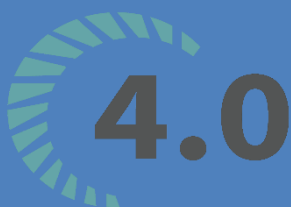


BỘ MÔN HỆ THỐNG THÔNG TIN – KHOA CÔNG NGHỆ THÔNG TIN
ĐẠI HỌC KHOA HỌC TỰ NHIÊN THÀNH PHỐ HỒ CHÍ MINH, ĐẠI HỌC QUỐC GIA TP HCM

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU



ĐỒ ÁN MÔN HỌC - HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU
2021

BẢNG THÔNG TIN CHI TIẾT NHÓM

Nhóm 2	MSSV	Họ tên
	19127376	Nguyễn Bảo Duy
	19127012	Vũ Nguyễn Thái Bình
	19127287	Nguyễn Hoài Thương

Bảng phân công công việc và đánh giá mức độ hoàn thành của từng công việc

Công việc thực hiện	Người thực hiện	Chi tiết Công việc	Mức độ hoàn thành
Thiết kế CSDL	Nguyễn Bảo Duy	Vẽ ERD	100%
		Kiểm tra video đặc tả	100%
	Vũ Nguyễn Thái Bình	Kiểm tra và hoàn chỉnh ERD	100%
		Kiểm tra video đặc tả	100%
	Nguyễn Hoài Thương	Vẽ ERD	100%
		Xuất ra cơ sở dữ liệu	100%
		Quay video đặc tả CSDL	100%
Phân quyền người dùng	Nguyễn Bảo Duy	Kiểm tra và chỉnh sửa kết quả phân quyền	100%
		Viết báo cáo phân II. Phân quyền	100%
	Vũ Nguyễn Thái Bình	Phân quyền người dùng	100%
	Nguyễn Hoài Thương	Kiểm tra và chỉnh sửa kết quả phân quyền	100%
Viết các Store procedure	Nguyễn Bảo Duy	Xem, cập nhật hợp đồng	100%
		Gia hạn hợp đồng	100%
		Xem, thêm hợp đồng	100%
		Xem, cập nhật tình trạng hợp đồng	100%
	Vũ Nguyễn Thái Bình	Tạo login	100%
		Tạo user	100%
		Cập nhật Role cho user	100%
		Xem, chọn đơn hàng trong khu vực đăng kí	100%
		Xem đối tác	100%

	Nguyễn Hoài Thương	Xem sản phẩm của đối tác	100%
		Thêm đơn hàng	100%
		Thên sản phẩm vào đơn hàng	100%
Viết script tranh chấp	Nguyễn Bảo Duy	Lost Update	100%
		Dirty Read	100%
	Vũ Nguyễn Thái Bình	Unrepeatable Read	100%
		Phantom Read	100%
	Nguyễn Hoài Thương	Conversion Deadlock	100%
		Cycle Deadlock	100%
Thiết kế layout Web (Sử dụng Node-RED)	Nguyễn Bảo Duy	Làm giao diện liên quan tranh chấp Dirty Read	100%
		Làm giao diện liên quan tranh chấp Lost Update	100%
	Vũ Nguyễn Thái Bình	Làm giao diện liên quan Conversion Deadlock	100%
		Làm giao diện liên quan Cycle Deadlock	100%
		Làm giao diện các chức năng còn lại	100%
		Làm báo cáo phần V. Web UI	100%
	Nguyễn Hoài Thương	Nhận xét layout và chỉnh sửa	100%
		Viết báo cáo phần IV. Tranh chấp	100%

Bảng đánh giá thành viên

Người thực hiện	Mức độ hoàn thành công việc được giao	Mức độ tham gia trong nhóm
Nguyễn Bảo Duy	100%	33%
Vũ Nguyễn Thái Bình	100%	34%
Nguyễn Hoài Thương	100%	33%



YÊU CẦU ĐỒ ÁN- BÀI TẬP

A. Yêu cầu của Đồ án/Bài tập

1. Thiết kế CSDL

2. Phân quyền người dùng

3. Viết các store procedure

4. Viết các tình huống tranh chấp

5. Thiết kế giao diện cho các tình huống tranh chấp



B. Kết quả

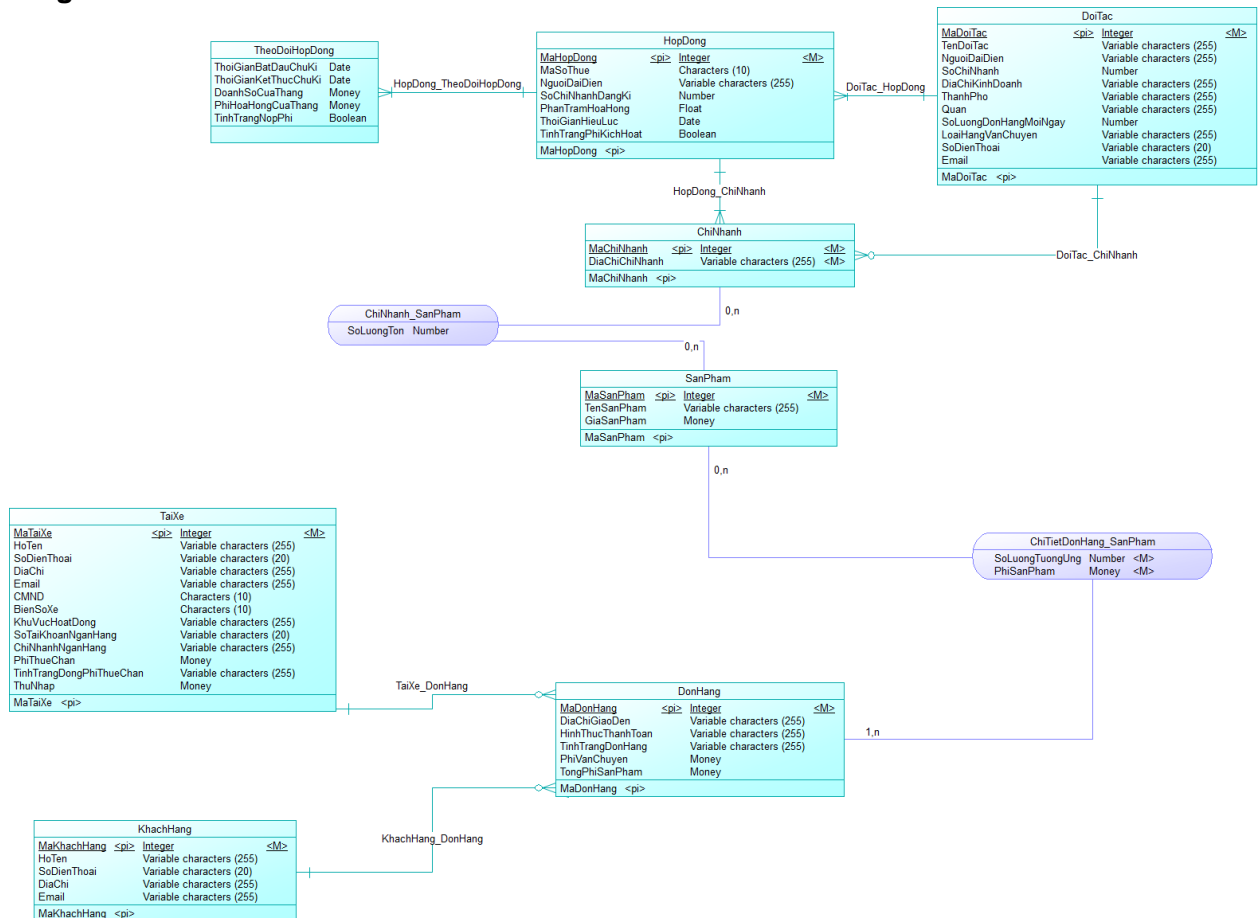
MỤC LỤC

Bảng phân công công việc và đánh giá mức độ hoàn thành của từng công việc	2
Bảng đánh giá thành viên.....	3
A. Yêu cầu của Đồ án/Bài tập	4
B. Kết quả.....	5
I. Diagram CSDL và mô tả bảng	6
II. Phân quyền CSDL.....	8
III. Store procedure	8
IV. Tranh chấp.....	13
1. Lost update	13
2. Dirty read	16
3. Unrepeatable Read	19
4. Phantom Read	22
5. Conversion Deadlock	25
6. Cycle deadlock	28
V. Web UI	31
1. Cài đặt	31
2. Khởi động node-red	32
3. Hướng dẫn sử dụng	33

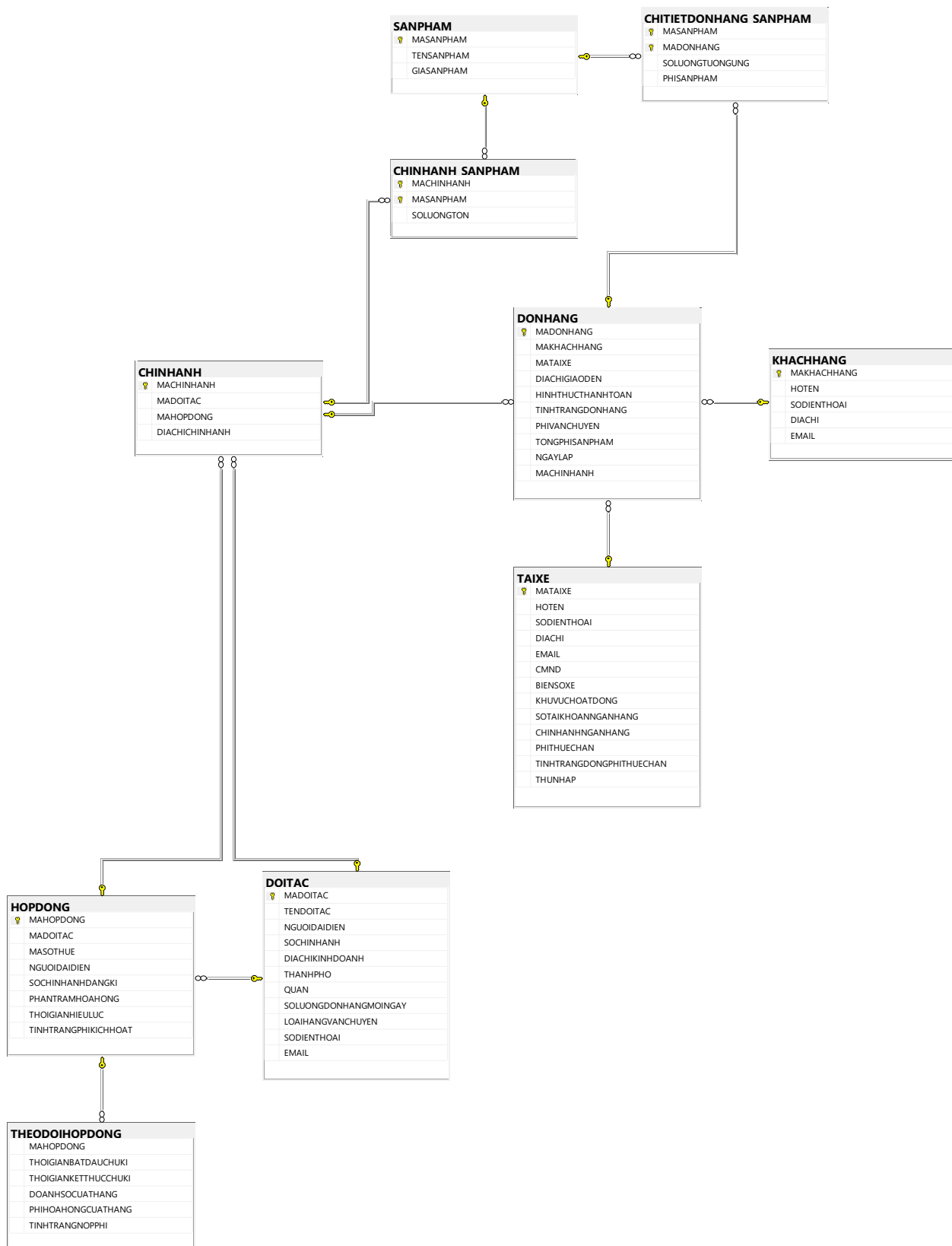


I. Diagram CSDL và mô tả bảng

a) Diagram CSDL



Mô hình dữ liệu mức quan niệm của hệ thống đặt và chuyển hàng Online.



