

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	- Sửa lỗi tình huống tranh chấp: LOST UPDATE, UNREPEATEABLE READ	100%
2	20127423	Đinh Thành Danh	- Sửa lỗi tình huống tranh chấp: DIRTY READ, PHANTOM READ	100%
3	20127503	Dương Hiền Lê Hoàng	- Sửa lỗi tình huống tranh chấp: DIRTY READ, UNREPEATEABLE READ	100%
4	20127204	Nguyễn Phụng Khanh	- Sửa lỗi tình huống tranh chấp: LOST UPDATE, UNREPEATEABLE READ	100%

XỬ LÝ 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

ERR01: Dirty read(FIX) 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
<u>Input:</u> @MaKH @SDTKH <u>Output:</u>		<u>Input:</u> @MaKH <u>Output:</u>	
		SET TRAN ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Kiểm tra thông tin (1) khách hàng IF EXISTS (SELECT * FROM KHACHHANG WHERE MaKH = @MaKH) BEGIN PRINT N'Không tồn khách hàng đó' ROLLBACK TRAN RETURN 0 END	S(KHACHHANG) //Xin khoá đọc trên bảng KHACHHANG với dk MaKH = @MaKH.		

B2: Cập nhật thông tin vào bảng KHACHHANG UPDATE KHACHHANG SET SDTKH = @SDTKH Where MaKH = @MaKH;	U,X(KHACHHANG) //Xin khóa update trên bảng KHACHHANG với dk MaKH = @MaKH		
WAITFOR DELAY '0:0:10';			
		BEGIN TRAN	
		B1: Kiểm tra thông tin khách hàng có tồn tại không IF EXISTS (SELECT * FROM KHACHHANG WHERE MaKH = @MaKH) BEGIN PRINT N'Không tồn khách hàng đó' ROLLBACK TRAN RETURN 0 END	S(KHACHHANG) //Không cấp khóa cho S do UpdatedThong TinKH đang giữ khóa X
		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

ERR01: UNREPEATEABLE READ 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
Input: @MaHS @NgayKham @MaKH Output:		Input: @MaHS Output:	
		SET TRAN ISOLATION LEVEL REPEATABLE READ	Thiết lập khóa S khi đọc và giữ đến khi hết giao tác
		BEGIN TRAN	
		B1: Kiểm tra thông tin khách hàng có tồn tại không IF EXISTS (SELECT * FROM HoSoBN WHERE MaHS = @MaHS) BEGIN	S(HoSoBN) //Xin khoá đọc trên bảng HoSoBN với

		PRINT N'Không tồn hồ sơ đó' ROLLBACK TRAN RETURN 0 END	đk MaHS = @MaHS.
		WAITFOR DELAY '0:0:5';	
BEGIN TRAN			
B1: Kiểm tra thông tin hồ sơ khách hàng IF EXISTS (SELECT * FROM HoSoBN WHERE MaHS = @MaHS) BEGIN PRINT N'Không tồn hồ sơ đó' ROLLBACK TRAN RETURN 0 END			
B2: Cập nhật thông tin vào bảng KHACHHANG UPDATE HoSoBN SET NgayKham = @NgayKham, MaKH = @MaKH Where MaHS = @MaHS;	U,X(HoSoBN) //Không xin được khóa trên bảng HoSoBN vì LayThongTinH oSo đang giữ khóa		
WAITFOR DELAY '0:0:5';			
		B2: Xem thông tin hồ sơ khách hàng	S(HoSoBN) //Xin khoá đọc trên bảng

		SELECT * FROM HoSoKH Where MaKH = @MaKH;	HoSoBN với đk MaKH = @MaKH.
COMMIT		COMMIT	

