

BÁO CÁO LẦN 3

A) Cài đặt khoá các stored procedures chính (mã giả)

A.1) Bộ phận chăm sóc khách hàng

A.1.a) <usp_DieuChinhPhanHangKhachHang>

<u><usp_DieuChinhPhanHangKhachHang></u>	Khóa trên bảng dữ liệu
Input : <ul style="list-style-type: none">- @SDT VARCHAR(11): Số điện thoại của khách hàng cần điều chỉnh phân hạng.	
Output : None <ul style="list-style-type: none">- Chỉ in thông báo thành công hoặc lỗi	
1. BEGIN TRANSACTION	
2. SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
3. Kiểm tra tính hợp lệ của số điện thoại: Truy vấn xem số điện thoại có tồn tại hay không trong bảng KHACHHANG.	READ(KHACHHANG) WITH READCOMMITTED
4. Tính tổng chi tiêu của khách hàng: Lấy thông tin tổng giá trị đơn hàng từ bảng DonHang với điều kiện phù hợp.	READ(DonHang) WITH REPEATABLEREAD
5. Điều chỉnh phân hạng khách hàng: Tính loại thẻ dựa trên tổng chi tiêu của khách hàng.	
6. Cập nhật loại thẻ: Cập nhật thông tin LoaiThe trong bảng KHACHHANGTHE.	WRITE(KHACHHANG THE) WITH HOLDLOCK, ROWLOCK
7. COMMIT TRANSACTION	

A.1.b) usp_CapNhatPhanHangDauThang

<u>usp_CapNhatPhanHangDauThang</u>	Khoá trên bảng dữ liệu
Input : None	
Output : None <ul style="list-style-type: none">- Chỉ in thông báo thành công hoặc lỗi	
1. BEGIN TRANSACTION	
2. SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
3. Truy vấn danh sách tất cả khách hàng: Truy vấn thông tin khách hàng từ bảng KHACHHANG và tổng chi tiêu từ bảng DonHang.	READ(KHACHHANG)

	WITH READCOMMITTED READ(DonHang) WITH REPEATABLEREAD
4. Tính toán loại thẻ: Xác định LoạiThe dựa trên tổng chi tiêu trong 1 năm qua.	Nội bộ trong bảng tạm (@DanhSachKhachHang)
5. Cập nhật loại thẻ: Cập nhật LoạiThe trong bảng KHACHHANGTHE dựa trên kết quả tính toán.	WRITE(KHACHHANGTHE) WITH HOLDLOCK, ROWLOCK
6. COMMIT TRANSACTION	

A.1.c) usp_QuanLyTaiKhoanKhachHang

<usp_QuanLyTaiKhoanKhachHang>	Khoá trên bảng dữ liệu
Input : <ul style="list-style-type: none"> - @SDT: Số điện thoại của khách hàng - @Ten: Tên khách hàng - @LoạiThe: Loại thẻ - @NgàyLapThe: Ngày lập thẻ - @Action: Xác định hành động ('C': Thêm, 'U': Sửa, 'D': Xóa) 	
Output : None <ul style="list-style-type: none"> - Chỉ in thông báo thành công hoặc lỗi. 	
1. BEGIN TRANSACTION	
2. SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
3. Kiểm tra hành động: Xác định hành động là Thêm ('C'), Sửa ('U'), Xóa ('D').	
4. Thêm tài khoản khách hàng	READ(KHACHHANG WITH UPDLOCK, HOLDLOCK WRITE(KHACHHANG) WITH HOLDLOCK, ROWLOCK WRITE(KHACHHANGTHE) WITH HOLDLOCK, ROWLOCK
5. Sửa thông tin khách hàng	READ(KHACHHANG) WITH UPDLOCK, HOLDLOCK WRITE(KHACHHANG) WITH ROWLOCK WRITE(KHACHHANGTHE) WITH ROWLOCK

6. Xóa tài khoản khách hàng	READ(KHACHHANG) WITH UPDLOCK, HOLDLOCK DELETE(KHACHHANG) WITH ROWLOCK DELETE(KHACHHANGT HE) WITH ROWLOCK
7. COMMIT TRANSACTION	

