

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



Báo cáo Project #2

STT 4 – 5 – 6

HỆ THỐNG ĐẶT VÀ GIAO THỨC ĂN ONLINE

Lớp: 20VP

Nhóm 20VP.HQT.05:

20126021 - Lê Viết Lực
20126039 - Trần Thiện Khiêm
20126040 - Lê Đa Khoa
20126044 - Nguyễn Cường Nam

HỌC PHẦN: Hệ quản trị cơ sở dữ liệu-CSC12003_20VP

Thành phố Hồ Chí Minh – 2023

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



Báo cáo Project #2

STT 4 – 5 – 6

HỆ THỐNG ĐẶT VÀ GIAO THỨC ĂN ONLINE

| Giảng viên hướng dẫn |

Phạm Thị Bạch Huệ

Hồ Thị Hoàng Vy

HỌC PHẦN: Hệ quản trị cơ sở dữ liệu-CSC12003_20VP

Thành phố Hồ Chí Minh – 2023

MỤC LỤC

THÔNG TIN NHÓM VÀ PHÂN CÔNG CÔNG VIỆC	6
TÌNH HUỐNG TRANH CHẤP	8
1. Xác định các tình huống tranh chấp :.....	8
A. LOST UPDATE	8
Tình huống 1:	8
Tình huống 2:	12
B. DIRTY READ	14
Tình huống 1:	14
Tình huống 2:	17
C. UNREPEATABLE READ	19
Tình huống 1:	19
Tình huống 2:	21
D. PHANTOM	23
Tình huống 1:	23
Tình huống 2	26
E. CYCLE DEADLOCK	28
F. CONVERSION DEADLOCK	30
2. Xử lý tranh chấp:	33

A. LOST UPDATE.....	33
Trường hợp 1:	33
Trường hợp 2:	38
B. DIRTY READ.....	40
Tình huống 1:	40
Tình huống 2:	42
C. UNREPEATABLE READ	44
Tình huống 1:	44
Tình huống 2:	46
D. PHANTOM.....	47
Tình huống 1	47
Tình huống 2	50
E. CYCLE DEADLOCK.....	53
F. CONVERSION DEADLOCK	54
CÀI ĐẶT HỆ THỐNG GIAO DIỆN ỨNG DỤNG	58
Tham khảo	72

THÔNG TIN NHÓM VÀ PHÂN CÔNG CÔNG VIỆC

STT	Thời gian	Lê Viết Lực	Trần Thiện Khiêm	Lê Đa Khoa	Nguyễn Cường Nam	Hoàn thành
1	13/2/2023 - 23/2/2023	+ Tham gia thiết kế cấu trúc cơ sở dữ liệu + Viết scripts tạo các bảng cho tài xế	+ Tham gia thiết kế cấu trúc cơ sở dữ liệu + Viết scripts tạo các bảng cho khách hàng	+ Tham gia thiết kế cấu trúc cơ sở dữ liệu + Viết scripts tạo các bảng cho đối tác	+ Tham gia thiết kế cấu trúc cơ sở dữ liệu + Viết scripts tạo các bảng cho nhân viên + Viết báo cáo	100%
2	24/2/2023 - 28/2/2023	+ Tham gia phân quyền cho cơ sở dữ liệu + Viết scripts phân quyền cho các bảng tài xế	+ Tham gia phân quyền cho cơ sở dữ liệu + Viết scripts phân quyền cho các bảng admin	+ Tham gia phân quyền cho cơ sở dữ liệu + Viết scripts phân quyền cho các bảng đối tác, khách hàng	+ Tham gia phân quyền cho cơ sở dữ liệu + Viết scripts phân quyền cho các bảng nhân viên. + Viết báo cáo	100%
3	01/3/2023 -10/3/2023	+ Nêu các tình huống tranh chấp cho 1 chức năng + Tham gia thiết kế giao diện (prototype) + Thiết kế giao diện cho tài xế, đăng nhập, đăng ký	+ Nêu các tình huống tranh chấp cho 1 vận t chia stt chức năng + Tham gia thiết kế giao diện (prototype) + Thiết kế giao diện cho admin, thực đơn, các món ăn	+ Nêu các tình huống tranh chấp cho 1 chức năng + Tham gia thiết kế giao diện (prototype) + Thiết kế giao diện cho đối tác, khách hàng	+ Nêu các tình huống tranh chấp cho 1 chức năng + Tham gia thiết kế giao diện (prototype) + Thiết kế giao diện cho nhân viên + Viết báo cáo	100%

STT	Thời gian	Lê Viết Lực	Trần Thiện Khiêm	Lê Đa Khoa	Nguyễn Cường Nam	Hoàn thành
4	11/3/2023 - 25/3/2023	+Làm tình huống tranh chấp 1 cái cho Unrepeatable read, 1 cái phantom read, 1 cái conversion deadlock +Viết báo cáo về tình huống	+ Làm tình huống tranh chấp 1 cái lost update và 1 cái dirty read +Viết báo cáo về tình huống	+ Làm tình huống tranh chấp 1 cái lost update và 1 cái dirty read +Viết báo cáo về tình huống	+Làm tình huống tranh chấp 1 cái cho Unrepeatable read, 1 cái phantom read, 1 cái cycle deadlock + Viết báo cáo tổng	100%
5	26/3/2023 - 2/4/2023	+ Thực hiện tìm hướng giải quyết cho các tranh chấp cho 3 tình huống đã được phân công +Viết báo cáo về tình huống	+ Thực hiện tìm hướng giải quyết cho 2 tình huống tranh chấp đã được giao. +Viết báo cáo về tình huống	+ Thực hiện tìm hướng giải quyết cho 2 tình huống tranh chấp đã được giao. +Viết báo cáo về tình huống	+ Thực hiện tìm hướng giải quyết cho các tranh chấp cho 3 tình huống đã được phân công +Viết báo cáo tổng	100%
6	3/4/2023 -17/4/2023	+ Cùng với Nam làm giao diện cho tài khoản admin, tài khoản nhân viên và làm phần đăng nhập.	+ Cùng với Khoa làm giao diện cho tài khoản khách hàng, tài xế, đối tác và đăng ký tài khoản.	+ Cùng với Khiêm làm giao diện cho tài khoản khách hàng, tài xế, đối tác và đăng ký tài khoản.	+ Cùng với Lực làm giao diện cho tài khoản admin, tài khoản nhân viên và làm phần đăng nhập. + Viết báo cáo tổng	100%

TÌNH HUỐNG TRANH CHẤP

1. Xác định các tình huống tranh chấp :

A. LOST UPDATE	
Tình huống 1:	
T1 Khách hàng 1 mua sản phẩm A. Input: @MADH CHAR(5), @MASP CHAR(5), @SLMUA INT	T2 Khách hàng 2 mua sản phẩm A. Input: @MADH CHAR(5), @MASP CHAR(5), @SLMUA INT
BEGIN TRAN	
DECLARE @SLTON INT SET @SLTON = (SELECT SLTON FROM SANPHAM WHERE MASP = @MASP)	
IF (@SLTON = 0) BEGIN PRINT N'Số lượng hàng trong kho đã hết' ROLLBACK TRAN END	
IF (@SLMUA > @SLTON) BEGIN PRINT N'Số lượng hàng trong kho không đủ đáp ứng với số lượng mua' ROLLBACK TRAN END	
WAITFOR DELAY '00:00:05'	
	BEGIN TRAN
	DECLARE @SLTON INT SET @SLTON = (SELECT SLTON FROM SANPHAM WHERE MASP = @MASP)
	IF (@SLTON = 0) BEGIN PRINT N'Số lượng hàng trong kho đã hết'

	ROLLBACK TRAN END
	IF (@SLMUA > @SLTON) BEGIN PRINT N'Số lượng hàng trong kho không đủ đáp ứng với số lượng mua' ROLLBACK TRAN END
	WAITFOR DELAY '00:00:05'
INSERT [dbo].[CTDONHANG] ([MASP], [MADH], [SLSANPHAM]) VALUES (@MASP, @MADH, @SLMUA)	
SET @SLTON = @SLTON - @SLMUA	
UPDATE SANPHAM SET SLTON = @SLTON WHERE MASP = @MASP	
COMMIT TRAN	
	INSERT [dbo].[CTDONHANG] ([MASP], [MADH], [SLSANPHAM]) VALUES (@MASP, @MADH, @SLMUA)
	SET @SLTON = @SLTON - @SLMUA
	UPDATE SANPHAM SET SLTON = @SLTON WHERE MASP = @MASP
	COMMIT TRAN

Tình huống giả định: Giả sử hai khách hàng 1 và 2 cùng đăng nhập vào hệ thống và thực hiện đồng thời hành động mua sản phẩm A. Ví dụ được minh họa cụ thể như sau:

Cụ thể, lúc đầu số lượng tồn trong kho của mã sản phẩm ‘SP008’ là 22.

SP008	DT097	Bàn chải đánh răng	22	445424.8568
-------	-------	--------------------	----	-------------

Khách hàng 1 mua sản phẩm ‘SP008’ với số lượng 6.

```

Transact-SQL
USE [QL_DH_GH_ONLINE]
GO
/*LOST UPDATE*/
/*T1*/
CREATE OR ALTER PROCEDURE KH1_MUASP @MADH CHAR(5), @MASP CHAR(5), @SLMUA INT
AS
BEGIN TRAN
DECLARE @SLTON INT
SET @SLTON = (SELECT SLTON FROM SANPHAM WHERE MASP = @MASP)
IF (@SLTON = 0)
BEGIN
PRINT N'Số lượng hàng trong kho đã hết'
ROLLBACK TRAN
END
IF (@SLMUA > @SLTON)
BEGIN
PRINT N'Số lượng hàng trong kho không đủ đáp ứng với số lượng mua'
ROLLBACK TRAN
END
WAITFOR DELAY '00:00:10'
INSERT [dbo].[CTDONHANG] ([MASP], [MADH], [SLSANPHAM]) VALUES (@MASP, @MADH, @SLMUA)
SET @SLTON = @SLTON - @SLMUA
UPDATE SANPHAM
SET SLTON = @SLTON
WHERE MASP = @MASP
COMMIT TRAN
GO
EXEC KH1_MUASP 'DH767', 'SP008', 6
  
```

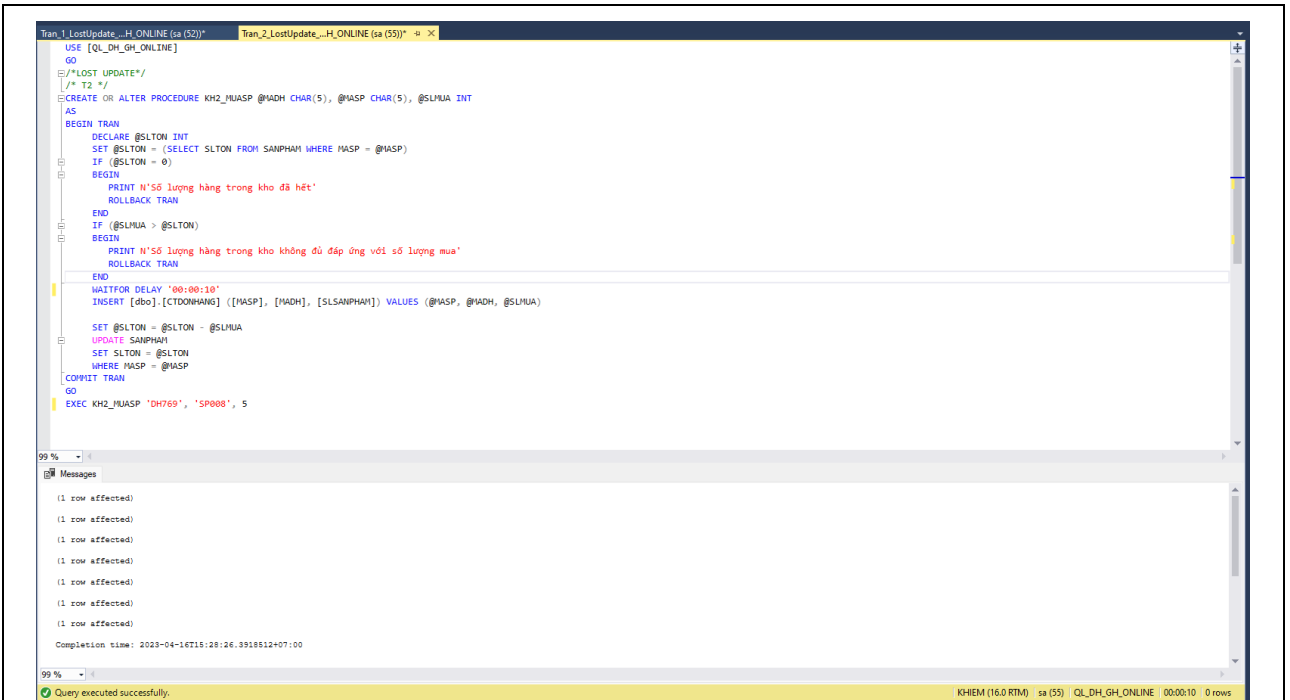
Messages

```

(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
Completion time: 2023-04-16T15:28:24.0409693+07:00
  
```

Query executed successfully. KHIEM (16.0 RTM) sa (52) QL_DH_GH_ONLINE 00:00:10 0 rows

Cùng lúc đó, khách hàng 2 cũng mua sản phẩm ‘SP005’ với số lượng 5.



KẾT QUẢ: Sau khi hai giao tác T1 & T2 được thực hiện đồng thời, kết quả mà T1 và T2 làm thay đổi số lượng tồn của 'SP006' là 17.

SP008	DT097	Bàn chải đánh răng	17	445424.8568
-------	-------	--------------------	----	-------------

Đáng lẽ, số lượng khi T1 mua là 6, T2 là 5 thì số lượng sản phẩm phải là 11. Điều này, có thể thấy rằng dữ liệu đã bị mất trong quá trình 2 giao tác cùng đọc và cập nhật cùng 1 đơn vị dữ liệu. Kết quả làm số lượng trong kho sau khi 2 giao tác thực hiện đã bị sai. Điều này là do tình trạng **LOST UPDATE**.

Tình huống 2:	
T1	T2
Đối tác cập nhật số lượng sản phẩm của mình. INPUT: @MASP CHAR(5), @SL INT	Đối tác cập nhật số lượng sản phẩm của mình. INPUT: @MASP CHAR(5), @SL INT
BEGIN TRAN	
SELECT SLTON FROM SANPHAM WHERE MASP = @MASP	
WAITFOR DELAY '00:00:05'	
	BEGIN TRAN
	SELECT SLTON FROM SANPHAM WHERE MASP = @MASP
	WAITFOR DELAY '00:00:05'
UPDATE SANPHAM SET SLTON = @SL WHERE MASP = @MASP	
COMMIT TRAN	
	UPDATE SANPHAM SET SLTON = @SL WHERE MASP = @MASP
	COMMIT TRAN

Tình huống giả định: Hai nhân viên của một đối tác cùng truy cập vào một tài khoản ở 2 khu vực khác nhau để cập nhật số lượng sản phẩm. Hai nhân viên của đối tác này đồng thời cập nhật khiến cho một trong hai giao tác bị mất dữ liệu khi cập nhật.

