

BÁO CÁO ĐỒ ÁN THỰC HÀNH  
MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU  
GVHD: Cô Hồ Thị Hoàng Vy

## THÔNG TIN NHÓM

STT	MSSV	Họ tên	Công việc	% Hoàn thành
1	20127561	Nguyễn Hoài Mẫn	- Cài đặt tình huống tranh chấp: LOST UPDATE, UNREPEATEABLE READ	100%
2	20127423	Đinh Thành Danh	- Cài đặt tình huống tranh chấp: DIRTY READ, PHANTOM READ - Giao tác chức năng	100%
3	20127503	Dương Hiền Lê Hoàng	- Cài đặt tình huống tranh chấp: DIRTY READ, UNREPEATEABLE READ	100%
4	20127204	Nguyễn Phụng Khanh	- Cài đặt tình huống tranh chấp: LOST UPDATE, UNREPEATEABLE READ	100%

## CÀI ĐẶT TÌNH HUỐNG TRANH CHẤP

## I. Sinh viên thực hiện: Dương Hiền Lê Hoàng

1. Tình huống 1: Nhân viên A update thông tin khách hàng B thì trong lúc đó nhân viên B truy vấn thông tin khách hàng B để xem thông tin khách hàng B

<b>ERR01: Dirtyread</b> T1 (User = Nhân viên): thực hiện cập nhật 1 thông tin khách hàng B. T2 (User = Nhân viên): thực hiện xem thông tin khách hàng B			
UpdatedThôngTinKH	Khóa	LayThôngTinKH	Khóa
<b>Input:</b> @MaKH @SDTKH <b>Output:</b> .....		<b>Input:</b> @MaKH <b>Output:</b>	
<b>BEGIN TRAN</b>			
<b>B1: Kiểm tra thông tin (1) khách hàng</b> IF EXISTS (SELECT * FROM KHACHHANG WHERE MaKH = @MaKH) BEGIN PRINT N'Không tồn tại khách hàng đó' ROLLBACK TRAN RETURN 0 END	S(KHACHHANG) //Xin khoá đọc trên bảng KHACHHANG với đk MaKH = @MaKH.		
<b>B2: Cập nhật thông tin vào bảng KHACHHANG</b> UPDATE KHACHHANG SET SDTKH = @SDTKH Where MaKH = @MaKH;	U,X(KHACHHANG)		

	//Xin khóa update trên bảng KHACHHANG với dk MaKH = @MaKH		
WAITFOR DELAY '0:0:5';			
		BEGIN TRAN	
		SET TRAN ISOLATION LEVEL READ UNCOMMITTED	Cho phép đọc dữ liệu đang được thực hiện bởi các giao dịch khác mà chưa được commit.
		B1: Kiểm tra thông tin khách hàng có tồn tại không IF EXISTS (SELECT * FROM KHACHHANG WHERE MaKH = @MaKH) BEGIN PRINT N'Không tồn tại khách hàng đó' ROLLBACK TRAN RETURN 0 END	S(KHACHHANG) //Xin khoá đọc trên bảng KHACHHANG với dk MaKH = @MaKH.

		B2: Xem thông tin khách hàng SELECT * FROM KHACHHANG Where MaKH = @MaKH;	S(KHACHHANG) //Xin khoá đọc trên bảng KHACHHANG với dk MaKH = @MaKH.
		COMMIT	
IF(@SDTKH > 999999999 OR  @SDTKH < 100000000) BEGIN ROLLBACK TRAN END			

2. Tình huống 2: Quản trị A cập nhật lại thông tin hồ sơ của khách hàng B thì trong lúc đó nhân viên C truy vấn thông tin khách hàng B

<b>ERR01: UNREPEATABLE READ</b> T1 (User = Quản trị): thực hiện cập nhật 1 thông tin hồ sơ của khách hàng B. T2 (User = Nhân viên): thực hiện xem thông tin hồ sơ của khách hàng B			
UpdatedThongTinHoSo	Khóa	LayThongTinHoSo	Khóa
<u>Input:</u> @MaHS @NgayKham @MaKH <u>Output:</u> .....		<u>Input:</u> @MaHS <u>Output:</u>	
		BEGIN TRAN	