A.1.d) usp_Gui1PhieuMuaHang

<usp_Gui1PhieuMuaHang>	Khoá trên bảng dữ liệu
Input : - @SDT: Số điện thoại của khách hàng - @Ten: Tên khách hàng - @LoaiThe: Loại thẻ - @NgayLapThe: Ngày lập thẻ - @Action: Xác định hành động ('C': Thêm, 'U': Sửa, 'D': Xóa)	
Output : @MaPhieu, @TienGiamGia, @NgayBatDau, @NgayKetThuc	
1. BEGIN TRANSACTION	
2. SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
3. Kiểm tra số điện thoại: Kiểm tra xem khách hàng có tồn tại trong hệ thống hay không.	READ(KHACHHANG) WITH UPDLOCK, HOLDLOCK
4. Tính giá trị tiền giảm giá: Dựa vào loại thẻ của khách hàng, xác định giá trị giảm giá.	
5. Tạo mã phiếu giảm giá ngẫu nhiên: Sinh một mã giảm giá dựa trên hàm CHECKSUM(NEWID()).	
6. Xác định ngày bắt đầu và kết thúc hiệu lực: Ngày bắt đầu là hiện tại (GETDATE()), ngày kết thúc là một tháng sau (DATEADD(MONTH, 1, @NgayBatDau)).	
7. Thêm phiếu giảm giá vào bảng PHIEUGIAMGIA: Lưu thông tin phiếu giảm giá vào bảng với các giá trị tính được.	WRITE(PHIEUGIAMGIA) WITH HOLDLOCK, ROWLOCK
8. COMMIT TRANSACTION	

A.1.e) usp_GuiPhieuMuaHangSinhNhat

<usp_GuiPhieuMuaHangSinhNhat>	Khoá trên bảng dữ liệu
Input : - @ThangSinh INT	
Output : None - Chỉ in thông báo thành công hoặc lỗi.	
1. SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
2. Truy vấn danh sách khách hàng có sinh nhật trong tháng: Truy xuất khách hàng từ bảng KHACHHANG và	READ(KHACHHANG)

KHACHHANGTHE	WITH UPDLOCK, HOLDLOCK READ(KHACHHANGTHE) WITH UPDLOCK, HOLDLOCK
3. Duyệt từng khách hàng trong danh sách: Sử dụng con trỏ (CURSOR) để xử lý từng khách hàng có sinh nhật.	
4. Tính tiền giảm giá: Dựa vào loại thẻ của khách hàng để xác định số tiền giảm giá.	
5. Sinh mã phiếu giảm giá: Tạo mã phiếu giảm giá ngẫu nhiên dựa trên hàm CHECKSUM(NEWID()).	
6. Gửi phiếu giảm giá vào bảng PHIEUGIAMGIA: Thêm phiếu giảm giá với thông tin chi tiết vào bảng PHIEUGIAMGIA.	WRITE(PHIEUGIAMGIA) WITH HOLDLOCK, ROWLOCK
7. Đóng và giải phóng con trỏ: Sau khi hoàn tất việc xử lý từng khách hàng, đóng và giải phóng con trỏ KháchHàngCursor.	
8. COMMIT TRANSACTION	

A.2) Bộ phận quản lý ngành hàng

A.2.a) usp_ThietLapChuongTrinhFlashSale

< usp_ThietLapChuongTrinhFlashSale >	Khóa trên bảng dữ liệu
Input : <ul style="list-style-type: none"> • @IDSanPham VARCHAR(15): ID sản phẩm muốn thiết lập khuyến mãi. • @TyleGiam INT: Phần trăm giảm giá muốn giảm trong chương trình. • @ThoiGianBatDau DATE: Ngày bắt đầu khuyến mãi. • @ThoiGianKetThuc DATE: Ngày kết thúc khuyến mãi. • @Soluong INT: Số lượng sản phẩm tối đa được áp dụng khuyến mãi. 	
Output : <ul style="list-style-type: none"> • @Ketqua NVARCHAR(50): Kết quả việc thêm chương trình sale thành công hay không 	
1. BEGIN TRANSACTION	
2. SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
3. Kiểm tra số lượng hàng tồn xem có sản phẩm có tồn nhiều hơn hoặc bằng @Soluong hay không (Sử dụng f_KiemTraSoLuongTonKho)	READ (KhoHang) WITH HOLD LOCK,ROW LOCK

