

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
KHOA CÔNG NGHỆ THÔNG TIN

---o0o---



BÁO CÁO ĐỒ ÁN CUỐI KỲ

MÔN HỌC: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

Nhóm sinh viên: Nguyễn Quang Phú – 19127507

Lâm Hoàng Phúc – 19127512

Võ Đình Phúc - 19127515

Lớp: 19HTTT1

TP. HỒ CHÍ MINH – THÁNG 12/2021

Mục lục

A. THÔNG TIN NHÓM	3
B. PHÂN CÔNG CÔNG VIỆC.....	3
C. NỘI DUNG BÁO CÁO	5
1. Mô hình quan hệ ER	5
2. Mô tả bảng – thuộc tính	5
3. Diagram cơ sở dữ liệu	7
4. Mô tả chức năng	8
5. Thiết kế giao diện	8
6. Mô tả các thủ tục	15
7. Mô tả lỗi tranh chấp, deadlock	19
8. Kết quả thực thi lỗi.....	26
9. Demo xử lý lỗi	35

A. THÔNG TIN NHÓM

Mã số sinh viên	Họ và tên	Email
19127507	Nguyễn Quang Phú	19127507@student.hcmus.edu.vn
19127512	Lâm Hoàng Phúc	19127512@student.hcmus.edu.vn
19127515	Võ Đình Phúc	19127515@student.hcmus.edu.vn

B. PHÂN CÔNG CÔNG VIỆC

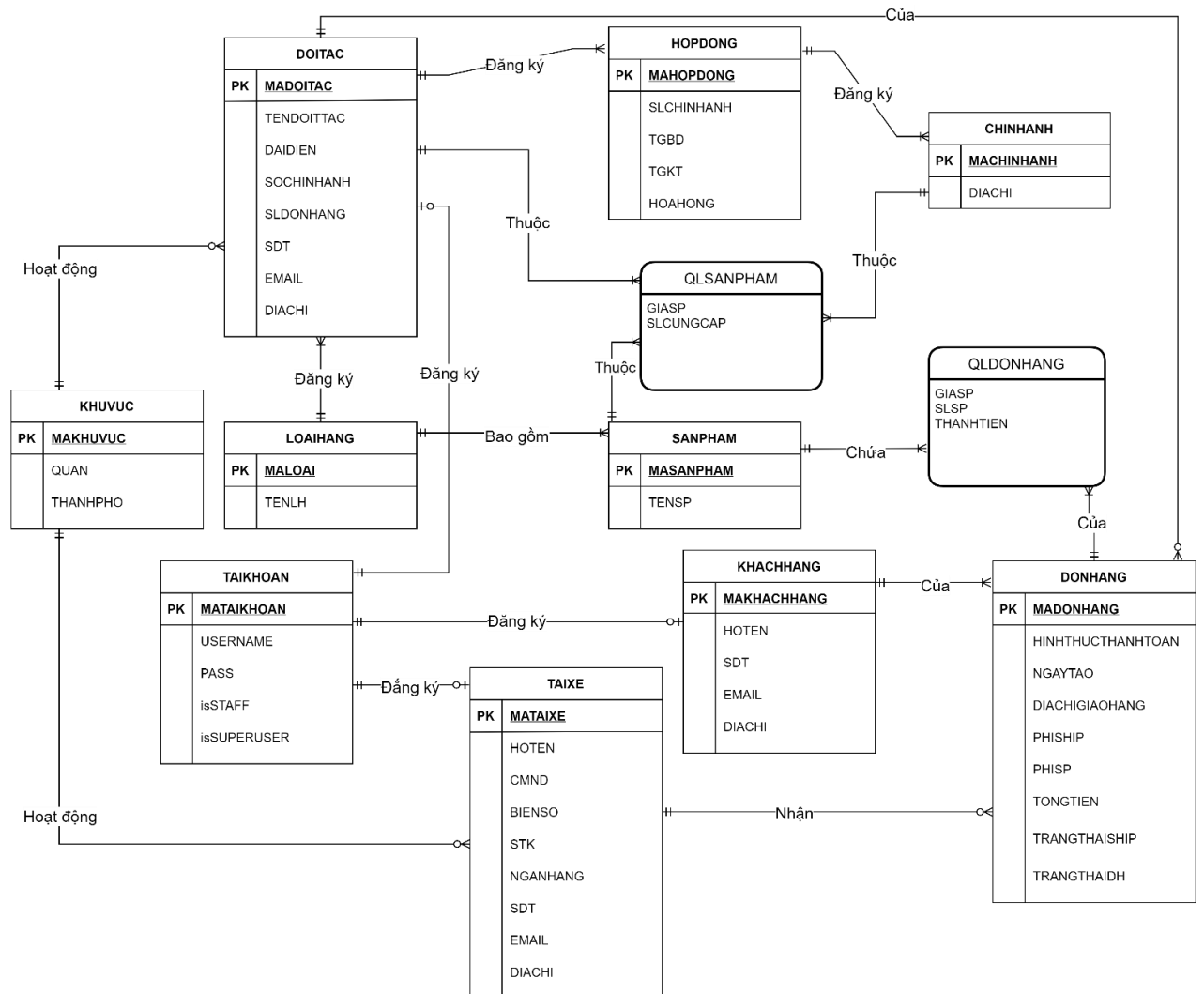
STT	Công việc	Chi tiết công việc	Phân công	Hoàn thành
1	Thiết kế cơ sở dữ liệu	Thiết kế, đặc tả và viết script các bảng đối tác, hợp đồng, chi nhánh, loại hàng.	19127515	100%
		Thiết kế, đặc tả và viết script các bảng tài xế, khu vực, tài khoản.	19127512	100%
		Thiết kế, đặc tả và viết script các bảng khách hàng, sản phẩm, đơn hàng.	19127507	100%
		Chỉnh sửa, bổ sung các thực thể, vẽ ER, hoàn thành đặc tả hệ thống và script.	Cả nhóm	100%
2	Phân quyền người dùng	Phân hệ đối tác	19127512	100%
		Phân hệ tài xế, khách hàng	19127507	100%
		Phân hệ nhân viên, quản trị	19127515	100%
3	Thiết kế giao diện	Đăng nhập, khách hàng	19127512	100%
		Tài xế, đối tác	19127507	100%
		Nhân viên, quản trị	19127515	100%

4	Tình huống tranh chấp	Mỗi thành viên phải tìm được từ 3 lỗi trở lên (script, hình ảnh minh họa)	19127512	100%
			19127507	100%
			19127515	100%
		Tổng hợp lỗi, hoàn thành script (cuộc họp qua Zoom)	Nhóm	100%
5	Hướng giải quyết	Giải quyết các lỗi mà mỗi thành viên tìm được	19127507	100%
			19127512	100%
			19127515	100%
		Hoàn thành script (cuộc họp qua Zoom)	Nhóm	100%
6	Cài đặt hệ thống (Tkinter)	Đăng nhập, khách hàng	19127512	100%
		Tài xế, đối tác	19127507	100%
		Nhân viên, quản trị	19127512	100%
4	Hoàn thành báo cáo		19127515	100%
			19127507	
5	Nộp bài		19127515	100%

MSSV	Họ và tên	Mức độ đóng góp
19127515	Võ Đình Phúc	34%
19127507	Nguyễn Quang Phú	33%
19127512	Lâm Hoàng Phúc	33%

C. NỘI DUNG BÁO CÁO

1. Mô hình quan hệ ER



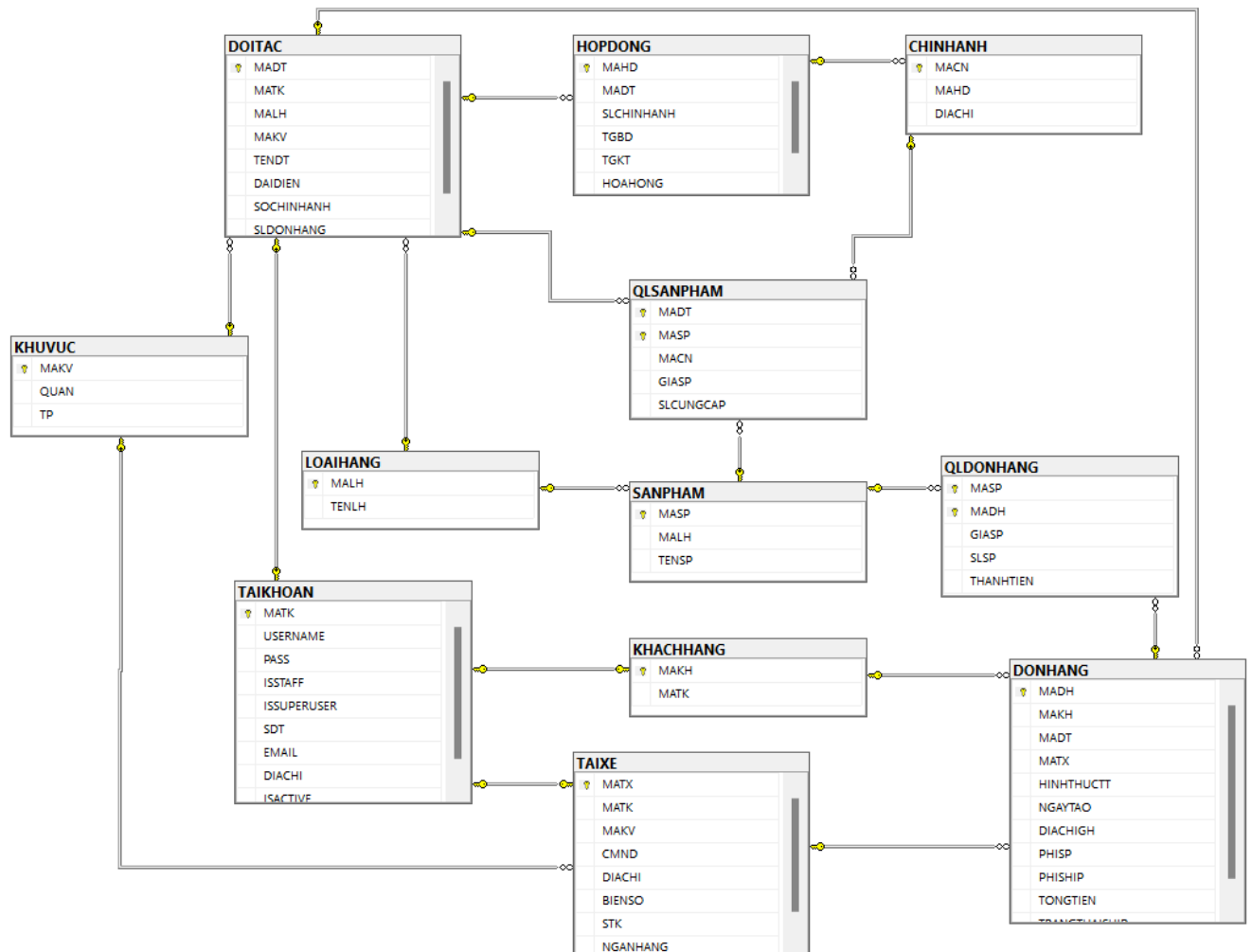
2. Mô tả bảng – thuộc tính

- Bảng DOITAC lưu thông tin các đối tác của hệ thống. Mỗi đối tác có một mã đối tác (MADOITAC) riêng biệt, tên đối tác, người đại diện, thành phố, quận, (thành phố, quận xác định bởi MAKHUVUC), số chi nhánh, số lượng đơn hàng mỗi ngày, loại hàng vận chuyển (xác định bởi MALOAIHANG), địa chỉ kinh doanh, số điện thoại, email. Mỗi đối tác sẽ có một tài khoản để truy cập vào hệ thống (xác định bởi MATAIKHOAN).
- Bảng CHINHANH lưu thông tin của các chi nhánh của mỗi đối tác của hệ thống có đăng ký trong hợp đồng. Mỗi chi nhánh có một mã chi nhánh (MACN) để phân biệt với các chi nhánh khác, mã hợp đồng (MAHD) để xác định hợp đồng nó được đăng ký, và địa chỉ của chi nhánh (DIACHI).

- Bảng KHUVUC lưu thông tin quận/huyện thuộc tỉnh/thành phố. Mỗi khu vực có một mã khu vực (MAKHUVUC) để phân biệt với các khu vực khác, một quận (QUAN) của một thành phố (THANHPHO).
- Bảng LOAIHANG lưu thông tin các loại hàng vận chuyển. Loại hàng có một mã (MALOAI) để phân biệt với các loại khác và tên loại hàng đó.
- Bảng HOPDONG lưu thông tin hợp đồng đối tác đã đăng ký thành công. Mỗi hợp đồng có một mã hợp đồng (MAHD) để phân biệt với các hợp đồng khác, mã đối tác (MADT) để xác định hợp đồng thuộc về đối tác nào. Bên cạnh đó, hợp đồng còn có số lượng chi nhánh (SLCHINHANH) được đối tác đăng ký, thời gian bắt đầu (TGBD) và kết thúc (TGKT) để xác định thời gian của hợp đồng, và tiền hoa hồng cho mỗi hợp đồng (HOAHONG), trạng thái duyệt (ISACCEPTED) để xác định hợp đồng đã duyệt hay chưa và trạng thái hạn của hợp đồng (ISUNEXPIRED) để xác định hợp đồng còn hạn hay không.
- Bảng SANPHAM lưu thông tin các sản phẩm của một loại hàng của một đối tác. Mỗi sản phẩm được xác định bởi mã sản phẩm (MASP), mỗi sản phẩm còn có mã loại hàng (MALH) và tên của sản phẩm (TENSP).
- Bảng QLSANPHAM quản lý các sản phẩm của mỗi chi nhánh thuộc từng đối tác, được xác định bởi mã đối tác (MADT) để xác định đối tác cung cấp sản phẩm và mã sản phẩm (MASP), bên cạnh đó Quản lý sản phẩm còn chứa mã chi nhánh (MACN) nó được sản xuất, giá của mỗi sản phẩm (GIASP) và số lượng sản phẩm (SLSP) được cung cấp.
- Bảng KHACHHANG lưu thông tin của khách hàng. Mỗi khách hàng có mã (MAKHACHHANG) để phân biệt với khách hàng khác, họ tên, số điện thoại, địa chỉ, email. Mỗi khách hàng sẽ có một tài khoản để truy cập vào hệ thống (xác định bởi MATAIKHOAN).
- Bảng DONHANG lưu thông tin của đơn hàng. Mỗi đơn hàng có một mã riêng biệt (MADONHANG), ngày tạo đơn hàng, địa chỉ giao hàng, tổng tiền sản phẩm của đơn hàng, phí vận chuyển, tổng tiền (tổng tiền sản phẩm + phí vận chuyển), trạng thái vận chuyển (gồm các trạng thái: chưa vận chuyển, đang vận chuyển, đã được giao), trạng thái thanh toán (đã thanh toán hoặc chưa thanh toán), hình thức thanh toán. Mỗi đơn hàng thuộc một khách hàng (MAKHACHHANG) và sẽ được một tài xế (MATAIXE) tiếp nhận vận chuyển.
- Bảng QLDONHANG quản lý các sản phẩm của đơn hàng và giá của nó. Quản lý đơn hàng được xác định bởi mã đơn hàng (MADH) và mã sản phẩm (MASP) mà nó quản lý. Quản lý sản phẩm còn quản lý giá thành sản phẩm (GIASP), số lượng của sản phẩm (SLSP) và tổng tiền (THANHTIEN) cho sản phẩm đó trên đơn hàng.