		<p><b>B1: Kiểm tra thông tin khách hàng có tồn tại không</b></p> <p>IF EXISTS (SELECT * FROM HoSoBN WHERE MaHS = @MaHS)</p> <p>BEGIN</p> <p>PRINT N'Không tồn tại hồ sơ đó'</p> <p>ROLLBACK TRAN</p> <p>RETURN 0</p> <p>END</p>	<p><b>S(HoSoBN )</b></p> <p>//Xin khoá đọc trên bảng HoSoBN với đk MaHS = @MaHS.</p>
		WAITFOR DELAY '0:0:5';	
<b>BEGIN TRAN</b>			
<p><b>B1: Kiểm tra thông tin hồ sơ khách hàng</b></p> <p>IF EXISTS (SELECT * FROM HoSoBN WHERE MaHS = @MaHS)</p> <p>BEGIN</p> <p>PRINT N'Không tồn tại hồ sơ đó'</p> <p>ROLLBACK TRAN</p> <p>RETURN 0</p> <p>END</p>	<p><b>S(HoSoBN )</b></p> <p>//Xin khoá đọc trên bảng HoSoBN với đk MaHS = @MaHS.</p>		
<p><b>B2: Cập nhật thông tin vào bảng KHACHHANG</b></p> <p>UPDATE HoSoBN SET NgayKham = @NgayKham, MaKH = @MaKH Where MaHS = @MaHS;</p>	<p><b>U,X(HoSoBN )</b></p> <p>//Xin khoá trên bảng HoSoBN với đk MaHS = @MaHS.</p>		
WAITFOR DELAY '0:0:5';			

		B2: Xem thông tin hồ sơ khách hàng  SELECT * FROM HoSoKH Where MaKH = @MaKH;	S(HoSoBN ) //Xin khoá đọc trên bảng HoSoBN với đk MaKH = @MaKH.
COMMIT		COMMIT	

## II. Sinh viên thực hiện: Đinh Thành Danh

1. Tình huống 1: Khách hàng A đăng nhập vào hệ thống chọn việc đặt lịch hẹn, trong khi đó nha sĩ B cũng đăng nhập vào hệ thống hiển thị lịch hẹn cũ

<b>ERR01: PHANTOM READ</b>  T1 (User = Khách hàng ):Khách hàng 1 thực hiện việc đặt lịch hẹn  T2 (User = Nha sĩ): Thực hiện xem danh sách cuộc hẹn			
Sp_ThemLichHen_1	Khóa	Sp_XemLichHen_2	Khóa

<p><b><u>Input:</u></b></p> <p><u>@NgayHen</u></p> <p><u>@ThoiGianHen</u></p> <p><u>@HoTenKH</u></p> <p><u>@MaKH</u></p> <p><u>@MaNS</u></p> <p><b><u>Output:</u></b> 1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</p>		<p><b><u>Input:</u></b></p>	
		<p>BEGIN TRAN</p>	



		<p>B1: Xem danh sách lịch hẹn:</p> <p><code>SELECT Ngay,ThoiGian,MaNS FROM LichCaNhanNS</code></p> <p><code>return 1</code></p>	<p><code>S(LichCaNhanNS)</code></p> <p>//Xin khoá đọc trên bảng LichCaNhanNS</p>
		<code>WAITFOR DELAY '0:0:10'</code>	
<code>BEGIN TRAN</code>			

<p>B1: Kiểm tra ngày và thời gian lịch hẹn xem có chưa ?</p> <pre>IF(EXISTS(SELECT * FROM LichHen WHERE Ngày = '01-10-2023' AND ThoiGian = '08:30:00')) RETURN -1</pre>	<p><b>S(LichHen)</b></p> <p>//Xin khoá đọc trên bảng LichHen với điều kiện Ngày = '01-10-2023' và ThoiGian ='08:30:00'</p>		
---	--	--	--

<p><b>B2: Kiểm tra mã nha sĩ xem có chưa ?</b></p> <p><b>IF</b>(<b>EXISTS</b>(<b>SELECT</b> * <b>FROM</b> LichHen <b>WHERE</b> MaNS = '1' )</p> <p><b>RETURN</b> -1</p>	<p><b>S(LichHen)</b></p> <p>//Xin khoá đọc trên bảng</p> <p>LichHen với điều kiện MaNS = '1'</p>		
<p><b>B3 : Thêm lịch hẹn mới</b></p> <p><b>INSERT INTO</b> LichHen(NgayHen, ThoiGianHen, TenKH, MaKH, MaNS) <b>VALUES</b> ('01-10-2023', '08:30:00', N'Nguyen Van E', '3', '2')</p> <p><b>return</b> 1</p>	<p><b>X(LichHen)</b> //xin khoá ghi trên bảng LichHen để thêm lịch hẹn mới</p>		