II. 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.

FIX01: 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>Input:</u> <u>@MaDV = 'DV001'</u> <u>@TenDV = 'Kham rang tong quat'</u> <u>@Loai = 'Dich vu nhanh'</u> <u>@Tien = '150'</u> <u>Output:</u> <u>1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u>		<u>Input:</u> <u>@MaDV = 'DV001'</u> <u>@TenDV = 'Kham rang tong quat'</u> <u>@Loai = 'Dich vu nhanh'</u> <u>@Tien = '150'</u> <u>Output:</u> <u>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 WITH (XLOCK) WHERE MADV = 'DV001') BEGIN PRINT 'Dịch vụ ' + @MaDV + N' Không Tồn Tại' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(DICHVU)</p> <p>//Xin khoá đọc trên bảng DICHVU với điều kiện MADV = 'DV001'</p>		
<pre> WAITFOR DELAY '0:0:5' </pre>			
		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 WITH (XLOCK) WHERE MADV = 'DV001') BEGIN PRINT 'Dịch vụ ' + @MaDV + N' Không Tồn Tại' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(DICHVU)</p> <p>//Xin khoá đọc trên bảng DICHVU với điều kiện MADV = 'DV001' nhưng do Sp_UpdateDichVu_1 đang giữ khóa S nên không xin được</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', 150</p> <p>UPDATE DichVu SET</p> <p>TenDV = @TenDV,</p> <p>Loai = @Loai,</p> <p>Tien = @Tien</p> <p>WHERE MaDV = @MaDV</p>	<p>U,X(DICHVU)</p> <p>//Xin khoá update trên bảng DICHVU với điều kiện MADV = 'DV001'</p>		
COMMIT			

		<p>B1: Kiểm tra thông tin mã dịch vụ(MADV) có tồn tại hay không ?</p> <pre> IF NOT EXISTS (SELECT * FROM DICHVU WITH (XLOCK) WHERE MADV = 'DV001') BEGIN PRINT 'Dịch vụ ' + @MaDV + N' Không Tồn Tại' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(DICHVU)</p> <p>//Xin khoá đọc trên bảng DICHVU với điều kiện MADV = 'DV001' do Sp_UpdateDichVu_1 đã nhả khóa S nên xin được</p>
--	--	--	--

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

FIX02: UNREPEATABLE READ 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
<u>Input:</u> <u>@SDT= '111111112'</u> <u>@MATKHAU = '12345'</u> <u>@LoaiTK = '1'</u> <u>Output:</u> <u>1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u>		<u>Input:</u> <u>@SDT = '111111112',</u> <u>@MATKHAU = '54321'</u> <u>Output:</u> <u>1 biến int có giá trị 1 hoặc 0. 1 – Thành công ngược lại 0 – Thất bại</u>	
BEGIN TRAN			

SET TRAN ISOLATION LEVEL REPEATABLE READ	Thiết lập khóa S khi đọc và giữ đến khi hết giao tác		
B1: Xử lý lấy thông tin mã tài khoản nha sĩ SELECT @Ma = NS.MaNS FROM TaiKhoan A INNER JOIN NhaSi NS ON A.SDT = NS.SDTNS WHERE A.SDT = <u>'111111112'</u> AND A.MatKhu = <u>'12345'</u> AND A.LoaiTK = <u>'1'</u>	S(TAIKHOAN) //Xin khoá đọc trên bảng TaiKhoan với điều kiện TENDANGNHAP = '111111112', MATKHAU = '12345', LoaiTK = '1'		
WAITFOR DELAY '0:0:5'			
		BEGIN TRAN	

		<p>B1: Kiểm tra xem tài khoản có tồn tại hay không?</p> <pre> IF NOT EXISTS (SELECT * FROM TAIKHOAN WHERE SĐT = '111111112') BEGIN PRINT 'Tài khoản Không Tồn Tại' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(TAIKHOAN)</p> <p>//Không cấp khóa cho S do Sp_NV_LayTongTinTK đang giữ khóa X</p>
		<p>B2: Xử lý cập nhật mật khẩu</p> <pre> UPDATE TAIKHOAN SET MATKHAU = '54321' WHERE SĐT = '111111112' </pre>	<p>U,X(TAIKHOAN)</p> <p>//Không cấp khóa cho S do Sp_NV_LayTongTinTK đang giữ khóa X</p>