- Bảng TAIXE lưu thông tin của tài xế. Mỗi tài xế có một mã riêng biệt (MATAIXE), họ tên, cmnd, điện thoại, địa chỉ, biển số xe, khu vực hoạt động, email, thông tin tài khoản ngân hàng để nhận tiền. Tài xế sẽ đăng ký khu vực hoạt động để theo dõi và cập nhật các đơn hàng trong khu vực (xác định bởi MAKHUVUC). Mỗi tài xế sẽ có một tài khoản để truy cập vào hệ thống (xác định bởi MATAIKHOAN).
- Bảng TAIKHOAN lưu thông tin các tài khoản của hệ thống. Mỗi tài khoản có một mã tài khoản (MATAIKHOAN) để phân biệt với các tài khoản khác, tên tài khoản, mật khẩu, tài khoản này có thể là tài khoản nhân viên hoặc admin (hoặc cả 2) (nhân viên xác định bởi isSTAFF, admin xác định bởi isSUPERUSER).

3. Diagram cơ sở dữ liệu

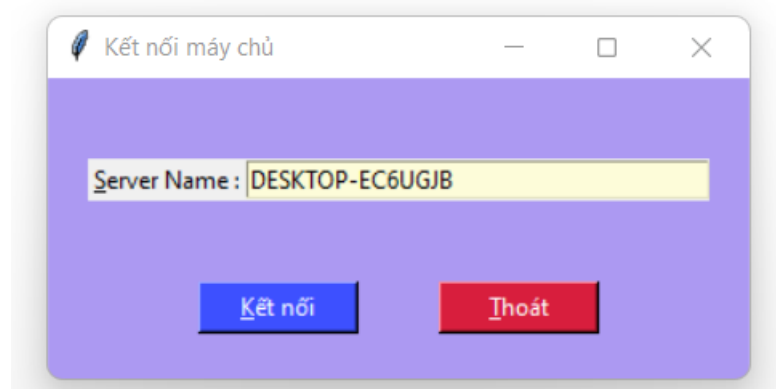


4. Mô tả chức năng

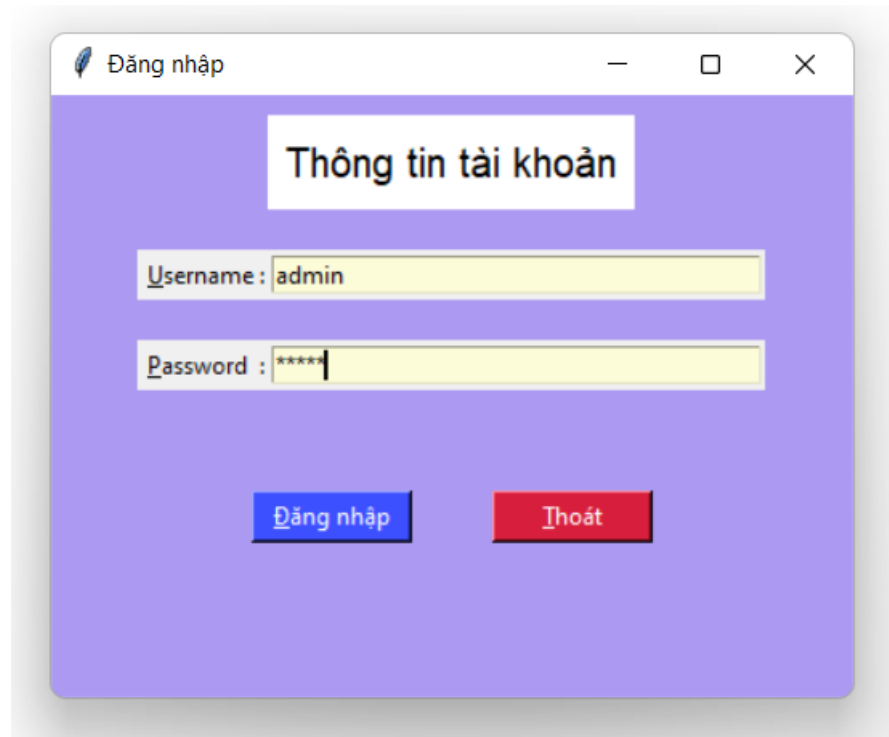
Phân hệ	Các chức năng
Đối tác	Đăng kí làm đối tác, đăng kí hợp đồng, đăng kí chi nhánh cho hợp đồng, đối tác thêm – xóa - sửa thông tin sản phẩm và chi nhánh có cung cấp sản phẩm này, đối tác xem thông tin đơn hàng và cập nhật tình trạng của đơn hàng.
Khách hàng	Đăng kí khách hàng, xem danh sách đối tác, xem danh sách sản phẩm của đối tác. Khách hàng chọn sản phẩm, số lượng tương ứng, hình thức thanh toán và địa chỉ giao hàng. Hệ thống tiến hành tính phí gồm: phí sản phẩm và phí vận chuyển. Khách hàng xác nhận đơn hàng. Khách hàng được phép xem tình trạng đơn hàng.
Tài xế	Tài có thể đăng ký tài làm tài xế của hệ thống. Sau khi đăng kí tài xế thì tài xế có thể xem danh sách đơn hàng theo khu vực mình đã chọn, nhận một đơn hàng, cập nhật trạng thái đơn hàng và xem thông tin của những đơn hàng mình đã nhận.
Nhân viên	Nhân viên được quyền xem danh sách hợp đồng chưa duyệt của đối tác, duyệt hợp đồng, sau đó thông báo thời hạn đến đối tác sau khi duyệt hợp đồng. Nhân viên xem danh sách hợp đồng của đối tác và thông báo giá hạn đến đối tác đối những hợp đồng sắp đáo hạn.
Quản trị	Quản trị viên được quyền xóa một tài khoản hoặc admin bất kì, khóa hoặc kích hoạt một tài khoản trong hệ thống. Cấp quyền dữ liệu và giao diện cho một tài khoản trong hệ thống.

5. Thiết kế giao diện

Kết nối đến máy chủ MS SQL Server

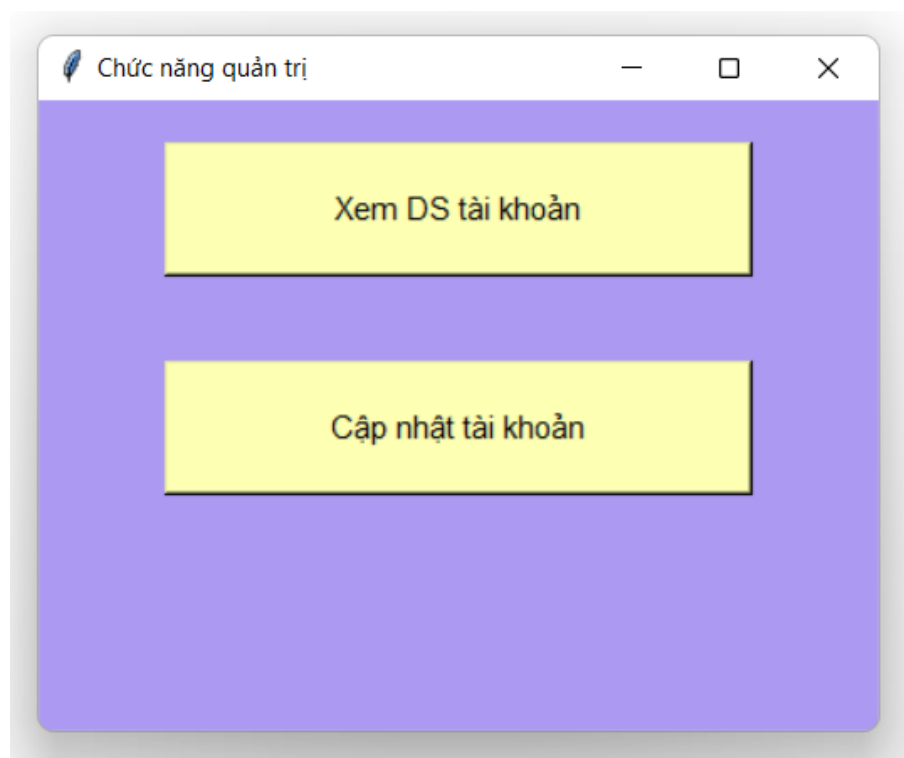


Giao diện đăng nhập



The screenshot shows a web application window titled "Đăng nhập". The main content area has a purple background. At the top, there is a white box containing the text "Thông tin tài khoản". Below this, there are two input fields: "Username:" with the value "admin" and "Password:" with the value "*****". At the bottom, there are two buttons: a blue button labeled "Đăng nhập" and a red button labeled "Thoát".

Giao diện chức năng của quản trị



The screenshot shows a web application window titled "Chức năng quản trị". The main content area has a purple background. There are two yellow buttons with black borders. The top button is labeled "Xem DS tài khoản" and the bottom button is labeled "Cập nhật tài khoản".

Cập nhật tài khoản

The screenshot shows a web application window titled "Cập nhật tài khoản". The window has a purple background and a white title bar with standard minimize, maximize, and close buttons. The form contains eight input fields, each with a label on the left and a light blue input box on the right. The labels are: "Mã tài khoản:", "Tài khoản:", "Mật khẩu:", "Là nhân viên? (0 or 1):", "Là quản trị? (0 or 1):", "Số điện thoại:", "Email:", and "Địa chỉ:". Below these fields is a yellow button with the text "Cập nhật".

Mã tài khoản:	<input type="text"/>
Tài khoản:	<input type="text"/>
Mật khẩu:	<input type="text"/>
Là nhân viên? (0 or 1):	<input type="text"/>
Là quản trị? (0 or 1):	<input type="text"/>
Số điện thoại:	<input type="text"/>
Email:	<input type="text"/>
Địa chỉ:	<input type="text"/>

Giao diện chức năng của nhân viên

The screenshot shows a web application window titled "Chức năng nhân viên". The window has a purple background and a white title bar with standard minimize, maximize, and close buttons. The interface displays four yellow buttons arranged in a 2x2 grid. The buttons are labeled: "Xem DS hợp đồng chưa duyệt", "Xem DS hợp đồng đã duyệt", "Duyệt hợp đồng", and "Thông báo gia hạn".

Xem DS hợp đồng chưa duyệt	Xem DS hợp đồng đã duyệt
Duyệt hợp đồng	Thông báo gia hạn

The screenshot shows a web application window with a title bar containing a feather icon, the text 'Danh sách hợp đồng c...', and standard window controls (minimize, maximize, close). The main content area has a purple background. It features a search bar with the placeholder text 'Điền mã đối tác:' and a yellow button labeled 'Xem' centered below it.

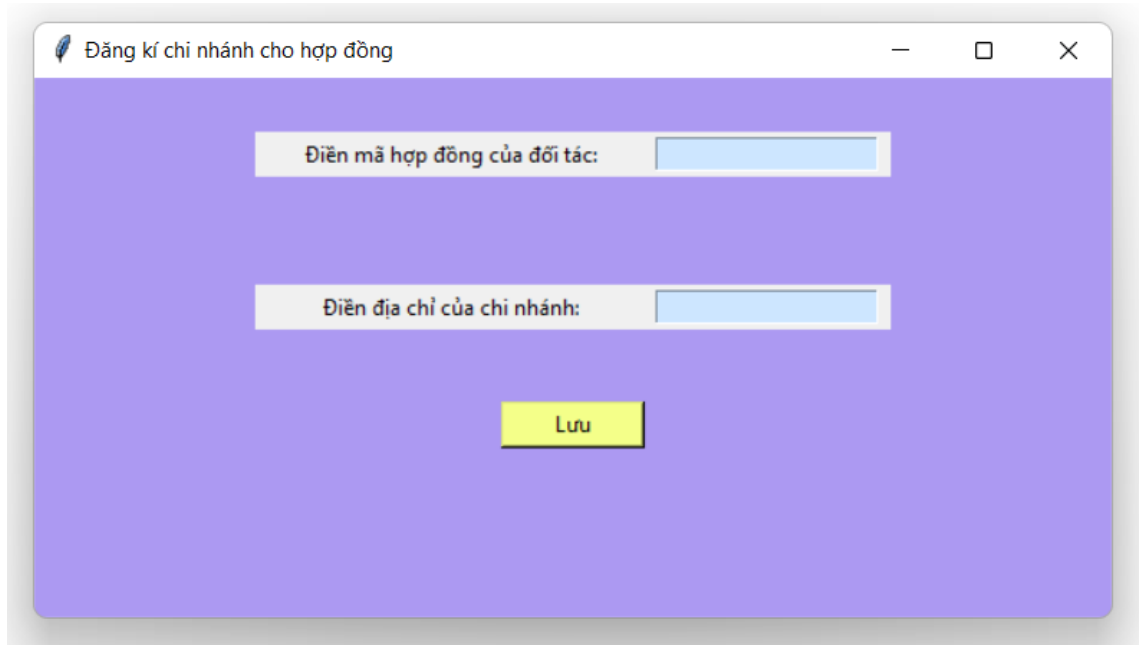
Giao diện chức năng của đối tác

The screenshot shows a web application window titled 'Chức năng đối tác' with a purple background. It contains six yellow buttons arranged in a 3x2 grid. The buttons are labeled: 'Đăng kí hợp đồng', 'Đăng kí chi nhánh cho hợp đồng', 'Xem danh sách đơn hàng', 'Gia hạn hợp đồng', 'Xem danh sách chi nhánh', and 'Xem danh sách hợp đồng'.

Đăng ký hợp đồng

The screenshot shows a web application window titled 'Đăng kí hợp đồng' with a purple background. It contains three input fields with labels: 'Thời gian bắt đầu:', 'Thời gian kết thúc:', and 'Hoa hồng:'. Below these fields is a yellow button labeled 'Đăng kí'.