COMMIT			
		<p>B1: Xem danh sách lịch hẹn:</p> <pre>SELECT Ngay,ThoiGian,MaNS FROM LichCaNhanNS</pre> <p>return 1</p>	<p>S(LichCaNhanNS)</p> <p>//Xin khoá đọc trên bảng LichCaNhanNS</p>
		COMMIT	

2. Tình huống 2: Trong hệ thống, khách hàng đang cập nhật thông tin cá nhân thì nhân viên xem thông tin của khách hàng đó.

<p><b>ERR02: DIRTY READ</b></p> <p>T1 (User = Khách hàng): Khách hàng đang thực hiện cập nhật thông tin</p> <p>T2 (User = Nhân viên): Nhân viên đang xem thông tin cũ của khách hàng</p>			
Sp_XemThongTinCaNhan	Khóa	Sp_XemThongTinKhachHang	Khóa

<p><b><u>Input:</u></b></p> <p><u>@HoTenKH</u></p> <p><u>@NgaySinhKH</u></p> <p><u>@DiaChiKH</u></p> <p><u>@SDTKH</u></p> <p><u>@EmailKH</u></p> <p><b><u>Output:</u></b> 1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</p>		<p><b><u>Input:</u></b></p>	
<p><b>BEGIN TRAN</b></p>			

<p><b>B1: Kiểm tra thông tin (1) khách hàng</b>  <b>IF EXISTS (SELECT * FROM KHACHHANG</b>  <b>WHERE SDTKH = @SDTKH)</b>  <b>BEGIN</b>          PRINT N'Không tồn tại tài khoản'          ROLLBACK TRAN          RETURN 0   <b>END</b></p>	<p><b>S(KHACHHANG)</b>          //Xin khoá đọc trên          bảng KHACHHANG          với dk SDTKH =          @SDTKH</p>		
<p><b>B2: Cập nhật thông tin vào bảng</b>  <b>KHACHHANG</b>  <b>UPDATE KHACHHANG SET EmailKH =</b>  <b>@EmailKH Where SDTKH = @SDTKH;</b></p>	<p><b>U,X(KHACHHANG)</b>          //Xin khóa update          trên bảng          KHACHHANG với          dk SDTKH =          @SDTKH</p>		
<p><b>WAITFOR DELAY '0:0:5';</b></p>			

		BEGIN TRAN	
		SET TRAN ISOLATION LEVEL READ UNCOMMITTED	Cho phép đọc dữ liệu đang được thực hiện bởi các giao dịch khác mà chưa được commit

		<p><b>B1: Kiểm tra thông tin khách hàng có tồn tại không</b></p> <pre> IF EXISTS (SELECT * FROM KHACHHANG WHERE MaKH = @MaKH) BEGIN     PRINT N'Không tồn tại khách hàng đó'     ROLLBACK TRAN     RETURN 0 END </pre>	<p><b>S(KHACHHANG)</b></p> <p>//Xin khoá đọc trên bảng KHACHHANG với đk MaKH = @MaKH.</p>
		<p><b>B2: Xem thông tin khách hàng</b></p> <pre> SELECT * FROM KHACHHANG Where MaKH = @MaKH; </pre>	<p><b>S(KHACHHANG)</b></p> <p>//Xin khoá đọc trên bảng KHACHHANG với đk MaKH = @MaKH.</p>
		<pre> COMMIT </pre>	



### III. Sinh viên thực hiện: Nguyễn Hoài Mẫn

1. Tình huống 1: Quản trị viên A đăng nhập vào hệ thống để cập nhật thông tin của một dịch vụ, chẳng hạn như điều chỉnh giá hoặc thay đổi mô tả dịch vụ. Trong khi đó, Quản trị viên B cũng đăng nhập và chọn cùng một dịch vụ để cập nhật thông tin.