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

III. Sinh viên thực hiện: Nguyễn Phụng 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

ERR01: LOST UPDATE 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			
Updated SLTonKhoThuoc 1	Khóa	UpdatedSoLuongTonKhoThuoc 2	Khóa
Input: @SLTonKho Output:		Input: @SLTonKho Output:	
BEGIN TRAN			
B1: Kiểm tra thông tin thuốc có tồn tại hay không IF EXISTS (SELECT * FROM THUOC WITH (XLOCK) WHERE MaThuoc = @MaThuoc) BEGIN PRINT N 'Không tồn tại loại thuốc đó' ROLLBACK TRAN RETURN 0 END	S(THUOC) //Xin khoá đọc trên bảng THUOC với đk MaThuoc = @MaThuoc.		
WAITFOR DELAY '0:0:5'			
		BEGIN TRAN	
		B1: Kiểm tra thông tin thuốc có tồn tại hay không IF EXISTS (SELECT * FROM THUOC WITH (XLOCK)	S(THUOC) //Xin khoá đọc trên bảng THUOC với đk

		WHERE MaThuoc = @MaThuoc) BEGIN PRINT N 'Không tồn tại loại thuốc đó' ROLLBACK TRAN RETURN 0 END	MaThuoc = @MaThuoc. nhưng do Updated_SLT onKhoThuoc_ 1 đang giữ khóa S nên không xin được
B2: Cập nhật thông tin SLTonKho vào bảng THUOC UPDATE THUOC SET @SLTonKho = 20 where MaThuoc = @MaThuoc;	U,X(THUOC) //Xin khóa update trên bảng THUOC với dk MaThuoc = @MaThuoc		
		B1: Kiểm tra thông tin thuốc có tồn tại hay không IF EXISTS (SELECT * FROM THUOC WITH (XLOCK) WHERE MaThuoc = @MaThuoc) BEGIN PRINT N 'Không tồn tại loại thuốc đó' ROLLBACK TRAN RETURN 0 END	S(THUOC) //Xin khoá đọc trên bảng THUOC với dk MaThuoc = @MaThuoc do Updated_SLT onKhoThuoc_ 1 đã nhả khóa S nên xin được
		B2: Cập nhật thông tin SLTonKho vào bảng THUOC	U,X(THUOC)

		UPDATE THUOC SET @SLTonKho = 15 where MaThuoc = @MaThuoc;	//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.

ERR01: UNREPEATABLE READ 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
Input: @SDTNS Output: @SDTNS='123456789'		Input: @SDTNS='987654321' Output: @SDTNS='123456789'	
BEGIN TRAN			
SET TRAN ISOLATION LEVEL REPEATABLE READ	Thiết lập khóa S khi đọc và giữ đến hết giao tác		
B1: Kiểm tra thông tin nha sĩ có tồn tại hay không IF EXISTS (SELECT * FROM NhaSi	S(NhaSi) //Xin khoá đọc trên bảng		

WHERE MaNS = @MaNS) BEGIN PRINT N 'Không tồn tại nha sĩ đó' ROLLBACK TRAN RETURN 0 END	NhaSi với đk MaNS = @MaNS.		
B2: Xem thông tin số điện thoại nha sĩ SELECT SDTNS FROM NhaSi WHERE MaNS = @MaNS;	S(NhaSi) //Xin khoá đọc trên bảng NhaSi với đk 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	S(NhaSi) //Không cấp khóa S do LaySoDienThoaiNS đang giữ khóa X
		B2: Cập nhật số điện thoại nha sĩ UPDATE NhaSi SET @SDTNS = 123456789 where MaNS = @MaNS;	U,X(NhaSi) //Không cấp khóa S do

			LaySoDienTh oaiNS đang giữ khóa X
B2: 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			
B3: Xem thông tin số điện thoại nha sĩ _ lần 2 SELECT SDTNS FROM NhaSi WHERE MaNS = @MaNS;	S(NhaSi) //Xin khoá đọc trên bảng NhaSi với đk MaNS = @MaNS.		
COMMIT		COMMIT	

IV. Sinh viên thực hiện: Đinh Thành Danh

3. 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

ERR01: PHANTOM READ			
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
<u>Input:</u> <u>@NgayHen</u> <u>@ThoiGianHen</u> <u>@HoTenKH</u> <u>@MaKH</u> <u>@MaNS</u> <u>Output: 1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u>			
SET TRANSACTION ISOLATION LEVEL		SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	

