

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

THÔNG TIN NHÓM

STT	MSSV	Họ tên	Công việc	% Hoàn thành
1	22424001	Trần Quốc Bảo	Tình huống 1, 2	100%
2	22424006	Lê Hà Giang	Tình huống 3, 4	100%
3	22424025	Phạm Minh Nhựt	Tình huống 5, 6	100%
4	22424034	Phạm Duy Tùng	Tình huống 7, 8	100%
5	22424037	Võ Minh Trung	Tình huống 9, 10	100%

CÀI ĐẶT TÌNH HUỐNG TRANH CHẤP

I. Sinh viên thực hiện: Trần Quốc Bảo

1. Tình huống 1: AD6 - AD5

ERR01: Phantom T1 (User = nhân viên): thực hiện xem danh sách các nhà T2 (User = nhân viên): thực hiện thêm nhà			
sp XemNha1	Khóa	sp ThemNha1	Khóa
<u>Input:</u> @Quan, @ThanhPho <u>Output:</u>		<u>Input:</u> @Duong, @Quan, @ThanhPho, @KhuVuc, @SLPhong, @TienThue, @NgayDang, @NgayHetHan, @TinhTrang, @LuotXem, @GiaBan, @DieuKienCN, @MaLoaiNha, @MaChuNha, @MaNhanVien, @MaChiNhanh <u>Output:</u>	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Kiểm tra thông tin (1) IF NOT EXISTS (SELECT * FROM NHA WHERE Quan= @Quan AND ThanhPho = @ThanhPho) BEGIN PRINT N'NHÀ KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 1 END	S(NHA) //Xin khoá đọc trên bảng NHA //Trả khóa		
B2: Hiển thị tổng số nhà DECLARE @TONG INT SELECT @TONG = COUNT(*) FROM NHA	S(NHA)		

WHERE Quan= @Quan AND ThanhPho = @ThanhPho PRINT @QUAN + N' CÓ ' + CAST(@TONG AS NVARCHAR(20)) + N' CĂN NHÀ'	//Xin khoá đọc trên bảng NHA //Trả khóa		
WAITFOR DELAY '00:00:20'			
		BEGIN TRAN	
		B1: Thêm nhà INSERT NHA(Duong, Quan, ThanhPho, KhuVuc, SLPhong, TienThue, NgayDang, NgayHetHan, TinhTrang, LuotXem, GiaBan, DieuKienCN, MaLoaiNha, MaChuNha, MaNhanVien, MaChiNhanh) VALUES (@Duong, @Quan, @ThanhPho, @KhuVuc, @SLPhong, @TienThue, @NgayDang, @NgayHetHan, @TinhTrang, @LuotXem, @GiaBan, @DieuKienCN, @MaLoaiNha, @MaChuNha, @MaNhanVien, @MaChiNhanh)	X(NHA) /Xin khoá ghi trên bảng NHA
		COMMIT	//Trả khóa
B2: Hiển thị danh sách nhà SELECT * FROM NHA WHERE Quan= @Quan AND ThanhPho = @ThanhPho	S(NHA) //Xin khoá đọc trên bảng NHA //Trả khóa		
COMMIT			

2. Tình huống 2: AD5 - CU1

ERR02: Dirty read T1 (User = nhân viên): thực hiện cập nhật địa chỉ nhà T2 (User = khách hàng): thực xem thông tin nhà			
sp_CapNhatTenDuong	Khóa	sp_XemNha2	Khóa
Input: @MaNha, @Duong Output:		Input: @MaNha Output:	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED	
BEGIN TRAN			
B1: Kiểm tra thông tin (1) IF NOT EXISTS (SELECT * FROM NHA WHERE MaNha = @MaNha) BEGIN PRINT N'NHÀ KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 1 END	S(NHA) //Xin khoá đọc trên bảng NHA //Trả khóa		
B2: Update tên đường UPDATE NHA SET Duong = @Duong WHERE MaNha = @MaNha	X(NHA) //Xin khoá ghi trên bảng NHA //Trả khóa khi commit		
WAITFOR DELAY '00:00:20'			
		BEGIN TRAN	
		B1: Kiểm tra thông tin (1) IF NOT EXISTS (SELECT * FROM NHA WHERE MaNha = @MaNha)	//Đọc không cần xin khóa

