**BỘ MÔN HỆ THỐNG THÔNG TIN – KHOA CÔNG NGHỆ THÔNG TIN**

**ĐẠI HỌC KHOA HỌC TỰ NHIÊN THÀNH PHỐ HỒ CHÍ MINH, ĐẠI HỌC QUỐC GIA TP HCM**

Sinh viên thực hiện: 19127366 – Long Mỹ Du (Nhóm trưởng)

19127304 – Trần Khải Trúc

19127649 – Tô Thanh Tuấn

GV phụ trách: Hồ Thị Hoàng Vy

Tiết Gia Hồng

Đồ án môn học: DA-01  - HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

HỌC KỲ I – NĂM HỌC 2021-2022

**môn hệ quản trị cơ sở dữ liệu**

**BẢNG THÔNG TIN CHI TIẾT NHÓM**

|  |  |  |
| --- | --- | --- |
| **Mã nhóm:** | Nhóm 10 | |
| **Số lượng:** | **3** | |
| **MSSV** | **Họ tên** | **Email** |
| 19127366 | Long Mỹ Du | lmdu19@clc.fitus.edu.vn |
| 19127304 | Trần Khải Trúc | tktruc19@clc.fitus.edu.vn |
| 19127649 | Tô Thanh Tuấn | tttuan19@clc.fitus.edu.vn |

|  |  |  |  |
| --- | --- | --- | --- |
| **Bảng phân công & đánh giá hoàn thành công việc** | | | |
| **Công việc thực hiện** | **Người thực hiện** | **Mức độ hoàn thành** | **Đánh giá của nhóm** |
| * Thiết kế cơ sở dữ liệu * Phân quyền cho admin * Viết trigger cho tài khoản * Tạo các procedure chức năng * Thiết kế giao diện hệ thống GUI * Viết báo cáo * Tìm và giải quyết tình huống tranh chấp: Dirty Read, Unrepeatable Read, Phantom Read. * Cài đặt hệ thống: Front-end | Long Mỹ Du | 100% | 10/10 |
| * Thiết kế cơ sở dữ liệu * Phân quyền cho đối tác * Viết trigger cho ràng buộc loại tài khoản chi phí đơn hàng * Tạo các procedure chức năng * Thiết kế giao diện hệ thống GUI * Viết báo cáo * Tìm và giải quyết tình huống tranh chấp: Lost Update, Dirty Read. * Cài đặt hệ thống: Back-end | Trần Khải Trúc | 100% | 10/10 |
| * Thiết kế cơ sở dữ liệu * Phân quyền cho khách hàng, nhân viên, tài xế * Viết trigger cho đơn hàng * Tạo các procedure chức năng * Thiết kế giao diện hệ thống GUI * Viết báo cáo * Tìm và giải quyết tình huống tranh chấp: Unrepeatable Read, Phantom Read. * Cài đặt hệ thống: Back-end. | Tô Thanh Tuấn | 100% | 10/10 |

**YÊU CẦU ĐỒ ÁN- BÀI TẬP**

|  |  |
| --- | --- |
| **Loại bài tập** | **Lý thuyết 🗹 Thực hành 🗹 Đồ án Bài tập** |
| **Ngày bắt đầu** | **20/10/2021** |
| **Ngày kết thúc** | **22/11/2021** |

# **Yêu cầu của đồ án**

1. Thiết kế cơ sở dữ liệu
2. Phân quyền người dùng
3. Xác định tình huống tranh chấp
4. Hướng giải quyết
5. Thiết kế giao diện hệ thống (GUI)
6. Cài đặt hệ thống

# **Kết quả**

Mục lục

[A. Yêu cầu của đồ án 2](#_Toc88388034)

[B. Kết quả 3](#_Toc88388035)

[I. Đặc tả cơ sở dữ liệu 4](#_Toc88388036)

[II. Mô hình ER 6](#_Toc88388037)

[III. Ràng buộc dữ liệu 7](#_Toc88388038)

[**1.** **Tài khoản người dùng** 7](#_Toc88388039)

[**2.** **Sản phẩm thuộc đơn hàng** 7](#_Toc88388040)

[**3.** **Loại tài khoản** 8](#_Toc88388041)

[**4.** **Phí sản phẩm** 8](#_Toc88388042)

[IV. Phân quyền người dùng 9](#_Toc88388043)

[V. Tranh chấp đồng thời 11](#_Toc88388044)

[**1.** **Lost Update** 11](#_Toc88388045)

[Tình huống 1 11](#_Toc88388046)

[Tình huống 2 11](#_Toc88388047)

[**2.** **Dirty Read** 17](#_Toc88388048)

[Tình huống 1 18](#_Toc88388049)

[Tình huống 2 18](#_Toc88388050)

[**3.** **Unrepeatable Read** 21](#_Toc88388053)

[Tình huống 1 22](#_Toc88388054)

[Tình huống 2 22](#_Toc88388055)

[**4.** **Phantom Read** 28](#_Toc88388057)

[Tình huống 1 28](#_Toc88388058)

[Tình huống 2 30](#_Toc88388059)

1. **Đặc tả cơ sở dữ liệu**

**Hệ thống đặt và chuyển hàng online**

1. **TAIKHOAN (TenTaiKhoan, MatKhau, PhanLoai, TinhTrangKhoa)**

Tân từ: Mỗi tài khoản gồm một tên tài khoản duy nhất, có một mật khẩu, thuộc một phân loại người dùng hệ thống. TinhTrangKhoa cho biết tài khoản bị khóa hay không.

1. **DOITAC (MaDoiTac, TenDoiTac, NguoiDaiDien, ThanhPho, Quan, SoChiNhanh, SoDonHangMoiNgay, LoaiHang, DiaChiKinhDoanh, SoDT, Email, TenTaiKhoan)**

Tân từ: Mỗi đối tác có một mã số duy nhất, một tên, một người đại diện, có một địa chỉ kinh doanh thuộc một quận của thành phố, kinh doanh một loại hàng, có một số điện thoại, một email và một tài khoản của hệ thống. SoChiNhanh cho biết số lượng chi nhánh của đối tác đó và SoDonHangMoiNgay là số lượng đơn hàng đối tác cần vận chuyển trong mỗi ngày.

1. **HOPDONG (MaHopDong, MaSoThue, ThanhToanPhiKichHoat, PhiHoaHong, ThoiGianHieuLuc, SoChiNhanh, TinhTrangDuyet, MaDoiTac)**

Tân từ: Mỗi hợp đồng có một mã hợp đồng duy nhất, có một mã số thuế thuộc một đối tác, có thời gian hiệu lực. Phí hoa hồng phải nộp được tính bằng doanh số bán x 10%. Thuộc tính ThanhToanPhiKichHoat cho biết đối tác đã thanh toán phí kích hoạt của hợp đồng hay chưa để xếp vào hàng chờ duyệt. TinhTrangDuyet cho biết nhân viên đã duyệt hợp đồng này hay chưa. SoChiNhanh là số lượng chi nhánh đăng ký trong hợp đồng.

1. **CHINHANH (MaChiNhanh, DiaChi, MaHopDong)**

Tân từ: Mỗi chi nhánh có một mã số duy nhất, có một địa chỉ và thuộc về một hợp đồng.

1. **SANPHAM (MaSP, TenSP, Gia, SoLuongTon, MaChiNhanh)**

Tân từ: Mỗi sản phẩm có một mã sản phẩm duy nhất, có một tên của sản phẩm, có một giá và được cung cấp bởi một chi nhánh. Có số lượng tồn của sản phẩm được cập nhật sau khi bán.

