

BỘ MÔN HỆ THỐNG THÔNG TIN – KHOA CÔNG NGHỆ THÔNG TIN  
ĐẠI HỌC KHOA HỌC TỰ NHIÊN THÀNH PHỐ HỒ CHÍ MINH

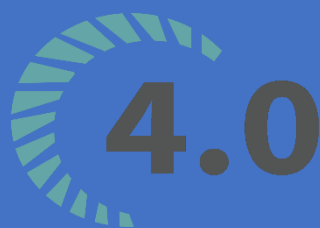
BÁO CÁO ĐỒ ÁN THỰC HÀNH (LẦN 3)

MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

GVHD

TS. PHẠM THỊ BẠCH HUỆ

THS. TIẾT GIA HỒNG



Nhóm sinh viên thực hiện: Nhóm 5

1. 19127041 – Lê Thị Phương Linh
2. 19127173 – Hoàng Lê Khanh
3. 19127361 – Nguyễn Thị Ngọc Diệu

## THÔNG TIN NHÓM

Số thứ tự nhóm: **Nhóm 5**

STT	MSSV	Họ tên	Công việc	% Hoàn thành
1	19127041	Lê Thị Phương Linh	Script lỗi Phantom Read, Unrepeatable Read, Cycle Deadlock	100 %
2	19127173	Hoàng Lê Khanh	Script lỗi Dirty Read, Unrepeatable Read	100 %
3	19127361	Nguyễn Thị Ngọc Diệu	Script lỗi Lost Update, Conversion Deadlock	100 %

## CÀI ĐẶT TÌNH HUỐNG TRANH CHẤP

## I. Sinh viên thực hiện: Lê Thị Phương Linh - 19127041

## 1. Tình huống 1:

<b>ERR01: Phantom Read</b>			
T1 (User = đối tác): thực hiện xem doanh thu của mình (đối với đơn hàng đã giao).			
T2 (User = tài xế): Thực hiện cập nhật trạng thái “Đã giao hàng” cho đơn hàng của đối tác → doanh thu của đối tác tăng			
USP_DT_XEMDOANHTHU	Khóa	USP_TX_DAGIAOHANG	Khóa
<u>Input:</u>  <u>Output:</u> Tổng doanh thu, tổng số lượng đơn hàng kèm danh sách đơn hàng đã giao.		<u>Input:</u> Trạng thái đơn hàng, mã đơn hàng, thời gian thực hiện giao hàng  <u>Output:</u> Thêm thành công trạng thái mới của đơn hàng.	
SET TRANSACTION ISOLATION  LEVEL REPEATABLE READ			
BEGIN TRAN			
B1: Tính tổng doanh thu và tổng đơn hàng với trạng thái đã giao	S(UV_DHDOITAC)  //Xin khoá đọc trên view UV_DHDOITAC		

<pre>SELECT SUM(PHI_SAN_PHAM - PHI_VAN_CHUYEN) AS DOANH_THU, COUNT(*) AS TONG_LUONG_DON  FROM UV_DHDOITAC  WHERE TRANG_THAI = N'Đã giao hàng'</pre> <p>* NOTE: UV_DHDOITAC là một view hiển thị danh sách đơn hàng của chính đối tác đó.</p>	với điều kiện trạng thái đơn hàng là “Đã giao hàng”		
<pre>WAITFOR DELAY '00:00:05'</pre>			
		<pre>BEGIN TRAN</pre>	
		<pre>B1: Kiểm tra đơn hàng có tồn tại / có phải là đơn hàng mà tài xế đã nhận hay không  IF @MaDH NOT IN (SELECT MADH FROM UV_DHTAIXE)  BEGIN  PRINT @MaDH + N' không phải của tài xế!!'  ROLLBACK TRAN  RETURN 1</pre>	<p><b>S(UV_DHTAIXE)</b></p> <p>//Xin khoá đọc trên view UV_DHTAIXE</p>

		<p>END</p> <p>* NOTE: UV_DHTAIXE là một view hiển thị danh sách đơn hàng mà tài xế đã nhận.</p>	
		<p>B2: Kiểm tra đơn hàng có phải đã được giao rồi hay không</p> <pre> IF EXISTS (SELECT *             FROM UV_TTDHTAIXE             WHERE TEN_TRANG_THAI = @TrangThai AND MADH = @MaDH) BEGIN     PRINT @MaDH + N' đã được giao trước đó!!' ROLLBACK TRAN RETURN 1 END </pre> <p>* NOTE: UV_TTDHTAIXE là một view hiển thị danh sách trạng thái đơn hàng mà tài xế đã nhận.</p>	<p><b>S(UV_TTDHTAIXE)</b></p> <p>//Xin khoá đọc trên view UV_TTDHTAIXE</p>

