

BÁO CÁO ĐỒ ÁN THỰC HÀNH

MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

GVHD: Tiết Gia Hồng, Hồ Thị Hoàng Vy

THÔNG TIN NHÓM

STT	MSSV	Họ tên	Công việc	% Hoàn thành
1	19120626	Hồ Minh Quân		
2	19120639	Lê Nam Thái Sơn		
3	19120640	Trần Minh Sơn		
4	19120674	Phạm Tân Tị		

STT	Chức năng 1	Người dùng	Chức năng 2	Người dùng	Lỗi tranh chấp
1	Đăng nhập	Người dùng	Khóa tài khoản	Hệ quản trị	unrepeatable read
2	Đặt hàng	Khách hàng	Cập nhật sản phẩm	Đối tác	unrepeatable read
3	Xem đơn hàng	Tài xế	Đặt hàng	Khách hàng	phantom read
4	Nhận đơn hàng	Tài xế	Đặt hàng nhưng bị lỗi Rollback	Khách hàng	DirtyRead
5	Nhận đơn hàng	Tài xế	Nhận đơn hàng	Tài xế	lost update

CÀI ĐẶT TÌNH HUỐNG TRANH CHẤP

I. Sinh viên thực hiện: 19120674

Tình huống 1: Khi người dùng đăng nhập và hệ quản trị khóa lại

ERR01: unrepeatable read T1 (User = người dùng (đối tác, khách hàng, ...)): Thực hiện đăng nhập T2 (User = hệ quản trị): Khóa tài khoản của user			
sp_login	Khóa	sp_enable_disable_login	Khóa
<u>Input:</u> @tk varchar(10), @mk varchar(10) <u>Output:</u>		<u>Input:</u> @tk varchar(10) <u>Output:</u>	
SET TRANSACTION ISOLATION LEVEL Read uncommitted		SET TRANSACTION ISOLATION LEVEL UNREPEATABLE READ	
BEGIN TRAN			
B1: Kiểm tra thông tin đăng nhập (1) if exists(SELECT * FROM TaiKhoan where @tk=taikhoan and @mk = matkhau and TrangThai='enable')	R(A)		
WAITFOR DELAY '00:00:20'			
		BEGIN TRAN	

		<p>B1: Tìm kiếm thông tin (1)</p> <pre>declare @stat varchar(10)= (select TrangThai from taikhoan where @tk=taikhoan) if(@stat='enable') set @stat='disable' else set @stat='enable'</pre>	R(A)
		<p>B2: Cập nhật thông tin</p> <pre>update taikhoan set TrangThai=@stat where TaiKhoan=@tk</pre>	X(A)
		COMMIT	
<pre>SELECT * FROM TaiKhoan where @tk=taikhoan and @mk = matkhau Else Print 'Wrong login'</pre>	R(A)		
COMMIT			

Tình huống 2: 2 tài xế nhận cùng một hóa đơn nhưng chỉ có tài xế nhập sau được

ERR01: Lost Update T1 (User = Tài xế): thực hiện nhận hóa đơn A T2 (User = Tài xế): thực hiện nhận hóa đơn A			
sp_TX NhanDH	Khóa	sp_TX NhanDH	Khóa
Input: @MaTX, @MaDH Output:		Input: @MaTX, @MaDH Output:	
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED		SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED	
BEGIN TRAN			
B1: T1 Nhận đơn hàng 'Chưa Giao' if (select MaTX from DonHang where MaDH = @MaDH) is NULL begin update DonHang set MaTX = @MaTX where MaDH = @MaDH exec sp_Update_TinhTrang @MaDH, 2 end	X(A)		
WAITFOR DELAY '00:00:20'			
		BEGIN TRAN	
		if (select MaTX from DonHang where MaDH = @MaDH) is NULL begin update DonHang set MaTX = @MaTX where MaDH = @MaDH	X(A)