ERR01: LOST UPDATE			
T1 (User = Admin): Admin 1 tiến hành cập nhật thông tin dịch vụ là : 'DV001', 'Kham rang tong quat', 'Dich vu nhanh', 150			
T2 (User = Admin): Admin 1 tiến hành cập nhật thông tin dịch vụ là : 'DV001', 'Kham rang tong quat', 'Dich vu nhanh', 200			
Sp_UpdateDichVu_1	Khóa	Sp_UpdateDichVu_2	Khóa
<u><b>Input:</b></u> <u>@MaDV = 'DV001'</u> <u>@TenDV = 'Kham rang tong quat'</u> <u>@Loai = 'Dich vu nhanh'</u> <u>@Tien = '150'</u> <u><b>Output:</b> 1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u>		<u><b>Input:</b></u> <u>@MaDV = 'DV001'</u> <u>@TenDV = 'Kham rang tong quat'</u> <u>@Loai = 'Dich vu nhanh'</u> <u>@Tien = '150'</u> <u><b>Output:</b> 1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u>	

BEGIN TRAN			
<p>B1: Kiểm tra thông tin mã dịch vụ(MADV) có tồn tại hay không ?</p> <pre> IF NOT EXISTS (SELECT * FROM DICHVU WHERE MADV = 'DV001' ) BEGIN PRINT 'Dịch vụ ' + @MaDV + N' Không Tồn Tại' ROLLBACK TRAN RETURN 0 END </pre>	<p><b>S(DICHVU)</b></p> <p>//Xin khoá đọc trên bảng DICHVU với điều kiện MADV = 'DV001'</p>		
WAITFOR DELAY '0:0:5'			
		BEGIN TRAN	

		<p>B1: Kiểm tra thông tin mã dịch vụ(MADV) có tồn tại hay không ?</p> <pre> IF NOT EXISTS (SELECT * FROM DICHVU WHERE MADV = 'DV001' ) BEGIN PRINT 'Dịch vụ ' + @MaDV + N' Không Tồn Tại' ROLLBACK TRAN RETURN 0 END </pre>	<p><b>S(DICHVU)</b></p> <p>//Xin khoá đọc trên bảng DICHVU với điều kiện MADV = 'DV001'</p>
--	--	---	---

		<p>B2: Cập nhật thông tin thuộc tính</p> <p>TenDV, Loai, Tien trong bảng DICHVU</p> <p>thành 'Kham rang tong quat', 'Dich vu nhanh', 200</p> <p>UPDATE DichVu SET</p> <p>TenDV = @TenDV,</p> <p>Loai = @Loai,</p> <p>Tien = @Tien</p> <p>WHERE MaDV = @MaDV</p>	<p><b>U,X(DICHVU)</b></p> <p>//Xin khoá update trên bảng DICHVU với điều kiện</p> <p>MADV = 'DV001'</p>
		COMMIT	

<p><b>B2: Cập nhật thông tin thuộc tính</b></p> <p><b>TenDV, Loai, Tien trong bảng DICHVU</b></p> <p><b>thành 'Kham rang tong quat', 'Dich vu nhanh', 150</b></p> <p>UPDATE DichVu SET</p> <p>TenDV = @TenDV,</p> <p>Loai = @Loai,</p> <p>Tien = @Tien</p> <p>WHERE MaDV = @MaDV</p>	<p><b>U,X(DICHVU)</b></p> <p>//Xin khoá update</p> <p>trên bảng DICHVU</p> <p>với điều kiện</p> <p>MADV = 'DV001'</p>		
COMMIT			

2. Tình huống 2: Trong hệ thống, có một nha sĩ vừa đăng nhập xong thì admin tiến hành đổi mật khẩu của tài khoản bác sĩ đó. Dẫn đến việc sau khi đăng nhập, bác sĩ đó không thể xử lý lấy được thông tin tài khoản của mình.

<b>ERR02: UNREPEATABLE READ</b>  T1 (User = Nha sĩ): Nha sĩ tiến hành đăng nhập vào hệ thống với tên đăng nhập là : “111111112”,  mật khẩu là : “12345”  T2 (User = admin): admin tiến hành đổi mật khẩu của tài khoản nha sĩ có mã số 1 từ “12345” thành “54321”			
Sp_NV_LayTongTinTK	Khóa	Sp_NV_DoiMK	Khóa
<b><u>Input:</u></b>  <u>@SDT= '111111112'</u>  <u>@MATKHAU = '12345'</u>  <u>@LoaiTK = '1'</u>  <b><u>Output: 1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u></b>		<b><u>Input:</u></b>  <u>@SDT = '111111112'</u>  <u>@MATKHAU = '54321'</u>  <b><u>Output: 1 biến int có giá trị 1 hoặc 0. 1 – Thành công ngược lại 0 – Thất bại</u></b>	
BEGIN TRAN			