		BEGIN PRINT N'NHÀ KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 1 END	
		B2: Xem chi tiết nhà SELECT * FROM NHA WHERE MaNha = @MaNha	//Đọc không cần xin khóa
		COMMIT	
B3: Kiểm tra lại dữ liệu DECLARE @RT INT SELECT @RT = COUNT(*) FROM NHA WHERE Duong = @Duong IF @RT >= 2 BEGIN ROLLBACK TRAN RETURN 1 END //Khi update đến bảng NHA: Nếu tên đường đã có trong bảng NHA, nhiều nhà không thể có cùng tên đường (địa chỉ). Điều này dẫn đến toàn dữ liệu sẽ bị ROLLBACK ROLLBACK	S(NHA) //Xin khoá đọc trên bảng NHA //Trả khóa		

II. Sinh viên thực hiện: Lê Hà Giang

3. Tình huống 3: AD5 -AD5

ERR01: Lost update T1 (User = nhân viên): Nhân viên tăng giá thuê nhà T2 (User = nhân viên): Nhân viên tăng giá thuê nhà			
sp_TangGiaThueNha	Khóa	sp_TangGiaThueNha	Khóa
Input: @MaNha, @PhanTramGia Output:		Input: @MaNha, @PhanTramGia Output:	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Kiểm tra thông tin Set @GiaThue = (SELECT GiaThue FROM NHA WHERE MaNha=@MaNha)	S(NHA) //Xin khoá đọc trên bảng NHA //Nhả khóa		
		B1: Kiểm tra thông tin Set @GiaThue = (SELECT GiaThue FROM NHA WHERE MaNha=@MaNha)	S(A) //Xin khoá đọc trên bảng NHA //Trả khóa
B2: Giảm tiền thuê nhà SET @GiaThue = @GiaThue * ((100+@PhanTramGia)/100)			
B3: Update giá thuê nhà BEGIN TRY UPDATE NHA SET GiaThue = @GiaThue WHERE MaNha = @MaNha END TRY	X(NHA) //Xin khóa ghi X trên bảng NHA		

WAITFOR DELAY '00:00:20'			
		B2: Tăng tiền thuê nhà SET @GiaThue = @GiaThue * ((100+@PhanTramGia)/100)	
		B3: Update giá thuê nhà BEGIN TRY UPDATE NHA SET GiaThue = @GiaThue WHERE MaNha = @MaNha END TRY	X(NHA) // Xin khóa ghi X trên bảng NHA
COMMIT	//Trả khoá X		
		COMMIT	//Trả khoá X
		//T2 thay đổi giá thuê nhà với dữ liệu cũ trước khi thay đổi giá thuê ở T1	

4. Tình huống 4: AD4 -AD7

ERR01: Unrepeatable read T1 (User = nhân viên): Thực hiện lấy danh sách chủ nhà T2 (User = nhân viên): Thực hiện xóa 1 chủ nhà			
sp LayDanhSachChuNha	Khóa	sp XoaChuNha	Khóa
Input: @Quan Output: Danh sách chủ nhà		Input: @MaChuNha Output:	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Kiểm tra thông tin IF NOT EXISTS (SELECT *	S(CHUNHA) //Trả khóa		

<pre> FROM CHUNHA WHERE ĐịaChi LIKE N'% ' + @QUAN + '%' BEGIN PRINT N'chủ nhà không tồn tại' ROLLBACK TRAN RETURN END </pre>			
<pre> B2: Kiểm tra thông tin SET @TONG = (SELECT COUNT(*) FROM CHUNHA WHERE ĐịaChi LIKE N'% ' + @QUAN + '%') BEGIN PRINT @Quan + N'CÓ' + CAST(@TONG AS NVARCHAR(20)) + N'CHỦ NHÀ' END </pre>	<pre> S(CHUNHA) //Trả khóa </pre>		
<pre> WAITFOR DELAY '00:00:20' </pre>			
.		<pre> B1: Kiểm tra thông tin IF NOT EXISTS (SELECT * FROM CHUNHA </pre>	<pre> S(CHUNHA) // Trả khóa </pre>

		WHERE MaChuNha = @MaChuNha)	
		B2: Xóa 1 chủ nhà DELETE FROM CHUNHA WHERE MaChuNha = @MaChuNha	X(CHUNHA) //Xin khóa ghi trên bảng CHUNHA
		COMMIT	// Trả khóa
B3: Đọc lại danh sách chủ nhà SELECT * FROM CHUNHA WHERE DIACHI LIKE N%' + @Quan + '%'	S(CHUNHA) // Trả khóa		
COMMIT			
// T1 in ra tổng khác với tổng trong danh sách chủ nhà được lấy ra (nhiều hơn 1 so với tổng chủ nhà trong danh sách)			

