

BÁO CÁO ĐỒ ÁN THỰC HÀNH
MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU
GVHD: LƯƠNG HÁN CƠ

MỤC LỤC:

THÔNG TIN NHÓM	4
YÊU CẦU, PHÂN TÍCH ĐỒ ÁN.....	5
1. Yêu cầu đồ án:.....	5
2. Phần trăm hoàn thành:	5
BẢNG THIẾT KẾ THỰC THỂ KẾT HỢP, CSDL, PHÂN QUYỀN	5
1. Lược đồ thực thể kết hợp (mô hình ER):	5
2. Ràng buộc dữ liệu:	6
3. Mô hình dữ liệu quan hệ:	7
4. Xác định [login, user], role và permission:	7
XÁC ĐỊNH CHỨC NĂNG CỦA HỆ THỐNG VÀ CÁC TÌNH HUỐNG TRANH CHẤP	8
1. Các chức năng của hệ thống:	8
2. Thiết kế giao diện chức năng:	11
TÌNH HUỐNG TRANH CHẤP	25
1. Xác định các tình huống tranh chấp:.....	25
2. Cài đặt tình huống tranh chấp.....	26
a. Sinh viên thực hiện: Trịnh Hữu Hiệp (20120079).....	26
Tình huống 1: LOST UPDATE	26
Tình huống 2: UNREPEATABLE READ	32
b. Sinh viên thực hiện: Nguyễn Thị Hồng Nhung (20120093)	36
Tình huống 1: Lỗi Unrepeatable Read	36
Tình huống 2: Lỗi Conversion Deadlock.....	40
c. Sinh viên thực hiện : Trần Thái San (20120177).....	44
Tình huống 1: DIRTY READ	44
Tình huống 2: PHANTOM.....	45
d. Sinh viên thực hiện: Nguyễn Nhật Trường (20120229).....	48
Tình huống 1: UNREPEATABLE READ	48
Tình huống 2: DIRTY READ	50
Tình huống 3: CIRCLE DEADLOCK.....	53
XỬ LÝ TÌNH HUỐNG TRANH CHẤP	56

1. Sinh viên thực hiện : Trịnh Hữu Hiệp (20120079)	56
Tình huống 1: LOST UPDATE	56
Tình huống 2: UNREPEATABLE READ	62
2. Sinh viên thực hiện: Nguyễn Thị Hồng Nhung (20120093)	65
Tình huống 1: Lỗi Unrepeatable Read	65
Tình huống 2: Lỗi Conversion Deadlock	69
3. Sinh viên thực hiện: Trần Thái San	73
Tình huống 1: DIRTY READ	73
Tình huống 2: PHANTOM READ	75
4. Sinh viên thực hiện: Nguyễn Nhật Trường	77
Tình huống 1: UNREPEATABLE READ	77
Tình huống 2: Dirty Read	79
Tình huống 3: Circle Deadlock	81

THÔNG TIN NHÓM

STT	MSSV	Họ tên	Công việc	% Hoàn thành	% tham gia
1	20120079	Trịnh Hữu Hiệp	Thiết kế ER + RBTV+ Script cài đặt CSDL QH Xác định chức năng và cài procedure Xác định + mô tả lỗi Unrepeatable Read và Lost Update Xử lý 2 tranh chấp trên Thiết kế giao diện nhân viên, tổng hợp code	100%	25%
2	20120093	Nguyễn Thị Hồng Nhung	Thiết kế ER+ Mô hình CSDL QH Tạo script phân quyền và cài procedure Xác định + mô tả lỗi Unrepeatable Read và Conversion Deadlock Xử lý 2 tranh chấp trên Thiết kế giao diện tài xế, khách hàng	100%	25%
3	20120177	Trần Thái San	Xác định Role + tạo role Tạo script phân quyền và cài procedure Xác định + mô tả lỗi Dirty Read và Phantom Xử lý 2 tranh chấp trên Thiết kế giao diện admin, đăng nhập, đăng ký	100%	25%
4	20120229	Nguyễn Nhật Trường	Xác định + Tạo(User +Login) Cài procedure Xác định + mô tả lỗi Dirty Read, Unrepeatable Read, Circle Deadlock Xử lý 3 tranh chấp trên Thiết kế giao diện đối tác (cửa hàng)	100%	25%

YÊU CẦU, PHÂN TÍCH ĐỒ ÁN

1. Yêu cầu đồ án:

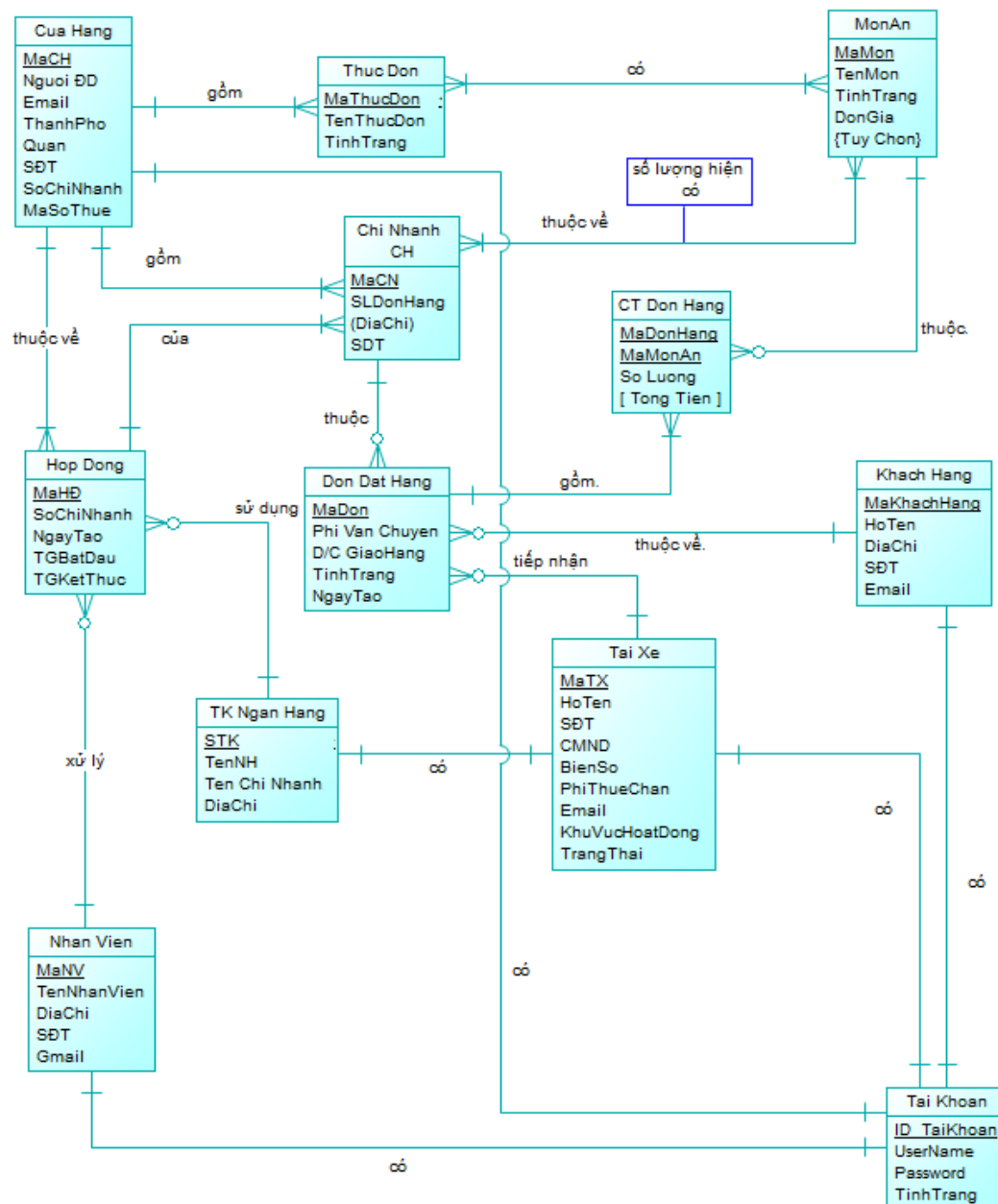
- Thiết kế cơ sở dữ liệu, phân tích quy trình nghiệp vụ và mô hình ER.
- Phân quyền người dùng.
- Thiết kế giao diện hệ thống, xây dựng các chức năng.
- Xác định các tình huống tranh chấp.
- Hướng giải quyết các tình huống tranh chấp.

2. Phần trăm hoàn thành:

Hoàn thành 100% yêu cầu đồ án

BẢNG THIẾT KẾ THỰC THỂ KẾT HỢP, CSDL, PHÂN QUYỀN

1. Lược đồ thực thể kết hợp (mô hình ER):



2. Ràng buộc dữ liệu:

<các ràng buộc toàn vẹn xác định được: miền giá trị, ràng buộc hệ thống>

Thời gian bắt đầu phải nhỏ hơn thời gian kết thúc trong hợp đồng

Trạng thái của tài xế phải là 'đã nhận đơn' hoặc null

Tình trạng của đơn hàng phải là chờ xác nhận hoặc đã xác nhận hoặc đang giao hoặc đã giao hoặc đã hủy

Số lượng món ăn trong chi tiết đơn hàng phải lớn hơn 0

Số lượng chi nhánh của cửa hàng phải lớn hơn 0

Số lượng đơn mỗi ngày trong mỗi chi nhánh phải lớn hơn 0

Tình trạng của món ăn phải là có bán hoặc hết hàng hôm nay hoặc tạm ngưng

Không thể hủy đơn hàng khi đơn hàng đang ở tình trạng là đã xác nhận hoặc đang giao hoặc đã giao

Tài xế chỉ tiếp nhận giao hàng trong khu vực đã đăng kí trước đó.

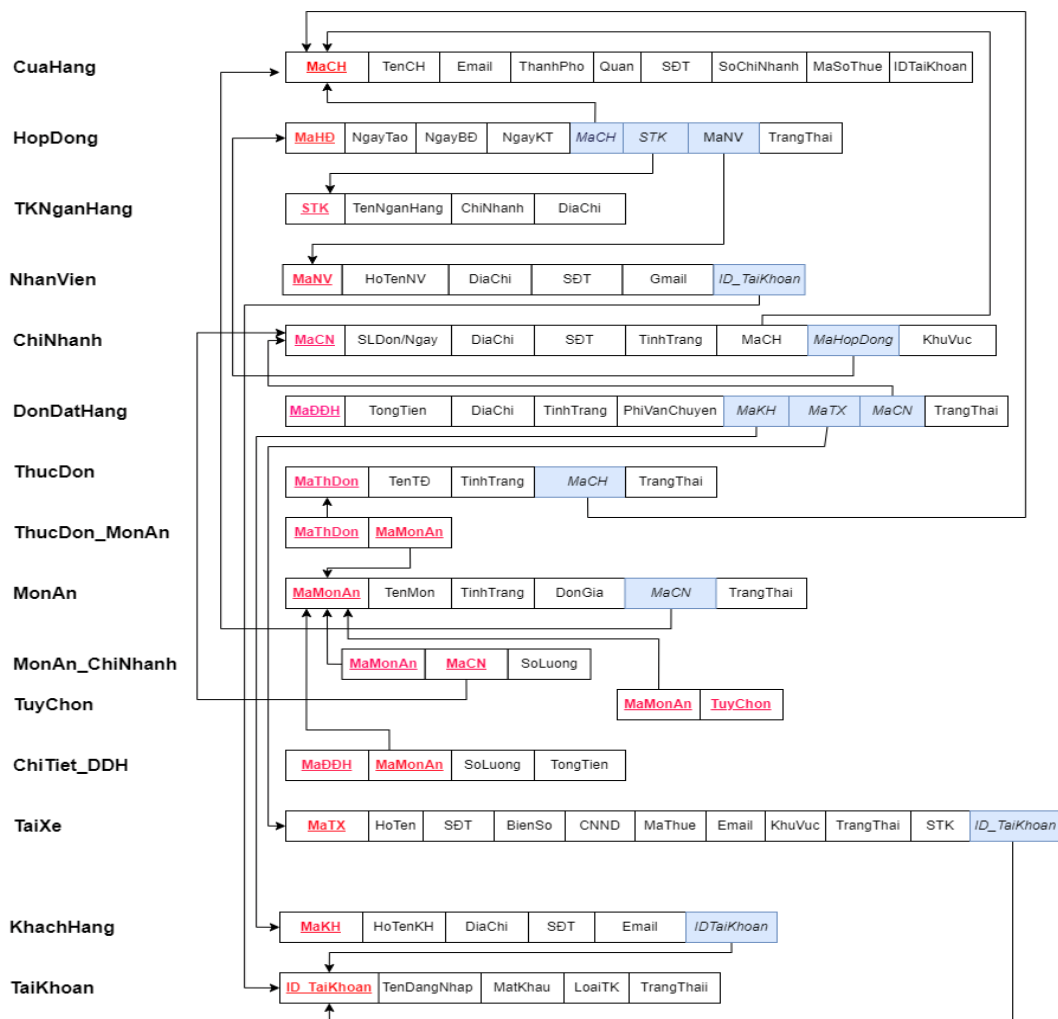
Tổng tiền trong CHITIET_DONHANG bằng số lượng* đơn giá của món ăn đó

Tổng tiền trong đơn đặt hàng bằng tổng tiền các chi tiết đơn hàng thuộc về đơn đặt hàng đó + phí vận chuyển.

