

BÁO CÁO LẦN 2

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

A.1) Bộ phận chăm sóc khách hàng

A.1.a) usp_DieuChinhPhanHangKhachHang

<usp_DieuChinhPhanHangKhachHang>	Thao tác thực hiện trên bảng dữ liệu
Input : - @SDT VARCHAR(11): Số điện thoại của khách hàng muốn thay đổi phân hạng.	
Output : None	
1. Kiểm tra tính hợp lệ của số điện thoại: Truy vấn xem số điện thoại có tồn tại trong bảng	Read(KhachHang)
2. Tính tổng chi tiêu năm trước: Truy vấn tổng chi tiêu từ bảng HoaDon.	Read(HoaDon) Write(HoaDon)
3. Điều chỉnh phân hạng: Dựa vào tổng chi tiêu, cập nhật phân hạng của khách hàng trong bảng KhachHang.	Write(KhachHang)

A.1.b) usp_CapNhatPhanHangDauThang

<usp_CapNhatPhanHangDauThang>	Thao tác thực hiện trên bảng dữ liệu
Input : None	
Output : None	
1. Truy vấn tất cả khách hàng: Lấy danh sách tất cả khách hàng từ bảng KhachHang.	Read(KhachHang)
2. Tính tổng chi tiêu của từng khách hàng: Lấy thông tin từ bảng HoaDon để tính tổng chi tiêu của từng khách hàng.	Read(HoaDon)
3. Cập nhật phân hạng cho tất cả khách hàng: Dựa vào tổng chi tiêu, cập nhật phân hạng của từng khách hàng trong bảng KhachHang.	Write(KhachHang)

A.1.c) <usp_QuanLyTaiKhoanKhachHang>

<usp_QuanLyTaiKhoanKhachHang>	Thao tác thực hiện trên bảng dữ liệu
-------------------------------	--------------------------------------

Input : <ul style="list-style-type: none"> - @SDT VARCHAR(11): Số điện thoại của khách hàng. - @Ten NVARCHAR(50): Tên khách hàng. - @LoaiThe NVARCHAR(30): Loại thẻ khách hàng. - @NgayLapThe DATE: Ngày lập thẻ. - @Action CHAR(1): Xác định hành động ('C' = Tạo, 'U' = Sửa, 'D' = Xóa). 	
Output : None	
1. Kiểm tra hành động (@Action): Kiểm tra xem hành động là Thêm (C), Sửa (U), hay Xóa (D).	
2. Insert (Thêm mới): Nếu @Action = 'C', thêm tài khoản khách hàng vào bảng KháchHang.	Write(KhachHang)
3. Update (Sửa): Nếu @Action = 'U', sửa thông tin khách hàng (Tên, Loại thẻ, Ngày lập thẻ) trong bảng KháchHang	Update(KhachHang)
4. Delete (Xóa): Nếu @Action = 'D', xóa khách hàng khỏi bảng KháchHang.	Delete(KhachHang)

A.1.d) usp_Gui1PhieuMuaHang

<usp_Gui1PhieuMuaHang>	Thao tác thực hiện trên bảng dữ liệu
Call by : <ul style="list-style-type: none"> - usp_QuanLyTaiKhoanKhachHang(có thể gọi để gửi phiếu giảm giá khi tạo mới tài khoản khách hàng). 	
Input : <ul style="list-style-type: none"> - @SDT VARCHAR(11): Số điện thoại của khách hàng. - @LoaiThe NVARCHAR(20): Loại thẻ khách hàng (để xác định giá trị của phiếu) 	
Output : <ul style="list-style-type: none"> - @MaPhieu VARCHAR(15): Mã phiếu giảm giá. - @TienGiamGia INT: Số tiền giảm giá từ phiếu. - @NgayBatDau DATE: Ngày bắt đầu phiếu. - @NgayKetThuc: Ngày kết thúc hiệu lực phiếu. 	
1. Sinh mã phiếu giảm giá: Tạo mã phiếu giảm giá ngẫu nhiên. (@MaPhieu).	
2. Tính toán số tiền giảm giá: Dựa vào loại thẻ khách hàng (@LoaiThe), tính toán giá trị phiếu giảm giá (@TienGiamGia).	Read(KhachHang)
3. Xác định thời gian bắt đầu và kết thúc của phiếu: Cập nhật @NgayBatDau và @NgayKetThuc.	
4. Thêm phiếu giảm giá vào bảng PhieuGiamGia	Write(PhieuGiamGia)