1. **KHACHHANG (MaKH, HoTen, SoDT, DiaChi, Email, TenTaiKhoan)**

Tân từ: Mỗi khách hàng có một mã số duy nhất, có một họ tên, một số điện thoại, một địa chỉ, một email và có một tài khoản của hệ thống.

1. **TAIXE (MaTaiXe, HoTen, CMND, SoDT, DiaChi, BienSoXe, KhuVucHoatDong, Email, TaiKhoanNganHang, TenTaiKhoan)**

Tân từ: Mỗi tài xế có một mã số duy nhất, có một họ tên, một chứng minh nhân dân, một số điện thoại, một địa chỉ, một biển số xe, có một khu vực hoạt động, một email, một tài khoản ngân hàng, và có một tài khoản của hệ thống.

1. **DONHANG (MaDonHang, NgayLap, HinhThucThanhToan, DiaChiGiaoHang, PhiSP, PhiVC, TinhTrangVanChuyen, MaKH, MaChiNhanh, MaTaiXe)**

Tân từ: Mỗi đơn hàng có một mã đơn hàng duy nhất, có ngày lập đơn hàng, một địa chỉ giao đơn hàng, có một hình thức để thanh toán đơn hàng. PhiSP và PhiVC là tiền mua sản phẩm và chi phí vận chuyển. TinhTrangVanChuyen được tài xế cập nhật để khách hàng theo dõi. Đơn hàng của một khách hàng được cung cấp bởi một chi nhánh và được một tài xế vận chuyển.

1. **CHITIETDONHANG (MaSP, MaDonHang, SoLuong, Gia)**

Tân từ: Mỗi dòng của chi tiết đơn hàng thể hiện số lượng và giá của một sản phẩm thuộc một đơn hàng.

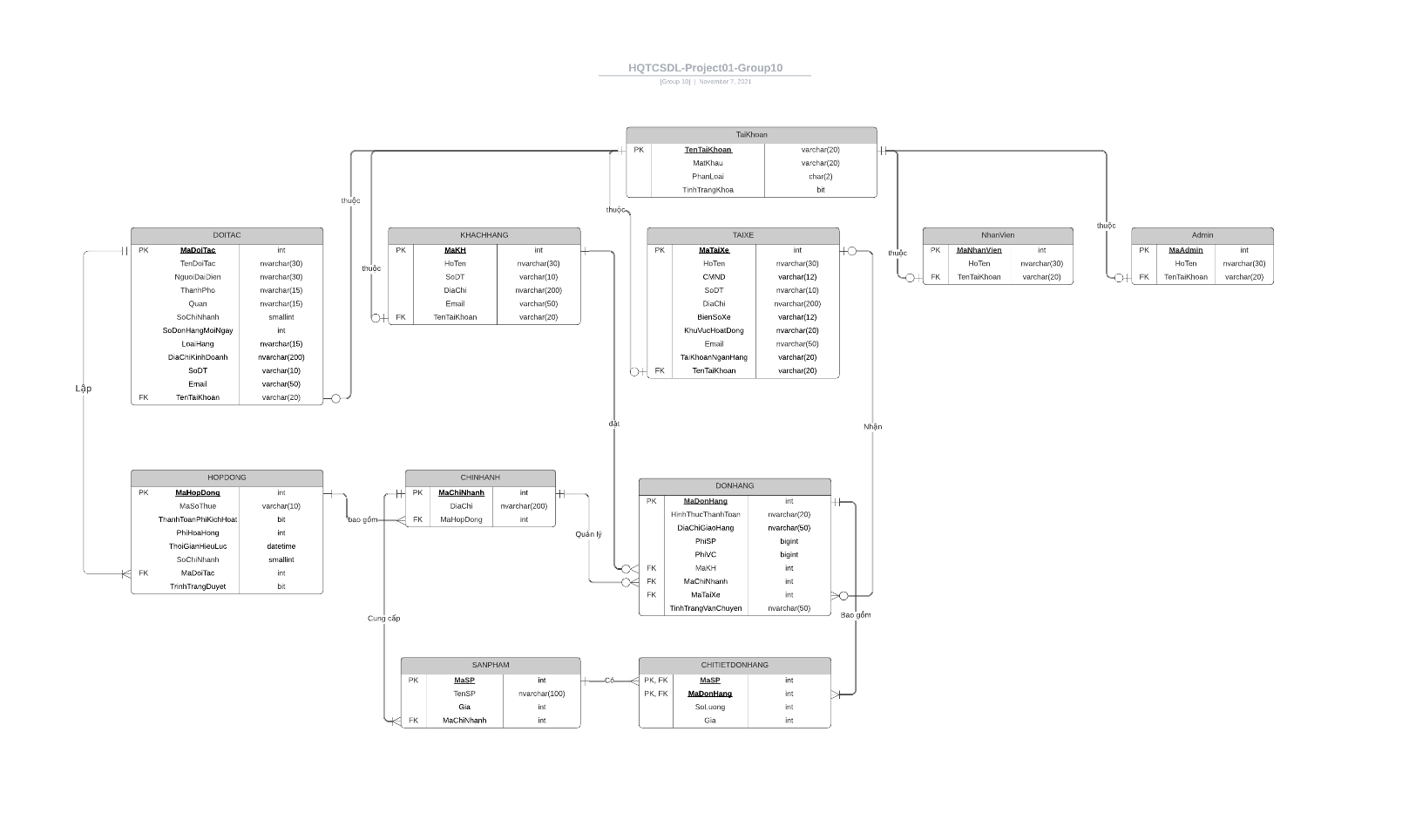
1. **NHANVIEN (MaNhanVien, HoTen, TenTaiKhoan)**

Tân từ: Mỗi nhân viên có một mã nhân viên duy nhất, một họ tên và có một tài khoản thuộc hệ thống.

1. **ADMIN (MaAdmin, HoTen, TenTaiKhoan)**

Tân từ: Mỗi admin có một mã số duy nhất, có một họ tên và một tài khoản của hệ thống.

1. **Mô hình ER**



1. **Ràng buộc dữ liệu**
2. **Tài khoản người dùng**
3. **Ràng buộc:** Mỗi tài khoản chỉ được thuộc về một người dùng

**Bảng tầm ảnh hưởng:**

|  |  |  |  |
| --- | --- | --- | --- |
| Bảng | Thêm | Xóa | Sửa |
| Doitac | + | - | +(TenTaiKhoan) |
| TaiXe | + | - | +(TenTaiKhoan) |
| KhachHang | + | - | +(TenTaiKhoan) |
| NhanVien | + | - | +(TenTaiKhoan) |
| Admin | + | - | +(TenTaiKhoan) |

1. **Ràng buộc**: Mỗi tài khoản phải cùng loại với người dùng sử dụng tài khoản đó

**Bảng tầm ảnh hưởng:**

|  |  |  |  |
| --- | --- | --- | --- |
| Bảng | Thêm | Xóa | Sửa |
| TaiKhoan | - | - | +(PhanLoai) |
| Doitac | + | - | +(TenTaiKhoan) |
| TaiXe | + | - | +(TenTaiKhoan) |
| KhachHang | + | - | +(TenTaiKhoan) |
| NhanVien | + | - | +(TenTaiKhoan) |
| Admin | + | - | +(TenTaiKhoan) |