Tất cả các tên món ăn phải phân biệt với nhau

Trạng thái của chi nhánh cửa hàng phải là 'bình thường' hoặc 'tạm nghỉ'

3. Mô hình dữ liệu quan hệ:



4. Xác định [login, user], role và permission:

a. Role:

Cửa Hàng, Tài Xế, Khách Hàng, Nhân Viên, Người Quản Trị

b. Phân quyền:

Cửa hàng được xem, xóa, chỉnh sửa, cập nhật thông tin của cửa hàng, thực đơn, món ăn

Cửa hàng được xem đơn đặt hàng, chi tiết đơn đặt hàng, thông tin tài xế, hợp đồng, thông tin khách hàng

Cửa hàng được phép cập nhật địa chỉ của chi nhánh

Cửa hàng được phép thêm chi nhánh

Tài xế được xem, xóa, chỉnh sửa, cập nhật thông tin của tài xế

Tài xế được xem thông tin của cửa hàng, đơn đặt hàng, chi tiết đơn đặt hàng, thông tin khách hàng

Tài xế được phép cập nhật tình trạng đơn hàng của mình

Khách hàng được xem, xóa, chỉnh sửa, cập nhật thông tin của khách hàng

Khách hàng được xem thông tin chi nhánh cửa hàng, thực đơn, món ăn, đơn đặt hàng, thông tin tài xế

Nhân Viên được phép xem thông tin, cập nhật hợp đồng, xem thông tin cửa hàng

Người Quản trị được phép thêm, xóa sửa, cập nhật thông tin của toàn bộ csdl

XÁC ĐỊNH CHỨC NĂNG CỦA HỆ THỐNG VÀ CÁC TÌNH HUỐNG TRANH CHẤP

1. Các chức năng của hệ thống:

a. Chức năng dùng chung:

STT	Chức năng	Mô tả hoạt động
ALL1	Đăng nhập	đăng nhập vào hệ thống đặt và giao thức ăn online dựa vào tài khoản và mật khẩu đăng kí.
ALL2	Đăng ký	Cung cấp thông tin, đăng ký tài khoản, sử dụng tài khoản đã tạo để đăng nhập
ALL3	Nhận thông báo	Nhận thông báo khi thao tác lỗi trên hệ thống
ALL4	Cập nhật tài khoản	Hệ thống hỗ trợ người dùng thay đổi và cập nhật lại các thông tin tài khoản
ALL5	Quên mật khẩu	Hỗ trợ tạo lại mật khẩu mới trong trường hợp quên mật khẩu khi đăng nhập
ALL6	Xem danh sách món ăn, cửa hàng	Hiện thị danh sách của hàng, các mon ăn, sản phẩm được đăng bán
ALL7	Xem chi tiết thông tin	Hiện thị thông tin của cửa hàng, hiện thị thông tin món ăn

b. Phân hệ đối tác:

STT	Chức năng	Mô tả hoạt động
AD1	Truy cập hợp đồng	Đối tác nhận được email và mã truy cập nhận hợp đồng được cung cấp từ hệ thống sau khi đã đăng kí thành công
AD2	Quản lý các chi nhánh cửa hàng	Xem thông tin các chi nhánh đã đăng kí và có quyền chỉnh sửa thông tin chi nhánh
AD3	Quản lý thực đơn	tìm kiếm điều chỉnh các thực đơn bao gồm thêm, xóa, cập nhật giá bán và tình trạng món ăn
AD4	Quản lý đơn đặt hàng	xem thông tin đơn hàng và cập nhật tình trạng của đơn hàng đã đặt của các chi nhánh
AD5	Xem chi tiết đơn hàng	Hiện thị chi tiết các đơn hàng của các chi nhánh của đối tác
AD6	Thống kê doanh thu	Hiện thị doanh thu theo tuần, tháng, năm của từng chi nhánh cửa hàng và tổng doanh thu của tất cả chi nhánh
AD7	Xem xu hướng món ăn	Hiện thị món ăn xu hướng, bán chạy nhất trong tuần hoặc trong tháng
AD8	Xem thông tin phản hồi	Hiện thị thông tin phản hồi của từng đơn hàng (nếu có) của khách hàng

c. Phân hệ khách hàng:

STT	Chức năng	Mô tả hoạt động
CTM1	Xem danh sách đối tác	Xem danh sách hiển thị đối tác, các chi nhánh của đối tác trong hệ thống
CTM2	Xem chi tiết danh sách món	Hiện thị toàn bộ món ăn cùng đơn giá, chi tiết món ăn của chi nhánh đối tác đã được chọn
CTM2	Tìm kiếm sản phẩm	tìm kiếm món ăn, đối tác bằng cách nhập tên món hoặc tên cửa hàng
CTM3	Chọn món	Chọn các món và thêm vào giỏ
CTM4	Quản lý giỏ hàng	Khách hàng thêm bớt các món ăn và giỏ, chọn số lượng món cùng các tùy chọn cho mỗi món
CTM5	Thanh toán	Chọn hình thức thanh toán, khách hàng tạo địa chỉ nhận hàng kèm số điện thoại và chọn 'đặt hàng' Hệ thống hỗ trợ người dùng thay đổi các thông tin về số điện thoại và địa chỉ nhận hàng
CTM4	Hủy đơn hàng	Chọn hủy đơn đặt hàng khi đơn hàng chưa được xác nhận từ phía chi nhánh cửa hàng Và hủy không thành công khi đơn hàng đã được tiếp nhận
CTM5	Theo dõi đơn hàng	Theo dõi quá trình vận chuyển, tài xế giao hàng của đơn hàng đã đặt

d. Phân hệ tài xế:

STT	Chức năng	Mô tả hoạt động
DRV1	Tạo TK đăng nhập	Cung cấp thông tin, admin thực hiện tạo tài khoản
DRV2	Xem danh sách đơn hàng	Hiện thị danh sách đơn hàng trong khu vực đã đăng kí của tài xế cần được giao
DRV3	Xem chi tiết đơn hàng	Hiện thị thông tin đơn hàng cần được giao

DRV4	Tiếp nhận và xử lý đơn hàng	Danh sách đơn hàng được hiển thị theo khu vực, tiếp nhận đơn hàng, cập nhật tình trạng 'đang giao' sau khi nhận đề khách hàng có thể theo dõi đơn hàng đã đặt
DRV5	Theo dõi thu nhập	Hiển thị danh sách các đơn hàng đã được nhận, phí vận chuyển từng đơn đã nhận

e. Phân hệ nhân viên:

STT	Chức năng	Mô tả hoạt động
NV1	Gia hạn hợp đồng	Thực hiện gia hạn hợp đồng, điều chỉnh ngày kết thúc
NV2	Xem danh sách hợp đồng	Hệ thống hiển thị danh sách hợp đồng đã được ký
NV3	Tìm kiếm hợp đồng	Tìm kiếm hợp đồng trên hệ thống, lọc các hợp đồng theo ngày, yêu cầu gửi thông báo khi sắp hết hạn hợp đồng
NV4	Duyệt hợp đồng	Hiển thị các hợp đồng mới cần ký kết, nhân viên thực hiện phê duyệt

f. Phân hệ admin:

STT	Chức năng	Mô tả hoạt động
AD1	Tạo tài khoản cho người dùng và new admin	Tạo tài khoản đăng nhập sẵn theo quy định cho người dùng và new admin
AD2	Xem thông tin danh sách tài khoản	Admin xem toàn bộ danh sách các tài khoản trên hệ thống
AD3	Thêm, xóa, sửa tài khoản	Admin thêm mới tài khoản, cập nhật, chỉnh sửa các tài khoản của người dùng
AD4	Khóa tài khoản	Admin có quyền khóa tài khoản của người dùng và các admin khác
AD5	Cấp quyền người dùng	Admin thao tác trên giao diện hệ thống, giới hạn quyền người dùng

2. Thiết kế giao diện chức năng:

a. Các chức năng chung:

Giao diện đăng nhập:

The login interface (ĐĂNG NHẬP) features a shopping cart illustration on the left with a 'BUY NOW' button. The login form on the right includes fields for Username (KFCTEST), Password (****), and a dropdown menu for 'Bạn là:' (Bạn là: Đối tác (Cửa hàng)). A 'Đăng nhập' button is located below the form. A link 'Đăng kí?' is provided for users who do not have an account.

The login interface (ĐĂNG NHẬP) shows a success message dialog box with the text 'Thông báo! Đăng nhập thành công!' (Thông báo! Đăng nhập thành công!). The dialog box has an 'OK' button. The background shows the same login form as the previous screenshot, but the 'Đăng nhập' button is highlighted.

Giao diện đăng ký (dành cho đối tác (cửa hàng), khách hàng, tài xế

Đăng kí

— □ ×

ĐĂNG KÝ

KHÁCH HÀNG

TÀI XẾ

ĐỐI TÁC(CỬA HÀNG)

ĐĂNG NHẬP

DKI_KH

— □ ×

KHÁCH HÀNG

Nhập thông tin cá nhân

Họ tên: Địa chỉ:

Số điện thoại: Email:

Quay lại **Tiếp tục**

DKi_TX

— □ ×

TÀI XẾ

Nhập thông tin cá nhân

Họ tên: Email: SDT: Biển số: CMND: Khu vực hoạt động: Mã thuế:

Quay lại

Tiếp tục

DKi_CH

— □ ×

ĐỐI TÁC

Nhập thông tin cá nhân

Tên CH: Email: TP: Quận: Số Chi Nhánh: Mã số thuế: SDT: Người đại diện:

Quay lại

Tiếp tục

b. Phân hệ đối tác (cửa hàng):

Giao diện xem danh sách hợp đồng

Cửa Hàng

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE

Mã cửa hàng: CH001

Tên cửa hàng: KFC

DS Hợp Đồng DS Chi Nhánh DS Sản Phẩm Đơn đặt hàng Cập nhật thông tin

Mã HĐ:

Mã HĐ	Mã CH	Ngày tạo	TG bắt đầu	TG kết thúc	STK	Mã nhân viên	Trạng thái
HD001	CH001	01/01/2018	02/01/2018	02/01/2023	101264783...	NV001	1

Giao diện xem danh sách chi nhánh và thêm chi nhánh mới:

Cửa Hàng

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE

Mã cửa hàng: CH001

Tên cửa hàng: KFC

DS Hợp Đồng DS Chi Nhánh DS Sản Phẩm Đơn đặt hàng Cập nhật thông tin

Mã cửa hàng: SDT:

Mã chi nhánh: Mã hợp đồng:

Địa chỉ: Khu vực:

Mã CN	SL đơn mỗi ngày	Địa chỉ	SDT	Tình trạng	Mã HĐ	Khu vực
CN001	100	227 NGUYỄN...	090505005		HD001	TPHCM
CN002	90	1 NGUYỄN ...	0987654321		HD001	TPHCM
CN010		45 Nhật Trườ...	0382112125		HD001	Bình Dương

Giao diện xem danh sách sản phẩm, cập nhật giá và số lượng:

Cửa Hàng

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE

Mã cửa hàng: CH001
Tên cửa hàng: KFC

DS Hợp Đồng DS Chi Nhánh DS Sản Phẩm Đơn đặt hàng Cập nhật thông tin

Mã sản phẩm:	Mã Món	Tên món	Tình	Đơn giá	Chi	Số
MA001	Ấn	món	Trạng		nhánh	lượng
Tên sản phẩm:	MA001	TEST1	có bán	55555	CN001	130
TEST1						
Số lượng thêm:						
0						
Đơn giá:						
55555						
Chi Nhánh:						
CN001						
CẬP NHẬT GIÁ						
CẬP NHẬT SỐ LƯỢNG						

ĐĂNG XUẤT

Giao diện xem danh sách đơn đặt hàng:

Cửa Hàng

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE

Mã cửa hàng: CH001
Tên cửa hàng: KFC

DS Hợp Đồng DS Chi Nhánh DS Sản Phẩm Đơn đặt hàng Cập nhật thông tin

DANH SÁCH ĐƠN ĐẶT HÀNG

Mã ĐH	Mã khách hàng	Tình trạng ĐH	Phí vận chuyển	Tổng tiền
DH001	KH001		30000	
DH002	KH002		40000	
DH003	KH002		30000	

ĐĂNG XUẤT

Giao diện cập nhật thông tin của cửa hàng:

Cửa Hàng

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE

Mã cửa hàng: CH001
Tên cửa hàng: KFC

DS Hợp Đồng DS Chi Nhánh DS Sản Phẩm Đơn đặt hàng **Cập nhật thông tin**

Mã cửa hàng:	CH001	NGƯỜI ĐẠI DIỆN
Tên cửa hàng:	KFC	Tên NDD: RONALDO
SĐT:	1012647837	
Quận:	QUẬN 6	
Thành phố (Tỉnh):	TPHCM	
Số chi nhánh	4	
Email:	KFC@GMAIL.COM	
MST:	MT001	

Cập nhật

ĐĂNG XUẤT

c. Phân hệ khách hàng:

Giao diện xem thông tin và chỉnh sửa thông tin tài khoản khách hàng:

Khách Hàng

THÔNG TIN KHÁCH HÀNG

Họ tên	
Số điện thoại	
Địa chỉ	
Email	

CHỈNH SỬA **HOÀN THÀNH**

Thông Tin Tài Khoản
Cập Nhật Mật Khẩu
Đặt Hàng
Đơn Hàng
Đăng Xuất

Giao diện cập nhật lại mật khẩu:

Khách Hàng

CẬP NHẬT MẬT KHẨU

Tên Đăng Nhập

Mật Khẩu Hiện Tại

Mật Khẩu Mới

Xác Nhận Mật Khẩu

CẬP NHẬT

Thông Tin Tài Khoản

Cập Nhật Mật Khẩu

Đặt Hàng

Đơn Hàng

Đăng Xuất

Giao diện đặt hàng:

Khách Hàng

DANH SÁCH ĐỐI TÁC

Mã Cửa Hàng

Tên Đối Tác

Địa Chỉ

XEM MÓN ĂN

LOAD

Thông Tin Tài Khoản

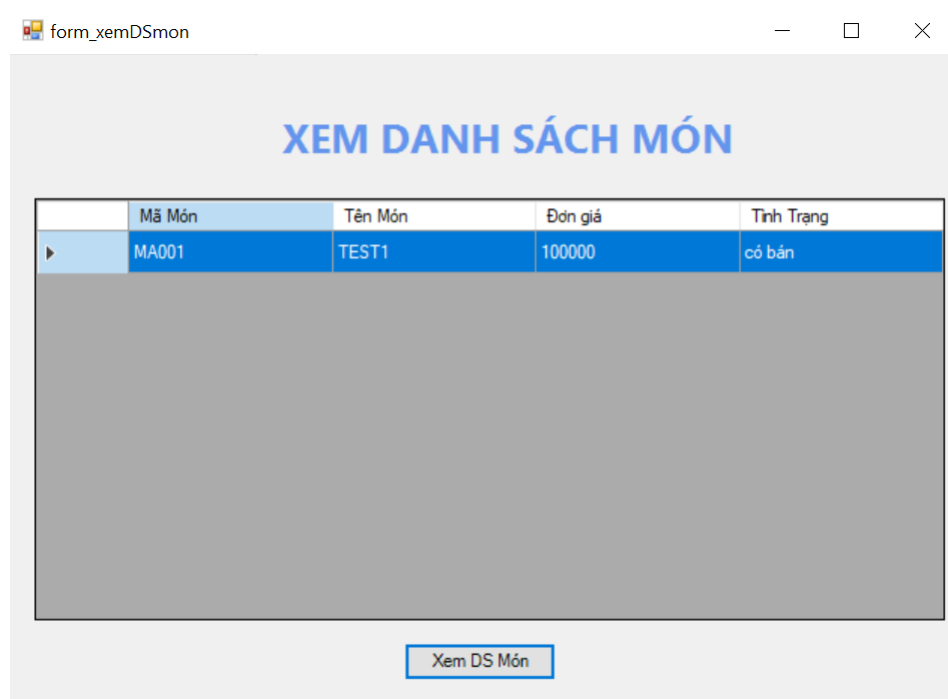
Cập Nhật Mật Khẩu

Đặt Hàng

Đơn Hàng

Đăng Xuất

Giao diện xem danh sách món:



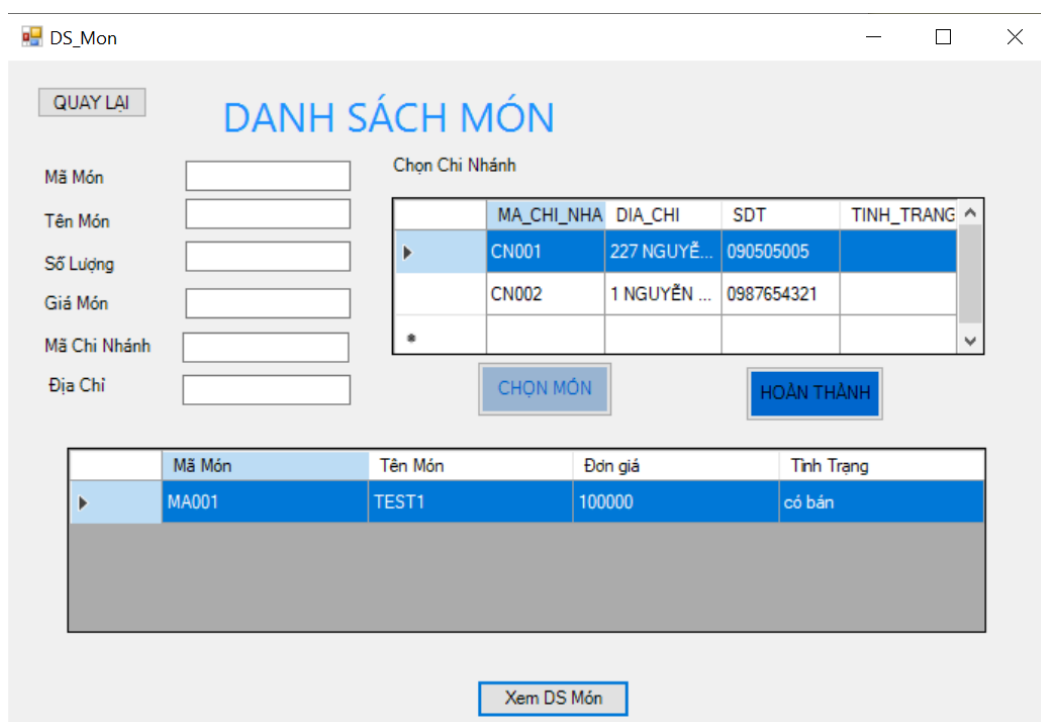
form_xemDSmon

XEM DANH SÁCH MÓN

	Mã Món	Tên Món	Đơn giá	Tình Trạng
▶	MA001	TEST1	100000	có bán

Xem DS Món

Giao diện chọn món và đặt hàng:



DS_Mon

QUAY LẠI

DANH SÁCH MÓN

Mã Món:

Tên Món:

Số Lượng:

Giá Món:

Mã Chi Nhánh:

Địa Chỉ:

Chọn Chi Nhánh

	MA_CHI_NHA	DIA_CHI	SDT	TINH_TRANG
▶	CN001	227 NGUYỄN...	090505005	
	CN002	1 NGUYỄN ...	0987654321	
*				

CHON MÓN

HOÀN THÀNH

	Mã Món	Tên Món	Đơn giá	Tình Trạng
▶	MA001	TEST1	100000	có bán

Xem DS Món

Giao diện xem đơn hàng đã đặt, hủy đơn:

Khách Hàng

ĐƠN HÀNG ĐÃ ĐẶT

Đơn đã giao

	MA_DC	TONG_	DIA_CI	TINH_	PHI_VA	MA_KH	MA_TA	MA_CH	TRANG

Đơn đang giao

	MA_DC	TONG_	DIA_CI	TINH_	PHI_VA	MA_KH	MA_TA	MA_CH	TRANG

Đơn đang chờ xác nhận

	MA_DC	TONG_	DIA_CI	TINH_	PHI_VA	MA_KH	MA_TA	MA_CH	TRANG
▶	DH004	1000...	quản...	chờ x...	0	KH001		CN002	1
	DH005	1000...		chờ x...	0	KH001		CN001	1

HỦY ĐƠN
XEM CHI TIẾT ĐƠN

Thông Tin Tài Khoản
 Cập Nhật Mật Khẩu
 Đặt Hàng
Đơn Hàng
 Đăng Xuất

d. Phân hệ tài xế

Giao diện xem thông tin tài khoản:

Tài Xế

THÔNG TIN TÀI XẾ

Họ tên

CMND/CCCD

Số điện thoại

Địa chỉ

Email

Biển số

Khu vực hoạt động

CHỈNH SỬA
HOÀN THÀNH

Thông Tin Tài Khoản
 Cập Nhật Mật Khẩu
 Danh Sách Đơn
 Đơn Đã Nhận
 Đăng Xuất

Giao diện cập nhật mật khẩu:

The screenshot shows a web application window titled 'Tài Xế'. On the left is a sidebar menu with five items: 'Thông Tin Tài Khoản', 'Cập Nhật Mật Khẩu' (highlighted in blue), 'Danh Sách Đơn', 'Đơn Đã Nhận', and 'Đăng Xuất'. The main content area is titled 'CẬP NHẬT MẬT KHẨU' in blue. It contains four input fields: 'Tên Đăng Nhập' with the value 'TX001', 'Mật Khẩu Hiện Tại' with '***', 'Mật Khẩu Mới' with '*****', and 'Xác Nhận Mật Khẩu' with '****'. A blue button labeled 'CẬP NHẬT' is positioned below the input fields.

Giao diện xem xem sách đơn để nhận đơn hàng:

The screenshot shows the 'Danh Sách Đơn' (Order List) interface. The sidebar menu is identical to the previous screen, with 'Danh Sách Đơn' highlighted in blue. The main content area is titled 'DANH SÁCH ĐƠN' in blue. It features five input fields for filtering: 'Mã Đơn', 'Địa Chỉ Giao', 'Tổng Tiền', 'Người Nhận', and 'Số Điện Thoại'. To the right of these fields are two buttons: 'NHẬN ĐƠN' and 'LOAD'. Below the input fields, the text 'Danh Sách Đơn Có Thể Nhận' is displayed above a large, empty gray rectangular box intended for the order list.

Giao diện xem đơn hàng đã đặt:

Tài Xế

Thông Tin Tài Khoản

Cập Nhật Mật Khẩu

Danh Sách Đơn

Đơn Đã Nhận

Đăng Xuất

ĐƠN HÀNG ĐÃ NHẬN

Họ Tên

Số Điện Thoại

Tình Trạng

Địa Chỉ Giao

Phí Vận Chuyển

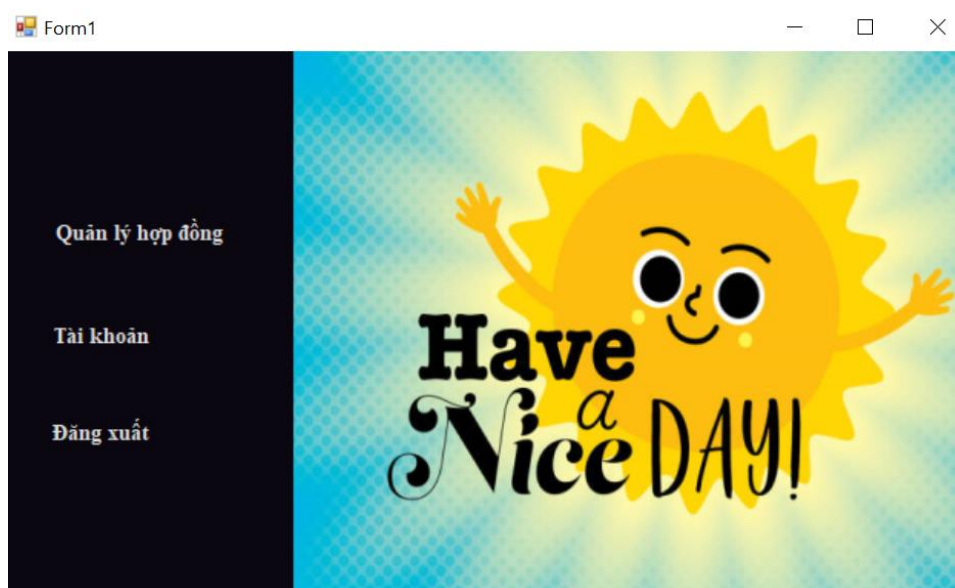
Tổng Chi Phí

CẬP NHẬT TÌNH TRẠNG

XEM CHI TIẾT ĐƠN

e. Phân hệ nhân viên

Giao diện trang chủ của nhân viên:



Giao diện quản lý hợp đồng (xem danh sách các hợp đồng, tìm kiếm hợp đồng để gia hạn):

QuanLyHopDong

Gia hạn Duyệt Tra cứu

Mã Hợp Đồng: ☐ Chọn ngày hết hạn mới 11 December 2022

*Note: Hệ thống hiển thị tất cả hợp đồng khi Mã hợp đồng trống
*Note: click vào hợp đồng để chọn trước khi thực hiện gia hạn

☐ Nhập số ngày gia hạn 0

MA HOP DONG	NGAY TAO	TG BAT DAU	TG KET THUC	MA CUA HANG	STK	MA NHAN VIEN	TRANG THAI
HD001	01/01/2018	02/01/2018	02/01/2023	CH001	1012647838	NV001	1
HD002	01/07/2017	01/09/2017	01/09/2025	CH002	1012647839	NV002	1
HD003	01/01/2022	02/01/2022	11/11/2022	CH003	1012647840	NV003	1

Giao diện quản lý hợp đồng (duyet hợp đồng):

QuanLyHopDong

Gia hạn Duyệt Tra cứu

☐ Sort dữ liệu Bắt đầu 06 December 2017 Kết thúc 11 December 2022 *Note: click vào hợp đồng để chọn trước khi thực hiện gia hạn

☐ Tìm kiếm thông tin Mã cửa hàng Tên cửa hàng

MA HOP DONG	NGAY TAO	TG BAT DAU	TG KET THUC	MA CUA HANG	STK	MA NHAN VIEN	TRANG THAI
-------------	----------	------------	-------------	-------------	-----	--------------	------------

Giao diện quản lý hợp đồng (tra cứu các chi nhánh trong hợp đồng):

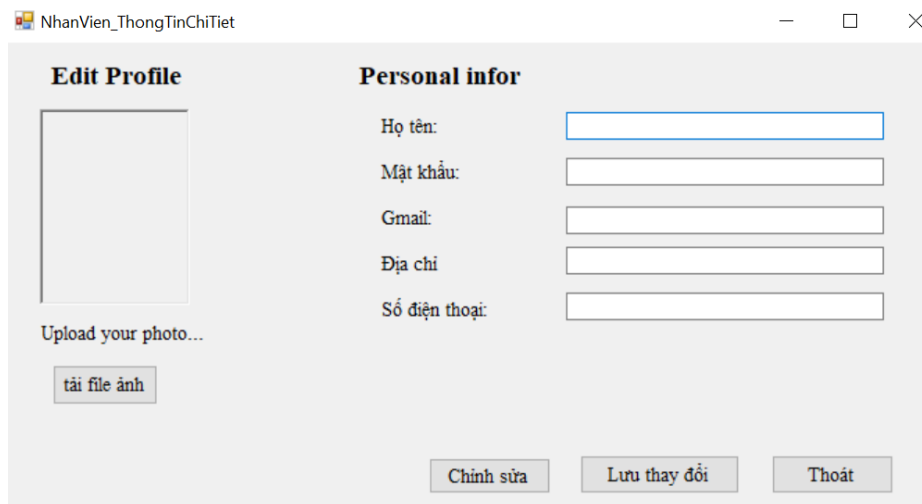
QuanLyHopDong

Gia hạn Duyệt Tra cứu

Thông tin chi nhánh trong hợp đồng Mã chi nhánh

MA HOP DONG	MA CUA HANG	TEN CUA HANG	MA CHI NHANH	SĐT	DIA CHI	KHUVUC
HD001	CH001	KFC	CN001	090505005	227 NGUYỄN VĂN CỨ	TPHCM
HD001	CH001	KFC	CN002	0987654321	1 NGUYỄN THỊ MINH	TPHCM
HD002	CH002	TEST	CN003	095505005	227 NGUYỄN VĂN LI	BÌNH DƯƠNG
HD002	CH002	TEST	CN004	090555005	22 TRƯỜNG CHÍNH	BÌNH DƯƠNG
HD003	CH003	L7U CHAY	CN005	090507005	ĐƯỜNG VÀNH ĐAI	BÌNH DƯƠNG
HD003	CH003	L7U CHAY	CN006	090505345	ĐƯỜNG SỐ 8	BÌNH DƯƠNG
HD003	CH003	L7U CHAY	CN007	090505567	SAU KTX KHU B ĐH	BÌNH DƯƠNG
HD001	CH001	KFC	CN010	0382112125	45 Nhật Trường	Bình Dương

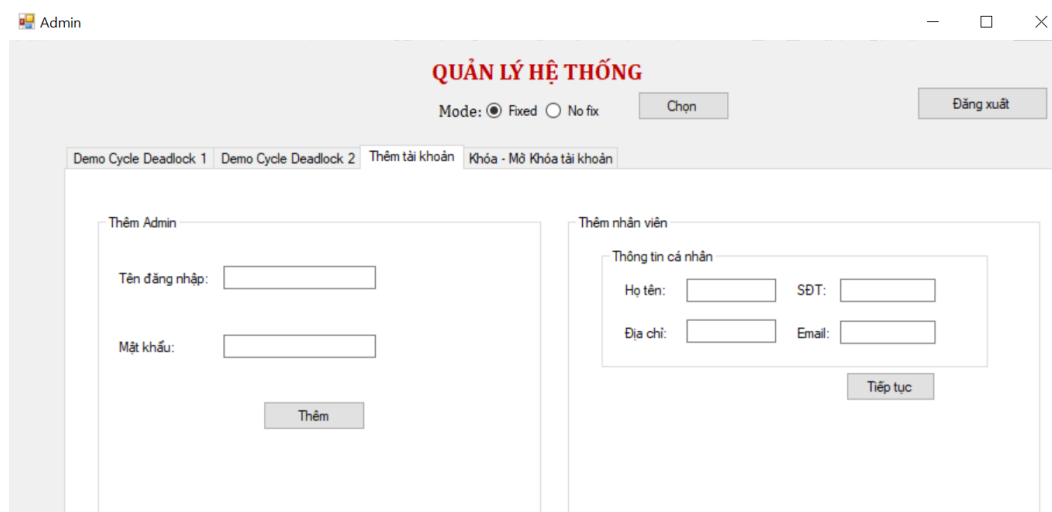
Giao diện xem thông tin của tài khoản nhân viên:



The screenshot shows a window titled 'NhanVien_ThongTinChiTiet'. It is divided into two main sections: 'Edit Profile' on the left and 'Personal infor' on the right. The 'Edit Profile' section contains a placeholder for a photo with the text 'Upload your photo...' and a button labeled 'tải file ảnh'. The 'Personal infor' section contains five input fields: 'Họ tên:', 'Mật khẩu:', 'Gmail:', 'Địa chỉ', and 'Số điện thoại:'. At the bottom of the window, there are three buttons: 'Chỉnh sửa', 'Lưu thay đổi', and 'Thoát'.

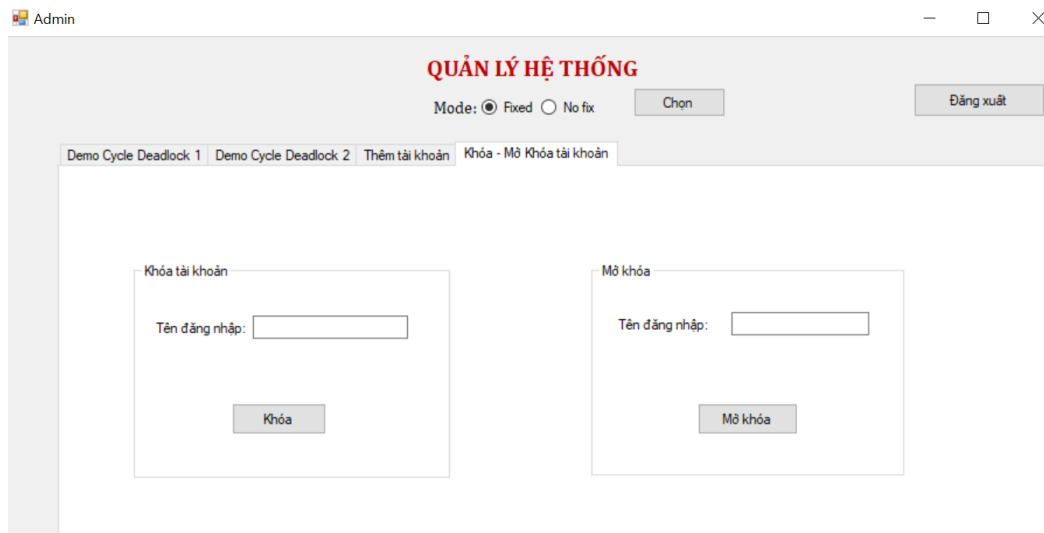
f. Phân hệ admin

Giao diện thêm tài khoản:



The screenshot shows a window titled 'Admin'. At the top, it says 'QUẢN LÝ HỆ THỐNG'. Below this, there is a 'Mode:' section with radio buttons for 'Fixed' (selected) and 'No fix', and a 'Chọn' button. To the right is a 'Đăng xuất' button. Below these are two tabs: 'Thêm tài khoản' (selected) and 'Khóa - Mở Khóa tài khoản'. The 'Thêm tài khoản' tab is divided into two sub-sections: 'Thêm Admin' and 'Thêm nhân viên'. The 'Thêm Admin' section has two input fields: 'Tên đăng nhập:' and 'Mật khẩu:', with a 'Thêm' button below them. The 'Thêm nhân viên' section has a 'Thông tin cá nhân' sub-section with four input fields: 'Họ tên:', 'SDT:', 'Địa chỉ:', and 'Email:', and a 'Tiếp tục' button below them.

