

Đồ án thực hành

**KHOA CÔNG NGHỆ THÔNG TIN**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

Môn học: Hệ quản trị cơ sở dữ liệu

Giáo viên: Phạm Thị Bạch Huệ

Giáo viên: Hồ Thị Hoàng Vy

TP Hồ Chí Minh – Năm 2021

Người thực hiện

Huỳnh Bá Huy – 19127420

Dương Lê Xuân Khang – 19127432

Huỳnh Thu Thảo – 19127551

**Xây dựng hệ thống đặt và chuyển hàng online**

Mục lục

[Thông tin nhóm và đồ án thực hiện 3](#_Toc90822668)

[Bảng đánh giá mức độ tham gia đồ án của từng thành viên (100%/nhóm) 4](#_Toc90822669)

[Bảng đánh giá phần trăm hoàn tất công việc được giao của từng thành viên (100%/thành viên) 5](#_Toc90822670)

[Bảng phân công công việc theo tuần 6](#_Toc90822671)

[Bảng đánh giá mức độ hoàn thành từng công việc (100%/công việc) 8](#_Toc90822672)

[Thiết kế cơ sở dữ liệu 10](#_Toc90822673)

[1. Mô hình cơ sở dữ liệu: 10](#_Toc90822674)

[2. Đặc tả cơ sở dữ liệu: 10](#_Toc90822675)

[3. Mô tả thuộc tính 12](#_Toc90822676)

[Phân quyền người dùng 16](#_Toc90822677)

[4. Phân quyền trên các role user: 16](#_Toc90822678)

[Xác định tình huống tranh chấp 18](#_Toc90822679)

[5. Mô tả chức năng: 18](#_Toc90822680)

[6. Mô tả stored procerdure: 18](#_Toc90822681)

[7. Mô tả kịch bản của các tình huống lỗi tranh chấp đồng thời: 27](#_Toc90822682)

[7.1. Lost Update 27](#_Toc90822683)

[ TEST CASE 1: 27](#_Toc90822684)

[ TEST CASE 2: 29](#_Toc90822685)

[ TEST CASE 3: 31](#_Toc90822686)

[7.2. Dirty Read 33](#_Toc90822687)

[ TEST CASE 1: 33](#_Toc90822688)

[ TEST CASE 2: 35](#_Toc90822689)

[ TEST CASE 3: 37](#_Toc90822690)

[7.3. Unrepeatable Read 39](#_Toc90822691)

[ TEST CASE 1: 39](#_Toc90822692)

[ TEST CASE 2: 44](#_Toc90822693)

[7.4. Phantom 48](#_Toc90822694)

[ TEST CASE 1: 48](#_Toc90822695)

[7.5. Deadlock 51](#_Toc90822696)

[7.5.1. Converson Deadlock: 51](#_Toc90822697)

[ TEST CASE 1: 51](#_Toc90822698)

[7.5.2. Cycle Deadlock: 54](#_Toc90822699)

[ TEST CASE 2: 54](#_Toc90822700)

# Thông tin nhóm và đồ án thực hiện

Mã nhóm: 04

Đề tài : Xây dựng ứng dụng đặt và chuyển hàng online

Nội quy họp nhóm:

1. Thời gian họp nhóm: 14h thứ 7 hằng tuần
2. Công cụ họp nhóm: Discord (voice chat) & Zoom meetings.
3. Nội quy họp nhóm:
   1. Tham dự đúng giờ
   2. Kỷ luật, năng suất
   3. Tôn trọng lẫn nhau

Thành viên trong nhóm:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | MSSV | Họ và tên | Email |
| 1 | 19127420 | Huỳnh Bá Huy | [19127420@student.hcmus.edu.vn](mailto:19127420@student.hcmus.edu.vn) |
| 2 | 19127432 | Dương Lê Xuân Khang | [19127432@student.hcmus.edu.vn](mailto:19127432@student.hcmus.edu.vn) |
| 3 | 19127551 | Huỳnh Thu Thảo | [19127551@student.hcmus.edu.vn](mailto:19127551@student.hcmus.edu.vn) |

Nội dung đề tài:

+ Phân tích và thiết kế cơ sở dữ liệu cho ứng dụng

+ Phân quyền cho từng loại người dùng của hệ thống

+ Thiết kế ứng dụng trên nền tảng web.