		<b>B3: Thêm trạng thái mới của đơn hàng</b> INSERT UV_TTDHTAIXE VALUES(@TrangThai, @MaDH, @ThoiGian)	<b>X(UV_TTDHTAIXE)</b> //Xin khoá ghi trên view UV_TTDHTAIXE
		<b>B4: Cập nhật thông tin đơn hàng</b> UPDATE UV_DHTAIXE SET TRANG_THAI = @TrangThai WHERE MADH = @MaDH	<b>X(UV_DHTAIXE)</b> //Xin khoá ghi trên view UV_DHTAIXE
		<b>B5: Bắt lỗi hệ thống nếu có</b>	
		<b>COMMIT</b>	
<b>B2: Hiện thị danh sách đơn hàng</b> SELECT MADH, MAKH, PHI_SAN_PHAM, PHI_GIAM, PHI_SAN_PHAM - PHI_GIAM AS THANH_TIEN, HINH_THUC_THANH_TOAN FROM UV_DHDOITAC WHERE TRANG_THAI = 'Đã giao hàng'	<b>S(UV_DHDOITAC)</b> //Xin khoá đọc trên view UV_DHDOITAC với điều kiện trạng thái đơn hàng là “Đã giao hàng”		
<b>COMMIT</b>			

## 2. Tình huống 2:

<b>ERR02: Cycle Deadlock</b>  T1 (User = đối tác): cập nhật thông tin của đối tác.  T2 (User = đối tác): cập nhật thông tin của đối tác.			
USP_DT_TTCASE1	Khóa	USP_DT_TTCASE2	Khóa
<u><b>Input:</b></u> Tên người đại diện, mã chi nhánh, địa chỉ của chi nhánh (đường)  <u><b>Output:</b></u> Deadlock		<u><b>Input:</b></u> Tên người đại diện, mã chi nhánh, địa chỉ của chi nhánh (đường)  <u><b>Output:</b></u> Deadlock	
<b>BEGIN TRAN</b>			
<b>B1: Kiểm tra chi nhánh</b>  IF @MaCN NOT IN (SELECT MACN FROM UV_CN_DOITAC)  BEGIN PRINT @MaCN + ' không phải chi nhánh của đối tác!!!'  ROLLBACK TRAN  RETURN 1  END	<b>S(UV_DHDOITAC)</b>  //Xin khóa đọc trên view UV_CN_DOITAC		

* NOTE: UV_CN_DOITAC là một view hiển thị <b>danh sách chi nhánh</b> của chính đối tác đó.			
<b>B2: Cập nhật tên người đại diện của đối tác</b> <b>UPDATE</b> UV_TTDOITAC <b>SET</b> NGUOI_DAI_DIEN = @NguoiDaiDien	<b>X(UV_DHDOITAC)</b>  //Xin khóa ghi trên view UV_TTDOITAC		
<b>WAITFOR DELAY '00:00:05'</b>			
		<b>BEGIN TRAN</b>	
		<b>B1: Kiểm tra chi nhánh</b>  <b>IF</b> @MaCN NOT IN ( <b>SELECT</b> MACN <b>FROM</b> UV_CN_DOITAC)  <b>BEGIN</b>  <b>PRINT</b> @MaCN + 'N' không phải chi nhánh của đối tác!!'  <b>ROLLBACK TRAN</b>  <b>RETURN</b> 1  <b>END</b>	<b>S(UV_DHDOITAC)</b>  //Xin khóa đọc trên view UV_CNDOITAC