III. Sinh viên thực hiện: Phạm Minh Nhựt

5. Tình huống 5: AD3- AD6: Cập nhật trạng thái thuê/bán nhà - Nhân viên quản lý các căn nhà

ERR01: Dirty read T1 (User =Nhân viên): Nhân viên thực hiện cập nhật trạng thái thuê/bán nhà T2 (User =Nhân viên): Nhân viên quản lý danh sách các căn nhà			
UPDATE_TRANG_THAI_NHA	Khóa	UPDATE_TRANG_THAI_NHA	Khóa
Input: MaNha Output: Thông báo		Input: MaNha Output: ThôngBao	
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED		SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED	
BEGIN TRAN			
B1: Kiểm tra thông tin (1) Nha IF EXISTS (SELECT * FROM Nha WHERE NHA.MaNha=@ma)	.		
B2: Cập nhật trạng thái (2) Nha @ma = 1 UPDATE NHA SET TinhTrang = 'Đã bán' WHERE MaNha = @ma IF @ERROR != 0 ROLLBACK RETURN "Có lỗi xảy ra khi cập trạng thái nhà!" END	R(NHA)		
WAITFOR DELAY '00:00:20'		BEGIN TRAN	
		B1: Xem thông tin Nha IF EXISTS (SELECT * FROM NHA WHERE NHA.MaNha=@ma) BEGIN IF @ERROR == 0 RETURN "Thông tin nhà" + NHA	

		ELSE RETURN "Sai thông tin nhà" END	
WAITFOR DELAY '00:00:20'			
		COMMIT	

6. Tình huống 6: AD1 - AD8: Tạo hợp đồng - Quản lý danh sách hợp đồng

ERR01: Bómg ma T1 (User = Nhân viên): Nhân viên tạo hợp đồng T2 (User = Nhân viên): Nhân viên quản lý danh sách hợp đồng			
TAO HOP DONG	Khóa	TAO HOP DONG	Khóa
Input: Thông tin hợp đồng Output: Thông báo tạo thành công hoặc thất bại	R(HOPDONG)	Input: Output: Thông báo hiển thị thông tin tất cả hợp đồng	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL UNREPEATEABLE READ	
BEGIN TRAN			
B1: Thêm thông tin vào bảng HOPDONG INSERT INTO NHA (LoaiHD, NgaBatDau, NgayKetThuc, MaNha, MaKhachHang, MaNhanVien) VALUES (@LoaiHD, @NgayBatDau, @NgayKetThuc, @MaNha, @MaKH, @MaNV)			
B2: Kiểm tra dữ liệu IF @ERROR !=0 ROLLBACK RETURN "KHÔNG THỂ TẠO HỢP ĐỒNG"			

END IF @ERROR == 0 COMMIT END			
WAITFOR DELAY '00:00:20'			
		BEGIN TRAN	
	S(HOPDONG)	B1: Xem danh sách hợp đồng SELECT * FROM HOPDONG	
		COMMIT	

IV. Sinh viên thực hiện: Phạm Duy Tùng

7. Tình huống 7: ALL1- ALL4

ERR01: Unrepeatable Read T1 (User = Khách hàng): Thực hiện đăng nhập vào hệ thống với mật khẩu cũ. T2 (User = Khách hàng): Thực hiện đổi mật khẩu			
sp_DangNhap	Khóa	sp_DoiMatKhau	Khóa

Input: TenDangNhap, MatKhai		Input: TenDangNhap, MatKhaiMoi	
Output: ThongBao		Output: ThongBao	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Kiểm tra tên đăng nhập IF NOT EXISTS (SELECT * FROM NGUOIDUNG WHERE NGUOIDUNG.TenDangNhap=@TenDangNhap) BEGIN RETURN @TenDangNhap + “Không Tồn Tại” END	S(NGUOIDUNG) //Trả ngay		
B2: Kiểm tra tên đăng nhập và mật khẩu IF NOT EXISTS (SELECT * FROM NGUOIDUNG WHERE NGUOIDUNG.TenDangNhap=@TenDangNhap AND TAIKHOAN.MatKhai=@MatKhaiCu) BEGIN RETURN “Sai Mật Khẩu” END	S(NGUOIDUNG) //Trả ngay		
WAITFOR DELAY '00:00:20'			
		BEGIN TRAN	
		UPDATE NGUOIDUNG SET MatKhai=@MatKhaiMoi WHERE TenDangNhap=@TenDangNhap IF @error != 0 BEGIN ROLLBACK	X(NGUOIDUNG)