A.1.e) usp_GuiPhieuMuaHangSinhNhat

<usp_GuiPhieuMuaHangSinhNhat>	Thao tác thực hiện trên bảng dữ liệu
Input : <ul style="list-style-type: none"> - @ThangSinh INT: Tháng sinh nhật của khách hàng (dùng để lọc 	

khách hàng trong tháng đó).	
Output : None	
1. Truy vấn khách hàng có sinh nhật trong tháng @ThangSinh: Lấy danh sách khách hàng từ bảng KhachHang	Read(KhachHang)
2. Tính toán phiếu giảm giá: Dựa trên phân hạng khách hàng, tính toán giá trị phiếu và thời gian hiệu lực của phiếu.	
3. Gửi phiếu giảm giá: Insert thông tin phiếu vào bảng PhieuGiamGia	Write(PhieuGiamGia)

A.2) Bộ phận quản lý ngành hàng

A.2.a) usp_ThietLapChuongTrinhFlashSale

< usp_ThietLapChuongTrinhFlashSale >	Thao tác thực hiện trên bảng dữ liệu
Call by: usp_QuanLyChuongTrinhKhuyenMai	
Input : <ul style="list-style-type: none"> - @IDSanPham VARCHAR(15): ID sản phẩm muốn thiết lập khuyến mãi. - @TyleGiam DECIMAL(3,2): 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 	
4. Kiểm tra số lượng hàng tồn xem có sản phẩm có tồn nhiều hơn hoặc bằng @Soluong hay không (Sử dụng usp_CheckSoLuongTonKho)	READ(KhoHang)
5. Thêm chương trình flashsale vào	WRITE(GiamGia) WRITE(FlashSale)
6. Trả kết quả chuỗi thông báo thêm thành công hay thất bại	

A.2.b) usp_ThietLapChuongTrinhComboSale

< usp_ThietLapChuongTrinhComboSale >	Thao tác thực hiện trên bảng dữ liệu
Call by: usp_QuanLyChuongTrinhKhuyenMai	
Input : <ul style="list-style-type: none"> - @IDSanPhamA VARCHAR(15): ID sản phẩm A muốn thiết lập khuyến mãi. - @IDSanPhamB VARCHAR(15): ID sản phẩm B muốn thiết lập khuyến mãi. 	

<ul style="list-style-type: none"> - @TyLeGiam DECIMAL(3,2): 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. Kiểm tra số lượng hàng tồn xem cả 2 sản phẩm có sản phẩm có tồn nhiều hơn hoặc bằng @Soluong hay không (Sử dụng usp_CheckSoLuongTonKho) <i>Note: 2 sản phẩm là 1 thì chỉ check 1 lần</i>	READ(KhoHang)
2. Thêm chương trình combosale vào	WRITE(GiamGia) WRITE(ComboSale)
3. Trả kết quả chuỗi thông báo thêm thành công hay thất bại	

A.2.c) usp_ThietLapChuongTrinhMemberSale

< usp_ThietLapChuongTrinhMemberSale >	Thao tác thực hiện trên bảng dữ liệu
Call by: usp_QuanLyChuongTrinhKhuyenMai	
Input : <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 DECIMAL(3,2): 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. Kiểm tra số lượng hàng tồn xem sản phẩm có sản phẩm có tồn nhiều hơn hoặc bằng @Soluong hay không (Sử dụng usp_CheckSoLuongTonKho)	READ(KhoHang)
2. Thêm chương trình membersale vào	WRITE(GiamGia) WRITE(MemberSale)
3. Trả kết quả chuỗi thông báo thêm thành công hay thất bại	

A.2.d) usp_QuanLyChuongTrinhKhuyenMai

< usp_QuanLyChuongTrinhKhuyenMai >	Thao tác thực hiện trên bảng dữ liệu
Input : <ul style="list-style-type: none">- @MaGiamGia NVARCHAR(30): Mã chương trình khuyến mãi.- @TyLeGiam DECIMAL(3,2): Tỷ lệ phần trăm giảm.- @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.- @TrangThaiKhuyenMai BIT: Trạng thái khuyến mãi muốn cập nhật.- @LoaiGiamGia VARCHAR(15): Loại sale muốn tạo.	
Output : <ul style="list-style-type: none">- @ThongBao NVARCHAR(50): Thông báo tạo chương trình sale thành công hay không.	
1. Kiểm tra mã sale đã tồn tại chưa.	READ(GiamGia)
2. Thêm chương trình giảm giá bằng các usp tương ứng	WRITE(GiamGia)
3. Trả về thông báo tạo thành công hay không	