<p>B1: Xử lý lấy thông tin mã tài khoản nha sĩ</p> <pre> SELECT @Ma = NS.MaNS FROM TaiKhoan A INNER JOIN NhaSi NS ON A.SDT = NS.SDTNS WHERE A.SDT = '111111112' AND A.MatKau = '12345' AND A.LoaiTK = '1' </pre>	<p><b>S(TAIKHOAN)</b></p> <p>//Xin khoá đọc trên bảng TaiKhoan với điều kiện TENDANGNHAP = '111111112', MATKHAU = '12345', LoaiTK = '1'</p>		
<pre> WAITFOR DELAY '0:0:5' </pre>			
		BEGIN TRAN	

		<p><b>B1: Kiểm tra xem tài khoản có tồn tại hay không?</b></p> <pre> IF NOT EXISTS (SELECT * FROM TAIKHOAN WHERE SDT = '111111112') BEGIN PRINT 'Tài khoản Không Tồn Tại' ROLLBACK TRAN RETURN 0 END </pre>	<p><b>S(TAIKHOAN)</b></p> <p>//Xin khoá đọc trên bảng TAIKHOAN với điều kiện SDT = '111111112'</p>
		<p><b>B2: Xử lý cập nhật mật khẩu</b></p> <pre> UPDATE TAIKHOAN SET MATKHAU = '54321' WHERE SDT = '111111112' </pre>	<p><b>U,X(TAIKHOAN)</b></p> <p>//Xin khoá update trên bảng TAIKHOAN với điều kiện</p>



			SDT = '111111112'
<p>B2: Kiểm tra xem mã tài khoản có tồn tại hay không?</p> <pre> IF @Ma IS NULL BEGIN PRINT N'Nhà số Không Tồn Tại' ROLLBACK TRAN RETURN 0 END </pre>			
<p>B3: Xử lý lấy thông tin của tài khoản?</p> <pre> SELECT * FROM NhaSi NS WHERE NS.MaNS = @Ma; </pre>	<p><b>S(NHASI)</b></p> <p>//Xin khoá đọc trên bảng NHASI, với điều kiện NS.MaNS = @Ma</p>		
COMMIT		COMMIT	

#### IV. Sinh viên thực hiện: Nguyễn Phương Khanh

- Tình huống 1: Quản trị viên A cập nhật số lượng tồn kho của loại thuốc A từ 30 vỉ còn 20 vỉ, trong lúc đó quản trị viên B cập nhật số lượng tồn kho của loại thuốc A từ 30 vỉ còn 15 vỉ

<b>ERROR1: LOST UPDATE</b>			
T1 (User = Quản trị viên A): thực hiện cập nhật số lượng tồn kho của loại thuốc A từ 30 vỉ còn 20 vỉ			
T2 (User = Quản trị viên B): thực hiện cập nhật số lượng tồn kho của loại thuốc A từ 30 vỉ còn 15 vỉ			
<b>UpdatedSoLuongTonKhoThuoc_1</b>	<b>Khóa</b>	<b>UpdatedSoLuongTonKhoThuoc_2</b>	<b>Khóa</b>
<b>Input:</b> @SLTonKho		<b>Input:</b> @SLTonKho	
<b>Output:</b> .....		<b>Output:</b>	
<b>BEGIN TRAN</b>			
<b>B1: Kiểm tra thông tin thuốc có tồn tại hay không</b> IF EXISTS (SELECT * FROM THUOC WHERE MaThuoc = @MaThuoc) BEGIN PRINT N 'Không tồn tại loại thuốc đó' ROLLBACK TRAN RETURN 0 END	<b>S(THUOC)</b> //Xin khoá đọc trên bảng THUOC với đk MaThuoc = @MaThuoc.		
<b>B2: Cập nhật thông tin SLTonKho vào bảng THUOC</b> UPDATE THUOC SET @SLTonKho = 20 where MaThuoc = @MaThuoc;	<b>U,X(THUOC)</b> //Xin khóa update trên		

	bảng THUOC với dk MaThuoc = @MaThuoc		
WAITFOR DELAY '0:0:05';			
		BEGIN TRAN	
		B1: Kiểm tra thông tin thuốc có tồn tại hay không IF EXISTS (SELECT * FROM THUOC WHERE MaThuoc = @MaThuoc) BEGIN PRINT N 'Không tồn tại loại thuốc đó' ROLLBACK TRAN RETURN 0 END	<b>S(THUOC)</b> //Xin khoá đọc trên bảng THUOC với dk MaThuoc = @MaThuoc.
		B2: Cập nhật thông tin SLTonKho vào bảng THUOC UPDATE THUOC SET @SLTonKho = 15 where MaThuoc = @MaThuoc;	<b>U,X(THUOC)</b> //Xin khóa update trên bảng THUOC với dk MaThuoc = @MaThuoc
		COMMIT	
COMMIT			