4. Thêm chương trình sale vào vào bảng sale	WRITE(GiamGia) WITH TABLOCK
5. Thêm chương trình flashsale vào bảng FlashSale	WRITE(FlashSale) WITH TABLOCK
6. Trả kết quả chuỗi thông báo thêm thành công hay thất bại	
7. COMMIT TRANSACTION	

A.2.b) usp_ThietLapChuongTrinhComboSale

< usp_ThietLapChuongTrinhComboSale >	Khóa trên bảng dữ liệu
<p>Input :</p> <ul style="list-style-type: none"> • @IDSanPham1 VARCHAR(15): ID sản phẩm A muốn thiết lập khuyến mãi. • @IDSanPham2 VARCHAR(15): ID sản phẩm B muốn thiết lập khuyến mãi. • @TyLeGiam INT: Phần trăm giảm giá muốn giảm trong chương trình. • @ThoiGianBatDau DATE: Ngày bắt đầu khuyến mãi. • @ThoiGianKetThuc DATE: Ngày kết thúc khuyến mãi. • @Soluong INT: Số lượng sản phẩm tối đa được áp dụng khuyến mãi. 	
<p>Output :</p> <ul style="list-style-type: none"> • @Ketqua NVARCHAR(50): Kết quả việc thêm chương trình sale thành công hay không 	
1. BEGIN TRANSACTION	
2. SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
3. Kiểm tra số lượng hàng tồn xem có sản phẩm có tồn nhiều hơn hoặc bằng @Soluong hay không (Sử dụng f_KiemTraSoLuongTonKho)	READ (KhoHang) WITH HOLD LOCK,ROW LOCK
4. Thêm chương trình sale vào vào bảng sale	WRITE(GiamGia) WITH TABLOCK
5. Thêm chương trình flashsale vào bảng ComboSale	WRITE(ComboSale) WITH TABLOCK

6. Trả kết quả chuỗi thông báo thêm thành công hay thất bại	
7. COMMIT TRANSACTION	

A.2.c) usp_ThietLapChuongTrinhMemberSale

< usp_ThietLapChuongTrinhMemberSale >	Khóa trên bảng dữ liệu
<p>Input :</p> <ul style="list-style-type: none"> • @IDSanPham VARCHAR(15): ID sản phẩm muốn thiết lập khuyến mãi. • – @LoaiKhachHang NVARCHAR(30): Loại khách hàng muốn thiết lập khuyến mãi. • – @TyLeGiam INT: Phần trăm giảm giá muốn giảm trong chương trình. • – @ThoiGianBatDau DATE: Ngày bắt đầu khuyến mãi. • – @ThoiGianKetThuc DATE: Ngày kết thúc khuyến mãi. • – @Soluong INT: Số lượng sản phẩm tối đa được áp dụng khuyến mãi. 	
<p>Output :</p> <ul style="list-style-type: none"> • @Ketqua NVARCHAR(50): Kết quả việc thêm chương trình sale thành công hay không 	
1. BEGIN TRANSACTION	
2. SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
3. Kiểm tra số lượng hàng tồn xem có sản phẩm có tồn nhiều hơn hoặc bằng @Soluong hay không (Sử dụng f_KiemTraSoLuongTonKho)	READ (KhoHang) WITH HOLD LOCK,ROW LOCK
4. Thêm chương trình sale vào vào bảng sale	WRITE(GiamGia) WITH TABLOCK
5. Thêm chương trình flashsale vào bảng MemberSale	WRITE(MemberSale) WITH TABLOCK
6. Trả kết quả chuỗi thông báo thêm thành công hay thất bại	
7. COMMIT TRANSACTION	