		RETURN "Có lỗi xảy ra khi cập nhật mật khẩu!" END	
		COMMIT	//Trả khóa
B3: Lấy thông tin người dùng SELECT * FROM NGUOIDUNG WHERE TenDangNhap=@TenDangNhap AND MatKhau=@MatKhauCu	S(NGUOIDUNG) //Trả ngay		
COMMIT			

8. Tình huống 8: AD2 - AD2

ERR02: Phantom Deadlock T1 (User = Nhân Viên): Đăng ký lịch xem nhà cho khách hàng. T1 (User = Nhân Viên): Đăng ký lịch xem nhà cho khách hàng.			
sp_DangKyXemNha	Khóa	sp_DangKyXemNha	Khóa
<u>Input:</u> MaNha, MaKhachHang, NgayXem, MaNhanVien <u>Output:</u> ThongBao		<u>Input:</u> MaNha, MaKhachHang, NgayXem, MaNhanVien <u>Output:</u> ThongBao	
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE		SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
BEGIN TRAN			

<p>B1: Kiểm tra thông tin lịch xem nhà IF EXISTS (SELECT * FROM XEMNHA WHERE MaNha=@MaNha AND MaKhachHang=@MaKhachHang AND NgayXem=@NgayXem AND MaNhanVien=@MaNhanVien) BEGIN RETURN "Đã tồn tại một lịch xem nhà!" END</p>	<p>S(XEMNHA) //Xin khoá đọc trên bảng XEMNHA. //Trả ngay</p>		
<p>WAITFOR DELAY '00:00:20'</p>			
		<p>BEGIN TRAN</p>	
		<p>B1: Kiểm tra thông tin lịch xem nhà IF EXISTS (SELECT * FROM XEMNHA WHERE MaNha=@MaNha AND MaKhachHang=@MaKhachHang AND NgayXem=@NgayXem) BEGIN RETURN "Đã tồn tại một lịch xem nhà!" END</p>	<p>S(XEMNHA) //Trả ngay</p>
		<p>B2: Đăng ký lịch xem nhà INSERT INTO XEMNHA (MaNha, MaKhachHang, NgayXem, GioXem) VALUES (@MaNha, @MaKhachHang, @NgayXem, @MaNhanVien) IF @error != 0 BEGIN ROLLBACK RETURN "Có lỗi xảy ra khi thêm lịch xem nhà!" END</p>	<p>X(XEMNHA) //Xin Khoá ghi trên bảng XEMNHA</p>

B2: Đăng ký lịch xem nhà INSERT INTO XEMNHA (MaNha, MaKhachHang, NgayXem, GioXem) VALUES (@MaNha, @MaKhachHang, @NgayXem, @MaNhanVien) IF @error != 0 BEGIN ROLLBACK RETURN "Có lỗi xảy ra khi thêm lịch xem nhà!" END COMMIT	X(XEMNHA) //Xin Khoá ghi trên bảng XEMNHA		
		COMMIT	

V. Sinh viên thực hiện: Võ Minh Trung

9. Tình huống 9: HO1 - AD6

ERR01: Dirty read T1 (User = chủ nhà): thực hiện cập nhật lại điều kiện cho thuê nhà. T2 (User = nhân viên): thực hiện xem danh sách nhà.			
sp_CapNhatDieuKienCN <u>Input:</u> @MaNha, @DieuKienCN <u>Output:</u> Thông báo SET TRANSACTION ISOLATION LEVEL READ COMMITTED BEGIN TRAN	Khóa	sp_XemDanhSachNha <u>Input:</u> @MaNha <u>Output:</u> Danh sách nhà SET TRANSACTION ISOLATION LEVEL READ COMMITTED	Khóa
B1: Kiểm tra thông tin (1) MaNha của NHA có tồn tại chưa: IF NOT EXISTS (SELECT * FROM NHA WHERE NHA.MaNha = @MaNha) BEGIN	S(NHA)		

