**d**

**d**

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A blue and white logo

Description automatically generated**

**BÁO CÁO ĐỒ ÁN THỰC HÀNH**

**MÔN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**GVHD: Trần Nguyễn Hoài Đức**

**Hồ Thị Hoàng Vy**

**Thành phố Hồ Chí Minh, tháng 12 năm 2023**

# **THÔNG TIN NHÓM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ tên** | **Công việc** | **% Hoàn thành** |
| **1** | 21127157 | Dương Phước Sang | * Xác định chức năng và thiết kế giao diện cho Dùng chung + Nha sĩ. * Xác định tình huống tranh chấp. | 100% |
| **2** | 21127235 | Nguyễn Xuân Quỳnh Chi | * Xác định chức năng và thiết kế giao diện cho Nhân viên + Quản trị viên. * Xác định tình huống tranh chấp. | 100% |
| **3** | 21127085 | Phan Trung Kiên | * Xác định chức năng và thiết kế giao diện cho Khách hàng. * Xác định tình huống tranh chấp. | 100% |

# CÀI ĐẶT TÌNH HUỐNG TRANH CHẤP

# Sinh viên thực hiện:…..

## Tình huống 1: Quản trị viên sử dụng hệ thống để cập nhật thuốc A (VD: số lượng thuốc A từ 10 lên 15) thì lúc này Nha sĩ sẽ xem được số lượng thuốc là 15. Tuy nhiên trong quá trình cập nhật của quản trị viên bị lỗi => rollback => nha sĩ đọc sai dữ liệu thuốc để kê cho bệnh nhân.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ERR01: *Dirty read***  T1 (User = Nha sĩ): thực hiện cập nhật thuốc.  T2 (User = Quản trị viên): thực hiện kê thuốc. | | | | |
| **sp\_CapNhatThuoc** | **Khóa** | **sp\_KeThuoc** | **Khóa** |
| ***Input:*** *…..*  ***Output***: ……. | ***Input***:  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Cập nhật thông tin thuốc  Update [dbo].[Thuoc] set [SoLuongTonKho] = @SoLuongTonKho  where [MaThuoc] = @MaThuoc and [TenT] = @TenT | **XL([dbo].[Thuoc)**  //Xin khoá ghi trên bảng Thuoc. |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Xem thông tin thuốc.  select [SoLuong], [MaHS]  from [dbo].[HS\_T] where [MaThuoc] = @MaThuoc  select [TenT], [ChiDinh], [NgHetHan] from [dbo].[Thuoc] where [MaThuoc] = @MaThuoc | //Không cần xin khoá |
|  |  | COMMIT |  |
| if @@ERROR <> null  begin  raiserror(N'Cập nhật không thành công', 16, 1) rollback  return  end |  |  |  |
| COMMIT |  |  |  |

## 

## Tình huống 2: Một khách hàng đang truy cập vào hệ thống đặt lịch hẹn để chọn lịch hẹn với nha sĩ B, cùng lúc đó phía nha sĩ B cập nhật lịch cá nhân của mình vào lịch được chọn ở trên => gây ra lỗi khi khách hàng xác nhận lịch hẹn.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ERR01: *Lost Update***  T1 (User = Khách hàng): thực hiện đặt lịch hẹn với nha sĩ B qua hệ thống.  T2 (User = Nha sĩ): thực hiện cập nhật lịch bận cá nhân. | | | | |
| **sp\_KhachHangDatLichHen** | **Khóa** | **sp\_CapNhatLichCaNhan** | **Khóa** |
| ***Input:*** *…..*  ***Output***: ……. | ***Input***:  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin  IF NOT EXISTS (SELECT \* FROM KhachHang WHERE MaKH = @MaKH)  BEGIN  ROLLBACK TRAN  RETURN 1  END | **R(KhachHang)**  **Xin khoá đọc trên bảng KhachHang, với điều kiện tồn tại mã khách hàng** |  |  |
| B2: Kiểm tra thông tin  IF NOT EXISTS (SELECT \* FROM NhaSi WHERE MaNS = @MaNS)  BEGIN  ROLLBACK TRAN  RETURN 1  END | **R(NhaSi)**  **Xin khoá đọc trên bảng NhaSi, với điều kiện tồn tại mã nha sĩ** |  |  |
| B3: Kiểm tra thông tin  IF EXISTS (SELECT \* FROM LichLamViec WHERE MaNS = @MaNS AND LichBan = @ThoiGian)  BEGIN  ROLLBACK TRAN  RETURN 1  END | **R(LichLamViec)**  **Xin khoá đọc trên bảng LichLamViec, với điều kiện lịch bận không trùng với thời gian đặt hẹn** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Nha sĩ cập nhật lịch cá nhân  UPDATE [dbo].[LichLamViec] SET [LichBan] = @ThoiGian  WHERE [MaNS] = @MaNS | **XL([dbo].[LichLamViec)**  //Xin khoá ghi trên bảng LichLamViec. |
| B4: Cập nhật lịch hẹn  UPDATE LichHen SET MaKH = @MaKH\_2  WHERE MaNS = @MaNS, ThoiGian = @ThoiGian | **XL(LichHen)**  **Xin khoá ghi trên bảng LichHen, cập nhật mã khách hàng** |  |  |
|  |  | COMMIT |  |
| COMMIT |  |  |  |

