

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


THÔNG TIN NHÓM




STT	MSSV	Họ tên	Công việc	% Hoàn thành
1	19127388	Bùi Nguyễn Nhật Hào	Tranh chấp 1, 2, 3, 4	100%
2	19127495	Lê Hoàng Nhật	Tranh chấp 5, 6, 7 , 11	100%
3	19127601	Trần Hữu Trọng	Tranh chấp 8, 9, 10, 12	100%


CÀI ĐẶT TÌNH HUỐNG TRANH CHẤP

I. Sinh viên thực hiện: Bùi Nguyễn Nhật Hào; Trần Hữu Trọng; Lê Hoàng Nhật

Tình huống 1: Cập nhật giá sản phẩm và xem thông tin sản phẩm




ERR01: Dirty read T1 (User = Đối tác): Đổi Mã số thuế T2 (User = Admin/Đối tác): Xem thông tin doanh nghiệp			
sp_CapNhatGiaSanPham	Khóa	sp_XemThongTinSanPham	Khóa
Input: @MaSP varchar(50), @GiaMoi smallmoney Output: Cập nhật lại giá sản phẩm		Input: @MaSP varchar(50) Output: Xuất ra thông tin sản phẩm	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
		BEGIN TRAN	
B1: Kiểm tra Mã sản phẩm có tồn tại hay không IF NOT EXISTS (SELECT * from sanpham where maps = @masp) BEGIN PRINT N'Mã sản phẩm này không tồn tại' ROLL BACK RETURN TRAN END	R(A) //Xin khoá đọc trên bảng SanPham		
			




		B1: Kiểm tra Mã sản phẩm có tồn tại hay không IF NOT EXISTS (SELECT * from sanpham where masp = @masp) BEGIN PRINT N'Mã sản phẩm này không tồn tại' ROLL BACK RETURN TRAN END	R(A) //Xin khoá đọc trên bảng SanPham
			
B2: Cập nhật giá sản phẩm Update SANPHAM SET GIA = @GIAMOI WHERE MASP = @MASP	X(A) //Xin khoá ghi trên bảng SanPham.		
WAITFOR DELAY '00:00:20'			
			
		Select * from sanpham with (NOLOCK) WHERE masp = @masp	Không xin khóa
			
IF @GiaMoi < 0 //Khi Update bảng doanh nghiệp, nếu mã số thuế bị trùng thì sẽ vào trường hợp này			



ROLLBACK			
			
		Commit	

Tình huống 2: Tăng giá sản phẩm và giảm giá sản phẩm

ERR01: Lost Update T1 (User = Đối tác): Tăng giá sản phẩm T2 (User = Đối tác): Giảm giá sản phẩm			
sp_TangGiaSanPham	Khóa	sp_GiamGiaSanPham	Khóa
Input: MaSP, SoTienTang Output: Tăng giá sản phẩm		Input: MaSP, SoTienGiam Output: Giảm giá sản phẩm	
SET TRANSACTION ISOLATION LEVEL Read Committed		SET TRANSACTION ISOLATION LEVEL Read COMMITTED	
BEGIN TRAN			
		BEGIN TRAN	
B1: Kiểm tra thông tin MaSP có tồn tại IF NOT EXISTS (Select * from SanPham sp where sp.MaSP = MaSP) BEGIN ROLL BACK; Return ; END	R(SanPham) //Xin khóa đọc trên bảng Sản phẩm		



			
		B1: Kiểm tra thông tin MaSP có tồn tại IF EXISTS (Select * from SanPham sp where sp.MaSP = MaSP) BEGIN ROLL BACK; Return ; END	R(SanPham) //Xin khóa đọc trên bảng Sản phẩm
			
B2: Kiểm tra thông tin GiaMoi có hợp lệ IF (SoTienTang <= 0) BEGIN Roll Back; Return; END			
			

		B2: Kiểm tra thông tin GiaMoi có hợp lệ IF (SoTienGiam <= 0) BEGIN Roll Back; Return; END	
			