		* NOTE: UV_CN_DOITAC là một view hiển thị <b>danh sách chi nhánh</b> của chính đối tác đó.	
		<b>B2: Cập nhật địa chỉ của chi nhánh</b> UPDATE UV_CN_DOITAC SET DUONG = @Duong WHERE MACN = @MaCN	<b>X(UV_CN_DOITAC)</b>  //Xin khóa ghi trên view UV_CN_DOITAC
		<b>WAITFOR DELAY '00:00:05'</b>	
<b>B3: Cập nhật địa chỉ của chi nhánh</b> UPDATE UV_CN_DOITAC SET DUONG = @Duong WHERE MACN = @MaCN	<b>X(UV_CN_DOITAC)</b>  //Xin khóa ghi trên view UV_CN_DOITAC		
<b>B4: Bắt lỗi hệ thống nếu có</b>			
		<b>B2: Cập nhật tên người đại diện của đối tác</b> UPDATE UV_TTDOITAC	<b>X(UV_DHDOITAC)</b>  //Xin khóa ghi trên view UV_TTDOITAC

		SET NGUOI_DAI_DIEN = @NguoiDaiDien	
		B3: Bắt lỗi hệ thống nếu có	
COMMIT		COMMIT	

### 3. Tình huống 3:

#### ERR03: Phantom Read

T1 (User = khách hàng): khách hàng tìm kiếm sản phẩm theo tên.

T2 (User = đối tác): thêm một sản phẩm mới cùng tên với sản phẩm khách hàng đang tìm.

USP_KH_TIMSP	Khóa	USP_DT_THEMSP	Khóa
<u>Input</u> : Tên sản phẩm <u>Output</u> : Số lượng tìm được và danh sách sản phẩm tương ứng		<u>Input</u> : Mã chi nhánh, mã sản phẩm, số lượng tồn, giá bán <u>Output</u> : Thêm sản phẩm thành công	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ			
BEGIN TRAN			
B1: Tính số lượng sản phẩm đã tìm được	S(UV_XEMSP_KH)		

<pre>SELECT COUNT(*) AS SL_TIM_KIEM FROM UV_XEMSP_KH WHERE TENSP = @TenSP</pre> <p>* NOTE: UV_XEMSP_KH là một view hiển thị <b>danh sách sản phẩm</b> đang được bán.</p>	<p>//Xin khoá đọc trên view UV_XEMSP_KH với điều kiện: theo tên sản phẩm mà KH tìm</p>		
<b>WAITFOR DELAY '00:00:05'</b>			
		<b>BEGIN TRAN</b>	
		<p><b>B1: Kiểm tra sản phẩm có trong danh sách sản phẩm mà hệ thống cho phép bán</b></p> <pre>IF @MaSP NOT IN (SELECT MASP FROM SAN_PHAM)  BEGIN  PRINT @MaSP + N' không tồn tại!!'  ROLLBACK TRAN  RETURN 1  END</pre>	<p><b>S(SAN_PHAM)</b></p> <p>//Xin khoá đọc trên bảng SAN_PHAM</p>

		* NOTE: SAN_PHAM là bảng <b>danh sách sản phẩm</b> mà hệ thống cho phép.	
		<b>B2: Kiểm tra chi nhánh</b>  IF @MaCN NOT IN (SELECT MACN FROM UV_CN_DOITAC WHERE MAHD IS NOT NULL)  BEGIN PRINT @MaCN + N' không thuộc quản lý của đối tác / chưa được ký hợp đồng!!' ROLLBACK TRAN RETURN 1 END  * NOTE: UV_CN_DOITAC là một view hiển thị <b>danh sách chi nhánh</b> của chính đối tác đó.	<b>S(UV_CN_DOITAC)</b>  //Xin khoá đọc trên view UV_CN_DOITAC
		<b>B3: Thêm sản phẩm</b>  INSERT PHAN_PHOI  VALUES (@MaSP, @MaCN, @SL_Ton, @Gia)	<b>X(PHAN_PHOI)</b>  //Xin khoá ghi trên bảng PHAN_PHOI

		B4: Bắt lỗi hệ thống nếu có	
		COMMIT	
<p>B2: Hiển thị sản phẩm</p> <pre>SELECT * FROM UV_XEMSP_KH WHERE TENSP = @TenSP</pre>	<p><b>S(UV_XEMSP_KH)</b></p> <p>//Xin khoá đọc trên view UV_XEMSP_KH với điều kiện: theo tên sản phẩm mà KH tìm</p>		
B3: Bắt lỗi hệ thống nếu có			
COMMIT			

#### 4. Tình huống 4:

##### ERR04: Unrepeatable read

T1 (User = khách hàng): khách hàng kiểm tra thành tiền của sản phẩm (trước khi đặt hàng).