A.2.e) usp_QuanLyKetThucChuongTrinhSale

< usp_QuanLyKetThucChuongTrinhSale >	Thao tác thực hiện trên bảng dữ liệu
Input : <ul style="list-style-type: none">- @MaGiamGia VARCHAR(15)	
Output :	
1. Kiểm tra số lượng hàng tồn xem sản phẩm có sản phẩm có còn hàng tồn hay không (Sử dụng usp_CheckHangTonSale)	READ(KhoHang)
1.1. Nếu số lượng hàng tồn không đủ hoặc số hàng sale của chương trình sale đã hết, đánh dấu chương trình giảm giá là kết thúc. (Các trường hợp là output của usp_CheckHangTonSale)	WRITE(GiamGia)
1.2. Nếu vẫn đủ hàng tồn sẽ không làm gì	

A.2.e) Các procedure phụ

A.2.e.1) usp_CheckSoLuongTonKho

< usp_CheckSoLuongTonKho >	Thao tác thực hiện trên bảng dữ liệu
Call by: <ul style="list-style-type: none">- usp_ThietLapChuongTrinhFlashSale- usp_ThietLapChuongTrinhComboSale- usp_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)
2. Trả kết quả chuỗi thông báo đủ hay không số lượng cần thiết	

A.2.e.2) usp_CheckHangTonSale

< usp_CheckHangTonSale >	Thao tác thực hiện trên bảng dữ liệu
Call by: usp_QuanLyKetThucChuongTrinhSale	
Input : <ul style="list-style-type: none"> - @MaGiamGia VARCHAR(15): Mã định dạng khuyến mãi cần xét. 	
Output : <ul style="list-style-type: none"> - @Ketqua NVARCHAR(50): Trạng thái thỏa các điều kiện đóng sale hay không 	
1. Truy cập vào thông tin của chương trình giảm giá đó và liên kết qua loại chương trình tương ứng.	READ(GiamGia)
2. Lấy thông tin của sản phẩm liên quan đến mã giảm giá đó	READ(FlashSale)/ READ(ComboSale)/ READ(MemberSale)
3. Xem xét số lượng của món hàng đó trong kho còn hay không và xem số lượng sale đã về 0 chưa	READ(KhoHang)
3.1 Nếu thỏa 1 trong 2 điều kiện trên, trả về kết quả “Hết hàng”	
3.2 Nếu không, trả về “Còn hàng”	

A.3) Bộ phận xử lý đơn hàng

A.3.a) usp_TaoDonHang

< usp_TaoDonHang >	Thao tác thực hiện 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 IDSanPham, Số lượng. - @SDT VARCHAR(15): Số điện thoại của khách hàng, dùng định danh khách hàng 	

<ul style="list-style-type: none"> - @isOnline bit: 1 nếu online, 0 nếu offline - @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. 	
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)
2. Với mọi dòng sản phẩm trong đơn hàng trong @ChiTietDonHang, nếu giá của hàng đó đang là null. <ul style="list-style-type: none"> a. Check trong kho còn đủ số hàng đó không b. Check món hàng này có khuyến mãi không(áp dụng theo thứ tự Flash - Combo – Member) c. Nếu có khuyến mãi, tính giá cho sản phẩm này theo khuyến mãi có trong hệ thống d. Tổng giá của mọi sản phẩm vào biến tạm e. Insert vào bảng sản phẩm đơn hàng các giá trị là MaDonHang, IDSanPham, Số lượng. 	READ(KhoHang) READ(GiamGia) WRITE(SanPhamDonHang)
3. Update tính tổng giá trị đơn hàng(usp_XuLyDonHang) với input là biến tạm lưu tổng giá trị ở trên.	
4. Thông báo thêm thành công hay thất bại	