2. Tình huống 2: Nhân viên xem số điện thoại của nha sĩ A để liên lạc, cùng lúc đó nha sĩ A cập nhật số điện thoại cá nhân của mình từ 123456789 thành 987654321 và lưu thay đổi thành công. Sau đó nhân viên load lại thông tin, đọc được số điện thoại mới.

<b>ERR01: UNREPEATEABLE READ</b> T1 (User = Nhân viên): thực hiện xem số điện thoại của nha sĩ A để liên lạc. T2 (User = Nha sĩ): thực hiện cập nhật số điện thoại cá nhân từ 123456789 thành 987654321 và lưu thay đổi.			
LaySoDienThoaiNS	Khóa	UpdatedSoDienThoaiNS	Khóa
<b>Input:</b> @SDTNS <b>Output:</b> @SDTNS='123456789'		<b>Input:</b> @SDTNS='987654321' <b>Output:</b> @SDTNS='123456789'	
<b>BEGIN TRAN</b>			
<b>B1: Kiểm tra thông tin nha sĩ có tồn tại hay không</b> IF EXISTS (SELECT * FROM NhaSi WHERE MaNS = @MaNS) BEGIN PRINT N 'Không tồn tại nha sĩ đó' ROLLBACK TRAN RETURN 0 END	<b>S(NhaSi)</b> //Xin khoá đọc trên bảng NhaSi với dk MaNS = @MaNS.		
<b>B2: Xem thông tin số điện thoại nha sĩ</b> SELECT SDTNS FROM NhaSi WHERE MaNS = @MaNS;	<b>S(NhaSi)</b> //Xin khoá đọc trên bảng NhaSi với dk MaNS = @MaNS.		

WAITFOR DELAY '0:0:05';			
		BEGIN TRAN	
		B1: Kiểm tra thông tin nha sĩ có tồn tại hay không IF EXISTS (SELECT * FROM NhaSi WHERE MaNS = @MaNS) BEGIN PRINT N 'Không tồn tại nha sĩ đó' ROLLBACK TRAN RETURN 0 END	<b>S(NhaSi)</b> //Xin khoá đọc trên bảng NhaSi với dk MaNS = @MaNS.
		B2: Cập nhật số điện thoại nha sĩ UPDATE NhaSi SET @SDTNS = 123456789 where MaNS = @MaNS;	<b>U,X(NhaSi)</b> //Xin khóa update trên bảng Nha Si với dk MaNS = @MaNS
B3: Xem thông tin số điện thoại nha sĩ _ lần 2 SELECT SDTNS FROM NhaSi WHERE MaNS = @MaNS;	<b>S(NhaSi)</b> //Xin khoá đọc trên bảng NhaSi với dk MaNS = @MaNS.		
COMMIT			
		COMMIT	

## V. Giao tác một số chức năng :

--Giao tác thêm khách hàng và hồ sơ bệnh nhân mới

BEGIN TRANSACTION;

-- Thêm khách hàng mới

INSERT INTO KHACHHANG (HoTenKH, NgaySinhKH, DiaChiKH, SDTKH, EmailKH, MaNV)

VALUES ('Nguyen Van X', '1990-01-01', 'TP.HCM', 123456789, 'nguyenvanx@example.com', 1);

-- Lấy mã khách hàng vừa thêm

DECLARE @MaKH INT;

SET @MaKH = SCOPE\_IDENTITY();

-- Thêm hồ sơ bệnh nhân mới

INSERT INTO HoSoBN (NgayKham, NguoiKham, MaKH, MaNS)

VALUES ('2024-01-03', 'NV001', @MaKH, 2);

COMMIT;

--Giao tác cập nhật thông tin khách hàng:

BEGIN TRANSACTION;

-- Cập nhật thông tin khách hàng

UPDATE KHACHHANG

SET DiaChiKH = '135B Trần Hưng Đạo, 1, TPHCM'