		<pre>exec sp_Update_TinhTrang @MaDH, 2 end</pre>	
		COMMIT	
COMMIT			

II. Sinh viên thực hiện: 19120639

Tình huống 3: Khách hàng xem/đặt hàng khi đối tác cập nhật sản phẩm

ERR03: unrepeatable read T1 (User = Khách hàng): thực hiện đặt hàng(xem sản phẩm). T2 (User = Đối tác): thực hiện cập nhật sản phẩm			
sp_KH_XemSP	Khóa	sp_ sp_Insert_SP_CN	Khóa
Input: Output:		Input: @MaDT varchar(10),@MaSP varchar(10), @SLTon int Output:	
SET TRANSACTION ISOLATION LEVEL		SET TRANSACTION ISOLATION LEVEL UNREPEATABLE READ	
BEGIN TRAN			
B1: Xem thông tin sản phẩm của đối tác select sp.MaSP, TenSP, GiaBan from SanPham SP join SanPhamChiNhanh spcn on sp.masp=spcn.masp where spcn.madt=@madt	R(A) //Xin khoá đọc trên bảng SanPham và SanPhamChiNhanh		
WAITFOR DELAY '00:00:20'			
		BEGIN TRAN	
		B1: Thêm thông tin sản phẩm insert into SanPham values(@MaDT, @MaSP, @SLTon)	W(A)
		COMMIT	
COMMIT			

III. Sinh viên thực hiện: 19120626

Tình huống 4: Khách hàng thêm đơn hàng khi tài xế đang xem danh sách các đơn hàng có thể nhận

ERR04: Phantom T1 (User = Tài xế): Xem danh sách đơn hàng mình có thể nhận T2 (User = Khách hàng): Thêm một đơn hàng mới			
sp_TX_XemDH	Khóa	sp_Insert_DonHang	Khóa
Input: @MaTX nvarchar(10) Output: Danh sách đơn hàng chưa được giao		Input: @MaDH varchar(10), @HTThanhToan nvarchar(50), @DiaChiGiaoHang nvarchar(50), @MaKH varchar(10), @MaDT varchar(10) Output: Đơn hàng vừa thêm	
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED		SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED	
BEGIN TRAN			
B1: Lấy địa chỉ của tài xế declare @KhuVuc as nvarchar(50) = (select KhuVucHoatDong from TaiXe where MaTX = @MaTX)	R(TaiXe) //Xin khoá đọc trên bảng TaiXe.		
B2: Hiển thị thông tin đơn hàng theo địa chỉ của tài xế select DH.* from DonHang DH join KháchHang KH on DH.MaKH = KH.MaKH where DiaChiGiaoHang like '%' + @KhuVuc + '%'	R(DonHang) //Xin khoá đọc trên bảng DonHang.		
WAITFOR DELAY '00:00:05'			

		BEGIN TRAN	
		B1: Tìm ngày lập <code>declare @NgayLap date</code> <code>set @NgayLap = GETDATE()</code>	
		B2: Kiểm tra, thêm đơn hàng và trả về kết quả: <code>if not exists(select * from DonHang</code> <code>where MaDH = @MaDH)</code> <code>begin</code> <code>insert into</code> <code>DonHang(MaDH,HTThanhToan,DiaChiGiaoHan</code> <code>g,MaKH,TinhTrang,MaDT,NgayLap)</code> <code>values(@MaDH, @HTThanhToan,</code> <code>@DiaChiGiaoHang, @MaKH, N'Chưa Giao',</code> <code>@MaDT, @NgayLap)</code> <code>end</code> <code>select * from DonHang</code>	X(DONHANG) //Không cần xin khoá
		COMMIT	
B3: Hiển thị tổng số lượng hóa đơn được chọn <code>select count(DH.MaDH) as 'SoLuongDH'</code> <code>from DonHang DH join KhachHang KH</code> <code>on DH.MaKH = KH.MaKH</code> <code>where DiaChiGiaoHang like '%' + @KhuVuc</code> <code>+ '%'</code>	R(DonHang) //Xin khoá đọc trên bảng DonHang.		
COMMIT			