A.3.b) usp_XuLyDonHang

< usp_XuLyDonHang >	Thao tác thực hiện trên bảng dữ liệu
Call by: usp_TaoHoaDon	
Input : <ul style="list-style-type: none"> - @MaDonHang VARCHAR(15): Mã đơn hàng cần xử lý - @SDT VARCHAR(15): Số điện thoại của khách hàng, dùng định danh khách hàng 	
Output : <ul style="list-style-type: none"> - @GiaTien Float: Tổng tiền cần thanh toán của đơn hàng 	
1. Đọc bảng Sản phẩm đơn hàng với MaDonHang = @MaDonHang	Read(SanPhamDonHang)
2. Xét xem có phiếu quà tặng của khách hàng @SDT không	Read(PhieuGiamGia)
3. Nếu có sẽ trừ trực tiếp vào tổng giá trị các sản phẩm ở trên. Lưu ý: Nếu hóa đơn có nhỏ hơn tổng giá trị phiếu tặng thì tổng giá trị hóa đơn cũng chỉ về 0, không bị âm.	
4. Update giá trị tổng hóa đơn vào Đơn hàng.	Update(DonHang)
5. Xóa phiếu tặng đã sử dụng của khách hàng	Delete(PhieuGiamGia)
6. Thông báo về giá trị của đơn hàng	

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 : <ul style="list-style-type: none">- @IDSanPham VARCHAR(15): Sản phẩm muốn đặt thêm.- @SL_DatThem INT: Số lượng hàng muốn đặt thêm.	
Output : <ul style="list-style-type: none">- @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)
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)
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 : <ul style="list-style-type: none">- @IDSanPham VARCHAR(15): Sản phẩm muốn đặt.	
Output : <ul style="list-style-type: none">- @SLNenDat INT: Số lượng hàng nên đặt cho sản phẩm này.	
1. Kiểm tra @IDSanPham có tồn tại hay không. Nếu không tồn tại thì báo lỗi không tìm thấy sản phẩm.	READ(SANPHAM)
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)	UPDATE(HANGDAT)
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 : <ul style="list-style-type: none">- @IDSanPham VARCHAR(15): Sản phẩm muốn đặt.- @SLDat INT: Lượng hàng muốn đặt. (Kết quả từ usp_ TinhHangCanDat).	
Output : <ul style="list-style-type: none">- @TrangThaiDatHang BIT: Trạng thái đặt hàng (1 nếu thành công, 0 nếu thất bại).	
1. Kiểm tra @IDSanPham có tồn tại hay không. Nếu không tồn tại thì báo lỗi không tìm thấy sản phẩm.	READ(SANPHAM)
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.	
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)
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.	UPDATE(KHOHANG)
4. Thông báo số lượng hàng tồn sau cập nhật.	

A.4.d.2. usp_CapNhatSL_HĐG

usp_CapNhatSL_HĐG	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. 	
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)
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_HDG	
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. 	
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)
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_HDG.	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_CapNhatSL_HCG

usp_CapNhatSL_HCG	Thao tác thực hiện trên bảng dữ liệu
Call by: usp_TinhHangCanDat	
Call by: usp_CapNhatSL_HDG	
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. 	
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. 	
5. Kiểm tra @IDSanPham có tồn tại hay không. Nếu không tồn tại thì báo lỗi không tìm thấy sản phẩm.	READ(SANPHAM)
6. 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)
7. 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)
8. Thông báo số lượng hàng tương lai của những hàng đã đặt nhưng mà chưa giao tới kho.	

A.4.d.5. 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. 	
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.6. 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. 	
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	Thao tác thực hiện trên bảng dữ liệu
Input :	
<ul style="list-style-type: none"> - @Ngay DATE: Ngày cần thống kê. 	
Output :	
<ul style="list-style-type: none"> - @SoLuongKhachHang INT: Tổng số khách trong ngày. - @Doanh Thu INT: Tổng doanh thu trong ngày. 	
1. Đọc bảng Đơn hàng để thống kê	Read(DonHang)
2. Trả về các dữ liệu tổng khách và tổng doanh thu của ngày	

A.5.b) usp_ThongKeTraHang

usp_ThongKeTraHang	Thao tác thực hiện trên bảng dữ liệu
Input :	
Output :	
<ul style="list-style-type: none"> - Table: danh sách thống kê số lượng của các sản phẩm bị trả lại 	
1. Đọc bảng Đơn hàng để thống kê các đơn bị trả (attribute TinhTrangDonHang = “Trả lại”)	Read(DonHang)
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)
3. Trả về các dữ liệu mã sản phẩm và số lần bị trả lại	