Mô hình dữ liệu đã được cài đặt trong MS SQL Server.



b) Đặc tả CSDL

Xem nội dung đặc tả CSDL lại link: [HQTCSDL_ER_Discuss.mp4 - Google Drive](#)

II. Phân quyền CSDL

Các role mới được thêm vào bao gồm:

a) Quản trị (AdminROLE):

- Có thể xem, thêm, xóa, sửa trên các bảng **DOITAC**, **TAIXE**, **KHACHHANG**
- Có thể tạo tài khoản dựa trên thông tin đăng kí (Đối tác, Khách hàng (tài xế)) và phân quyền truy cập vào csdl

b) Nhân viên (StaffROLE):

- Có thể xem danh sách hợp đồng và update thời gian hiệu lực, tình trạng kích hoạt của hợp đồng trên bảng **HOPDONG**

c) Đối tác (PartnerROLE)

- Có thể xem và update thông tin trên bảng **DOITAC**
- Xem các đơn hàng và update tình trạng đơn hàng trên bảng **DONHANG**
- Xem, thêm, xóa, sửa và update các sản phẩm trong chi nhánh của mình trong bảng **CHINHANH_SANPHAM**
- Xem và lập thêm hợp đồng mới trên bảng **HOPDONG**
- Xem các hợp đồng mà mình đã kí trên bảng **THEODOIHOPDONG**

d) Khách hàng (CustomerROLE)

- Có thể xem và update thông tin trên bảng **KHACHHANG**
- Có thể xem danh sách đối tác, sản phẩm của đối tác và hóa đơn của khách hàng

e) Tài xế (DriverROLE)

- Có thể xem và update thông tin trên bảng **KHACHANG**
- Có thể xem danh sách đơn hàng có thể nhận và update tình trạng đơn hàng trên bảng **DONHANG**

III. Store procedure

Store procedure	Chức năng	Input	Output	Thuật toán
Nhân viên				
View_Con tract	Xem hợp đồng của đối tác	Mã đối tác	Danh sách các hợp đồng của đối tác	1. Kiểm tra mã đối tác có hợp lệ, nếu không thì roll back 2. Vào bảng hợp đồng chọn các hợp đồng với mã đối tác tương ứng

insertContract	Thêm hợp đồng mới	<ul style="list-style-type: none"> - Mã đối tác - Mã số thuế - Người đại diện - Số chi nhánh - Phần trăm hoa hồng - Thời gian hiệu lực 	Hợp đồng mới được tạo ra và chờ duyệt	<ol style="list-style-type: none"> 1. Kiểm tra tính hợp lệ của thông tin nhập vào, roll back khi không hợp lệ 2. Nếu số chi nhánh đã đăng kí + số chi nhánh định đăng kí > tổng số chi nhánh của đối tác thì rollback 3. Insert hợp đồng mới với tình trạng chờ duyệt
updateTinhTrangDonHang	Cập nhật tình trạng đơn hàng	<ul style="list-style-type: none"> - Mã đối tác - Mã đơn hàng - Tình trạng đơn hàng 	Tình trạng đơn hàng được cập nhật	<ol style="list-style-type: none"> 1. Kiểm tra tính hợp lệ của thông tin nhập vào, roll back khi không hợp lệ 2. Kiểm tra đơn hàng có nằm trong chi nhánh mà đối tác quản lý không, nếu không thì rollback 3. Cập nhật tình trạng đơn hàng 4. Nếu tình trạng đơn hàng = “Đã nhận” thì cập nhật doanh thu hàng tháng của hợp đồng tương ứng
updateDiaChi	Cập nhật địa chỉ các chi nhánh	<ul style="list-style-type: none"> - Mã đối tác - Mã chi nhánh - Địa chỉ 	Địa chỉ chi nhánh tương ứng sẽ được cập nhật	<ol style="list-style-type: none"> 1. Kiểm tra tính hợp lệ của thông tin nhập vào, roll back khi không hợp lệ 2. Kiểm tra chi nhánh có do đối tác quản lý không, nếu không thì rollback 3. Cập nhật địa chỉ chi nhánh
View_PendingContract	Xem danh sách các hợp đồng chờ duyệt	Mã đối tác	Danh sách tất cả các hợp đồng đang chờ duyệt	<ol style="list-style-type: none"> 1. Kiểm tra xem mã đối tác có hợp lệ, nếu không thì rollback 2. Nếu mã đối tác = NULL thì sẽ hiển thị tất cả các hợp đồng chờ duyệt của tất cả các đối tác 3. Nếu như mã đối tác được nhập thì sẽ hiển thị hợp đồng chờ duyệt của đối tác đó
Approve_Contract	Duyệt hợp đồng	Mã đối tác	<ul style="list-style-type: none"> - Duyệt hợp đồng - Thêm các chi nhánh của hợp đồng vào bảng CHINHANH với địa chỉ NULL - Thêm các chu kì phải đóng phí vào bảng THEODOIHOPDONG 	<ol style="list-style-type: none"> 1. Kiểm tra xem mã hợp đồng có hợp lệ, nếu không thì rollback 2. Kiểm tra nếu hợp đồng được duyệt rồi thì rollback 3. Cập nhật tình trạng đóng phí kích hoạt của hợp đồng, thêm các chi nhánh và chu kì đóng phí hàng tháng của hợp đồng



Extend_Contract	Gia hạn hợp đồng	- Mã hợp đồng - Phần trăm hoa hồng - Thời gian hiệu lực	Cập nhật thời gian hiệu lực và phần trăm hoa hồng của hợp đồng	<ol style="list-style-type: none"> 1. Kiểm tra tính hợp lệ của thông tin nhập vào, roll back khi không hợp lệ 2. Kiểm tra xem hợp đồng có còn hiệu lực đến ngày hôm nay hay không, nếu còn thì roll back 3. Cập nhật thời gian hoạt động và phần trăm hoa hồng mới của hợp đồng và đặt tình trạng đóng phí là chưa đóng
Approve_Monthly Fee	Cập nhật phí tháng của hợp đồng	- Mã hợp đồng - Ngày bắt đầu chu kì	Cập nhật tình trạng đóng phí của chu kì	<ol style="list-style-type: none"> 1. Kiểm tra tính hợp lệ của thông tin nhập vào, roll back khi không hợp lệ 2. Kiểm tra xem chu kì này có tồn tại trong hợp đồng này hay không 3. Cập nhật tình trạng đóng phí của chu kì thuộc hợp đồng này
deleteSanPham_ChiNhánh	Xóa sản phẩm khỏi 1 chi nhánh nhất định	- Mã đối tác - Mã chi nhánh - Mã sản phẩm	Sản phẩm sẽ được xóa khỏi chi nhánh	<ol style="list-style-type: none"> 1. Kiểm tra tính hợp lệ của thông tin nhập vào, roll back khi không hợp lệ 2. Xóa sản phẩm khỏi chi nhánh
updateSLSanPham_ChiNhánh	Cập nhật số lượng của sản phẩm tại 1 chi nhánh	- Mã đối tác - Mã sản phẩm - Mã chi nhánh - Số lượng cần cập nhật	Sản phẩm sẽ được cập nhật số lượng	<ol style="list-style-type: none"> 1. Kiểm tra tính hợp lệ của thông tin nhập vào, roll back khi không hợp lệ 2. Cập nhật số lượng của sản phẩm
update_GiaSanPham	Cập nhật giá của sản phẩm	- Mã đối tác - Mã sản phẩm - Giá cần cập nhật	Sản phẩm sẽ được cập nhật giá	<ol style="list-style-type: none"> 1. Kiểm tra tính hợp lệ của thông tin nhập vào, roll back khi không hợp lệ 2. Cập nhật giá của sản phẩm
Khách hàng				
View_DoiTac	Xem danh sách các đối tác		- Danh sách mã đối tác và tên đối tác.	1. Xem danh sách mã sản phẩm, tên sản phẩm, giá sản phẩm của các chi nhánh của đối tác thông qua câu lệnh select.
View_DoiTac_SanPham	Xem danh sách sản phẩm của 1 đối tác	- Mã đối tác	- Danh sách mã sản phẩm, tên sản phẩm, giá sản phẩm của các chi nhánh của đối tác.	1. Xem danh sách mã sản phẩm, tên sản phẩm, giá sản phẩm của các chi nhánh của đối tác thông qua câu lệnh select.

View_DonHang	Xem thông tin 1 đơn hàng	- Mã đơn hàng	Thông tin của đơn và danh sách sản phẩm của đơn hàng	<ol style="list-style-type: none"> Kiểm tra xem mã đơn hàng có hợp lệ, nếu không thì roll back. Trả về thông tin của đơn và danh sách sản phẩm của đơn hàng.
Insert_DonHang	Thêm một đơn hàng mới ở trạng thái "Chưa đồng ý"	- Mã khách hàng Địa chỉ giao đến Hình thức thanh toán	Thêm 1 dòng mới vào bảng "Đơn hàng"	<ol style="list-style-type: none"> Kiểm tra xem mã đơn hàng có hợp lệ, nếu không thì roll back. Tạo mã đơn hàng mới. Thêm thông tin vào bảng "Đơn hàng" Trả về mã đơn hàng.
Insert_ChiTietDonHang	Thêm 1 sản phẩm vào đơn hàng	- Mã đơn hàng - Mã sản phẩm - Mã chi nhánh - Số lượng	Thêm 1 dòng mới vào bảng "Chi tiết đơn hàng – sản phẩm" hoặc cập nhật số lượng của dòng tương ứng trong bảng "Chi tiết đơn hàng – sản phẩm"	<ol style="list-style-type: none"> Kiểm tra tính hợp lệ của thông tin nhập vào, roll back khi không hợp lệ. Kiểm tra trạng thái của đơn hàng đầu vào, nếu không phải là "Chưa đồng ý" thì rollback. So sánh số lượng muốn đặt với số lượng sản phẩm tồn của chi nhánh. Nếu vượt quá thì rollback. Tính và kiểm tra phí sản phẩm tương ứng, rollback khi không hợp lệ. Kiểm tra xem sản phẩm đã có trong chi tiết đơn hàng hay chưa, nếu có rồi thì chỉ cần cập nhật lại số lượng tương ứng. Ngược lại thì thêm 1 dòng mới vào bảng chi tiết đơn hàng.
DongY_DonHang	Chuyển trạng thái 1 đơn hàng từ "Chưa đồng ý" sang "Đồng ý"	- Mã đơn hàng	- Cập nhật tình trạng đơn hàng thành "Đồng ý"	<ol style="list-style-type: none"> Kiểm tra trạng thái của đơn hàng đầu vào, nếu không phải là "Chưa đồng ý" thì rollback. Kiểm tra danh sách chi tiết sản phẩm của đơn hàng, nếu chưa có sản phẩm nào thì rollback. Cập nhật tình trạng của đơn hàng thành "Đồng ý" Cập nhật lại danh sách số lượng tồn của chi nhánh (trừ đi cho số lượng sản phẩm đơn hàng đã đặt) Kiểm tra lại số lượng tồn sau khi cập nhật. Nếu phát hiện có chi nhánh có số lượng sản phẩm sau khi cập nhật không hợp lệ thì rollback. Nếu không có thì commit giao tác.



