,	X(Users)		

<pre> 1) IF @@Error <> 0 GOTO ABORT END ELSE BEGIN B3.2: Cập nhật dữ liệu nhân viên UPDATE [dbo].[Employee] SET [Name] = @pName, [Address] = @pAddress, [Phone] = @pPhone, [Gender] = @pGender, [Birthday] = @pBirthday, [Salary] = @pSalary, [IDBranchOffice]= @pIDBranchOffice WHERE [ID] = @pID IF @@Error <> 0 GOTO ABORT END </pre>	X(Employee)		
		BEGIN TRANSACTION	
		<pre> B1: Lấy danh sách thỏa SELECT EMP.[ID], EMP.[Name], EMP.[Address], EMP.[Phone], EMP.[Gender], EMP.[Birthday], EMP.[Salary], EMP.[IDBranchOffice], BRO.[Name] AS BranchOfficeName, EMP.[IsActive] FROM Employee EMP LEFT JOIN BranchOffice BRO ON BRO.ID = EMP.IDBranchOffice WHERE EMP.[IsActive] = 1 AND (EMP.[ID] = @pID OR @pID IS NULL OR @pID = -1) AND (EMP.[Name] LIKE N'%' + @pName + '%' OR @pName IS NULL OR @pName = '') </pre>	R(Employee)

		<pre> AND (EMP.[Address] LIKE N'%' + @pAddress + '%' OR @pAddress IS NULL OR @pAddress = '') AND (EMP.[Phone] LIKE N'%' + @pPhone + '%' OR @pPhone IS NULL OR @pPhone = '') AND (EMP.[Gender] = @pGender OR @pGender IS NULL OR @pID = -1) AND (CONVERT(varchar(10), EMP.[Birthday], 103) = @pBirthday OR @pBirthday IS NULL OR @pBirthday = CONVERT(varchar(10), getdate(), 103)) AND (EMP.[Salary] = @pSalary OR @pSalary IS NULL OR @pSalary = -1) AND (EMP.[IDBranchOffice] = @pIDBranchOffice OR @pIDBranchOffice IS NULL OR @pIDBranchOffice = -1) </pre>	
		COMMIT TRANSACTION	
<p>B4.1: Thành công trả kết quả</p> <pre> SELECT 1 AS Result, @vMessage ErrorDesc RETURN 1 </pre> <p>ABORT:</p> <pre> BEGIN </pre> <p>B4.1: Thất bại delay rồi Rollback – Trả kết quả</p> <p>ROLLBACK TRANSACTION</p> <pre> SET @vMessage = 'Thêm nhân viên thất bại!' SELECT 0 AS Result, @vMessage ErrorDesc RETURN 0 </pre> <p>END</p>			
<p>Cài đặt Trigger để phát sinh lỗi</p> <pre> IF((Select Salary From inserted) < 0) BEGIN WAITFOR DELAY '0:0:5' ROLLBACK TRAN </pre> <p>END</p>			

2. Tình huống 2:

ERR01: Dirty read T1 (User = quản lý): thực hiện thêm 1 thông tin chuyển xe vào ngày A. T2 (User = tổ trưởng): thực hiện thống kê số chuyển xe sẽ thực hiện trong ngày A			
sp_Them1ChuyenXe	Khóa	sp_ThongKeChuyenXe-CoDieuKien	Khóa
Input: Output:		Input: Output:	

III. Sinh viên thực hiện: Trần Văn Khuê

1. Tình huống: sửa – xem thông tin người thuê nhà

ERR01: Dirty read T1 (User = nhân viên 1 (chi nhánh B1)): thực hiện giao tác sửa thông tin chi tiết của người thuê nhà A T2 (User = nhân viên 2 (B1)): thực hiện giao tác lấy thông tin chi tiết của người thuê nhà A			
sp_SuaChiTietNguoiThueNha	Khóa	sp_ChiTietNguoiThueNha	Khóa
Input: @pId nvarchar(255) = " Output: NguoiThueNha (1 row)		Input: @pId (nvarchar) Output: NguoiThueNha (1 row)	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ		SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED	
BEGIN TRAN			
B1: Cập nhật thông tin vào bảng NguoiThueNha với điều kiện maNguoiThue=@maNguoiThue UPDATE NguoiThueNha SET tenNguoiThue='B' WHERE maNguoiThue=@maNguoiThue	X(A)		
WAITFOR DELAY '00:00:20'			
		BEGIN TRAN	
		B1: Lấy thông tin người thuê nhà SELET * FROM NguoiThueNha WHERE maNguoiThue = @maNguoiThue	R(A) //Không cần xin khoá
		COMMIT	
IF @error <> 0			