A.5.c) usp_ThangMuaSam

usp_ThangMuaSam	Thao tác thực hiện trên bảng dữ liệu
Input :	
Output :	
<ul style="list-style-type: none"> - Table: Trả về dữ liệu tháng nào là có nhiều lượt mua nhất 	
1. Đọc bảng Đơn hàng để thống kê qua attribute NgayBan	Read(DonHang)
2. Trả về table gồm 12 dòng biểu trưng 12 tháng và số đơn mua mỗi tháng	

A.5.d) usp_SoLuongDaBanVaKhachMua

usp_SoLuongDaBanVaKhachMua	Thao tác thực hiện trên bảng dữ liệu
Input : <ul style="list-style-type: none">- @Ngày DATE: Ngày cần thống kê.- @IDSanPham VARCHAR(15): Sản phẩm cần thống kê	
Output : <ul style="list-style-type: none">- @SoLuongKhachMua INT: Tổng số khách mua sản phẩm đó trong ngày.- @SoLuongBan INT: Tổng số lượng sản phẩm bán ra trong ngày.	
1. Đọc bảng Đơn hàng để lấy số lượng khách mua nối với bảng sản phẩm đơn hàng để lấy số lượng bán ra	Read(DonHang) Read(SanPhamDonHang)
2. Trả về tổng số lượng sản phẩm được bán và tổng khách mua sản phẩm đó	

B) Các tình huống xảy ra xung đột

B.1) Lost Update

Liệt kê những cặp stored procedure chính nếu cùng thực hiện đồng thời :

Tên procedure 1	Tên procedure 2	Mô tả sự xung đột
usp_CapNhatSL_HT	usp_CapNhatSL_HT	<ul style="list-style-type: none">- Lúc đầu sản phẩm A có số lượng là 10- Step 1: Transaction 1 sử dụng usp_CapNhatSL_HT để cập nhật số lượng hàng tồn sản phẩm A. $10 - 5 = 5$- Step 2: Transaction 2 sử dụng usp_CapNhatSL_HT khi T1 chưa commit, vẫn lấy giá trị gốc của hàng tồn là 10, đi trừ cho số lượng bán được là 3 $10 - 3 = 7$- Step 3: T1 commit, cập nhật SL_HT = 5- Step 4: T2 commit, cập nhật SL_HT = 7 <p>=> Sai, đáng ra phải là $10 - 5 - 3 = 2$</p>

Tên procedure 1	Tên procedure 2	Mô tả sự xung đột
usp_DieuChinhPhanHangKhachHang	usp_TraHang	<ul style="list-style-type: none">- Lúc đầu Khách hàng A có tiêu dùng là 14 triệu của năm đó.- Step 1: Transaction 1 sử dụng usp_DieuChinhPhanHangKhachHang và đọc được giá trị của tiêu dùng năm đó của Khách A là 16 triệu- Step 2: Tiêu dùng của khách hàng giảm xuống khi mà khách đó mới usp_TraHang 1 đơn hàng 2 triệu. Tiêu dùng giờ là 14 triệu.

		<ul style="list-style-type: none"> - Step 3: Transaction 2 sử dụng usp_DieuChinhPhanHangKhachHang và đọc được giá trị của tiêu dùng năm đó của Khách A là 14 triệu - Step 3: T2 commit, cập nhật hạng khách đó là hạng Bạc. - Step 4: T1 commit, cập nhật hạng khách đó là hạng Vàng. <p>=> Sai, đáng ra phải là hạng Bạc do tiêu dùng chỉ có 14 triệu</p>
--	--	--

Tên procedure 1	Tên procedure 2	Mô tả sự xung đột
usp_XuLyDonHang	Usp_UpdateSanPhamDonHang	<ul style="list-style-type: none"> - Step 1: Transaction 1 usp_XuLyDonHang tính ra được tổng giá trị đơn nhưng do lí do gì nên bị delay chưa commit được. - Step 2: Transaction 2 usp_UpdateSanPhamDonHang thay đổi số lượng của 1 sản phẩm trong đơn hàng trên và commit - Step 3: Transaction 3 usp_XulyDonHang tính ra được tổng giá trị mới. - Step 4: Transaction 3 commit. Tổng đơn hàng được tính theo T3. - Step 5: Transaction 1 commit. Tổng đơn hàng được tính theo T1. <p>=> Sai. Giá trị đơn hàng phải là giá trị mà T3 tính ra nhưng bị T1 lấn át lên làm mất.</p>