B3: Lấy biến GiaTien Set Gia = (Select Gia from SanPham sp where sp.MaSP = MaSP)	R(SanPham) //Xin khóa đọc trên bảng Sản phẩm		
			
		B3: Lấy biến GiaTien Set @Gia = (Select Gia from SanPham sp where sp.MaSP = MaSP)	R(SanPham) //Xin khóa đọc trên bảng Sản phẩm
			
B4: Tăng giá tiền lên một đơn vị	X(SanPham)		

Update table SanPham sp Set Gia = @Gia + SoTienTang where sp.MaSP = MaSP	//Xin khóa ghi trên bảng Sản phẩm		
		B4: Tăng giá tiền lên một đơn vị Update table SanPham sp Set Gia = @Gia - SoTienGiam where sp.MaSP = MaSP	X(A) //Xin khóa ghi trên bảng Sản phẩm nhưng không được
		COMMIT	
COMMIT			

Tình huống 3: Xóa tài khoản nhân viên và Update tài khoản nhân viên




ERR01: PhanTom Read T1 (User = Admin): Xóa tài khoản nhân viên T2 (User = Admin): Update tài khoản nhân viên			
sp_XoaTaiKhoan	Khóa	sp_UpdateTaiKhoan	Khóa
Input: ID		Input: ID, DataUpdate(Email, DiaChi, ...)	
Output: Xóa tài khoản nhân viên ID		Output: Update tài khoản nhân viên	
SET TRANSACTION ISOLATION LEVEL Read Committed		SET TRANSACTION ISOLATION LEVEL Read Committed	
BEGIN TRAN			
		BEGIN TRAN	


B1: Kiểm tra tài khoản có tồn tại hay không IF Not EXISTS (Select * from TKNhanVien tk where tk.ID = ID) BEGIN ROLL BACK; Return ; END	R(TKNhanVien) //Xin khóa đọc trên bảng Tài Khoản Nhân Viên		
			
		B1: Kiểm tra tài khoản có tồn tại hay không IF Not EXISTS (Select * from TKNhanVien tk where tk.ID = ID) BEGIN ROLL BACK; Return ; END	R(TKNhanVien) //Xin khóa đọc trên bảng Tài Khoản Nhân Viên
			
B2: Xóa tài khoản Delete from TKNhanVien tknv where tknv.ID = ID	X(tkNhanVien) //Xin khóa ghi trên bảng TKNhanVien		

↓		<p>B2: Update Tài Khoản (Ví dụ cập nhật địa chỉ) Update TKNhanVien tk Set tk.DiaChi = DiaChi where tk.ID = ID</p>	<p>X(tkNhanVien) //Xin khóa ghi trên bảng TKNhanVien</p>
	↗	COMMIT	
COMMIT			

Tình huống 4: Xóa sản phẩm và Update sản phẩm




<p>ERR01: Unrepeatable Read T1 (User = Admin): Xóa sản phẩm. T2 (User = Admin): Update sản phẩm.</p>			
sp_XoaSanPham	Khóa	sp_UpdateSanPham	Khóa
<p>Input: MaSP Output: Xóa sản phẩm</p>		<p>Input: MaSP, TenSP Output: Update sản phẩm</p>	
<p>SET TRANSACTION ISOLATION LEVEL Read Committed</p>		<p>SET TRANSACTION ISOLATION LEVEL Read Committed</p>	
BEGIN TRAN			
		BEGIN TRAN	
<p>B1: Kiểm tra sản phẩm có tồn tại hay không IF Not EXISTS (Select * from SanPham sp where sp.MaSP = MaSP)</p>	R(SanPham)		

BEGIN ROLL BACK; Return ; END	//Xin khóa đọc trên bảng sản phẩm		
			
		B1: Kiểm tra sản phẩm có tồn tại hay không IF Not EXISTS (Select * from SanPham sp where sp.MaSP = MaSP) BEGIN ROLL BACK; Return ; END	R(SanPham) //Xin khóa đọc trên bảng sản phẩm
			