A.2.d) usp_KiemTraHoatDongSale

< usp_KiemTraHoatDongSale >	Khóa trên bảng dữ liệu
Input : <ul style="list-style-type: none">@MaGiamGia VARCHAR(15): ID sản phẩm muốn thiết lập trạng thái khuyến mãi.	
Output :	
BEGIN TRANSACTION	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ	
1. Truy cập bảng GIAMGIA và update TinhTrangGiamGia thành 0 nếu số lượng của GIAMGIA có MaGiamGia = @MaGiamGia = 0 hoặc ngày kết thúc giảm giá đã đến	UPDATE(GIAMGIA) WITH HOLD LOCK,ROW LOCK
COMMIT	

A.2.e) Các procedure phụ

A.2.e.1) f_KiemTraSoLuongTonKho (đây là function nên không thể begin transaction)

< f_KiemTraSoLuongTonKho >	Khóa trên bảng dữ liệu
Call by: <ul style="list-style-type: none">usp_ThietLapChuongTrinhFlashSaleusp_ThietLapChuongTrinhComboSaleusp_ThietLapChuongTrinhMemberSale	
Input : <ul style="list-style-type: none">@IDSanPham VARCHAR(15): ID sản phẩm muốn thiết lập khuyến mãi.@SoLuong INT: Số lượng cần để thiết lập chương trình sale	
Output : <ul style="list-style-type: none">@Ketqua NVARCHAR(50): Kết quả việc có đủ sản phẩm mở sale không.	

1. Truy cập kho hàng check số hàng tồn của sản phẩm có @IDSanPham và so sánh với @SoLuong	READ(KhoHang) WITH HOLD LOCK,ROW LOCK
2. Trả kết quả chuỗi thông báo đủ hay không số lượng cần thiết	

A.3) Bộ phận xử lý đơn hàng

A.3.a) usp_TaoDonHang

< usp_TaoDonHang >	Khóa trên bảng dữ liệu
Input : <ul style="list-style-type: none"> • @ChiTietDonHang TABLE: 1 table chứa các dòng dữ liệu về sản phẩm của 1 đơn hàng với các cột là IDSảnPhẩm, Số lượng. • @SDT VARCHAR(15): Số điện thoại của khách hàng, dùng định danh khách hàng • @isOnline bit: 1 nếu online, 0 nếu offline • @CachThanhToan NVARCHAR(50): phương thức thanh toán • @diaChiGiaoHang: chỉ có nếu @isOnline = 1, nếu không thì null, là dữ liệu địa chủ giao hàng cho khách hàng 	
Output : <ul style="list-style-type: none"> • @Ketqua NVARCHAR(50): Kết quả tạo đơn thành công. 	
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
1. Insert 1 đơn hàng mới , generate mã đơn hàng bằng (max(mã đơn hàng) IN ĐơnHàng) +1. Tổng giá trị đơn hàng để null, bước xử lý đơn hàng sẽ xử lý.	WRITE(DonHang) WITH TABLOCK
2. Với mọi dòng sản phẩm trong đơn hàng trong @ChiTietDonHang.	
BEGIN TRANSACTION	
2.1 Check trong kho hàng có đủ sản phẩm đó không	READ(KhoHang) WITH (ROW LOCK, HOLD LOCK)

2.2 Check món hàng này có khuyến mãi không(áp dụng theo thứ tự Flash - Combo – Member)	READ(GiamGia) WITH (ROW LOCK, HOLD LOCK)
2.3 Nếu có khuyến mãi, tính giá cho sản phẩm này theo khuyến mãi có trong hệ thống	
2.4 Tổng giá của mọi sản phẩm vào biến tạm. Đồng thời cập nhật số lượng tồn kho, số lượng giảm giá	UPDATE(KhoHang),(GiamGia) WITH (ROWLOCK, HOLDLOCK)
2.5 Insert vào bảng sản phẩm đơn hàng các giá trị là MaDonHang, IDSanPham, Số lượng.	INSERT(SanPhamDonHang) WITH TABLOCK
2.6 Update Số lượng hàng tồn của kho hàng	UPDATE(KhoHang) WITH (ROWLOCK, HOLDLOCK)
2.7 Update Số lượng giảm giá còn lại của giảm giá được áp dụng	UPDATE(GIAMGIA) WITH (ROWLOCK, HOLDLOCK)
COMMIT TRANSACTION	
3. Sử dụng usp_XuLyDonHang để tính giá cuối cùng của đơn hàng. Input là biến tạm lưu tổng giá trị đơn hàng được nói ở 2.4	
4. Thông báo thêm thành công hay thất bại	