T1:

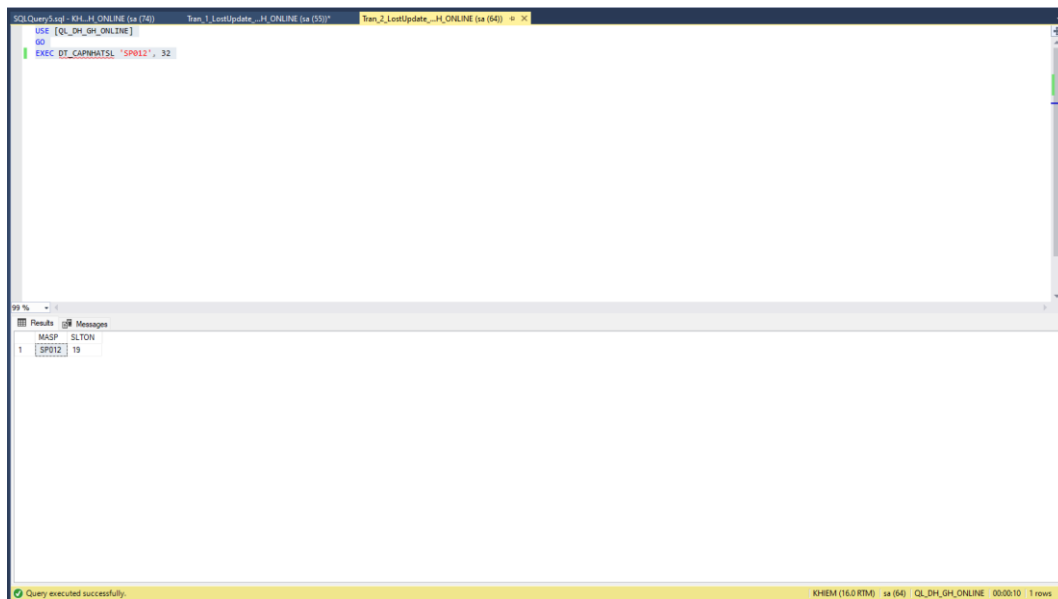
```

USE [OL_OLW_ONLINE]
GO
CREATE OR ALTER PROCEDURE DT_CAPNHATSL @MASP CHAR(5), @SL INT
AS
BEGIN TRAN
    SELECT MASP, SLTON
    FROM SANPHAM
    WHERE MASP = @MASP
    WAITFOR DELAY '00:00:10'
    UPDATE SANPHAM
    SET SLTON = @SL
    WHERE MASP = @MASP
COMMIT TRAN
GO
EXEC DT_CAPNHATSL 'SPWEL', 20
  
```

Results

MAASP	SLTON
SPWEL	10

Query executed successfully.

T2:

Ta thấy kết quả đọc của 1 trong hai giao tác không đúng như mong đợi là do tình trạng **LOST UPDATE**.

B. DIRTY READ	
Tình huống 1:	
T1 Đối tác cập nhật ngày kết thúc hợp đồng	T2 Nhân viên xem tình trạng hợp đồng của đối tác đó
Input: @MAHD CHAR(5), @NGAYCAPNHAT DATE	Input: @MAHD CHAR(5), @MADT CHAR(5)
BEGIN TRAN	
DECLARE @KTRANGAY DATE SET @KTRANGAY = (SELECT NGAYKT FROM HOPDONG WHERE MAHD = @MAHD)	
UPDATE HOPDONG SET NGAYKT = @NGAYCAPNHAT WHERE MAHD = @MAHD	
WAITFOR DELAY '00:00:10'	
	BEGIN TRAN
	SELECT * FROM HOPDONG WITH(NOLOCK) WHERE MAHD = @MAHD AND MADT = @MADT
	COMMIT TRAN
IF (@KTRANGAY > @NGAYCAPNHAT) BEGIN ROLLBACK TRAN RETURN END	
COMMIT TRAN	

Khi đối tác cập nhật thời gian hiệu lực của hợp đồng của mình và nhân viên xem tình trạng hợp đồng của đối tác cùng lúc. Nếu ngày cập nhật bé hơn ngày kết thúc của hợp đồng thì T1 thì ROLLBACK nhưng lúc này T2 vẫn lấy kết quả nên bị lỗi.

Ví dụ: đối tác 541 cập nhật ngày kết thúc của hợp đồng HD469 và nhân viên xem hợp đồng HD469. Và ban đầu ngày kết thúc của hợp đồng HD469 là 2023-04-30.

→T1

```
--T1
USE [QL_DH_GH_ONLINE]
GO

CREATE OR ALTER PROCEDURE CAPNHAT_HOPDONG @MAHD CHAR(5), @NGAYCAPNHAT DATE
AS
BEGIN TRAN
    DECLARE @KTRANGAY DATE
    SET @KTRANGAY = (SELECT NGAYKT FROM HOPDONG WHERE MAHD = @MAHD)
    UPDATE HOPDONG
    SET NGAYKT = @NGAYCAPNHAT
    WHERE MAHD = @MAHD

    WAITFOR DELAY '00:00:10'
    IF (@KTRANGAY > @NGAYCAPNHAT)
    BEGIN
        ROLLBACK TRAN
        RETURN
    END
    COMMIT TRAN
GO
EXEC CAPNHAT_HOPDONG 'HD469', '2021-03-09'
```

121 %

Messages

(1 row affected)

Completion time: 2023-04-16T22:23:23.6648302+07:00

→T2

```
--T2
USE [QL_DH_GH_ONLINE]
GO

CREATE OR ALTER PROCEDURE XEM_DS_MAHM @MAHD CHAR(5), @MADT CHAR(5)
AS
BEGIN TRAN
    SELECT * FROM HOPDONG WITH(NOLOCK) WHERE MAHD = @MAHD AND MADT = @MADT
COMMIT TRAN
GO
EXEC XEM_DS_MAHM 'HD469', 'DT541'
```

21 %

Results Messages

	MAHD	MADT	MACN	NGAYKY	NGAYKT	PHANTRAMHOAHONG	TINHTRANGHD
1	HD469	DT541	CN654	2020-03-15	2021-03-09	10	Còn hiệu lực

Ngày kết thúc của hợp đồng sau khi 2 giao tác thực hiện:

HD469	DT541	CN654	2020-03-15	2023-04-30	10	Còn hiệu lực
-------	-------	-------	------------	------------	----	--------------

Vì đối tác nhập ngày cập nhật bé hơn ngày kết thúc của hợp đồng nên giao tác được ROLLBACK, trong lúc đó nhân xem hợp đồng chưa được commit. Do đó dẫn đến lỗi **DIRTY READ**.

DIRTY READ	
Tình huống 2:	
T1	T2
Đối tác cập nhật giá sản phẩm	Khách hàng xem giá sản phẩm
Input: @MASP CHAR(5), @GIA MONEY	Input: @MASP CHAR(5)
BEGIN TRAN	
DECLARE @GIASP MONEY SET @GIASP = (SELECT DONGIA FROM SANPHAM WHERE MASP = @MASP)	
UPDATE SANPHAM SET DONGIA = @GIA WHERE MASP = @MASP	
WAITFOR DELAY '00:00:10'	
	BEGIN TRAN
	SELECT * FROM SANPHAM WITH(NOLOCK) WHERE MASP = @MASP
	COMMIT TRAN
IF (@GIA >= 5 * @GIASP) BEGIN ROLLBACK TRAN RETURN END	
COMMIT TRAN	

Khi đối tác cập nhật giá của sản phẩm và khách hàng xem giá sản phẩm cùng lúc. Nếu giá cập nhật lớn hơn 5 lần giá sản phẩm thì T1 thì ROLLBACK nhưng lúc này T2 vẫn lấy kết quả nên bị lỗi.

Ví dụ: đối tác cập nhật giá sản phẩm SP015 và khách hàng xem giá sản phẩm SP015. Và ban đầu giá của SP015 là 2000.

→T1

```
--DIRTY READ
--T1
USE [QL_DH_GH_ONLINE]
GO

CREATE OR ALTER PROCEDURE DT_CAPNHAT_GIASP @MASP CHAR(5), @GIA MONEY
AS
BEGIN TRAN
    DECLARE @GIASP MONEY
    SET @GIASP = (SELECT DONGIA FROM SANPHAM WHERE MASP = @MASP)
    UPDATE SANPHAM
        SET DONGIA = @GIA
        WHERE MASP = @MASP
    WAITFOR DELAY '00:00:10'

    IF (@GIA >= 5 * @GIASP)
    BEGIN
        ROLLBACK TRAN
        RETURN
    END

    COMMIT TRAN
GO
EXEC DT_CAPNHAT_GIASP 'SP015', 200000
```

21 %

Messages

(1 row affected)

Completion time: 2023-04-16T14:26:39.5651195+07:00

→T2

```

--DIRTY READ
--T2
USE [QL_DH_GH_ONLINE]
GO

CREATE OR ALTER PROCEDURE KH_XEMGIA_SP @MASP CHAR(5)
AS
BEGIN TRAN
    SELECT * FROM SANPHAM WITH(NOLOCK)
    WHERE MASP = @MASP
COMMIT TRAN
GO

EXEC KH_XEMGIA_SP 'SP015'

```

121 %

Results Messages

	MASP	MADT	TENSP	SLTON	DONGIA	MATD
1	SP015	DT125	Chuối	66	200000.00	NULL

Giá của sản phẩm sau khi 2 giao tác thực hiện:

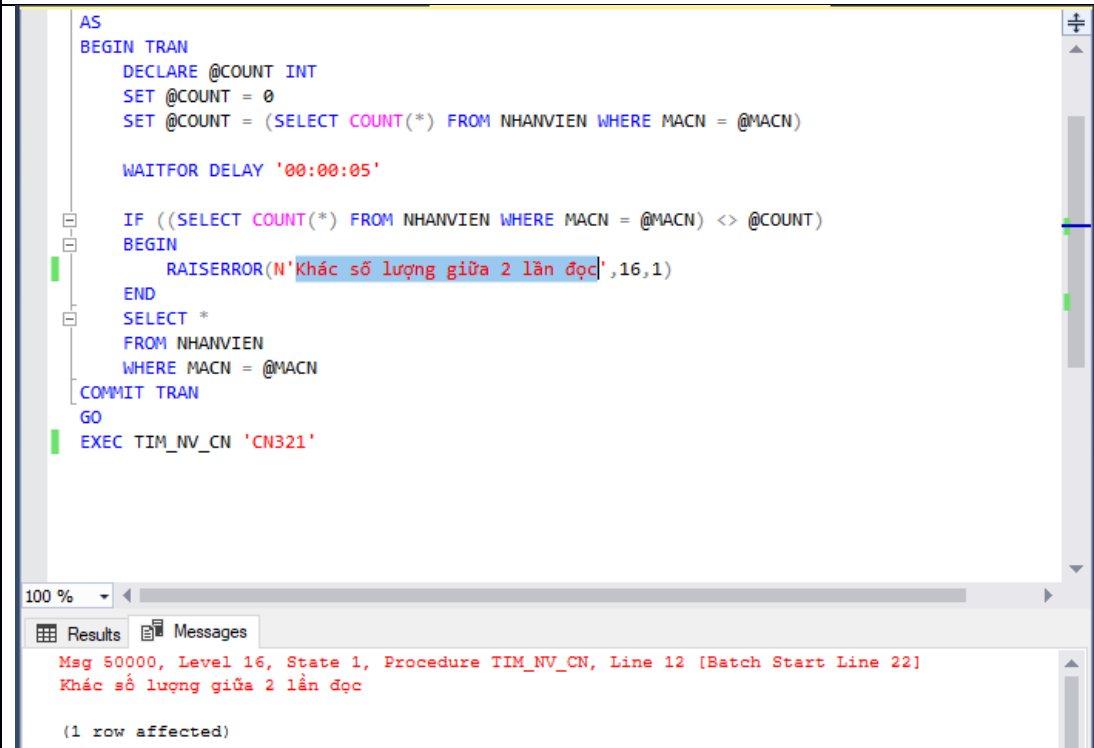
SP015	DT125	Chuối	66	2000.00
-------	-------	-------	----	---------

- ➔ Vì đối tác nhập giá sản phẩm lớn hơn 5 lần giá gốc nên giao tác được ROLLBACK, trong lúc đó khách hàng xem giá sản phẩm chưa được commit. Do đó dẫn đến lỗi **DIRTY READ**.

C. UNREPEATABLE READ

Tình huống 1:

T1	T2
Quản lý muốn tìm thông tin tất cả nhân viên ở chi nhánh X. Input: @MACN CHAR(5)	Nhân viên chi nhánh Y đổi chi nhánh làm việc. Input: @MANV CHAR(5), @MACN CHAR(5)
BEGIN TRAN	
DECLARE @COUNT INT SET @COUNT = 0 SET @COUNT = (SELECT COUNT(*) FROM NHANVIEN WHERE MACN = @MACN)	

WAITFOR DELAY '00:00:05'	
	BEGIN TRAN
	UPDATE NHANVIEN SET MACN = @MACN WHERE MANV = @MANV
	COMMIT TRAN
IF ((SELECT COUNT(*) FROM NHANVIEN WHERE MACN = @MACN) <> @COUNT) BEGIN RAISERROR(Khác số lượng giữa 2 lần đọc',16,1) END SELECT * FROM NHANVIEN WHERE MACN = @MACN	
COMMIT TRAN	
<p>Tình huống giả định: Quản lý hệ thống muốn thống kê tìm danh sách các nhân viên của chi nhánh 'CN321'. Cùng lúc, 'NV313' chi nhánh 'CN321' được điều động đến chi nhánh 'CN301' để làm việc. Điều này được minh họa qua kết quả sau:</p> <p><input type="checkbox"/> T1</p>	
 <pre> AS BEGIN TRAN DECLARE @COUNT INT SET @COUNT = 0 SET @COUNT = (SELECT COUNT(*) FROM NHANVIEN WHERE MACN = @MACN) WAITFOR DELAY '00:00:05' IF ((SELECT COUNT(*) FROM NHANVIEN WHERE MACN = @MACN) <> @COUNT) BEGIN RAISERROR(N'Khác số lượng giữa 2 lần đọc',16,1) END SELECT * FROM NHANVIEN WHERE MACN = @MACN COMMIT TRAN GO EXEC TIM_NV_CN 'CN321' </pre> <p>Results Messages</p> <p>Msg 50000, Level 16, State 1, Procedure TIM_NV_CN, Line 12 [Batch Start Line 22] Khác số lượng giữa 2 lần đọc (1 row affected)</p> <p><input type="checkbox"/> T2</p>	

```

/*UNREPEATABLE READ*/
/*T2*/
CREATE OR ALTER PROCEDURE CAPNHAT_MACN_NV @MANV CHAR(5), @MACN CHAR(5)
AS
BEGIN TRAN
    UPDATE NHANVIEN
    SET MACN = @MACN
    WHERE MANV = @MANV
COMMIT TRAN
GO
EXEC CAPNHAT_MACN_NV 'NV313', 'CN301'

```

100 %

Messages

(1 row affected)

KẾT QUẢ: Khi giao tác T1 thực hiện trước, nó sẽ xuất ra danh sách các nhân viên ở chi nhánh 'CN321'. Tại thời điểm này, giao tác T2 bắt đầu và nó cập nhật nhân viên 'NV313' chi nhánh 'CN321' thành chi nhánh 'CN301'. Sau đó T1 đọc lại và đã xảy ra lỗi khi 2 lần đọc đều cho ra kết quả số lượng khác nhau đó gọi là Unrepeatable read.

Tình huống 2:

T1	T2
Khách hàng kiểm tra tình trạng giao hàng của đơn hàng. Input: @MAKH CHAR(5), @MADH CHAR(5)	Tài xế cập nhật tình trạng giao hàng. Input: @MATX CHAR(5), @MADH CHAR(5), @TTGH NVARCHAR(40)

BEGIN TRAN	
<pre> IF NOT EXISTS(SELECT DGH.* FROM DONGIAOHANG DGH, DONHANG DH WHERE DGH.MADH = DH.MADH AND DH.MAKH = @MAKH AND DGH.MADH = @MADH) BEGIN PRINT N'DH không tồn tại' RETURN END </pre>	
<pre> DECLARE @TTGH NVARCHAR(40) SET @TTGH = 0 SET @TTGH = (SELECT TTGIAOHANG FROM </pre>	

DONGIAOHANG WHERE MADH = @MADH)	
WAITFOR DELAY '00:00:05'	
	BEGIN TRAN
	UPDATE DONGIAOHANG SET TTGIAOHANG = @TTGH WHERE MADH = @MADH AND MATX = @MATX
	COMMIT TRAN
IF ((SELECT TTGIAOHANG FROM DONGIAOHANG WHERE MADH = @MADH) <> @TTGH) BEGIN RAISERROR(N'2 lần đọc dữ liệu khác nhau',16,1) END SELECT* FROM DONGIAOHANG WHERE MADH = @MADH	
COMMIT TRAN	
<p>Tình huống giả định: Khách hàng 'KH249' thực hiện kiểm tra tình trạng giao hàng của đơn hàng 'DH001'. Đồng thời, cùng lúc đó, tài xế cập nhật tình trạng giao hàng 'DH001'.</p> <p><input type="checkbox"/> T1</p>	

```

SET @TTGH = (SELECT TTGIAOHANG FROM DONGIAOHANG WHERE MADH = @MADH)
WAITFOR DELAY '00:00:05'

IF ((SELECT TTGIAOHANG FROM DONGIAOHANG WHERE MADH = @MADH) <> @TTGH)
BEGIN
    RAISERROR(N'2 lần đọc dữ liệu khác nhau',16,1)
END
SELECT *
FROM DONGIAOHANG
WHERE MADH = @MADH
COMMIT TRAN
GO
EXEC KTRA_TTGH 'KH165', 'DH016'

```

100 %

Results Messages

Msg 50000, Level 16, State 1, Procedure KTRA_TTGH, Line 17 [Batch Start Line 27]
2 lần đọc dữ liệu khác nhau

(1 row affected)

□ T2

```

USE [QL_DH_GH_ONLINE]
GO
/*NON-REPEATABLE READ*/
/*T2*/
CREATE OR ALTER PROCEDURE TX_CAPNHAT_TTGH @MATX CHAR(5), @MADH CHAR(5), @TTGH NVARCHAR(40)
AS
BEGIN TRAN
    UPDATE DONGIAOHANG
    SET TTGIAOHANG = @TTGH
    WHERE MADH = @MADH AND MATX = @MATX
COMMIT TRAN
GO
EXEC TX_CAPNHAT_TTGH 'TX014', 'DH016', N'Đã vận chuyển'

```

100 %

Messages

(1 row affected)

KẾT QUẢ: Khi giao tác T1 thực hiện trước, nó sẽ xuất ra hợp đồng với mã 'DH016' với tình trạng là 'Đang vận chuyển'. Tại thời điểm này, giao tác T2 bắt đầu và nó cập nhật tình trạng HĐ của hợp đồng 'DH016' là 'Đã vận chuyển'. Sau đó, T1 thực hiện đọc lại và xảy ra lỗi đây là lỗi Unrepeatable read.