1. **Sản phẩm thuộc đơn hàng**

**Ràng buộc:** Mỗi đơn hàng chỉ được có các sản phẩm thuộc cùng một chi nhánh

**Bảng tầm ảnh hưởng:**

|  |  |  |  |
| --- | --- | --- | --- |
| Bảng | Thêm | Xóa | Sửa |
| DonHang | - | - | +(MaChiNhanh) |
| ChiTietDonHang | + | - | +(MaDonHang, MaSP) |

1. **Loại tài khoản**

**Mô tả:**

Phân loại tài khoản trong bảng TaiKhoan bao gồm: DT (đối tác), KH (khách hàng), TX (tài xế), NV (nhân viên), AD (admin)

**Nội dung:**

Thêm Constraint CK\_TaiKhoan\_PhanLoai trên bảng TaiKhoan

1. **Phí sản phẩm**

**Ràng buộc:**

Phí sản phẩm (PhiSP) trong bảng đơn hàng (DonHang) bằng tổng giá trị các sản phẩm thuộc đơn hàng đó trong bảng chi tiết đơn hàng (ChiTietDonHang)

**Bảng tầm ảnh hưởng:**

|  |  |  |  |
| --- | --- | --- | --- |
| Bảng | Thêm | Xóa | Sửa |
| ChiTietDonHang | + | + | +(SoLuong,Gia) |

1. **Phân quyền người dùng**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vai trò  Thực thể | **Đối tác** | **Khách Hàng** | **Tài Xế** | **Nhân Viên** | **Admin** |
| **TAIKHOAN** | **Xem** (TenTaiKhoan, TinhTrangKhoa)  **Sửa** (TenTaiKhoan, MatKhau) | **Xem** (TenTaiKhoan, TinhTrangKhoa)  **Sửa** (TenTaiKhoan, MatKhau) | **Xem** (TenTaiKhoan, TinhTrangKhoa)  **Sửa** (TenTaiKhoan, MatKhau) | **Xem** (TenTaiKhoan, TinhTrangKhoa)  **Sửa** (TenTaiKhoan, MatKhau) | **Xem**  **Thêm**  **Xóa**  **Sửa**  **Cấp quyền** |
| **DOITAC** | **Xem**  **Sửa** | **Xem** (TenDoiTac, Thanhpho, Quan, LoaiHang, SoDT, DiaChiKinhDoanh) | x | Xem | x |
| **HOPDONG** | **Xem**  **Thêm** | x | x | **Xem**  **Sửa** (PhiHoaHong, ThoiGianHieuLuc TinhTrangDuyet) | x |
| **CHINHANH** | **Xem**  **Thêm**  **Xóa**  **Sửa** (DiaChi) | **Xem** (DiaChi) | x | x | x |
| **KHACHHANG** | x | **Xem**  Sửa(Hoten, Sodt, Diachi, Email) | **Xem** (MaKH, HoTen, SoDT) | x | x |
| **DONHANG** | **Xem**  **Sửa** (TinhTrangVanChuyen) | **Xem**  **Thêm**  **Sửa** (HìnhThucThanhToan, DiaChiGiaoHang, MaChiNhanh) | **Xem**  **Sửa** (TinhTrangVanChuyen) | x | x |
| **CHITIETDONHANG** | **Xem** | **Xem**  **Thêm**  **Xóa**  **Sửa** (MaSP, SoLuong) | x | x | x |
| **SANPHAM** | **Xem**  **Thêm**  **Xóa**  **Sửa** | Xem (MaSp, Tensp, Gia, MaChiNhanh) | x | x | x |
| **TAIXE** | x | x | **Xem**  **Sử**a | x | x |
| **NHANVIEN** | x | x | x | x | **Xem**  **Thêm**  **Xóa**  **Sửa**  **Cấp quyền** |
| **ADMIN** | x | x | x | x | **Xem**  **Thêm**  **Xóa**  **Sửa**  **Cấp quyền** |

1. **Tranh chấp đồng thời**
2. **Lost Update**

Tình huống 1

Transaction 1: Bán được 3 sản phẩm ‘100001’ và cập nhật lượng tồn của sản phẩm.

Transaction 2: Thêm vào 5 sản phẩm ‘100001’ và cập nhật lượng tồn của sản phẩm.

|  |  |  |
| --- | --- | --- |
| **Transaction 1** | **Transaction 2** | **Giải thích** |
| Set @soluongton = (Select SoLuongTon from SanPham where MaSP = @masp) |  | T1 lấy số lượng tồn của sản phẩm ‘100001’ |
|  | Set @soluongton = (Select SoLuongTon from SanPham where MaSP = @masp) | T2 lấy số lượng tồn của sản phẩm ‘100001’ |
|  | Set @soluongton =  @soluongton + @soluongthem | T2 thêm vào 5 sản phẩm |
| Set @soluongton =  @soluongton - @soluongban |  | T1 bán được 3 sản phẩm |
|  | Update SanPham  Set SoLuongTon = @soluongton  Where MaSP = @masp | T2 cập nhật số lượng tồn của sản phẩm |
| Update SanPham  Set SoLuongTon = @soluongton  Where MaSP = @masp |  | T1 cập nhật số lượng tồn của sản phẩm |
|  | Commit |  |
| Commit |  |  |

**Giải thích**

Ảnh có chứa bàn

Mô tả được tạo tự độngXem số lượng tồn ban đầu của sản phẩm ‘100001’ là 687.

* Chạy transaction 1 trước và chạy transaction 2 ngay sau đó.
* Ta thấy transaction 2 sẽ hoàn thành trước và kết quả số lượng tồn lúc này là 692 ( = 687 + 5)
* Ảnh có chứa văn bản, máy tính, trong nhà, ảnh chụp màn hình

  Mô tả được tạo tự độngSau đó vài giây, transaction 1 hoàn thành và kết quả số lượng tồn là 684 ( = 687 – 3)
* Tuy nhiên, kết quả đúng phải là 687 + 5 – 3 = 689
* Do transaction 1 không đọc dữ liệu cập nhật từ transaction 2 **🡪 Lost Update**

**Giải quyết tranh chấp**

Thiết lập mức cô lập cho cả 2 transaction là REPEATABLE READ 🡪 Dẫn tới **Deadlock**

|  |  |  |
| --- | --- | --- |
| **Transaction 1** | **Transaction 2** | **Giải thích** |
| Set @soluongton = (Select SoLuongTon from SanPham where MaSP = @masp) |  | T1 xin cấp S-lock trên bảng SANPHAM  SQL: Cấp S-lock trên bảng SANPHAM cho T1 |
|  | Set @soluongton = (Select SoLuongTon from SanPham where MaSP = @masp) | T2 xin cấp S-lock trên bảng SANPHAM  SQL: Cấp S-lock trên bảng SANPHAM cho T2 |
|  | Set @soluongton =  @soluongton + @soluongthem |  |
| Set @soluongton =  @soluongton - @soluongban |  |  |
|  | Update SanPham  Set SoLuongTon = @soluongton  Where MaSP = @masp  **(CHỜ)** | T2 xin cấp X-lock trên bảng SANPHAM  SQL: Không cấp X-lock trên bảng SANPHAM cho T2 vì T1 đang giữ S-lock |
| Update SanPham  Set SoLuongTon = @soluongton  Where MaSP = @masp  **(CHỜ)** |  | T1 xin cấp X-lock trên bảng SANPHAM  SQL: Không cấp X-lock trên bảng SANPHAM cho T1 vì T2 đang giữ S-lock |
|  |  | T1 chờ T2 trả S-lock, T2 chờ T1 trả S-lock **🡪 Deadlock** |