T2 (User = đối tác): thực hiện cập nhật giá cho sản phẩm

USP_KH_CTDH	Khóa	USP_DT_THEMSP	Khóa
<p><b>Input:</b> Mã sản phẩm, mã chi nhánh phân phối, số lượng mua</p> <p><b>Output:</b> Đơn giá và thành tiền tương ứng</p>		<p><b>Input:</b> Mã chi nhánh, mã sản phẩm, số lượng tồn, giá bán</p> <p><b>Output:</b> Thêm sản phẩm thành công</p>	

SET TRANSACTION ISOLATION LEVEL <b>READ COMMITED</b>		SET TRANSACTION ISOLATION LEVEL <b>READ COMMITED</b>	
<b>BEGIN TRAN</b>			
<b>B1: Kiểm tra sản phẩm</b>  IF @MaSP NOT IN (SELECT MASP FROM SAN_PHAM)  BEGIN PRINT @MaSP + <b>N' không tồn tại!!'</b> ROLLBACK TRAN RETURN 1 END	<b>S(SAN_PHAM)</b>  //Xin khoá đọc trên bảng SAN_PHAM		
<b>B2: Kiểm tra chi nhánh</b>  IF @MaCN NOT IN (SELECT MACN FROM PHAN_PHOI)  BEGIN PRINT @MaCN + <b>N' không tồn tại / chưa ký hợp  đồng!!'</b>	<b>S(PHAN_PHOI)</b>  //Xin khoá đọc trên bảng PHAN_PHOI		

ROLLBACK TRAN  RETURN 1  END  * NOTE: PHAN_PHOI là bảng danh sách sản phẩm ứng với từng chi nhánh đã được ký hợp đồng mua bán.			
<b>B3: Thông tin sản phẩm kèm số lượng và đơn giá</b>  SELECT MASP, TENSPP, @SoLuong AS SO_LUONG_MUA, GIA  FROM UV_XEMSP_KH  WHERE MASP = @MaSP AND MACN = @MaCN  * NOTE: UV_XEMSP_KH là một view hiển thị <b>danh sách sản phẩm</b> đang được bán.	<b>S(UV_XEMSP_KH)</b>  //Xin khoá đọc trên view UV_XEMSP_KH với điều kiện theo sản phẩm và chi nhánh		
WAITFOR DELAY '00:00:05'			
		BEGIN TRAN	
		B1: Kiểm tra sản phẩm có trong danh sách sản phẩm mà hệ thống cho phép bán	S(SAN_PHAM)

		<pre> IF @MaSP NOT IN (SELECT MASP                   FROM SAN_PHAM)  BEGIN  PRINT @MaSP + N' không tồn tại!!'  ROLLBACK TRAN  RETURN 1  END  * NOTE: SAN_PHAM là bảng <b>danh sách</b> <b>sản phẩm</b> mà hệ thống cho phép. </pre>	//Xin khoá đọc trên bảng SAN_PHAM
		<p><b>B2: Kiểm tra chi nhánh</b></p> <pre> IF @MaCN NOT IN (SELECT MACN                   FROM UV_CN_DOITAC                   WHERE MAHD IS NOT NULL)  BEGIN  PRINT @MaCN + N' không thuộc quản lý của đối tác / chưa được ký hợp đồng!!'  ROLLBACK TRAN </pre>	<p><b>S(UV_CN_DOITAC)</b></p> <p>//Xin khoá đọc trên view UV_CN_DOITAC</p>



		RETURN 1  END  * NOTE: UV_CN_DOITAC là một view hiển thị <b>danh sách chi nhánh</b> của chính đối tác đó.	
		<b>B3: Cập nhật giá của sản phẩm</b>  UPDATE UV_SPDOITAC SET GIA = @GiaMoi WHERE MASP = @MaSP AND MACN = @MaCN	<b>X(UV_SPDOITAC)</b>  //Xin khoá ghi trên bảng UV_SPDOITAC
		COMMIT	
<b>B2: Hiển thị thành tiền của sản phẩm</b>  SELECT GIA * @SoLuong AS THANH_TIEN FROM UV_XEMSP_KH WHERE MASP = @MaSP AND MACN = @MaCN	<b>S(UV_XEMSP_KH)</b>  //Xin khoá đọc trên view UV_XEMSP_KH với điều kiện theo sản phẩm và chi nhánh		
COMMIT			