D. PHANTOM

Tình huống 1:

T1	T2
Chức năng : Quản lý xem các thông tin tài khoản người dùng có trong hệ thống	Chức năng: Khách hàng đăng ký tài khoản mới vào hệ thống
BEGIN TRAN	
DECLARE @COUNT INT SET @COUNT = 0 SET @COUNT = (SELECT COUNT(*) FROM LOGIN) WAITFOR DELAY '00:00:10'	

	BEGIN TRAN
	INSERT INTO LOGIN(USERID, USERNAME,PASSWORD,ROLE_USER) VALUES (@USERID,@USERNAME, @PW,@ROLE)
	COMMIT TRAN
IF ((SELECT COUNT(*) FROM LOGIN) <> @COUNT) BEGIN RAISERROR(N'Dữ liệu hai lần đọc khác nhau',16,1) END	
SELECT * FROM LOGIN	
COMMIT TRAN	
Ví dụ mẫu: Quản lý truy cập vào hệ thống để xem danh sách tài khoản người dùng thì cùng lúc đó có một khách hàng muốn đăng ký tài khoản để có thể mua hàng trên hệ thống.	
➔ T1	


```
USE [QL_DH_GH_ONLINE]
GO
/*PHANTOM TH1*/
/* T1 */
CREATE OR ALTER PROCEDURE XEM_DS_LOGIN
AS
BEGIN TRAN
    DECLARE @COUNT INT
    SET @COUNT = 0
    SET @COUNT = (SELECT COUNT(*) FROM LOGIN)
    WAITFOR DELAY '00:00:10'

    IF ((SELECT COUNT(*) FROM LOGIN) <> @COUNT)
    BEGIN
        RAISERROR(N'Dữ liệu hai lần đọc khác nhau',16,1)
    END
    SELECT * FROM LOGIN
COMMIT TRAN
GO
EXEC XEM_DS_LOGIN
```

133 %

Results Messages

Msg 50000, Level 16, State 1, Procedure XEM_DS_LOGIN, Line 13 [Batch Start Line 19]
Dữ liệu hai lần đọc khác nhau

(17 rows affected)

Completion time: 2023-04-13T20:54:01.2277496+07:00

→ T2

```
USE [QL_DH_GH_ONLINE]
GO
/*PHANTOM TH1*/
/* T2 */
EXEC SP_INSERT_LOGIN 'KH008', 'khachhang8', '123', N'Khách hàng'
```

133 %

Messages

(1 row affected)

Completion time: 2023-04-13T20:53:54.0233529+07:00

KẾT QUẢ: Khi T1 được thực hiện, đọc dữ liệu trên bảng LOGIN. Thì tại thời điểm đó, T2 cũng được thực hiện và nó chèn một dữ liệu mới vào bảng. Và giao tác T1 khi đọc dữ liệu lại. Tuy nhiên, kết quả 2 lần đọc lại khác nhau. Việc đọc dữ liệu ra 2 kết quả khác nhau làm cho người sử dụng không Xác định được dữ liệu đúng hay sai. Đây được gọi là PHANTOM, tức là dữ liệu bóng ma, khi mà một giao tác thực thực hiện hai lần truy vấn đọc thì giao tác hai lại chèn một dữ liệu mới vào giữa 2 lần đọc

PHANTOM	
Tình huống 2	
T1	T2
Chức năng: Nhân viên xem danh sách hợp đồng của chi nhánh	Chức năng: Đối tác thêm hợp đồng mới vào chi nhánh
BEGIN TRAN	
DECLARE @COUNT INT SET @COUNT = 0 SET @COUNT = (SELECT COUNT(*) FROM HOPDONG WHERE MACN = @MACN) WAITFOR DELAY '00:00:10'	
	BEGIN TRAN
	INSERT INTO HOPDONG(MADT,MACN,NGAYKY,NGAYKT,PHANTRAMHOAHONG, TINHTRANGHD) VALUES (@MADT,@MACN,@NGAYKY,@NGAYKT,@PTHOAHONG, N'Đang chờ duyệt')
	IF (@NGAYKY > @NGAYKT) BEGIN PRINT N'NGÀY KÝ VÀ NGÀY KT KHÔNG HỢP LỆ' ROLLBACK TRAN END
	COMMIT TRAN
IF ((SELECT COUNT(*) FROM HOPDONG WHERE MACN = @MACN) <>	

<pre> @COUNT) BEGIN RAISERROR(N 'Dữ liệu hai lần đọc khác nhau',16,1) END </pre>	
<pre> SELECT * FROM HOPDONG WHERE MACN = @MACN </pre>	
<pre> COMMIT TRAN </pre>	

Ví dụ mẫu: Nhân viên kiểm tra danh sách hợp đồng của chi nhánh “CN184”. Ngay lúc đó đối tác thực hiện việc lập hợp đồng với chi nhánh “CN184”

→ T1:

```

USE [QL_DH_GH_ONLINE]
GO
/*PHANTOM - TH2*/
/* T1 */
CREATE OR ALTER PROCEDURE XEM_DSHD @MACN CHAR(5)
AS
BEGIN TRAN
    DECLARE @COUNT INT
    SET @COUNT = 0
    SET @COUNT = (SELECT COUNT(*) FROM HOPDONG WHERE MACN = @MACN)
    WAITFOR DELAY '00:00:10'

    IF ((SELECT COUNT(*) FROM HOPDONG WHERE MACN = @MACN) <> @COUNT)
    BEGIN
        RAISERROR(N'Dữ liệu hai lần đọc khác nhau',16,1)
    END
    SELECT * FROM HOPDONG WHERE MACN = @MACN
COMMIT TRAN
GO
EXEC XEM_DSHD 'CN184'

```

133 %

Results Messages

Msg 50000, Level 16, State 1, Procedure XEM_DSHD, Line 13 [Batch Start Line 19]
Dữ liệu hai lần đọc khác nhau

(4 rows affected)

Completion time: 2023-04-13T22:12:55.7013713+07:00

→ T2

```

USE [QL_DH_GH_ONLINE]
GO
/*PHANTOM - TH2*/
/* T2 */
CREATE OR ALTER PROCEDURE THEM_HD @MADT CHAR(5), @MACN CHAR(5), @NGAYKY DATE, @NGAYKT DATE, @PTHOAHONG FLOAT
AS
BEGIN TRAN
    INSERT INTO HOPDONG(MADT,MACN,NGAYKY,NGAYKT,PHANTRAMHOAHONG, TINHTRANGHD)
    VALUES (@MADT,@MACN,@NGAYKY,@NGAYKT,@PTHOAHONG, N'Đang chờ duyệt')

    IF (@NGAYKY > @NGAYKT)
    BEGIN
        PRINT N'NGÀY KÝ VÀ NGÀY KT KHÔNG HỢP LỆ'
        ROLLBACK TRAN
    END
    COMMIT TRAN
GO
EXEC THEM_HD 'DT195', 'CN184', '2023-03-16', '2024-03-16', 0

```

133 %

Messages

(1 row affected)

Completion time: 2023-04-13T22:12:48.2347009+07:00

KẾT QUẢ: Khi T1 được thực hiện để nhận viên xem danh sách hợp đồng của chi nhánh “CN184” trong bảng HOPDONG. Thì tại thời điểm đó, T2 thực hiện hành động thêm một hợp đồng mới vào chi nhánh “CN184” và sau đó T1 đọc lại dữ liệu. Tuy nhiên, kết quả 2 lần đọc lại khác nhau. Việc đọc dữ liệu ra 2 kết quả khác nhau làm cho người sử dụng không xác định được dữ liệu đúng hay sai. Đây được gọi là PHANTOM, tức là dữ liệu bóng ma, khi mà một giao tác thực thực hiện hai lần truy vấn đọc thì giao tác hai lại chèn một dữ liệu mới vào giữa 2 lần đọc

E. CYCLE DEADLOCK

T1	T2
<p>Quản lý 1 cập nhật thông tin tài khoản của khách hàng (đổi mật khẩu) và cập nhật thông tin của tài xế (CMND).</p> <p>Input: @USERID CHAR(5), @PASS_NEW CHAR(10), @MATX CHAR(5), @CMND_NEW CHAR(10)</p>	<p>Quản lý 2 cập nhật thông tin của tài xế (CMND) và cập nhật thông tin tài khoản của khách hàng (đổi mật khẩu).</p> <p>Input: @MATX CHAR(5), @CMND_NEW CHAR(10), @USERID CHAR(5), @PASS_NEW CHAR(10)</p>
BEGIN TRAN	
UPDATE LOGIN SET PASSWORD = @PASS_NEW WHERE USERID = @USERID	
WAITFOR DELAY '00:00:05'	
	BEGIN TRAN
	<p>UPDATE TAIXE SET CMND = @CMND_NEW WHERE MATX = @MATX WAITFOR DELAY '00:00:05'</p>

	<pre>UPDATE LOGIN SET PASSWORD = @PASS_NEW WHERE USERID = @USERID</pre>
<pre>UPDATE TAIXE SET CMND = @CMND_NEW WHERE MATX = @MATX</pre>	
COMMIT TRAN	
	COMMIT TRAN
<p>Tình huống giả định: Giao tác T1 và giao tác T2 thực hiện xen kẽ các giao tác đã nêu trong bảng:</p> <p><input type="checkbox"/> T1</p> <pre> /* ^ I1 ^ */ CREATE PROCEDURE QL1_DEADLOCK @USERID CHAR(5), @PASS_NEW CHAR(10), @MATX CHAR(5), @CMND_NEW AS BEGIN TRAN UPDATE LOGIN SET PASSWORD = @PASS_NEW WHERE USERID = @USERID WAITFOR DELAY '00:00:05' UPDATE TAIXE SET CMND = @CMND_NEW WHERE MATX = @MATX COMMIT TRAN GO /*Demo deadlock*/ EXEC QL1_DEADLOCK 'KH001', '1234', 'TX001', '20126044'</pre> <p>100 %</p> <p>Messages</p> <p>(1 row affected)</p> <p>Msg 1205, Level 13, State 51, Procedure QL1_DEADLOCK, Line 9 [Batch Start Line 17] Transaction (Process ID 52) was deadlocked on lock resources with another process and has been</p> <p><input type="checkbox"/> T2</p>	

```

/*CYCLE DEADLOCK*/
/* T2 */
CREATE PROCEDURE QL2_DEADLOCK @USERID CHAR(5), @PASS_NEW CHAR(10), @MATX CHAR(5), @CMND_NEW
AS
BEGIN TRAN
    UPDATE TAIXE
    SET CMND = @CMND_NEW
    WHERE MATX = @MATX
    WAITFOR DELAY '00:00:05'

    UPDATE LOGIN SET PASSWORD = @PASS_NEW WHERE USERID = @USERID
COMMIT TRAN
GO

/*Demo deadlock*/
EXEC QL2_DEADLOCK 'KH001', '1234', 'TX001', '20126044'

```

0 %

Messages

(1 row affected)

(1 row affected)

KẾT QUẢ: Giao tác thứ nhất T1 thực hiện hành động đổi mật khẩu của KH001, tức là nó xin khóa XL trên bảng LOGIN, và trì hoãn 5s. Cùng lúc T2 cập nhật thông tin tài xế (CMND), tức là xin khóa XL trên bảng TAIXE. Sau đó T1 tiếp tục thực hiện cập nhật thông tin TX001, nhưng T2 đang giữ khóa XL trên bảng TAIXE nên T1 chờ T2, và sau đó T2 thực hiện đổi mật khẩu KH001, nhưng vì T1 đang giữ khóa XL trên bảng LOGIN nên T2 phải chờ T1. Hai giao tác phải chờ lẫn nhau dẫn đến không thể kết thúc được nên xảy ra tình trạng cycle deadlock.

F. CONVERSION DEADLOCK

T1	T2
Chức năng : Nhân viên 1 chi nhánh x cập nhật tình trạng hợp đồng	Chức năng : Nhân viên 2 chi nhánh x cập nhật tình trạng hợp đồng
BEGIN TRAN	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ	
SELECT *	

FROM HOPDONG WHERE MAHD = @MAHD WAITFOR DELAY '00:00:05'															
	BEGIN TRAN														
	SET TRANSACTION ISOLATION LEVEL REPEATABLE READ														
	SELECT * FROM HOPDONG WHERE MAHD = @MAHD WAITFOR DELAY '00:00:05'														
UPDATE HOPDONG SET TINHTRANGHD = @TTHD WHERE MAHD = @MAHD															
COMMIT TRAN															
	UPDATE HOPDONG SET TINHTRANGHD = @TTHD WHERE MAHD = @MAHD														
	COMMIT TRAN														
<div><pre>GO EXEC SP_INSERT_LOGIN 'NV283', 'nv283', '123', N'Nhân viên' EXEC SP_INSERT_LOGIN 'NV646', 'nv646', '123', N'Nhân viên' GO EXEC CPNHAT_TTHDONG 'HD146', N'Hết hạn'</pre></div> <div><div>Results</div><div>Messages</div><table><tr><th>MAHD</th><th>MADT</th><th>MACN</th><th>NGAYKY</th><th>NGAYKT</th><th>PHANTRAMHOAHONG</th><th>TINHTRANGHD</th></tr><tr><td>HD146</td><td>DT339</td><td>CN405</td><td>2020-02-21</td><td>2023-12-13</td><td>45.4695487373832</td><td>Còn hiệu lực</td></tr></table></div>		MAHD	MADT	MACN	NGAYKY	NGAYKT	PHANTRAMHOAHONG	TINHTRANGHD	HD146	DT339	CN405	2020-02-21	2023-12-13	45.4695487373832	Còn hiệu lực
MAHD	MADT	MACN	NGAYKY	NGAYKT	PHANTRAMHOAHONG	TINHTRANGHD									
HD146	DT339	CN405	2020-02-21	2023-12-13	45.4695487373832	Còn hiệu lực									
<p>KẾT QUẢ :</p> <p>Giao tác 1: Lần đọc thứ nhất đọc bảng HOPDONG và xin khóa SL. Ngay thời điểm đó, ở giao tác 2 tiếp tục đọc bảng HOPDONG và xin khóa SL. Sau đó, T1 cập nhật trên bảng HOPDONG nhưng vì T2 đang giữ SL nên T1 chờ, tương tự với T2, T2 cũng cập nhật trên bảng HOPDONG nên T2 cũng xin khóa XL nhưng T1 đang giữ khóa SL nên T2 chờ T1. Tạo ra tình trạng 2 giao tác chờ nhau. Vậy nên đã xảy ra tình trạng conversion deadlock</p>															
<p>Ví dụ mẫu 2: Có 2 nhân viên của cùng 1 chi nhánh cùng cập nhật hợp đồng của hợp đồng « HD146 ». Nhân viên 1 cập nhật hợp đồng là hết hạn. Nhân viên 2 cập nhật hợp đồng lại là có hiệu lực</p> <p>➔ T1</p>															

```

USE [QL_DH_GH_ONLINE]
GO
/*CONVERSION DEADLOCK*/
/* T1 */
CREATE OR ALTER PROCEDURE CPNHAT_TTHDONG @MAHD CHAR(5), @TTHD NVARCHAR(40)
AS
BEGIN TRAN
    SET TRANSACTION ISOLATION
    LEVEL REPEATABLE READ
    SELECT *
    FROM HOPDONG
    WHERE MAHD = @MAHD
    WAITFOR DELAY '00:00:10'

    UPDATE HOPDONG
    SET TINHTRANGHD = @TTHD
    WHERE MAHD = @MAHD
COMMIT TRAN
GO
EXEC SP_INSERT_LOGIN 'NV283', 'nv283', '123', N'Nhân viên'
--EXEC SP_INSERT_LOGIN 'NV646', 'nv646', '123', N'Nhân viên'
GO
EXEC CPNHAT_TTHDONG 'HD146', N'Hết hạn'