**Giải quyết deadlock**

Thiết lập mức cô lập cho cả 2 transaction là REPEATABLE READvà sử dụng UPDATE LOCK khi đọc.

|  |  |  |
| --- | --- | --- |
| **Transaction 1** | **Transaction 2** | **Giải thích** |
| Set @soluongton = (Select SoLuongTon  from SanPham **with (Updlock)**  where MaSP = @masp) |  | T1 xin cấp U-lock trên bảng SANPHAM  SQL: Cấp U-lock trên bảng SANPHAM cho T1 |
|  | Set @soluongton = (Select SoLuongTon  from SanPham **with (Updlock)**  where MaSP = @masp)  **(CHỜ)** | T2 xin cấp U-lock trên bảng SANPHAM  SQL: Không cấp U-lock cho T2 vì T1 đang giữ U-lock. |
| Set @soluongton =  @soluongton - @soluongban |  |  |
| Update SanPham  Set SoLuongTon = @soluongton  Where MaSP = @masp |  | T1 xin cấp X-lock trên bảng SANPHAM  SQL: Cấp X-lock trên bảng SANPHAM cho T1. |
| commit tran |  | T1 trả khóa X-lock trên bảng SANPHAM. |
|  | Set @soluongton = (Select SoLuongTon  from SanPham **with (Updlock)**  where MaSP = @masp) | T2 xin cấp U-lock trên bảng SANPHAM  SQL: Cấp U-lock trên bảng SANPHAM cho T2 |
|  | Set @soluongton =  @soluongton + @soluongthem |  |
|  | Update SanPham  Set SoLuongTon = @soluongton  Where MaSP = @masp  (chờ) | T2 xin cấp X-lock trên bảng SANPHAM  SQL: Cấp X-lock trên bảng SANPHAM cho T2 |
|  | commit tran |  |

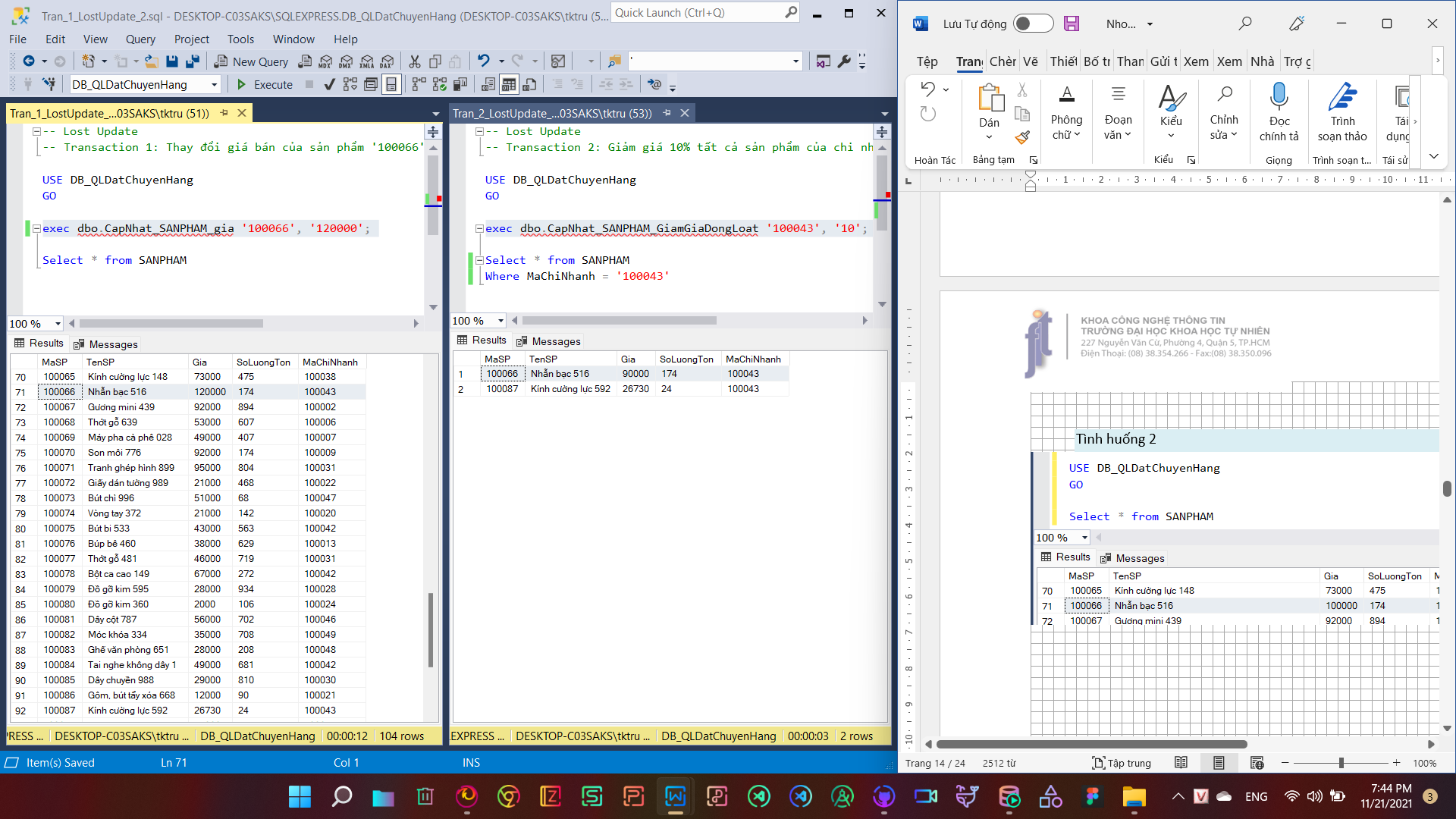
Tình huống 2

Transaction 1: Thay đổi giá bán của sản phẩm ‘100066’ thành 120.000đ.

Transaction 2: Giảm giá 10% tất cả sản phẩm của chi nhánh ‘100043’.

|  |  |  |
| --- | --- | --- |
| **Transaction 1** | **Transaction 2** | **Giải thích** |
| Set @gia = (Select Gia  from SanPham  where MaSP = @masp) |  | T1 lấy giá của sản phẩm ‘100066’ |
|  | Set @gia = (Select Gia  from SanPham  where MaChiNhanh = @macn) | T2 lấy giá của các sản phẩm thuộc chi nhánh ‘100043’ |
|  | Set @gia = @gia \*  (100% - @giamgia%) | T2 giảm giá 10% các sản phẩm của chi nhánh |
| Set @gia = @giamoi |  | T1 đổi giá của sản phẩm ‘100066’ |
|  | Update SanPham  Set Gia = @gia  Where MaChiNhanh = @macn | T2 cập nhật giá của các sản phẩm thuộc chi nhánh |
| Update SanPham  Set Gia = @gia  Where MaSP = @masp |  | T1 cập nhật giá của sản phẩm ‘100066’ |
|  | Commit |  |
| Commit |  |  |

* Ảnh có chứa bàn

  Mô tả được tạo tự độngXem giá ban đầu của sản phẩm ‘100066’ là 100.000đ.
* Chạy transaction 1 trước và chạy transaction 2 ngay sau đó.
* Ta thấy transaction 2 sẽ hoàn thành trước và kết quả giá là 90.000 (giảm giá 10% với 100.000).
* Sau đó vài giây, transaction 1 hoàn thành và kết quả giá là 120.000đ.
* Do cả 2 transaction không đọc dữ liệu cập nhật của nhau **🡪 Lost Update.**