Giao diện khóa, mở khóa tài khoản:



Admin

QUẢN LÝ HỆ THỐNG

Mode: ☒ Fixed ☐ No fix

Demo Cycle Deadlock 1 Demo Cycle Deadlock 2 Thêm tài khoản Khóa - Mô Khóa tài khoản

Khóa tài khoản

Tên đăng nhập:

Mô khóa

Tên đăng nhập:

Giao diện đổi mật khẩu và cập nhật thông tin:



Admin

QUẢN LÝ HỆ THỐNG

Mode: ☒ Fixed ☐ No fix

Demo Cycle Deadlock 1 Demo Cycle Deadlock 2 Thêm tài khoản Khóa - Mô Khóa tài khoản

Đổi mật khẩu

Username:

Password:

Cập nhật thông tin

Mã TX:

Email:

TÌNH HUỐNG TRANH CHẤP

1. Xác định các tình huống tranh chấp:

STT	Chức năng 1	Người dùng	Chức năng 2	Người dùng	Lỗi tranh chấp	Người thực hiện
1	Cập nhật số lượng món ăn tại 1 chi nhánh	Cửa hàng (Đối tác)	Thêm đơn đặt hàng	Khách Hàng	Lost Update	Trịnh Hữu Hiệp-20120079
2	Xem tổng tiền một món ăn	Khách hàng	Cập nhật đơn giá món ăn	Cửa hàng (đối tác)	Unrepeatable Read	
3	Đăng nhập	Khách hàng	Cập nhật mật khẩu	Khách Hàng	Unrepeatable Read	Nguyễn Thị Hồng Nhung-20120093
4	Gia hạn hợp đồng	Nhân viên	Gia hạn hợp đồng	Nhân viên	Conversion Deadlock	
5	Cập nhật giá món ăn	Đối tác	Xem giá món ăn	Khách hàng	Dirty Read	Trần Thái San-20120177
6	Thực hiện thêm 1 chi nhánh mới	Đối tác	Thực hiện xem chi nhánh của đối tác	Khách hàng	Phantom	
7	Xem tình trạng đơn hàng	Khách hàng	Cập nhật tình trạng đơn hàng	Tài xế	Unrepeatable Read	Nguyễn Nhật Trường-20120229
8	Cập nhật giá món	Đối tác	Xem giá món	Khách Hàng	Dirty Read	
9	Cập nhật thông tin khách hàng (Đổi mật khẩu), cập nhật thông tin tài xế (Email)	Người quản trị 1 (admin 1)	Cập nhật thông tin tài xế (Email), cập nhật thông tin khách hàng (Đổi mật khẩu),	Người quản trị 2 (admin 2)	Circle Deadlock	

2. Cài đặt tình huống tranh chấp

a. Sinh viên thực hiện: Trịnh Hữu Hiệp (20120079)

Tình huống 1: LOST UPDATE

ERR01: lost update T1 (User = cửa hàng (đối tác)): đối tác thực hiện nhập xuất số lượng món ăn, cập nhật số lượng món ăn tại một chi nhánh T2 (User = cửa hàng (đối tác)): đối tác thực hiện thêm mới đơn hàng (khi khách hàng mua tại chi nhánh)			
SP_CH_CAP_NHAT_SL_MON_AN Input: @MA_CHI_NHANH = 'CN001' @MA_MON_AN = 'MA001' @SO_LUONG_THEM = 100 Output: 1: cập nhật thành công 0: cập nhật thất bại		SP_KHACH_HANG_DAT_HANG_MON_AN Input: @MA_DON = 'DH001' @MA_MON_AN = 'MA001' @SO_LUONG = 10 @DIA_CHI = 'QUANG NAM' @PHI_VAN_CHUYEN = 10000 @MA_KHACH_HANG = 'KH001' @MA_TAI_XE = 'TX001' @MA_CHI_NHANH = 'CN001' Output: 1: cập nhật thành công 0: cập nhật thất bại	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN B1: Kiểm tra thông tin tồn tại của @MA_CHI_NHANH và @MA_MON_AN IF NOT EXISTS (SELECT * FROM MON_AN_CHI_NHANH WHERE MA_CHI_NHANH = @MA_CHI_NHANH AND MA_MON_AN = @MA_MON_AN) --(1) BEGIN PRINT @MA_CHI_NHANH + ', ' + @MA_MON_AN + N' KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0	S(MON_AN_CHI_NHANH) //Xin khoá đọc trên bảng MON_AN_CHI_NHANH //(xin thành công) Sau khi đọc xong thực hiện giải phóng khóa Do mức cô lập (RC)		

END			
B2: Kiểm tra @SO_LUONG_THEM IF @SO_LUONG_THEM <1 BEGIN PRINT N'SỐ LƯỢNG THÊM KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0 END	//kiểm tra @SO_LUONG_THEM hợp lệ		
DECLARE @SO_LUONG_HIEN_CO INT SET @SO_LUONG_HIEN_CO = (SELECT SELECT SO_LUONG FROM MON_AN_CHI_NHANH WHERE MA_CHI_NHANH = @MA_CHI_NHANH AND MA_MON_AN = @MA_MON_AN) BEGIN PRINT ' SỐ LƯỢNG HIỆN CÓ KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0 END	S(MON_AN_CHI_NHANH) //Xin khoá đọc trên bảng MON_AN_CHI_NHANH //(xin thành công) Sau khi đọc xong thực hiện giải phóng khóa		
WAITFOR DELAY '00:00:10'			
		BEGIN TRAN	
		B1.1: Kiểm tra thông tin tồn tại của @MA_KHACH_HANG ,@MA_TAI_XE, @MA_CHI_NHANH IF(@MA_KHACH_HANG NOT IN(SELECT MA_KHACH_HANG FROM KHACHHANG) OR @MA_TAI_XE NOT IN(SELECT MA_TAI_XE FROM TAI_XE) OR @MA_CHI_NHANH NOT IN (SELECT MA_CHI_NHANH FROM CHINHANH)) BEGIN PRINT N'KHÔNG TỒN TẠI ' + @MA_KHACH_HANG +N' HOẶC '+ @MA_TAI_XE +N' HOẶC '+ @MA_CHI_NHANH	S(KHACHHANG) S(TAI_XE) S(CHINHANH) //Xin khoá đọc trên bảng KHACHHANG,TAI_XE, CHINHANH (xin thành công) Sau khi đọc xong thực hiện giải phóng khóa Do mức cô lập (RC)

		<pre> ROLLBACK TRAN RETURN 0 END </pre>	
		<p>B1.2: Kiểm tra thông tin tồn tại của @MA_DON và @MA_MON_AN</p> <pre> IF (EXISTS(SELECT * FROM DONDATHANG DH WHERE DH.MA_DON = @MA_DON)) BEGIN PRINT @MA_DON + N' ĐÃ TỒN TẠI' ROLLBACK TRAN RETURN 0 END IF (@MA_MON_AN NOT IN (SELECT MA_MON_AN FROM MON_AN WHERE TINH_TRANG = N'có bán')) BEGIN PRINT @MA_MON_AN + N' KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(DON_DAT_HANG) //Xin khoá đọc trên bảng (xin thành công) DON_DAT_HANG đọc xong thực hiện giải phóng khóa(RC)</p> <p>S(MON_AN) //Xin khoá đọc trên bảng MON_AN (xin thành công) đọc xong thực hiện giải phóng khóa(RC)</p>
		<p>B2: Kiểm tra thông tin tồn tại @SO_LUONG</p> <pre> IF (@SO_LUONG > (SELECT A.SO_LUONG FROM MON_AN_CHI_NHANH A WHERE A.MA_CHI_NHANH = @MA_CHI_NHANH AND A.MA_MON_AN =@MA_MON_AN)) --(2) BEGIN PRINT N'SỐ LƯỢNG HIỆN TẠI CỦA CHI NHÁNH '+ @MA_CHI_NHANH+' KHÔNG ĐỦ ĐÁP ỨNG' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(MON_AN_CHI_NHANH) //Xin khoá đọc trên bảng MON_AN_CHI_NHANH (xin thành công) đọc xong thực hiện giải phóng khóa</p>

		<p>B3: lấy đơn giá món ăn , tạo biến tổng tiền trước khi insert</p> <pre>DECLARE @DON_GIA_MON FLOAT SET @DON_GIA_MON = (SELECT DON_GIA FROM MON_AN WHERE MA_MON_AN = @MA_MON_AN) DECLARE @TONG_TIEN FLOAT SET @TONG_TIEN = @DON_GIA_MON * @SO_LUONG</pre>	<p>S(MON_AN) //Xin khoá đọc trên bảng MON_AN (xin thành công) đọc xong thực hiện giải phóng khóa (mức cô lập RC)</p>
--	--	--	---

		<p>B4: insert dữ liệu vào bảng DONDATHANG VÀ CHITIET_DONHANH</p> <pre>-- INSERT DON DAT HÀNG INSERT INTO DONDATHANG VALUES(@MA_DON,@TONG_TIEN,@DIA_CHI,N 'chờ xác nhận',@PHI_VAN_CHUYEN, @MA_KHACH_HANG,@MA_TAI_XE,@MA_CHI_NH ANH,1) -- INSERT CHI TIẾT ĐƠN HÀNG INSERT INTO CHITIET_DONHANG VALUES(@MA_DON , @MA_MON_AN, @SO_LUONG, @TONG_TIEN)</pre>	<p>X(DONDATHANG) //Xin khoá ghi trên bảng DONDATHANG (xin thành công) Và giữ khóa đến hết giao tác do mức cô lập RC</p> <p>X(CHITIET_DON_HANG) //Xin khoá ghi trên bảng CHITIET_DONHANG (xin thành công) Và giữ khóa đến hết giao tác do mức cô lập RC</p>
--	--	--	---

		<p>B5: cập nhật số lượng món ăn tại chi nhánh khách hàng mua hàng ('CN001')</p> <pre> UPDATE MON_AN_CHI_NHANH SET SO_LUONG = SO_LUONG - @SO_LUONG --(3) WHERE @MA_CHI_NHANH = MA_CHI_NHANH AND @MA_MON_AN = MA_MON_AN </pre>	<p>X(MON_AN_CHI_NHANH) //Xin khoá GHI trên bảng MON_AN_CHI_NHANH (xin thành công) Và giữ khóa đến hết giao tác do mức cô lập RC</p>
		<pre> COMMIT TRAN PRINT N'CẬP NHẬT THÀNH CÔNG' RETURN 1 </pre>	
<p>B3:cập nhật số lượng món ăn tại chi nhánh</p> <pre> UPDATE MON_AN_CHI_NHANH SET SO_LUONG = @SO_LUONG_HIEN_CO + @SO_LUONG_THEM WHERE MA_CHI_NHANH = @MA_CHI_NHANH AND MA_MON_AN = @MA_MON_AN </pre>	<p>X(MON_AN_CHI_NHANH) //Xin khoá ghi trên bản MON_AN_CHI_NHANH (xin thành công) Giữ đến hết giao tác Do T2 đã commit và nhả khóa X</p>		
<pre> COMMIT TRAN PRINT N'CẬP NHẬT THÀNH CÔNG' RETURN 1 </pre>			

Tình huống 2: UNREPEATABLE READ

ERR01: UNREPEATABLE READ T1 (User = cửa hàng(đối tác)): đối tác thực hiện cập nhật đơn giá của món ăn (@MA_MON_AN = 'MA001') T2 (User = khách hàng): xem tổng tiền của một món ăn (@MA_MON_AN = 'MA001') với số lượng nhất định			
SP_KH_XEM_TONG_TIEN_MOT_MON_AN	Khóa	SP_CH_CAP_NHAT_DON_GIA_MOT_MON_AN	Khóa
Input: @MA_CUA_HANG = 'CH001' @MA_MON_AN = 'MA001' @SOUONG = 10 Output: 1: hiển thị thành công 0: thất bại		Input: @MA_CUA_HANG = 'CH001' @MA_MON_AN = 'MA001' @NEW_DONGIA = 100000 Output: 1: cập nhật thành công 0: cập nhật thất bại	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: kiểm tra @MA_CUA_HANG tồn tại <pre>IF(@MA_CUA_HANG NOT IN (SELECT MA_CUA_HANG FROM CUAHANG)) BEGIN PRINT @MA_CUA_HANG + N' KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0 END</pre>	S(CUAHANG) //Xin khoá đọc trên bảng (xin thành công) đọc xong thực hiện giải phóng khoá (mức cô lập RC)		
B1: kiểm tra @MA_MON_AN tồn tại và thuộc về của hàng 'CH001' <pre>IF(NOT EXISTS (SELECT * FROM MON_AN MA WHERE MA.MA_MON_AN = @MA_MON_AN AND MA.TINH_TRANG = N'có bán' AND MA.TRANG_THAI = 1 AND MA.MA_CUA_HANG = @MA_CUA_HANG)) BEGIN</pre>	S(MON_AN) //Xin khoá đọc trên bảng MON_AN (xin thành công)		

<pre>PRINT @MA_MON_AN + N' KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0 END</pre>	đọc xong thực hiện giải phóng khóa (mức cô lập RC)		
WAITFOR DELAY '00:00:15'			
		BEGIN TRAN	
		<pre>IF (@MA_CUA_HANG NOT IN (SELECT MA_CUA_HANG FROM CUAHANG)) BEGIN PRINT @MA_CUA_HANG + N' KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0 END</pre>	S(CUAHANG) // Xin khoá đọc trên bảng CUAHANG (xin thành công) đọc xong thực hiện giải phóng khóa (do mức cô lập RC)
		<pre>IF (NOT EXISTS (SELECT * FROM MON_AN MA WHERE MA.MA_MON_AN = @MA_MON_AN AND TRANG_THAI !=0 AND MA.MA_CUA_HANG = @MA_CUA_HANG)) BEGIN PRINT @MA_MON_AN + N' KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0 END --(1)</pre>	S(MON_AN) // Xin khoá đọc trên bảng MON_AN (xin thành công) đọc xong thực hiện giải phóng khóa (do mức cô lập RC)

		<pre> UPDATE MON_AN SET DON_GIA = @NEW_DONGIA WHERE MA_MON_AN = @MA_MON_AN1 </pre>	<p>X(MON_AN) //Xin khoá ghi trên bảng MON_AN (xin thành công) Giữ khóa đến hết giao tác</p>
		<pre> COMMIT TRAN PRINT N'CẬP NHẬT THÀNH CÔNG' RETURN 1 </pre>	
<pre> SELECT @SO_LUONG* DON_GIA FROM MON_AN WHERE @MA_MON_AN = MA_MON_AN </pre>	<p>X(MON_AN) //Xin khoá ghi trên bảng MON_AN (xin thành công do T2 đã commit và nhả khóa ghi trên bảng MON_AN) Giữ khóa đến hết giao tác</p>		
<pre> COMMIT TRAN PRINT N'HIỂN THỊ THÀNH CÔNG' RETURN 1 </pre>			

b. Sinh viên thực hiện: Nguyễn Thị Hồng Nhung (20120093)