A.3.b) usp_XuLyDonHang

< usp_XuLyDonHang >	Khóa trên bảng dữ liệu
Call by: <ul style="list-style-type: none"> usp_TaoHoaDon 	
Input : <ul style="list-style-type: none"> @MaGiamGia VARCHAR(15): ID sản phẩm muốn thiết lập khuyến mãi. @SDT VARCHAR(15): Định danh của Khách hàng @TongGiaTriDonHang INT: Tổng giá trị đơn hàng 	
Output :	
BEGIN TRANSACTION	

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
1. Truy cập bảng PHIEUGIAMGIA để lấy Phiếu Quà Tặng của khách hàng.	READ PHIEUGIAMGIA WITH HOLD LOCK, ROW LOCK
2. Tính tổng số tiền sau giảm giá	
3. Cập nhật tổng giá trị đơn hàng vào DonHang , cập nhật tình trạng đơn hàng là N'Thành công'	UPDATE(DonHang) WITH HOLD LOCK, ROW LOCK
4. Xóa phiếu quà tặng của khách hàng	DELETE(PHIEUGIAMGIA) WITH ROWLOCK
COMMIT	

A.4) Bộ phận quản lý kho hàng

A.4.a) usp_KiemTraDieuKienDatHang

usp_KiemTraDieuKienDatHang	Thao tác thực hiện trên bảng dữ liệu
Call by: usp_TinhHangCanDat	
Input : - @IDSanPham VARCHAR(15): Sản phẩm muốn đặt thêm. - @SL_DatThem INT: Số lượng hàng muốn đặt thêm.	
Output : - @Ketqua INT: 1 tức có thể đặt và 0 tức không thỏa điều kiện.	
1. Kiểm tra sản phẩm muốn đặt thêm là sản phẩm có sẵn trong Bảng “Sản phẩm” hay là sản phẩm mới.	READ(SANPHAM) WITH UPD LOCK
1.1. Nếu sản phẩm muốn đặt là sản phẩm mới chưa có trong bảng “Sản phẩm”	
1.2. Insert sản phẩm mới vào bảng “Sản phẩm”.	WRITE(SANPHAM)
1.3. Insert kho hàng mới cho sản phẩm mới với tham số @SL_SP_TĐ mới.	WRITE(KHOHANG)
1.4. Nếu sản phẩm muốn đặt là sản phẩm có sẵn thì ta thực hiện bước 2.	
2. Kiểm tra số lượng hàng đặt @SL_DatThem có lớn hơn 10% số lượng hàng tồn tối đa @SL_SP_TĐ và số lượng hàng tồn kho của sản phẩm @SL_HT trong bảng “Kho Hàng”.	READ(KHOHANG) WITH UPD LOCK
3. Thông báo có đủ điều kiện đặt hàng hay không (1 là có thể đặt, 0 là không thể đặt).	

A.4.b) usp_TinhHangCanDat

usp_TinhHangCanDat	Thao tác thực hiện trên bảng dữ liệu
Input : - @IDSanPham VARCHAR(15): Sản phẩm muốn đặt.	
Output : - @SLNenDat INT: Số lượng hàng nên đặt cho sản phẩm này.	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ	
BEGIN TRANSACTION	
1. Kiểm tra @IDSanPham có tồn tại hay không. Nếu không tồn tại thì báo lỗi không tìm thấy sản phẩm.	READ(SANPHAM) WITH UPD LOCK
2. Thực hiện usp_KiemtraDieuKienDatHang để kiểm tra xem có đủ điều kiện để đặt hàng hay không	
3. Tính số lượng hàng đặt bằng cách lấy số lượng hàng tồn tối đa trừ cho số lượng hàng tồn còn trong kho và trừ đi số lượng hàng chưa giao từ phiếu đặt trước đây. (@SL_SP_TĐ - @SL_HT - @SLHCG)	READ(KHOHANG) WITH UPD LOCK
4. Thông báo số lượng hàng nên đặt cho sản phẩm.	