B.2) Phantom Read

Tên procedure 1	Tên procedure 2	Mô tả sự xung đột
usp_GuiPhieuMuaHangSinhNhat	usp_ThemKhachHang	<ul style="list-style-type: none"> - Step 1: T1 truy vấn danh sách khách hàng có ngày sinh nhật trong tháng (giả sử ban đầu có 3 khách hàng). - Step 2: T2 thêm một khách hàng mới vào cơ sở dữ liệu có ngày sinh nhật trong tháng. - Step 3: T1 A gửi thông báo phiếu tặng cho danh sách ban đầu (3 khách hàng), bỏ sót khách hàng mới. - Kết quả: Danh sách khách hàng trong giao dịch A không phản

		ảnh chính xác dữ liệu hiện tại trong cơ sở dữ liệu.
--	--	---

Tên procedure 1	Tên procedure 2	Mô tả sự xung đột
usp_CapNhatPhanHangDauThang	usp_QuanLyTaiKhoanKhachHang	<ul style="list-style-type: none"> - Step 1: T1 đọc dữ liệu danh sách khách hàng có trong đơn hàng. - Step 2: T2 tạo khách hàng mới và insert 1 dòng đơn hàng của khách hàng đó mua - Step 3: T1 tiếp tục tính toán tổng giá trị hóa đơn theo kết quả trước khi xóa sản phẩm. => Kết quả: Kết quả phép tính bị sai do số phần tử trong tập SanPhamDonHang của đơn hàng đó bị thay đổi.

Tên procedure 1	Tên procedure 2	Mô tả sự xung đột
usp_KhachHangDoanhThu	usp_TraLaiDonHang	<ul style="list-style-type: none"> - Step 1: T1 đọc dữ liệu các sản phẩm có thuộc tính TinhTrangDonHang < “trả hàng” để thống kê tổng doanh thu - Step 2: T2 xử lý trả hàng cho 1 đơn hàng vô tình làm mất đi 1 đơn hàng đang thống kê trong ngày - Step 3: T1 vẫn tiếp tục thống kê các đơn trả hàng mà không biết sự tồn tại của đơn bị trả mới => Kết quả: Hai phần thống kê của T1 xảy ra sự không đồng nhất dữ liệu.

B.3) Unrepeatable Read

Tên procedure 1	Tên procedure 2	Mô tả sự xung đột
usp_DieuChinhPhanHangKhachHang	usp_TaoHoaDon	<ul style="list-style-type: none"> - Step 1: T1 thực hiện usp_TaoHoaDon bình thường và mới add 1 dòng dữ liệu về 1 sản phẩm có member sale. Khi đó thực

		<p>hiện việc check kết quả hạng trả ra hạng “Bạc”</p> <ul style="list-style-type: none"> - Step 2: T2 thực hiện usp_DieuChinhPhanHangKhachHang và khách hàng đó được tăng hạng thành “Vàng”. - Step 3: T1 xét đến dòng chứa sản phẩm khác và sản phẩm đó vẫn có áp dụng member sale. - Step 4: T1 xét hạng khách hàng. Do việc tăng hạng tại step 2 làm kết quả trả ra của hạng khách hàng là “Vàng”, gây không nhất quán dữ liệu.
--	--	---

Tên procedure 1	Tên procedure 2	Mô tả sự xung đột
usp_ThongKeTraHang	usp_UpdateSanPhamDonHang	<ul style="list-style-type: none"> - Step 1: T1 đọc dữ liệu và thống kê số lượng đơn hàng được trả lại - Step 2: T2 thay đổi số lượng của 1 sản phẩm trong đơn hàng trên. - Step 3: T1 thống kê khách hàng và đơn hàng lần nữa. Kết quả số lượng đã thay đổi => Kết quả: 2 lần thống kê xuất hiện 2 kết quả khác nhau làm kết quả không nhất quán.

Tên procedure 1	Tên procedure 2	Mô tả sự xung đột
usp_TimMonTheoDanhMuc	usp_XoaSanPham	<ul style="list-style-type: none"> - Step 1: T1 tìm món với danh mục “Đồ ăn” Trả về: “Sushi”, “Cơm tấm sườn bì chả 2 trứng” - Step 2: T2 xóa sản phẩm “Sushi” trong bảng khách hàng. - Step 3: T1 tiếp tục tìm lại danh mục “Đồ ăn” Trả về: “Cơm tấm sườn bì chả 2 trứng” => Gây mất đồng bộ dữ liệu

B.4) Dirty Read

Với hệ quản trị này thì việc có dirty read là không thể.