		BEGIN TRAN	
	.	<p>B1: Xem danh sách lịch hẹn:</p> <p>SELECT Ngay,ThoiGian,MaNS FROM LichCaNhanNS</p> <p>return 1</p>	<p>S(LichCaNhanNS) //Xin khoá đọc trên bảng LichCaNhanNS</p>
		WAITFOR DELAY '0:0:10'	
BEGIN TRAN			
<p>B1: Kiểm tra ngày và thời gian lịch hẹn xem có chưa ?</p> <p>IF(EXISTS(SELECT * FROM LichHen WHERE Ngay = '01-10-2023' AND ThoiGian = '08:30:00'))</p> <p>RETURN -1</p>	<p>S(LichHen)</p> <p>//Xin khoá đọc trên bảng LichHen với điều kiện</p> <p>Ngay = '01-10-2023' và ThoiGian = '08:30:00'</p>		

<p>B2: Kiểm tra mã nha sĩ xem có chưa ?</p> <pre>IF(EXISTS(SELECT * FROM LichHen WHERE MaNS = '1')</pre> <p>RETURN -1</p>	<p>S(LichHen)</p> <p>//Xin khoá đọc trên bảng</p> <p>LichHen với điều kiện MaNS = '1'</p>		
<p>B3 : Thêm lịch hẹn mới</p> <pre>INSERT INTO LichHen(NgayHen, ThoiGianHen, TenKH, MaKH, MaNS) VALUES ('01-10-2023', '08:30:00', N'Nguyen Van E', '3', '2')</pre> <p>return 1</p>	<p>X(LichHen)</p> <p>//xin khoá ghi trên bảng LichHen để thêm lịch hẹn mới</p>		
<p>B4: Cập nhật cuộc hẹn:</p> <pre>UPDATE LichHen SET NgayHen = @NgayHen AND ThoiGianHen = @ThoiGianHen WHERE MAKH = @MaKH</pre>	<p>UX(KHACHHANG) //xin khoá ghi trên bảng DOITAC</p>		

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

4. 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.

ERR02: DIRTY READ T1 (User = Khách hàng): Khách hàng đang thực hiện cập nhật thông tin T2 (User = Nhân viên): Nhân viên đang xem thông tin cũ của khách hàng			
Sp_XemThongTinCaNhan	Khóa	Sp_XemThongTinKhachHang	Khóa
<u>Input: @HoTenKH</u> <u>@NgaySinhKH</u> <u>@DiaChiKH</u> <u>@SDTKH</u>			

<u>@EmailKH</u> <u>Output:</u> 1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại			
		SET TRAN ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Kiểm tra thông tin (1) khách hàng IF EXISTS (SELECT * FROM KHACHHANG WHERE SDTKH = @SDTKH) BEGIN PRINT N'Không tồn tại khoản' ROLLBACK TRAN RETURN 0 END	S(KHACHHANG) //Xin khoá đọc trên bảng KHACHHANG với dk SDTKH = @SDTKH		
B2: Cập nhật thông tin vào bảng KHACHHANG UPDATE KHACHHANG SET EmailKH = @EmailKH Where SDTKH = @SDTKH;	U,X(KHACHHANG) //Xin khóa update trên bảng KHACHHANG G với dk		

	SDTKH = @SDTKH		
WAITFOR DELAY '0:0:05';			
		BEGIN TRAN	
		<p>B1: Kiểm tra thông tin khách hàng có tồn tại không</p> <p>IF EXISTS (SELECT * FROM KHACHHANG WHERE MaKH = @MaKH)</p> <p>BEGIN</p> <p> PRINT N'Không tồn khách hàng đó'</p> <p> ROLLBACK TRAN</p> <p> RETURN 0</p> <p>END</p>	<p>S(KHACHHANG)</p> <p>//Xin khoá đọc trên bảng KHACHHANG với đk MaKH = @MaKH.</p>
		<p>B2: Xem thông tin khách hàng</p> <p>SELECT * FROM KHACHHANG Where MaKH = @MaKH;</p>	<p>S(KHACHHANG)</p> <p>//Xin khoá đọc trên bảng KHACHHANG với đk MaKH = @MaKH.</p>
		COMMIT	