**Giải quyết tranh chấp**

Thiết lập mức cô lập cho cả 2 transaction là REPEATABLE READ 🡪 Dẫn tới **Deadlock**

|  |  |  |
| --- | --- | --- |
| **Transaction 1** | **Transaction 2** | **Giải thích** |
| Set @gia = (Select Gia  from SanPham  where MaSP = @masp) |  | T1 xin cấp S-lock trên bảng SanPham  SQL cấp S-lock cho T1. |
|  | Set @gia = (Select Gia  from SanPham  where MaChiNhanh = @macn) | T2 xin cấp S-lock trên bảng SanPham  SQL cấp S-lock cho T2. |
|  | Set @gia = @gia \*  (100% - @giamgia%) |  |
| Set @gia = @giamoi |  |  |
|  | Update SanPham  Set Gia = @gia  Where MaChiNhanh = @macn  **(CHỜ)** | T2 xin cấp X-lock trên bảng SanPham.  SQL không cấp X-lock cho T2 vì T1 đang giữ S-lock. |
| Update SanPham  Set Gia = @gia  Where MaSP = @masp  **(CHỜ)** |  | T1 xin cấp X-lock trên bảng SanPham.  SQL không cấp X-lock cho T1 vì T đang giữ S-lock. |
|  | | T1 chờ T2 trả S-lock, T2 chờ T1 trả S-lock  **🡪 Deadlock** |

**Giải quyết deadlock**

Thiết lập mức cô lập cho cả 2 transaction là REPEATABLE READvà sử dụng UPDATE LOCK khi đọc.

|  |  |  |
| --- | --- | --- |
| **Transaction 1** | **Transaction 2** | **Giải thích** |
| Set @gia = (Select Gia  from SanPham **with (Updlock)**  where MaSP = @masp) |  | T1 xin cấp U-lock trên bảng SanPham  SQL cấp U-lock trên bảng SanPham cho T1 |
|  | Set @gia = (Select Gia  from SanPham **with (Updlock)**  where MaChiNhanh = @macn)  **(CHỜ)** | T2 xin cấp U-lock trên bảng SanPham  SQL không cấp U-lock cho T2 vì T1 đang giữ U-lock |
| Set @gia = @giamoi |  |  |
| Update SanPham  Set Gia = @gia  Where MaSP = @masp |  | T1 xin cấp X-lock trên bảng SanPham  SQL cấp X-lock trên bảng SanPham cho T1 |
| Commit |  | T1 trả khóa X-lock trên bảng SanPham |
|  | Set @gia = (Select Gia  from SanPham **with (Updlock)**  where MaChiNhanh = @macn) | T2 xin cấp U-lock trên bảng SanPham  SQL cấp U-lock trên bảng SanPham cho T2 |
|  | Set @gia = @gia \*  (100% - @giamgia%) |  |
|  | Update SanPham  Set Gia = @gia  Where MaChiNhanh = @macn | T2 xin cấp X-lock trên bảng SanPham  SQL cấp X-lock trên bảng SanPham cho T2 |
|  | Commit |  |

1. **Dirty Read**

Tình huống 1

Transaction 1: Bán 8 sản phẩm ‘100028’, sau đó bị hủy do lỗi số lượng tồn.

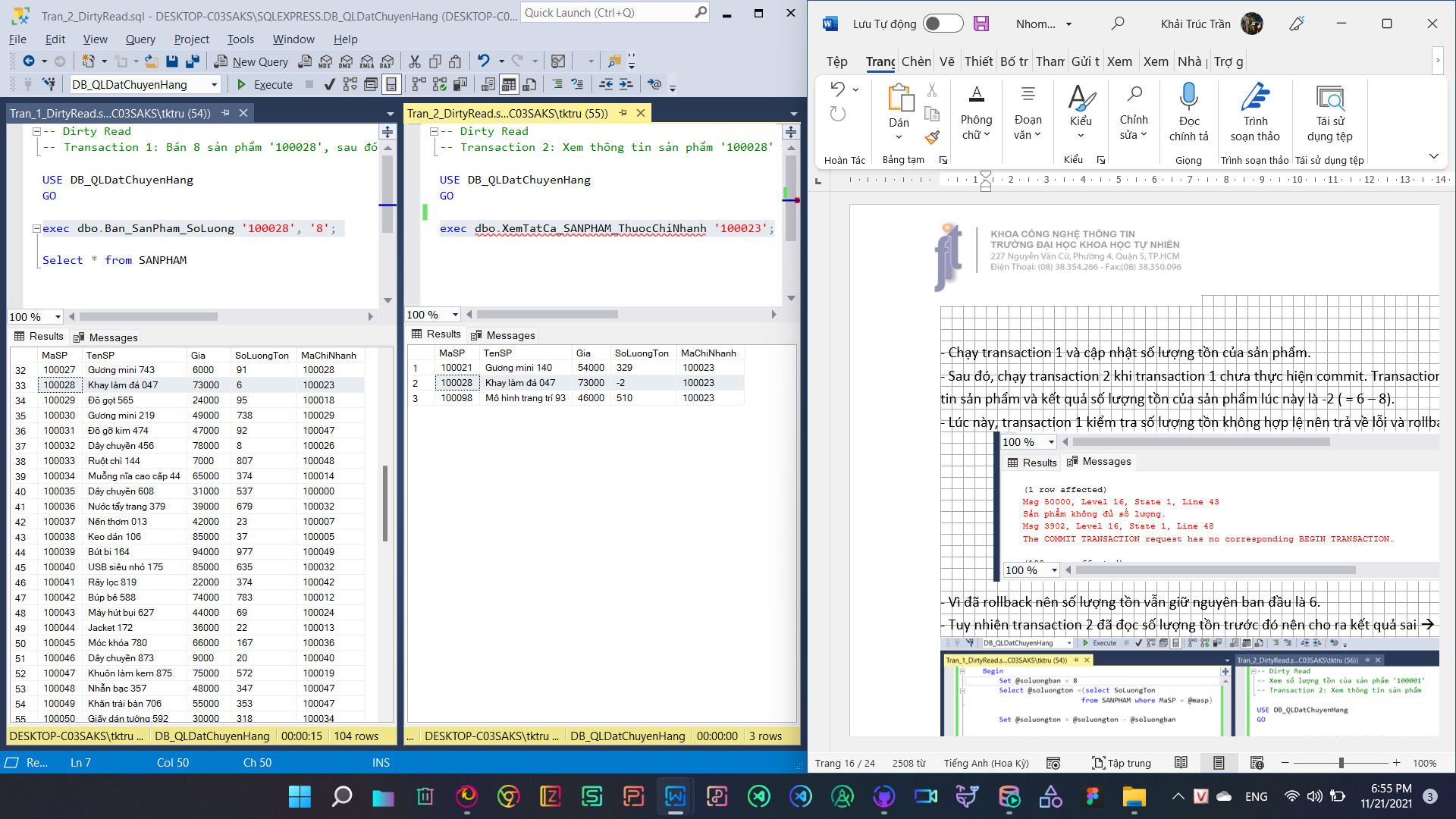
Transaction 2: Xem sản phẩm ‘100028’ thuộc chi nhánh ‘100023’.