## II. Sinh viên thực hiện: Hoàng Lê Khanh - 19127173

## 1. Tình huống 1:

<b>ERROR 01: Dirty read</b>  T1 (User = tài xế): Thêm đơn hàng vào phiếu giao hàng  T2 (User = tài xế): Xem đơn hàng đã nhận giao			
<b>USP_THEMPHIEUGIAOHANGTX</b>	<b>Khóa</b>	<b>USP_KIEMTRADONHANGTX</b>	<b>Khóa</b>
<u><b>Input:</b></u>  @MADH <b>CHAR</b> (8),  @NGAYGIAO <b>DATE</b>		<u><b>Input:</b></u>  @MADHCANTIM <b>CHAR</b> (8)	
<b>SET TRAN ISOLATION LEVEL READ COMMITTED (DEFAULT)</b>		<b>SET TRAN ISOLATION LEVEL READ UNCOMMITTED</b>	
<b>BEGIN TRAN</b>			

<p>B1: Thêm thông tin phiếu giao hàng của tài xế về đơn hàng đã nhận.</p> <pre>INSERT PHIEU_GIAO_HANG VALUES ((SELECT TT.MATX FROM UV_TTTAIXE TT), @MADH, @NGAYGIAO)</pre>	<p>S(PHIEU_GIAO_HANG)</p> <p>// Ghi không cần cấp khóa</p>		
<pre>WAITFOR DELAY '0:0:05'</pre>			
		<pre>BEGIN TRAN</pre>	

		<p>B1: Kiểm tra thông tin mã đơn hàng có trống không.</p> <p>IF @MADHCANTIM IS NULL</p> <p>BEGIN</p> <p>PRINT N'MÃ ĐƠN HÀNG TÌM KIẾM KHÔNG ĐƯỢC ĐỂ TRỐNG'</p> <p>ROLLBACK TRAN</p> <p>RETURN</p> <p>END</p>	
--	--	---	--

		<p>B2: Kiểm tra thông tin đơn hàng có thuộc đơn hàng có thể nhận của tài xế không.</p> <pre> IF NOT EXISTS(SELECT * FROM UV_DHTAIXE WHERE MADH = @MADHCANTIM)  BEGIN  PRINT @MADHCANTIM + N'DƠN HÀNG NÀY KHÔNG THUỘC KHU VỰC CỦA TÀI XẾ'  ROLLBACK TRAN  RETURN  END </pre>	<p>S(UV_DHTAIXE)</p> <p>//Đọc không cần cấp khóa</p>
--	--	---	--

		<p>B3: Tìm thông tin đơn hàng trong danh sách đơn hàng đã nhận.</p> <pre> SELECT * FROM UV_DHTAIXEDANHAN WHERE MADH = @MADHCANTIM </pre>	<p>S(UV_DHTAIXEDANHAN)</p> <p>//Đọc không cần cấp khóa</p>
		<p>B4: Bắt lỗi hệ thống (nếu có)</p> <pre> BEGIN CATCH PRINT 'LỖI HỆ THỐNG ' + ERROR_MESSAGE() ROLLBACK TRAN END CATCH </pre>	

<p>B2: Kiểm tra thông tin đơn hàng được thêm vào thuộc vào đơn tài xế có thể nhận không.</p> <pre>IF NOT EXISTS(SELECT * FROM UV_DHTAIXE WHERE MADH = @MADH) BEGIN PRINT @MADH + N' ĐƠN HÀNG NÀY KHÔNG THUỘC KHU VỰC CỦA TÀI XẾ' ROLLBACK TRAN RETURN 0 END</pre>	<p>S(UV_DHTAIXE)</p> <p>//Cấp khóa đọc</p>		
---	--	--	--

<p>B3: Kiểm tra thông tin ngày giao hàng</p> <pre>IF @NGAYGIAO IS NULL OR @NGAYGIAO &lt; (SELECT MAX(THOI_GIAN) FROM TRANG_THAI_DON_HANG WHERE @MADH = MADH)  BEGIN  PRINT N'KIỂM TRA LẠI NGÀY GIAO HÀNG (KHÔNG ĐƯỢC TRỎNG/ NGÀY GIAO KHÔNG HỢP LỆ)'  ROLLBACK TRAN  RETURN 0  END</pre>	<p>S(TRANG_THAI_DON_HANG)</p> <p>//Cấp khóa đọc</p>		
--	---	--	--