```

133 %

Results Messages

	MAHD	MADT	MACN	NGAYKY	NGAYKT	PHANTRAMHOAHONG	TINHTRANGHD
1	HD146	DT339	CN405	2020-02-21	2023-12-13	45.4695487373832	Hết hạn

➔ T2

```

USE [QL_DH_GH_ONLINE]
GO
/*CONVERSION DEADLOCK*/
/* T2 */
EXEC CPNHAT_TTHDONG 'HD146', N'Còn hiệu lực'

```

133 %

Results Messages

(1 row affected)
 Msg 1205, Level 13, State 51, Procedure CPNHAT_TTHDONG, Line 13 [Batch Start Line 2]
 Transaction (Process ID 54) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.
 Completion time: 2023-04-13T23:51:55.5738244+07:00

KẾT QUẢ: Như kết quả trên 2 giao tác đã xảy ra tình trạng **conversion deadlock**

2. Xử lý tranh chấp:

A. LOST UPDATE			
Trường hợp 1:			
T1		T2	
BEGIN TRAN			
SET TRAN ISOLATION LEVEL REPEATABLE READ			
DECLARE @SLTON INT SET @SLTON = (SELECT SLTON FROM SANPHAM WITH(UPDLOCK) WHERE MASP = @MASP)	UL(SANPHAM) T1 xin khóa tăng cấp đọc trên bảng SANPHAM ứng với MASP = @MASP		
IF (@SLTON = 0) BEGIN <div style="color: red;">PRINT</div> <div style="color: red;">N'Số lượng hàng trong kho đã hết'</div> ROLLBACK TRAN END			
IF (@SLMUA > @SLTON) BEGIN <div style="color: red;">PRINT N'Số lượng hàng trong kho không đủ đáp ứng với số lượng mua'</div> ROLLBACK TRAN END			
WAITFOR DELAY <div style="color: red;">'00:00:05'</div>			
		BEGIN TRAN	
		SET TRAN ISOLATION LEVEL REPEATABLE READ	
		DECLARE @SLTON INT	UL(SAN PHAM)

		<pre> SET @SLTON = (SELECT SLTON FROM SANPHAM WITH(XLOCK) WHERE MASP = @MASP) </pre>	T1 xin khóa tăng cấp để đọc trên bảng SANPHAM ứng với MASP = @MASP
		<pre> IF (@SLTON = 0) BEGIN PRINT N'Số lượng hàng trong kho đã hết' ROLLBACK TRAN END </pre>	
		<pre> IF (@SLMUA > @SLTON) BEGIN PRINT N'Số lượng hàng trong kho không đủ đáp ứng với số lượng mua' ROLLBACK TRAN END </pre>	
<pre> INSERT [dbo].[CTDONHANG] ([MASP], [MADH], [SLSANPHAM]) VALUES (@MASP, @MADH, @SLMUA) </pre>			
<pre> UPDATE SANPHAM SET SLTON = @SLTON - @SLMUA WHERE MASP = @MASP </pre>	<pre> XL(SAN PHAM) T1 chuyển thành khóa ghi trên bảng SANPHAM ứng với MASP = @MASP </pre>		
COMMIT TRAN			
		<pre> WAITFOR DELAY '00:00:05' </pre>	
		<pre> INSERT [dbo].[CTDONHANG] ([MASP], [MADH], [SLSANPHAM]) VALUES (@MASP, @MADH, @SLMUA) </pre>	

		UPDATE SANPHAM SET SLTON = @SLTON - @SLMUA WHERE MASP = @MASP	XL(SAN PHAM) T1 xin khóa ghi để ghi trên bảng SANPHAM ứng với MASP = @MASP
		COMMIT TRAN	

KẾT QUẢ: Đối với lỗi LOST UPDATE ở trên, nhóm đề xuất áp dụng mức cô lập **REPEATABLE READ** ở cả 2 giao tác để tránh tình trạng giao tác này đang thực thi, giao tác khác lại chen ngang. Tuy nhiên, khi áp dụng mức cô lập **REPEATABLE READ**, thì các giao tác phải chờ lẫn nhau → Xảy ra tình trạng **DEADLOCK** (cụ thể là **CONVERSION DEADLOCK**).

Chỉ áp dụng mức cô lập **REPEATABLE READ**:

```

CREATE OR ALTER PROCEDURE KHI_HUASP_FIX @MAKH CHAR(5), @MASP CHAR(5), @SLMUA INT
AS
BEGIN TRAN
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
DECLARE @SLTON INT
SELECT @SLTON = SLTON FROM SANPHAM WHERE MASP = @MASP
PRINT @SLTON
PRINT @SLMUA
IF (@SLTON = 0)
BEGIN
PRINT N'Số lượng hàng trong kho đã hết'
ROLLBACK TRAN
END
IF (@SLMUA > @SLTON)
BEGIN
PRINT N'Số lượng hàng trong kho không đủ đáp ứng với số lượng mua'
ROLLBACK TRAN
END
WAITFOR DELAY '00:00:10'
INSERT [dbo].[CTDOANHANG] ([MASP], [MAKH], [SLSANPHAM]) VALUES (@MASP, @MAKH, @SLMUA)
UPDATE SANPHAM
SET SLTON = @SLTON - @SLMUA
WHERE MASP = @MASP
COMMIT TRAN
GO
EXEC KHI_HUASP_FIX 'DH001', 'SP043', 6

```

Messages

```

(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
Completion time: 2023-04-14T17:08:58.8248713+07:00

```

Query executed successfully.

```

SQLQuery2.sql - KH-M-ONLINE (sa (72))    SQLQuery2.sql - KH-M-ONLINE (sa (72))    Tran_1_Update_M-ONLINE (sa (51))    Tran_2_Update_M-ONLINE (sa (52))
SELECT @SLTON = SLTON FROM SANPHAM WHERE MASP = @MASP
PRINT @SLTON
IF (@SLTON = 0)
BEGIN
    PRINT N'Số lượng hàng trong kho đã hết'
    ROLLBACK TRAN
END
IF (@SLTON > @SLTON)
BEGIN
    PRINT N'Số lượng hàng trong kho không đủ đáp ứng với số lượng mua'
    ROLLBACK TRAN
END
WAITFOR DELAY '00:00:10'
INSERT [dbo].[CTDONGHANG] ([MASP], [HACH], [SLSANPHAM]) VALUES (@MASP, @HACH, @SLMUA)
UPDATE SANPHAM
SET SLTON = @SLTON - @SLMUA
WHERE MASP = @MASP
COMMIT TRAN
GO
EXEC KH2_HUASP_FIX 'DH082', 'SP043', 5

```

99 % -

Messages

62

Msg 1205, Level 13, State 61, Procedure Dstfang, Line 14 (Batch Start Line 31): Transaction (Process ID 52) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

Completion time: 2023-04-14T17:08:58.760012407:00

99 % -

Query completed with errors.

KHEEM (16.0 RTM) sa (52) QL_DH_GH_ONLINE 00:00:10 0 rows

→ Một trong 2 giao tác đã xuất hiện DEADLOCK.

Vì thế nhóm em đề xuất thêm áp dụng khóa UPDLOCK đối với giao tác T1 và giao tác T2. Kết quả, đã giải quyết được tình trạng **dirty read**, cũng như **conversion deadlock**. Áp dụng khóa UPDLOCK trên T1 để có thể đọc trên bảng SANPHAM, sau đó, sẽ được chuyển thành khóa XL để ghi trên bảng SANPHAM, khóa UPDLOCK sẽ giữ cho đến khi giao tác T1 kết thúc. Đối với T2 cũng được thực hiện tương tự. Vì T1 và T2 đều thực hiện trên cùng 1 ĐVDL nên T2 phải chờ T1 commit thì T2 mới được thực hiện.

Kết quả thực hiện:

```

--//T1//
CREATE OR ALTER PROCEDURE KHI_HUASP_FIX @HACH CHAR(5), @HUASP CHAR(5), @SLHUA INT
AS
BEGIN TRAN
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
DECLARE @SLTON INT
SELECT @SLTON = SLTON FROM SANPHAM WITH(UPDLOCK) WHERE HUASP = @HUASP
PRINT @SLTON
IF (@SLTON = 0)
BEGIN
PRINT N'Số lượng hàng trong kho đã hết'
ROLLBACK TRAN
END
IF (@SLHUA > @SLTON)
BEGIN
PRINT N'Số lượng hàng trong kho không đủ đáp ứng với số lượng mua'
ROLLBACK TRAN
END
WAITFOR DELAY '00:00:10'
INSERT [dbo].[CTDOANHANG] ([HUASP], [HACH], [SLSANPHAM]) VALUES (@HUASP, @HACH, @SLHUA)
UPDATE SANPHAM
SET SLTON = @SLTON - @SLHUA
WHERE HUASP = @HUASP
COMMIT TRAN
GO
EXEC KHI_HUASP_FIX 'DH014', 'SP014', 6

```

99 %
Messages
36
6
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
Completion time: 2023-04-14T16:10:14.5584725+07:00
Query executed successfully. KHIEM (16.0 RTM) sa (51) QL_DH_GH_ONLINE 00:00:10 0 rows

```

--//T2//
CREATE OR ALTER PROCEDURE KHI_HUASP_FIX @HACH CHAR(5), @HUASP CHAR(5), @SLHUA INT
AS
BEGIN TRAN
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
DECLARE @SLTON INT
SELECT @SLTON = SLTON FROM SANPHAM WITH(UPDLOCK) WHERE HUASP = @HUASP
PRINT @SLTON
IF (@SLTON = 0)
BEGIN
PRINT N'Số lượng hàng trong kho đã hết'
ROLLBACK TRAN
END
IF (@SLHUA > @SLTON)
BEGIN
PRINT N'Số lượng hàng trong kho không đủ đáp ứng với số lượng mua'
ROLLBACK TRAN
END
WAITFOR DELAY '00:00:10'
INSERT [dbo].[CTDOANHANG] ([HUASP], [HACH], [SLSANPHAM]) VALUES (@HUASP, @HACH, @SLHUA)
UPDATE SANPHAM
SET SLTON = @SLTON - @SLHUA
WHERE HUASP = @HUASP
COMMIT TRAN
GO
EXEC KHI_HUASP_FIX 'DH015', 'SP014', 5

```

99 %
Messages
30
90
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
Completion time: 2023-04-14T16:10:24.8783070+07:00
Query executed successfully. KHIEM (16.0 RTM) sa (52) QL_DH_GH_ONLINE 00:00:10 0 rows

Cả hai giao tác đều thực hiện thành công và cho ra số lượng SP chính xác:

SL sản phẩm ban đầu:

SP014	DT474	Cây lau nhà	96	155738.5821
-------	-------	-------------	----	-------------

SL sản phẩm sau khi thực hiện 2 giao tác:

SP014	DT474	Cây lau nhà	85	155738.5821
-------	-------	-------------	----	-------------

Trường hợp 2:			
T1		T2	
BEGIN TRAN			
SET TRAN ISOLATION LEVEL REPEATABLE READ			
SELECT SLTON FROM SANPHAM WITH(UPDLOCK) WHERE MASP = @MASP	UL(SAN PHAM) T1 xin khóa tăng cấp đọc trên bảng SANPHAM ứng với MASP = @MASP		
WAITFOR DELAY '00:00:05'			
		BEGIN TRAN	
		SET TRAN ISOLATION LEVEL REPEATABLE READ	
		SELECT SLTON FROM SANPHAM WITH(UPDLOCK) WHERE MASP = @MASP	UL(SAN PHAM) T1 xin khóa tăng cấp đọc trên bảng SANPHAM ứng với MASP = @MASP
		WAITFOR DELAY '00:00:05'	
UPDATE SANPHAM SET SLTON = @SL WHERE MASP = @MASP	XL(SAN PHAM) T1 chuyển thành khóa ghi trên bảng SANPHAM ứng với MASP = @MASP		
COMMIT TRAN			
		UPDATE SANPHAM SET SLTON = @SL WHERE MASP =	XL(SAN PHAM) T1 chuyển thành khóa ghi trên bảng

		@MASP	SANPHAM ứng với MASP = @MASP
		COMMIT TRAN	

GIẢI THÍCH: Tương tự như tình huống 1, áp dụng mức cô lập **REPEATABLE READ** đối với cả 2 giao tác, việc áp dụng mức cô lập này giúp cho giao tác này không thể chen ngang để thực hiện hành động ghi trên giao tác khác. Tuy nhiên, áp dụng mức cô lập này thì dẫn đến hai giao tác T1 và T2 xảy ra tình trạng **CONVERSION DEADLOCK**, vì thế bên cạnh áp dụng mức cô lập, cần áp dụng thêm khóa **UPDLOCK**, lúc này giao tác 1 sẽ đọc dữ liệu trên bảng SANPHAM bằng cách xin khóa tăng cấp, nếu như giao tác 1 muốn ghi trên bảng SANPHAM thì khóa tăng cấp sẽ chuyển thành khóa ghi để thực hiện hành động ghi. Cả 2 giao tác đã được xử lý thành công mà không còn tình trạng **LOST UPDATE**.

T1:

```

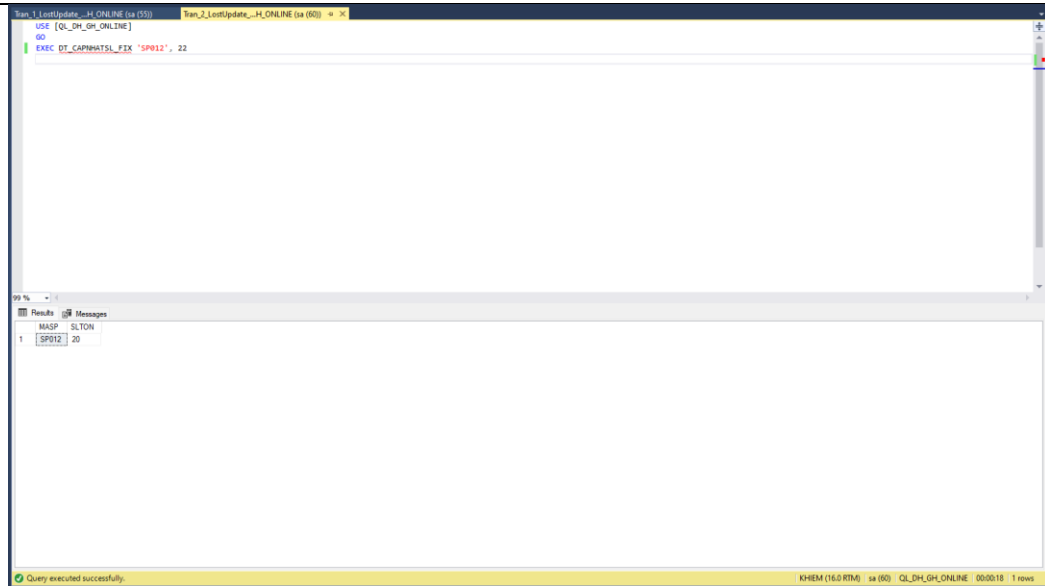
Trans_1_LostUpdate...H_ONLINE (sa (55))
GO
USE [QL_DH_ONLINE]
GO
--CREATE OR ALTER PROCEDURE DT_CAPHATSL_FEX @MASP CHAR(5), @SL INT
AS
BEGIN TRAN
SET TRAN ISOLATION LEVEL REPEATABLE READ
SELECT MASP, SLTON
FROM SANPHAM WITH(UPDLOCK)
WHERE MASP = @MASP
WAITFOR DELAY '00:00:10'
UPDATE SANPHAM
SET SLTON = @SL
WHERE MASP = @MASP
COMMIT TRAN
GO
EXEC DT_CAPHATSL_FEX 'SP012', 20

```

Results	Messages
MASP SLTON	
1 SP012 30	

Query executed successfully. KHIEM (16.0 RTM) sa (55) QL_DH_ONLINE 00:00:10 1 rows

T2:



Giao tác 1 đọc số lượng sản phẩm ‘SP001’ là 30, sau đó cập nhật lại thành 20. Giao tác 2 sẽ đọc giá trị 20 của T1 vừa cập nhật và cập nhật lên thành 22.