|  |  |  |
| --- | --- | --- |
| **Transaction 1** | **Transaction 2** | **Giải thích** |
| Set @soluongton = (Select SoLuongTon from SanPham where MaSP = @masp) |  | T1 lấy số lượng tồn của sản phẩm ‘100028’ |
| Set @soluongton = @soluongton - @soluongban |  | T1 bán 8 sản phẩm ‘100028’ |
| Update SanPham  Set SoLuongTon = @soluongton  Where MaSP = @masp |  | T1 cập nhật số lượng tồn của sản phẩm ‘100028 |
|  | Select \* from SanPham where MaSP = @masp | T2 xem thông tin sản phẩm ‘100028’ |
| Rollback |  | T1 lỗi khi lượng tồn < 0, hủy cập nhật |

* Ảnh có chứa bàn

  Mô tả được tạo tự độngXem số lượng tồn ban đầu của sản phẩm ‘100028’ là 6.
* Chạy transaction 1 và cập nhật số lượng tồn của sản phẩm.
* Sau đó, chạy transaction 2 khi transaction 1 chưa thực hiện commit. Transaction 2 đọc thông tin sản phẩm và kết quả số lượng tồn của sản phẩm lúc này là -2 ( = 6 – 8).
* Ảnh có chứa văn bản

  Mô tả được tạo tự độngLúc này, transaction 1 kiểm tra số lượng tồn không hợp lệ nên trả về lỗi và rollback.
* Vì transaction 1 đã rollback nên số lượng tồn vẫn giữ nguyên ban đầu là 6.
* Tuy nhiên transaction 2 đã đọc số lượng tồn trước đó nên cho ra kết quả sai 🡪 **Dirty Read**



**Giải quyết tranh chấp**

Thiết lập mức cô lập READ COMMITTED cho transaction 2.

|  |  |  |
| --- | --- | --- |
| **Transaction 1** | **Transaction 2** | **Giải thích** |
| Set @soluongton = (Select SoLuongTon from SanPham where MaSP = @masp) |  |  |
| Set @soluongton = @soluongton - @soluongban |  |  |
| Update SanPham  Set SoLuongTon = @soluongton  Where MaSP = @masp |  | T1 xin cấp khóa X-lock trên bảng SanPham.  SQL cấp khóa X-lock cho T1. |
|  | Select \* from SanPham where MaSP = @masp  **(CHỜ)** | T2 xin cấp khóa S-lock trên bảng SanPham.  SQL không cấp S-lock cho T2 vì T1 đang giữ X-lock |
| Rollback |  |  |
|  | Select \* from SanPham where MaSP = @masp | T1 đã trả X-lock.  SQL cấp S-lock cho T2. |
|  | Commit |  |

Tình huống 2

Transaction 1: Giảm giá 10% cho tất cả sản phẩm được cung cấp bởi chi nhánh 1.

Transaction 2: Đọc thông tin các sản phẩm được cung cấp bởi chi nhánh 1.

|  |  |
| --- | --- |
| Transaction 1 | Transaction 2 |
| update SANPHAM  set Gia = convert(bigint, Gia \* (100-10)/100)  where MaChiNhanh = 1 |  |
|  | Select \* from SANPHAM  Where MaChiNhanh = 1 |
|  | commit tran |
| raiserror('Đã xảy ra sự cố', 16, 1);  rollback tran |  |

**Giải thích:**

Dữ liệu ban đầu:

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

Thực thi transaction 1 giảm giá 10% các sản phẩm thuộc chi nhánh 1. Sau đó, vì xảy ra sự cố nên transaction 1 bị rollback, và giá của các sản phẩm được giữ nguyên. Thế nhưng, trước khi transaction 1 rollback, transaction 2 đọc giá đã được giảm của các sản phẩm thuộc chi nhánh 1:

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Demo lỗi trên giao diện:**

Thực hiện giảm giá 10% các sản phẩm của chi nhánh 1:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Mặc dù việc thực hiện giảm giá thất bại, nhưng khi transaction 2 xem danh sách sản phẩm của chi nhánh 1 thì giá của sản phẩm vẫn là giá đã giảm:

Ảnh có chứa văn bản

Mô tả được tạo tự động

**Giải quyết tranh chấp:**

Thiết lập mức cô lập READ COMMITTED cho transaction 2.

|  |  |  |
| --- | --- | --- |
| Transaction 1 | Transaction 2 | Giải thích |
| update SANPHAM  set Gia = convert(bigint, Gia \* (100-10)/100)  where MaChiNhanh = 1 |  | T1 xin cấp khóa X-lock trên bảng SANPHAM.  SQL: cấp khóa X-lock trên bảng SANPHAM cho T1. |
|  | Select \* from SANPHAM  Where MaChiNhanh = 1  **(CHỜ)** | T2 xin cấp khóa S-lock trên bảng SANPHAM.  SQL: không cấp S-lock cho T2 vì T1 đang giữ X-lock. |
| raiserror('Đã xảy ra sự cố', 16, 1);  rollback tran |  |  |
|  | Select \* from SANPHAM  Where MaChiNhanh = 1 | T1 đã trả X-lock, SQL cấp S-lock cho T2. |
|  | commit tran |  |

1. **Unrepeatable Read**

Tình huống 1

Transaction 1: Đăng nhập vào tài khoản: ‘\_hopeful’ mật khẩu: ‘tOxd4cLQkFyI’

Transaction 2 : Đổi mật khẩu tài khoản : ‘\_hopeful’ thành ‘123456’

|  |  |
| --- | --- |
| Transaction 1 | Transaction 2 |
| declare @MK varchar(20), @TK varchar(20)  select @TK = '\_hopeful', @MK = 'tOxd4cLQkFyI'  if (LEN(@TK) > 20)  begin  raiserror(N'Tên đăng nhập không tồn tại', 16,1)  rollback tran  end |  |
| else if (LEN(@MK) > 20)  begin  raiserror(N'Sai mật khẩu', 16,1)  rollback tran  end |  |
| else if not exists(select \* from TaiKhoan  where TaiKhoan.TenTaiKhoan = @TK and TaiKhoan.MatKhau = @MK)  begin  raiserror (N'Sai tên đăng nhập hoặc mật khẩu', 16,1)  rollback tran  end |  |
| else  print (N'Đăng nhập thành công') |  |
|  | declare @TK varchar(20), @MK varchar(20), @MKMoi varchar(20), @MKMoiLan2 varchar(20)  select @TK = '\_hopeful', @MK = 'tOxd4cLQkFyI', @MKMoi = '123456',@MKMoiLan2 = '123456'  if not exists(select \* from TaiKhoan where taikhoan.TenTaiKhoan = @TK and TaiKhoan.MatKhau = @MK )  begin  raiserror(N'Tài khoản hoặc mật khẩu không chính xác',16,1)  rollback  end |
|  | else if (@MKMoi = null or len(@MKMoi) > 20 )  begin  raiserror (N'Mật khẩu mới không hợp lệ',16,1)  rollback  end |
|  | else if (@MKMoi != @MKMoiLan2 )  begin  raiserror (N'Nhập sai mật khẩu mới',16,1)  rollback  end |
|  | else  begin  update TaiKhoan  set MatKhau = @MKMoi  where TaiKhoan.TenTaiKhoan = @TK  print N'Đổi mật khẩu thành công'  end |
| if not exists (select \* from TaiKhoan  where TaiKhoan.TenTaiKhoan = @TK and TaiKhoan.MatKhau = @MK)  begin  raiserror(N'Không tìm thấy thông tin tài khoản này',16,1)  rollback tran  end |  |

**Giải thích:**

T1: Nhập đúng tài khoản và mật khẩu, hệ thống báo đăng nhập thành công nhưng T2 đã đổi mật khẩu trước khi T1 truy vấn được thông tin tài khoản ‘\_hopeful’

Graphical user interface, application

Description automatically generated

T2: Đổi mật khẩu thành công của tài khoản ‘\_hopeful’

Graphical user interface, text, application

Description automatically generated

Tình huống 2

Transaction 1: Tìm kiếm sản phẩm có từ khóa “hữu cơ”.