## 

## Tình huống 3: Khách hàng A một lịch hẹn X, thì trong hệ thống lịch hẹn đã được giữ chỗ. Khách hàng B khi xem hệ thống lịch thì không còn thấy lịch X, nhưng quá trình xác nhận lịch của A có lỗi hệ thống => rollback => Khách hàng B không xem được lịch hẹn X.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ERR01: *Dirty Read***  T1 (User = Khách hàng): thực hiện đặt lịch hẹn X qua hệ thống.  T2 (User = Khách hàng): thực hiện đặt lịch hẹn X qua hệ thống. | | | | |
| **sp\_KhachHangDatLichHen** | **Khóa** | **sp\_KhachHangDatLichHen** | **Khóa** |
| ***Input:*** *…..*  ***Output***: ……. | ***Input***:  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin  IF NOT EXISTS (SELECT \* FROM KhachHang WHERE MaKH = @MaKH)  BEGIN  ROLLBACK TRAN  RETURN 1  END | **R(KhachHang)**  **Xin khoá đọc trên bảng KhachHang, với điều kiện tồn tại mã khách hàng** |  |  |
| B2: Kiểm tra thông tin  IF NOT EXISTS (SELECT \* FROM NhaSi WHERE MaNS = @MaNS)  BEGIN  ROLLBACK TRAN  RETURN 1  END | **R(NhaSi)**  **Xin khoá đọc trên bảng NhaSi, với điều kiện tồn tại mã nha sĩ** |  |  |
| B3: Kiểm tra thông tin  IF EXISTS (SELECT \* FROM LichLamViec WHERE MaNS = @MaNS AND LichBan = @ThoiGian)  BEGIN  ROLLBACK TRAN  RETURN 1  END | **R(LichLamViec)**  **Xin khoá đọc trên bảng LichLamViec, với điều kiện lịch bận không trùng với thời gian đặt hẹn** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin  IF NOT EXISTS (SELECT \* FROM KhachHang WHERE MaKH\_2 = @MaKH)  BEGIN  ROLLBACK TRAN  RETURN 1  END | **R(KhachHang)**  **Xin khoá đọc trên bảng KhachHang, với điều kiện tồn tại mã khách hàng** |
|  |  | B2: Kiểm tra thông tin  IF NOT EXISTS (SELECT \* FROM NhaSi WHERE MaNS = @MaNS)  BEGIN  ROLLBACK TRAN  RETURN 1  END | **R(NhaSi)**  **Xin khoá đọc trên bảng NhaSi, với điều kiện tồn tại mã nha sĩ** |
|  |  | B3: Kiểm tra thông tin  IF EXISTS (SELECT \* FROM LichLamViec WHERE MaNS = @MaNS AND LichBan = @ThoiGian)  BEGIN  ROLLBACK TRAN  RETURN 1  END | **R(LichLamViec)**  **Xin khoá đọc trên bảng LichLamViec, với điều kiện lịch bận không trùng với thời gian đặt hẹn** |
|  |  | WAITFOR DELAY '00:00:10' |  |
| B4: Cập nhật lịch hẹn  UPDATE LichHen SET MaKH = @MaKH WHERE MaNS = @MaNS, ThoiGian = @ThoiGian | **X(LichHen)**  **Xin khoá ghi trên bảng LichHen, cập nhật mã khách hàng** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | B4: Cập nhật lịch hẹn  UPDATE LichHen SET MaKH = @MaKH\_2 WHERE MaNS = @MaNS, ThoiGian = @ThoiGian | **X(LichHen)**  **Xin khoá ghi trên bảng LichHen, cập nhật mã khách hàng** |
|  |  | COMMIT |  |
| COMMIT |  |  |  |