+ Xác định tình huống tranh chấp và đề ra hướng giải quyết phù hợp

# Bảng đánh giá mức độ tham gia đồ án của từng thành viên (100%/nhóm)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | MSSV | Họ và tên | Mức độ tham gia | Đánh giá |
| 1 | 19127420 | Huỳnh Bá Huy | 100% | Tốt |
| 2 | 19127432 | Dương Lê Xuân Khang | 100% | Tốt |
| 3 | 19127551 | Huỳnh Thu Thảo | 90% | Tốt |

# Bảng đánh giá phần trăm hoàn tất công việc được giao của từng thành viên (100%/thành viên)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Nội dung đồ án | Họ và tên | Phần trăm hoàn thành | Đánh giá |
| 1 | Thiết kế cơ sở dữ liệu | Huỳnh Bá Huy | 100% | Tốt |
| Dương Lê Xuân Khang | 100% | Tốt |
| Huỳnh Thu Thảo | 100% | Tốt |
| 2 | Phân quyền người dùng | Huỳnh Bá Huy | 100% | Tốt |
| Dương Lê Xuân Khang | 100% | Tốt |
| Huỳnh Thu Thảo | 100% | Tốt |
| 3 | Cài đặt giao diện | Huỳnh Bá Huy | 100% | Tốt |
| Dương Lê Xuân Khang | 100% | Tốt |
| Huỳnh Thu Thảo | 100% | Tốt |
| 3 | Phân tích và xử lý tình huống tranh chấp | Huỳnh Bá Huy | 100% | Tốt |
| Dương Lê Xuân Khang | 100% | Tốt |
| Huỳnh Thu Thảo | 100% | Tốt |

# Bảng phân công công việc theo tuần

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thời gian | Công việc của Huy | Công việc của Khang | Công việc của Thảo |
| 1 | 24/10/2021  –  28/10/2021 | + Tham gia thiết kế cấu trúc cơ sở dữ liệu  + Viết scripts tạo các bảng cho tài xế  + Viết báo cáo | + Tham gia thiết kế cấu trúc cơ sở dữ liệu  + Viết scripts tạo các bảng cho khách hàng | + Tham gia thiết kế cấu trúc cơ sở dữ liệu  + Viết scripts tạo các bảng cho đối tác |
| 2 | 28/10/2021  – 02/11/2021 | + Tham gia phân quyền cho cơ sở dữ liệu  + Viết scripts phân quyền cho các bảng tài xế, nhân viên.  + Viết báo cáo | + Tham gia phân quyền cho cơ sở dữ liệu  + Viết scripts phân quyền cho các bảng admin | + Tham gia phân quyền cho cơ sở dữ liệu  + Viết scripts phân quyền cho các bảng đối tác, khách hàng |
| 3 | 09/11/2021  – 22/11/2021 | + Tham gia phân tích yêu cầu và nhận biết lỗi  + Viết scripts store procedure cho tài xế và nhân viên  + Viết script và document mô tả tình huống của lỗi deadlock  + Viết báo cáo nội dung phân công công việc | + Tham gia phân tích yêu cầu và nhận biết lỗi  + Viết scripts store procedure cho admin  + Viết script và document mô tả tình huống của lỗi unreapeatable read và phantom read.  + Viết báo cáo cho tổng hợp giao diện | + Tham gia phân tích yêu cầu và nhận biết lỗi  + Viết scripts store procedure cho đối tác và khách hàng.  + Viết script và document mô tả tình huống của lỗi dirty read và lost update.  + Xây dựng dữ liệu mẫu  + Tổng hợp báo cáo cho nhóm |
| 4 | 22/11/2021  – 29/11/2021 | +Viết script và document mô tả sửa lỗi cho deadlock | + Viết script và document mô tả sửa lỗi cho unreapeatable read và phantom | + Viết script và document mô tả sửa lỗi cho lost update và dirty read  + Thiết kế layout GUI |
| 5 | 29/11/2021  – 11/12/2021  (2 tuần) | + Code giao diện cho web.  + Quay clip mô tả cho ER | + Code giao diện cho web.  + Code tính năng đăng nhập | + Chuẩn bị dữ liệu mẫu và kiểm tra tất cả các scripts sửa lỗi |
| 6 | 11/12/2021  – 18/12/2021 | + Code tính năng gia hạn hợp đồng, xóa hợp đồng cho nhân viên  + Code tính năng chấp nhận đơn hàng và cập nhật tình trạng đơn hàng cho tài xế | + Code tính năng cập nhật tình trạng đơn hàng, cập nhật sản phẩm cho đối tác  + Chọn mua đơn hàng cho khách hàng.  + Quay video demo. | + Lên kế hoạch kịch bản để quay clip demo.  + Chỉnh sửa video.  + Chỉnh sửa báo cáo. |