Transaction 2: Thay đổi tên sản phẩm 1 từ “Hạt hạnh nhân hữu cơ” thành “Hạt hạnh nhân”.

|  |  |
| --- | --- |
| Transaction 1 | Transaction 2 |
| if exists  (  Select \*  From SANPHAM SP  Where SP.TenSP LIKE ('%' + @TenSP + '%')  )  begin  Print (N'Tồn tại sản phẩm') |  |
|  | if not exists  (  select\*  from SANPHAM SP  where SP.MaSP = @MaSP  )  begin  raiserror('Không tìm thấy sản phẩm.', 16, 1)  rollback tran  end |
|  | else  begin  update SANPHAM  set [TenSP] = @TenMoi  where [MaSP] = @MaSP  commit tran  end |
| Select \*  From SANPHAM SP  Where SP.TenSP LIKE ('%' + @TenSP + '%')  end |  |
| commit tran |  |

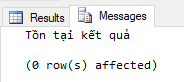
**Giải thích:**

Dữ liệu ban đầu:

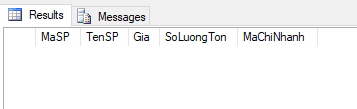


Thực thi transaction 1 để tìm kiếm sản phẩm có chứa từ khóa “hữu cơ”. Trong lúc transaction 1 đang thực thi, transaction 2 thay đổi tên của sản phẩm 1 từ “Hạt hạnh nhân hữu cơ” thành “Hạt hạnh nhân”.

Transaction 1 bị “unrepeatable read” khi tìm thấy sản phẩm có chứa từ khóa “hữu cơ” ở lần đọc đầu tiên (bên trong điều kiện của câu lệnh if):



Nhưng khi đọc lần 2 thì không xuất ra được thông tin sản phẩm nào:



**Demo trên giao diện**

Thực hiện tìm kiếm sản phẩm có từ khóa ”hữu cơ”. Trong lúc transaction 1 tìm kiểm thì thực hiện đổi tên sản phẩm ở transaction 2:

Ảnh có chứa văn bản

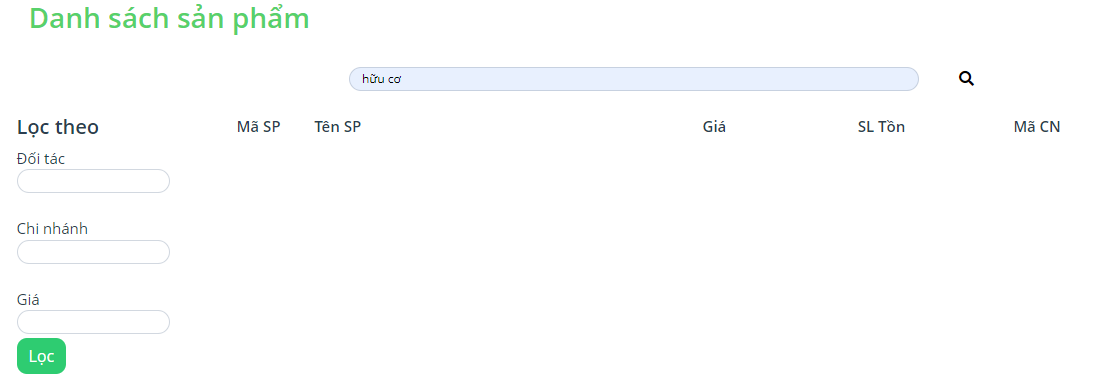
Mô tả được tạo tự động

Khi transaction 1 tìm kiếm kết thúc thì có thông báo tồn tại kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Nhưng danh sách kết quả thì không hiện thông tin sản phẩm nào:



**Giải quyết tranh chấp**

Thiết lập mức cô lập cho transaction 1 là REPEATABLE READ.

|  |  |  |
| --- | --- | --- |
| Transaction 1 | Transaction 2 | Giải thích |
| if exists  (  Select \*  From SANPHAM SP  Where SP.TenSP LIKE ('%' + @TenSP + '%')  )  begin  Print (N'Tồn tại sản phẩm') |  | T1 xin cấp khóa S-lock trên bảng SANPHAM  SQL: cấp khóa S-lock trên bảng SANPHAM cho T1  và T1 giữ S-lock tới hết giao tác |
|  | if not exists  (  select\*  from SANPHAM SP  where SP.MaSP = @MaSP  )  begin  raiserror('Không tìm thấy sản phẩm.', 16, 1)  rollback tran  end |  |
|  | else  begin  update SANPHAM  set [TenSP] = @TenMoi  where [MaSP] = @MaSP  commit tran  end  **(CHỜ)** | T2 xin cấp khóa X-lock trên bảng SANPHAM  SQL: không cấp khóa X-lock cho T2 vì T1 đang giữ S-lock. |
| Select \*  From SANPHAM SP  Where SP.TenSP LIKE ('%' + @TenSP + '%')  end |  |  |
| commit tran |  | T1 trả khóa S-lock trên bảng SANPHAM. |
|  | begin  update SANPHAM  set [TenSP] = @TenMoi  where [MaSP] = @MaSP  commit tran  end | SQL: cấp khóa X-lock trên bảng SANPHAM cho T2. |

1. **Phantom Read**

Tình huống 1

Transaction 1: Khách hàng xem các mặt hàng của chi nhánh 1

Transaction 2: Đối tác thêm vào chi nhánh 1 một mặt hàng

|  |  |
| --- | --- |
| **Transaction 1** | **Transaction 2** |
| declare @MaChiNhanh int  select @MaChiNhanh = 1  if not exists (select \* from CHINHANH  where CHINHANH.MaChiNhanh = 1 )  begin  raiserror(N'Không có chi nhánh này', 16,1)  rollback  end |  |
| declare @TongSoSP int  set @TongSoSP = (select count (sp.MaSP) from SANPHAM sp  where sp.MaChiNhanh = @MaChiNhanh)  print N'Có ' + cast (@TongSoSP as nvarchar(10))+N' sản phẩm' |  |
|  | Declare  @MaSP int,  @TenSP nvarchar(100),  @Gia bigint,  @SoLuongTon int = NULL,  @MaChiNhanh int = NULL  select  @MaSP = 5,  @TenSP = N'Bánh oreo ngon giòn 150g',  @Gia = 15000,  @SoLuongTon = 100,  @MaChiNhanh = 1  if @Gia < 0  begin  raiserror('Giá sản phẩm không hợp lệ.', 16, 1)  rollback tran  end |
|  | else  begin  insert into SANPHAM (TenSP, Gia, SoLuongTon, MaChiNhanh)  values (@TenSP, @Gia, @SoLuongTon, @MaChiNhanh) |
| select \* from SANPHAM sp where sp.MaChiNhanh = @MaChiNhanh |  |

**Kết quả:**

Transaction 1: Đếm được có tất cả 4 sản phẩm thuộc chi nhánh 1 nhưng xuất ra thông tin 5 sản phẩm do trước khi truy vấn thông tin của các sản phẩm thuộc chi nhánh 1 thì T2 đã thêm vào một sản phẩm

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application, table

Description automatically generated

Transaction 2:Thêm vào một sản phẩm thuộc chi nhánh 1 làm cho kết quả đọc của T1 trước và sau không cùng số bộ dữ liệu

Tình huống 2

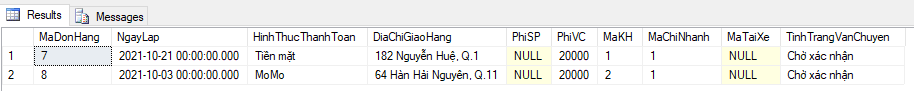
Transaction 1: Xem tất cả đơn hàng thuộc chi nhánh 1 trong tình trạng “Chờ xác nhận”.