<p>B4: Bắt lỗi hệ thống (nếu có)</p> <p>BEGIN CATCH</p> <p>PRINT N'LỖI HỆ THỐNG ' + ERROR_MESSAGE()</p> <p>ROLLBACK TRAN</p> <p>END CATCH</p>			
COMMIT		COMMIT	

## 2. Tình huống 2:

<p><b>ERROR 02: Unrepeatable read</b></p> <p>T1 (User = khách hàng): <b>Tìm thông tin sản phẩm bằng tên sản phẩm.</b></p> <p>T2 (User = đối tác): <b>Chỉnh sửa tên sản phẩm.</b></p>			
USP_TIMSP TEN	Khóa	USP_DOITEN SP	Khóa

<u><i>Input:</i></u> @TENS <b>NVARCHAR</b> (100)		<u><i>Input:</i></u> @TENCANCAPNHAT <b>NVARCHAR</b> (100), @TENCAPNHAT <b>NVARCHAR</b> (100)	
<b>SET TRAN ISOLATION LEVEL READ COMMITTED (DEFAULT)</b>		<b>SET TRAN ISOLATION LEVEL READ COMMITTED (DEFAULT)</b>	
<b>BEGIN TRAN</b>			

<p>B1: Kiểm tra xem sản phẩm có tồn tại trong danh sách sản phẩm ứng với tên cần tìm.</p> <pre> IF NOT EXISTS(SELECT * FROM SAN_PHAM WHERE TENSPP = @TENSPP) BEGIN PRINT @TENSPP + N' KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN END </pre>	<p>S(SAN_PHAM)</p> <p>//Cấp khóa đọc</p>		
<pre> WAITFOR DELAY '0:0:05' </pre>			
		<pre> BEGIN TRAN </pre>	

		<p>B1: Kiểm tra xem sản phẩm có thuộc đối tác không.</p> <pre>IF NOT EXISTS (SELECT * FROM UV_SPDOITAC WHERE TENSEP = @TENCANCAPNHAT) BEGIN PRINT @TENCANCAPNHAT + N'SẢN PHẨM NÀY KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0 END</pre>	<p>S(UV_SPDOITAC)</p> <p>//Cấp khóa đọc</p>
--	--	---	---

		<p>B2: Kiểm tra tên cập nhật có trống hay không.</p> <pre>IF @TENCAPNHAT IS NULL BEGIN PRINT 'TÊN CẬP NHẬT KHÔNG ĐƯỢC TRỐNG' ROLLBACK TRAN RETURN 0 END</pre>	
--	--	---	--

		<p>B3: Kiểm tra tên cập nhật không được trùng tên trước đó.</p> <pre> IF @TENCAPNHAT = @TENCANCAPNHAT BEGIN PRINT N'TÊN CẬP NHẬT KHÔNG ĐƯỢC TRÙNG VỚI TÊN CŨ' ROLLBACK TRAN RETURN 0 END </pre>	
		<p>B4: Bắt lỗi hệ thống (nếu có)</p> <pre> UPDATE UV_SPDOITAC SET TENS = @TENCAPNHAT WHERE @TENCANCAPNHAT = TENS </pre>	//Không cấp khóa để cập nhật

<p>B2: Tìm sản phẩm ứng với tên sản phẩm nhận vào.</p> <pre>SELECT * FROM SAN_PHAM WHERE TENSEP = @TENSEP</pre>	<p>S(SAN_PHAM)</p> <p>//Cấp khóa đọc</p>	<p>B5: Bắt lỗi hệ thống (nếu có)</p> <pre>BEGIN CATCH  PRINT N'LỖI HỆ THỐNG ' + ERROR_MESSAGE()  ROLLBACK TRAN  END CATCH</pre>	
<p>B3: Bắt lỗi hệ thống (nếu có)</p> <pre>BEGIN CATCH  PRINT N'LỖI HỆ THỐNG ' + ERROR_MESSAGE()  ROLLBACK TRAN  END CATCH</pre>			
COMMIT		COMMIT	