A.4.c) usp_ThucHienDatHang

usp_TinhHangCanDat	Thao tác thực hiện trên bảng dữ liệu
Input : - @IDSanPham VARCHAR(15): Sản phẩm muốn đặt. - @SLDat INT: Lượng hàng muốn đặt. (Kết quả từ usp_TinhHangCanDat).	
Output : - @TrangThaiDatHang BIT: Trạng thái đặt hàng (1 nếu thành công, 0 nếu thất bại).	
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
BEGIN TRANSACTION	
1. Kiểm tra @IDSanPham có tồn tại hay không. Nếu không tồn tại thì báo lỗi không tìm thấy sản phẩm.	READ(SANPHAM) WITH HOLDLOCK, UPDLOCK
2. Tạo phiếu đặt mới bằng usp_TaoPhieuDat	
3. Tạo thông tin hàng đặt chi tiết với tham số @IDSanPham, @SLDat và mã phiếu đặt từ phiếu đặt vừa mới tạo ra bằng usp_TaoHangDat.	
4. Thông báo trạng thái đặt hàng của sản phẩm.	

A.4.d) Các procedure phụ

A.4.d.1. usp_CapNhatSL_HT

usp_CapNhatSL_HT	Thao tác thực hiện trên bảng dữ liệu
Call by: usp_TinhHangCanDat	
Input : <ul style="list-style-type: none">- @IDSanPham VARCHAR(15): Sản phẩm muốn cập nhật SL_HT sau khi đã bán.- @SL_HangDaBan INT: Số lượng hàng tồn bán ra.	
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
BEGIN TRANSACTION	
Output : <ul style="list-style-type: none">- @SL_HangTonMoi INT: Số lượng hàng tồn sau khi bán ra, nhập vào hoặc bị hủy đơn.	
1. Kiểm tra @IDSanPham có tồn tại hay không. Nếu không tồn tại thì báo lỗi không tìm thấy sản phẩm.	READ(SANPHAM) WITH UPD LOCK
2. Kiểm tra @SL_HangDaBan có vượt hơn số lượng hàng tồn hiện có trong kho hàng. Nếu vượt hơn thì báo lỗi @SL_HangDaBan không thể vượt hơn hàng tồn hiện có.	READ(KHOHANG)
3. Tính số lượng hàng tồn còn lại của sản phẩm sau khi đã bán ra bằng cách lấy @SL_HT trong bảng “Kho Hàng” trừ đi @SL_HangTonMoi số lượng hàng tồn đã bán ra. 4. Thông báo số lượng hàng tồn sau cập nhật.	UPDATE(KHOHANG)

A.4.d.2. usp_CapNhatSL_HDG

usp_CapNhatSL_HDG	Thao tác thực hiện trên bảng dữ liệu
Call by: usp_TinhHangCanDat	
Input : <ul style="list-style-type: none">- @IDSanPham VARCHAR(15): Sản phẩm được giao tới.- @SL_HangDaGiao INT: Số lượng hàng đã được giao.	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ	
BEGIN TRANSACTION	
Output : <ul style="list-style-type: none">- @SL_HangDaGiaoSau INT: Số lượng hàng đã giao sau khi được cập nhật.	
1. Kiểm tra @IDSanPham có tồn tại hay không. Nếu không tồn tại thì báo lỗi không tìm thấy sản phẩm.	READ(SANPHAM) WITH UPD LOCK
2. Cập nhật số lượng hàng đã được giao khi có đợt giao hàng đến kho. 3. Thông báo số lượng hàng đã giao sau khi được cập nhật.	UPDATE(HANGDAT)