SELECT SELECT N'Không thể cập nhật người thuê nhà' ROLLBACK TRAN RETURN End			

2. Tình huống: sửa – xem chi tiết nhà

ERR02: Dirty read T1 (User = nhân viên 1 (chi nhánh B1)): thực hiện giao tác sửa thông tin chi tiết của nhà A T2 (User = nhân viên 2 (B1)): thực hiện giao tác lấy thông tin chi tiết của nhà A			
sp_SuaChiTietNha	Khóa	sp_ChiTietNha	Khóa
Input: @pID (bigint) @RoomNumber int Output: Nha (1 row)		Input: pID (bigint) Output: Nha (1 row)	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ		SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED	
BEGIN TRAN			
B1: Cập nhật thông tin vào bảng Nha với điều kiện maNha = @maNha UPDATE Nha SET soluongPhong=5 WHERE maNha=@maNha	X(A)		
WAITFOR DELAY '00:00:20'			
		BEGIN TRAN	
		B1: Lấy thông tin người thuê nhà SELET * FROM Nha WHERE maNha = @maNha	R(A) //Không cần xin khoá

		COMMIT	
IF @error <> 0 SELECT SELECT N'Không thể cập nhật nhà' ROLLBACK TRAN RETURN End			

3. Tình huống: sửa – sửa thông tin chi tiết người thuê nhà

ERR03: Lost Updated T1 (User = nhân viên 1 (chi nhánh B1)): thực hiện giao tác sửa thông tin chi tiết của người thuê nhà A T2 (User = nhân viên 2 (B1)): thực hiện giao tác sửa thông tin chi tiết của người thuê nhà A			
sp_SuaChiTietNguoiThueNha	Khóa	sp_ChiTietNguoiThueNha	Khóa
Input: @pID (bigint) @Name nvarchar Output: NguoiThueNha (1 row)		Input: @pID (bigint) @Name(nvarchar) Output: NguoiThueNha (1 row)	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Cập nhật thông tin vào bảng NguoiThueNha với điều kiện maNguoiThue=@maNguoiThue UPDATE NguoiThueNha SET tenNguoiThue='B' WHERE maNguoiThue=@maNguoiThue	X(A)		
WAITFOR DELAY '00:00:20'			
		BEGIN TRAN	

		B1: Cập nhật thông tin vào bảng NguoiThueNha với điều kiện maNguoiThue=@maNguoiThues UPDATE NguoiThueNha SET tenNguoiThue ='C' WHERE maNguoiThue=@maNguoiThue	X(A)
		COMMIT	
SELECT * FROM NguoiThueNha Where maNguoiThue = @maNguoiThue			

4. Tình huống: sửa – sửa thông tin chi tiết nhà

ERR04: Lost Updated T1 (User = nhân viên 1 (chi nhánh B1)): thực hiện giao tác sửa thông tin chi tiết của nhà A T2 (User = nhân viên 2 (B1)): thực hiện giao tác sửa thông tin chi tiết của nhà A			
sp_SuaChiTietNha	Khóa	sp_ChiTietNha	Khóa
<u>Input:</u> @pId (bigint) @RoomNumber int <u>Output:</u> Nha (1 row)		<u>Input:</u> @pId (bigint) @RoomNumber int <u>Output:</u> Nha (1 row)	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Cập nhật thông tin vào bảng Nha với điều kiện maNha=@maNha UPDATE Nha SET soluongPhong =4 WHERE maNha=@maNha	X(A)		
WAITFOR DELAY '00:00:20'			
		BEGIN TRAN	

		B1: Cập nhật thông tin vào bảng Nha với điều kiện maNha = @maNha UPDATE Nha SET soluongPhong=5 WHERE maNha=@maNha	X(A)
		COMMIT	
SELECT * FROM Nha Where maNha = @maNha			