Transaction 2: Thêm đơn hàng mới thuộc chi nhánh 1 cũng trong tình trạng “Chờ xác nhận”.

|  |  |
| --- | --- |
| Transaction 1 | Transaction 2 |
| if @MaChiNhanh != NULL and not exists  (  select \*  from CHINHANH CN  where CN.MaChiNhanh = @MaChiNhanh  )  begin  raiserror('Không tìm thấy chi nhánh.', 16, 1);  rollback tran  end |  |
| Set @SoDonHang = (  Select count(\*)  From DONHANG  Where MaChiNhanh = @MaChiNhanh and TinhTrangVanChuyen = @TinhTrang  ) |  |
| Print N'Tổng số đơn hàng trong tình trạng "Chờ xác nhận": ' + CAST(@SoDonHang AS VARCHAR(10)) |  |
|  | insert into DONHANG (HinhThucThanhToan, DiaChiGiaoHang, PhiVC, MaKH, MaChiNhanh, TinhTrangVanChuyen)  values (@HinhThucThanhToan, @DiaChiGiaoHang, @PhiVC, @MaKH, @MaChiNhanh, N'Chờ xác nhận') |
|  | commit tran |
| Select \*  From DONHANG  Where MaChiNhanh = @MaChiNhanh  and TinhTrangVanChuyen = @TinhTrang |  |
| commit tran |  |

**Giải thích:**

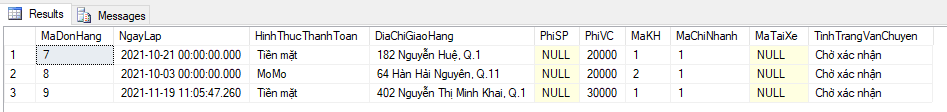
Dữ liệu ban đầu:



Thực thi transaction 1 để thống kê số lượng và xem thông tin các đơn hàng đang trong tình trạng “Chờ xác nhận”. Trong lúc transaction 1 đang thực thi, thực thi transaction 2 để thêm 1 đơn hàng mới với cũng tình trạng “Chờ xác nhận”.

Kết quả: Transaction 1 bị “phantom read” khi thống kê tổng số đơn hàng đang trong tình trạng “Chờ xác nhận” là 2, nhưng lại xuất ra 3 dòng kết quả:





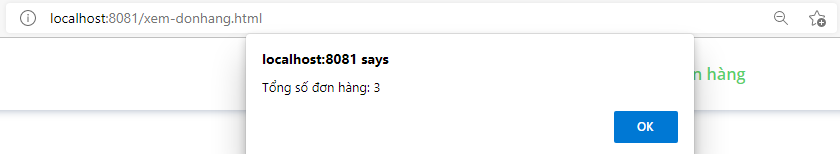
**Demo trên giao diện:**

Thực thi transaction 1 để lọc ra những đơn hàng thuộc chi nhánh 1 và có tình trạng là “Chờ xác nhận”.

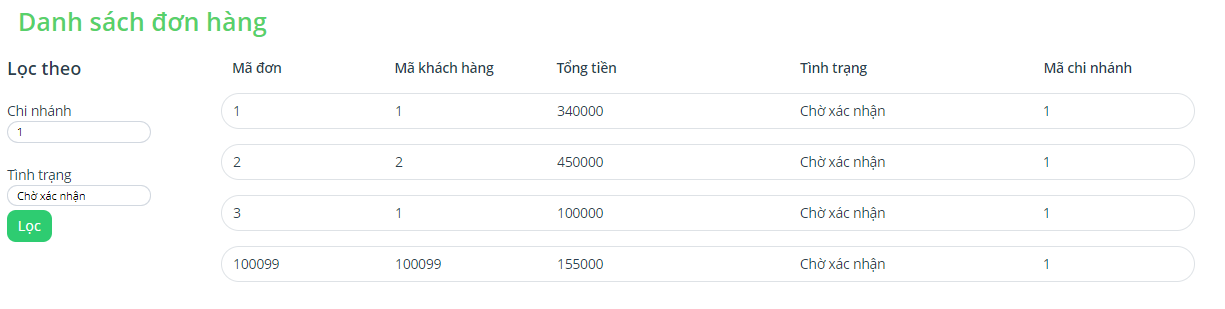
Trong lúc đó, thực hiện đặt hàng các sản phẩm thuộc chi nhánh 1 ở transaction 2:

**Ảnh có chứa văn bản

Mô tả được tạo tự động**Khi transaction 1 kết thúc việc lọc đơn hàng, hệ thống thông báo tổng số đơn hàng thỏa mãn điều kiện là 3:



Nhưng danh sách đơn hàng lại hiển thị thông tin của 4 đơn hàng:



**Giải quyết tranh chấp:**

Thiết lập mức cô lập SERIALIZABLE trên transaction 1.

|  |  |  |
| --- | --- | --- |
| Transaction 1 | Transaction 2 | Giải thích |
| if @MaChiNhanh != NULL and not exists  (  select \*  from CHINHANH CN  where CN.MaChiNhanh = @MaChiNhanh  )  begin  raiserror('Không tìm thấy chi nhánh.', 16, 1);  rollback tran  end |  |  |
| Set @SoDonHang = (  Select count(\*)  From DONHANG  Where MaChiNhanh = @MaChiNhanh and TinhTrangVanChuyen = @TinhTrang  ) |  | T1 xin S-lock trên bảng DONHANG  SQL: Cấp S-lock trên bảng DONHANG cho T1 và ngăn không cho thêm dòng dữ liệu |
| Print N'Tổng số đơn hàng trong tình trạng "Chờ xác nhận": ' + CAST(@SoDonHang AS VARCHAR(10)) |  |  |
|  | insert into DONHANG (HinhThucThanhToan, DiaChiGiaoHang, PhiVC, MaKH, MaChiNhanh, TinhTrangVanChuyen)  values (@HinhThucThanhToan, @DiaChiGiaoHang, @PhiVC, @MaKH, @MaChiNhanh, N'Chờ xác nhận')  **(CHỜ)** | T2 xin X-lock trên bảng DONHANG  SQL: không cấp X-lock cho T2 vì T1 ngăn không cho thêm dữ liệu trên bảng DONHANG |
| Select \*  From DONHANG  Where MaChiNhanh = @MaChiNhanh  and TinhTrangVanChuyen = @TinhTrang |  |  |
| commit tran |  | T1 trả S-lock trên bảng đơn hàng |
|  | insert into DONHANG (HinhThucThanhToan, DiaChiGiaoHang, PhiVC, MaKH, MaChiNhanh, TinhTrangVanChuyen)  values (@HinhThucThanhToan, @DiaChiGiaoHang, @PhiVC, @MaKH, @MaChiNhanh, N'Chờ xác nhận') | SQL cấp X-lock cho T2. |