IV. Sinh viên thực hiện: 19120640

Tình huống 5: Khách hàng đặt hàng không thành công và rollback, tài xế nhận đơn hàng đã bị rollback

ERR05: Dirty read T1 (User = Khách hàng): Thực hiện đặt hàng. T2 (User = Tài Xế): Thực hiện nhận đơn hàng			
sp_DatHang Input: @MaDH varchar(10), @MaSP varchar(10), @MaDT varchar(10), @MaKH varchar(10), @DiaChi nvarchar(50), @ThanhToan int, @SoLuong int Output: Danh sách đơn hàng đã thêm	Khóa	sp_TX_NhanDH Input: @MaTX varchar(10), @MaDH varchar(10) Output: Danh sách đơn vừa nhận	Khóa
SET TRANSACTION ISOLATION LEVEL UNREPEATABLE READ		SET TRANSACTION ISOLATION LEVEL UNREPEATABLE READ	
BEGIN TRAN			
B1: Chọn hình thức thanh toán và ngày lập declare @HTThanhToan as nvarchar(50) if (@ThanhToan = 1) set @HTThanhToan = N'Chuyển Khoản' else set @HTThanhToan = N'Tiền Mặt' declare @Ngay as date = getdate()			
B2: Thêm đơn hàng insert into DonHang(MaDH, HTThanhToan, DiaChiGiaoHang, MaKH, MaDT, NgayLap, TinhTrang)	X(DonHang) //Xin khoá ghi trên bảng DonHang.		

<code>values (@MaDH,@HTThanhToan,@DiaChi,@MaKH, @MaDT,@Ngay,N'Chưa Giao')</code>			
<code>WAITFOR DELAY '00:00:10'</code>			
		<code>BEGIN TRAN</code>	
		<p>B1: Kiểm tra đơn hàng có tồn tại không:</p> <pre> if not exists (select * from DonHang where MaDH = @MaDH) begin print(N'Dơn Hàng không tồn tại!') rollback tran return 1 end </pre>	<p>R(DonHang) //Đọc không xin khóa</p>
		<p>B2: Kiểm tra đơn hàng đã được nhận chưa:</p> <pre> if exists (select * from DonHang where MaDH = @MaDH and MaTX is not null) begin print(N'Dơn Hàng đã có người nhận giao hàng!') rollback tran return 1 end </pre>	<p>R(DonHang) //Đọc không xin khóa</p>

		B2: Cập nhật thông tin đơn hàng: --Cập nhật tài xế giao đơn hàng update DonHang set MaTX = @MaTX where MaDH = @MaDH --cập nhật tình trạng đơn hàng update DonHang set TinhTrang = N'Dang Giao' where MaDH = @MaDH	X(DonHang) //Xin khóa ghi trên bản DonHang
		COMMIT	
B3: Tìm giá bán của sản phẩm declare @Gia int = (Select GiaBan from SanPham where MaSP = @MaSP)	R(SanPham) //Xin khoá đọc trên bảng SanPham.		
B4: Kiểm tra số lượng hàng đặt với số lượng tồn kho -> nếu số lượng không đủ -> ROLLBACK if(@SoLuong > (Select sum(SLTon) from SanPhamChiNhanh where MaDT = @MaDT and MaSP = @MaSP)) begin print(N'Số lượng hàng tồn kho không đủ!') rollback tran return 1 end			
B5: Thêm chi tiết đơn hàng và trả về kết quả:			

<pre>insert into CT_DonHang(MaDH, MaSP, SoLuong, GiaBan) values(@MaDH, @MaSP, @SoLuong, @Gia) select * from DonHang</pre>			
COMMIT			

Ghi chú:

Tuy còn nhiều lỗi tranh chấp nhưng hiện tại nhóm em chỉ mới tìm được 5 ví dụ trên, mong cô có thể thông cảm cho nhóm em. Nhóm em xin cảm ơn