Đăng ký chi nhánh cho hợp đồng



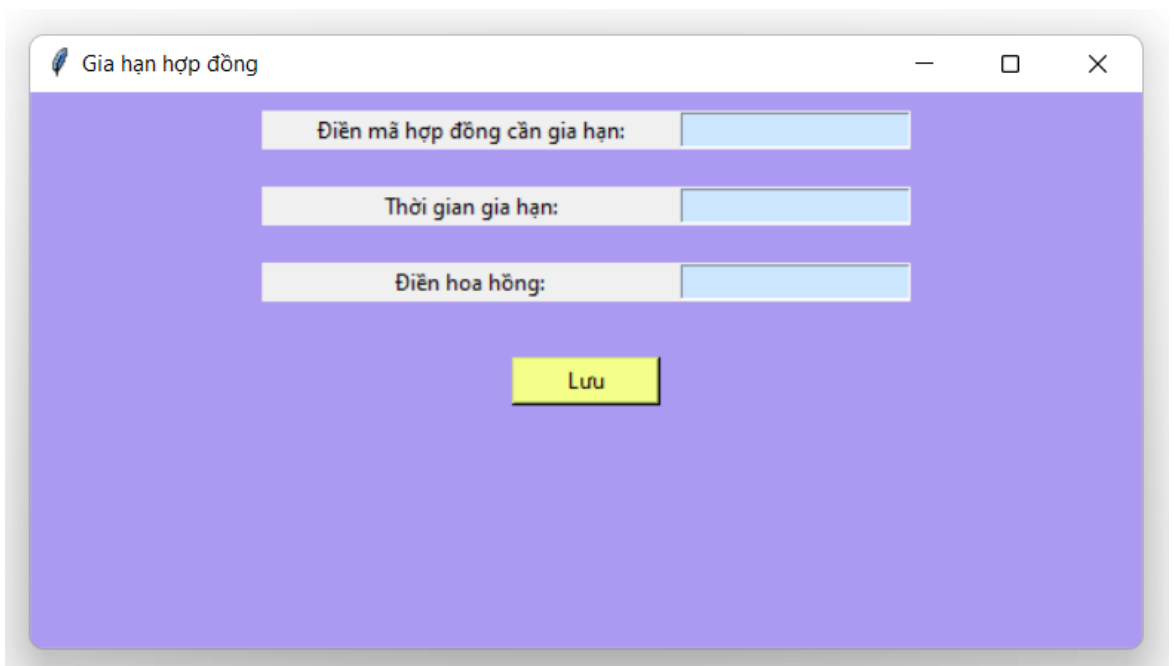
Đăng ký chi nhánh cho hợp đồng

Điền mã hợp đồng của đối tác:

Điền địa chỉ của chi nhánh:

Lưu

Gia hạn hợp đồng



Gia hạn hợp đồng

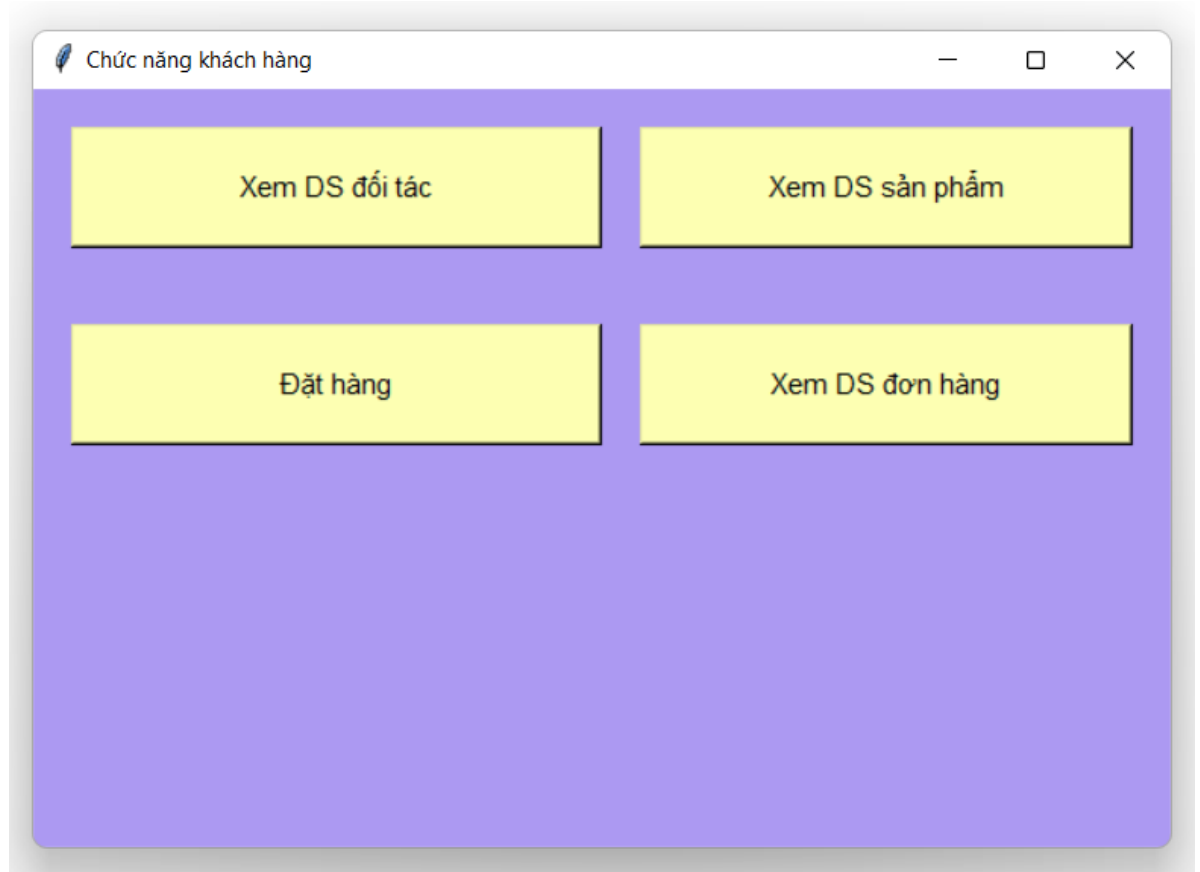
Điền mã hợp đồng cần gia hạn:

Thời gian gia hạn:

Điền hoa hồng:

Lưu

Giao diện chức năng của khách hàng

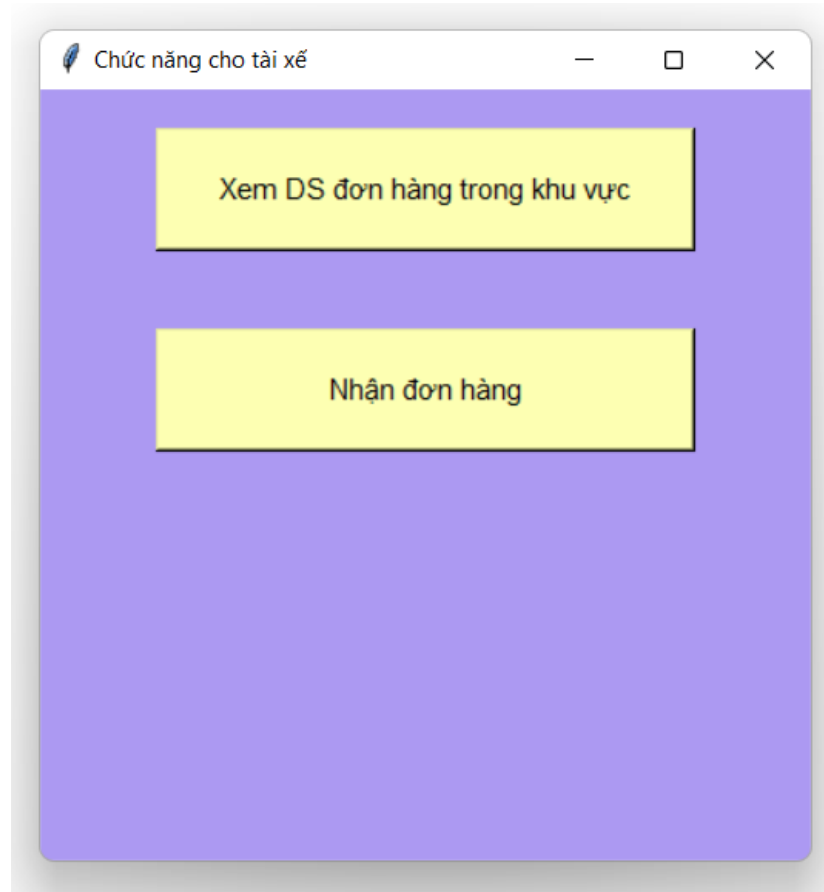


Đặt hàng

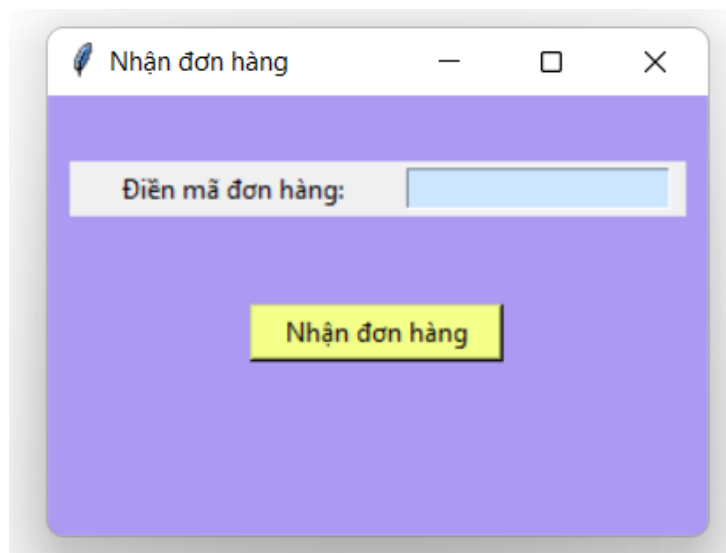
The screenshot shows a window titled "Đặt hàng" (Place Order) with a purple background. It contains five input fields and a button:

- Điền mã sản phẩm: (Enter product code)
- Điền mã đối tác: (Enter partner code)
- Số lượng: (Quantity)
- Hình thức thanh toán: (Payment method)
- Địa chỉ giao hàng: (Delivery address)
- Đặt hàng (Place Order button)

Giao diện chức năng của tài xế



Nhận đơn hàng



6. Mô tả các thủ tục

STT	Tên thủ tục	Tham số	Mô tả
Phân hệ đối tác			
1	XEM_DS_DONHANG_DT	@MADT	Xem danh sách đơn hàng tồn tại của một đối tác.
2	XEM_TT_DONHANG_DT	@MADT	Xem danh sách đơn hàng và chi tiết của đơn hàng của một đối tác.
3	XEM_DS_CHINHANH	@MASP, @MADT	Xem danh sách chi nhánh cung cấp sản phẩm đã chọn.
4	DANGKI_DT	@USERNAME_DT @PASS_DT @SDT_DT @EMAIL_DT @DIACHI_DT @MALH @MAKV @TENDT @DAIDIEN	Hệ thống sẽ tạo một tài khoản cho đối tác sau đó, insert một đối tác với thông tin do người dùng nhập vào bảng đối tác.
5	DANGKI_HOPDONG	@MADT @TGBD @TGKT @HOAHONG	Tạo một hợp đồng cho đối tác với các thông tin do người dùng nhập vào, với trạng thái kích hoạt mặc định là 0.
6	DANGKI_CHINHANH_HOPDONG	@MAHD @DIACHI	Thêm một chi nhánh vào hợp đồng với thông tin do người dùng nhập vào, sau đó thực hiện tăng số lượng chi nhánh của đối tác trong hợp đồng.
7	GIAHAN_HOPDONG	@MAHD @TGKT @HOAHONG @ISACCEPTED	Gia hạn hợp đồng với thông tin do người dùng nhập vào
8	THEM_SANPHAM	@MALH @TENSP	Thêm sản phẩm với thông tin do đối tác nhập vào
9	SUA_SANPHAM	@MASP @MALH @TENSP	Sửa sản phẩm với thông tin do đối tác nhập vào
10	XOA_SANPHAM	@MASP	Xóa sản phẩm đã chọn đồng thời xóa các QLSANPHAM của sản phẩm đó
11	THEM_CHINHANH_SP	@MADT @MASP @MACN @GIASP @SLCUNGCAP	Tạo QLSANPHAM để thêm các thông tin về chi nhánh và đối tác cung cấp theo thông tin do người dùng nhập

12	SUA_CHINHAN H_SP	@MADT @MASP @MACN @GIASP @SLCUNGCAP	Sửa QLSANPHAM theo thông tin mà người dùng đã nhập
13	XOA_CHINHAN H_SP	@MADT @MASP	Xóa QLSANPHAM của đối tác đã chọn với một sản phẩm đã chọn
14	CAPNHAT_DON HANG_DT	@MADH @TRANGTHAISHIP @TRANGTHAITTOAN	Cập nhật trạng thái của đơn hàng theo người dùng nhập

Phân hệ khách hàng

1	DANGKI_KH	@USERNAME_KH @PASS_KH @SDT_KH @EMAIL_KH @DIACHI_KH	Đăng ký tài khoản cho khách hàng trong bảng tài khoản bằng proc TAO_TAIKHOAN, sau đó hệ thống thực hiện thêm một khách hàng vào bảng khách hàng.
2	XEM_DS_DT		Cho phép khách hàng xem danh sách đối tác của hệ thống.
3	XEM_DS_SP	@MADT	Sau khi khách hàng đã chọn được đối tác, khách hàng có thể xem danh sách sản phẩm được cung cấp bởi đối tác này.
4	TAO_DONHANG	@MAKH @MADT @MATX @HINHTHUCTT @NGAYTAO @DIACHIGH @PHISP @PHISHIP @TONGTIEN @TRANGTHAISHIP @TRANGTHAITTOAN @MADH OUTPUT	Chức năng tạo một đơn hàng, khi khách hàng tiến hành chọn sản phẩm và những thông tin cần thiết và hệ thống sẽ tạo một đơn hàng cho khách hàng.
5	THEMSANPHAM	@MADH @MASP @GIASP @SLSP @MADT	Đối với mỗi loại sản phẩm được khách hàng thêm vào đơn hàng của mình thì hệ thống sẽ tạo ra một quản lý đơn hàng để quản lý sản phẩm này trong đơn hàng của khách hàng (Giả sử một khách hàng được phép đặt nhiều sản phẩm của một đối tác). Đồng thời trừ số sản phẩm đặt vào số lượng cung cấp sản phẩm này của đối tác.
6	DATHANG	@MAKH @MASP @MADT @SLSP @HINHTHUCTT @DIACHIGH	Khách hàng sau khi chọn được sản phẩm và số lượng tương ứng cũng như một số thông tin cần thiết thì hệ thống thực hiện tạo một đơn hàng cho khách hàng.

		@PHISHIP	Kiểm tra nếu số lượng sản phẩm cung cấp nhỏ hơn số lượng sản phẩm đặt thì cho phép khách hàng thêm sản phẩm. Thêm sản phẩm này vào đơn hàng cho khách hàng.
7	XACNHAN_DONHANG	@MADH	Khi khách hàng xác nhận đơn hàng thì hệ thống tiến hành set lại trạng thái cho đơn hàng là đã thanh toán.
8	XEM_TTHAI_DONHANG	@MADH	Khách hàng được phép xem trạng thái ship của đơn hàng sau khi xác nhận đơn hàng.

Phân hệ tài xế

1	DANGKI_TAIXE	@USERNAME_TX @PASS_TX @SDT_TX @EMAIL_TX @DIACHI_TX @MAKV @CMND @DIACHI @BIENSO @STK @NGANHANG	Đầu tiên hệ thống tiến hành tạo một tài khoản cho tài xế, sau đó thêm một đối tượng tài xế vào bảng tài xế với những thông tin được nhập vào.
2	HIENTHI_DSDH	@MAKV	Cho phép tài xế xem danh sách đơn hàng của những đối tác trong khu vực mình đã đăng kí. Và những đơn hàng này phải thỏa điều kiện là đã được thanh toán và mã tài xế là null.
3	NHAN_DONHANG	@MADH @MATX	Khi tài xế nhận đơn hàng thì hệ thống sẽ gắn mã tài xế này vào đơn hàng.
4	CAPNHAT_DONHANG_TX	@MADH @TRANGTHAISHIP	Tài xế được phép cập nhật lại trạng thái ship của đơn hàng, khi giao hàng.
5	XEM_DS_DONHANG_DANHAN	@MATX	Tài xế được phép xem thông tin của những đơn hàng mình đã nhận.

Phân hệ nhân viên và quản trị viên

1	XEM_DSHD_DADUYET	@MADT	Xem danh sách những hợp đồng của đối tác đã được xét duyệt
2	XEM_DSHD_CHUADUYET	@MADT	Xem danh sách những hợp đồng đối tác đã đăng kí nhưng chưa thực hiện xét duyệt
3	THONGBAO_GIAHAN	@MADT	Thực hiện tìm ra danh sách những hợp đồng sắp đáo hạn của đối tác thông qua mã đối tác. Thời gian thông mặc định là 10 ngày trước khi đáo hạn.