Tình huống 1: Lỗi Unrepeatable Read

ERR01: UNREPEATABLE READ T1 (User = Khách hàng): khách hàng A thực hiện đăng nhập T2 (User = Khách hàng): ở thiết bị khác của khách hàng A lại cập nhật mật khẩu			
dang_nhap	Khóa	cap_nhat_mat_khau	Khóa
Input: tên đăng nhập (@ten_tk) Mật khẩu (@mat_khau) Output: nếu thành công: output ra table gồm: loại tài khoản, mã khách hàng		Input: tên đăng nhập (@ten_tk) Mật khẩu (@mat_khau) Mật khẩu mới (@mat_khau_moi) Output: nếu thành công mật khẩu mới được cập nhật, nếu không thành công rollback	
SET TRANSACTION ISOLATION LEVEL READ COMMITTED		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Kiểm tra tên đăng nhập có tồn tại không if not exists (select tk.TEN_DANG_NHAP from TAIKHOAN tk where tk.TEN_DANG_NHAP=@ten_tk) begin print N'Không tồn tại tên đăng nhập' rollback tran return -1 end	S(TAIKHOAN) //Xin khóa đọc trên bảng TAIKHOAN sau khi đọc xong thì giải phóng khóa do mức cô lập RC (Xin thành công)		
B2: Kiểm tra mật khẩu có đúng không if not exists (select tk.TEN_DANG_NHAP from TAIKHOAN tk where tk.TEN_DANG_NHAP=@ten_tk and tk.MAT_KHAU= @mat_khau) begin print N'Sai mật khẩu' rollback tran	S(TAIKHOAN) //Xin khóa đọc trên bảng TAIKHOAN sau khi đọc xong thì giải		

<pre> end return -1 </pre>	phóng khóa do mức cô lập RC (Xin thành công)		
<pre> declare @loai_tk int set @loai_tk = NULL SET @loai_tk = (SELECT TOP 1 tk.LOAI_TK FROM TAIKHOAN tk WHERE tk.TEN_DANG_NHAP = @ten_tk AND tk.MAT_KHAU = @mat_khau AND tk.TRANG_THAI = 1) </pre>	S(TAIKHOAN) //Xin khóa đọc trên bảng TAIKHOAN sau khi đọc xong thì giải phóng khóa do mức cô lập RC (Xin thành công)		
<pre> WAITFOR DELAY '00:00:10' </pre>			
		BEGIN TRAN	
		B1: Kiểm tra thông tin (1) if not exists (select tk.TEN_DANG_NHAP from TAIKHOAN tk where tk.TEN_DANG_NHAP=@ten_tk) begin print N'Sai tên đăng nhập' rollback tran return -1 end	S(TAIKHOAN) //Xin khóa đọc trên bảng TAIKHOAN (Xin thành công vì ở mức cô lập RC thì T1 đọc xong sẽ giải phóng khóa S nên T2 có thể xin được khóa S)
		B2: Thống kê Chuyển Xe if not exists (select tk.TEN_DANG_NHAP from TAIKHOAN tk where tk.TEN_DANG_NHAP=@ten_tk and tk.MAT_KHAU= @mat_khau) begin	S(TAIKHOAN) //Xin khóa đọc trên bảng TAIKHOAN (Xin thành công) như trên

		<pre> print N'Sai mật khẩu. Vui lòng nhập đúng mật khẩu để sửa đổi thông tin' rollback tran return -1 end </pre>	
		<pre> update TAIKHOAN set MAT_KHAU=@mat_khau_moi where TEN_DANG_NHAP= @ten_tk print N'Đổi mật khẩu thành công' end try begin catch print N'Thay đổi mật khẩu thất bại' rollback tran end catch </pre>	<p>X(TAIKHOAN) //Xin khóa ghi trên bảng TAIKHOAN(xin thành công cho T2 đến khi hoàn thành T2)</p>
		COMMIT TRAN	
<pre> IF @loai_tk IS NOT NULL begin if @loai_tk=0 begin SELECT distinct @loai_tk AS 'loai_tk', CH.MA_CUA_HANG AS 'ma' FROM CUAHANG CH WHERE CH.MA_CUA_HANG = @ten_tk COMMIT TRAN RETURN end else if @loai_tk=1 BEGIN SELECT distinct @loai_tk AS 'loai_tk', KH.MA_KHACH_HANG AS 'ma' </pre>	<p>S(TAIKHOAN) //Xin khóa đọc trên bảng TAIKHOAN sau khi đọc xong thì giải phóng khóa do mức cô lập RC và T2 đã COMMIT xong (Xin thành công)</p>		

```

FROM KHACHHANG KH, TAIKHOAN tk
WHERE KH.TEN_DANG_NHAP = @ten_tk and
tk.MAT_KHAU=@mat_khau
        COMMIT TRAN
        RETURN

    END
else if @loai_tk=2
    BEGIN
SELECT distinct @loai_tk AS 'loai_tk', TX.MA_TAI_XE
AS 'ma'
FROM TAIXE TX, TAIKHOAN tk
WHERE TX.TEN_DANG_NHAP = @ten_tk and
tk.MAT_KHAU=@mat_khau
        COMMIT TRAN
        RETURN

    END
    else if @loai_tk=3
    begin
select distinct @loai_tk as 'loai_tk',
NV.MA_NHAN_VIEN from NHANVIEN NV,
TAIKHOAN tk
where NV.TEN_DANG_NHAP=@ten_tk and
tk.MAT_KHAU=@mat_khau
        end
    else if @loai_tk=3
    begin
        declare @ad char(10)
        set @ad= 'admin'
select distinct @loai_tk as 'loai_tk',
tk.TEN_DANG_NHAP from TAIKHOAN tk
where tk.TEN_DANG_NHAP=@ten_tk and
tk.MAT_KHAU=@mat_khau
        end

    end
end

```

<pre> end try begin catch print N'Đăng nhập thất bại' rollback tran end catch commit tran go COMMIT TRAN </pre>			
---	--	--	--

Tình huống 2: Lỗi Conversion Deadlock

ERR02: CONVERSION DEADLOCK T1 (User = Nhân viên): nhân viên A thực hiện gia hạn hợp đồng của cửa hàng C T2 (User = Nhân viên): nhân viên B cũng thực hiện gia hạn hợp đồng của cửa hàng C			
Gia_han_hop_dong	Khóa	Gia_han_hop_dong	Khóa
<u>Input:</u> Mã hợp đồng A (@maHD), số lượng ngày thêm gia hạn hợp đồng (@so_ngay_them) <u>Output:</u> thời gian kết thúc được gia hạn thêm với số ngày gia hạn hợp đồng thêm		<u>Input:</u> Mã hợp đồng A(@maHD), số lượng ngày thêm gia hạn hợp đồng (@so_ngay_them) <u>Output:</u> thời gian kết thúc được gia hạn thêm với số ngày gia hạn hợp đồng thêm	
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE		SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
BEGIN TRAN			
B1: Kiểm tra mã Hợp đồng có tồn tại không if @maHD not in (select hd.MA_HOP_DONG from HOPDONG hd where hd.MA_HOP_DONG=@maHD) begin print N'Không tồn tại hợp đồng này' rollback tran end	S(HOPDONG) //Xin khóa đọc bảng HOPDONG (Xin thành công)		

B2: tạo biến @ngay_hien_tai và set bằng ngày cuối hợp đồng declare @ngay_hien_tai date set @ngay_hien_tai=(select top 1 hd.TG_KET_THUC from HOPDONG hd where hd.MA_HOP_DONG=@maHD)	S(HOPDONG) //Xin khóa đọc bảng HOPDONG (Xin thành công)		
if @ngay_hien_tai is not null begin -- bắt đầu chờ WAITFOR DELAY '00:00:15'			
		BEGIN TRAN	
		B1.1: Kiểm tra mã Hợp đồng có tồn tại không if @maHD not in (select hd.MA_HOP_DONG from HOPDONG hd where hd.MA_HOP_DONG=@maHD) begin print N'Không tồn tại hợp đồng này' rollback tran end	S(HOPDONG) //Xin khóa đọc bảng HOPDONG (Xin thành công vì 2 khóa S tương thích nhau nên sẽ được phát)
		B2.1: tạo biến @ngay_hien_tai và set bằng ngày cuối hợp đồng declare @ngay_hien_tai date set @ngay_hien_tai=(select top 1 hd.TG_KET_THUC from HOPDONG hd where hd.MA_HOP_DONG=@maHD)	S(HOPDONG) //Xin khóa đọc bảng HOPDONG (Xin thành công)
		if @ngay_hien_tai is not null begin	

		-- bắt đầu chờ WAITFOR DELAY '00:00:15'	
-- gia hạn ngày hợp đồng set @ngay_hien_tai = (select DATEADD(DAY, @so_ngay_them, @ngay_hien_tai)) -- cập nhật vào hợp đồng update HOPDONG SET TG_KET_THUC = @ngay_hien_tai WHERE MA_HOP_DONG = @maHD; end	X(HOPDONG) //Xin khóa ghi ở bảng HOPDONG để update gia hạn hợp đồng (Không thành công chờ T2 giải phóng khóa S thì T2 mới có thể xin khóa X để ghi)		
end try begin catch print N'Gia hạn hợp đồng thất bại' rollback tran end catch			
COMMIT TRAN			
		-- gia hạn ngày hợp đồng set @ngay_hien_tai = (select DATEADD(DAY, @so_ngay_them, @ngay_hien_tai)) -- cập nhật vào hợp đồng update HOPDONG SET TG_KET_THUC = @ngay_hien_tai WHERE MA_HOP_DONG = @maHD; end	X(HOPDONG) //Xin khóa ghi ở bảng HOPDONG để update gia hạn hợp đồng (Không thành công Chờ T1 giải phóng khóa S thì T2 mới xin được khóa X để ghi)
		end try begin catch print N'Gia hạn hợp đồng thất bại'	

		<code>rollback tran</code> <code>end catch</code> <code>COMMIT TRAN</code>	
--	--	--	--

c. Sinh viên thực hiện : Trần Thái San (20120177)

Tình huống 1: DIRTY READ

ERR03: DIRTY READ T1 (User = cửa hàng (đối tác)): thực cập nhật đơn giá món ăn T2 (User = khách hàng): khách hàng thực hiện xem thông tin món ăn			
SP_CAPNHAPGIAMONAN Input: @MAMONAN = 'MA001' @MA_CUA_HANG = 'CH001' @GIAMOI = 100000 Output: 1: cập nhật thành công 0: cập nhật thất bại		SP_XEMMONAN Input: @MA_MON_AN = 'MA001' Output: :	
SET TRANSACTION ISOLATION LEVEL		SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED	
BEGIN TRAN			
B1: Kiểm tra món ăn có tồn tại hay không IF NOT EXISTS(SELECT * FROM MON_AN WHERE @MAMONAN=MA_MON_AN AND @MACH=MA_CUA_HANG) BEGIN PRINT N'KHÔNG TỒN TẠI MÓN ĂN' ROLLBACK TRAN RETURN 1 END	S(MON_AN) //Xin khoá đọc trên bảng MON_AN với điều kiện MaCH = 'CH001' và MaMonAn = 'MA001' (xin thành công)		
B2: Thực hiện cập nhật giá mới UPDATE MON_AN SET DON_GIA=@GIAMOI WHERE @MAMONAN=MA_MON_AN AND @MACH=MA_CUA_HANG	U,X(MON_AN) //Xin khoá U,X trên bảng MON_AN với điều kiện @MAMONAN='MA001' @MACH = 'CH001' @GIAMOI =0		

	(xin thành công)		
WAITFOR DELAY '00:00:10'			
		BEGIN TRAN	
		Thực hiện xem món ăn của đối tác với MAMONAN = 'MA001' SELECT MA.MA_MON_AN, MA.TEN_MON, MA.DON_GIA, CH.TEN_CUA_HANG, MA.TINH_TRANG, MA.TRANG_THAI FROM MON_AN MA, CUAHANG CH WHERE MA.MA_MON_AN=@MAMONAN AND MA.MA_CUA_HANG=CH.MA_CUA_HANG	S(MON_AN) // Xin khoá đọc trên bảng MON_AN với điều kiện MA_MON_AN = 'MA001' (xin thành công)
		COMMIT TRAN	
B3: Kiểm tra giá mới có bằng 0 hay không, nếu bằng thì rollbacktran IF @GIAMOI=0 BEGIN ROLLBACK TRAN RETURN 1 END			
COMMIT TRAN			

Tình huống 2: PHANTOM

ERR03: PHANTOM T1 (User = cửa hàng (đối tác)): thực hiện thêm một chi nhánh mới T2 (User = khách hàng): thực hiện xem danh sách đối tác			
sp_DT_ThemChiNhanh		sp_KH_XemDSDoiTac	

<u>Input:</u> @machinhanh @diachi @sdt @macuahang @mahopdong @khuvuc <u>Output:</u> 1: thêm thành công 0: thêm thất bại		<u>Input:</u> <u>Output:</u> : 1: cập nhật thành công 0: cập nhật thất bại	
SET TRANSACTION ISOLATION LEVEL		SET TRANSACTION ISOLATION LEVEL	
		BEGIN TRAN	
		B1: Xem danh sách đối tác(cửa hàng) SELECT MA_CUA_HANG, TEN_CUA_HANG, EMAIL, THANH_PHO, QUAN, SDT, SO_CHI_NHANH FROM CUAHANG	S(CUAHANG) //Xin khóa đọc trên bảng CUAHANG (xin thành công)
		WAITFOR DELAY '00:00:10'	
BEGIN TRAN			
B1: Kiểm tra trùng lặp của @machinhanh và @diachi IF(EXISTS(SELECT * FROM CHINHANH WHERE MA_CUA_HANG = @macuahang AND DIA_CHI = @diachi)) begin rollback tran RETURN -1 end IF(EXISTS(SELECT * FROM CHINHANH WHERE MA_CHI_NHANH = @machinhanh)) begin rollback tran RETURN -1 end	S(CHI_NHANH) //Xin khoá đọc trên bảng CHI_NHANH với điều kiện @machinhanh = 'CH001' (xin thành công) S(CHI_NHANH) //Xin khoá đọc trên bảng CHI_NHANH với điều kiện @diachi = 'Dĩ An, Bình Dương' (xin thành công)		

B2: Thêm chi nhánh <code>INSERT INTO CHINHANH(MA_CHI_NHANH, SL_DON_MOINGAY, DIA_CHI, SDT, TINH_TRANG, MA_CUA_HANG, MA_HOP_DONG, KHUVUC) VALUES(@machinhanh, NULL, @diachi, @sdt, null, @macuahang, @mahopdong, @khuvuc)</code>	X(CHI_NHANH) //Xin khóa ghi trên bảng CHINHANH (xin thành công)		
B3: Cập nhập số lượng chi nhánh <code>UPDATE CUAHANG SET SO_CHI_NHANH = SO_CHI_NHANH + 1 WHERE MA_CUA_HANG = @macuahang</code>	UX(CUAHANG) //Xin khóa ghi trên bảng CUAHANG (xin thành công)		
COMMIT TRAN			
		B1: Xem danh sách đối tác(cửa hàng) <code>SELECT MA_CUA_HANG, TEN_CUA_HANG, EMAIL, THANH_PHO, QUAN, SDT, SO_CHI_NHANH FROM CUAHANG</code>	S(CUAHANG) //Xin khóa đọc trên bảng CUAHANG (xin thành công)
		COMMIT TRAN	

d. Sinh viên thực hiện: Nguyễn Nhật Trường (20120229)