WHERE MaKH = 1;

COMMIT;

--Giao tác xoá hồ sơ bệnh nhân và thông tin liên quan:

BEGIN TRANSACTION;

-- Lấy mã khách hàng và mã nhân sự từ hồ sơ bệnh nhân

DECLARE @MaKH INT, @MaNS INT;

SELECT @MaKH = MaKH, @MaNS = MaNS FROM HoSoBN WHERE MaHS = 1;

-- Xoá hồ sơ bệnh nhân

DELETE FROM HoSoBN WHERE MaHS = 1;

-- Xoá các liên quan trong các bảng khác

DELETE FROM LichHen WHERE MaKH = @MaKH OR MaNS = @MaNS;

-- Nếu cần, xoá khách hàng

DELETE FROM KHACHHANG WHERE MaKH = @MaKH;

COMMIT;

--Giao tác xoá dịch vụ và các liên kết trong chi tiết hóa đơn:

BEGIN TRANSACTION;

-- Lấy mã dịch vụ và xoá chi tiết hóa đơn liên quan

DECLARE @MaDV CHAR(5);

SELECT @MaDV = MaDV FROM DichVu WHERE MaDV = 'DV003';

DELETE FROM CT\_HOADON WHERE MaDV\_Thuoc = @MaDV;

-- Xoá dịch vụ

DELETE FROM DichVu WHERE MaDV = @MaDV;

COMMIT;

--Giao tác cập nhật số lượng tồn kho của một loại thuốc:

BEGIN TRANSACTION;

-- Cập nhật số lượng tồn kho của thuốc

UPDATE THUOC

SET SLTonKho = SLTonKho - 10

WHERE MaThuoc = 'T001';



COMMIT;

--Giao tác cập nhật thông tin một đơn hàng và chi tiết đơn hàng:

BEGIN TRANSACTION;

-- Cập nhật thông tin đơn hàng

UPDATE HOADON

SET PhiThanhToan = 1800000

WHERE MaHD = 1;

-- Cập nhật chi tiết đơn hàng

UPDATE CT\_HOADON

SET SL = 5, ThanhTien = 900000

WHERE MaHD = 1 AND MaDV\_Thuoc = 'DV001';

COMMIT;

--Giao tác cập nhật lịch hẹn và thông tin bệnh nhân:

BEGIN TRANSACTION;

-- Cập nhật lịch hẹn

UPDATE LichHen

```
SET ThoiGianHen = '15:00:00'
```

```
WHERE MaLichHen = 2;
```

```
-- Cập nhật thông tin bệnh nhân
```

```
UPDATE KHACHHANG
```

```
SET HoTenKH = 'Tran Van F', NgaySinhKH = '1988-11-20'
```

```
WHERE MaKH = 5;
```

```
COMMIT;
```

```
--Giao tác xoá một khách hàng và liên kết liên quan:
```

```
BEGIN TRANSACTION;
```

```
-- Lấy mã khách hàng và xoá liên kết liên quan
```

```
DECLARE @MaKHToremove INT;
```

```
SELECT @MaKHToremove = MaKH FROM KHACHHANG WHERE HoTenKH = 'Nguyen Van E';
```

```
DELETE FROM LichHen WHERE MaKH = @MaKHToremove;
```

```
DELETE FROM HoSoBN WHERE MaKH = @MaKHToremove;
```

```
DELETE FROM KHACHHANG WHERE MaKH = @MaKHToremove;
```

COMMIT;

--Giao tác thêm một nhân viên mới và liên kết với tài khoản:

BEGIN TRANSACTION;

-- Thêm nhân viên mới

```
INSERT INTO NhanVien (HoTenNV, NgaySinhNV, DiaChiNV, SDTNV, EmailNV)  
VALUES ('Nguyen Thi G', '1995-03-10', 'Vung Tau', 555555556, 'ntg@gmail.com');
```

-- Lấy thông tin nhân viên vừa thêm

```
DECLARE @MaNVToAdd INT, @SDTNVToAdd INT;  
SELECT @MaNVToAdd = MaNV, @SDTNVToAdd = SDTNV FROM NhanVien WHERE SDTNV = 555555556;
```

-- Thêm tài khoản cho nhân viên

```
INSERT INTO TAIKHOAN (SDT, MatKhau, LoaiTK)  
VALUES (@SDTNVToAdd, 'newpassword', '0');
```

COMMIT;