4	DUYET_HOPDO NG	@MADT	Nhân viên thực hiện tìm ra danh sách những hợp đồng đã lập của đối tác những chưa thông qua xét duyệt, sau đó duyệt những hợp đồng này, và thông báo thời hạn đến đối tác.
5	TAO_TAIKHOA N	@ISSTAFF @ISSUPERUSER @USERNAME @PASS @SDT @EMAIL @DIACHI @ISACTIVE @MATK	Tạo một tài khoản dựa vào những thông tin người dùng nhập vào.
6	CAPNHAT_TAIK HOAN	@MATK @USERNAME @PASS @ISSTAFF @ISSUPPERUSER @SDT @EMAIL @DIACHI	Cập nhật tài khoản người dùng dựa trên username của người dùng, và những thông tin và người dùng muốn thay đổi.
7	XEM_DS_TK		Nhân viên xem những tài khoản hiện tại có trong hệ thống.
8	XOA_TK_NV_A D	@USERNAME	Nhân viên thực hiện xóa một tài khoản của admin hoặc nhân viên khác, thông qua username.
9	KHOA_TK	@USERNAME	Nhân viên thực hiện khóa một tài khoản bất kì, thông qua username
10	KICHHOAT_TK	@USERNAME	Nhân viên thực hiện kích hoạt một tài khoản bất kì trong hệ thống thông qua username.
11	CAPQUYEN_DU LIEU	@USERNAME	Người quản trị thực hiện cấp quyền cho một tài khoản bất kì trong hệ thống thông qua user

7. Mô tả lỗi tranh chấp, deadlock

Test case	Loại lỗi	Kịch bản	T1	T2
Testcase1	Lost update	<p>Xét giao tác Đăng kí chi nhánh cho hợp đồng.</p> <p>T1: Đọc số lượng chi nhánh sau đó cập nhật số lượng chi nhánh.</p> <p>T1: Đọc số lượng chi nhánh sau đó cập nhật số lượng chi nhánh. Giả sử hai giao tác này cùng thực hiện trên 1 hợp đồng.</p>	DECLARE @slchinhanh AS INT SET @slchinhanh = (SELECT SLCHINHANH FROM dbo.HOPDONG WHERE MAHD=1)	
				DECLARE @slchinhanh AS INT SET @slchinhanh= (SELECT SLCHINHANH FROM dbo.HOPDONG WHERE MAHD=1)
			INSERT INTO CHINHANH (MAHD, DIACHI) VALUES (1,'Tp hcm');	
				INSERT INTO CHINHANH (MAHD, DIACHI) VALUES (1,'quan 8');
			SET @slchinhanh = @slchinhanh+1 UPDATE dbo.HOPDONG SET SLCHINHANH = @slchinhanh WHERE MAHD=1	
			COMMIT	
				SET @slchinhanh= @slchinhanh+1 UPDATE dbo.HOPDONG SET SLCHINHANH=@slchinha nh WHERE MAHD=1

				COMMIT
Giải thích: Sau khi T1 đọc được số lượng chi nhánh trên bảng hợp đồng thì cùng lúc nào T2 cũng đi vào hệ thống và đọc số lượng chi nhánh tương tự,(vì cả 2 đều phát Slock nên không bị ngăn cản),sau đó cả hai thực hiện insert chi nhánh vào bảng chi nhánh và update lại số lượng sản phẩm trên bảng hợp đồng, dẫn đến số lượng chi nhánh bị ghi đè, chỉ tăng lên 1 trong khi có hai chi nhánh được thêm vào.				
Giải quyết: Để giải quyết vấn đề lost update trên ta có thể set mức cô lập cho giao tác là repeatable read, tuy nhiên nếu set cho giao tác mức cô lập này rất có thể sẽ gây ra deadlock do, T1 và T2 sẽ chờ đợi tài nguyên của nhau. Thay vào đó, vì hai giao tác này thực hiện đọc trước và ghi sau, nên ở thao tác đọc ban đầu để lấy số lượng hiện tại của chi nhánh chúng ta có thể cấp khóa Xlock trên bảng chi nhánh cho giao tác đọc luôn thay vì là Slock như mặc định của SQL Server, điều này giúp cho giao tác chạy trước ngăn giao tác chạy sau đọc số lượng chi nhánh khi nó chưa chạy xong.				
Testcase2	Unrepeatable read	Nhân viên thực hiện xem danh sách hợp đồng của đối tác sau đó gửi những thông báo đến đối tác đối với những hợp đồng sắp đến hạn, tuy nhiên trong lúc nhân viên đang thực hiện xem thì khách hàng thực hiện gia hạn một số hợp đồng, dẫn đến danh sách thông báo gia hạn sau đó bị sai so với ban đầu dự định của nhân viên.	XEM_DSHD_DADUYET (ĐỐI TÁC A)	
				GIAHAN_HD (ĐỐI TÁC A)
				COMMIT
			THONGBAO_GIAHAN (ĐÔI TÁC A)	
			COMMIT	
Giải thích: Nhân viên thứ nhất thực hiện xem danh sách hợp đồng chưa duyệt của đối tác A, tuy nhiên sau đó đối tác A thực hiện gia hạn hợp đồng (vì T1 sinh SL nhưng nhả ra ngay sau đó nên T2 có thể ghi được trên bảng hợp đồng) sau đó T2 commit T1 tiếp tục thực hiện thông báo gia hạn đến đối tác A thì phát hiện danh sách thông báo đã sai.				
Giải quyết: Để giải quyết vấn đề Unrepeatable read trên t chỉ cần set mức cô lập cho thao tác T1 là repeatable read, giúp cho T1 giữ khóa Slock trên bảng hợp đồng cho đến khi giao tác kết thúc, và giao tác T2 sẽ không thể ghi cho đến khi T1 commit.				
Testcase3	Dirty read	Nhân viên thứ nhất thực hiện xem danh sách hợp đồng chưa duyệt của đối tác, sau đó thực hiện duyệt một số hợp đồng, trong khi đang thực hiện duyệt thì nhân viên thứ hai đi vào xem danh sách hợp	XEM_DSHD _CHUADUYET (ĐỐI TÁC A) DUYET_HD (ĐỐI TÁC A)	

		đồng đã duyệt của đối tác để thông báo hạn đến cho khách hàng. Tuy nhiên sau đó nhân viên thứ nhất gặp lỗi phải rollback, dẫn đến danh sách thông báo bị sai.		XEM_DSHD_DADUYET (ĐỐI TÁC A)
			ROLLBACK	
				COMMIT
Giải thích: T1 thực hiện đọc trên bảng hợp đồng, sau đó thực hiện ghi trên bảng hợp đồng, tuy nhiên T2 sau đó đọc trên bảng hợp đồng, và T1 rollback ngay sau đó, dẫn đến dữ liệu T2 đọc bị sai.				
Giải quyết: Để giải quyết vấn đề Dirty read trên, ta thực hiện set mức cô lập cho giao tác T2 là read committed để đảm bảo rằng T2 chỉ đọc khi T1 thực hiện xong.				
Testcase4	Phantom read	Nhân viên thứ nhất thực hiện xem danh sách hợp đồng chưa duyệt của đối tác A Đối tác A thực hiện đăng ký hợp đồng mới. Nhân viên thứ nhất duyệt hợp đồng cho đối tác A, và thông báo thời hạn đối với những hợp đồng mới duyệt. => Sau khi duyệt thì nhân viên phát hiện ra một hợp đồng khác của đối tác, thay vì những hợp đồng như danh sách ban đầu.	XEM_DSHD_CHUADUYET (ĐT A)	
			DUYET_HOPDONG (ĐT A)	
				DANGKI_HOPDONG (ĐỐI TÁC A)
				COMMIT
			XEM_DSHD_CHUADUYET (ĐT A)	
			COMMIT	
Giải thích: T1 đọc và ghi trên bảng hợp đồng => T1 sinh XL trên bảng hợp đồng, nhưng vì T2 thực hiện Insert trên bảng hợp đồng nên không bị chặn bởi XL, sau đó T1 đọc lại bảng hợp đồng thì dữ liệu bị dư ra do T2 insert.				
Giải quyết: Để giải quyết Phantom read trên chúng ta thực hiện set mức cô lập cho giao tác T1 là Serializable, vì T2 là thao tác insert nên mặc dù T1 đang giữ khóa thì T2 vẫn có thể ghi lên bản hợp đồng, vì vậy để tránh trường hợp này thì chúng ta set mức cô lập cho T1 là Serializable để ngăn T2 insert chen ngang.				
Testcase5	Lost update	Hai khách cùng đặt sản phẩm của một đối tác với số lượng sản phẩm là 10, và số lượng sản phẩm cung cấp còn lại của đối tác là 15. Cả hai khách	TAO_DONHANG @MADH, @MADT, @MATX, @HINHTHUCTT, @NGAYTAO, @DIACHI, @PHISP, @PHISHIP, @TONGTIEN	

		hàng kiểm tra số lượng sản phẩm cùng lúc, nên điều pass qua được điều kiện của hệ thống. Sau đó cả 2 cùng đặt và đặt được hàng. Tuy nhiên, số lượng sản phẩm còn lại của đối tác bị giảm thành -5.	, @ TRANGTHAISHIP, @MADH.	
				TAO_DONHANG @MADH, @MADT, @MATX, @HINHTHUCTT, @ NGAYTAO, @DIACHI, @PHISP, @PHISHIP, @TONGTIEN, @ TRANGTHAISHIP, @MADH.
			DECLARE @SLSPCC AS INT SELECT @SLSPCC= SLCUNGCAP FROM dbo.QLSANPHAM WHERE MADT=@MADT AND MASP=@MASP	
				DECLARE @SLSPCC AS INT SELECT @SLSPCC= SLCUNGCAP FROM dbo.QLSANPHAM WHERE MADT=@MADT AND MASP=@MASP
			IF (@SLSP>@SLSPCC) ROLLBACK	
				IF (@SLSP>@SLSPC) ROLLBACK
				THEMSANPHAM @MADH, @MASP, @GIASP, @SLSP, @MADT
				COMMIT
			THEMSANPHAM @MADH, @MASP, @GIASP, @SLSP, @MADT	
			COMMIT	

Giải thích: T1 thực hiện đọc trên bảng QLSANPHAM và thấy số lượng sản phẩm hợp lệ, sau đó T2 cũng đọc trên bảng hợp đồng và cũng thấy dữ liệu hợp lệ (vì cả hai đều sinh ra SL và nhả ra ngay sau đọc nên không bị xung đột). Sau đó cả hai lần lượt Update số lượng sản phẩm lên bảng QLSANPHAM thì dẫn đến SLSP bị âm vì tổng số lượng nhỏ hơn tổng số được đặt.

Giải quyết: Đối với trường hợp lost update trên để giải quyết vấn đề trên thì chúng ta có thể set mức cô lập cho hai bảng Serializable, tuy nhiên việc set mức cô lập này rất có thể sẽ gây ra conversion deadlock bởi vì, T1 thực hiện đọc và giữ khóa đọc sau đó ghi trên bảng QLSanPham và T2 cũng làm điều tương tự. Vì vậy để giải quyết trường hợp này và tránh deadlock thì chúng ta sử dụng có chế khóa UPDLOCK khi thực hiện đọc trên bảng QLSanPham ở cả hai giao tác để đảm bảo khi T1 vào trước và thực hiện UPDLOCK nếu T2 đi vào và cũng thực hiện việc tương tự thì UPDLOCK sẽ ngăn cản T2 vì 2 UPDLOCK không tương thích với nhau.

Testcase6	Deadlock	Khách hàng thực hiện đặt hàng và xác nhận đơn hàng, sau đó khách hàng khách hàng tiếp tục xem danh sách sản phẩm của đối tác, tuy nhiên ngay sau khi khách hàng xác nhận đơn hàng thì đối tác thực hiện sửa một số sản phẩm, sau đó đối tác xem trạng thái của đơn hàng mà khách hàng mới xác nhận	XACNHAN_DONHANG 1	
				SUA_SANPHAM 1 1 ‘GA’
			XEM_DS_SP 1	
				XEM_TTHAI_DONHANG_DT 1
			COMMIT	
				COMMIT

Giải thích: Deadlock vì sau khi khách hàng thực hiện xác nhận đơn hàng thì T1 đã XL trên bảng DONHANG, sau đó đối tác thực hiện sửa sản phẩm thì T2 đã XL trên bảng sản phẩm. Sau đó T1 yêu cầu đọc bảng SANPHAM phải đợi, tương tự T2 yêu cầu đọc bảng DONHANG cũng phải đợi, dẫn đến deadlock

Testcase7	Deadlock	Đối tác T1 thực hiện sửa thông tin sản phẩm, sau đó thực hiện chỉnh sửa chi nhánh của sản phẩm này, nhưng sau khi T1 sửa thông tin sản phẩm thì đối tác T2 thực hiện sửa thông tin chi nhánh của 1 sản phẩm, sau đó mới sửa thông tin của sản phẩm.	SUA_SANPHAM 1, 1, ‘THITGA’	
				SUA_CHINHANH_SP 2, 2, 1, 120, 100
			SUA_CHINHANH_SP 1, 1, 1, 100, 100	
				SUA_SANPHAM 2, 2, ‘MAYTINH’
			COMMIT	

				COMMIT
Giải thích: Đầu tiên đối tác T1 thực hiện sửa sản phẩm nên đã giữ khóa Xlock trên bảng SANPHAM sau đó T2 thực hiện sửa chi nhánh sản phẩm nên giữ khóa XLock trên bảng QLSanPham, sau đó T1 lại thực hiện sửa chi nhánh sản phẩm, nên bắt buộc phát sinh XLock trên bảng QLSanPham những T2 lại đang giữ khóa trên bảng này, nên T1 phải đợi, sau đó T2 thực hiện sửa sản phẩm nên yêu cầu XLock trên bảng SanPham tuy nhiên T1 lại đang XLock trên bảng này nên T2 phải đợi T1, từ đây dẫn đến hai giao tác này đợi lẫn nhau.				
Testcase8	Lost update	Hai tài xế cùng nhận cùng thực hiện nhận một đơn hàng, tài xế thứ nhất thực hiện kiểm tra đơn hàng đây thấy rằng đơn hàng chưa được tài xế nào nhận, sau đó tài xế 2 vào và cũng kiểm tra và cũng hợp lệ, sau đó hai tài xế cùng nhận đơn hàng này, dẫn đến sẽ có một tài xế bị ghi đè kết quả	IF NOT EXISTS (SELECT * FROM DONHANG WHERE MATX IS NULL AND MADH=1) ROLLBACK	
				IF NOT EXISTS (SELECT * FROM DONHANG WHERE MATX IS NULL AND MADH=1) ROLLBACK
			UPDATE dbo.DONHANG SET MATX=1 WHERE MADH=1	
			COMMIT	
				UPDATE dbo.DONHANG SET MATX=2WHERE MADH=1
				COMMIT
Giải thích: T1 thực hiện hiện đọc trên bảng DONHANG được cấp khóa SLock tuy nhiên sau đó nhả khóa ngay vì mức cô lập lúc này chỉ là Read Committed, sau đó T2 cũng thực tương tự như T1, sau đó T1 thực hiện Update trên bảng DONHANG và T2 cũng làm điều tương tự dẫn đến lost update.				
Giải quyết: chúng ta set isolation level repeatable read cho giao tác này để khi T1 đọc thì sẽ giữ khóa đến cuối giao tác, tránh T2 vào hệ thống sau đó.				
Testcase9	Dirty Read	Khách hàng T1 tạo một đơn hàng cho đối tác đó, sau đó T2 xem danh sách đơn hàng của đối tác, nhưng T1 xảy ra lỗi (số lượng sản phẩm đặt lớn hơn số lượng sản phẩm còn lại) nên rollback sau khi tạo đơn hàng.	DATHANG	
				XEM_DS_DONHANG_DT
			ROLLBACK	
				COMMIT
Giải thích: T1 thực hiện tạo đơn hàng (thao tác insert), lúc này T2 thực hiện đọc trên bảng đơn hàng (vì T1 insert nên không phát sinh khóa -> T2 được cấp SL), sau đó T1 kiểm tra số lượng sản phẩm đặt thì thấy lớn hơn số lượng sản phẩm còn lại, nên rollback dẫn đến T2 đọc dữ liệu bần.				