Tài xế

XemDonHhang	Xem danh sách các đơn hàng có thể nhận	- Mã tài xế	Danh sách đơn hàng có thể giao	<ol style="list-style-type: none"> 1. Kiểm tra xem mã tài xế có hợp lệ, nếu không thì roll back 2. Kiểm tra khu vực hoạt động của tài xế, roll back nếu là NULL 3. In ra danh sách đơn hàng có tình trạng là 'Đồng ý' và địa chỉ giao đến trong khu vực hoạt động
ChonDonHang	Xác nhận giao 1 đơn hàng	- Mã tài xế - Mã đơn hàng	- Cập nhật tình trạng = 'Đang giao' - Cập nhật tài xế giao = Mã tài xế	<ol style="list-style-type: none"> 1. Kiểm tra tính hợp lệ của thông tin nhập vào, roll back khi không hợp lệ 2. Cập nhật tình trạng và tài xế của Mã đơn hàng
TraCuuDonHangDaGiao	Xem danh sách các đơn hàng đã giao	- Mã tài xế	Danh sách đơn hàng đã giao	<ol style="list-style-type: none"> 1. Kiểm tra xem mã tài xế có hợp lệ, nếu không thì roll back 2. In ra danh sách Đơn hàng có tình trạng là 'Đã nhận' và Tài xế là Mã tài xế

Admin

newlogin_Doitac	Tạo login cho đối tác	<ul style="list-style-type: none"> - Tên đối tác - Người đại diện - Địa chỉ - Thành phố - Quận - Số chi nhánh - Số Đơn hàng mỗi ngày - Loại hàng - Số điện thoại - Email 	<ul style="list-style-type: none"> - Thêm thông tin đối tác vào bảng "Đối tác". - Tạo login, user và add role cho tài khoản đối tác. 	<ol style="list-style-type: none"> 1. Kiểm tra số điện thoại, rollback nếu sđt đã được sử dụng. 2. Thêm các thông tin đối tác vào bảng "Đối tác". 3. Tạo biến chứa script tạo login, user, add role và cho HQTCSDDL chạy.
newlogin_KhachHang	Tạo login cho khách hàng	<ul style="list-style-type: none"> - Tên khách hàng - Số điện thoại - Địa chỉ - Email 	<ul style="list-style-type: none"> - Thêm thông tin đối tác vào bảng "Khách hàng". - Tạo login, user và add role cho tài khoản khách hàng. 	<ol style="list-style-type: none"> 1. Kiểm tra số điện thoại, rollback nếu sđt đã được sử dụng. 2. Thêm các thông tin đối tác vào bảng "Khách hàng". 3. Tạo biến chứa script tạo login, user, add role và cho HQTCSDDL chạy.

newlogin_TaiXe	Tạo login cho tài xế	<ul style="list-style-type: none"> - Tên địa chỉ - Số điện thoại - Địa chỉ - Email - Chứng minh nhân dân - Biển số xe Khu vực hoạt động - Số tài khoản ngân hàng - Chi nhánh tài khoản ngân hàng 	<ul style="list-style-type: none"> - Thêm thông tin đối tác vào bảng "Tài xế". - Tạo login, user và add role cho tài khoản tài xế. 	<ul style="list-style-type: none"> I. Kiểm tra số điện thoại, rollback nếu sđt đã được sử dụng. II. Thêm các thông tin đối tác vào bảng "Tài xế". III. Tạo biến chứa script tạo login, user, add role và cho HQTCSDL chạy.
----------------	----------------------	---	--	---

IV. Tranh chấp

1. Lost update

- Tình huống 1:

- (1) Giao tác T1: Một tài xế (Tài xế TX01) chọn đi giao một đơn hàng (Đơn hàng DH01) (Cập nhật thông tin trên bảng **DONHANG**).
- (2) Giao tác T2: Cùng lúc đó, một tài xế khác (Tài xế TX02) cũng chọn đi giao đơn hàng DH01 (Cập nhật thông tin trên bảng **DONHANG**).
- Vấn đề xảy ra: Mất dữ liệu khi cập nhật.
- Hình ảnh lỗi khi chạy thử trên MS SQL SERVER:
 - o Dữ liệu ban đầu: Đơn hàng chưa có tài xế nào nhận

	MADONHANG	MATAIXE
1	1	NULL

- o Sau khi chạy đồng thời 2 giao tác:

T1

T2

Cả 2 giao tác đều được Commit. Tuy nhiên khi kiểm tra lại thì thấy bị mất đi thông tin cập nhật MaTaiXe của T1

	MADONHANG	MATAIXE
1	1	2

- Nguyên nhân:
 - Trong lúc T1 đang thực hiện, hệ thống chưa commit dữ liệu của T1 thì T2 cũng thực hiện hành động tương tự.

— Việc cập nhật dữ liệu bảng **DONHANG** của T2 sẽ ghi đè lên việc cập nhật dữ liệu của T1 dẫn đến dữ liệu bị cập nhật sai.

- Hướng giải quyết:

- Dùng cơ chế khóa trên dòng lệnh: Đặt khóa đọc-ghi Update khi đọc dữ liệu trên bảng **DONHANG**. Hành động này sẽ giúp chuyển khóa đọc trên bảng **DONHANG** thành khóa ghi.

a) Quá trình thực hiện ban đầu.

T1	T2
Begin Tran	
Read (DonHang)	
	Begin Tran
	Read (DonHang)
Update tình trạng đơn hàng của 'DH01' = Đang giao	
Update tài xế của 'DH01' = 'TX01'	
	Update tình trạng đơn hàng của 'DH01' = Đang giao
	Update tài xế của 'DH01' = 'TX02'
	Commit tran
Commit tran	

b) Quá trình thực hiện sau khi giải quyết tranh chấp.

T1	T2
Begin Tran	
UL(DonHang)	
Read (DonHang)	
	Begin Tran
	UL(DonHang) chờ
Update tình trạng đơn hàng của 'DH01' = Đang giao	
Update tài xế của 'DH01' = 'TX01'	
UNL(DonHang)	
	UL(DonHang)
	Read (DonHang)
	Báo là đơn hàng đã có người giao
	Rollback Tran
Commit tran	

- Hình ảnh kết quả thực thi chương trình sau khi giải quyết lỗi:

T1	T2
<pre>Messages (1 row affected) Completion time: 2021-12-18T21:04:02.6810983+07:00</pre>	<pre>Messages Mã đơn hàng không hợp lệ hoặc đã có người giao. Completion time: 2021-12-18T21:04:02.7346662+07:00</pre>

Tài xế 2 nhận đơn sau nên không nhận được đơn

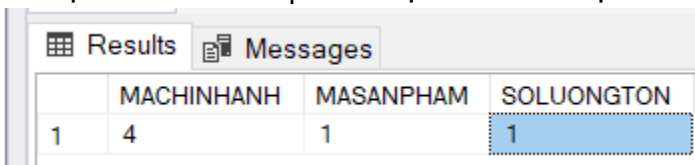
- **Tình huống 2:**

Sản phẩm có @MaSanPham tại chi nhánh có @MaChiNhanh hiện số lượng tồn chỉ còn 1, nhưng vô tình lại có 2 đơn hàng đang tiến hành thanh toán đơn đều chứa mặt hàng này.

(1) Giao tác T1: Khách hàng 1 thanh toán đơn hàng của mình

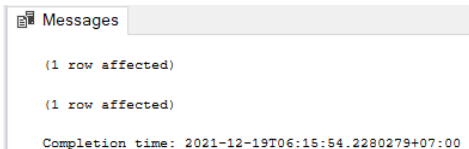
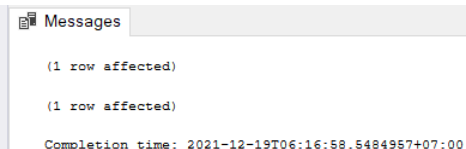
(2) Giao tác T2: Khách hàng 2 thanh toán đơn hàng của mình sau khách hàng 1

- Vấn đề xảy ra: Mất dữ liệu khi cập nhật.
- Hình ảnh lỗi khi chạy thử trên MS SQL SERVER:
 - o Dữ liệu ban đầu: Sản phẩm 1 tại chi nhánh hiện chỉ còn 1



	MACHINHANH	MASANPHAM	SOLUONGTON
1	4	1	1

- o Sau khi chạy đồng thời 2 giao tác:

T1	T2
 <p>Completion time: 2021-12-19T06:15:54.2280279+07:00</p>	 <p>Completion time: 2021-12-19T06:16:58.5484957+07:00</p>

2 Đơn hàng đều đồng ý được dù thực tế là không đủ sản phẩm.

- Nguyên nhân:
 - Trong lúc T1 đang thực hiện, hệ thống chưa commit dữ liệu của T1 thì T2 cũng thực hiện hành động tương tự.
 - Việc cập nhật dữ liệu bảng **DONHANG** của T2 sẽ ghi đè lên việc cập nhật dữ liệu của T1 dẫn đến dữ liệu bị cập nhật sai.
- Hướng giải quyết:
 - o Dùng cơ chế khóa trên dòng lệnh: Đặt khóa đọc-ghi Update khi đọc dữ liệu trên bảng **DONHANG**. Hành động này sẽ giúp chuyển khóa đọc trên bảng **DONHANG** thành khóa ghi.

a) Quá trình thực hiện ban đầu.