A.4.d.3. usp_CapNhatSL_HCG

usp_CapNhatSL_HCG	Thao tác thực hiện trên bảng dữ liệu
Call by: usp_TinhHangCanDat	
Call by: usp_CapNhatSL_HĐG	
Input : <ul style="list-style-type: none"> - @IDSanPham VARCHAR(15): Sản phẩm được giao tới. - @SL_HangGiaoThem INT: Số lượng hàng đã được giao nhưng chưa tới kho hàng. 	
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
BEGIN TRANSACTION	
Output : <ul style="list-style-type: none"> - @SL_HangTonSau INT: Số lượng hàng tồn trong kho sau khi được cập nhật. - @SL_HangCGSau INT: Số lượng hàng chưa giao tới kho sau khi được cập nhật. 	
1. Kiểm tra @IDSanPham có tồn tại hay không. Nếu không tồn tại thì báo lỗi không tìm thấy sản phẩm.	READ(SANPHAM) WITH UPD LOCK
2. Cập nhật số lượng hàng tồn kho sau khi có đợt giao hàng đến kho. Tổng của hàng tồn hiện tại có trong kho và số lượng hàng đã giao, kết quả từ usp_CapNhatSL_HĐG.	UPDATE(KHOHANG)
3. Cập nhật số lượng hàng chưa giao sau khi có đợt giao hàng đến kho. Tổng của kết quả của bước 2 và @SL_HangGiaoThem.	UPDATE(KHOHANG)
4. Thông báo số lượng hàng tương lai của những hàng đã đặt nhưng mà chưa giao tới kho.	

A.4.d.4. usp_TaoPhieuDat

usp_TaoPhieuDat	Thao tác thực hiện trên bảng dữ liệu
Call by: usp_ThucHienDatHang	
Input : Không có	
Output : <ul style="list-style-type: none"> - @MaPhieuDatMoi VARCHAR(15): Mã phiếu đặt mới sau khi được tạo (In ra thông tin phiếu đặt).@SL_HangCGSau INT: Số lượng hàng chưa giao tới kho sau khi được cập nhật. - @NgayDatMoi DATE: Ngày đặt hàng/phiếu sau khi được đặt mới. 	
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
BEGIN TRANSACTION	
1. Insert thêm thông tin phiếu đặt mới bằng cách @MaPhieuDat + 1 và @NgayDatMoi là ngày đặt hôm nay tạo phiếu đặt	WRITE(PHIEUDAT)

2. Thông báo thông tin phiếu đặt vừa mới được tạo ra.	
---	--

A.4.d.5. usp_TaoHangDat

usp_HangDat	Thao tác thực hiện trên bảng dữ liệu
Call by: usp_ThucHienDatHang	
Input : <ul style="list-style-type: none"> - @MaPhieuDat VARCHAR(15): Mã phiếu đặt. - @IDSanPham VARCHAR(15): Sản phẩm muốn đặt. - @SLDat INT: Số lượng hàng muốn đặt. - @SLDaGiao INT: Số lượng hàng đã giao. 	
Output : <ul style="list-style-type: none"> - @MaPhieuDat2 VARCHAR(15): Mã phiếu đặt (Xuất thông tin phiếu đặt chi tiết sau khi tạo). - @IDSanPham2 VARCHAR(15): Sản phẩm muốn đặt. - @SLDat2 INT: Số lượng hàng muốn đặt. - @SLDaGiao2 INT: Số lượng hàng đã giao. 	
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
BEGIN TRANSACTION	
1. Insert thêm thông tin mã phiếu đặt, ID sản phẩm, số lượng hàng muốn đặt, số lượng hàng đã giao.	WRITE(HANGDAT)
2. Thông báo thông tin phiếu đặt chi tiết vừa mới được tạo ra.	

A.5) Bộ phận kinh doanh

A.5.a) usp_KhachHangDoanhThu

usp_KhachHangDoanhThu	Khóa trên bảng dữ liệu
Input : <ul style="list-style-type: none"> - @Ngày DATE: Ngày cần thống kê. 	
Output : <ul style="list-style-type: none"> - Table gồm tổng số lượng khách hàng và doanh thu trong ngày 	
BEGIN TRANSACTION	
SET TRAN ISOLATION LEVEL SERIALIZABLE	
1. Đọc bảng Đơn hàng để lấy những đơn hàng không bị hủy trong ngày và tính tổng doanh thu Commit tran	Read(DonHang) WITH ROW LOCK, HOLDLOCK

A.5.b) usp_SoLuongDaBanVaKhachMua

usp_SoLuongDaBanVaKhachMua	Khóa trên bảng dữ liệu
Input : <ul style="list-style-type: none"> - @Ngày DATE: Ngày cần thống kê. 	