Giải quyết: Để giải quyết vấn đề trên chúng ta chỉ cần set mức cô lập cho giao tác T2 là Read Committed để đảm bảo rằng T2 chỉ đọc khi T1 đã thực hiện xong.

Testcase10	Phantom Read	T1: nhân viên thực hiện xem danh sách tài khoản, sau đó thực hiện cập nhật thông tin tài khoản của khách hàng A, sau đó T2 khách hàng B đăng kí tài khoản, sau đó T1 nhân viên kiểm tra lại danh sách tài khoản, nhận thấy trong danh sách xuất hiện tài khoản B.	XEM_DS_TK	
			CAPNHAT_TAIKHOAN 1	
				DANGKI_KH
			XEM_DS_TK	
				COMMIT
			COMMIT	

Giải thích: T1 đọc và update trên bảng TAIKHOAN -> XLOCK trên TAIKHOAN. T2 thực hiện insert một tài khoản vào bảng TAIKHOAN và KHACHANG (thao tác insert không bị ngăn cản bởi XLOCK) dẫn đến sau đó T2 đọc dữ liệu lại thì thấy trong hệ thống xuất hiện một tài khoản nữa.

Giải quyết: vì T2 thực hiện insert nên để giải quyết vấn đề này thì chúng ta set mức cô lập cho T1 là Serializable để đảm bảo khi T1 chưa thực hiện xong thì T2 không được insert.

Testcase11	Unrepeatable Read	T1: Nhân viên thực hiện xem danh sách đã duyệt của đối tác 1. T2: nhân viên thực hiện duyệt hợp đồng của đối tác 1. T1: Thực hiện thông báo gia hạn hợp đồng của đối tác 1 => Nhưng danh sách hợp đồng sắp đáo hạn của đối tác 1 có thay đổi, vì nhân viên T2 đã thực hiện duyệt thêm hợp đồng cho đối tác 1	XEM_DS_HD_DADUYET (1)	
				DUYET_HOPDONG (1)
				COMMIT
			THONGBAO_GIAHAN(1)	
			COMMIT	

Giải thích: T1 một đọc trên bảng HOPDONG, sau đó T2 update trên bảng HOPDONG (phát sinh XL nhưng không bị ngăn cản vì SL của T1 trên HOPDONG đã được unlock sau khi T1 đọc xong), sau đó T2 rollback và T1 tiếp tục đọc trên bảng HOPDONG thì dữ liệu có thay đổi.

Giải quyết: để giải quyết vấn đề unrepeatable read trên thì chúng ta thực hiện set mức cô lập cho T1 ở mức repeatable read, để đảm bảo khi T1 đọc và nhận được khóa Slock thì sẽ giữ khóa Slock đến cuối giao tác, từ đó T2 không thể thực hiện Update trên bảng hợp đồng khi T1 chưa hoàn thành.

Testcase12	Dirty Read	Khách hàng T1 tạo một đơn hàng cho đối tác đó, sau đó T2 tài xế, T2 vào hệ thống xem danh sách đơn hàng trong khu vực của mình và trong đó có đơn hàng của khách hàng T1 tạo, tuy nhiên sau đó, vì số lượng sản phẩm mà T1 đặt lớn hơn số lượng còn lại của đối tác nên giao tác T1 rollback -> T2 đọc bị lỗi dirty read	TAODONHANG	
				XEM_DS_DONHANG_DT
			THEMSANPHAM	
			ROLLBACK	
				COMMIT

Giải thích: T1 thực hiện tạo đơn hàng (thao tác insert), lúc này T2 thực hiện đọc trên bảng đơn hàng (vì T1 insert nên không phát sinh khóa -> T2 được cấp SL), sau đó T1 kiểm tra số lượng sản phẩm đặt thì thấy lớn hơn số lượng sản phẩm còn lại, nên rollback dẫn đến T2 đọc dữ liệu bẩn.

Giải quyết: Để giải quyết vấn đề trên chúng ta chỉ cần set mức cô lập cho giao tác T2 là Read Committed để đảm bảo rằng T2 chỉ đọc khi T1 đã thực hiện xong.

8. Kết quả thực thi lỗi

Testcase1:

Query4.sql - D...9QD9GB\ACER (53))*

```

1  USE ONLINESHOP
2  GO
3  SELECT * FROM dbo.HOPDONG WHERE MAHD=1
4  BEGIN TRAN
5  DECLARE @slchinhanh AS INT
6  SET @slchinhanh= (SELECT SLCHINHANH
7  FROM dbo.HOPDONG WHERE MAHD=1)
8  WAITFOR DELAY '00:00:10'
9  INSERT INTO CHINHANH
10 (MAHD,DIACHI) VALUES
11 (
12     1,
13     'Quan 7'
14 );
15 SET @slchinhanh=
16 @slchinhanh+1
17 UPDATE dbo.HOPDONG
18 SET SLCHINHANH=@slchinhanh
19 WHERE MAHD=1
20 COMMIT

```

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACCEPTED
1	1	1	2021-11-21	2021-11-21	0	1

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACCEPTED
1	1	2	2021-11-21	2021-11-21	0	1

Query3.sql - D...9QD9GB\ACER (54))*

```

1  USE ONLINESHOP
2  GO
3  SELECT * FROM dbo.HOPDONG WHERE MAHD=1
4  BEGIN TRAN
5  DECLARE @slchinhanh AS INT
6  SET @slchinhanh= (SELECT SLCHINHANH
7  FROM dbo.HOPDONG WHERE MAHD=1)
8  INSERT INTO CHINHANH
9  (MAHD,DIACHI) VALUES
10 (
11     1,
12     'Quan 8'
13 );
14 SET @slchinhanh=
15 @slchinhanh+1
16 UPDATE dbo.HOPDONG
17 SET SLCHINHANH=@slchinhanh
18 WHERE MAHD=1
19 COMMIT
20

```

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACCEPTED
1	1	1	2021-11-21	2021-11-21	0	1

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACCEPTED
1	1	2	2021-11-21	2021-11-21	0	1

Ta thấy rằng ban đầu số lượng chi nhánh của hợp đồng 1 là 1:

	MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
1	1	1	1	2021-11-21	2021-11-21	0	1

	MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
1	1	1	1	2021-11-21	2021-11-21	0	1

Tuy nhiên, sau khi cả hai giao tác cũng thực hiện đăng kí chi nhánh cho hợp đồng này thì số chi nhánh chỉ tăng lên 1, vì vậy đã rơi vào lost update.

	MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
1	1	1	2	2021-11-21	2021-11-21	0	1

Testcase2:

SQLQuery5.sql - D...9QD9GB\ACER (53))*							
1	USE ONLINESHOP						
2	GO						
3	BEGIN TRAN						
4	EXEC dbo.XEM_DSHD_DADUYET @MADT = 1 -- int						
5	WAITFOR DELAY '00:00:10'						
6	EXEC dbo.THONGBAO_GIAHAN @MADT = 1 -- bigint						
7	COMMIT						
8	-- như ta thấy ban đầu thì danh sách hợp đồng thì hợp						
9	-- ngày em test là 22 (cách thời hạn nhỏ hơn 10 ngày						
10	-- 1 thực hiện Gia hạn nên dẫn đến danh sách thông b						
11							
12							
13							

	MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
1	1	1	2	2021-11-21	2021-11-25	12	1
2	2	1	0	2021-11-21	2021-12-25	12	1

SQLQuery4.sql - D...9QD9GB\ACER (54))*							
1	USE ONLINESHOP						
2	GO						
3	BEGIN TRAN						
4	EXEC dbo.GIAHAN_HOPDONG @MAHD = 1,						
5	@TGKT = '2021-12-22', --						
6	@HOAHONG = 10,						
7	@ISACEPTED = 1						
8	COMMIT						
9							

98 %							
Messages							
(1 row affected)							
Completion time: 2021-11-22T11:55:41.0970818+07:00							

Ta thấy rằng, ban đầu số lượng hợp đồng cần thông báo gia hạn của đối tác 1 là hợp đồng 1, vì hợp đồng 1 còn 2 ngày nữa hết hạn (test vào ngày 22/11/2021)

	MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
1	1	1	2	2021-11-21	2021-11-25	12	1
2	2	1	0	2021-11-21	2021-12-25	12	1

Tuy nhiên do sau đó Đối tác 1 thực hiện gia hạn hợp đồng (T2), nên dẫn đến danh sách thông báo gia hạn của đối tác T1 bị rỗng.

	MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
--	------	------	------------	------	------	---------	-----------

Testcase3:

T1.sql - DESKTOP-...9QD9GB\ACER (53)

```

1 USE ONLINESHOP
2 GO
3 BEGIN TRAN
4 EXEC dbo.XEM_DSHD_CHUADUYET @MADT = 2 -- int
5 EXEC dbo.DUYET_HOPDONG @MADT = 2 -- bigint
6 WAITFOR DELAY '00:00:05'
7 ROLLBACK
8
9 EXEC dbo.XEM_DSHD_DADUYET @MADT = 2 -- int
10
11 -- Ta thấy được rằng sau khi gặp lỗi buộc rollback thì
12 -- duyệt của đối tác 2 là rỗng
13
14 -- Tuy nhiên ở Giao tác T2 thì vẫn xuất hiện danh sách
15
16
17
18

```

Results

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACCEPTED
1	3	2	0	2021-11-22	2021-11-30	10
2	4	2	0	2021-11-22	2021-11-30	15

Danh sách hợp đồng đã duyệt

MAHD	MADT	EXPRIEDTIME	ACCEPTEDDATE
1	3	2	2021-11-30
2	4	2	2021-11-30

T2.sql - DESKTOP-...9QD9GB\ACER (54)

```

1 USE ONLINESHOP
2 GO
3 BEGIN TRAN
4 SET TRAN ISOLATION LEVEL READ UNCOMMITTED
5 EXEC dbo.XEM_DSHD_DADUYET @MADT = 2 -- int
6 COMMIT
7
8
9
10 -- vì sql server tự cấp khóa XLOCK khi ghi ở T1, nên em
11 -- set mức cô lập của T2 là read uncommitted để có thể test
12 -- được dirty read

```

Results

Danh sách hợp đồng đã duyệt

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACCEPTED
1	3	2	0	2021-11-22	2021-11-30	10
2	4	2	0	2021-11-22	2021-11-30	15

Ta thấy rằng danh sách hợp đồng đã duyệt của đối tác 2 là rỗng (do T1 rollback), tuy nhiên T2 vẫn đọc được danh sách hợp đồng đã duyệt của đối tác 2, do T2 đọc khi T1 chưa thực hiện rollback.

Testcase4:

T1.sql - DESKTOP-...9QD9GB\ACER (53)

```

1 USE ONLINESHOP
2 GO
3 BEGIN TRAN
4 EXEC dbo.XEM_DSHD_CHUADUYET @MADT = 2 -- int
5 WAITFOR DELAY '00:00:05'
6 EXEC dbo.DUYET_HOPDONG @MADT = 2 -- bigint
7 COMMIT
8
9 -- Ta thấy được rằng ban đầu danh sách hợp đồng chưa
10 -- chỉ hợp đồng 3 4, tuy nhiên sau khi duyệt và thông
11 -- xuất hiện thêm hợp đồng 5, do T2 thực hiện Đăng kí
12 -- sau khi T1 xem danh sách
13

```

Results

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACCEPTED
1	3	2	0	2021-11-22	2021-11-30	10
2	4	2	0	2021-11-22	2021-11-30	15

Danh sách hợp đồng cần duyệt

MAHD	MADT	EXPRIEDTIME	ACCEPTEDDATE
1	3	2	2021-11-30
2	4	2	2021-11-30
3	5	2	2021-11-30

Danh sách hợp đồng được duyệt

T2.sql - DESKTOP-...9QD9GB\ACER (54)

```

1 USE ONLINESHOP
2 GO
3 BEGIN TRAN
4 EXEC dbo.DANGKI_HOPDONG @MADT = 2,
5 @TGBD = '2021-11-22',
6 @TGKT = '2021-11-30',
7 @HOAHONG = 10
8 COMMIT
9

```

Messages

(1 row affected)

Completion time: 2021-11-22T12:17:55.2283498+07:00

Ta thấy rằng danh sách hợp đồng cần duyệt và danh sách hợp đồng được duyệt của 2 lần đọc ở T1 khác nhau, do trong T1 đang đọc và duyệt hợp đồng, thì T2 thực hiện đăng kí hợp đồng.

Testcase5:

```
ESKTOP-...9QD9GB\ACER (53))
BEGIN TRANSACTION
DECLARE @MADH AS BIGINT, @GIASP AS DECIMAL(15,2), @PHISP AS DECIMAL(15,2), @TONGTIEN AS DECIMAL(15,2)
SELECT @GIASP=GIASP FROM dbo.QLSANPHAM WHERE MASP=1 AND MADT=1
SET @PHISP=@GIASP*10
SET @TONGTIEN= @PHISP+10000

DECLARE @MADH1 BIGINT;
EXEC dbo.TAO_DONHANG @MAKH = 1, -- bigint
                    @MADT = 1, -- bigint
                    @MATX = NULL, -- bigint
                    @HINHTHUCTT = 'CARD', -- varchar(20)
                    @NGAYTAO = '2021-11-21 13:20:04', -- datetime
                    @DIACHIGH = 'QUAN BT', -- varchar(100)
                    @PHISP = @PHISP, -- decimal(15, 2)
                    @PHISHIP = 10000, -- decimal(15, 2)
                    @TONGTIEN = @TONGTIEN, -- decimal(15, 2)
                    @TRANGTHAISHIP = 0, -- int
                    @TRANGTHAITTOAN = 0, -- bit
                    @MADH = @MADH1 OUTPUT -- bigint

DECLARE @SLSPCC AS INT
SELECT @SLSPCC= SLCUNGCAP FROM dbo.QLSANPHAM WHERE MADT=1 AND MASP=1

WAITFOR DELAY '00:00:05'

IF (10>@SLSPCC)
BEGIN
    RAISERROR('số lượng sp không đủ',15,1)
    ROLLBACK
END
EXEC dbo.THEMSANPHAM @MADH = @MADH1, -- bigint
                    @MASP = 1, -- bigint
                    @GIASP = @GIASP, -- decimal(15, 2)
                    @SLSP = 10, -- int
                    @MADT=1

COMMIT
```

```
ESKTOP-...9QD9GB\ACER (54))  procedure_phu.sql...9QD9GB\ACER (55))
GO
BEGIN TRANSACTION
DECLARE @MADH AS BIGINT, @GIASP AS DECIMAL(15,2), @PHISP AS DECIMAL(15,2), @TONGTIEN AS DECIMAL(15,2)
SELECT @GIASP=GIASP FROM dbo.QLSANPHAM WHERE MASP=1 AND MADT=1
SET @PHISP=@GIASP*10
SET @TONGTIEN= @PHISP+10000

DECLARE @MADH1 BIGINT;
EXEC dbo.TAO_DONHANG @MAKH = 2, -- bigint
                    @MADT = 1, -- bigint
                    @MATX = NULL, -- bigint
                    @HINHTHUCTT = 'CARD', -- varchar(20)
                    @NGAYTAO = '2021-11-21 13:20:04', -- datetime
                    @DIACHIGH = 'QUAN 8', -- varchar(100)
                    @PHISP = @PHISP, -- decimal(15, 2)
                    @PHISHIP = 10000, -- decimal(15, 2)
                    @TONGTIEN = @TONGTIEN, -- decimal(15, 2)
                    @TRANGTHAISHIP = 0, -- int
                    @TRANGTHAITTOAN = 0, -- bit
                    @MADH = @MADH1 OUTPUT -- bigint

DECLARE @SLSPCC AS INT
SELECT @SLSPCC= SLCUNGCAP FROM dbo.QLSANPHAM WHERE MADT=1 AND MASP=1]

IF (10>@SLSPCC)
BEGIN
    RAISERROR('số lượng sp không đủ',15,1)
    ROLLBACK
END
EXEC dbo.THEMSANPHAM @MADH = @MADH1, -- bigint
                    @MASP = 1, -- bigint
                    @GIASP = @GIASP, -- decimal(15, 2)
                    @SLSP = 10, -- int
                    @MADT=1

COMMIT
```

DESKTOP-G9QD9G...- dbo.QLSANPHAM					
	MADT	MASP	MACN	GIASP	SLCUNGCAP
▶	1	1	1	15000.00	-5
	1	2	1	125.00	100
*	NULL	NULL	NULL	NULL	NULL

Ở trong test case trên thì số lượng sản phẩm 1 của đối tác 1 ban đầu là 15, sau đó T1 và T2 cùng vào hệ thống và thực hiện đặt hàng với sản phẩm 1 của đối tác 1 với số lượng 10, và cả hai đều qua được giai đoạn kiểm tra số lượng sản phẩm, do cả hai đều nhỏ hơn 15, tuy nhiên sau đó cả hai cùng thực hiện thêm sản phẩm thì số lượng sản phẩm còn lại của sản phẩm 1 của đối tác 1 bị giảm xuống -5.