B. DIRTY READ

Tình huống 1:

T1		T2	
Input: @MAHD CHAR(5), @NGAYCAPNHAT DATE		Input: @MAHD CHAR(5), @MADT CHAR(5)	
BEGIN TRAN			
DECLARE @KTRANGAY DATE SET @KTRANGAY = (SELECT NGAYKT FROM HOPDONG WHERE MAHD = @MAHD)	SL T1 xin khóa đọc trên bảng HOPDONG ứng với MAHD = @MAHD		
UPDATE HOPDONG SET NGAYKT = @NGAYCAPNHAT WHERE MAHD = @MAHD	XL T1 xin khóa ghi trên bảng HOPDONG		
WAITFOR DELAY			

'00:00:10'			
		BEGIN TRAN	
		SET TRAN ISOLATION LEVEL READ COMMITTED	
		SELECT * FROM HOPDONG WITH(NOLOCK) WHERE MAHD = @MAHD AND MADT = @MADT	SL T2 xin khóa đọc trên bảng HOPDONG
		COMMIT TRAN	
IF (@KTRANGAY > @NGAYCAPNHAT) BEGIN ROLLBACK TRAN RETURN END			
COMMIT TRAN			

- Để tránh tình trạng DIRTY READ, thì đưa T2 về mức cô lập READ COMMITTED. Khi 2 giao tác thực hiện thì đầu tiên T1 thực hiện select và update trên bảng HOPDONG, trong khi đó thì T2 thực hiện đọc hợp đồng của T1 sau khi cập nhật. Sau đó vì nhập ngày kết thúc sai nên T1 ROLLBACK và theo cơ chế mức cô lập READ COMMITTED thì T2 chỉ được đọc dữ liệu sau khi đã commit, vì vậy T2 sẽ trả lại kết quả ban đầu khi chưa thực hiện update.

Đây là kết quả khi T2 được xét mức cô lập

```

USE [QL_DH_GH_ONLINE]
GO
CREATE OR ALTER PROCEDURE XEM_DS_MAHD_FIX @MAHD CHAR(5), @MADT CHAR(5)
AS
BEGIN TRAN
    SET TRAN ISOLATION LEVEL READ COMMITTED
    SELECT * FROM HOPDONG WHERE MAHD = @MAHD AND MADT = @MADT
COMMIT TRAN
GO
EXEC XEM_DS_MAHD_FIX 'HD469', 'DT541'

```

21 %

Results Messages

	MAHD	MADT	MACN	NGAYKY	NGAYKT	PHANTRAMHOAHONG	TINHTRANGHD
1	HD469	DT541	CN654	2020-03-15	2023-04-30	10	Còn hiệu lực

Lúc này thì ngày kết thúc của hợp đồng vẫn giữ nguyên như cũ là 2023-04-30.

DIRTY READ			
Tình huống 2:			
T1		T2	
Input: @MASP CHAR(5), @GIA MONEY		Input: @MASP CHAR(5)	
BEGIN TRAN			
DECLARE @GIASP MONEY SET @GIASP = (SELECT DONGIA FROM SANPHAM WHERE MASP = @MASP)	SL T1 xin khóa đọc trên bảng SANPHAM ứng với MASP = @MASP		
UPDATE SANPHAM SET DONGIA = @GIA WHERE MASP = @MASP	XL T1 xin khóa ghi trên bảng SANPHAM		
WAITFOR DELAY '00:00:10'			
		BEGIN TRAN	

		SET TRAN ISOLATION LEVEL READ COMMITTED	
		SELECT * FROM SANPHAM WHERE MASP = @MASP	SL T2 xin khóa đọc trên bảng SANPHAM
		COMMIT TRAN	
IF (@GIA >= 5 * @GIASP) BEGIN ROLLBACK TRAN RETURN END			
COMMIT TRAN			

- Để tránh tình trạng DIRTY READ, thì đưa T2 về mức cô lập READ COMMITTED. Khi 2 giao tác thực hiện thì đầu tiên T1 thực hiện select và update trên bảng SANPHAM, trong khi đó thì T2 thực hiện đọc giá sản phẩm của T1 sau khi cập nhật. Sau đó vì nhập giá sai nên T1 ROLLBACK và theo cơ chế mức cô lập READ COMMITTED thì T2 chỉ được đọc dữ liệu sau khi đã commit, vì vậy T2 sẽ trả lại kết quả ban đầu khi chưa thực hiện update.

Đây là kết quả khi T2 được xét mức cô lập

```

--DIRTY READ
--T2
USE [QL_DH_GH_ONLINE]
GO

CREATE OR ALTER PROCEDURE KH_XEMGIA_SP_FIX @MASP CHAR(5)
AS
BEGIN TRAN
    SET TRAN ISOLATION LEVEL READ COMMITTED
    SELECT *
    FROM SANPHAM
    WHERE MASP = @MASP
COMMIT TRAN
GO
EXEC KH_XEMGIA_SP_FIX 'SP015'

```

21 %

Results Messages

	MASP	MADT	TENSP	SLTON	DONGIA	MATD
1	SP015	DT125	Chuối	66	2000.00	NULL

Lúc này thì giá sản phẩm vẫn giữ nguyên như cũ là 2000.

c. UNREPEATABLE READ			
Tình huống 1:			
T1		T2	
BEGIN TRAN			
SET TRAN ISOLATION LEVEL REPEATABLE READ			
DECLARE @COUNT INT SET @COUNT = 0 SELECT @COUNT = (SELECT COUNT(*) FROM NHANVIEN WHERE MACN = @MACN)	SL(NHANVIEN) T1 xin khóa đọc trên bảng NHANVIEN		
WAITFOR DELAY '00:00:05'			
		BEGIN TRAN	

		UPDATE NHANVIEN SET MACN = @MACN WHERE MANV = @MANV	XL(NHANVIEN) T2 xin khóa ghi trên bảng NHANVIEN. Tuy nhiên, T2 cần đợi T1 commit để nhả khóa thì mới có thể xin khóa ghi được.
		COMMIT TRAN	
IF ((SELECT COUNT (*) FROM NHANVIEN WHERE MACN = @MACN) <> @COUNT) BEGIN RAISERROR (N'Số lượng dữ liệu hai lần đọc khác nhau',16,1) END SELECT * FROM NHANVIEN WHERE MACN = @MACN	SL(NHANVIEN) T1 xin khóa đọc trên bảng NHANVIEN thành công. T1 sẽ nhả khóa sau khi thực hiện commit.		
COMMIT TRAN			
GIẢI THÍCH: sử dụng mức cô lập Repeatable Read Với mức cô lập này ngăn cho việc giao tác T2 thực hiện chen ngang vào giao tác T1 từ đó làm không xảy ra tình trạng Unrepeatable Read nữa . Bởi vậy T1 sau khi thực hiện xong tra khóa thì T2 mới được thực hiện.			

```

BEGIN TRAN
SET TRAN ISOLATION LEVEL REPEATABLE READ
DECLARE @COUNT INT
SET @COUNT = 0
SET @COUNT = (SELECT COUNT(*) FROM NHANVIEN WHERE MACN = @MACN)
WAITFOR DELAY '00:00:05'

IF ((SELECT COUNT(*) FROM NHANVIEN WHERE MACN = @MACN) <> @COUNT)
BEGIN
    RAISERROR(N'Khác số lượng giữa 2 lần đọc',16,1)
END
SELECT *
FROM NHANVIEN
WHERE MACN = @MACN
COMMIT TRAN
GO
EXEC TIM_NV_CN_FIX 'CN321'

```

```

GO
/*UNREPEATABLE READ*/
/*T2*/
EXEC CAPNHAT_MACN_NV 'NV579', 'CN301'

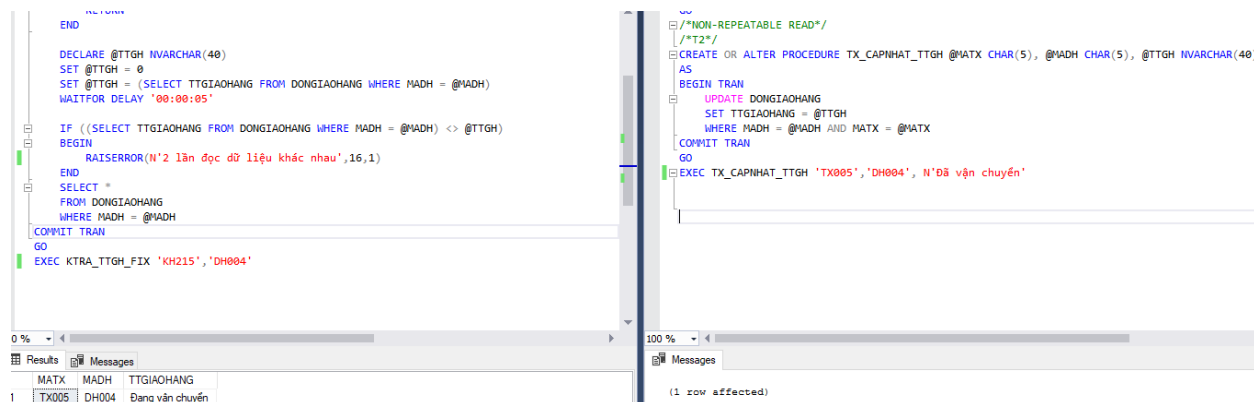
```

MANV	TENNV	SDT	DIACHI	MACN
NV579	Ông Hiếu Kiệt	0552 876 232	19 Đào Duy Từ	CN321

(1 row affected)

Tình huống 2:			
T1		T2	
BEGIN TRAN			
SET TRAN ISOLATION LEVEL REPEATABLE READ			
IF NOT EXISTS (SELECT DGH.* FROM DONGIAOHANG DGH, DONHANG DH WHERE DGH.MADH = DH.MADH AND DH.MAKH = @MAKH AND DGH.MADH = @MADH)	SL(DONHANG,DONGIAOHANG) T1 xin khóa đọc trên bảng DONHANG, DONGIAOHANG. T1 sẽ nhả khóa sau khi commit.		
DECLARE @TTGH NVARCHAR(40) SET @TTGH = 0 SET @TTGH = (SELECT TTGIAOHANG FROM DONGIAOHANG WHERE MADH = @MADH)	SL(DONGIAOHANG) T1 xin khóa đọc trên bảng DONGIAOHANG. T1 sẽ nhả khóa sau khi commit.		
WAITFOR DELAY '00:00:05'			
		BEGIN TRAN	
		UPDATE DONGIAOHANG SET TTGIAOHANG = @TTGH WHERE MADH = @MADH AND MATX = @MATX	XL(DONGIAOHANG) T1 xin khóa ghi trên bảng DONGIAOHANG. Thành công. T2 sẽ chờ T1 commit để xin khóa.
		COMMIT TRAN	
IF ((SELECT TTGIAOHANG FROM DONGIAOHANG WHERE MADH = @MADH) <> @TTGH) BEGIN RAISERROR(N'Dữ liệu hai lần đọc khác nhau',16,1)	SL(DONGIAOHANG) T1 xin khóa đọc trên bảng DONGIAOHANG. T1 sẽ nhả khóa		

END SELECT * FROM DONGIAOHANG WHERE MADH = @MADH	sau khi commit.		
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.

D. PHANTOM			
Tình huống 1			
T1	T2		
BEGIN TRAN			
SET TRAN ISOLATION LEVEL SERIALIZABLE			
DECLARE @COUNT INT SET @COUNT = 0 SET @COUNT = (SELECT COUNT(*) FROM LOGIN) WAITFOR DELAY '00:00:10'	SL(LO GIN) T1 xin khóa đọc bảng LOGIN		
		BEGIN TRAN	
		INSERT INTO LOGIN (USERID, USERNAME,PASSWORD,ROL E_USER) VALUES (@USERID,@USERNAME, @PW,@ROLE)	XL(LO GIN) T2 xin khóa đọc bảng LOGIN, nhưng T2 phải chờ T1

		COMMIT TRAN	
IF ((SELECT COUNT(*) FROM LOGIN) <> @COUNT) BEGIN RAISERROR(N'Dữ liệu hai lần đọc khác nhau',16,1) END	SL(LO GIN) T1 xin khóa đọc bảng LOGIN		
SELECT * FROM LOGIN			
COMMIT TRAN			
Việc áp dụng mức cô lập cao nhất như là SERIALIZABLE giúp ta có thể ngăn chặn việc INSERT, UPDATE của giao tác khác chen vào khi một giao tác nào đó đang thực hiện để tránh tình trạng dữ liệu bị sai KẾT QUẢ ➔ T1			


```
USE [QL_DH_GH_ONLINE]
GO
/*PHANTOM - TH1 */
/* T1 */
CREATE OR ALTER PROCEDURE XEM_DS_LOGIN_FIX
AS
BEGIN TRAN
    SET TRAN ISOLATION LEVEL SERIALIZABLE
    DECLARE @COUNT INT
    SET @COUNT = 0
    SET @COUNT = (SELECT COUNT(*) FROM LOGIN)
    WAITFOR DELAY '00:00:10'

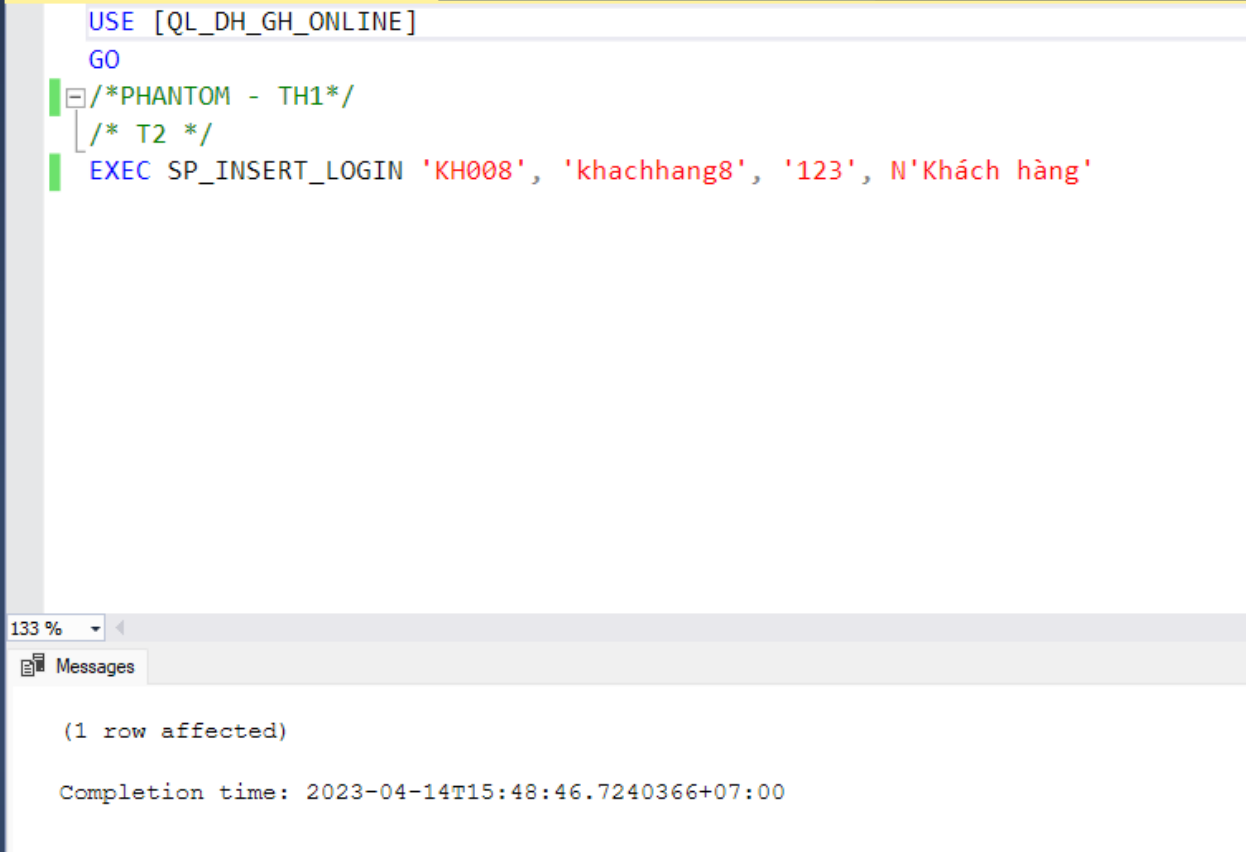
    IF ((SELECT COUNT(*) FROM LOGIN) <> @COUNT)
    BEGIN
        RAISERROR(N'Dữ liệu hai lần đọc khác nhau',16,1)
    END
    SELECT * FROM LOGIN
COMMIT TRAN
GO
EXEC XEM_DS_LOGIN_FIX
```