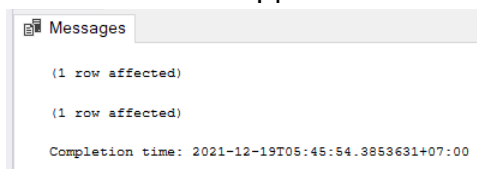
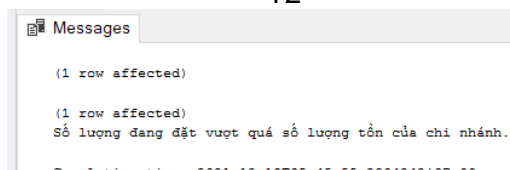
T1	T2
Begin Tran	
Update tình trạng đơn hàng của khách hàng 1 = Đồng ý	
	Begin Tran
	Update tình trạng đơn hàng của khách hàng 2 = Đồng ý
	Trừ số lượng tồn của sản phẩm trong bảng CHINHANH_SANPHAM
	Commit Tran
Trừ số lượng tồn của sản phẩm trong bảng CHINHANH_SANPHAM	
ROLL BACK TRAN (Số lượng sản phẩm = 0)	

b) Quá trình thực hiện sau khi giải quyết tranh chấp.

T1	T2
Begin Tran	

UL(DonHang)	
Update tình trạng đơn hàng của khách hàng 1 = Đồng ý	
	Begin Tran
	UL(DonHang) chờ
Trừ số lượng tồn của sản phẩm trong bảng CHINHANH_SANPHAM	
Commit Tran	
UNL(DonHang)	
	UL(DonHang)
	Update tình trạng đơn hàng của khách hàng 2 = Đồng ý
	Trừ số lượng tồn của sản phẩm trong bảng CHINHANH_SANPHAM
	ROLL BACK TRAN (Số lượng sản phẩm = 0)
Commit Tran	

- Hình ảnh kết quả thực thi chương trình sau khi giải quyết lỗi:

T1	T2
 <p>Messages</p> <p>(1 row affected)</p> <p>(1 row affected)</p> <p>Completion time: 2021-12-19T05:45:54.3853631+07:00</p>	 <p>Messages</p> <p>(1 row affected)</p> <p>(1 row affected)</p> <p>Số lượng đang đặt vượt quá số lượng tồn của chi nhánh.</p> <p>Completion time: 2021-12-19T05:45:55.2304242+07:00</p>

Khách hàng 2 thanh toán sau nên không đặt được hàng.

2. Dirty read

- Tình huống 1:

- Giao tác T1: Một khách hàng tiến hành chọn đồng ý đơn hàng của mình (Cập nhật thông tin trên bảng **CHINHANH_SANPHAM** và bảng **DONHANG**). Tuy nhiên do xảy ra lỗi nên giao tác này bị ROLLBACK.
- Giao tác T2: Cùng lúc đó, một tài xế xem thông tin trên bảng **DONHANG** (Đọc dữ liệu trên bảng **DONHANG**).

Chú thích: T2 đang có mức cô lập là READ UNCOMMITTED.

- Vấn đề xảy ra: Tài xế đọc phải dữ liệu rác.
- Hình ảnh lỗi khi chạy thử trên MS SQL SERVER:
 - Dữ liệu ban đầu: Tình trạng của đơn hàng có mã 5 là 'Chưa đồng ý'

Results		Messages	
	MADONHANG	MATAIXE	TINHTRANGDONHANG
1	5	3	Chưa đồng ý

- Sau khi chạy đồng thời 2 giao tác:



Results		Messages			
	HOTEN	DIACHIGIAODEN	HINHTHUCTHANHTOAN	PHIVANCHUYEN	TONGPHISANPHAM
1	Addie	Virginia Water	Chuyển khoản	79.28	NULL
	MADONHANG	MATAIXE	TINHTRANGDONHANG		
1	5	3	Đồng ý		

Tài xế T2 vẫn nhìn thấy đơn hàng dù khách hàng T1 chưa cập nhật giao tác.

- Nguyên nhân:
 - READ UNCOMMITTED có đặc điểm là không phát khóa khi đọc, do đó giao tác không phải chờ khi đọc dữ liệu. Vì vậy T2 có thể đọc những dữ liệu chưa được commit.
- Hướng giải quyết:
 - Thay đổi mức cô lập của T2 thành READ COMMITTED. Hành động này sẽ giúp T2 chờ cho đến khi T1 thực hiện xong thì T2 mới vào đọc giá trị.

a) Quá trình thực hiện ban đầu.

T1	T2
Begin Tran	
Read (DonHang)	
Update tình trạng đơn hàng của khách hàng = 'Đồng ý'	
	Begin Tran
	Read (DonHang)
	Commit tran
Rollback Tran	

b) Quá trình thực hiện sau khi giải quyết tranh chấp.

T1	T2
Begin Tran	
SET → READ COMMITTED	
Read (DonHang)	
Update tình trạng đơn hàng của 'DonHang' = 'Đồng ý'	
	Begin Tran
	Read (DonHang) chờ
Rollback Tran	
	Read (DHang)
	Commit tran

- Hình ảnh kết quả thực thi chương trình sau khi giải quyết lỗi

Results		Messages			
	HOTEN	DIACHIGIAODEN	HINHTHUCTHANHTOAN	PHIVANCHUYEN	TONGPHISANPHAM
	MADONHANG	MATAIXE	TINHTRANGDONHANG		
1	5	3	Chưa đồng ý		

Tài xế T2 không nhìn thấy đơn hàng mã 5 nữa vì tình trạng của nó là 'Chưa đồng ý'.

- **Tình huống 2:**

- (1) Giao tác T1: Một đối tác cập nhập lại giá của sản phẩm (update bảng SanPham). Tuy nhiên do xảy ra lỗi nên giao tác này bị ROLLBACK.
- (2) Giao tác T2: Cùng lúc đó, khách hàng xem các sản phẩm của đối tác đó (xem bảng SanPham).

Chú thích: T2 đang có mức cô lập là READ UNCOMMITTED.

- Vấn đề xảy ra: khách hàng đọc dữ liệu rác.
- Hình ảnh lỗi khi chạy thử trên MS SQL SERVER:
 - Dữ liệu ban đầu: Giá của sản phẩm có mã 1 là 30.00

	MASANPHAM	TENSANPHAM	GIASANPHAM	SOLUONGTON	MACHINHANH
1	1	Cabjectentor	30.00	30	1
2	2	Anlicscope	40.00	50	1
3	3	Armleator	50.00	30	1
4	4	Charculimator	60.00	300	2
5	5	Cartculphone	70.00	200	2
6	6	Antaar	80.00	400	2

- Sau khi chạy đồng thời 2 giao tác:

	MASANPHAM	TENSANPHAM	GIASANPHAM	SOLUONGTON	MACHINHANH
1	1	Cabjectentor	5.00	30	1
2	2	Anlicscope	40.00	50	1
3	3	Armleator	50.00	30	1
4	4	Charculimator	60.00	300	2
5	5	Cartculphone	70.00	200	2
6	6	Antaar	80.00	400	2

Khách hàng T2 nhìn thấy giá sản phẩm có mã 1 là 5.00 mặc dù giao tác T1 vẫn chưa được cập nhật.

- Nguyên nhân:
 - READ UNCOMMITTED có đặc điểm là không phát khóa khi đọc, do đó giao tác không phải chờ khi đọc dữ liệu. Vì vậy T2 có thể đọc những dữ liệu chưa được commit.
- Hướng giải quyết:
 - Thay đổi mức cô lập của T2 thành READ COMMITTED. Hành động này sẽ giúp T2 chờ cho đến khi T1 thực hiện xong thì T2 mới vào đọc giá trị.

a) Quá trình thực hiện ban đầu.

T1	T2
Begin Tran	
Update SANPHAM	
	Begin Tran
	Read (SANPHAM)

	Commit tran
Rollback Tran	

b) Quá trình thực hiện sau khi giải quyết tranh chấp.

T1	T2
Begin Tran	
SET → READ COMMITTED	
Update SANPHAM	
	Begin Tran
	Read (DHang) chờ
Rollback Tran	
	Read (DHang)
	Commit tran

- Hình ảnh kết quả thực thi chương trình sau khi giải quyết lỗi:

	MASANPHAM	TENSANPHAM	GIASANPHAM	SOLUONGTON	MACHINHANH
1	1	Cabjectentor	30.00	30	1
2	2	Anlicscope	40.00	50	1
3	3	Armleator	50.00	30	1
4	4	Charculimator	60.00	300	2
5	5	Cartculphone	70.00	200	2
6	6	Antaar	80.00	400	2

Khách hàng T2 đã đọc được giá sản phẩm đúng

3. Unrepeatable Read

- Tình huống 1:

- (1) Giao tác T1: Một quản trị viên xem thông tin trên bảng **DONHANG** (Đọc dữ liệu trên bảng **DONHANG**) và cập nhật lại PHIVANCHUYEN (Cập nhật dữ liệu bảng **DONHANG**)
- (2) Giao tác T2: Một tài xế tiến hành đếm số lượng đơn hàng đã giao (Đọc dữ liệu bảng **DONHANG**) và sau đó là tính tổng tiền (Đọc dữ liệu bảng **DONHANG**) (2 lần đọc này là trước và sau khi quản trị viên commit việc cập nhật)

Chú thích: T2 đang có mức cô lập là READ UNCOMMITTED.

- Vấn đề xảy ra: 2 lần đọc của tài xế là 2 bộ dữ liệu không đồng nhất về dữ liệu
- Hình ảnh lỗi khi chạy thử trên MS SQL SERVER:



Results		Messages								
	MADONHANG	MAKHACHHANG	MATAIXE	DIACHIGIAODEN	HINHTHUCTHANHTOAN	TINHTRANGDONHANG	PHIVANCHUYEN	TONGPHISANPHAM	NGAYLAP	MACHINHANH
1	19	1	3	NULL	NULL	Đã nhận	25000.00	NULL	NULL	NULL
2	20	1	3	NULL	NULL	Đã nhận	25000.00	NULL	NULL	NULL
3	21	1	3	NULL	NULL	Đã nhận	24000.00	NULL	NULL	NULL

	MADONHANG	MAKHACHHANG	MATAIXE	DIACHIGIAODEN	HINHTHUCTHANHTOAN	TINHTRANGDONHANG	PHIVANCHUYEN	TONGPHISANPHAM	NGAYLAP	MACHINHANH
1	19	1	3	NULL	NULL	Đã nhận	25000.00	NULL	NULL	NULL
2	20	1	3	NULL	NULL	Đã nhận	25000.00	NULL	NULL	NULL
3	21	1	3	NULL	NULL	Đã nhận	25000.00	NULL	NULL	NULL

2 lần đọc bảng DONHANG của tài xế trả về 2 bộ dữ liệu không đồng nhất với nhau

- Nguyên nhân:
 - READ UNCOMMITTED có đặc điểm là không phát khóa, do đó giao tác không phải chờ khi đọc dữ liệu. Vì vậy T2 có thể cập nhật dữ liệu dù T1 đã hoàn tất việc đọc hay chưa.
- Hướng giải quyết:
 - Thay đổi mức cô lập của T1 thành SERIALIZABLE để đảm bảo nhiều tác vụ đọc dữ liệu trong cùng một giao tác có kết quả giống nhau.. Hành động này sẽ giúp T2 chờ cho đến khi T1 thực hiện xong thì T2 mới vào cập nhật giá trị.

a) Quá trình thực hiện ban đầu.