Testcase6:

```

1  USE ONLINESHOP
2  GO
3  BEGIN TRAN
4  EXEC dbo.DATHANG @MAKH = 1,
5      @MASP = 1,
6      @MADT = 1,
7      @SLSP = 10,
8      @HINHTHUCTT = 'card',
9      @DIACHIGH = 'hcm',
10     @PHISHIP = '12000'
11  WAITFOR DELAY '00:00:05'
12  EXEC dbo.XEM_DS_SP @MADT = 1 -- bigint
13  COMMIT
14
15
16

```

98 %

	MASP	MALH	TENSP
1	1	1	gao
2	2	1	ga

```

1  USE ONLINESHOP
2  GO
3  BEGIN TRAN
4  EXEC dbo.SUA_SANPHAM @MASP = 1, -- big
5      @MALH = 1, -- big
6      @TENSP = 'phuc' --
7  WAITFOR DELAY '00:00:05'
8  EXEC dbo.XEM_TT_DONHANG_DT @MADT = 1 --
9  COMMIT
10
11
12

```

98 %

(1 row affected)

Msg 1205, Level 13, State 51, Procedure XEM_TT_DONHANG_DT, Line 4
Transaction (Process ID 54) was deadlocked on lock resources with

Completion time: 2021-11-22T10:34:40.7608720+07:00

Deadlock do T1 giữ khóa trên bảng QLSanPham, sau đó yêu cầu tài nguyên trên bảng SanPham, mà bảng này bị T2 khóa XLock và T2 lại yêu cầu tài nguyên trên bảng QLSanPham đang bị T1 khóa, dẫn đến circle deadlock.

Testcase7:

QLQuery3.sql - D...9QD9GB\ACER (53))*

```
1 USE ONLINESHOP
2 GO
3 BEGIN TRAN
4 EXEC dbo.SUA_SANPHAM @MASP = 1, -- begin
5 @MALH = 1, -- begin
6 @TENSP = 'Thit ga'
7 WAITFOR DELAY '00:00:05'
8 EXEC dbo.SUA_CHINHANH_SP @MADT = 1,
9 @MASP = 1,
10 @MACN = 1,
11 @GIASP = 15000,
12 @SLCUNGCAP = 10
13 COMMIT
14
15
```

8 %

Messages

Msg 1206, Level 13, State 51, Procedure SUA_CHINHANH_SP, Line 8 (Batch Transaction (Process ID 53)) was deadlocked on lock resources with another process and has chosen to abort because it has no activity in the process. If you are aborting this process, you may also want to consider aborting the other process. Completion time: 2021-11-22T11:39:14.2986199+07:00

SQLQuery2.sql - D...9QD9GB\ACER (54))*

```
1 USE ONLINESHOP
2 GO
3 BEGIN TRAN
4 EXEC dbo.SUA_CHINHANH_SP @MADT = 1,
5 @MASP = 1,
6 @MACN = 1,
7 @GIASP = 15000,
8 @SLCUNGCAP = 15
9 WAITFOR DELAY '00:00:05'
10 EXEC dbo.SUA_SANPHAM @MASP = 1, -- begin
11 @MALH = 1, -- begin
12 @TENSP = 'Thit bo'
13 COMMIT
14
15
```

98 %

Messages

(1 row affected)
Completion time: 2021-11-22T11:39:14.2776713+07:00

Testcase8:

T2.sql - DESKTOP-...9QD9GB\ACER (55))*

```
1 USE ONLINESHOP
2 GO
3 EXEC NHAN_DONHANG_LOI @MADH=1, @MATX=2
```

100 %

Messages

(1 row affected)
Completion time: 2021-12-18T22:04:13.9376098+07:00

T1.sql - DESKTOP-...9QD9GB\ACER (54))*

```
1 USE ONLINESHOP
2 GO
3 CREATE PROC NHAN_DONHANG_LOI
4 @MADH BIGINT,
5 @MATX BIGINT
6 AS
7 BEGIN TRAN
8 IF NOT EXISTS (SELECT * FROM DONHANG WHERE MATX IS NULL AND MADH=@MADH)
9 ROLLBACK
10 ELSE
11 WAITFOR DELAY '00:00:10'
12 BEGIN
13 UPDATE dbo.DONHANG SET MATX=@MATX WHERE MADH=@MADH
14 COMMIT TRAN
15 END
16 GO
17 EXEC NHAN_DONHANG_LOI @MADH=1, @MATX=1
18 SELECT MADH,MATX FROM DONHANG
19 -- Ta thấy sau khi T1 thêm 1 sản phẩm thì, T2 ghi đè nên đã xóa sản phẩm
20 -- dẫn đến trong cơ sở dữ liệu không có sản phẩm bánh trung thu
```

100 %

Results

MADH	MATX
1	2
2	NULL
3	NULL
4	NULL
5	NULL

Messages

Ta thấy mặc dù T1 thực hiện trước tuy nhiên T2 mới là người nhận được đơn hàng này.

Testcase9:

```

4 BEGIN TRAN
5 DECLARE @MADH AS BIGINT, @GIASP AS DECIMAL(15,2), @PHISP AS DECIMAL(15,2), @TONGTIEN AS DECIMAL(15,2)
6 SELECT @GIASP=GIASP FROM dbo.QLSANPHAM WHERE MASP=1 AND MADT=1
7 SET @PHISP=@GIASP*15
8 SET @TONGTIEN= @PHISP+10000
9
10 DECLARE @MADH1 BIGINT;
11 EXEC dbo.TAO_DONHANG @MAKH = 2, -- bigint
12 @MADT = 1, -- bigint
13 @MATX = NULL, -- bigint
14 @HINHTHUCTT = 'CARD', -- varchar(20)
15 @NGAYTAO = '2021-11-21 13:20:04', -- datetime
16 @DIACHIGH = 'QUAN 11', -- varchar(100)
17 @PHISP =@PHISP, -- decimal(15, 2)
18 @PHISHIP = 15000, -- decimal(15, 2)
19 @TONGTIEN = @TONGTIEN, -- decimal(15, 2)
20 @TRANGTHAISHIP = 0, -- int
21 @TRANGTHAITTOAN = 0, -- bit
22 @MADH = @MADH1 OUTPUT -- bigint
23
24 DECLARE @SLSPCC AS INT
25 SELECT @SLSPCC= SLCUNGCAP FROM dbo.QLSANPHAM WHERE MADT=1 AND MASP=1
26
27 WAITFOR DELAY '00:00:05'
28
29 IF (15>@SLSPCC)
30 BEGIN
31 RAISERROR('số lượng sp không đủ',15,1)
32 ROLLBACK
33 END
34 EXEC dbo.THEMSANPHAM @MADH = @MADH1, -- bigint
35 @MASP = 1, -- bigint
36 @GIASP = @GIASP, -- decimal(15, 2)
37 @SLSP = 10, -- int
38 @MADT=1
39 COMMIT
40 EXEC dbo.XEM_DS_DONHANG_DT @MADT = 1 -- bigint

```

Results Messages

(1 row affected)
Msg 50050, Level 15, State 1, Line 30
số lượng sp không đủ
Msg 547, Level 16, State 0, Procedure THEMSANPHAM, Line 10 [Batch Start Line 2]
The INSERT statement conflicted with the FOREIGN KEY constraint "FK_QLDH_SH". The conflict occurred in database "ONLINESHOP", table "THEMSANPHAM". The statement has been terminated.

T2.sql - DESKTOP-...9QD9GB\ACER (54)

```

1 USE ONLINESHOP
2 GO
3 BEGIN TRAN
4 SET TRAN ISOLATION LEVEL READ UNCOMMITTED
5 EXEC dbo.XEM_DS_DONHANG_DT @MADT = 1 -- bigint
6 COMMIT TRAN
7 -- Ta thấy rằng mặc dù sau khi không đủ hàng thì T1 đã roll back nhưng T2
8 -- vẫn đọc được đơn hàng này của T1 đã hủy.

```

98 %

Results Messages

	DH	MAKH	MADT	MATX	HINHTHUCTT	NGAYTAO	DIACHIGH	PHISP	PHISHIP	TONGTIEN	TRAN
1		1	1	NULL	card	2019-02-25 18:11:00.160	quan 7	125.00	124.00	124.00	0
2		1	1	NULL	card	2021-11-21 13:20:04.000	quan 9	1200.00	120.00	1320.00	0
3		1	1	NULL	card	2021-11-21 13:20:04.000	hcm	1200.00	12000.00	13200.00	0
4		1	1	NULL	card	2021-11-21 13:20:04.000	hcm	1200.00	12000.00	13200.00	0
5		1	1	NULL	CARD	2021-11-21 13:20:04.000	QUAN BT	150000.00	10000.00	160000.00	0
6		2	1	NULL	CARD	2021-11-21 13:20:04.000	QUAN 8	150000.00	10000.00	160000.00	0
7		2	1	NULL	CARD	2021-11-21 13:20:04.000	QUAN 11	225000.00	15000.00	235000.00	0

Ta thấy, rằng T1 thực hiện giao tác mua hàng, sản phẩm 1 của đối tác 1, với số lượng hàng là 15 tuy nhiên trong hệ thống chỉ còn 10 sản phẩm, dẫn đến bị rollback, tuy nhiên trước khi rollback thì T2 đọc bảng đơn hàng, do T1 đã tạo đơn hàng trước đó nên T2 đọc bảng đơn hàng bị sai, rơi vào dirty read.

Testcase10:

T1.sql - DESKTOP-...9QD9GB\ACER (53))

```

1  USE ONLINESHOP
2  GO
3  BEGIN TRAN
4      EXEC dbo.XEM_DS_TK
5      EXEC dbo.CAPNHAT_TAIKHOAN @MATK = 1, -- begin
6                                     @USERNAME = 'phuccccc',
7                                     @PASS = '1234564',
8                                     @ISSTAFF = 1, -- bit
9                                     @ISSUPPERUSER = 0, -- bit
10                                    @SDT = '12345',
11                                    @EMAIL = 'phuc@gmail.commm',
12                                    @DIACHI = 'Quan 10'
13      WAITFOR DELAY '00:00:05'
14      EXEC dbo.XEM_DS_TK
15  COMMIT
16  -- ta thấy danh sách tài khoản đã xuất hiện thêm một tài kh
17  -- do T2 đã thực hiện đăng kí khách hàng nên thêm một tài kh
18  -- vào bảng
19

```

100 %

Results Messages

	MATK	USERNAME	PASS	ISSTAFF	ISSUPERUSER	SDT	EMAIL
1	1	phuc	lp_2-"6PFI-[@	0	0	123	phuc@gmail.co
2	2	phu	lp_2-"6PFI-[@	0	0	123	phuc@gmail.co
3	3	phucc	lp_2-"6PFI-[@	0	0	1234	phuc@gmail.co
4	4	PHUC11	_xü%"JAwxypE>	0	0	12345	PHUC11@GM
5	5	phu1	#%+Ç;.,WUİLO&A i	0	0	123456789	phu@gmail.co
6	6	quang	_xü%"JAwxypE>	0	0	123456	quang@gmail

	MATK	USERNAME	PASS	ISSTAFF	ISSUPERUSER	SDT	EMAIL
1	1	phucccc	1234564	1	0	12345	phuc@gmail.co
2	2	phu	lp_2-"6PFI-[@	0	0	123	phuc@gmail.co
3	3	phucc	lp_2-"6PFI-[@	0	0	1234	phuc@gmail.co
4	4	PHUC11	_xü%"JAwxypE>	0	0	12345	PHUC11@GM
5	5	phu1	#%+Ç;.,WUİLO&A i	0	0	123456789	phu@gmail.co
6	6	quang	_xü%"JAwxypE>	0	0	123456	quang@gmail
7	7	dinhphuc	27f&@TnB"ôl...	0	0	12345678	dinhphuc@gm

T2.sql - DESKTOP-...9QD9GB\ACER (54))

```

1  USE ONLINESHOP
2  GO
3  BEGIN TRAN
4      EXEC dbo.DANGKI_KH @USERNAME_KH = 'dinhphuc', -- char(30)
5                          @PASS_KH = '123456', -- char(30)
6                          @SDT_KH = '12345678', -- char(30)
7                          @EMAIL_KH = 'dinhphuc@gmail.com', -- char(50)
8                          @DIACHI_KH = 'Quan binhthanh'
9  COMMIT
10

```

98 %

Messages

(1 row affected)

(1 row affected)

Completion time: 2021-11-22T14:44:19.6096391+07:00

Testcase11:

T1.sql - DESKTOP-...9QD9GB\ACER (53))

```

1  USE ONLINESHOP
2  GO
3
4  BEGIN TRAN
5      EXEC dbo.XEM_DS_HD_DADUYET @MADT = 1 -- int
6      WAITFOR DELAY '00:00:05'
7      EXEC dbo.THONGBAO_GIAHAN @MADT = 1 -- bigint
8  COMMIT
9
10  -- Ta thấy ban đầu chỉ có 2 hợp đồng cần thông báo G
11  -- hợp đồng 1 và 2 vì khoảng cách đến ngày hôm nay (
12  -- nhỏ hơn 10 ngày.
13  -- Tuy nhiên thực tế thì có thêm hợp đồng 6, vì T2 đ
14  -- 6 và vô tình hợp đồng này cũng sắp đáo hạn dẫn đế
15  -- thay đổi đi.
16
17
18
19

```

100 %

Results Messages

	MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACCEPTED
1	1	1	2	2021-11-21	2021-11-22	10	1
2	2	1	0	2021-11-21	2021-11-25	12	1

	MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACCEPTED
1	1	1	2	2021-11-21	2021-11-22	10	1
2	2	1	0	2021-11-21	2021-11-25	12	1
3	6	1	0	2021-11-22	2021-11-26	10	1

T2.sql - DESKTOP-...9QD9GB\ACER (54))

```

1  USE ONLINESHOP
2  GO
3
4  BEGIN TRAN
5      EXEC dbo.DUYET_HOPDONG @MADT = 1
6  COMMIT
7

```

98 %

Results Messages

	MAHD	MADT	EXPRIEDTIME	ACCEPTEDDATE
1	6	1	2021-11-26	2021-11-22 14:35:25.740

Testcase12:

Số lượng sản phẩm 1 của đối tác 2:

MADT	MASP	MACN	GIASP	SLCUNGCAP
1	1	1	9000.00	4
1	2	2	9000.00	9
2	1	3	9000.00	10
2	3	4	9000.00	10
3	4	5	18000.00	10
3	6	6	18000.00	10

T1: Khách hàng 1 đặt 15 sản phẩm 1 của đối tác 2 (Vì số lượng đặt hàng lớn hơn nên sau khi tạo đơn hàng thì T1 đã thực hiện rollback -> đơn hàng này bị hủy):

```
EXEC DATHANG_LOI 1,1,2,15,'CARD','QUAN 11', 10000
```

T2: Tài xế 1 xem danh sách đơn hàng trong khu vực (trùng khu vực của đối tác 2 nên sẽ thấy được đơn hàng của đối tác 2):

```
BEGIN TRAN
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
EXEC HIENHI_DSDH 1
COMMIT
```

Kết quả T2 đọc được:

esults Messages						
MADH	MAKH	MADT	MATX	HINHTHUCTT	NGAYTAO	DIACHIGH
6	3	2	NULL	NULL	2021-10-11 12:00:00.000	44 Cach Mang Thang
9	6	2	NULL	NULL	2021-10-11 12:00:00.000	80 Ly Thai To, phuong
65	1	2	NULL	CARD	2021-12-19 13:16:42.223	QUAN 11

Kết quả sau khi T1 và T2 hoàn thành:

esults Messages						
MADH	MAKH	MADT	MATX	HINHTHUCTT	NGAYTAO	DIACHIGH
6	3	2	NULL	NULL	2021-10-11 12:00:00.000	44 Cach Mang Thang
9	6	2	NULL	NULL	2021-10-11 12:00:00.000	80 Ly Thai To, phuong

9. Demo xử lý lỗi

Test case 1:

- Ban đầu:

	MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
1	1	1	4	2021-01-02	2022-02-02	15	1

Số lượng chi nhánh ban đầu là 4.

T1: thêm UPDLOCK cho giao tác

```
CREATE PROC DANGKI_CHINHANH_HOPDONG_FIX
    @MAHD BIGINT,
    @DIACHI VARCHAR(100)
AS
BEGIN TRAN
    DECLARE @slchinhanh AS INT
    SET @slchinhanh= (SELECT SLCHINHANH FROM dbo.HOPDONG (UPDLOCK) WHERE MAHD=@MAHD)
    WAITFOR DELAY '00:00:10'
    INSERT INTO CHINHANH(MAHD,DIACHI) VALUES (
        @MAHD,
        @DIACHI
    );
    SET @slchinhanh=@slchinhanh+1
    UPDATE dbo.HOPDONG SET SLCHINHANH=@slchinhanh WHERE MAHD=@MAHD
COMMIT TRAN

EXEC DANGKI_CHINHANH_HOPDONG_FIX 1, 'Quận 7'
```

T2: Số lượng hợp chi nhánh lúc sau đã tăng lên 6 (đúng)

```
GO
EXEC DANGKI_CHINHANH_HOPDONG_FIX 1, 'Quận 8'

SELECT * FROM dbo.HOPDONG WHERE MAHD=1
```

	MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
1	1	1	6	2021-01-02	2022-02-02	15	1

Test case 2:

T1: SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

```
CREATE PROC XEM_DSHD_DADUYET_TB_GIAHAN
    @MADT BIGINT
AS
BEGIN TRAN
    SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
    EXEC dbo.XEM_DSHD_DADUYET @MADT -- int
    WAITFOR DELAY '00:00:10'
    EXEC dbo.THONGBAO_GIAHAN @MADT-- bigint
COMMIT
EXEC XEM_DSHD_DADUYET_TB_GIAHAN 1
```

T2:

```
BEGIN TRAN
EXEC dbo.GIAHAN_HOPDONG @MAHD = 1,
                        @TGKT = '2021-12-22',
                        @HOAHONG = 10,
                        @ISACEPTED = 1
COMMIT
```

Danh sách hợp đồng chưa duyệt (T1 xem lần 1)

	MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
1	1	1	6	2021-01-02	2022-02-02	15	1
2	6	1	3	2021-01-02	2022-01-02	10	1

Danh sách thông báo gia hạn (T2 xem lần 2)

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
------	------	------------	------	------	---------	-----------

Ta thấy, vì thời gian hết hạn của hai hợp đồng đều còn hơn 10 ngày, nên danh sách thông báo gia hạn rỗng (đúng), mặc dù T2 thực hiện thêm hợp đồng có thời gian hết hạn là 22-12 (test ngày 18-12, nên dưới 10 ngày).

Test case 3:

T1:

```
BEGIN TRAN
EXEC dbo.XEM_DSHD_CHUADUYET @MADT = 2 -- int
EXEC dbo.DUYET_HOPDONG @MADT = 2 -- bigint
WAITFOR DELAY '00:00:05'
ROLLBACK
```

T2: set isolation level read committed

```
BEGIN TRAN
SET TRAN ISOLATION LEVEL READ COMMITTED
EXEC dbo.XEM_DSHD_DADUYET @MADT = 2 -- int
COMMIT
```

Danh sách hợp đồng chưa duyệt của đối tác 2 (T1)

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
2	2	2	2021-11-20	2022-11-20	8	0
7	2	2	2021-11-20	2022-11-20	8	0

Danh sách hợp đồng duyệt của đối tác 2(trước khi T1 rollback):

MAHD	MADT	EXPRIEDTIME	ACCEPTEDDATE
2	2	2022-11-20	2021-12-19 00:09:44.897
7	2	2022-11-20	2021-12-19 00:09:44.897

Tuy nhiên, T1 rollback nên danh sách hợp đồng đã duyệt của T2 đọc là rỗng (đúng):

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
------	------	------------	------	------	---------	-----------

Vậy ta đã giải quyết được dirty read.

Test case 4:

T1: SET TRANSACTION ISOLATION LEVEL SERIALIZABLE

```

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
BEGIN TRAN
EXEC dbo.XEM_DSHD_CHUADUYET @MADT=2 -- in
WAITFOR DELAY '00:00:05'
EXEC dbo.DUYET_HOPDONG @MADT=2 -- bigint
EXEC dbo.XEM_DSHD_DADUYET 2
COMMIT

```

T2:

```

BEGIN TRAN
EXEC dbo.DANGKI_HOPDONG @MADT = 2,
                        @TGBD = '2021-12-20',
                        @TGKT = '2022-1-30',
                        @HOAHONG = 10
COMMIT

```

Danh sách hợp đồng chưa duyệt của đối tác 2 (lần đọc thứ nhất của T1)

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
2	2	2	2021-11-20	2022-11-20	8	0
7	2	2	2021-11-20	2022-11-20	8	0

Danh sách hợp đồng đã duyệt của đối tác 2 (lần đọc thứ 2 của T1)

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
2	2	2	2021-11-20	2022-11-20	8	1
7	2	2	2021-11-20	2022-11-20	8	1

Mặc dù T2 đã insert thêm một hợp đồng tuy nhiên danh sách hợp đồng chưa duyệt và đã duyệt trong một giao tác vẫn đồng nhất, chúng ta đã giải quyết được phantom read.

Test case 5:

Số lượng cung cấp sản phẩm 1 của đối tác 1 là 10:

	MADT	MASP	MACN	GIASP	SLCUNGCAP
1	1	1	1	9000.00	10
2	1	2	2	9000.00	9
3	2	1	3	9000.00	10
4	2	3	4	9000.00	10
5	3	4	5	18000.00	10
6	3	6	6	18000.00	10

Ta thực hiện đặt UPDLOCK trên bảng QLSANPHAM cho giao tác đặt hàng, để đảm bảo hai khách hàng không thể kiểm tra cùng lúc.

```

ALTER PROC DATHANG_FIX @MAKH BIGINT, @MASP BIGINT, @MADT BIGINT, @SLSP INT,
@HINHTHUUCT VARCHAR(20),
@DIACHIGH VARCHAR(100),
@PHISHIP DECIMAL(15,2)
AS
BEGIN TRAN
DECLARE @CURRENTDATE AS DATETIME, @MADH AS BIGINT, @GIASP AS DECIMAL(15,2),
@PHISP AS DECIMAL(15,2), @TONGTIEN AS DECIMAL(15,2)
SELECT @GIASP=GIASP FROM dbo.QLSANPHAM WHERE MASP=@MASP AND MADT=@MADT
SET @PHISP=@GIASP*@SLSP
SET @TONGTIEN= @PHISP*@PHISHIP
SET @CURRENTDATE=(Select GetDate())
DECLARE @MADH1 BIGINT;
EXEC dbo.TAO_DONHANG @MAKH = @MAKH, @MADT = @MADT, @MATX = NULL,
@HINHTHUUCT = @HINHTHUUCT, @NGAYTAO = @CURRENTDATE,
@DIACHIGH = @DIACHIGH, @PHISP = @PHISP,
@PHISHIP = @PHISHIP, @TONGTIEN = @TONGTIEN,
@TRANGTHAISHIP = 0, @TRANGTHAITTOAN = 0,
@MADH = @MADH1 OUTPUT

DECLARE @SLSPCC AS INT
SELECT @SLSPCC= SLCUNGCAP FROM dbo.QLSANPHAM (UPDLOCK) WHERE MADT=@MADT AND MASP=@MASP
WAITFOR DELAY '00:00:05'
print (@SLSP)
print(@SLSPCC)
IF (@SLSP>@SLSPCC)
rollback tran
ELSE
begin
EXEC dbo.THEMSANPHAM @MADH = @MADH1, -- bigint
@MASP = @MASP, -- bigint
@GIASP = @GIASP, -- decimal(15, 2)
@SLSP = @SLSP, -- int
@MADT=@MADT

COMMIT tran
end

```

T1: Khách hàng 1 đặt 10 sản phẩm 1 của đối tác 1

```
EXEC DATHANG_FIX 1,1,1,10, 'CARD', 'QUAN BT', 10000
```

T2: Khách hàng 2 đặt 10 sản phẩm 1 của đối tác 1

```
EXEC DATHANG_FIX 2,1,1,10, 'CARD', 'QUAN BT', 10000
```

Số lượng sản phẩm sau khi khách hàng đặt hàng:

MADT	MASP	MACN	GIASP	SLCUNGCAP
1	1	1	9000.00	0
1	2	2	9000.00	9
2	1	3	9000.00	10
2	3	4	9000.00	10
3	4	5	18000.00	10
3	6	6	18000.00	10

Bảng đơn hàng:

MADH	MAKH	MADT	MATX	HINHTHUUCT	NGAYTAO	DIACHIGH	PHISP	PHISHIP	TONGTIEN	TRANGTHAISHIP	TRANGTHAITTOAN
1	1	1	1	NULL	2021-10-11 12:00:00.000	36 Tran Binh Trong, phuong 5, quan 5, Thanh pho Ho Chi Minh	NULL	NULL	NULL	NULL	NULL
2	2	1	NULL	NULL	2021-10-11 12:00:00.000	265 Truong Chinh, phuong 3, quan Go Vap, Thanh pho Ho Chi Minh	NULL	NULL	NULL	NULL	NULL
3	2	3	NULL	NULL	2021-10-11 12:00:00.000	265 Truong Chinh, phuong 3, quan Go Vap, Thanh pho Ho Chi Minh	NULL	NULL	NULL	NULL	NULL
4	3	4	NULL	NULL	2021-10-11 12:00:00.000	98 Ly Chinh Thang, phuong 10, quan Binh Thanh, Thanh pho Ho Chi Minh	NULL	NULL	NULL	NULL	NULL
5	3	5	NULL	NULL	2021-10-11 12:00:00.000	44 Cach Mang Thang Tam, phuong 2, quan Tan Binh, Thanh pho Ho Chi Minh	NULL	NULL	NULL	NULL	NULL
6	3	2	1	NULL	2021-10-11 12:00:00.000	44 Cach Mang Thang Tam, phuong 2, quan Tan Binh, Thanh pho Ho Chi Minh	NULL	NULL	NULL	NULL	NULL
7	4	4	NULL	NULL	2021-10-11 12:00:00.000	80 Ly Thai To, phuong 6, quan 5, Thanh pho Ho Chi Minh	NULL	NULL	NULL	NULL	NULL
8	4	4	NULL	NULL	2021-12-11 12:00:00.000	80 Ly Thai To, phuong 6, quan 5, Thanh pho Ho Chi Minh	NULL	NULL	NULL	NULL	NULL
9	6	2	NULL	NULL	2021-10-11 12:00:00.000	80 Ly Thai To, phuong 6, quan 5, Thanh pho Ho Chi Minh	NULL	NULL	NULL	NULL	NULL
10	1	1	NULL	card	2021-11-21 13:20:04.000	Quan Cam	90000.00	10000.00	100000.00	0	0
12	1	1	NULL	card	2021-11-21 13:20:04.000	quan 1	9000.00	10000.00	19000.00	0	0
20	1	1	NULL	CARD	2021-12-19 00:28:53.157	QUAN BINH THANH T1	90000.00	10000.00	100000.00	0	0

T1 đặt được hàng vì T1 đến trước, T2 không đặt được hàng -> giải quyết được lost update

Test case 8:

Tình trạng đơn hàng 1 trước khi thực hiện nhận đơn hàng (MATX là null)

MADH	MAKH	MADT	MATX	HINHTHUC...	NGAYTAO	DIACHIGH	PHISP
1	1	1	NULL	NULL	2021-10-11 ...	36 Tran Binh...	NULL

Thêm UPDLOCK cho giao tác nhận hàng

T1:

```
CREATE PROC NHAN_DONHANG_FIX
    @MADH BIGINT,
    @MATX BIGINT
AS
    set transaction isolation level repeatable read
    BEGIN TRAN
    IF NOT EXISTS (SELECT * FROM DONHANG WHERE MATX IS NULL AND MADH=@MADH)
        ROLLBACK TRAN
    ELSE
        BEGIN
            WAITFOR DELAY '00:00:10'
            UPDATE dbo.DONHANG SET MATX=@MATX WHERE MADH=@MADH
            COMMIT TRAN
        END
```

T2:

EXEC NHAN_DONHANG_FIX 1, 2

Sau khi T1 và T2 thực hiện nhận đơn hàng (T1 đến trước):

MADH	MAKH	MADT	MATX	HINHTHUC...	NGAYTAO	DIACHIGH	PHISP	PHISHIP
1	1	1	1	NULL	2021-10-11 ...	36 Tran Binh...	NULL	NULL

Vì Taixe 1 đến trước nên Taixe 1 nhận được đơn hàng này.