Tình huống 1: UNREPEATABLE READ

ERR01: UNREPEATABLE READ T1 (User = khách hàng): xem tình trạng đơn hàng T2 (User = tài xế): tài xế thực hiện cập nhật tình trạng đơn hàng			
USP_KIEMTRA_TTGH	Khóa	USP_TX_CAPNHAT_TTGH	Khóa
<u>Input:</u> @MA_KHACH_HANG CHAR(5), @MA_DON CHAR(5) <u>Output:</u>		<u>Input:</u> @MA_TAI_XE CHAR(5), @MA_DON CHAR(5), @TTGH NVARCHAR(255) <u>Output:</u>	
BEGIN TRAN			
B1: kiểm tra @MA_DON_HANG tồn tại <pre> IF NOT EXISTS(SELECT DDH.* FROM DONDATHANG DDH WHERE DDH.MA_KHACH_HANG=@MA_KHACH_HANG AND DDH.MA_DON=@MA_DON) BEGIN PRINT N'DH không tồn tại' </pre>	S(DONDATHANG)		

RETURN END			
DECLARE @TTGH NVARCHAR(255) SET @TTGH = 0 SET @TTGH = (SELECT TINH_TRANG FROM DONDATHANG WHERE MA_DON = @MA_DON)	S(DONDATHANG)		
WAITFOR DELAY '00:00:015'			
		BEGIN TRAN	
		UPDATE DONDATHANG SET TINH_TRANG = @TTGH WHERE MA_DON = @MA_DON AND MA_TAI_XE = @MA_TAI_XE	U.X(DONDATHANG)
		COMMIT TRAN	

<pre> IF ((SELECT TINH_TRANG FROM DONDATHANG WHERE MA_DON = @MA_DON) <> @TTGH) BEGIN RAISERROR(N'Dữ liệu hai lần đọc khác nhau',16,1) END SELECT * FROM DONDATHANG WHERE MA_DON = @MA_DON </pre>	S(DONDATHANG)		
COMMIT TRAN			

Tình huống 2: DIRTY READ

ERR01: DIRTY READ			
<p>T1 (User = Đối tác (cửa hàng)): Đối tác thực hiện cập nhật giá của món ăn B (@MA_MON_AN = 'MA002')</p> <p>T2 (User = Khách hàng): Khách hàng xem giá món ăn B (@MA_MON_AN = 'MA002')</p>			
SP_DT_CAPNHAT_GIA_MONAN	Khóa	USP_KH_XEMGIA_MONAN	Khóa
<p>Input: @MA_MON_AN CHAR(5), @GIA_UPDATE MONEY</p> <p>Output:</p>		<p>Input: @MA_MON_AN CHAR(5)</p> <p>Output:</p>	

BEGIN TRAN			
B1: kiểm tra @MA_MON_AN tồn tại IF NOT EXISTS (SELECT * FROM MON_AN WHERE MA_MON_AN=@MA_MON_AN) BEGIN PRINT N'MÓN ĂN KHÔNG TỒN TẠI' RETURN END	S(MON_AN)		
DECLARE @GIASP MONEY SET @GIASP = (SELECT DON_GIA FROM MON_AN WHERE MA_MON_AN=@MA_MON_AN) UPDATE MON_AN SET DON_GIA = @GIA_UPDATE WHERE MA_MON_AN=@MA_MON_AN	U,X(MON_AN)		
WAITFOR DELAY '00:00:05'			

		BEGIN TRAN	
		B1: kiểm tra @MA_MON_AN tồn tại IF NOT EXISTS (SELECT * FROM MON_AN WHERE MA_MON_AN=@MA_MON_AN) BEGIN PRINT N'MÓN ĂN KHÔNG TỒN TẠI' RETURN END	S(MON_AN)
		SELECT * FROM MON_AN WITH(NOLOCK) WHERE MA_MON_AN=@MA_MON_AN	S(MON_AN)
		COMMIT TRAN	
IF (@GIA_UPDATE = 0) BEGIN ROLLBACK TRAN RETURN			

END			
COMMIT TRAN			

Tình huống 3: CIRCLE DEADLOCK

ERR01: CIRCLE DEADLOCK

T1: Người quản trị 1 cập nhật thông tin tài khoản của khách hàng (đổi mật khẩu) và cập nhật thông tin của tài xế (Email).

T2: Quản trị 2 cập nhật thông tin của tài xế(Email) và cập nhật thông tin tài khoản của khách hàng (đổi mật khẩu).

USP_QT1_CAPNHAT_TT @TEN_DANG_NHAP	Khóa	USP_QT2_CAPNHAT_TT	Khóa
<p><u>Input:</u> @TEN_DANG_NHAP NVARCHAR(255), @MAT_KHAU_MOI CHAR(100), @MATX CHAR(5), @EMAIL_MOI CHAR(100)</p> <p><u>Output:</u></p>		<p><u>Input:</u> @TEN_DANG_NHAP NVARCHAR(255), @MAT_KHAU_MOI CHAR(100), @MATX CHAR(5), @EMAIL_MOI CHAR(100)</p> <p><u>Output:</u></p>	
BEGIN TRAN			

<p>B1: kiểm tra @TEN_DANG_NHAP tồn tại</p> <pre> IF NOT EXISTS(SELECT TK.* FROM TAIKHOAN TK WHERE TEN_DANG_NHAP=@TEN_DANG_NHAP) BEGIN PRINT N'Tài khoản không tồn tại' RETURN END </pre>	S(TAIKHOAN)		
<pre> UPDATE TAIKHOAN SET MAT_KHAU = @MAT_KHAU_MOI WHERE TEN_DANG_NHAP=@TEN_DANG_NHAP </pre>	U,X(TAIKHOAN)		
WAITFOR DELAY '00:00:05'			
		BEGIN TRAN	
		<p>B1: kiểm tra @TEN_DANG_NHAP tồn tại</p>	S(TAIKHOAN)

		<pre> IF NOT EXISTS(SELECT TK.* FROM TAIKHOAN TK WHERE TEN_DANG_NHAP=@TEN_DANG_NHAP) BEGIN PRINT N'Tài khoản không tồn tại' RETURN END </pre>	
		<pre> UPDATE TAIXE SET EMAIL = @EMAIL_MOI WHERE MA_TAI_XE = @MATX </pre>	U,X(TAIXE)
		WAITFOR DELAY '00:00:05'	
		<pre> UPDATE TAIKHOAN SET MAT_KHAU = @MAT_KHAU_MOI WHERE TEN_DANG_NHAP=@TEN_DANG_NHAP </pre>	S(TAIKHOAN)
<pre> UPDATE TAIXE SET EMAIL = @EMAIL_MOI WHERE MA_TAI_XE = @MATX </pre>	U,X(TAIXE)		

COMMIT TRAN			
		COMMIT TRAN	

XỬ LÝ TÌNH HUỐNG TRANH CHẤP

1. Sinh viên thực hiện : Trịnh Hữu Hiệp (20120079)

Tình huống 1: LOST UPDATE

ERR01: lost update T1 (User = cửa hàng (đối tác)): đối tác thực hiện nhập xuất số lượng món ăn, cập nhật số lượng món ăn tại một chi nhánh T2 (User = cửa hàng (đối tác)): đối tác thực hiện thêm mới đơn hàng (khi khách hàng mua tại chi nhánh)			
SP_CH_CAP_NHAT_SL_MON_AN		SP_KHACH_HANG_DAT_HANG_MON_AN	
Input: @MA_CHI_NHANH = 'CN001' @MA_MON_AN = 'MA001' @SO_LUONG_THEM = 100 Output: 1: cập nhật thành công 0: cập nhật thất bại		Input: @MA_DON = 'DH001' @MA_MON_AN = 'MA001' @SO_LUONG = 10 @DIA_CHI = 'QUANG NAM' @PHI_VAN_CHUYEN = 10000 @MA_KHACH_HANG = 'KH001' @MA_TAI_XE = 'TX001' @MA_CHI_NHANH = 'CN001' Output: 1: cập nhật thành công 0: cập nhật thất bại	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ			

BEGIN TRAN			
B1: Kiểm tra thông tin tồn tại của @MA_CHI_NHANH và @MA_MON_AN IF NOT EXISTS (SELECT * FROM MON_AN_CHI_NHANH WHERE MA_CHI_NHANH = @MA_CHI_NHANH AND MA_MON_AN = @MA_MON_AN) --(1) BEGIN PRINT @MA_CHI_NHANH + ', ' + @MA_MON_AN + N' KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0 END	S(MON_AN_CHI_NHANH) //Xin khoá đọc trên bảng MON_AN_CHI_NHANH // (xin thành công) Giữ các khóa đọc đến cuối giao tác do mức cô lập RR		
B2: Kiểm tra @SO_LUONG_THEM IF @SO_LUONG_THEM <1 BEGIN PRINT N'SỐ LƯỢNG THÊM KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0 END	//kiểm tra @SO_LUONG_THEM hợp lệ		
DECLARE @SO_LUONG_HIEN_CO INT SET @SO_LUONG_HIEN_CO = (SELECT SELECT SO_LUONG FROM MON_AN_CHI_NHANH WHERE MA_CHI_NHANH = @MA_CHI_NHANH AND MA_MON_AN = @MA_MON_AN) BEGIN PRINT ' SỐ LƯỢNG HIỆN CÓ KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0 END	S(MON_AN_CHI_NHANH) //không cần xin khoá đọc trên bảng MON_AN_CHI_NHANH Do trước đó đã xin rồi		
WAITFOR DELAY '00:00:10'			
		BEGIN TRAN	
		B1.1: Kiểm tra thông tin tồn tại của @MA_KHACH_HANG	S(KHACHHANG) S(TAIXE)

		<pre> ,@MA_TAI_XE, @MA_CHI_NHANH IF(@MA_KHACH_HANG NOT IN(SELECT MA_KHACH_HANG FROM KHACHHANG) OR @MA_TAI_XE NOT IN(SELECT MA_TAI_XE FROM TAI_XE) OR @MA_CHI_NHANH NOT IN (SELECT MA_CHI_NHANH FROM CHINHANH)) BEGIN PRINT N'KHÔNG TỒN TẠI ' + @MA_KHACH_HANG +N' HOẶC ' + @MA_TAI_XE +N' HOẶC ' + @MA_CHI_NHANH ROLLBACK TRAN RETURN 0 END </pre>	<p>S(CHINHANH) //Xin khoá đọc trên bảng KHACHHANG,TAI_XE, CHINHANH (xin thành công) Giữ các khóa đọc đến cuối giao tác do mức cô lập RR</p>
		<p>B1.2: Kiểm tra thông tin tồn tại của @MA_DON và @MA_MON_AN</p> <pre> IF(EXISTS(SELECT * FROM DONDATHANG DH WHERE DH.MA_DON = @MA_DON)) BEGIN PRINT @MA_DON + N' ĐÃ TỒN TẠI' ROLLBACK TRAN RETURN 0 END IF(@MA_MON_AN NOT IN (SELECT MA_MON_AN FROM MON_AN WHERE TINH_TRANG = N'có bán')) BEGIN PRINT @MA_MON_AN + N' KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(DON_DAT_HANG) //Xin khoá đọc trên bảng (xin thành công) DON_DAT_HANG Giữ các khóa đọc đến cuối giao tác do mức cô lập RR</p> <p>S(MON_AN) //Xin khoá đọc trên bảng MON_AN (xin thành công) Giữ các khóa đọc đến cuối giao tác do mức cô lập RR</p>

		<p>B2: Kiểm tra thông tin tồn tại @SO_LUONG</p> <pre> IF (@SO_LUONG > (SELECT A.SO_LUONG FROM MON_AN_CHI_NHANH A WHERE A.MA_CHI_NHANH = @MA_CHI_NHANH AND A.MA_MON_AN =@MA_MON_AN)) --(2) BEGIN PRINT N'SỐ LƯỢNG HIỆN TẠI CỦA CHI NHÁNH '+ @MA_CHI_NHANH+' KHÔNG ĐỦ ĐÁP ỨNG' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(MON_AN_CHI_NHANH) //Xin khoá đọc trên bảng MON_AN_CHI_NHANH (xin không thành công) Do T1 đã giữ khóa đọc trên MON_AN_CHI_NHANH và chưa commit Chờ T1 Nhả khóa (sau Khi COMMIT)</p>
		<p>B3: lấy đơn giá món ăn , tạo biến tổng tiền trước khi insert</p> <pre> DECLARE @DON_GIA_MON FLOAT SET @DON_GIA_MON = (SELECT DON_GIA FROM MON_AN WHERE MA_MON_AN = @MA_MON_AN) DECLARE @TONG_TIEN FLOAT SET @TONG_TIEN = @DON_GIA_MON * @SO_LUONG </pre>	<p>S(MON_AN) //Xin khoá đọc trên bảng MON_AN (xin thành công)</p>

		<p>B4: insert dữ liệu vào bảng DONDATHANG VÀ CHITIET_DONHANH</p> <pre>-- INSERT DON DAT HÀNG INSERT INTO DONDATHANG VALUES(@MA_DON,@TONG_TIEN,@DIA_CHI,N 'chờ xác nhận',@PHI_VAN_CHUYEN, @MA_KHACH_HANG,@MA_TAI_XE,@MA_CHI_NH ANH,1) -- INSERT CHI TIẾT ĐƠN HÀNG INSERT INTO CHITIET_DONHANG VALUES(@MA_DON , @MA_MON_AN, @SO_LUONG, @TONG_TIEN)</pre>	<p>X(DONDATHANG) //Xin khoá ghi trên bảng DONDATHANG (xin thành công)</p> <p>X(CHITIET_DON_HANG) //Xin khoá ghi trên bảng CHITIET_DONHANG (xin thành công)</p>
--	--	--	--

		<p>B5: cập nhật số lượng món ăn tại chi nhánh khách hàng mua hàng ('CN001')</p> <pre> UPDATE MON_AN_CHI_NHANH SET SO_LUONG = SO_LUONG - @SO_LUONG --(3) WHERE @MA_CHI_NHANH = MA_CHI_NHANH AND @MA_MON_AN = MA_MON_AN </pre>	<p>X(MON_AN_CHI_NHANH) //Xin khoá GHI trên bảng MON_AN_CHI_NHANH (xin thành công)</p>
		<pre> COMMIT TRAN PRINT N'CẬP NHẬT THÀNH CÔNG' RETURN 1 </pre>	
<p>B3:cập nhật số lượng món ăn tại chi nhánh</p> <pre> UPDATE MON_AN_CHI_NHANH SET SO_LUONG = @SO_LUONG_HIEN_CO + @SO_LUONG_THEM WHERE MA_CHI_NHANH = @MA_CHI_NHANH AND MA_MON_AN = @MA_MON_AN </pre>	<p>X(MON_AN_CHI_NHANH) //Xin khoá ghi trên bản MON_AN_CHI_NHANH (xin thành công) Giữ khóa đến hết giao tác</p>		
<pre> COMMIT TRAN PRINT N'CẬP NHẬT THÀNH CÔNG' RETURN 1 </pre>	<p>Nhả khóa trên các đơn vị dữ liệu</p>		