### III. Sinh viên thực hiện: Nguyễn Thị Ngọc Diệu - 19127361

#### 1. Tình huống 1:

<b>ERROR 01: Lost Update</b>  T1 (User = Khách hàng): Thực hiện đặt hàng (tạo mới một chi tiết cho đơn hàng) với sản phẩm A tại chi nhánh X  T2 (User = Đối tác): Cập nhật thêm số lượng tồn cho sản phẩm A tại chi nhánh X			
USP_THEMCTDH	Khóa	USP_THEMSLSP	Khóa
<u><b>Input:</b></u>  @MADH CHAR(8),  @MASP CHAR(8),  @MACN CHAR(8),  @SO_LUONG INT		<u><b>Input:</b></u>  @MASP CHAR(8),  @MACN CHAR(8),  @SO_LUONG INT	
READ COMMITTED (DEFAULT)		READ COMMITTED (DEFAULT)	



BEGIN TRAN			
<p>B1: Kiểm tra sản phẩm có tồn tại trong danh sách hay không</p> <pre>IF NOT EXISTS (SELECT * FROM UV_XEMSP_KH WHERE MASP = @MASP AND MACN = @MACN ) BEGIN PRINT N 'SẢN PHẨM KHÔNG CÓ Ở CHI NHÁNH NÀY' ROLLBACK TRAN RETURN 1 END</pre>	S( UV_XEMSP_KH)		

<p>B2: Kiểm tra số lượng có hợp lệ không</p> <pre> IF @SO_LUONG &lt;= 0 BEGIN PRINT N 'SỐ LƯỢNG PHẢI LỚN HƠN 0' ROLLBACK TRAN RETURN 1 END </pre>			
<p>B3: Kiểm tra số lượng tồn có đáp ứng đủ cho số lượng khách đặt không</p> <pre> DECLARE @SLT INT SET @SLT = (SELECT SO_LUONG_TON FROM PHAN_PHOI WHERE MASP = @MASP AND MACN = @MACN) IF @SLT &lt; @SO_LUONG </pre>	S(PHAN_PHOI)		

BEGIN PRINT N'SỐ LƯỢNG TỒN KHÔNG ĐỦ' ROLLBACK TRAN RETURN 1 END			
WAITFOR DELAY '0:0:05'			
		BEGIN TRAN	

		<p>B1: Kiểm tra sản phẩm có tồn tại trong danh sách hay không</p> <pre>IF NOT EXISTS (SELECT * FROM UV_SPDOITAC WHERE MASP = @MASP AND MACN = @MACN )  BEGIN  PRINT N 'SẢN PHẨM KHÔNG CÓ Ở CHI NHÁNH NÀY'  ROLLBACK TRAN  RETURN 1  END</pre>	S(UV_SPDOITAC)
--	--	---	----------------

		<p>B2: Kiểm tra số lượng có hợp lệ không</p> <p>IF @SO_LUONG &lt;= 0</p> <p>BEGIN</p> <p>PRINT N 'SỐ LƯỢNG PHẢI LỚN HƠN 0'</p> <p>ROLLBACK TRAN</p> <p>RETURN 1</p> <p>END</p>	
--	--	--	--

		<p>B3: Đọc số lượng tồn hiện tại từ bảng PHAN_PHOI</p> <pre> DECLARE @SLT INT  SET @SLT = (SELECT SO_LUONG_TON  FROM PHAN_PHOI  WHERE MASP = @MASP AND MACN = @MACN) </pre>	S(PHAN_PHOI)
		<pre> WAITFOR DELAY '0:0:05' </pre>	
<p>B4: Chèn chi tiết đơn hàng vào dbo.CHI_TIET_DON_HANG</p> <pre> INSERT dbo.CHI_TIET_DON_HANG (MADH, MASP, SO_LUONG) VALUES (@MADH, @MASP, @SO_LUONG) </pre>	X(CHI_TIET_DON_HANG)		

<p>B5: Cập nhật số lượng tồn cho bảng PHAN_PHOI</p> <p>UPDATE PHAN_PHOI</p> <p>SET SO_LUONG_TON = @SLT - @SO_LUONG</p> <p>WHERE MASP = @MASP AND MACN = @MACN</p>	X(PHAN_PHOI)		
<p>B4: Bắt lỗi hệ thống (nếu có)</p> <p>BEGIN CATCH</p> <p>PRINT N'LỖI HỆ THỐNG ' + ERROR_MESSAGE()</p> <p>ROLLBACK TRAN</p> <p>RETURN 1</p> <p>END CATCH</p>			
COMMIT			