Test case 9:

```
ALTER PROC DATHANG_FIX @MAKH BIGINT, @MASP BIGINT, @MADT BIGINT, @SLSP INT,
    @HINHTHUCTT VARCHAR(20),
    @DIACHIGH VARCHAR(100),
    @PHISHIP DECIMAL(15,2)
AS
    BEGIN TRAN
    DECLARE @CURRENTDATE AS DATETIME, @MADH AS BIGINT, @GIASP AS DECIMAL(15,2),
            @PHISP AS DECIMAL(15,2), @TONGTIEN AS DECIMAL(15,2)
    SELECT @GIASP=GIASP FROM dbo.QLSANPHAM WHERE MASP=@MASP AND MADT=@MADT
    SET @PHISP=@GIASP*@SLSP
    SET @TONGTIEN= @PHISP*@PHISHIP
    SET @CURRENTDATE=(Select GetDate())
    DECLARE @MADH1 BIGINT;
    EXEC dbo.TAO_DONHANG @MAKH = @MAKH, @MADT = @MADT, @MATX = NULL,
        @HINHTHUCTT = @HINHTHUCTT, @NGAYTAO = @CURRENTDATE,
        @DIACHIGH = @DIACHIGH, @PHISP = @PHISP,
        @PHISHIP = @PHISHIP, @TONGTIEN = @TONGTIEN,
        @TRANGTHAISHIP = 0, @TRANGTHAITTOAN = 0,
        @MADH = @MADH1 OUTPUT
    DECLARE @SLSPCC AS INT
    SELECT @SLSPCC= SLCUNGCAP FROM dbo.QLSANPHAM (UPDLOCK) WHERE MADT=@MADT AND MASP=@MASP
    WAITFOR DELAY '00:00:05'
    print (@SLSP)
    print (@SLSPCC)
    IF (@SLSP>@SLSPCC)
        rollback tran
    ELSE
        begin
            EXEC dbo.THEMSANPHAM @MADH = @MADH1, -- bigint
                @MASP = @MASP, -- bigint
                @GIASP = @GIASP, -- decimal(15, 2)
                @SLSP = @SLSP, -- int
                @MADT=@MADT
            COMMIT tran
        end
```

Số lượng của sản phẩm 1 của đối tác 1:

MADT	MASP	MACN	GIASP	SLCUNGCAP
1	1	1	9000.00	0
1	2	2	9000.00	9
2	1	3	9000.00	10
2	3	4	9000.00	10
3	4	5	18000.00	10
3	6	6	18000.00	10

T1: khách hàng đặt sản phẩm 1 của đối tác 1 số lượng 10

```
exec DATHANG_FIX 1,1,1,10,'CARD','QUAN 10',10000
```

T2: SET INSOLATION LEVEL READ COMMITTED

```
BEGIN TRAN
    SET TRAN ISOLATION LEVEL READ COMMITTED
    EXEC dbo.XEM_DS_DONHANG_DT @MADT=1 -- bigint
COMMIT TRAN
```

Bảng danh sách đơn hàng của giao tác T2 đọc được:

MADH	MAKH	MADT	MATX	HINHTHUCT	NGAYTAO	DIACHIGH	PHISP	PHISHIP	TONGTIEN	TRANGTHAISHIP	TRANG
1	1	1	1	NULL	2021-10-11 12:00:00.000	36 Tran Binh Trong, phuong 5, quan 5, Thanh pho H...	NULL	NULL	NULL	NULL	NULL
2	2	1	NULL	NULL	2021-10-11 12:00:00.000	265 Trung Chinh, phuong 3, quan Go Vap, Thanh p...	NULL	NULL	NULL	NULL	NULL
10	1	1	NULL	card	2021-11-21 13:20:04.000	Quan Cam	90000.00	10000.00	100000.00	0	0
12	1	1	NULL	card	2021-11-21 13:20:04.000	quan 1	9000.00	10000.00	19000.00	0	0
20	1	1	NULL	CARD	2021-12-19 00:28:53.157	QUAN BINH THANH T1	90000.00	10000.00	100000.00	0	0

Vì số lượng sản phẩm mua (10) lớn hơn số lượng hàng còn lại (0), nên dẫn đến T1 rollback (hủy tạo đơn hàng), dẫn đến T2 không đọc được đơn hàng đã hủy của T1 => chúng ta đã giải quyết được dirty read.

Test case 10:

T1: set isolation level serializable

```
CREATE PROC XEM_DS_TK_CAPNHAT
    @MATK BIGINT,
    @USERNAME CHAR(30),
    @PASS CHAR(30),
    @ISSTAFF BIT,
    @ISSUPPERUSER BIT,
    @SDT CHAR(15),
    @EMAIL CHAR(50),
    @DIACHI VARCHAR(100)
AS
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
BEGIN TRAN
    EXEC dbo.XEM_DS_TK
    EXEC dbo.CAPNHAT_TAIKHOAN @MATK, @USERNAME, @PASS, @ISSTAFF, @ISSUPPERUSER, @SDT, @EMAIL, @DIACHI
    WAITFOR DELAY '00:00:05'
    EXEC dbo.XEM_DS_TK
COMMIT
```

T2:


```

BEGIN TRAN
EXEC dbo.DANGKI_KH @USERNAME_KH = 'vodinhphuc', -- ch
    @PASS_KH = '123456', -- char(3
    @SDT_KH = '12345678', -- char
    @EMAIL_KH = 'dinhphuc@gmail.com',
    @DIACHI_KH = 'Quan binhthanh'
COMMIT

```

Kết quả lần đọc thứ nhất của T1:

MATK	USERNAME	PASS	ISSTAFF	ISSUPERUSER	SDT	EMAIL	DIACHI	ISACTIVE
1	admin1	admin1	0	1	0123456789	admin1@gmail.com	326 Bui Minh Truc, phuong 6, quan 8, Thanh pho Ho...	1
2	admin2	admin2	0	1	0123456789	admin2@gmail.com	268 Tran Hung Dao, phuong 6, quan 1, Thanh pho ...	1
3	nquangphu	123456	0	0	0123456789	nquangphu@gmail.com	123 Nguyen Trai, phuong 4, quan 5, Thanh pho Ho ...	1
4	lhoangphuc	123456	0	0	0123456789	lhoangphuc@gmail.com	60 Nguyen Thi Minh Khai, phuong 3, quan 1, Thanh ...	1
5	khachhang1	khach...	0	0	0123456789	khachhang1@gmail.co...	200 Nguyen Van Cu, phuong 4, quan 5, Thanh pho ...	1
6	khachhang2	khach...	0	0	0123456789	khachhang2@gmail.co...	190 An Duong Vuong, phuong 4, quan 5, Thanh pho ...	1
7	khachhang3	khach...	0	0	0123456789	khachhang3@gmail.co...	10 Dien Bien Phu, phuong 1, quan Binh Thanh, Tha...	1
8	khachhang4	khach...	0	0	0123456789	khachhang4@gmail.co...	426 Su Van Hanh, phuong 7, quan 10, Thanh pho H...	1
9	khachhang5	khach...	0	0	0123456789	khachhang5@gmail.co...	300 Nguyen Van Cu, phuong 4, quan 5, Thanh pho ...	1
10	khachhang6	khach...	0	0	0123456789	khachhang6@gmail.co...	62 Tran Phu, phuong 1, quan 5, Thanh pho Ho Chi ...	1
11	khachhang7	khach...	0	0	0123456789	khachhang7@gmail.co...	15 Vinh Vien, phuong 3, quan 3, Thanh pho Ho Chi ...	1
12	doitac1	doitac1	0	0	0123456789	doitac1@gmail.com	98 Ly Chinh Thang, phuong 10, quan Binh Thanh, T...	1
13	doitac2	doitac2	0	0	0123456789	doitac2@gmail.com	193 Nguyen Thi Minh Khai, phuong 3, quan 1, Than...	1
14	doitac3	doitac3	0	0	0123456789	doitac3@gmail.com	36 Tran Binh Trong, phuong 5, quan 5, Thanh pho H...	1
15	doitac4	doitac4	0	0	0123456789	doitac4@gmail.com	269 Le Hong Phong, phuong 2, quan 10, Thanh pho...	1
16	doitac5	doitac5	0	0	0123456789	doitac5@gmail.com	500 Ly Chinh Thang, phuong 10, quan Binh Thanh, ...	1
17	taixe1	taixe1	0	0	0123456789	taixe1@gmail.com	326 Bui Minh Truc, phuong 6, quan 8, Thanh pho Ho...	1
18	taixe2	taixe2	0	0	0123456789	taixe2@gmail.com	265 Truong Chinh, phuong 3, quan Go Vap, Thanh p...	1
19	taixe3	taixe3	0	0	0123456789	taixe3@gmail.com	44 Cach Mang Thang Tam, phuong 2, quan Tan Bin...	1
20	taixe4	taixe4	0	0	0123456789	taixe4@gmail.com	326/5 Bui Minh Truc, phuong 6, quan 8, Thanh pho ...	1
21	taixe5	taixe5	0	0	0123456789	taixe5@gmail.com	80 Ly Thai To, phuong 6, quan 5, Thanh pho Ho Chi ...	1
22	dinhphuc1...	277êž...	0	0	12345678	dinhphuc@gmail.com	Quan binhthanh	1

Lần 2:

MATK	USERNAME	PASS	ISSTAFF	ISSUPERUSER	SDT	EMAIL	DIACHI	ISACTIVE
1	phucccc	1234564	1	0	12345	phuc@gmail.commmmm	Quan 10	1
2	admin2	admin2	0	1	0123456789	admin2@gmail.com	268 Tran Hung Dao, phuong 6, quan 1, Thanh pho ...	1
3	nquangphu	123456	0	0	0123456789	nquangphu@gmail.co...	123 Nguyen Trai, phuong 4, quan 5, Thanh pho Ho ...	1
4	lhoangphuc	123456	0	0	0123456789	lhoangphuc@gmail.c...	60 Nguyen Thi Minh Khai, phuong 3, quan 1, Thanh ...	1
5	khachhang1	khachh...	0	0	0123456789	khachhang1@gmail.c...	200 Nguyen Van Cu, phuong 4, quan 5, Thanh pho ...	1
6	khachhang2	khachh...	0	0	0123456789	khachhang2@gmail.c...	190 An Duong Vuong, phuong 4, quan 5, Thanh pho ...	1
7	khachhang3	khachh...	0	0	0123456789	khachhang3@gmail.c...	10 Dien Bien Phu, phuong 1, quan Binh Thanh, Tha...	1
8	khachhang4	khachh...	0	0	0123456789	khachhang4@gmail.c...	426 Su Van Hanh, phuong 7, quan 10, Thanh pho H...	1
9	khachhang5	khachh...	0	0	0123456789	khachhang5@gmail.c...	300 Nguyen Van Cu, phuong 4, quan 5, Thanh pho ...	1
10	khachhang6	khachh...	0	0	0123456789	khachhang6@gmail.c...	62 Tran Phu, phuong 1, quan 5, Thanh pho Ho Chi ...	1
11	khachhang7	khachh...	0	0	0123456789	khachhang7@gmail.c...	15 Vinh Vien, phuong 3, quan 3, Thanh pho Ho Chi ...	1
12	doitac1	doitac1	0	0	0123456789	doitac1@gmail.com	98 Ly Chinh Thang, phuong 10, quan Binh Thanh, T...	1
13	doitac2	doitac2	0	0	0123456789	doitac2@gmail.com	193 Nguyen Thi Minh Khai, phuong 3, quan 1, Than...	1
14	doitac3	doitac3	0	0	0123456789	doitac3@gmail.com	36 Tran Binh Trong, phuong 5, quan 5, Thanh pho H...	1
15	doitac4	doitac4	0	0	0123456789	doitac4@gmail.com	269 Le Hong Phong, phuong 2, quan 10, Thanh pho...	1
16	doitac5	doitac5	0	0	0123456789	doitac5@gmail.com	500 Ly Chinh Thang, phuong 10, quan Binh Thanh, ...	1
17	taixe1	taixe1	0	0	0123456789	taixe1@gmail.com	326 Bui Minh Truc, phuong 6, quan 8, Thanh pho Ho...	1
18	taixe2	taixe2	0	0	0123456789	taixe2@gmail.com	265 Truong Chinh, phuong 3, quan Go Vap, Thanh p...	1
19	taixe3	taixe3	0	0	0123456789	taixe3@gmail.com	44 Cach Mang Thang Tam, phuong 2, quan Tan Bin...	1
20	taixe4	taixe4	0	0	0123456789	taixe4@gmail.com	326/5 Bui Minh Truc, phuong 6, quan 8, Thanh pho ...	1
21	taixe5	taixe5	0	0	0123456789	taixe5@gmail.com	80 Ly Thai To, phuong 6, quan 5, Thanh pho Ho Chi ...	1
22	dinhphuc1...	277êž...	0	0	12345678	dinhphuc@gmail.com	Quan binhthanh	1

Ta thấy mặc dù, T2 có insert thêm tài khoản những vì ta đã set mức cô lập là serializable nên tranh được phantom.

Test case 11:

T1: SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

```

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
BEGIN TRAN
    EXEC dbo.XEM_DSHD_DADUYET @MADT = 1 -- int
    WAITFOR DELAY '00:00:05'
    EXEC dbo.THONGBAO_GIAHAN @MADT = 1 -- bigint
COMMIT

```

T2:

```

BEGIN TRAN
    EXEC dbo.DUYET_HOPDONG @MADT = 1
COMMIT

```

Danh sách hợp đồng đã duyệt của đối tác 1 (T1 đọc lần 1):

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
6	1	3	2021-01-02	2022-01-02	10	1

Ta thấy ngày hết hạn của hợp đồng 6 của đối tác 1 cách hiện tại (19/12/2021) hơn 10 ngày nên bảng thông báo gia hạn của chúng ta mong muốn sẽ rỗng.

Danh sách thông báo gia hạn:

MAHD	MADT	SLCHINHANH	TGBD	TGKT	HOAHONG	ISACEPTED
------	------	------------	------	------	---------	-----------

Vậy ta đã giải quyết được unrepeatable read.

Testcase12:

Số lượng sản phẩm 1 của đối tác 2:

MADT	MASP	MACN	GIASP	SLCUNGCAP
1	1	1	9000.00	4
1	2	2	9000.00	9
2	1	3	9000.00	10
2	3	4	9000.00	10
3	4	5	18000.00	10
3	6	6	18000.00	10

T1: Khách hàng 1 đặt 15 sản phẩm 1 của đối tác 2 (Vì số lượng đặt hàng lớn hơn nên sau khi tạo đơn hàng thì T1 đã thực hiện rollback -> đơn hàng này bị hủy):

```

EXEC DATHANG_LOI 1,1,2,15,'CARD','QUAN 11', 10000

```

T2: (SET TRANSACTION ISOLATION LEVEL READ COMMITTED): Tài xế 1 xem danh sách đơn hàng trong khu vực (trùng khu vực của đối tác 2 nên sẽ thấy được đơn hàng của đối tác 2):

```

BEGIN TRAN
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED
    EXEC HIENTHI_DSDH 1
COMMIT

```

Kết quả T2 đọc được:

Results Messages						
MADH	MAKH	MADT	MATX	HINHTHUCTT	NGAYTAO	DIACHIGH
6	3	2	NULL	NULL	2021-10-11 12:00:00.000	44 Cach Mang Thang
9	6	2	NULL	NULL	2021-10-11 12:00:00.000	80 Ly Thai To, phuong

Kết quả sau khi T1 và T2 hoàn thành:

Results Messages						
MADH	MAKH	MADT	MATX	HINHTHUCTT	NGAYTAO	DIACHIGH
6	3	2	NULL	NULL	2021-10-11 12:00:00.000	44 Cach Mang Thang
9	6	2	NULL	NULL	2021-10-11 12:00:00.000	80 Ly Thai To, phuong