

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	Sửa lỗi Phantom	100%
2	19120639	Lê Nam Thái Sơn	Sửa lỗi Unrepeatable-read và Deadlock	100%
3	19120640	Trần Minh Sơn	Sửa lỗi Dirty-read	100%
4	19120674	Phạm Tân Tị	Sửa lỗi Lost-Update và Unrepeatable-read	100%
5	19120640 19120674	Trần Minh Sơn Phạm Tân Tị	Viết source code các chức năng của giao diện để giao tiếp với cơ sở dữ liệu	100%
	19120626 19120639	Hồ Minh Quân Lê Nam Thái Sơn	Kiểm tra và tìm lỗi của hệ thống	100%

## XỬ LÝ TÌNH HUỐNG TRANH CHẤP

## 1. Các tình huống và xử lý

## 1.1. Tình huống 1: Đăng nhập và khóa đăng nhập

Người thực hiện: 19120674

<b>ERR01: unrepeatable read</b>  T1 (User = người dùng (đối tác, khách hàng, ...)): Thực hiện đăng nhập  T2 (User = hệ quản trị): Khóa tài khoản của user			
<b>sp_Them1ChuyenXe</b>	<b>Khóa</b>	<b>sp_ThongKeChuyenXe-CoDieuKien</b>	<b>Khóa</b>
<b><u>Input:</u></b> @tk varchar(50), @mk varchar(50) <b><u>Output:</u></b> .....		<b><u>Input:</u></b> <b><u>Output:</u></b>	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ		SET TRANSACTION ISOLATION LEVEL <b>REPEATABLE READ</b>	
<b>BEGIN TRAN</b>			
<b>B1: Kiểm tra thông tin (1) .....</b> <b>IF EXISTS (SELECT * FROM TaiKhoan WHERE</b> TaiKhoan=@tk and MatKhau = @mk)			
<b>WAITFOR DELAY '00:00:20'</b>			
		<b>BEGIN TRAN</b>	
		<b>B1: Kiểm tra thông tin (1) .....</b> declare @stat varchar(10)= (select TrangThai from taikhoan where @tk=taikhoan) if(@stat='enable') set @stat='disable' else set @stat='enable'	
		update taikhoan	

		set TrangThai=@stat where TaiKhoan=@tkWHERE	
		COMMIT	
Select * from TaiKhoan where TaiKhoan = @tk			
Else begin raiserror('Wrong Login',16,1) rollback tran			
commit			

## 1.2. Tình huống 2: Đặt hàng trong khi đối tác thêm sản phẩm

Người thực hiện: 19120639

<b>ERR02:</b> unrepeatable read T1 (User = Khách hàng): thực hiện đặt hàng. T2 (User = Đối tác): thực hiện cập nhật sản phẩm			
	<b>Khóa</b>	<b>sp_ sp_Insert_SP_CN</b>	<b>Khóa</b>
<u><b>Input:</b> .....</u> <u><b>Output:</b> .....</u>		<u><b>Input:</b> @MaDT varchar(10),@MaSP varchar(10), @SLTon int</u> <u><b>Output:</b></u>	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ		SET TRANSACTION ISOLATION LEVEL <b>REPEATEABLE READ</b>	
<b>BEGIN TRAN</b>			
<b>B1: Xem thông tin sản phẩm của đối tác</b> select sp.MaSP, TenSP, GiaBan from SanPham SP join SanPhamChiNhanh spcn on sp.masp=spcn.masp where spcn.madt=@madt	<b>R(A)</b> //Xin khoá đọc trên bảng SanPham và SanPham ChiNhanh		
<b>WAITFOR DELAY '00:00:20'</b>			

		BEGIN TRAN	
		B1: Thêm thông tin sản phẩm insert into SanPham values(@MaDT, @MaSP, @SLTon)	W(A)
		COMMIT	
COMMIT			

### 1.3. Tình huống 3: Khách hàng thêm đơn hàng khi tài xế đang xem danh sách các đơn hàng có thể nhận

Người thực hiện: 19120626

<b>ERR01: Phantom</b> 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
<u>Input:</u> @MaTX nvarchar(10) <u>Output:</u> Danh sách đơn hàng chưa được giao		<u>Input:</u> @MaDH varchar(10), @HTThanhToan nvarchar(50), @DiaChiGiaoHang nvarchar(50), @MaKH varchar(10), @MaDT varchar(10) <u>Output:</u> Đơn hàng vừa thêm	
SET TRANSACTION ISOLATION LEVEL <b>SERIALIZABLE</b>		SET TRANSACTION ISOLATION LEVEL <b>SERIALIZABLE</b>	
BEGIN TRAN			
B1: Lấy địa chỉ của tài xế declare @KhuVuc as nvarchar(50) = (select KhuVucHoatDong from TaiXe where MaTX = @MaTX)	<b>R(TaiXe)</b> //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 KhachHang KH	<b>R(DonHang)</b>		

<pre>on DH.MaKH = KH.MaKH where DiaChiGiaoHang like '%' + @KhuVuc + '%'</pre>	<pre>//Xin khoá đọc trên bảng DonHang.</pre>		
<pre>WAITFOR DELAY '00:00:05'</pre>			
		<pre>BEGIN TRAN</pre>	
		<pre>B1: Tìm ngày lập declare @NgayLap date set @NgayLap = GETDATE()</pre>	
		<pre>B2: Kiểm tra, thêm đơn hàng và trả về kết quả: if not exists(select * from DonHang where MaDH = @MaDH)  begin insert into DonHang(MaDH,HTThanhToan,DiaChiGiaoHang ,MaKH,TinhTrang,MaDT,NgayLap) values(@MaDH, @HTThanhToan, @DiaChiGiaoHang, @MaKH, N'Chưa Giao', @MaDT, @NgayLap) end  select * from DonHang</pre>	<p><b>X(DONHANG)</b> //Không cần xin khoá</p>
		<pre>COMMIT</pre>	
<pre>B3: Hiện thị tổng số lượng hóa đơn được chọn select count(DH.MaDH) as 'SoLuongDH' from DonHang DH join KhachHang KH on DH.MaKH = KH.MaKH where DiaChiGiaoHang like '%' + @KhuVuc + '%'</pre>	<p><b>R(DonHang)</b> //Xin khoá đọc trên bảng DonHang.</p>		