Tình huống 2: UNREPEATABLE READ

ERR01: UNREPEATABLE READ T1 (User = cửa hàng(đối tác)): đối tác thực hiện cập nhật đơn giá của món ăn (@MA_MON_AN = 'MA001') T2 (User = khách hàng): xem tổng tiền của một món ăn (@MA_MON_AN = 'MA001') với số lượng nhất định			
SP_KH_XEM_TONG_TIEN_MOT_MON_AN	Khóa	SP_CH_CAP_NHAT_DON_GIA_MOT_MON_AN	Khóa
Input: @MA_CUA_HANG = 'CH001' @MA_MON_AN = 'MA001' @SOUONG = 10 Output: 1: hiển thị thành công 0: thất bại		Input: @MA_CUA_HANG = 'CH001' @MA_MON_AN = 'MA001' @NEW_DONGIA = 100000 Output: 1: cập nhật thành công 0: cập nhật thất bại	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ			
BEGIN TRAN			
B1: kiểm tra @MA_CUA_HANG tồn tại <pre>IF(@MA_CUA_HANG NOT IN (SELECT MA_CUA_HANG FROM CUAHANG)) BEGIN PRINT @MA_CUA_HANG + N' KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0 END</pre>	S(CUAHANG) //Xin khoá đọc trên bảng (xin thành công) giữ khoá đến hết giao tác do mức cô lập RR		
B1: kiểm tra @MA_MON_AN tồn tại và thuộc về của hàng 'CH001' <pre>IF(NOT EXISTS (SELECT * FROM MON_AN MA WHERE MA.MA_MON_AN = @MA_MON_AN AND MA.TINH_TRANG = N'có bán' AND MA.TRANG_THAI = 1 AND MA.MA_CUA_HANG = @MA_CUA_HANG)) BEGIN PRINT @MA_MON_AN + N' KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0</pre>	S(MON_AN) //Xin khoá đọc trên bảng MON_AN (xin thành công) giữ khoá đến hết giao tác do mức cô lập RR		

END			
<pre> DECLARE @DONGIA FLOAT SET @DONGIA = (SELECT DON_GIA FROM MON_AN WHERE MA_MON_AN = @MA_MON_AN) PRINT N'Giá Món ăn là : ' + CAST(@DONGIA AS NVARCHAR(30)) </pre>	S(MON_AN) Không cần xin khóa do trước đó đã xin rồi		
WAITFOR DELAY '00:00:15'			
		BEGIN TRAN	
		<pre> IF(@MA_CUA_HANG NOT IN (SELECT MA_CUA_HANG FROM CUAHANG)) BEGIN PRINT @MA_CUA_HANG + N' KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0 END </pre>	S(CUAHANG) // Xin khoá đọc trên bảng CUAHANG (xin không thành công) Chờ T1 nhả khóa (Sau khi COMMIT) do khóa được giữ đến hết giao tác)
		<pre> IF(NOT EXISTS (SELECT * FROM MON_AN MA WHERE MA.MA_MON_AN = @MA_MON_AN AND TRANG_THAI !=0 AND MA.MA_CUA_HANG = @MA_CUA_HANG)) BEGIN PRINT @MA_MON_AN + N' KHÔNG HỢP LỆ !' ROLLBACK TRAN RETURN 0 END --(1) </pre>	S(MON_AN) // Xin khoá đọc trên bảng MON_AN (xin thành công)

		<pre> UPDATE MON_AN SET DON_GIA = @NEW_DONGIA WHERE MA_MON_AN = @MA_MON_AN1 </pre>	X(MON_AN) //Xin khoá ghi trên bảng MON_AN (xin thành công)
		<pre> COMMIT TRAN PRINT N'CẬP NHẬT THÀNH CÔNG' RETURN 1 </pre>	
<pre> DECLARE @TONGTIEN FLOAT SET @TONGTIEN = (SELECT @SO_LUONG* DON_GIA FROM MON_AN WHERE @MA_MON_AN = MA_MON_AN) PRINT N'Tổng tiền món ăn với số lượng : ' + CAST(@SO_LUONG AS NVARCHAR(30)) + N' là : ' + CAST(@TONGTIEN AS NVARCHAR(30)) </pre>	X(MON_AN) //Xin khoá ghi trên bảng MON_AN (xin thành công do T2 đã commit và nhả khóa ghi trên bảng MON_AN) Giữ khóa đến hết giao tác		
<pre> COMMIT TRAN PRINT N'HIỂN THỊ THÀNH CÔNG' RETURN 1 </pre>	T1 Nhả các khóa và T2 thực hiện các bước sau đó		

2. Sinh viên thực hiện: Nguyễn Thị Hồng Nhung (20120093)

Tình huống 1: Lỗi Unrepeatable Read

Mô tả cách sửa lỗi: Đặt mức cô lập **REPEATABLE READ** cho T1, Tạo Shared Lock trên đơn vị dữ liệu được đọc và T1 giữ shared lock này đến hết giao tác, giao tác T2 phải chờ đến khi giao tác T1 kết thúc nếu muốn cập nhật, thay đổi giá trị trên đơn vị dữ liệu này.

ERR01: UNREPEATABLE READ

T1 (User = Khách hàng): khách hàng A thực hiện đăng nhập

T2 (User = Khách hàng): ở thiết bị khác của khách hàng A lại cập nhật mật khẩu

dang_nhap	Khóa	cap_nhat_mat_khau	Khóa
<u>Input:</u> tên đăng nhập (@ten_tk) Mật khẩu (@mat_khau) <u>Output:</u> nếu thành công: output ra table gồm: loại tài khoản, mã khách hàng		<u>Input:</u> tên đăng nhập (@ten_tk) Mật khẩu (@mat_khau) Mật khẩu mới (@mat_khau_moi) <u>Output:</u> nếu thành công mật khẩu mới được cập nhật, nếu không thành công rollback	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ		SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Kiểm tra tên đăng nhập có tồn tại không if not exists (select tk.TEN_DANG_NHAP from TAIKHOAN tk where tk.TEN_DANG_NHAP=@ten_tk) begin print N'Không tồn tại tên đăng nhập' rollback tran return -1 end	S(TAIKHOAN) //Xin khóa đọc trên bảng TAIKHOAN sau khi đọc xong thì giữ khóa đến khi kết thúc giao tác		
B2: Kiểm tra mật khẩu có đúng không			

<pre> if not exists (select tk.TEN_DANG_NHAP from TAIKHOAN tk where tk.TEN_DANG_NHAP=@ten_tk and tk.MAT_KHAU= @mat_khau) begin print N'Sai mật khẩu' rollback tran return -1 end </pre>	(Xin thành công)		
<pre> declare @loai_tk int set @loai_tk = NULL SET @loai_tk = (SELECT TOP 1 tk.LOAI_TK FROM TAIKHOAN tk WHERE tk.TEN_DANG_NHAP = @ten_tk AND tk.MAT_KHAU = @mat_khau AND tk.TRANG_THAI = 1) </pre>			
WAITFOR DELAY '00:00:10'			
		BEGIN TRAN	
		<p>B1: Kiểm tra thông tin</p> <pre> if not exists (select tk.TEN_DANG_NHAP from TAIKHOAN tk where tk.TEN_DANG_NHAP=@ten_tk) begin print N'Sai tên đăng nhập' rollback tran return -1 end </pre>	<p>S(TAIKHOAN) //Xin khóa đọc trên bảng TAIKHOAN Vì khóa S tương thích nhau nên vẫn phát khóa S, sau khi đọc xong sẽ giải phóng khóa (Xin thành công)</p>
		<p>B2: Kiểm tra mật khẩu</p> <pre> if not exists (select tk.TEN_DANG_NHAP from TAIKHOAN tk where tk.TEN_DANG_NHAP=@ten_tk and tk.MAT_KHAU= @mat_khau) </pre>	<p>S(TAIKHOAN) //Xin khóa đọc trên bảng TAIKHOAN Vì khóa S tương thích nhau nên vẫn phát khóa S, sau khi đọc</p>

		<pre> begin print N'Sai mật khẩu. Vui lòng nhập đúng mật khẩu để sửa đổi thông tin' rollback tran return -1 end </pre>	<p>xong sẽ giải phóng khóa (Xin thành công)</p>
			<p>X(TAIKHOAN) //Xin khóa ghi trên bảng TAIKHOAN (Chờ T1 commit thì mới có thể xin khóa X để ghi vì T1 đang giữ khóa S đến hết giao tác) (Xin chưa thành công) Đang chờ</p>
<pre> IF @loai_tk IS NOT NULL begin if @loai_tk=0 begin SELECT distinct @loai_tk AS 'loai_tk', CH.MA_CUA_HANG AS 'ma' FROM CUAHANG CH WHERE CH.MA_CUA_HANG = @ten_tk COMMIT TRAN RETURN end else if @loai_tk=1 BEGIN SELECT distinct @loai_tk AS 'loai_tk', KH.MA_KHACH_HANG AS 'ma' FROM KHACHHANG KH, TAIKHOAN tk </pre>	<p>S(TAIKHOAN) //Xin khóa đọc trên bảng TAIKHOAN Giữ khóa S đến hết giao tác (Xin thành công)</p>		

```

WHERE KH.TEN_DANG_NHAP = @ten_tk and
tk.MAT_KHAU=@mat_khau
        COMMIT TRAN
        RETURN

    END
else if @loai_tk=2
    BEGIN
SELECT distinct @loai_tk AS 'loai_tk', TX.MA_TAI_XE
AS 'ma'
FROM TAIXE TX, TAIKHOAN tk
WHERE TX.TEN_DANG_NHAP = @ten_tk and
tk.MAT_KHAU=@mat_khau
        COMMIT TRAN
        RETURN

    END
    else if @loai_tk=3
    begin
select distinct @loai_tk as 'loai_tk',
NV.MA_NHAN_VIEN from NHANVIEN NV,
TAIKHOAN tk
where NV.TEN_DANG_NHAP=@ten_tk and
tk.MAT_KHAU=@mat_khau
        end
        else if @loai_tk=3
        begin
        declare @ad char(10)
        set @ad= 'admin'
select distinct @loai_tk as 'loai_tk',
tk.TEN_DANG_NHAP from TAIKHOAN tk
where tk.TEN_DANG_NHAP=@ten_tk and
tk.MAT_KHAU=@mat_khau
        end

    end
    end
    
```





<pre> end try begin catch print N'Dăng nhập thất bại' rollback tran end catch commit tran go </pre>			
COMMIT TRAN	Giải phóng khóa S		
		<pre> update TAIKHOAN set MAT_KHAU=@mat_khau_moi where TEN_DANG_NHAP= @ten_tk print N'Dổi mật khẩu thành công' end try begin catch print N'Thay đổi mật khẩu thất bại' rollback tran end catch </pre>	<p>X(TAIKHOAN) //Xin khóa ghi trên bảng TAIKHOAN(xin thành công do T1 đã commit và giải phóng khóa S)</p>
		COMMIT TRAN	Giải phóng khóa X

Tình huống 2: Lỗi Conversion Deadlock

Mô tả sửa lỗi: Đối với trường hợp **CONVERSION DEADLOCK**, thực hiện đề xuất sử dụng khóa tăng cấp (Update lock) để giải quyết tình trạng lỗi này. Khi thực hiện, T1 sẽ được cấp khóa tăng cấp, khóa này chỉ cho phép đọc dữ liệu trên bảng HOPDONG, đồng thời T2 cũng thực hiện xin khóa tăng cấp. Tuy nhiên, T2 phải chờ T1 nhả khóa thì mới có thể thực hiện các thao tác của mình, vì do T2 cũng xin khóa tăng cấp cùng đơn vị dữ liệu với T1. Sau đó, T1 sẽ update lên khóa ghi để thực hiện cập nhật dữ liệu và commit. Lúc này, T2 bắt đầu thực hiện tương tự như T1.

ERR02: CONVERSION DEADLOCK T1 (User = Nhân viên): nhân viên A thực hiện gia hạn hợp đồng của cửa hàng C T2 (User = Nhân viên): nhân viên B cũng thực hiện gia hạn hợp đồng của cửa hàng C			
Gia_han_hop_dong	Khóa	Gia_han_hop_dong	Khóa
Input: Mã hợp đồng A (@maHD), số lượng ngày thêm gia hạn hợp đồng (@so_ngay_them) Output: thời gian kết thúc được gia hạn thêm với số ngày gia hạn hợp đồng thêm		Input: Mã hợp đồng A (@maHD), số lượng ngày thêm gia hạn hợp đồng (@so_ngay_them) Output: thời gian kết thúc được gia hạn thêm với số ngày gia hạn hợp đồng thêm	
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE		SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
BEGIN TRAN			
B1: Kiểm tra mã Hợp đồng có tồn tại không if @maHD not in (select hd.MA_HOP_DONG from HOPDONG hd with (UPDLOCK) where hd.MA_HOP_DONG=@maHD) begin print N'Không tồn tại hợp đồng này' rollback tran end	U(HOPDONG) //Xin khóa update lock trên bảng HOPDONG (Xin thành công) Giữ đến cuối giao tác		
B2: tạo biến @ngay_hien_tai và set bằng ngày cuối hợp đồng declare @ngay_hien_tai date set @ngay_hien_tai=(select top 1 hd.TG_KET_THUC from HOPDONG hd where hd.MA_HOP_DONG=@maHD)			
if @ngay_hien_tai is not null begin -- bắt đầu chờ WAITFOR DELAY '00:00:15'			
		BEGIN TRAN	

		B1.1: Kiểm tra mã Hợp đồng có tồn tại không if @maHD not in (select hd.MA_HOP_DONG from HOPDONG hd where hd.MA_HOP_DONG=@maHD) begin print N'Không tồn tại hợp đồng này' rollback tran end	U(HOPDONG) //Xin khóa tăng cấp trên bảng HOPDONG Tuy nhiên vì T1 đang giữ khóa tăng cấp trên bảng HOPDONG nên T2 phải chờ T1 giải phóng khóa. Sau khi T1 commit thì T2 sẽ xin thành công khóa U
		B2.1: tạo biến @ngay_hien_tai và set bằng ngày cuối hợp đồng declare @ngay_hien_tai date set @ngay_hien_tai=(select top 1 hd.TG_KET_THUC from HOPDONG hd where hd.MA_HOP_DONG=@maHD)	
		if @ngay_hien_tai is not null begin -- bắt đầu chờ WAITFOR DELAY '00:00:15'	
-- gia hạn ngày hợp đồng set @ngay_hien_tai = (select DATEADD(DAY, @so_ngay_them, @ngay_hien_tai)) -- cập nhật vào hợp đồng update HOPDONG SET TG_KET_THUC = @ngay_hien_tai WHERE MA_HOP_DONG = @maHD; end	X(HOPDONG) //T1 update khóa U thành khóa X để có thể ghi trên bảng HOPDONG và giữ khóa X đến cuối giao tác(Thành công)		
end try			

begin catch print N'Gia hạn hợp đồng thất bại' rollback tran end catch			
COMMIT TRAN	Giải phóng khóa X		
		-- gia hạn ngày hợp đồng set @ngay_hien_tai = (select DATEADD(DAY, @so_ngay_them, @ngay_hien_tai)) -- cập nhật vào hợp đồng update HOPDONG SET TG_KET_THUC = @ngay_hien_tai WHERE MA_HOP_DONG = @maHD; end	X(HOPDONG) //T1 update khóa U thành khóa X để có thể ghi trên bảng HOPDONG và giữ khóa X đến cuối giao tác (Thành công) 
		end try begin catch print N'Gia hạn hợp đồng thất bại' rollback tran end catch	
		COMMIT TRAN	Giải phóng khóa X