T1	T2
Begin Tran	
Read (ChiNhanh_SanPham)	
	Begin Tran
	Update (ChiNhanh_SanPham)
Read (ChiNhanh_SanPham)	
Commit tran	
	Read (DHang)
	Commit tran

b) Quá trình thực hiện sau khi giải quyết tranh chấp.

T1	T2
Begin Tran	
Read (DHang)	
	Begin Tran
	SET → SERIALIZABLE
	Read (DHang)
Update PHIVANCHUYEN chờ	
	Read (DHang)
	Commit tran
Update PHIVANCHUYEN của 'DHang' = 26000	
Commit tran	

- Hình ảnh kết quả thực thi chương trình sau khi giải quyết lỗi

	MADONHANG	MAKHACHHANG	MATAIXE	DIACHIGIAODEN	HINHTHUCTHANHTOAN	TINHTRANGDONHANG	PHIVANCHUYEN	TONGPHISANPHAM	NGAYLAP	MACHINHANH
1	29	1	3	NULL	NULL	Đã nhận	24000.00	NULL	NULL	NULL

	MADONHANG	MAKHACHHANG	MATAIXE	DIACHIGIAODEN	HINHTHUCTHANHTOAN	TINHTRANGDONHANG	PHIVANCHUYEN	TONGPHISANPHAM	NGAYLAP	MACHINHANH
1	29	1	3	NULL	NULL	Đã nhận	24000.00	NULL	NULL	NULL

2 lần đọc bảng DONHANG của tài xế trả về 2 bộ dữ liệu đồng nhất với nhau

- **Tình huống 2:**

- (3) Giao tác T1: Một quản trị viên xem thống kê thông tin các sản phẩm của chi nhánh có mã @MaChiNhanh. Sau đó tiếp tục tra cứu tổng số sản phẩm đang cso tại chi nhánh này
- (4) Giao tác T2: Cùng lúc đó, một khách hàng đang tiến hành đồng ý đơn hàng của mình, đơn hàng này có chứa một số sản phẩm của chi nhánh có mã @MaChiNhanh

Chú thích: T1 đang có mức cô lập là READ UNCOMMITTED.

- Vấn đề xảy ra: 2 lần đọc của quản trị viên là 2 bộ dữ liệu không đồng nhất về dữ liệu
- Hình ảnh lỗi khi chạy thử trên MS SQL SERVER:

Results		Messages	
	MACHINHANH	MASANPHAM	SOLUONGTON
1	1	1	10
2	1	2	9
3	1	3	8
4	1	4	7
5	1	5	6
6	1	6	19
	Tổng số lượng sản phẩm		
1	49		

Tổng số lượng sản phẩm ở lần đọc 1 là 59, lần 2 là 49

2 lần đọc bảng DONHANG của tài xế trả về 2 bộ dữ liệu không đồng nhất với nhau

- Nguyên nhân:
 - READ UNCOMMITTED có đặc điểm là không phát khóa, do đó giao tác không phải chờ khi đọc dữ liệu. Vì vậy T2 có thể cập nhật dữ liệu dù T1 đã hoàn tất việc đọc hay chưa.
- Hướng giải quyết:
 - Thay đổi mức cô lập của T2 thành SERIALIZABLE để đảm bảo nhiều tác vụ đọc dữ liệu trong cùng một giao tác có kết quả giống nhau. Hành động này sẽ giúp T1 chờ cho đến khi T2 thực hiện xong thì T1 mới vào cập nhật giá trị.

c) Quá trình thực hiện ban đầu.

T1	T2
Begin Tran	
SET → READ UNCOMMITTED	
Read (ChiNhanh_SanPham)	
	Begin Tran
	Update (ChiNhanh_SanPham)
	Commit tran
Read (ChiNhanh_SanPham)	
Commit tran	

d) Quá trình thực hiện sau khi giải quyết tranh chấp.



T1	T2
Begin Tran	
SET → SERIALIZEABLE	
Read (ChiNhanh_SanPham)	
	Update (ChiNhanh_SanPham) CHỜ
Read (ChiNhanh_SanPham)	
Commit tran	
	Update (ChiNhanh_SanPham)
	Commit tran

- Hình ảnh kết quả thực thi chương trình sau khi giải quyết lỗi

	MACHINHANH	MASANPHAM	SOLUONGTON
1	1	1	10
2	1	2	9
3	1	3	8
4	1	4	7
5	1	5	6
6	1	6	19
Tổng số lượng sản phẩm			
1	59		

2 lần đọc bảng CHINHANH_SANPHAM của quản trị viên trả về 2 bộ dữ liệu đồng nhất với nhau

4. Phantom Read

- Tình huống 1:**

- Giao tác T1: Một quản trị viên xem thông tin trên bảng **DONHANG** (Đọc dữ liệu trên bảng **DONHANG**) và xóa hóa đơn (Xóa dữ liệu bảng **DONHANG**)
- Giao tác T2: Một tài xế tiến hành đếm số lượng đơn hàng đã giao (Đọc dữ liệu bảng **DONHANG**) và sau đó là tính tổng tiền (Đọc dữ liệu bảng **DONHANG**) (2 lần đọc này là trước và sau khi quản trị viên commit việc xóa)

Chú thích: T2 đang có mức cô lập là READ UNCOMMITTED.

- Vấn đề xảy ra:** lần đọc thứ 2 của tài xế là bộ dữ liệu “ma”
- Hình ảnh lỗi khi chạy thử trên MS SQL SERVER:**

Results Messages										
	MADONHANG	MAKHACHHANG	MATAIXE	DIACHIGIAODEN	HINHTHUCTHANHTOAN	TINHTRANGDONHANG	PHIVANCHUYEN	TONGPHISANPHAM	NGAYLAP	MACHINHANH
1	3	1	3	Ystrad Meurig	Chuyển khoản	Đã nhận	90.68	50.00	2021-11-16	1
2	4	1	3	Knaresborough	Chuyển khoản	Đã nhận	84.95	30.00	2021-10-26	2
3	5	1	3	Virginia Water	Chuyển khoản	Đã nhận	79.28	40.00	2021-10-06	2
	MADONHANG	MAKHACHHANG	MATAIXE	DIACHIGIAODEN	HINHTHUCTHANHTOAN	TINHTRANGDONHANG	PHIVANCHUYEN	TONGPHISANPHAM	NGAYLAP	MACHINHANH
1	4	1	3	Knaresborough	Chuyển khoản	Đã nhận	84.95	30.00	2021-10-26	2
2	5	1	3	Virginia Water	Chuyển khoản	Đã nhận	79.28	40.00	2021-10-06	2

Ở lần đọc 2 của tài xế, đơn hàng có mã 3 đã biến mất

- Nguyên nhân:**

- READ UNCOMMITTED có đặc điểm là không phát khóa, do đó giao tác không phải chờ khi đọc dữ liệu. Vì vậy T1 có thể xóa, cập nhật dữ liệu dù T2 đã hoàn tất việc đọc hay chưa.
- Hướng giải quyết:
 - Thay đổi mức cô lập của T2 thành SERIALIZABLE. Hành động này sẽ giúp T1 chờ cho đến khi T2 thực hiện xong thì T1 mới xóa dữ liệu.

a) Quá trình thực hiện ban đầu.

T1	T2
Begin Tran	
Read (DonHang)	
	Begin Tran
	Read (DonHang)
Delete đơn hàng trong bảng DonHang	
Commit tran	
	Read (DonHang)
	Commit tran

b) Quá trình thực hiện sau khi giải quyết tranh chấp.

T1	T2
Begin Tran	
Read (DonHang)	
	Begin Tran
	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
	Read (DonHang)
Delete đơn hàng trong bảng DonHang chờ	
	Read (DonHang)
	Commit tran
Delete đơn hàng trong bảng DonHang	
Commit tran	

- Hình ảnh kết quả thực thi chương trình sau khi giải quyết lỗi

Results Messages										
	MADONHANG	MAKHACHHANG	MATAIXE	DIACHIGIAODEN	HINHTHUCTHANHTOAN	TINHTRANGDONHANG	PHIVANCHUYEN	TONGPHISANPHAM	NGAYLAP	MACHINHANH
1	4	1	3	Knaresborough	Chuyển khoản	Đã nhận	84.95	30.00	2021-10-26	2
2	5	1	3	Virginia Water	Chuyển khoản	Đã nhận	79.28	40.00	2021-10-06	2
	MADONHANG	MAKHACHHANG	MATAIXE	DIACHIGIAODEN	HINHTHUCTHANHTOAN	TINHTRANGDONHANG	PHIVANCHUYEN	TONGPHISANPHAM	NGAYLAP	MACHINHANH
1	4	1	3	Knaresborough	Chuyển khoản	Đã nhận	84.95	30.00	2021-10-26	2
2	5	1	3	Virginia Water	Chuyển khoản	Đã nhận	79.28	40.00	2021-10-06	2

2 lần đọc bảng DONHANG của tài xế trả về 2 bộ dữ liệu đồng nhất với nhau

- **Tình huống 2:**
 - (1) Giao tác T1: Một khách hàng tính tổng phí sản phẩm trên tất cả các hóa đơn của mình (dữ liệu từ bảng **DONHANG**), sau đó đếm tất cả các đơn hàng của mình (dữ liệu từ bảng **DONHANG**) (2 lần đọc này là trước và sau khi quản trị viên commit việc xóa)
 - (2) Giao tác T2: Một quản trị viên xóa 1 đơn hàng ra khỏi bảng **DONHANG** (đơn hàng của khách hàng ở giao tác T1)
- Vấn đề xảy ra: lần đọc thứ 2 của khách hàng là bộ dữ liệu “ma”

- Hình ảnh lỗi khi chạy thử trên MS SQL SERVER:

Results Messages										
	MADONHANG	MAKHACHHANG	MATAIXE	DIACHIGIAODEN	HINHTHUCHANHTOAN	TINHTRANGDONHANG	PHIVANCHUYEN	TONGPHISANPHAM	NGAYLAP	MACHINHANH
1	1	1	1	Fairbourne	Tiền mặt	Đã nhận	89.44	30.00	2021-10-01	1
2	2	1	2	Colne	Chuyển khoản	Đã nhận	36.94	40.00	2021-10-14	1
3	3	1	3	Ystrad Meurig	Chuyển khoản	Đã nhận	90.68	50.00	2021-11-16	1
4	4	1	3	Knaresborough	Chuyển khoản	Đã nhận	84.95	30.00	2021-10-26	2
5	5	1	3	Virginia Water	Chuyển khoản	Đã nhận	79.28	40.00	2021-10-06	2
	MADONHANG	MAKHACHHANG	MATAIXE	DIACHIGIAODEN	HINHTHUCHANHTOAN	TINHTRANGDONHANG	PHIVANCHUYEN	TONGPHISANPHAM	NGAYLAP	MACHINHANH
1	2	1	2	Colne	Chuyển khoản	Đã nhận	36.94	40.00	2021-10-14	1
2	3	1	3	Ystrad Meurig	Chuyển khoản	Đã nhận	90.68	50.00	2021-11-16	1
3	4	1	3	Knaresborough	Chuyển khoản	Đã nhận	84.95	30.00	2021-10-26	2
4	5	1	3	Virginia Water	Chuyển khoản	Đã nhận	79.28	40.00	2021-10-06	2

Ở lần đọc thứ 2, đơn hàng có mã 1 đã biến mất

- Nguyên nhân:

- T1 đang tính toán dữ liệu thì T2 xóa mất dữ liệu, làm thay đổi số dòng trong câu truy vấn của T1 đang dùng
- T1 đọc lại dữ liệu bằng câu truy vấn cũ nhưng số dòng đã thay đổi dẫn đến việc tính toán sai dữ liệu

- Hướng giải quyết:

- Thay đổi mức cô lập của T1 thành **SERIALIZABLE**. Hành động này sẽ giúp T2 chờ cho đến khi T1 thực hiện xong thì T2 mới xóa dữ liệu.

a) Quá trình thực hiện ban đầu.