		<p>B4: Cập nhật số lượng tồn cho bảng PHAN_PHOI</p> <p>UPDATE PHAN_PHOI</p> <p>SET SO_LUONG_TON = @SLT + @SO_LUONG</p> <p>WHERE MASP = @MASP AND MACN = @MACN</p>	X(PHAN_PHOI)
		<p>B4: Bắt lỗi hệ thống (nếu có)</p> <p>BEGIN CATCH</p> <p>PRINT N'LỖI HỆ THỐNG ' + ERROR_MESSAGE()</p> <p>ROLLBACK TRAN</p> <p>RETURN 1</p> <p>END CATCH</p>	
		COMMIT	



## 2. Tình huống 2:

**ERROR02: Conversion Deadlock**

T1 (User = Đối tác): Cập nhật thông tin đối tác (sửa tên người đại diện)

T2 (User = Đối tác): Cập nhật thông tin đối tác (sửa địa chỉ - tên đường)

USP_DT_NDD	Khóa	USP_DT_Duong	Khóa
<b><u>Input:</u></b> @NGUOI_DAI_DIEN <b>nvarchar</b> (30)		<b><u>Input:</u></b> @MASP CHAR(8), @MACN CHAR(8), @SO_LUONG INT	
SET TRANSACTION ISOLATION LEVEL <b>REPEATABLE READ</b>		SET TRANSACTION ISOLATION LEVEL <b>REPEATABLE READ</b>	
<b>BEGIN TRAN</b>			

<p>B1: Kiểm tra người đại diện muốn đổi có trùng với người đại diện hiện tại không</p> <pre> DECLARE @NGUOI_DAI_DIEN_HT nvarchar(30)  SET @NGUOI_DAI_DIEN_HT = (SELECT NGUOI_DAI_DIEN  FROM UV_TTDOITAC)  IF (@NGUOI_DAI_DIEN = @NGUOI_DAI_DIEN_HT)  BEGIN  PRINT N'Người đại diện muốn đổi trùng với người đại diện hiện tại'  ROLLBACK TRAN  RETURN  END </pre>	<p><b>S(UV_TTDOITAC)</b></p> <p>// Xin khóa đọc</p>		
<p><b>WAITFOR DELAY '0:0:05'</b></p>			

		<b>BEGIN TRAN</b>	
		<p>B1: Kiểm tra tên đường muốn đổi có trùng với tên đường hiện tại không</p> <p><b>DECLARE</b> @DUONG_HT <b>nvarchar</b>(30)</p> <p><b>SET</b> @DUONG_HT = (<b>SELECT</b> DUONG</p> <p><b>FROM</b> UV_TTDOITAC)</p> <p><b>IF</b> (@DUONG = @DUONG_HT)</p> <p><b>BEGIN</b></p> <p><b>PRINT</b> N'Tên đường muốn đổi trùng với tên đường hiện tại'</p> <p><b>ROLLBACK TRAN</b></p> <p><b>RETURN</b></p> <p><b>END</b></p>	<p><b>S( UV_TTDOITAC)</b></p> <p>//Xin khóa đọc</p>
		<b>WAITFOR DELAY</b> '0:0:05'	

<p>B2: Cập nhật tên người đại diện</p> <pre>UPDATE UV_TTDOITAC SET NGUOI_DAI_DIEN = @NGUOI_DAI_DIEN</pre>	<p><b>X(UV_TTDOITAC)</b></p> <p>// Xin khóa ghi</p>		
<p>B3: Bắt lỗi hệ thống và in ra lỗi (nếu có)</p> <pre>BEGIN CATCH DECLARE @ErrorMsg VARCHAR(2000) SELECT @ErrorMsg = N'Lỗi: ' + ERROR_MESSAGE() RAISERROR(@ErrorMsg, 16,1) ROLLBACK TRAN RETURN END CATCH</pre>			
<p><b>COMMIT</b></p>			

		<p>B2: Cập nhật tên đường</p> <pre>UPDATE UV_TTDOITAC SET DUONG = @DUONG</pre>	<p><b>X(UV_TTDOITAC)</b></p> <p>// Xin khóa ghi</p>
		<p>B3: Bắt lỗi hệ thống và in ra lỗi (nếu có)</p> <pre>BEGIN CATCH DECLARE @ErrorMsg VARCHAR(2000) SELECT @ErrorMsg = N'Lỗi: ' + ERROR_MESSAGE() RAISERROR(@ErrorMsg, 16,1) ROLLBACK TRAN RETURN END CATCH</pre>	
		<b>COMMIT</b>	