3. Sinh viên thực hiện: Trần Thái San

Tình huống 1: DIRTY READ

ERR03: Dirty Read T1 (User = cửa hàng (đối tác)): thực cập nhật đơn giá món ăn T2 (User = khách hàng): khách hàng thực hiện xem thông tin món ăn			
SP_CAPNHAPGIAMONAN Input: @MAMONAN = 'MA001' @MA_CUA_HANG = 'CH001' @GIAMOI = 100000 Output: 1: cập nhật thành công 0: cập nhật thất bại		SP_XEMONAN Input: @MA_MON_AN = 'MA001' Output: :	
BEGIN TRAN			
B1: Kiểm tra món ăn có tồn tại hay không <pre> IF NOT EXISTS(SELECT * FROM MON_AN WHERE @MAMONAN=MA_MON_AN AND @MACH=MA_CUA_HANG) BEGIN PRINT N'KHÔNG TỒN TẠI MÓN ĂN' ROLLBACK TRAN RETURN 1 END </pre>	S(MON_AN) //Xin khoá đọc trên bảng MON_AN với điều kiện MaCH = 'CH001' và MaMonAn = 'MA001' (xin thành công)		
B2: Thực hiện cập nhật giá mới <pre> UPDATE MON_AN SET DON_GIA=@GIAMOI WHERE @MAMONAN=MA_MON_AN AND @MACH=MA_CUA_HANG </pre>	U,X(MON_AN) //Xin khoá U,X trên bảng MON_AN với điều kiện @MAMONAN='MA001' @MACH = 'CH001' @GIAMOI =0 (xin thành công)		
WAITFOR DELAY '00:00:10'			

		BEGIN TRAN	
		Thực hiện xem món ăn của đối tác với MAMONAN = 'MA001' <code>SELECT MA.MA_MON_AN, MA.TEN_MON, MA.DON_GIA, CH.TEN_CUA_HANG, MA.TINH_TRANG, MA.TRANG_THAI FROM MON_AN MA, CUAHANG CH WHERE MA.MA_MON_AN=@MAMONAN AND MA.MA_CUA_HANG=CH.MA_CUA_HANG</code>	S(MON_AN) // Xin khoá đọc trên bảng MON_AN với điều kiện MA_MON_AN = 'MA001' (xin thành công)
		COMMIT TRAN	
B3: Kiểm tra giá mới có bằng 0 hay không, nếu bằng thì rollbacktran <code>IF @GIAMOI=0 BEGIN ROLLBACK TRAN RETURN 1 END</code>			
COMMIT TRAN			

GIẢI THÍCH: Để xử lý Dirty Read, nhóm em đề xuất mức cô lập mặc định cho T2 (Read Committed) khi đó khi T1 thực hiện select và update giá trên bảng MON_AN thì T2 cũng thực hiện đọc giá sản phẩm rồi tiếp tục thực hiện T1. Khi đó theo cơ chế của READ COMMITTED thì T2 chỉ được phép đọc dữ liệu đã commit mà T1 chưa commit nên T2 sẽ được trả lại kết quả ban đầu chưa update

Tình huống 2: PHANTOM READ

ERR03: Phantom Read T1 (User = cửa hàng (đối tác)): thực hiện thêm một chi nhánh mới T2 (User = khách hàng): thực hiện xem danh sách đối tác			
sp_DT_ThemChiNhanh		sp_KH_XemDSDoiTac	
Input: @machinhanh @diachi @sdt @macuahang @mahopdong @khuvuc Output: 1: thêm thành công 0: thêm thất bại		Input: Output: : 1: cập nhật thành công 0: cập nhật thất bại	
SET TRANSACTION ISOLATION LEVEL		SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
		BEGIN TRAN	
		B1: Xem danh sách đối tác(cửa hàng) SELECT MA_CUA_HANG, TEN_CUA_HANG, EMAIL, THANH_PHO, QUAN, SDT, SO_CHI_NHANH FROM CUAHANG	S(CUAHANG) //Xin khóa đọc trên bảng CUAHANG (xin thành công)
		WAITFOR DELAY '00:00:10'	
BEGIN TRAN			
B1: Kiểm tra trùng lặp của @machinhanh và @diachi IF(EXISTS(SELECT * FROM CHINHANH WHERE MA_CUA_HANG = @macuahang AND DIA_CHI = @diachi)) begin rollback tran RETURN -1 end	S(CHI_NHANH) //Xin khoá đọc trên bảng CHI_NHANH với điều kiện @machinhanh = 'CN001' (xin thành công) S(CHI_NHANH)		

<pre>IF(EXISTS(SELECT * FROM CHINHANH WHERE MA_CHI_NHANH = @machinhanh)) begin rollback tran RETURN -1 end</pre>	<pre>//Xin khoá đọc trên bảng CHI_NHANH với điều kiện @diachi = 'Dĩ An, Bình Dương' (xin thành công)</pre>		
<p>B2: Thêm chi nhánh</p> <pre>INSERT INTO CHINHANH(MA_CHI_NHANH, SL_DON_MOINGAY, DIA_CHI, SDT, TINH_TRANG, MA_CUA_HANG, MA_HOP_DONG, KHUVUC) VALUES(@machinhanh, NULL, @diachi, @sdt, null, @macuahang, @mahopdong, @khuvuc)</pre>	<p>X(CHI_NHANH)</p> <pre>//Xin khóa ghi trên bảng CHINHANH (xin thành công)</pre>		
<p>B3: Cập nhật số lượng chi nhánh</p> <pre>UPDATE CUAHANG SET SO_CHI_NHANH = SO_CHI_NHANH + 1 WHERE MA_CUA_HANG = @macuahang</pre>	<p>UX(CUAHANG)</p> <pre>//Xin khóa ghi trên bảng CUAHANG</pre>		
COMMIT TRAN			
		<p>B1: Xem danh sách đối tác(cửa hàng)</p> <pre>SELECT MA_CUA_HANG, TEN_CUA_HANG, EMAIL, THANH_PHO, QUAN, SDT, SO_CHI_NHANH FROM CUAHANG</pre>	<p>S(CUAHANG)</p> <pre>//Xin khóa đọc trên bảng CUAHANG (xin thành công)</pre>
		COMMIT TRAN	

Giải thích: Để khắc phục lỗi Phantom Read, ở đây nhóm đề xuất đặt T2 ở mức cô lập Serializable, khi đó nó sẽ đặt 1 khóa ở T2 ngăn chặn các giao tác khác thêm hay cập nhật trong T2, qua đó giải quyết được Phantom Read. Khi thực hiện T2, T1 bị chặn cho đến khi T2 hoàn thành và kết thúc T2, cả 2 lần đọc đều nhận được giá trị như nhau

4. Sinh viên thực hiện: Nguyễn Nhật Trường

Tình huống 1: UNREPEATABLE READ

ERR01: UNREPEATABLE READ			
T1 (User = khách hàng): xem tình trạng đơn hàng T2 (User = tài xế): tài xế thực hiện cập nhật tình trạng đơn hàng			
USP_KIEMTRA_TTGH	Khóa	USP_TX_CAPNHAT_TTGH	Khóa
Input: @MA_KHACH_HANG CHAR(5), @MA_DON CHAR(5) Output:		Input: @MA_TAI_XE CHAR(5), @MA_DON CHAR(5), @TTGH NVARCHAR(255) Output:	
BEGIN TRAN			
SET TRAN ISOLATION LEVEL REPEATABLE READ	Thiết lập khóa S khi đọc và giữ đến khi hết giao tác		
<pre>IF NOT EXISTS(SELECT DDH.* FROM DONDATHANG DDH WHERE DDH.MA_KHACH_HANG=@MA_KHACH_HANG AND DDH.MA_DON=@MA_DON) BEGIN PRINT N'DH không tồn tại' RETURN END</pre>	S(DONDATHANG) T1 xin khóa đọc trên bảng DONDATHANG. T1 sẽ nhả khóa sau khi commit.		
<pre>DECLARE @TTGH NVARCHAR(255) SET @TTGH = 0 SET @TTGH = (SELECT TINH_TRANG FROM DONDATHANG WHERE MA_DON = @MA_DON)</pre>	S(DONDATHANG) T1 xin khóa đọc trên bảng DONDATHANG. T1 sẽ nhả khóa sau khi commit.		
WAITFOR DELAY '00:00:015'			

		BEGIN TRAN	
		<pre> UPDATE DONDATHANG SET TINH_TRANG = @TTGH WHERE MA_DON = @MA_DON AND MA_TAI_XE = @MA_TAI_XE </pre>	U,X(DONDATHANG) T1 xin khóa ghi trên bảng DONDATHANG thành công. T2 sẽ chờ T1 commit để xin khóa.
		COMMIT TRAN	
<pre> IF ((SELECT TINH_TRANG FROM DONDATHANG WHERE MA_DON = @MA_DON) <> @TTGH) BEGIN RAISERROR(N'Dữ liệu hai lần đọc khác nhau',16,1) END SELECT * FROM DONDATHANG WHERE MA_DON = @MA_DON </pre>	S(DONDATHANG) T1 xin khóa ghi trên bảng DONGIAOHANG thành công. T2 sẽ chờ T1 commit để xin khóa.		
COMMIT TRAN			

GIẢI THÍCH: Áp dụng mức cô lập REPEATABLE READ đối với T1, làm cho T2 không thể chen ngang UPDATE dữ liệu khi T1 đang thực hiện. Kết quả hai lần đọc cho ra TINHTRANG đều cho kết quả giống nhau.

Tình huống 2: Dirty Read

ERR01: DIRTY READ			
T1 (User = Đối tác (cửa hàng)): Đối tác thực hiện cập nhật giá của món ăn B (@MA_MON_AN = 'MA002')			
T2 (User = Khách hàng): Khách hàng xem giá món ăn B (@MA_MON_AN = 'MA002')			
USP_DT_CAPNHAT_GIA_MONAN	Khóa	USP_KH_XEMGIA_MONAN	Khóa
Input: @MA_MON_AN CHAR(5), @GIA_UPDATE MONEY Output:		Input: @MA_MON_AN CHAR(5) Output:	
BEGIN TRAN			
B1: kiểm tra @MA_MON_AN tồn tại <pre>IF NOT EXISTS (SELECT * FROM MON_AN WHERE MA_MON_AN=@MA_MON_AN) BEGIN PRINT N'MÓN ĂN KHÔNG TỒN TẠI' RETURN END</pre>	S(MON_AN) T1 xin khóa đọc trên bảng MON_AN ứng với MA_MON_AN=@MA_M ON_AN		
<pre>DECLARE @GIASP MONEY SET @GIASP = (SELECT DON_GIA FROM MON_AN WHERE MA_MON_AN=@MA_MON_AN) UPDATE MON_AN SET DON_GIA = @GIA_UPDATE WHERE MA_MON_AN=@MA_MON_AN</pre>	S(MON_AN) T1 xin khóa đọc trên bảng MON_AN ứng với MA_MON_AN=@MA_M ON_AN		
WAITFOR DELAY '00:00:05'			
		BEGIN TRAN	
		SET TRAN ISOLATION LEVEL READ COMMITTED	

		<pre>SELECT * FROM MON_AN WITH(NOLOCK) WHERE MA_MON_AN=@MA_MON_AN</pre>	<p>S(MON_AN) T2 xin khóa đọc trên bảng MON_AN ứng với MA_MON_AN=@MA_MON_AN</p>
		COMMIT TRAN	
<pre>IF (@GIA_UPDATE = 0) BEGIN ROLLBACK TRAN RETURN END</pre>			
COMMIT TRAN			

GIẢI THÍCH: Việc áp dụng NOLOCK trên bảng MON_AN để cho kết quả nhanh chóng đã dẫn đến tình trạng DIRTY READ, giao tác T2 đọc dữ liệu chưa được commit bởi T1. Để tránh tình trạng Dirty Read, nhóm đề xuất xóa NOLOCK đối với bảng MON_AN, để đưa T2 về mức cô lập READ COMMITTED. Lúc này, quá trình thực hiện của hai giao tác trên được thực hiện dựa trên mũi tên màu đỏ ở trên. Đầu tiên, T1 thực hiện select và update giá trên bảng MON_AN, cùng lúc đó, T2 thực hiện đọc giá sản phẩm của T1 vừa cập nhật. Tiếp đến, vì giá sản phẩm bị nhập sai, nên T1 đã rollback, theo cơ chế của mức cô lập READ COMMITTED, thì T2 chỉ được phép đọc dữ liệu đã commit, vì thế kết quả đọc dữ liệu của T2 sẽ được trả lại kết quả ban đầu chưa thực hiện update.

Tình huống 3: Circle Deadlock

ERR01: CIRCLE DEADLOCK

T1: Người quản trị 1 cập nhật thông tin tài khoản của khách hàng (đổi mật khẩu) và cập nhật thông tin của tài xế (Email).

T2: Quản trị 2 cập nhật thông tin của tài xế(Email) và cập nhật thông tin tài khoản của khách hàng (đổi mật khẩu).

USP_QT1_CAPNHAT_TT @TEN_DANG_NHAP	Khóa	USP_QT2_CAPNHAT_TT	Khóa
Input: @TEN_DANG_NHAP NVARCHAR(255), @MAT_KHAU_MOI CHAR(100), @MATX CHAR(5), @EMAIL_MOI CHAR(100) Output:		Input: @TEN_DANG_NHAP NVARCHAR(255), @MAT_KHAU_MOI CHAR(100), @MATX CHAR(5), @EMAIL_MOI CHAR(100) Output:	
BEGIN TRAN			
B1: kiểm tra @TEN_DANG_NHAP tồn tại <pre> IF NOT EXISTS(SELECT TK.* FROM TAIKHOAN TK WHERE TEN_DANG_NHAP=@TEN_DANG_NHAP) BEGIN PRINT N'Tài khoản không tồn tại' RETURN END </pre>	S(TAIKHOAN) T1 xin khóa ghi trên bảng TAIKHOAN.		
<pre> UPDATE TAIKHOAN SET MAT_KHAU = @MAT_KHAU_MOI WHERE TEN_DANG_NHAP=@TEN_DANG_NHAP </pre>	U,X(TAIKHOAN) T1 xin khóa trên bảng TAIKHOAN		
WAITFOR DELAY '00:00:05'			
		BEGIN TRAN	

		UPDATE TAIKHOAN SET MAT_KHAU = @MAT_KHAU_MOI WHERE TEN_DANG_NHAP=@TEN_DANG_NHAP	U,X(TAIXE) T2 xin khóa trên bảng TAIKHOAN. T2 chờ T1 commit thì bắt đầu thực hiện.
		WAITFOR DELAY '00:00:05'	
UPDATE TAIXE SET EMAIL = @EMAIL_MOI WHERE MA_TAI_XE = @MATX	U,X(TAIXE) T1 xin khóa trên bảng TAIKHOAN		
		UPDATE TAIXE SET EMAIL = @EMAIL_MOI WHERE MA_TAI_XE = @MATX	S(TAIKHOAN) T1 xin khóa trên bảng TAIKHOAN
COMMIT TRAN			
		COMMIT TRAN	

GIẢI THÍCH: Về lỗi Circle Deadlock không có cách nào để xử lý tranh chấp theo các cách thông thường nên em đề xuất việc thay đổi thứ tự thao tác đối với T2, để tránh tình trạng gây ra CYCLE DEADLOCK. Khi đó, vì cả 2 giao tác ở mức cô lập COMMITTED cho nên T1 thực hiện thực hiện và commit thì T2 sẽ bắt đầu thực hiện. → Cả 2 giao tác đều được thực thi thành công