B2: Xóa sản phẩm Delete from SanPham sp where sp.MaSP = MaSP	X(SanPham) //Xin khóa ghi trên bảng sản phẩm		
		B2: Update Sản Phẩm Update SanPham sp Set sp.TenSP = TenSP where sp.MaSP = MaSP	X(tkNhanVien) //Xin khóa ghi trên bảng TKNhanVien

		COMMIT	
COMMIT			

Tình huống 4: Xóa chi nhánh và sửa chi nhánh



ERR01: Phantom Read T1 (User = Đối tác): Xóa chi nhánh. T2 (User = Đối tác): Sửa chi nhánh.			
sp_XoaChiNhanh	Khóa	sp_SuaChiNhanh	Khóa
<u>Input:</u> @MaDoanhNghiep varchar(50), @MaChiNhanh varchar(50) <u>Output:</u> Xóa chi nhánh		<u>Input:</u> @MaDoanhNghiep varchar(50), @MaChiNhanh varchar(50), @DiaChi varchar(255) <u>Output:</u> Sửa chi nhánh	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Kiểm tra chi nhánh có tồn tại hay không ? IF Not EXISTS (Select * from ChiNhanh cn where cn.MaDoanhNghiep = @MaDoanhNghiep and cn.MaChiNhanh = @MaChiNhanh) BEGIN ROLL BACK Return Tran END	R(ChiNhanh) //Xin khoá đọc trên bảng Chi nhánh.		

			
		B1: Kiểm tra chi nhánh có tồn tại hay không ? IF Not EXISTS (Select * from ChiNhanh cn where cn.MaDoanhNghiep = @MaDoanhNghiep and cn.MaChiNhanh = @MaChiNhanh) BEGIN ROLLBACK TRAN Return END	R(ChiNhanh) //Xin khoá đọc trên bảng Chi nhánh.
			
B2: Xóa chi nhánh Delete from ChiNhanh cn where cn.MaDoanhNghiep = @MaDoanhNghiep and cn.MaChiNhanh =@ MaChiNhanh.	X(ChiNhanh) //Xin khoá ghi trên bảng Chi nhánh		
COMMIT			
			
		B2: Cập nhật chi nhánh	
		Update ChiNhanh	X(ChiNhanh)

		SET DiaChi = @DiaChi where MaDoanhNghiep = @MaDoanhNghiep and MaChiNhanh = @ MaChiNhanh.	//Xin khoá ghi trên bảng chi nhánh
		COMMIT	

Tình huống 5: Xóa tài khoản nhân viên tranh chấp đăng nhập của nhân viên

ERR05: Phantom Read T1 (User = Admin): thực hiện xóa tài khoản của nhân viên. T2 (User = Nhân viên): đăng nhập vào tài khoản của mình			
sp_XoaTaiKhoanNhanVien	Khóa	sp_DangNhapNhanVien	Khóa
<u>Input:</u> @taikhoan varchar (50) <u>Output:</u> Xóa tài khoản		<u>Input:</u> @taikhoan varchar(50), @matkhau varchar(50) <u>Output:</u> Đăng nhập ở tài khoản nhân viên	
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE		SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
		BEGIN TRAN	
		B1: Kiểm tra tài khoản (tên đăng nhập) IF NOT EXISTS (SELECT * FROM TKNhanVien tknv WHERE tknv.id = @taikhoan BEGIN Rollback transaction Return END	R(TKNhanVi en)
		B2: Kiểm tra xem tài khoản bị khoá hay không. IF 0 = (SELECT tknv.trangthai FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) BEGIN Rollback transaction Return	

		END	
		B3: Kiểm tra mật khẩu. IF @matkhau != (SELECT * FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) BEGIN Rollback transaction Return END	
		WAITFOR DELAY '0:0:010'	
BEGIN TRAN			
B1: Kiểm tra tài khoản(1) IF NOT EXISTS (SELECT * FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) BEGIN Rollback transaction Return END	R(TKNhanVien)		
B2: Xóa tài khoản DELETE FROM TKNhanVien WHERE id = @taikhoan	X(TKNhanVien)		
COMMIT			