133 %

Results Messages

	IDLG	USERID	USERNAME	PASSWORD	ROLE_USER
17	LG018	KH008	khachhang8	123	Khách hàng

→ T2



USE [QL_DH_GH_ONLINE]
GO
/*PHANTOM - TH1*/
/* T2 */
EXEC SP_INSERT_LOGIN 'KH008', 'khachhang8', '123', N'Khách hàng'

133 %
Messages
(1 row affected)
Completion time: 2023-04-14T15:48:46.7240366+07:00

➔ T1 sau 2 lần đọc cho dữ liệu giống nhau và trả ra kết quả bảng LOGIN. Nên đã giải quyết được tình trạng PHANTOM

PHANTOM			
Tình huống 2			
T1		T2	
BEGIN TRAN			
SET TRAN ISOLATION LEVEL SERIALIZABLE			
DECLARE @COUNT INT SET @COUNT = 0 SET @COUNT = (SELECT COUNT(*) FROM HOPDONG WHERE MACN = @MACN) WAITFOR DELAY	SL(HOP DONG) T1 xin khóa đọc bảng LOGIN		

'00:00:10'			
		BEGIN TRAN	
		INSERT INTO HOPDONG(MADT,MACN, NGAYKY,NGAYKT, PHANTRAMHOAHONG, TINHTRANGHD) VALUES (@MADT,@MACN, @NGAYKY,@NGAYKT, @PTHOAHONG, N'Dang chờ duyet')	XL(HOP DONG) T2 xin khóa đọc bảng HOPDONG, nhưng T2 phải chờ T1
		IF (@NGAYKY >@NGAYKT) BEGIN PRINT N'NGÀY KÝ VÀ NGÀY KT KHÔNG HỢP LỆ' ROLLBACK TRAN END	
		COMMIT TRAN	
IF ((SELECT COUNT(*) FROM HOPDONG WHERE MACN = @MACN) <> @COUNT) BEGIN RAISERROR(N'Dữ liệu hai lần đọc khác nhau',16,1) END	SL(LOG IN) T1 xin khóa đọc bảng LOGIN		
SELECT * FROM HOPDONG WHERE MACN = @MACN			
COMMIT TRAN			
<p>Tương tự tình huống 1, việc áp dụng mức cô lập cao nhất như là SERIALIZABLE giúp ta có thể ngăn chặn việc INSERT, UPDATE của giao tác khác chen vào khi một giao tác nào đó đang thực hiện để tránh tình trạng dữ liệu bị sai</p> <p>➔ T1</p>			

```
USE [QL_DH_GH_ONLINE]
GO
/*PHANTOM - TH2*/
/* T1 */
CREATE OR ALTER PROCEDURE XEM_DS_HD_FIX @MACN CHAR(5)
AS
BEGIN TRAN
    SET TRAN ISOLATION LEVEL SERIALIZABLE
    DECLARE @COUNT INT
    SET @COUNT = 0
    SET @COUNT = (SELECT COUNT(*) FROM HOPDONG WHERE MACN = @MACN)
    WAITFOR DELAY '00:00:10'

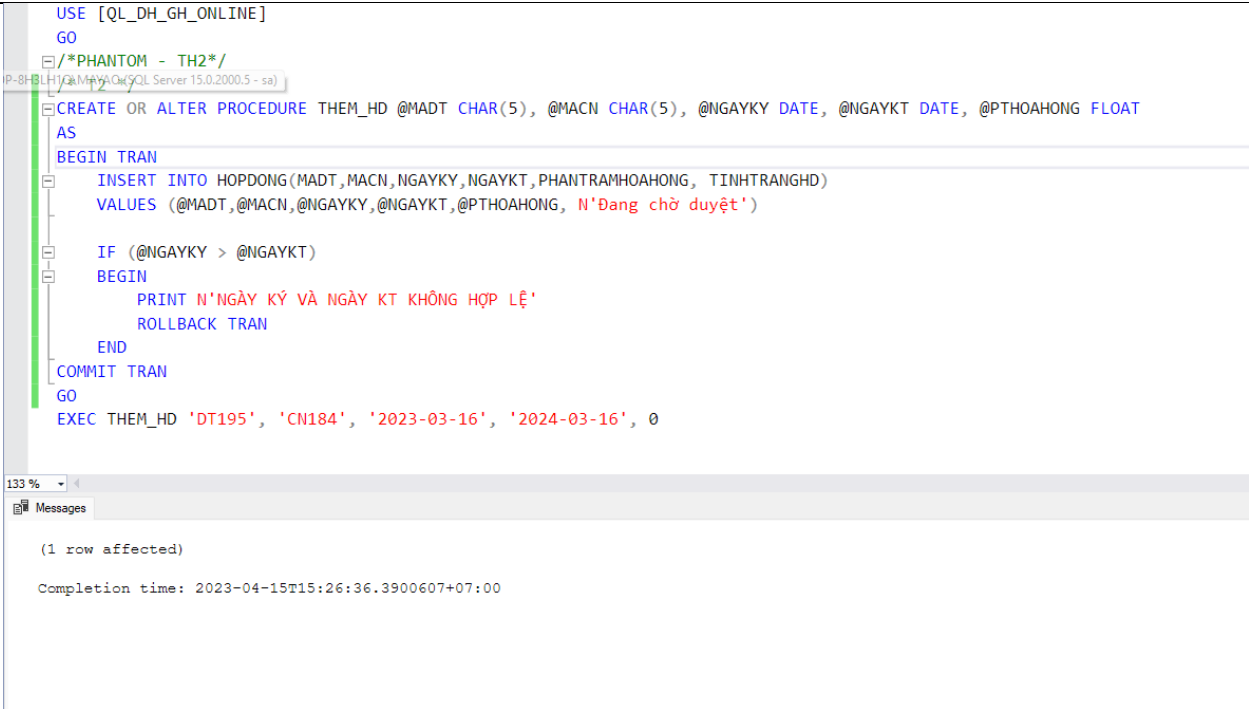
    IF ((SELECT COUNT(*) FROM HOPDONG WHERE MACN = @MACN) <> @COUNT)
    BEGIN
        RAISERROR(N'Dữ liệu hai lần đọc khác nhau',16,1)
    END
    SELECT * FROM HOPDONG WHERE MACN = @MACN
COMMIT TRAN
GO
EXEC XEM_DS_HD_FIX 'CN184'
```

133 %

Results Messages

	MAHD	MADT	MACN	NGAYKY	NGAYKT	PHANTRAMHOAHONG	TINHTRANGHD
1	HD094	DT152	CN184	2020-02-02	2021-09-20	39.3037949871755	Hết hạn
2	HD501	DT152	CN184	2020-10-30	2021-12-08	5.17763933407964	Còn hiệu lực
3	HD774	DT152	CN184	2020-02-16	2022-07-02	7.89198074857331	Còn hiệu lực
4	HD804	DT195	CN184	2023-03-16	2024-03-16	0	Đang chờ duyệt

→ T2



```

USE [QL_DH_GH_ONLINE]
GO
/*PHANTOM - TH2*/
CREATE OR ALTER PROCEDURE THEM_HD @MADT CHAR(5), @MACN CHAR(5), @NGAYKY DATE, @NGAYKT DATE, @PTHOAHONG FLOAT
AS
BEGIN TRAN
    INSERT INTO HOPDONG(MADT,MACN,NGAYKY,NGAYKT,PHANTRAMHOAHONG, TINHTRANGHD)
    VALUES (@MADT,@MACN,@NGAYKY,@NGAYKT,@PTHOAHONG, N'Đang chờ duyệt')
    IF (@NGAYKY > @NGAYKT)
    BEGIN
        PRINT N'NGÀY KÝ VÀ NGÀY KT KHÔNG HỢP LỆ'
        ROLLBACK TRAN
    END
    COMMIT TRAN
GO
EXEC THEM_HD 'DT195', 'CN184', '2023-03-16', '2024-03-16', 0
  
```

Messages

(1 row affected)

Completion time: 2023-04-15T15:26:36.3900607+07:00

➔ T1 sau 2 lần đọc cho dữ liệu giống nhau và trả ra kết quả bảng HOPDONG. Nên đã giải quyết được tình trạng PHANTOM

E. CYCLE DEADLOCK			
T1		T2	
BEGIN TRAN			
UPDATE LOGIN SET PASSWORD = @PASS_NEW WHERE USERID = @USERID	XL(LOGIN) T1 xin khóa ghi trên bảng LOGIN.		
WAITFOR DELAY '00:00:05'			
		BEGIN TRAN	
		UPDATE LOGIN SET PASSWORD = @PASS_NEW WHERE USERID = @USERID	XL(LOGIN) T2 xin khóa ghi trên bảng LOGIN. T2 chờ T1 commit thì bắt đầu thực hiện.
		WAITFOR DELAY '00:00:05'	

UPDATE TAIXE SET CMND = @CMND_NEW WHERE MATX = @MATX COMMIT TRAN	XL(TAIXE) T1 xin khóa ghi trên bảng LOGIN.		
		UPDATE TAIXE SET CMND = @CMND_NEW WHERE MATX = @MATX COMMIT TRAN	XL(TAIXE) T2 xin khóa ghi trên bảng LOGIN.
GIẢI THÍCH:. 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:			

```

/*CYCLE DEADLOCK*/
/* T1 */
CREATE OR ALTER PROCEDURE QL1_DEADLOCK_FIX @USERID CHAR(5), @PASS_NEW CHAR(10), @MATX CHAR(5)
AS
BEGIN TRAN
    UPDATE LOGIN
    SET PASSWORD = @PASS_NEW
    WHERE USERID = @USERID
    WAITFOR DELAY '00:00:05'

    UPDATE TAIXE
    SET CMND = @CMND_NEW
    WHERE MATX = @MATX
COMMIT TRAN
GO
EXEC QL1_DEADLOCK_FIX 'KH001', '1234', 'TX001', '20126044'

(1 row affected)
(1 row affected)
  
```

```

/*CYCLE DEADLOCK*/
/* T2 */
CREATE OR ALTER PROCEDURE QL2_DEADLOCK_FIX @USERID CHAR(5), @PASS_NEW CHAR(10), @MATX CHAR(5)
AS
BEGIN TRAN
    UPDATE LOGIN
    SET PASSWORD = @PASS_NEW
    WHERE USERID = @USERID
    WAITFOR DELAY '00:00:05'

    UPDATE TAIXE
    SET CMND = @CMND_NEW
    WHERE MATX = @MATX
COMMIT TRAN
GO
/*Demo deadlock*/
EXEC QL2_DEADLOCK_FIX 'KH001', '1234', 'TX001', '20126044'

(1 row affected)
(1 row affected)
  
```

F. CONVERSION DEADLOCK

T1	T2
BEGIN TRAN	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ	
SELECT * FROM HOPDONG WITH(UPDLOCK) WHERE MAHD = @MAHD WAITFOR DELAY '00:00:10'	UL(HOPD ONG) T1 xin khóa tăng cấp trên bảng để đọc HOPDONG ứng với MAHD = @MAHD

		BEGIN TRAN	
		SET TRANSACTION ISOLATION LEVEL REPEATABLE READ	
		SELECT * FROM HOPDONG WITH(UPDLOCK) WHERE MAHD = @MAHD WAITFOR DELAY '00:00:10'	UL(HOPD ONG) T1 xin khóa tăng cấp trên bảng để đọc HOPDONG ứng với MAHD = @MAHD. Tuy nhiên vì T1 đang giữ khóa tăng cấp trên bảng HOPDONG nên T2 phải chờ T1 nhả khóa
UPDATE HOPDONG SET TINHTRANGHD = @TTHD WHERE MAHD = @MAHD	XL(HOPD ONG) T1 update khóa UL thành XL để có thể ghi trên bảng HOPDONG		
COMMIT TRAN		UPDATE HOPDONG SET TINHTRANGHD = @TTHD WHERE MAHD = @MAHD	XL(HOPD ONG) T2 update khóa UL thành XL để có thể ghi trên bảng HOPDONG
		COMMIT TRAN	
<p>Để giải quyết tình trạng lỗi CONVERSION DEADLOCK, có thể sử dụng khóa tăng cấp (Update lock) trong quá trình thực hiện các thao tác trên bảng dữ liệu HOPDONG. Khi T1 được cấp khóa tăng cấp, nó chỉ được phép đọc dữ liệu trên bảng HOPDONG và T2 cũng xin khóa tăng cấp. Tuy nhiên, T2 phải chờ T1 nhả khóa trước khi thực hiện các thao tác của mình, vì cả hai đều xin khóa tăng cấp cùng đơn vị dữ liệu. Sau khi T1 cập nhật dữ liệu và commit, T2 sẽ được cấp khóa tăng cấp và thực hiện tương tự như T1. Quá trình này giúp tránh tình trạng deadlock và đảm bảo tính nhất quán của dữ liệu</p> <p>KẾT QUẢ</p> <p>➔ T1</p>			

```
USE [QL_DH_GH_ONLINE]
GO
/*CONVERSION DEADLOCK*/
/* T1 */
CREATE OR ALTER PROCEDURE CPNHAT_TTHDONG_FIX @MAHD CHAR(5), @TTHD NVARCHAR(40)
AS
BEGIN TRAN
    SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
    SELECT *
    FROM HOPDONG WITH(UPDLOCK)
    WHERE MAHD = @MAHD
    WAITFOR DELAY '00:00:10'
    UPDATE HOPDONG
    SET TINHTRANGHD = @TTHD
    WHERE MAHD = @MAHD
COMMIT TRAN
GO
EXEC CPNHAT_TTHDONG_FIX 'HD146', N'Hết hạn'
```

133 %

Results Messages

	MAHD	MADT	MACN	NGAYKY	NGAYKT	PHANTRAMHOAHONG	TINHTRANGHD
1	HD146	DT339	CN405	2020-02-21	2023-12-13	45.4695487373832	Hết hạn

→ T2


```
USE [QL_DH_GH_ONLINE]
GO
/*CONVERSION DEADLOCK*/
/* T2 */
GO
EXEC CPNHAT_TTHDONG_FIX 'HD146', N'Còn hiệu lực'
```