T1	T2
Begin Tran	
Read (DonHang)	
Tính tổng phí trên các đơn hàng của mình	
	Begin Tran
	Read (DonHang)
	Delete đơn hàng trong bảng DonHang
	Commit tran
Read (DonHang)	
Tính số đơn hàng	
Tính tiền Trung bình	
Commit tran	

b) Quá trình thực hiện sau khi giải quyết tranh chấp.

T1	T2
Begin Tran	
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
Read (DonHang)	
Tính tổng phí trên các đơn hàng của mình	
	Begin Tran
	Read (DonHang)
	Delete đơn hàng trong bảng DonHang chờ
Read (DonHang)	
Tính số đơn hàng	
Tính tiền Trung bình	
Commit tran	
	Delete đơn hàng trong bảng DonHang

Commit Tran

- Hình ảnh kết quả thực thi chương trình sau khi giải quyết lỗi

Results Messages										
	MADONHANG	MAKHACHHANG	MATAIXE	DIACHIGIAODEN	HINHTHUCTHANHTOAN	TINHTRANGDONHANG	PHIVANCHUYEN	TONGPHISANPHAM	NGAYLAP	MACHINHANH
1	1	1	1	Fairbourne	Tiền mặt	Đã nhận	89.44	30.00	2021-10-01	1
2	2	1	2	Colne	Chuyển khoản	Đã nhận	36.94	40.00	2021-10-14	1
3	3	1	3	Ystrad Meurig	Chuyển khoản	Đã nhận	90.68	50.00	2021-11-16	1
4	4	1	3	Knaresborough	Chuyển khoản	Đã nhận	84.95	30.00	2021-10-26	2
5	5	1	3	Virginia Water	Chuyển khoản	Đã nhận	79.28	40.00	2021-10-06	2

	MADONHANG	MAKHACHHANG	MATAIXE	DIACHIGIAODEN	HINHTHUCTHANHTOAN	TINHTRANGDONHANG	PHIVANCHUYEN	TONGPHISANPHAM	NGAYLAP	MACHINHANH
1	1	1	1	Fairbourne	Tiền mặt	Đã nhận	89.44	30.00	2021-10-01	1
2	2	1	2	Colne	Chuyển khoản	Đã nhận	36.94	40.00	2021-10-14	1
3	3	1	3	Ystrad Meurig	Chuyển khoản	Đã nhận	90.68	50.00	2021-11-16	1
4	4	1	3	Knaresborough	Chuyển khoản	Đã nhận	84.95	30.00	2021-10-26	2
5	5	1	3	Virginia Water	Chuyển khoản	Đã nhận	79.28	40.00	2021-10-06	2

2 lần đọc bảng DONHANG của tài xế trả về 2 bộ dữ liệu đồng nhất với nhau

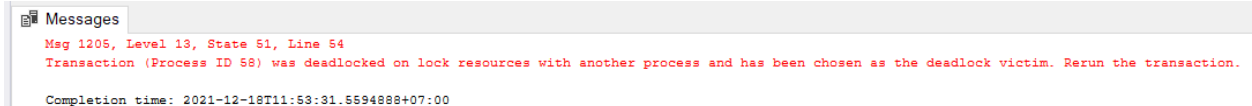
5. Conversion Deadlock

- Tình huống 1:

- (1) Giao tác T1: Một nhân viên vào xem danh sách các hợp đồng và tiến hành gia hạn hợp đồng (Cập nhật thông tin trên bảng **HOPDONG**).
- (2) Giao tác T2: Cùng lúc đó, một nhân viên khác cũng vào xem danh sách các hợp đồng và tiến hành gia hạn cùng một hợp đồng đó.

Chú thích: T1 và T2 đang có mức cô lập là Serializable.

- Vấn đề xảy ra: Hệ thống bị treo.
- Hình ảnh thông báo lỗi trả về khi chạy thử trên MS SQL SERVER:



- Nguyên nhân:

- T2 phải chờ T1 trả khóa đọc trên bảng **HOPDONG** để thực hiện việc ghi. Trong khi đó T1 cũng đang chờ T2 trả khóa đọc trên bảng **HOPDONG** để thực hiện việc ghi.
- 2 Giao tác chờ nhau nên gây ra DEADLOCK.

- Hướng giải quyết:

- Cách 1: Để hệ quản trị tự giải quyết DEADLOCK bằng cách ROLLBACK 1 giao tác. (Không thể kiểm soát được hệ thống đã hủy bỏ giao tác nào)
- Cách 2: Dùng cơ chế khóa trên dòng lệnh: Giảm mức cô lập và đặt khóa đọc-ghi Update khi đọc dữ liệu trên bảng **HOPDONG**. Hành động này sẽ giúp chuyển khóa đọc trên bảng **HOPDONG** thành khóa ghi.

a) Quá trình thực hiện ban đầu.

T1	T2
SET → SERIALIZABLE	
Select(HOPDONG)	
T1: Xin khóa S	
SQL: Cấp khóa S	
T1: Giữ khóa S đến hết giao tác, ngăn chèn dữ liệu vào tập đang khóa	

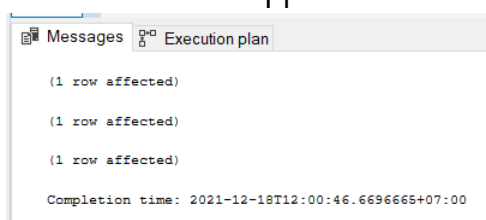
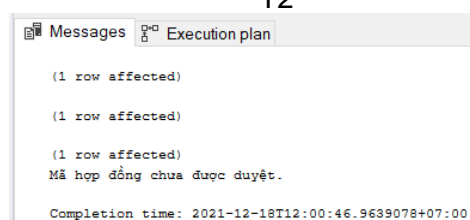


	SET → SERIALIZABLE
	Select(HOPDONG) T2: Xin khóa S SQL: Cấp khóa S T2: Giữ khóa S đến hết giao tác
	Update(HOPDONG) CHỜ T2: Xin khóa X SQL: Không cấp khóa X do T1 đang giữ khóa S
Update(HOPDONG) CHỜ T1: Xin khóa X SQL: Không cấp khóa X do T2 đang giữ khóa S	
DEADLOCK	

b) Quá trình thực hiện sau khi giải quyết tranh chấp.

T1	T2
SET → READ COMMITTED	
Select(HOPDONG) with UpDLock T1: Xin khóa đọc SQL: Cấp khóa đọc T1: Trả khóa ngay khi đọc xong	
	SET → READ COMMITTED
	Select(HOPDONG) with UpDLock T2: Xin khóa đọc SQL: Cấp khóa đọc T2: Trả khóa ngay khi đọc xong
	Update(HOPDONG) T2: Xin khóa X SQL: Cấp khóa X
	Commit
Update(HOPDONG) T2: Xin khóa X SQL: Cấp khóa X	
Commit	

- Hình ảnh kết quả thực thi chương trình sau khi giải quyết lỗi:

T1	T2
	

2 giao tác đều thực hiện thành công, không còn xảy ra tình trạng DEADLOCK



- **Tình huống 2:**

Một đối tác yêu cầu nhân viên của mình cập nhật tình trạng của đơn hàng có mã @MaDonHang thành “Đã nhận”, vì không yêu cầu cụ thể người nào nên vô tình đã có 2 nhân viên của đối tác thực hiện hành động đó cùng một lúc.

(1) Giao tác T1: Nhân viên phía đối tác tiến hành cập nhật thông tin của dòng có MaDonHang = @MaDonHang trên bảng **DONHANG**.

(2) Giao tác T2: Cùng lúc đó, một nhân viên khác cũng của phía đối tác này vào xem danh sách các hợp đồng và tiến hành hành động giống hệt T1 (cập nhật thông tin của dòng có MaDonHang = @MaDonHang trên bảng **DONHANG**).

Chú thích: T1 và T2 đang có mức cô lập là Serializable.

- Vấn đề xảy ra: Hệ thống bị treo.

- Hình ảnh thông báo lỗi trả về khi chạy thử trên MS SQL SERVER:

```
Messages
Msg 1205, Level 13, State 51, Line 44
Transaction (Process ID 54) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

Completion time: 2021-12-18T11:34:42.2729443+07:00
```

- Nguyên nhân:

- T2 phải chờ T1 trả khóa đọc trên bảng **DONHANG** để thực hiện việc ghi. Trong khi đó T1 cũng đang chờ T2 trả khóa đọc trên bảng **DONHANG** để thực hiện việc ghi.
- 2 Giao tác chờ nhau nên gây ra DEADLOCK.

- Hướng giải quyết:

- *Cách 1:* Để hệ quản trị tự giải quyết DEADLOCK bằng cách ROLLBACK 1 giao tác. (Không thể kiểm soát được hệ thống đã hủy bỏ giao tác nào)
- *Cách 2:* Dùng cơ chế khóa trên dòng lệnh: Giảm mức cô lập và đặt khóa đọc-ghi Update khi đọc dữ liệu trên bảng **DONHANG**. Hành động này sẽ giúp chuyển khóa đọc trên bảng **DONHANG** thành khóa ghi.

a) Quá trình thực hiện ban đầu.

T1	T2
SET → SERIALIZABLE	
Select(DONHANG) T1: Xin khóa S SQL: Cấp khóa S T1: Giữ khóa S đến hết giao tác, ngăn chèn dữ liệu vào tập đang khóa	
	SET → SERIALIZABLE
	Select(DONHANG) T2: Xin khóa S SQL: Cấp khóa S T2: Giữ khóa S đến hết giao tác
	Update(DONHANG) CHỜ T2: Xin khóa X SQL: Không cấp khóa X do T1 đang giữ khóa S
Update(DONHANG) CHỜ T1: Xin khóa X	

SQL: Không cấp khóa X do T2 đang giữ khóa S	
DEADLOCK	

b) Quá trình thực hiện sau khi giải quyết tranh chấp.