		<p>B4: In thông tin tài khoản mật khẩu người dùng</p> <pre> print '-----Dang nhap thanh cong-----' print '-Thong tin user' Declare @tk varchar(50) Declare @mk varchar(50) Declare @tt int Set @tk = (SELECT tknv.id FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) Set @mk = (SELECT tknv.mk FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) Set @tt = (SELECT tknv.TRANGTHAI FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) Print '-- Tai Khoan: ' + @tk Print '-- Mat khau: ' + @mk Print '-- Trang thai: ' + cast(@tt as varchar(10)) </pre>	R (TKNhanVien)
		COMMIT	

Tình huống 6: Đổi mật khẩu và đăng nhập của tất cả các phân hệ

VD: Phân hệ nhân viên.

<p>ERR06: Unrepeatable Read</p> <p>T1 (User = nhân viên): nhân viên đổi mật khẩu.</p> <p>T2 (User = nhân viên): nhân viên đăng nhập</p>			
sp_DoiMatKhau	Khóa	sp_DangNhapNhanVien	Khóa

<u>Input:</u> @taikhoan varchar(50), @matkhaucu varchar(50), @matkhaumoi varchar(50) <u>Output:</u> đổi mật khẩu		<u>Input:</u> @taikhoan varchar(50), @matkhou varchar(50) <u>Output:</u> đăng nhập ở tài khoản nhân viên	
SET TRANSACTION ISOLATION LEVEL repeatable read		SET TRANSACTION ISOLATION LEVEL repeatable read	
		BEGIN TRAN	
		B1: Kiểm tra tài khoản (tên đăng nhập) IF NOT EXISTS (SELECT * FROM TKNhanVien tknv WHERE tknv.id = @taikhoan BEGIN Rollback transaction Return END	R(TKNhanVie n)
		B2: Kiểm tra xem tài khoản bị khoá hay không. IF 0 = (SELECT tknv.trangthai FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) BEGIN Rollback transaction Return END	R(TKNhanVie n)
		B3: Kiểm tra mật khẩu. IF @matkhou != (SELECT * FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) BEGIN Rollback transaction Return END	R(TKNhanVie n)


		WAITFOR DELAY '0:0:010''	
BEGIN TRAN			
B1: Kiểm tra tài khoản(1) IF NOT EXISTS (SELECT * FROM TKNhanVien tknv WHERE tknv.id = @taikhoan BEGIN Rollback transaction Return END	R(TKNhanVie n)		
B2: Kiểm tra mật khẩu cũ IF @matkhaucu != (SELECT * FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) BEGIN Rollback transaction Return END	R(TKNhanVie n)		
B3: Cập nhật khẩu mới UPDATE TKNhanVien SET mk= @matkhaumoi WHERE id = @taikhoan;	X(TKNhanVie n)		
Commit			

		B4: In thông tin tài khoản mật khẩu người dùng print '-----Dang nhap thanh cong-----' print '-Thong tin user' Declare @tk varchar(50) Declare @mk varchar(50) Declare @tt int Set @tk = (SELECT tknv.id FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) Set @mk = (SELECT tknv.mk FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) Set @tt = (SELECT tknv.TRANGTHAI FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) Print '-- Tai Khoan: ' + @tk Print '-- Mat khau: ' + @mk Print '-- Trang thai: ' + cast(@tt as varchar(10))	R (TKNhanVien)
		COMMIT	

Tình huống 7: Đăng nhập và admin khoá tài khoản

ERR07: unrepeatable read T1 (User = admin): thực hiện khoá tài khoản của nhân viên đó T2 (User = Nhân viên): thực hiện đăng nhập.			
sp_KhoaTaiKhoanNhanVien	Khóa	sp_DangNhapNhanVien	Khóa
Input: @taikhoan varchar(50)		Input: @taikhoan varchar(50),	