Output :	
- Table gồm những sản phẩm được mua, số lượng khách mua cũng như số lượng bán ra của sản phẩm đó.	
BEGIN TRANSACTION	
SET TRAN ISOLATION LEVEL READ COMMITTED	
1. Đọc sản phẩm, số lượng bán và đơn bán sản phẩm để biết các đơn bán sản phẩm đó trong bảng SanPhamDonHang	Read(SanPhamDonHang) WITH ROW LOCK
2. Đọc tiếp sang bảng DonHang để xác định những đơn hàng trong @Ngày này	Read(DonHang) WITH ROW LOCK
3. Xử lý và trả ra những sản phẩm được mua, số lượng khách mua và số lượng bán ra. Commit tran	

A.5.c) usp_ThongKeTraHang

usp_ThongKeTraHang	Khóa trên bảng dữ liệu
Input :	
Output :	
- Table: danh sách thống kê số lượng của các sản phẩm bị trả lại	
BEGIN TRANSACTION	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ	
1. Đọc bảng Đơn hàng để thống kê các đơn bị trả (attribute TinhTrangDonHang = “Trả lại”)	Read(DonHang) WITH REPEATABLE READ
2. Đọc bảng Sản phẩm đơn hàng để thống kê các sản phẩm trong đơn bị trả gồm mã sản phẩm và số lượng	Read(SanPhamDonHang) WITH REPEATABLE READ
3. Đọc bảng Sản phẩm để biết giá niêm yết của 1 sản phẩm đó	Read(SanPham) WITH REPEATABLE READ
4. Trả ra kết quả những mặt hàng bị trả lại cũng như số tiền mất mát do không bán được lô hàng đó. Commit tran	

A.5.d) usp_ThangMuaSam

usp_ThangMuaSam	Khóa trên bảng dữ liệu
Input :	
Output :	
- Table: Trả về bảng gồm tháng và sản phẩm được mua nhiều nhất trong tháng đó, cùng với số lượng của nó.	
BEGIN TRANSACTION	
SET TRAN ISOLATION LEVEL SERIALIZABLE	
1. Xác định những đơn hàng nào đã được mua mà không bị hủy	Read(DonHang) WITH TAB LOCK
2. Xác định số lượng của những sản phẩm không bị hủy đó	Read(SanPhamDonHang) WITH TAB LOCK

3. Kiểm tra đơn hàng đó có phải số lượng bán nhiều nhất trong tháng đó hay không	
4. Trả về kết quả sản phẩm được mua nhiều nhất theo tháng. COMMIT tran	

B) Cài đặt khoá các stored procedures phát sinh (mã giả)

B.1) usp_TraLaiDonHang

< usp_TraLaiDonHang >	Khoá trên bảng dữ liệu
Input : @MaDonHang VARCHAR(15): Mã đơn cần trả hàng	
Output :	
1. Check trong bảng DonHang có mã đơn hàng input không	READ(DonHang) WITH (ROWLOCK, HOLDLOCK)
2. Update trạng thái đơn thành trả lại	UPDATE(DonHang) WITH (ROWLOCK, HOLDLOCK)
3. Lấy dữ liệu các sản phẩm trong đơn hàng cần trả lại	READ(SanPhamDonHang) WITH (ROWLOCK, HOLDLOCK)
4. Với mỗi sản phẩm trong bảng dữ liệu các sản phẩm trong đơn hàng cần trả lại	
BEGIN TRANSACTION	
4.1 Update bảng KhoHang, cập nhật cộng lại số lượng vào SL_HT	UPDATE (KhoHang) WITH (ROWLOCK, HOLDLOCK)
COMMIT TRANSACTION	