T1	T2
SET → READ COMMITTED	
Select(DONHANG) with UpDLock	
T1: Xin khóa đọc SQL: Cấp khóa đọc T1: Trả khóa ngay khi đọc xong	
	SET → READ COMMITTED
	Select(DONHANG) with UpDLock
	T2: Xin khóa đọc SQL: Cấp khóa đọc T2: Trả khóa ngay khi đọc xong
	Update(DONHANG)
	T2: Xin khóa X SQL: Cấp khóa X
	Commit
Update(DONHANG)	
T2: Xin khóa X SQL: Cấp khóa X	
Commit	

- Hình ảnh kết quả thực thi chương trình sau khi giải quyết lỗi:

T1	T2

2 giao tác đều thực hiện thành công, không còn xảy ra tình trạng DEADLOCK

6. Cycle deadlock

- Tình huống:

- Giao tác T1: Sau khi thêm các sản phẩm muốn mua vào đơn hàng, một khách hàng tiến hành chọn đồng ý đơn hàng của mình (Cập nhật thông tin trên bảng **CHINHANH_SANPHAM**, sau đó cập nhật thông tin trên bảng **DONHANG**).
- Giao tác T2: Cùng lúc đó, một khách hàng khác cũng thực hiện hành động tương tự (Cập nhật thông tin trên bảng **DONHANG**, sau đó cập nhật thông tin trên bảng **CHINHANH_SANPHAM**).

- Vấn đề xảy ra: Hệ thống bị treo.
- Hình ảnh lỗi khi chạy thử trên MS SQL SERVER:

- Nguyên nhân:
 - T2 phải chờ T1 trả khóa ghi trên bảng **DONHANG** để thực hiện việc ghi. Trong khi đó T1 cũng đang chờ T2 trả khóa ghi trên bảng **CHINHANH_SANPHAM** để thực hiện việc ghi.
 - 2 Giao tác chờ nhau nên gây ra DEADLOCK.
- Hướng giải quyết:
 - Cách 1: Để hệ quản trị tự giải quyết DEADLOCK bằng cách ROLLBACK 1 giao tác. (Không thể kiểm soát được hệ thống đã hủy bỏ giao tác nào)
 - Cách 2: Sử dụng logic nhất quán trong cả ứng dụng (Giao tác truy cập vào bảng **DONHANG** và **CHINHANH_SANPHAM** phải luôn truy cập vào các bảng theo cùng một thứ tự, hạn chế tối đa việc tạo thêm các giao tác thực hiện chữ năng giống nhau).

a) Quá trình thực hiện ban đầu.

T1	T2
Update(DONHANG) T1: Xin khóa X trên HOPDONG SQL: Cấp khóa X T1: Giữ khóa X đến hết giao tác	
	Update(CHINHANH_SANPHAM) T2: Xin khóa X trên CHINHANH_SANPHAM SQL: Cấp khóa X T2: Giữ khóa X đến hết giao tác
Update(CHINHANH_SANPHAM) CHỜ T1: Xin khóa X trên CHINHANH_SANPHAM SQL: Không cấp khóa X do T2 đang giữ	
	Update(DONHANG) CHỜ T2: Xin khóa X trên HOPDONG SQL: Không cấp khóa X do T1 đang giữ
DEADLOCK	

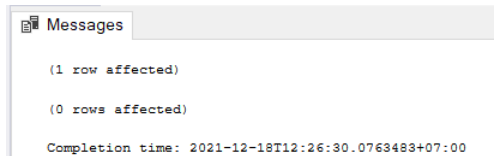
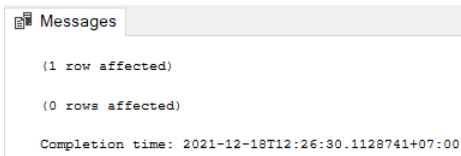
b) Quá trình thực hiện sau khi giải quyết tranh chấp.

T1	T2
Update(DONHANG) T1: Xin khóa X trên HOPDONG SQL: Cấp khóa X T1: Giữ khóa X đến hết giao tác	
	Update(DONHANG) CHỜ T2 Xin khóa X trên HOPDONG SQL: Không cấp khóa X do T1 đang giữ
Update(CHINHANH_SANPHAM) T1: Xin khóa X trên CHINHANH_SANPHAM SQL: Cấp khóa X T1: Giữ khóa X đến hết giao tác	
Commit	Update(DONHANG) T2: Xin khóa X trên HOPDONG



	SQL: Cấp khóa X
	Update(CHINHANH_SANPHAM) T2: Xin khóa X trên CHINHANH_SANPHAM SQL: Cấp khóa X
	Commit

- Hình ảnh kết quả thực thi chương trình sau khi giải quyết lỗi:

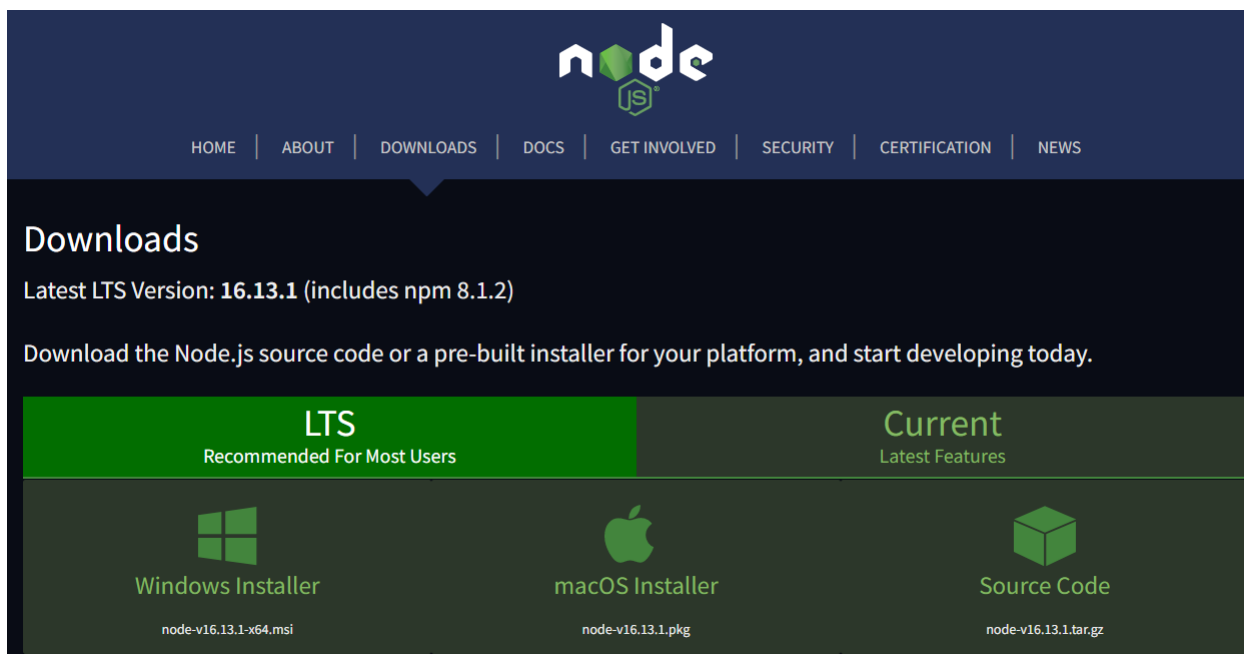
T1	T2
	

2 giao tác đều thực hiện thành công, không còn xảy ra tình trạng DEADLOCK

V. Web UI

1. Cài đặt

a) Cài node.js: (Tải và cài node.js (và npm) tùy vào loại thiết bị)



b) Cài node-red:

Installing with npm

To install Node-RED you can use the `npm` command that comes with node.js:

```
sudo npm install -g --unsafe-perm node-red
```

If you are using Windows, do not start the command with `sudo`.

Những thư viện node-red được dùng: (chạy trên cmd/terminal)

<code>npm install node-red-dashboard</code>	(UI web)	.
<code>npm install node-red-contrib-mssql-plus</code>	(SQL connect)	.
<code>npm install node-red-node-ui-table</code>	(view data as table)	.

Những thư viện node-red được dùng:



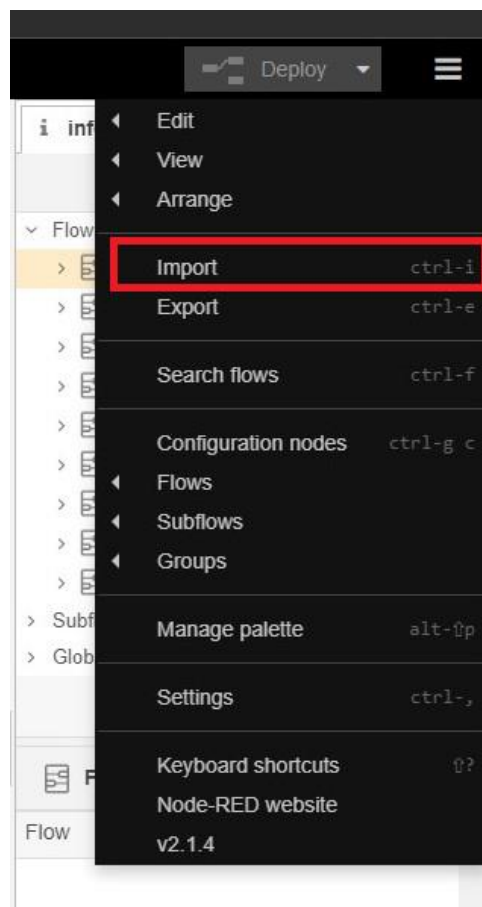
2. Khởi động node-red

```
node-red
C:\Windows\System32>node-red
17 Dec 10:21:51 - [info]

Welcome to Node-RED
=====

17 Dec 10:21:51 - [info] Node-RED version: v2.1.4
17 Dec 10:21:51 - [info] Node.js version: v14.16.1
17 Dec 10:21:51 - [info] Windows_NT 10.0.19043 x64 LE
17 Dec 10:21:51 - [info] Loading palette nodes
BAIL ON node-red-contrib-firebase/firebase_config
BAIL ON node-red-contrib-firebase/firebase_modify
BAIL ON node-red-contrib-firebase/firebase_once
BAIL ON node-red-contrib-firebase/firebase_on
BAIL ON node-red-contrib-firebase/firebase_ondisconnect
BAIL ON node-red-contrib-firebase/firebase_auth
17 Dec 10:21:52 - [info] Dashboard version 3.1.2 started at /ui
17 Dec 10:21:53 - [info] Settings file : C:\Users\admin\.node-red\settings.js
17 Dec 10:21:53 - [info] Context store : 'default' [module=memory]
17 Dec 10:21:53 - [info] User directory : \Users\admin\.node-red
17 Dec 10:21:53 - [warn] Projects disabled : editorTheme.projects.enabled=false
17 Dec 10:21:53 - [info] Flows file : \Users\admin\.node-red\flows.json
17 Dec 10:21:53 - [info] Server now running at http://127.0.0.1:1880/
```

- Node-red bây giờ sẽ chạy trên **localhost(127.0.0.1):1880**, ta truy cập nó bằng trình duyệt web
- Load file “*.json” ui được cung cấp
- Chọn **[Deploy]**
- Để truy cập vào web giao diện **127.0.0.1:1880/ui**





3. Hướng dẫn sử dụng

a) Login:

Khi mới đăng nhập, chỉ có 1 tab hiện ra

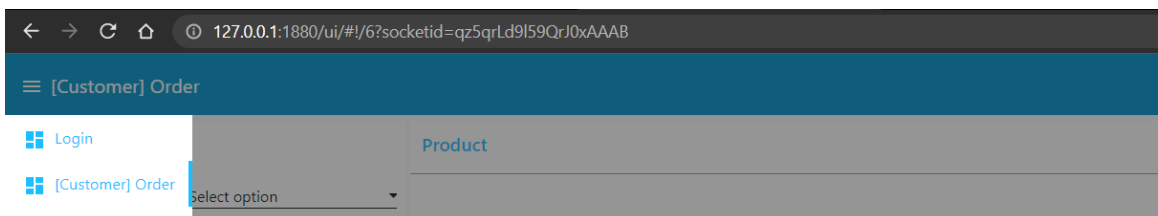
Đăng nhập tài khoản Quản trị (username: **Admin01**, password: **Admin**)

Đăng nhập tài khoản Nhân viên (username: **Staff01**, password: **Staff**)

Đăng nhập tài khoản Đối tác (username: **DTA08001204087**, password: **Partner**)*

Đăng nhập tài khoản Tài xế (username: **TXE08003740430**, password: **Driver**)*

Đăng nhập tài khoản Khách hàng (username: **KHG08089928570**, password: **Customer**)*



*Lưu ý: tên tài khoản của đối tác, khách hàng và tài xế được tạo từ SĐT đã đăng ký

b) Quản trị: Thêm thông tin đăng nhập của (Đối tác – Tài xế - Khách hàng):

Partner	Driver	Customer
Tên Đối Tác *	Tên Tài Xế *	Tên Khách Hàng *
Người Đại Diện *	SĐT *	SĐT *
Địa Chỉ	Địa chỉ	Địa chỉ
Thành Phố	Email	Email
Quận	CMND	<input type="button" value="SUBMIT"/> <input type="button" value="CANCEL"/>
Số Chi Nhánh	Số Xe Đăng Kí	
Số đơn hàng / ngày	Khu Vực Đăng Kí	
Loại Hàng	STK Ngân Hàng	
SĐT	Chi Nhánh Ngân Hàng	
Email	<input type="button" value="SUBMIT"/> <input type="button" value="CANCEL"/>	
<input type="button" value="SUBMIT"/> <input type="button" value="CANCEL"/>		

Procedure:

```
EXEC newlogin_DoiTac {{TENDOITAC}}, ..., {{EMAIL}}  
EXEC newlogin_TaiXe {{HOTEN}}, ..., {{CHINHANHNGANHANG}}  
EXEC newlogin_KhachHang {{HOTEN}}, ..., {{EMAIL}}
```

c) Nhân viên:
Duyệt hợp đồng:

[Staff] PendingContract

User

Đối Tác

North Protection Corp.

Mã Hợp Đồng

PendingContract

HD	MSTHUE	Đại Diện	Chỉ Nhanh	%	Hieu Luc
12					
13					
17	FI59217615	Antoine	15	15.62	60

DUYỆT HỢP ĐỒNG

Procedure:

EXEC View DoiTac

```
EXEC View PendingContract {{MADOITAC}}
```

```
EXEC Approve Contract {{MAHOPDONG}}
```

Gia hạn các hợp đồng đã duyệt và cập nhật thời gian đóng phí:

[illegible]

Procedure:

EXEC Extend Contract {{MAHOPDONG}} , {{PHANTRAMHOAHONG}} ,{{THOIGIANHIEULUC}}

EXEC Approve MonthlyFee {{MAHOPDONG}} , ' {{THOIGIANBATDAUCHUKI}} '

- d) Đối tác:
Xem và đăng ký hợp đồng:

[Partner] Contract								
User		Contract						
Mã Đối Tác	3	HD	MSTHUE	Dai Dien	Chi Nhanh	%	Hieu Luc	Kich Hoat
Người Đại Diện Vũ Ng. Thái Bình		5	CZ21940835	Lean	15	15.06	18	✓
		7	NL98505948	Alfreda	12	12.22	14	✓
		11						✓
		12						✗
		13						✗
Mã Số Thuế TEMPMASOTHUE08		14						✓
Số Chi Nhanh 4		15	FI59217615	Antoine	15	15.62	60	✓
		16	HU43857230	Edythe	16	16.93	32	✓
		17	FI59217615	Antoine	15	15.62	60	✗
		18	HU43857230	Edythe	16	16.93	32	✓
Hoa Hồng (%) 21.5								
Đăng kí (Tháng) 48								
ĐĂNG KÍ HỢP ĐỒNG								

Procedure:

EXEC View Contract **{{MADOITAC}}**

```
EXEC insertContract {{MADOITAC}}, '{{MASOTHUE}}', ..., {{THOIGIANHIEULUC}}
```

Xem sản phẩm của các chi nhánh (thuộc quản lý của Đối tác):

User

Mã Đối Tác

Chi Nhánh

Địa chỉ

Địa Chỉ Mới

CẬP NHẬT

Mã Sản Phẩm

Số lượng

Giá

XÓA SẢN PHẨM

Product

MASANPHAM	TENSANPHAM	GIASANPHAM	SOLUONGTON
10	Playpickentor	1393.72	459
11	Chartoplet	1854.46	1554

Procedure:

```
SELECT * FROM CHINHANH WHERE MADDOITAC = {{MADDOITAC}}
```

SELECT * FROM SANPHAM ... (Danh Sách sản phẩm của chi nhánh)

```
EXEC updateDiaChi {{MADOITAC}} , {{MACHINHANH}} , N'{{DIACHICHINHANH}}'
```

```
EXEC updateSLSanPham_ChiNhanh {{MADOITAC}}, {{MACHINHANH}}, {{MASANPHAM}},
{{SOLUONGTON}}
```

```
EXEC update GiaSanPham {{MADOITAC}}, {{MASANPHAM}}, {{GIASANPHAM}}
```

```
EXEC deleteSanPham ChiNhanh {{MADOITAC}}, {{MACHINHANH}}, {{MASANPHAM}}
```

Cập nhật tình trạng của 1000 đơn hàng gần nhất:

[Partner] Order

User

Mã Đối Tác

3

Mã Đơn Hàng

2

ĐÃ NHẬN

GIAO HÀNG

Order

MADON...	MAKHA...	MATAIXE	DIACHIG...	HINHTH...	TINHTR...	PHIVAN...	TONGPH...	NGAYLAP	MACHIN...
2	3	3	Colne	Chuyển khoản	Đang giao	36.94		2021-10-13T11...	29

Procedure:

```
SELECT top(1000) DONHANG.*
FROM DONHANG JOIN CHINHANH ON DONHANG.MACHINHANH = CHINHANH.MACHINHANH
WHERE MADOITAC = {{MADOITAC}}
ORDER BY MADONHANG DESC
EXEC updateTinhTrangDonHang {{MADOITAC}}, {{MADONHANG}}, N'{{TINHTRANGDONHANG}}'
```

e) Khách hàng:

[Customer] Order

User

Đối Tác

Global Mobile Group

Mã Sản Phẩm

5

Số lượng

12

Địa chỉ giao đến

23/45 đường 3/2

Hình thức thanh

COD

XÁC NHẬN ĐƠN HÀNG

Product

MASANPHAM	TENSANPHAM	GIASANPHAM	SOLUONGTON	MACHINHANH
4	Charculmator	182.68	798	5
5	Cartculphone	1110.03	360	5
10	Playpickentor	1393.72	459	11
11	Chartoplet	1854.46	1554	11

Cart

Mã Đối Tác

3

Chartoplet: 12

Cartculphone: 24

THANH TOÁN

HỦY

- Chọn Đối tác
- Thêm/bớt sản phẩm thuộc đối tác
Lưu ý: sau khi chọn sản phẩm không thể thay đổi đối tác (Phải xóa hết sản phẩm ra khỏi giỏ hàng)
- Thêm địa chỉ giao đến
- Chọn hình thức thanh toán
- **Xác nhận đơn hàng:** sau khi xác nhận sẽ tiến hành thanh toán
Không thể thay đổi giỏ hàng

[Customer] Order

User

Đổi Tác
Select option

Mã Sản Phẩm
11

Số lượng
12

+
-

Địa chỉ giao đến
23/45 đường 3/2

Hình thức thanh
COD

XÁC NHẬN ĐƠN HÀNG

[Customer] Order

User

Đổi Tác
Global Mobile Group

Mã Sản Phẩm
11

Số lượng
3

+
-

Địa chỉ giao đến

Hình thức thanh
Select option

XÁC NHẬN ĐƠN HÀNG

THANH TOÁN

HỦY

Mã Đổi Tác
3

- Chartoplet: 5
- Charculimator: 1

(trước và sau khi xác nhận đơn hàng)

Procedure:

EXEC View_DoiTac_SanPham {{MADOITAC}}

EXEC DongY_DonHang {{MADONHANG}}

Tác vụ thêm đơn hàng (sau xác nhận)

```

DECLARE @CART SanPham_SoLuong_Type;
INSERT INTO @CART VALUES ( {{MACHINHANH_1}}, {{MASANPHAM_1}}, {{SOLUONGTUONGUNG_1}} );
INSERT INTO @CART VALUES ( {{MACHINHANH_2}}, {{MASANPHAM_2}}, {{SOLUONGTUONGUNG_2}} );
...
INSERT INTO @CART VALUES ( {{MACHINHANH_N}}, {{MASANPHAM_N}}, {{SOLUONGTUONGUNG_N}} );
EXEC Insert_DonHang {{MAKHACHHANG}}, N'{{DIACHIGIAODEN}}', N'{{HINHTHUCTHANHTOAN}}', @CART;

```

- Tạo 1 biến tạm kiểu table lưu danh sách các sản phẩm trong giỏ hàng (lưu trên web)
- Chạy thủ tục Insert_DonHang

f) Tài xế:

Xem danh sách đơn hàng đang chờ được giao (trong khu vực đã đăng ký), và chọn đơn hàng



[Driver] Order

User

Charlena

HOTEN

SDT 08003740430

Số xe BSEAQ0M

Khu vực hoạt động

Fairbourne

Tổng thu nhập

1667

Mã Đơn Hàng

1

NHẬN ĐƠN HÀNG

ĐÃ GIAO

MADONHANG	HOTEN	DIACHIGIAODEN	HINHTHUCTHANH...	PHIVANCHUYEN	TONGPHISANPHAM
1	Addie	Fairbourne	Tiền mặt	89.44	
26	Addie	45, Đ.HUY, Fairbourne		34	

GIAO

Procedure:

EXEC XemDonHang {{MATAIXE}}

EXEC ChonDonHang {{MATAIXE}}, {{MADONHANG}}

Xem danh sách đơn hàng đã giao

[Driver] Order

User

Charlena

HOTEN

SDT 08003740430

Số xe BSEAQ0M

Khu vực hoạt động

Fairbourne

Tổng thu nhập

1667

Mã Đơn Hàng

1

NHẬN ĐƠN HÀNG

ĐÃ GIAO

MADONHANG	NGAYLAP	PHIVANCHUYEN
23		1234
24		433

GIAO

Procedure:

EXEC TraCuuDonHangDaGiao {{MATAIXE}}

Link demo: [Video](#)

Link github: https://github.com/NBDuy201/HQTCSDL_DA_TH_1