<u>Output:</u> Khóa tài khoản nhân viên		@matkhau varchar(50)	
SET TRANSACTION ISOLATION LEVEL repeatable read		<u>Output:</u> Đăng nhập tài khoản nhân viên	
		SET TRANSACTION ISOLATION LEVEL repeatable read	
		BEGIN TRAN	
		B1: Kiểm tra tài khoản (tên đăng nhập) IF NOT EXISTS (SELECT * FROM TKNhanVien tknv WHERE tknv.id = @taikhoan BEGIN Rollback transaction Return END	R(TKNhanVien)
		B2: Kiểm tra xem tài khoản bị khoá hay không. IF 0 = (SELECT tknv.trangthai FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) BEGIN Rollback transaction Return END	R(TKNhanVien)
		B3: Kiểm tra mật khẩu. IF @matkhau != (SELECT * FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) BEGIN Rollback transaction Return END	R(TKNhanVien)
		WAITFOR DELAY '0:0:010'	
BEGIN TRAN			

B1: Kiểm tra tài khoản IF NOT EXISTS (SELECT * FROM TKNhanVien tknv WHERE tknv.id = @taikhoan BEGIN Rollback transaction Return END	R (TKNhanVien)		
B2: Cập nhật trạng thái của tài khoản UPDATE TKNhanVien SET trangthai = 0 WHERE id = @taikhoan;	X (TKNhanVien)		
COMMIT			
		B4: In thông tin tài khoản mật khẩu người dùng print '-----Dang nhap thanh cong-----' print '-Thông tin user' Declare @tk varchar(50) Declare @mk varchar(50) Declare @tt int Set @tk = (SELECT tknv.id FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) Set @mk = (SELECT tknv.mk FROM TKNhanVien tknv WHERE tknv.id = @taikhoan)	R (TKNhanVien)

		Set @tt = (SELECT tknv.TRANGTHAI FROM TKNhanVien tknv WHERE tknv.id = @taikhoan) Print '-- Tai Khoan: ' + @tk Print '-- Mat khau: ' + @mk Print '-- Trang thai: ' + cast(@tt as varchar(10))	
		COMMIT	

Tình huống 8: Thêm một sản phẩm vào đơn hàng và sửa giá sản phẩm


ERR08: Unrepeatable Read T1 (User = Đối tác): Sửa giá sản phẩm T2 (User = Khách hàng): Thêm một sản phẩm vào đơn hàng			
sp_SuaGiaSanPham	Khóa	sp_DatHang	Khóa
<u>Input:</u> @MaSP, @GiaMoi <u>Output:</u> Sửa giá sản phẩm		<u>Input:</u> @MaSP varchar(50), @MaDH varchar(50), @Gia varchar(50), @SLSP varchar(50) <u>Output:</u> Thêm một sản phẩm vào đơn hàng	
SET TRANSACTION ISOLATION LEVEL Read Committed		SET TRANSACTION ISOLATION LEVEL Read committed	
		BEGIN TRAN	
BEGIN TRAN			
		B1: Kiểm tra mã Đơn hàng IF not exists(select * from donhang where @MaDH = MaDH) begin ROLLBACK TRAN	R(DonHang)



		RETURN end	
			
B1: Kiểm tra thông tin MaSP có tồn tại IF EXISTS (Select * from SanPham sp where sp.MaSP = @MaSP) BEGIN. ROLLBACK; Return; END	R(SanPham)		
B2: Kiểm tra giá mới có hợp lệ không IF (@giamoi <= 0) BEGIN Roll Back; Return; END			
B3: Cập nhật giá Update SanPham SP set SP.GIA = @giamoi where SP.MaSP = @MaSP	X(SanPham)		
			

COMMIT			
		<p>B2 : Cập nhật phí sản phẩm của đơn hàng</p> <p>Update DonHang set PhiSanPham = PhiSanPham + SLSP * Gia where MaDH = @MaDH</p>	X(SanPham)
		↓	
		<p>B3 : Cập nhật sản phẩm vào DH_SP</p> <p>Update_DHSP(MaSP, MaDH)</p>	X(SanPham)
		COMMIT	