<pre> PRINT N'LỖI MaNha KHÔNG TỒN TẠI' SET @KQ = N'LỖI' + @MaNha ROLLBACK TRAN RETURN 1; -- Lỗi do dữ liệu CSDL END B2: Kiểm tra thông tin (2) @MaNha nhập vào null IF @MaNha IS NULL BEGIN PRINT N'LỖI MaNha KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 2; -- Lỗi do dữ liệu input END </pre>	<pre> //Xin khoá đọc trên bảng NHA //Trả khóa </pre>		
<pre> B2: Cập nhật lại điều kiện của chủ nhà (3): BEGIN TRY UPDATE NHA SET DieuKienCN = @DieuKienCN WHERE MaNha = @MaNha END TRY </pre>	<p>S(NHA)</p> <pre> /Xin khoá ghi trên bảng NHA //Trả khóa </pre>		
<pre> WAITFOR DELAY '00:00:20' </pre>			
		<pre> BEGIN TRAN </pre>	
		<pre> B1: Kiểm tra thông tin (4) MaNha của NHA có tồn tại chưa: IF NOT EXISTS (SELECT * FROM NHA WHERE NHA.MaNha = @MaNha) BEGIN PRINT N'LỖI MaNha KHÔNG TỒN TẠI' SET @KQ = N'LỖI' + @MaNha ROLLBACK TRAN RETURN 1; -- Lỗi do dữ liệu CSDL END </pre>	<p>X(NHA)</p> <p>// Đọc không cần xin khoá</p>

		B2: Xem danh sách nhà (5): SELECT * FROM NHA WHERE MaNha = @MaNha	X(NHA) // Đọc không cần xin khoá
		COMMIT	
IF @error //Khi cập nhật đến bảng NHA: dữ liệu bị lỗi, không thể cập nhật vào bảng Nhà được hoặc dữ liệu trùng với ban đầu. Điều này dẫn đến toàn dữ liệu sẽ bị ROLLBACK ROLLBACK	S(NHA) //Xin khoá đọc trên bảng NHA //Trả khóa		

10. Tình huống 10: AD4 - AD7

ERR02: Phantom T1 (User = nhân viên): thực hiện xem danh sách chủ nhà T2 (User = nhân viên): thực hiện thêm chủ nhà			
sp XemChuNha	Khóa	sp ThemChuNha	Khóa
Input: @MaChuNha, @TenChuNha Output: Danh sách chủ nhà		Input: @TenChuNha, @DiaChi, @SDT Output: Thêm thành công	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Kiểm tra thông tin (1) MaChuNha của CHUNHA có tồn tại chưa: IF NOT EXISTS (SELECT * FROM CHUNHA WHERE CHUNHA.MaChuNha = @MaChuNha) BEGIN	S(CHUNHA) //Xin khoá đọc trên bảng CHUNHA		

<pre> PRINT N'LỖI MaChuNha KHÔNG TỒN TAI' SET @KQ = N'LỖI' + @MaChuNha ROLLBACK TRAN RETURN 1; -- Lỗi do dữ liệu CSDL END B2: Kiểm tra thông tin (2) @MaChuNha nhập vào null IF @MaChuNha IS NULL BEGIN PRINT N'LỖI MaChuNha KHÔNG TỒN TAI' ROLLBACK TRAN RETURN 2; -- Lỗi do dữ liệu input END </pre>	//Trả khóa		
<pre> B3: Đếm số lượng họ Nguyễn (3) DECLARE @Count INT = 0; SELECT @Count = COUNT(*) FROM CHUNHA WHERE @TenChuNha LIKE N'Nguyễn%'; PRINT 'Số lượng chủ nhà có họ Nguyễn: ' + CAST(@Count AS NVARCHAR(10)); </pre>	<p>S(CHUNHA) //Xin khoá đọc trên bảng CHUNHA</p> <p>//Trả khóa</p>		
WAITFOR DELAY '00:00:20'			
		BEGIN TRAN	
		<pre> B1: Kiểm tra thông tin (1) Thêm chủ nhà INSERT CHUNHA(TenChuNha, DiaChi, SDT) VALUE (@TenChuNha, @DiaChi, @SDT) </pre>	X(NHA)

			/Xin khoá ghi trên bảng CHUNHA
		COMMIT	//Trả khóa
<p>B3: Kiểm tra thông tin (2) Hiển thị danh sách và số lượng</p> <pre> SELECT * FROM CHUNHA WHERE MaChuNha = @MaChuNha AND WHERE @TenChuNha LIKE N'Nguyễn%'; PRINT 'Số lượng chủ nhà có họ Nguyễn: ' + CAST(@Count AS NVARCHAR(10)); </pre>	<p>S(CHUNHA) //Xin khoá đọc trên bảng CHUNHA //Trả khóa</p>		
COMMIT			