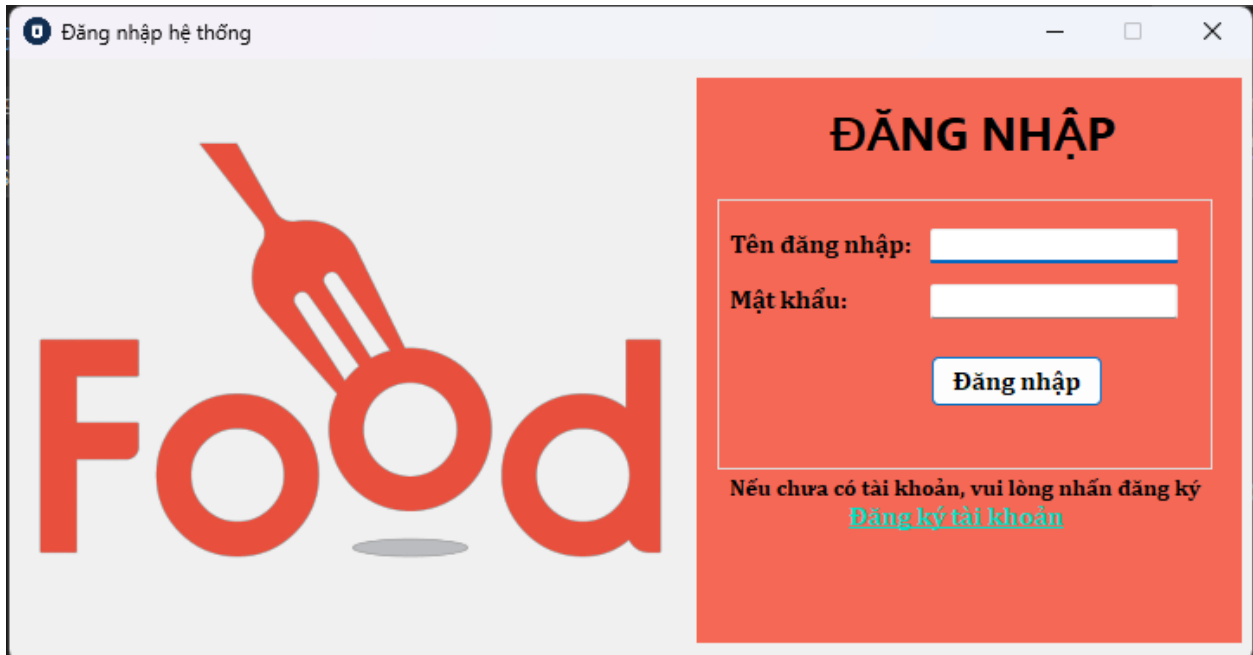
133 %

Results Messages

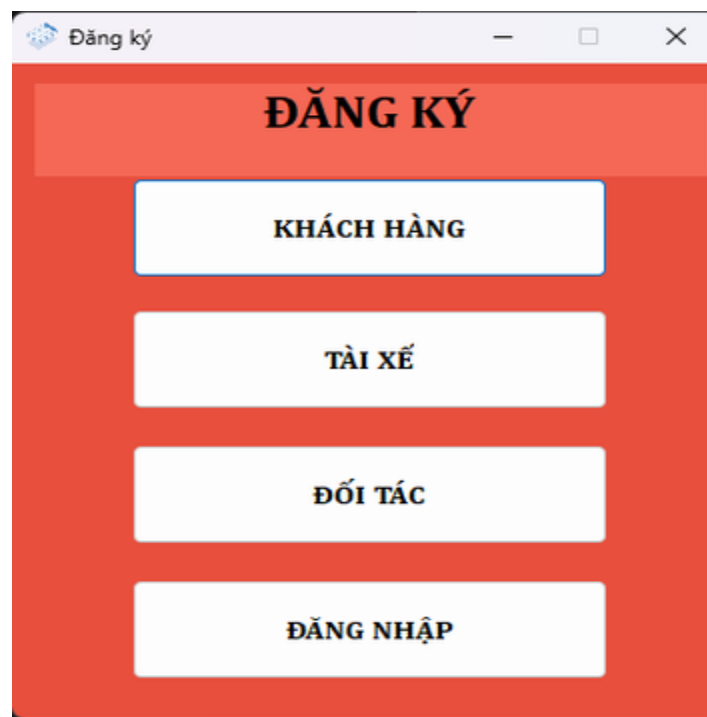
	MAHD	MADT	MACN	NGAYKY	NGAYKT	PHANTRAMHOAHONG	TINHTRANGHD
1	HD146	DT339	CN405	2020-02-21	2023-12-13	45.4695487373832	Hết hạn

CÀI ĐẶT HỆ THỐNG GIAO DIỆN ỨNG DỤNG

- Giao diện đăng nhập



- Đăng ký tài khoản



- Đăng ký tài khoản khách hàng

The screenshot shows a web browser window titled "Đăng ký tài khoản". The main heading is "KHÁCH HÀNG". Below it, a section titled "Nhập thông tin cá nhân" (Enter personal information) contains a form with four input fields: "Họ tên:" (Last name), "Địa chỉ:" (Address), "Số điện thoại:" (Phone number), and "Email:". A "Tiếp tục" (Continue) button is located at the bottom right of the form.

- Đăng ký tài khoản tài xế

The screenshot shows a web browser window titled "Đăng ký tài khoản". The main heading is "TÀI XẾ". Below it, a section titled "Nhập thông tin cá nhân" (Enter personal information) contains a form with eight input fields: "Họ tên:" (Last name), "Số điện thoại:" (Phone number), "CMND:" (ID card), "TT đóng phí:" (Payment type), "Địa chỉ:" (Address), "Email:" (Email), "Biển số xe:" (Vehicle license plate), and "Tài khoản NH:" (Bank account). There are also two dropdown menus for "Khu vực hoạt động:" (Operating area) and "Tài khoản NH:" (Bank account). A "Tiếp tục" (Continue) button is located at the bottom right of the form.

- Đăng ký tài khoản đối tác

Đăng ký tài khoản

ĐỐI TÁC

Nhập thông tin cá nhân

Họ tên: Địa chỉ:

Quận: Thành phố:

Số điện thoại: Email:

Loại hàng ĐK: Mã số thuế:

Người đại diện:

- Giao diện Admin hệ thống nhân viên

Admin

QUẢN LÝ HỆ THỐNG

Mode: ☒ Fixed ☐ No fix

Hệ thống nhân viên | Hệ thống khách hàng | Hệ thống đối tác | Hệ thống tài xế | Thông tin tài khoản | Demo cycle deadlock 1 | Demo cycle deadlock 2

DANH SÁCH NV

Mã nhân viên:

Họ tên:

Số điện thoại:

Địa chỉ:

Mã chi nhánh:

Số lượng nhân viên: 699

Mã NV	Tên NV	Số điện thoại	Địa chỉ	Mã chi nhánh
NV001	Hồ Hữu Vinh	0224 809 029	05 Nguyễn Tri ...	CN450
NV002	Dương Phúc Ph...	0426 816 675	35 Hoàng Văn T...	CN410
NV003	Ngô Huy Ninh	0797 956 446	16 Cao Bá Quát	CN301
NV005	Bùi Hữu Hiếu	0274 446 186	36 Đặng Tất	CN336
NV006	Đàm Ân Công	0288 442 673	49 Nguyễn Mìn...	CN560
NV007	Lê Hoàng Đạt	0966 511 253	84 Mac Mahon ...	CN481
NV008	Đỗ Hoa Tuấn	0513 775 805	43 Abattoire de	CN097
NV009	Đặng Ninh Hiếu	0788 782 873	78 Thích Quảng...	CN795
NV010	Phan Vinh Tú	0154 785 319	61 Lê Văn Duyệt	CN282
NV011	Võ Hiền Vy	0118 940 292	64 Hòa Mỹ	CN793
NV012	Phó Kỳ Phục	0171 580 872	55 Thái Lập Th...	CN140
NV013	Hồ Ninh Lụa	0557 334 051	96 Huỳnh Thúc...	CN252
NV014	Lý Trinh Sơn	0653 967 122	31 Douaumont	CN639
NV015	Bùi Văn Hải	0461 831 108	01 Thuận Kiều	CN455
NV016	Trương Quý Tu...	0439 952 009	25 Lý Chính Th...	CN358

- Giao diện Admin hệ thống khách hàng

MAKH	HOTEN	SĐT	EMAIL	DIACHI
KH001	Mạch Hiếu Kiệt	0707 235 985	ypbi@gmail.com	... 582 Hoàng Sa
KH002	Triệu Văn Bình	0468 032 842	benh@gmail.com	... 643 Lý Chính Thắng
KH003	Tô Gấm Huy	0663 654 842	jswh@gmail.com	... 36 Cầu Kho
KH004	Bùi Phong Nga	0243 778 111	frhe@gmail.com	... 93 Ngõ Tắt Tở
KH005	Tô Bửu Ánh	0429 542 261	hlpj@gmail.com	... 03 Hồ Xuân Hương
KH006	Triệu Xuyên Quý	0615 899 728	knbt@gmail.com	... 05 Abattoire de
KH007	Lê Phúc Đức	0771 352 047	sxfy@gmail.com	... 923 Đặng Trần Côn
KH008	Trương Thắng Thanh	0409 797 879	kcru@gmail.com	... 926 Chu Mạnh Trinh
KH009	Phan Danh Hoa	0684 644 128	ntat@gmail.com	... 464 Chu Mạnh Trinh
KH010	Trần Xuân Nghĩa	0237 890 439	jviv@gmail.com	... 502 Lê Ngõ Cát
KH011	Bùi Phúc Thọ	0333 911 479	tilb@gmail.com	... 854 Nguyễn Công Trứ
KH012	Phan Linh Tín	0765 385 361	kdmk@gmail.com	... 743 Đào Duy Từ
KH013	Phan Hùng Chấn	0735 713 671	vkny@gmail.com	... 441 Hồ Xuân Hương
KH014	Tôn Phát Phương	0642 561 608	ushx@gmail.com	... 34 Thích Quảng Đức
KH015	Đỗ Trung Tài	0656 475 965	woys@gmail.com	... 86 Huyền Quang

Số lượng khách hàng: 720

- Giao diện Admin hệ thống đối tác

MADT	TENDOITAI	THANHPHỐ	QUAN	DIACHI	SĐT	EMAIL	MST	MALH	MANDD	SLDH
DT001	Huỳnh Hi...	Phủ Thọ	Việt Trì	81 Nguyễ...	0648 859 ...	svb.p@g...	38385113...	LH004	DD057	1
DT004	Nghiêm N...	Long An	Tân An	537 Cao T...	0513 745 ...	ykl2@ma...	47952572...	LH006	DD098	5
DT005	Tôn Hoàn...	Hòa Bình	Hòa Bình	363 Nguy...	0533 973 ...	khn.n@g...	37024688...	LH001	DD013	17
DT006	Lê Ninh T...	Hòa Bình	Hòa Bình	252 Đổng ...	0123 144 ...	cmh.v@g...	67633839...	LH004	DD054	2
DT007	Nguyễn Ki...	Trà Vinh	Trà Vinh	390 Đổng ...	0528 270 ...	mxy.r@gm...	08417243...	LH004	DD053	12
DT008	Triệu Vy ...	Hà Giang	Hà Giang	70 Phan T...	0226 740 ...	hkr2@gm...	93770140...	LH006	DD096	10
DT009	Võ Vi Hải	Sơn La	Sơn La	442 Cao T...	0669 508 ...	idd.e@gm...	74050186...	LH005	DD077	16
DT010	Trần Hải ...	Yên Bái	Yên Bái	48 Hoàng ...	0964 067 ...	ztg.l@gma...	73359400...	LH003	DD039	18
DT011	Tô Trang ...	Hòa Bình	Hòa Bình	29 Phan T...	0392 147 ...	ipm0@gm...	52036088...	LH006	DD093	3
DT012	Ngô Võ Ân	Hậu Giang	Vị Thanh	92 Lưu Vã...	0215 498 ...	bbv.x@g...	58145156...	LH006	DD095	7
DT013	Nguyễn T...	Vĩnh Long	Vĩnh Long	871 Đê T...	0877 839 ...	cgc.e@g...	38926041...	LH006	DD083	16
DT014	Ngô Nghĩ...	Kon Tum	Kon Tum	49 Phan ...	0145 543 ...	ghl2@gm...	18316130...	LH005	DD071	8
DT015	Phó Duy ...	Bến Tre	Bến Tre	32 Đặng ...	0847 201 ...	ql.p@gma...	73431354...	LH003	DD036	0
DT016	Bùi Võ Đức	Bạc Liêu	Bạc Liêu	583 Đổng ...	0722 463 ...	qsn.c@g...	59160067...	LH001	DD009	17
DT017	Đặng Quả...	Trà Vinh	Trà Vinh	67 Phan T...	0955 683 ...	tjn1@gma...	85042321...	LH003	DD047	20

Số lượng đối tác: 599

- Giao diện Admin hệ thống tài xế

Admin

QUẢN LÝ HỆ THỐNG

Mode: ☒ Fixed ☐ No fix

Hệ thống nhân viên Hệ thống khách hàng Hệ thống đối tác **Hệ thống tài xế** Thông tin tài khoản Demo cycle deadlock 1 Demo cycle deadlock 2

MATX	MAKV	HOTEN	CMND	SĐT	DIACHI	BIENSOXE	TKGANHA	EMAIL	TTDONGPH	THUNHAP
TX001	KV050	Huỳnh Ng...	24091578...	02887795...	25 Mac M...	52B-84015	2354168266	sx250@g...	Chưa đóng	18000.0000
TX002	KV058	Hoàng Lo...	24899877...	05695592...	736 Huyn...	81E-73208	4826850123	wi855@g...	Chưa đóng	0.0000
TX003	KV019	Phan Hòa...	24175865...	03793572...	73 Cô Bắc	16C-57728	0034332358	fq276@g...	Đã đóng	20639.0000
TX004	KV071	Lê Đại Lo...	24986285...	03391300...	61 Nguyễ...	48E-51028	0580101339	ik037@g...	Chưa đóng	0.0000
TX005	KV055	Dương Lợi...	24819068...	08389610...	11 Lê Côn...	28D-29901	9577727418	kw183@g...	Đã đóng	899.2646
TX006	KV043	Hồ Minh P...	24967930...	05676716...	564 Nguy...	11E-75120	3984811483	qk567@g...	Đã đóng	553.1168
TX007	KV020	Ông Sơn ...	24364563...	07598264...	92 Cách ...	50D-99174	3396469269	yv234@g...	Đã đóng	391.3222
TX008	KV097	Đặng Văn...	24228315...	08944346...	41 Amiral	56AB-46075	8218863490	hv067@g...	Đã đóng	784.5578
TX009	KV066	Phó Nghị ...	24927891...	05405199...	57 Tự Do	89B-01652	3353215432	wz840@g...	Đã đóng	871.3022
TX010	KV024	Hoàng Vin...	24083385...	09252359...	09 Mac M...	19E-37817	2532841164	kh783@g...	Đã đóng	166.8647
TX011	KV003	Ông Khải ...	24303326...	09367464...	35 Cao B...	58F-06085	3902533570	pf073@g...	Chưa đóng	0.0000
TX012	KV011	Tôn Thiên ...	24203742...	04942405...	312 Nguy...	94F-85410	5164744017	dg417@g...	Chưa đóng	0.0000
TX013	KV082	Trần Lan ...	24830922...	02462261...	860 Jean	82F-61511	2593589093	ue047@g...	Đã đóng	423.7022
TX014	KV061	Bùi Vi Lâm	24402470...	09004856...	96 Hồ Tú...	33F-64351	1634548080	df469@g...	Chưa đóng	0.0000
TX015	KV006	Phó Đạt T...	24407220...	07036341...	53 Verdun	37B-07414	8388374922	fd538@g...	Chưa đóng	0.0000

Số lượng tài xế: 700

- Giao diện Admin xem thông tin tài khoản

Admin

QUẢN LÝ HỆ THỐNG

Mode: ☒ Fixed ☐ No fix

Hệ thống nhân viên Hệ thống khách hàng Hệ thống đối tác Hệ thống tài xế **Thông tin tài khoản** Demo cycle deadlock 1 Demo cycle deadlock 2

Cập nhật tài khoản

ID LG: User ID:

User Name:

Password:

Role:

IDLG	USERID	USERNAME	PASSWORD	ROLE_USER
LG001	DT001	doitac1	doitac1	Đối tác
LG002	NV002	nhanvien2	nhanvien2	Nhân viên
LG003	KH001	khachhang1	123	Khách hàng
LG004	TX003	taixe3	taixe3	Tài xế
LG005	QT001	admin	admin	Quản trị
LG006	TX001	tx1	123	Tài xế
LG007	KH249	kh1	123	Khách hàng
LG008	NV457	dr2	dr2	Nhân viên
LG009	NV220	nv220	123	Nhân viên
LG010	DTS41	dt541	123	Đối tác
LG011	KH249	kh249	123	Khách hàng
LG012	QT002	admin2	admin2	Quản trị
LG013	NV283	nv283	123	Nhân viên
LG014	QT001	admin1	admin1	Quản trị
LG015	TX298	tx298	123	Tài xế
LG016	NV363	nv363	123	Nhân viên

- Giao diện demo tranh chấp

- Giao diện đối tác xem danh sách hợp đồng

MAHD	MADT	MACN	NGAYKY	NGAYKT	PHANTRAMH	TINHTRANGH
HD469	DT541	CN654	15/03/2020	30/04/2023	10	Còn hiệu lực

- Giao diện đối tác tạo hợp đồng mới

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE ĐĂNG XUẤT

Mã đối tác: DT541 Tên đối tác: Huỳnh Thu Lan

Mode: ☒ Fixed ☐ No fix Chọn

DS hợp đồng Tạo hợp đồng Danh sách sản phẩm Quản lý đơn đặt hàng Cập nhật thông tin Doanh thu

Mã hợp đồng: Mã chi nhánh: Ngày ký: Ngày kết thúc:

THÊM HỢP ĐỒNG

DANH SÁCH HỢP ĐỒNG ĐANG CHỜ PHÊ DUYỆT

Mã HĐ	Mã CN	Ngày ký	Ngày kết thúc	Phần trăm HH	Tình trạng HĐ
HD803	CN185	17/04/2023	19/04/2023	10	Đang chờ duyệt