Tình huống 9: Sửa giá sản phẩm: Đổi mật khẩu


ERR09: Phantom Read T1 (User = nhân viên): nhân viên đổi mật khẩu. T2 (User = admin): xóa tài khoản			
sp_DoiMatKhau	Khóa	sp_XoaTaiKhoan	Khóa
<u>Input:</u> @taikhoan varchar(50), @matkhaucu varchar(50), @matkhaumoi varchar(50) <u>Output:</u> Đổi mật khẩu		<u>Input:</u> @taikhoan varchar(50), @matkhaucu varchar(50) <u>Output:</u> Xóa tài khoản	



SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Kiểm tra tài khoản(1) IF NOT EXISTS (SELECT * FROM TKNhanVien tknv WHERE tknv.tk = @taikhoan BEGIN Rollback transaction Return END	R (TKNhanVie n)		
B2: Kiểm tra mật khẩu cũ IF @matkhaucu != (SELECT * FROM TKNhanVien tknv WHERE tknv.tk = @taikhoan) BEGIN Rollback transaction Return END	R (TKNhanVie n)		
			
		BEGIN TRAN	
		B1: Kiểm tra tài khoản có tồn tại không IF NOT EXISTS (SELECT * FROM TKNhanVien tknv WHERE tknv.tk = @taikhoan BEGIN Rollback transaction Return END	X (TKNhanVie n)

			
		B2: Xóa tài khoản Delete FROM TKNhanVien tknv WHERE tknv.tk = @taikhoan	R(TKNhanVien)
		COMMIT	
B3: Cập nhật khẩu mới UPDATE TKNhanVien SET mk= @matkhaumoi WHERE tk = @taikhoan;	R(TKNhanVien)		
Commit			

Tình huống 10: Đổi mật khẩu



ERR010: Unrepeatable Read T1 (User = all): đổi mật khẩu. T2 (User = all): đổi mật khẩu.			
sp_DoiMatKhau	Khóa	sp_DoiMatKhau	Khóa
<u>Input:</u> @taikhoan varchar(50), @matkhaucu varchar(50), @matkhaumoi varchar(50) <u>Output:</u> đổi mật khẩu		<u>Input:</u> @taikhoan varchar(50), @matkhaucu varchar(50), @matkhaumoi varchar(50) <u>Output:</u> đổi mật khẩu	

SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Kiểm tra tài khoản(1) IF NOT EXISTS (SELECT * FROM TKNhanVien tknv WHERE tknv.tk = @taikhoan BEGIN Rollback transaction Return END	R(TKNhanVie n)		
B2: Kiểm tra mật khẩu cũ IF @matkhaucu != (SELECT * FROM TKNhanVien tknv WHERE tknv.tk = @taikhoan) BEGIN Rollback transaction Return END	R(TKNhanVie n)		
			
		BEGIN TRAN	
		B1: Kiểm tra tài khoản(1) IF NOT EXISTS (SELECT * FROM TKNhanVien tknv WHERE tknv.tk = @taikhoan BEGIN Rollback transaction Return END	R(TKNhanVie n)

		B2: Kiểm tra mật khẩu cũ IF @matkhaucu != (SELECT * FROM TKNhanVien tknv WHERE tknv.tk = @taikhoan) BEGIN Rollback transaction Return END	R (TKNhanVie n)
		COMMIT	
			
B3: Cập nhật khẩu mới UPDATE TKNhanVien SET mk= @matkhaumoi WHERE tk = @taikhoan;	X (TKNhanVie n)		
Commit			
			
		B3: Cập nhật khẩu mới UPDATE TKNhanVien SET mk= @matkhaumoi WHERE tk = @taikhoan;	X (TKNhanVie n)
		Commit	


Tình huống 11: Cập nhật tài khoản Nhân Viên của admin và Đổi mật khẩu của nhân viên