COMMIT

#### 1.4. Tình huống 4: Khách hàng đặt hàng không thành công và rollback, tài xế nhận đơn hàng đã bị rollback

Người thực hiện: 19120640

<b>ERR01: Dirty read</b> T1 (User = Khách hàng): Thực hiện đặt hàng. T2 (User = Tài Xế): Thực hiện nhận đơn hàng			
<b>sp_DatHang</b>	Khóa	<b>sp_TX_NhanDH</b>	Khóa
<b>Input:</b> @MaDH varchar(10), @MaSP varchar(10), @MaDT varchar(10), @MaKH varchar(10), @DiaChi nvarchar(50), @ThanhToan int, @SoLuong int  <b>Output:</b> Danh sách đơn hàng đã thêm		<b>Input:</b> @MaTX varchar(10), @MaDH varchar(10) <b>Output:</b> Danh sách đơn vừa nhận	
SET TRANSACTION ISOLATION LEVEL <b>READ COMMITED</b>		SET TRANSACTION ISOLATION LEVEL <b>READ COMMITED</b>	
<b>BEGIN TRAN</b>			
<b>B1: Chọn hình thức thanh toán và ngày lập</b> declare @HTThanhToan as nvarchar(50) if (@ThanhToan = 1) set @HTThanhToan = N'Chuyển Khoản' else set @HTThanhToan = N'Tiền Mặt' declare @Ngay as date = getdate()			

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



		<pre> rollback tran return 1 end </pre>	
		<pre> 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 </pre>	<b>X(DonHang)</b> //Xin khóa ghi trên bản DonHang
		<b>COMMIT</b>	
<b>B3: Tìm giá bán của sản phẩm</b> <pre> declare @Gia int = (Select GiaBan from SanPham where MaSP = @MaSP) </pre>	<b>R(SanPham)</b> //Xin khoá đọc trên bảng SanPham.		
<b>B4: Kiểm tra số lượng hàng đặt với số lượng tồn kho -&gt; nếu số lượng không đủ -&gt; ROLLBACK</b> <pre> if(@SoLuong &gt; (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 </pre>			
<b>B5: Thêm chi tiết đơn hàng và trả về kết quả:</b>			

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

### 1.5. Tình huống 5: 2 tài xế cùng nhận một đơn hàng

Người thực hiện: 19120674

<b>ERR01: Lost Update</b> 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
<u>Input:</u> ..... <u>Output:</u> .....		<u>Input:</u> ..... <u>Output:</u> .....	
SET TRANSACTION ISOLATION LEVEL REPEATABLE <b>READ</b>		SET TRANSACTION ISOLATION LEVEL REPEATABLE <b>READ</b>	
BEGIN TRAN			
<b>B1: T1 Nhận đơn hàng 'Chưa Giao'</b> <pre>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</pre>			
WAITFOR DELAY '00:00:20'			
		BEGIN TRAN	

		<pre> 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 </pre>	
		<b>COMMIT</b>	
<b>COMMIT</b>			

### 1.6. Tình huống 6:

#### ERR01: Dead Lock

T1 (User = Đối Tác): Sửa mật khẩu và thông tin sản phẩm

T2 (User =Nhân viên): Sửa mật khẩu và thông tin sản phẩm

	Khóa		Khóa
<b><u>Input:</u></b> .....		<b><u>Input:</u></b>	
<b><u>Output:</u></b> .....		<b><u>Output:</u></b>	
SET TRANSACTION ISOLATION LEVEL REPEATABLE <b>READ</b>		SET TRANSACTION ISOLATION LEVEL REPEATABLE <b>READ</b>	
<b>BEGIN TRAN</b>			
update TaiKhoan set MatKhu = 'password' where TaiKhoan like 'test_user'			
<b>WAITFOR DELAY '00:00:20'</b>			
		<b>BEGIN TRAN</b>	
		update SanPham set GiaBan = 15000 where TenSP like 'SP Test'	
		<b>WAITFOR DELAY '00:00:20'</b>	
update SanPham set GiaBan = 20000 where TenSP like 'SP Test'			
<b>WAITFOR DELAY '00:00:20'</b>			

		update TaiKhoan set MatKau = 'testpass' where TaiKhoan like 'test_user'	
		WAITFOR DELAY '00:00:20'	
COMMIT		COMMIT	

## 2. Cài đặt hệ thống:

Sử dụng các file sql theo thứ tự: Database.sql-> FunctionList.sql-> TriggerList.sql->ProcedureList.sql

Có thể sử dụng Data\_sample.sql để thêm vào các data cho đối tác

\*Lưu ý do generate bằng phần mềm nên sử dụng chức năng đăng nhập cho các tài khoản có sẵn có thể bị lỗi, nên đăng ký để sử dụng an toàn.

Các giao diện được liệt kê trong file báo cáo số 2.

## 3. Link Github:



#### 4. Link video demo:



Do các video được quay sau khi đã nộp báo cáo nên có thể cập nhật chậm trễ. Sẽ cập nhật trước buổi vấn đáp, mong cô thông cảm cho nhóm.