## 

## Tình huống 4: Một quản trị viên đang cập nhật lại giá thuốc A, thì có một nhân viên tính tiền và xuất hóa đơn cho khách hàng, sau đó việc chỉnh giá của quản trị viên bị lỗi rollback.

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Unrepeatable Read***  T1 (User = Nhân viên): thực hiện thanh toán hoá đơn cho khách hàng A.  T2 (User = Nha sĩ): thực hiện cập nhật chi phí khám cá nhân. | | | |
| **sp\_LapHoaDon** | **Khóa** | **sp\_CapNhatChiPhiKham** | **Khóa** |
| ***Input:*** *…..*  ***Output***: ……. | ***Input***:  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin  IF NOT EXISTS (SELECT \* FROM HoSoKham WHERE MaHS = @MaHS)  BEGIN  ROLLBACK TRAN  RETURN 1  END | **R(HoSoKham)**  **Xin khoá đọc trên bảng HoSoKham, với điều kiện tồn tại mã hồ sơ** |  |  |
| B2: Kiểm tra thuốc được kê đơn còn trong kho  IF NOT EXISTS (SELECT \* FROM HS\_T, Thuoc T WHERE HS\_T.MaThuoc = T.MaThuoc AND HS\_T.SoLuong <= T.SoLuongTonKho)  BEGIN  ROLLBACK TRAN  RETURN 1  END | **R(HS\_T), R(Thuoc)**  **Xin khoá đọc trên bảng HS\_T và Thuoc với điều kiện tồn tại mã thuốc và số lượng thuốc được kê <= số lượng tồn kho** |  |  |
| B2: Tính tổng tiền cần thanh toán  DECLARE @PhiDV INT, @ChiPhiKham INT, @TienThuoc INT, @TongTien INT  SELECT @PhiDV = SUM(DV.ChiPhi) FROM HS\_DV, DichVu DV, WHERE HS\_DV.MaHS = @MaHS AND HS\_DV.MaDV = DV.MaDV  SELECT @ChiPhiKham = NS.ChiPhiKham FROM HoSoKham HSK, NhaSi NS WHERE HSK.MaHS = @MaHS AND HSK.NguoiKham = NS.MaNS  SELECT @TienThuoc = SUM(T.GiaTien\*HS\_T.SoLuong)  FROM HS\_T, Thuoc T WHERE HS\_T.MaHS = @MaHS AND HS\_T.MaThuoc = T.MaThuoc  @TongTien = @PhiDV + @ChiPhiKham + @TienThuoc | **R(HS\_DV), R(DichVu). R(NhaSi)**  **Xin khoá đọc trên bảng HS\_DV, DichVu và NhaSi tính tổng tiền thanh toán bao gồm phí dịch vụ, phí khám và tiền thuốc** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin  IF NOT EXISTS (SELECT \* FROM NhaSi WHERE MaNS = @MaNS AND ChiPhiKham < > NULL)  BEGIN  ROLLBACK TRAN  RETURN 1  END | **R(Thuoc)**  **Xin khoá đọc trên bảng NhaSi, với điều kiện chi phí khám không được để trống.** |
|  |  | B2: Cập nhật chi phí khám  UPDATE [dbo].[NhaSi] SET [ChiPhiKham[= @ChiPhiKham WHERE [MaNS]= @MaNS | **X(Thuoc)**  **Xin khoá ghi trên bảng Thuoc, cập nhật giá thuốc mới** |
|  |  | COMMIT |  |
| COMMIT |  |  |  |

## 