ERR010: Conversion Deadlock			
T1 (User = Nhân Viên): Nhân Viên Đổi mật khẩu			
T2 (User = Quản Trị): Cập nhật trạng thái tài khoản			
sp_DoiMatKhauNV	Khóa	sp_CapNhatTrangThaiNV	Khóa
Input: @taikhoan varchar(50), @matkhaucu varchar(50), @matkhaumoi varchar(50) Output: Đổi mật khẩu nhân viên		Input: @taikhoan varchar(50), @TrangThai int Output: Cập nhật trạng thái tài khoản nhân viên	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ		SET TRANSACTION ISOLATION LEVEL REPEATABLE READ	
BEGIN TRAN			
B1: Kiểm tra tài khoản(1) IF NOT EXISTS (SELECT * FROM TKNhanVien tknv WHERE tknv.tk = @taikhoan BEGIN Rollback transaction Return END	R(TKNhanVien)		
	↘		
		BEGIN TRAN	
		B1: Kiểm tra tài khoản(1) IF NOT EXISTS (SELECT * FROM TKNhanVien tknv WHERE tknv.tk = @taikhoan BEGIN Rollback transaction	R(TKNhanVien)



		Return END	
			
B2: Cập nhật khẩu mới UPDATE TKNhanVien SET mk= @matkhaumoi WHERE tk = @taikhoan;	X(TKNhanVien)		
			
		B2 : Kiểm tra trạng thái và cập nhật IF (TrangThai != 0 and TrangThai != 1) Begin ROLLBACK TRANSACTION RETURN END	
		Update NhanVien Set TrangThai = @TrangThai Where TaiKhoan = @taikhoan	X(TKNhanVien) Deadlock

Tình huống 12: Xóa chi nhánh và Xóa chi nhánh khỏi hợp đồng

ERR010: Cycle Deadlock T1 (User = Đối tác): Xóa chi nhánh T2 (User = Đối tác): Xóa chi nhánh ra khỏi hợp đồng			
sp_XoaChiNhanh	Khóa	sp_XoaHopDongCuaChiNhanh	Khóa

<u>Input:</u> @MaChiNhanh varchar(50), @MaDoanhNghiep varchar(50) <u>Output:</u> Xóa chi nhánh		<u>Input:</u> @MaChiNhanh varchar(50), @MaDoanhNghiep varchar(50) <u>Output:</u> Xóa chi nhánh khỏi hợp đồng	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ		SET TRANSACTION ISOLATION LEVEL REPEATABLE READ	
BEGIN TRAN			
B1: Kiểm tra chi nhánh IF NOT EXISTS (SELECT * FROM ChiNhanh where MaChinhanh = @MaChiNhanh and MaDoanhNGhiep = @MaDoanhNGhiep) BEGIN Rollback transaction Return END	R(ChiNhanh)		
			
		BEGIN TRAN	
		B1: Kiểm tra chi nhánh có tồn tại IF NOT EXISTS (SELECT * FROM ChiNhanh where MaChinhanh = @MaChiNhanh and MaDoanhNGhiep = @MaDoanhNGhiepCu) BEGIN Rollback transaction Return END	R(ChiNhanh)

			
B2: Giảm số chi nhánh đăng kí trong hợp đồng xuống 1 đơn vị	R(TKNhanVien)		
Update HopDong Set SoChiNhanhDK = SoChiNhanhDK - 1 Where MaHD = (Select MaHD from ChiNhanh where @MaChiNhanh = MaChiNhanh and @MaDoanhNghiep = MaDoanhNGhiep)	X(HopDong)		
			
		B2 : Lấy mã hợp đồng Set @MaHD = (Select MaHD where @MaChiNhanh = MaChiNhanh and @MaHopDong = MaHopDong)	
		B2 : Xóa chi nhánh khỏi hợp đồng Update ChiNhanh Set MaHD = NULL Where MaChiNhanh = @MaChiNhanh and @MaHopDong = @MaHopDong	X(ChiNhanh)

			
B3: Xóa chi nhánh			
Delete ChiNhanh where MaDoanhNghiep = @MaDoanhNGhiep and MaChiNhanh = @MaChiNhanh	X(ChiNhanh)		
		B3 : Giảm đơn vị chi nhánh trong hợp đồng Update HopDong Set SoChiNhanhDK = SoChiNhanhDK - 1 Where MaHD = @MaHD	X(HopDong) Deadlock