- Giao diện đối tác xem danh sách sản phẩm

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE ĐĂNG XUẤT

Mã đối tác: DT541 Tên đối tác: Huỳnh Thu Lan

Mode: ☒ Fixed ☐ No fix Chọn

DS hợp đồng Tạo hợp đồng Danh sách sản phẩm Quản lý đơn đặt hàng Cập nhật thông tin Doanh thu

Mã sản phẩm:

Tên sản phẩm:

Số lượng:

Đơn giá:

CẬP NHẬT GIÁ

XÓA CN SL

THÊM RESET

Mã SP	Tên sản phẩm	Số lượng tồn	Đơn giá (đồng)
SP007	Phở mai	21	50960
SP068	Giấy in	65	76239

- Giao diện đối tác quản lí các đơn hàng

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE

Mã đối tác: DT541 Tên đối tác: Huỳnh Thu Lan

Mode: ☒ Fixed ☐ No fix Chọn

DS hợp đồng Tạo hợp đồng Danh sách sản phẩm **Quản lý đơn đặt hàng** Cập nhật thông tin Doanh thu

DANH SÁCH LOẠI HÀNG

DANH SÁCH ĐƠN ĐẶT HÀNG

Cập nhật TTĐH:
Mã ĐH: DH000

Cập nhật

Mã ĐH	Mã khách hàng	Tình trạng ĐH	Phí vận chuyển	Tổng tiền
DH772	KH001	Đã nhận	0	254798
DH773	KH001	Đã nhận	0	254798
DH775	KH001	Đã nhận	1000	305758
DH776	KH001	Đã nhận	0	458637
DH777	KH001	Đã nhận	0	50960
DH778	KH001	Đã nhận	0	101919
DH779	KH001	Đã nhận	0	1019193
DH780	KH001	Đã nhận	10000	101919
DH774	KH001	Đã nhận	0	914865

- Giao diện đối tác xem thông tin đối tác

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE

Mã đối tác: DT541 Tên đối tác: Huỳnh Thu Lan

Mode: ☒ Fixed ☐ No fix Chọn

DS hợp đồng Tạo hợp đồng Danh sách sản phẩm Quản lý đơn đặt hàng **Cập nhật thông tin** Doanh thu

Mã đối tác: DT541

Tên đối tác: Huỳnh Thu Lan

SĐT: 0715 940 999

Quận: Quảng Ngãi

Thành phố (Tỉnh): Quảng Ngãi

Địa chỉ: 826 Nguyễn Cảnh Chân

Email: hmm8@gmail.com

MST: 3860923838

NGƯỜI ĐẠI DIỆN

Mã NDD: DD021

Tên NDD: Hồ Lựa Thọ

TÀI KHOẢN

Tài khoản đối tác: dt541

Mật khẩu: 123

Cập nhật Đổi mật khẩu

- Giao diện đối tác xem doanh thu và phí hoa hồng theo tháng

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE **ĐĂNG XUẤT**

Mã đối tác: DT541 Tên đối tác: Huỳnh Thu Lan

Mode: ☒ Fixed ☐ No fix

DS hợp đồng Tạo hợp đồng Danh sách sản phẩm Quản lý đơn đặt hàng Cập nhật thông tin **Doanh thu**

Năm	Tháng	Doanh thu	Phí hoa hồng
2023	4	2446064.0496	244606.40496

Doanh thu

- Giao diện tài xế xem danh sách đơn hàng

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE **ĐĂNG XUẤT**

Mã tài xế: TX298 Tên tài xế: Hoàng Duyên Khuyển

Mode: ☒ Fixed ☐ No fix

Danh sách đơn hàng đã nhận Cập nhật tình trạng giao hàng Thu nhập Cập nhật thông tin

Cập nhật phí VC: 24137 Mã DH: DH292

Mã DH	Mã KH	Tên KH	Tình trạng DH	Địa chỉ GH	Phí VC	Tổng tiền
DH292	KH630	Dương Hồ...	Đang xử lý	02 Võ Tánh	24137	6344122
DH773	KH001	Mạch Hiế...	Đã nhận	111	0	254798
DH774	KH001	Mạch Hiế...	Đã nhận	123	0	914865
DH775	KH001	Mạch Hiế...	Đã nhận	123	1000	625662
DH777	KH001	Mạch Hiế...	Đã nhận	123	0	387411
DH778	KH001	Mạch Hiế...	Đã nhận	123	0	1701439
DH779	KH001	Mạch Hiế...	Đã nhận	123	0	1019193
DH780	KH001	Mạch Hiế...	Đã nhận	123	10000	101919

- Giao diện tài xế cập nhật tình trạng đơn hàng

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE ĐĂNG XUẤT

Mã tài xế: TX298 Tên tài xế: Hoàng Duyên Khuyến

Mode: ☒ Fixed ☐ No fix Chọn

Danh sách đơn hàng đã nhận Cập nhật tình trạng giao hàng Thu nhập Cập nhật thông tin

Cập nhật TTGH: Đã vận chuyển Mã ĐH: DH292 Cập nhật

Mã Tài xế	Mã đơn hàng	Tình trạng giao hàng
TX298	DH292	Đã vận chuyển
TX298	DH773	Đang vận chuyển
TX298	DH774	Đang vận chuyển
TX298	DH775	Đã vận chuyển
TX298	DH777	Đang vận chuyển
TX298	DH778	Đang vận chuyển
TX298	DH779	Đang vận chuyển
TX298	DH780	Đã vận chuyển

- Giao diện thu nhập của tài xế

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE ĐĂNG XUẤT

Mã tài xế: TX298 Tên tài xế: Hoàng Duyên Khuyến

Mode: ☒ Fixed ☐ No fix Chọn

Danh sách đơn hàng đã nhận Cập nhật tình trạng giao hàng **Thu nhập** Cập nhật thông tin

Số lượng đơn hàng đã nhận: 8

Số lượng đơn hàng đã giao: 3

Tổng thu nhập: 31624 đồng

Thu nhập được tính bằng 90% phí vận chuyển/1 đơn hàng đã vận chuyển.

- Giao diện thông tin của tài xế

Tài xế

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE **ĐĂNG XUẤT**

Mã tài xế: TX298 Tên tài xế: Hoàng Duyên Khuyến

Mode: ☒ Fixed ☐ No fix **Chọn**

Danh sách đơn hàng đã nhận Cập nhật tình trạng giao hàng Thu nhập Cập nhật thông tin

Mã tài xế: TX298 Email: cx012@gmail.com

Tên tài xế: Hoàng Duyên Khuyến TTĐP: Đã đóng

CMND: 2488429052

SĐT: 0368050011

Địa chỉ: 225 Mac Mahon nối dài

Biển số xe: 29C-73797

Tài khoản NH: 4423883997

TÀI KHOẢN:

Tài khoản tài xế: tx298

Mật khẩu: 123

Cập nhật **Đổi mật khẩu**

- Giao diện nhân viên xem danh sách hợp đồng của chi nhánh

Nhân viên

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE **ĐĂNG XUẤT**

Mã nhân viên: NV363 Tên nhân viên: Phạm Sừ Sừ

Mode: ☒ Fixed ☐ No fix **Chọn**

Danh sách hợp đồng của chi nhánh Cập nhật thông tin

Mã HD: HD469 Cập nhật TTĐ: Còn hiệu lực **Cập nhật**

Cập nhật ngày KT: 17/04/2023 Phần trăm HH: 10 **Cập nhật**

Danh sách hợp đồng của chi nhánh

MAHD	MADT	MACN	NGAYKY	NGAYKT	PHANTRAMH	TINHTRANGH
HD469	DT541	CN654	15/03/2020	30/04/2023	10	Còn hiệu lực
HD579	DT541	CN654	23/01/2020	28/12/2022	10	Còn hiệu lực

Chỉ nhánh: CN654

- Giao diện thông tin của nhân viên

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE **ĐĂNG XUẤT**

Mã nhân viên: NV363 Tên nhân viên: Phạm Sầu Sầu

Mode: ☒ Fixed ☐ No fix Chọn

Danh sách hợp đồng của chi nhánh Cập nhật thông tin

Mã nhân viên: NV363 Tài khoản nhân viên: nv363

Tên nhân viên: Phạm Sầu Sầu Mật khẩu: 123

SĐT: 0249 605 212 **Đổi mật khẩu**

Địa chỉ: 57 Lê Công Kiều

Mã chi nhánh: CN654 **Cập nhật**

Chi nhánh: CN654

- Giao diện đặt hàng của khách hàng

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE **ĐĂNG XUẤT**

Mã khách hàng: KH001 Tên khách hàng: Mạch Hiếu Kiệt

Mode: ☒ Fixed ☐ No fix Chọn

Đặt hàng Tìm sản phẩm Đơn hàng của bạn Cập nhật thông tin

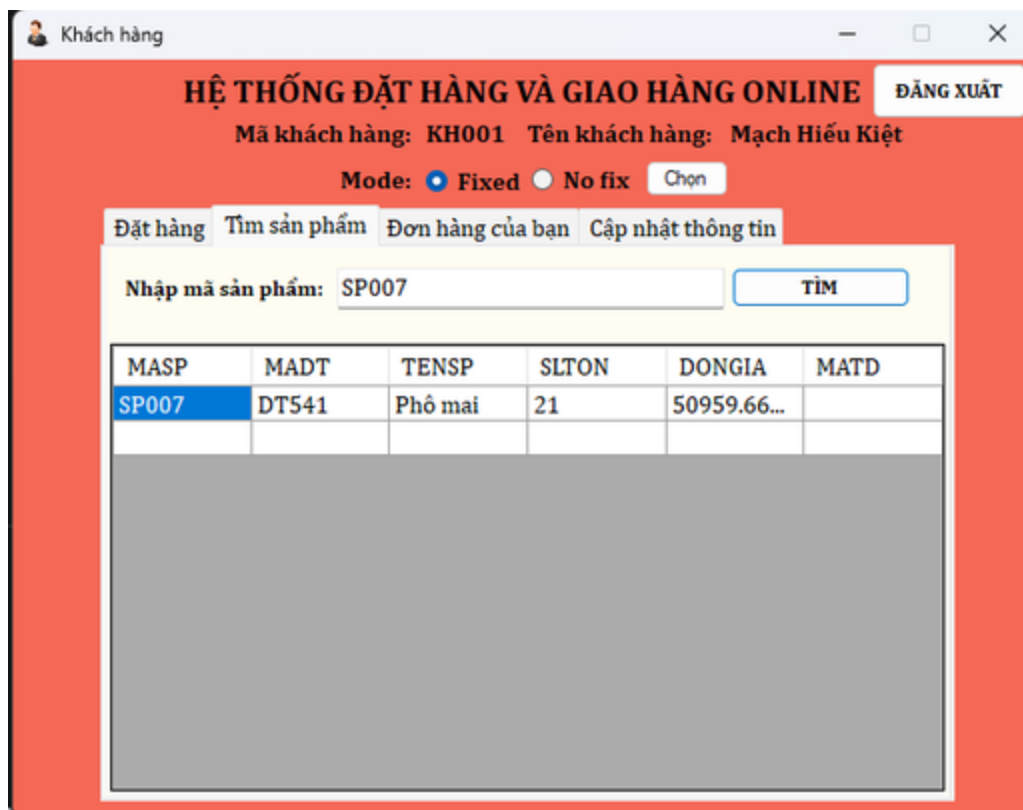
Địa chỉ giao hàng: 123 Tỉnh (TP): Kon Tum **XÁC NHẬN**

Mã DH: DH781 Mã SP: Số lượng: 0 **THÊM**

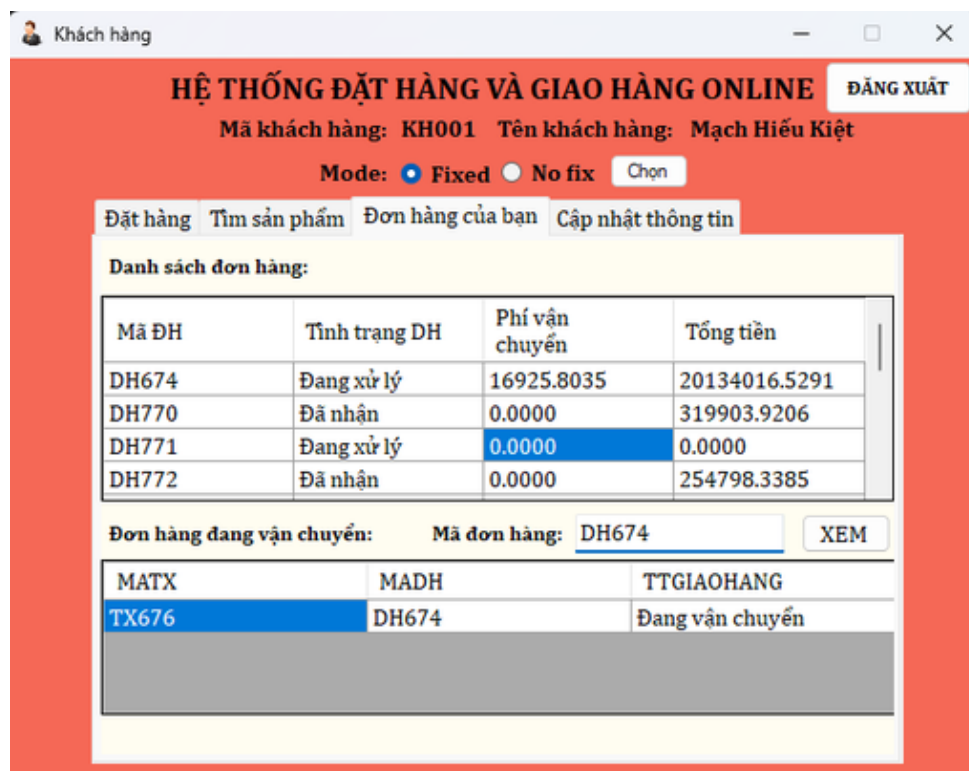
XEM SẢN PHẨM **XÁC NHẬN GH** **XÓA SẢN PHẨM**

Mã sản phẩm	Mã đối tác	Tên SP	Số lượng tồn	Đơn giá	Mã thực đơn
SP001	DT147	Nước tẩy ...	77	159951.9...	
SP002	DT619	Nước ép	44	220003.4...	
SP003	DT430	Cháo	66	112150.5...	
SP004	DT617	Lipton	91	394533.7...	
SP005	DT349	Thảm	29	457206.6...	
SP006	DT456	Sữa rửa ...	81	20.0000	
SP007	DT541	Phô mai	21	50959.66...	

- Giao diện tìm sản phẩm của khách hàng



- Giao diện xem đơn hàng của khách hàng



- Giao diện thông tin khách hàng

The screenshot shows a web browser window titled "Khách hàng". The main heading is "HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE" with a "ĐĂNG XUẤT" button. Below the heading, it displays "Mã khách hàng: KH001" and "Tên khách hàng: Mạch Hiếu Kiệt". There is a "Mode:" section with radio buttons for "Fixed" (selected) and "No fix", and a "Chọn" button. A navigation bar contains four tabs: "Đặt hàng", "Tìm sản phẩm", "Đơn hàng của bạn", and "Cập nhật thông tin" (which is active). The active tab shows a form with the following fields: "Mã khách hàng:" (KH001), "Tên khách hàng:" (Mạch Hiếu Kiệt), "SốT:" (0707 235 985), "Địa chỉ:" (582 Hoàng Sa), "Email:" (ypbi@gmail.com), "Tài khoản:" (khachhang1), and "Mật khẩu:" (123). There is a "Đổi mật khẩu" button next to the password field and a "Cập nhật" button at the bottom of the form.

Khách hàng

HỆ THỐNG ĐẶT HÀNG VÀ GIAO HÀNG ONLINE **ĐĂNG XUẤT**

Mã khách hàng: KH001 Tên khách hàng: Mạch Hiếu Kiệt

Mode: ☒ Fixed ☐ No fix

Đặt hàng Tìm sản phẩm Đơn hàng của bạn **Cập nhật thông tin**

Mã khách hàng: KH001 Tài khoản: khachhang1

Tên khách hàng: Mạch Hiếu Kiệt Mật khẩu: 123

SốT: 0707 235 985

Địa chỉ: 582 Hoàng Sa

Email: ypbi@gmail.com

Tham khảo

- Tham khảo từ file Tutorials của cô Hồ Thị Hoàng Vy
- Tham khảo các Slide bài giảng của môn Cơ sở dữ liệu.
- Tham khảo tài liệu của Microsoft

[Tutorials for SQL Server - SQL Server | Microsoft Learn](#)