# Bảng đánh giá mức độ hoàn thành từng công việc (100%/công việc)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Nội dung | Người thực hiện | Mức độ hoàn thành | Đánh giá |
| 1 | Thiết kế mô hình cơ sở dữ liệu | Cả nhóm (họp) | 100% | Tốt |
| Viết scripts tạo các bảng cho tài xế | Huỳnh Bá Huy | 100% | Tốt |
| Viết scripts tạo các bảng cho khách hàng | Dương Lê Xuân Khang | 100% | Tốt |
| Viết scripts tạo các bảng cho đối tác | Huỳnh Thu Thảo | 100% | Tốt |
| 2 | Viết scripts phân quyền cho các bảng tài xế, nhân viên. | Huỳnh Bá Huy | 100% | Tốt |
| Viết scripts phân quyền cho các bảng admin | Dương Lê Xuân Khang | 100% | Tốt |
| Viết scripts phân quyền cho các bảng đối tác, khách hàng | Huỳnh Thu Thảo | 100% | Tốt |
| 3 | Viết store procedure cho đối tác và khách hàng | Huỳnh Thu Thảo | 100% | Tốt |
| Viết store procedure cho admin | Dương Lê Xuân Khang | 100% | Tốt |
| Viết store procedure cho nhân viên và tài xế. | Huỳnh Bá Huy | 100% | Tốt |
| 4 | Viết scripts sửa lỗi và fix lỗi cho lostupdate và dirty read | Huỳnh Thu Thảo | 80% | Tốt |
| Viết scripts sửa lỗi và fix lỗi cho unrepeatable read và phantom | Dương Lê Xuân Khang | 100% | Tốt |
| Viết scripts sửa lỗi và fix lỗi cho deadlock | Huỳnh Bá Huy | 100% | Tốt |
| 5 | Code tính năng cho nhân viên và tài xế | Huỳnh Bá Huy | 100% | Tốt |
| Code tính năng cho đối tác và khách hàng | Dương Lê Xuân Khang | 100% | Tốt |
| Chuẩn bị dữ liệu mẫu.  Lên kịch bản chi tiết quay video demo. | Huỳnh Thu Thảo | 100% | Tốt |
| 6 | Quay video giới thiệu ER | Huỳnh Bá Huy | 100% | Tốt |
| Quay video demo lỗi | Dương Lê Xuân Khang | 100% | Tốt |
| Chỉnh sửa video và tổng hợp báo cáo | Huỳnh Thu Thảo | 100% | Tốt |

# Thiết kế cơ sở dữ liệu

## Mô hình cơ sở dữ liệu:



Hình 1.1: Diagram của cơ sở dữ liệu của hệ thống

## Đặc tả cơ sở dữ liệu:

DOI\_TAC(**MaDT**, TenDT, NguoiDaiDien, MaKV, SoChiNhanh, SLDH, MaLoai, DiaChiKD, SoDT, Email, MaSoThue)

Tân từ: Đối tác của hệ thống là các nhà doanh nghiệp có nhu cầu vận chuyển hàng hóa, sản phẩm, thực phẩm, ... đến khách hàng. Mỗi đối tác cần có một mã để phân biệt với các đối tác khác, tên đối tác, người đại diện, mã khu vực, số chi nhánh, số lượng đơn hàng mỗi ngày, mã loại hàng vận chuyển, địa chỉ kinh doanh, số điện thoại, email, mã số thuế

CHI\_NHANH (**MaCN**, MaDT, MaKV , DiaChiCuThe)

Tân từ: Mỗi đối tác sẽ có một hay nhiều chi nhánh. Mỗi chi nhánh cần có một mã để phân biệt với các chi nhánh khác và có những thông tin khác gồm: mã đối tác, mã khu vực, địa chỉ cụ thể

LOAI\_HANG (**MaLoai**, TenLoai)

Tân từ: Mỗi loại hàng có một mã để phân biệt và tên của loại hàng đó

HOP\_DONG (**MaHD**, MaDT, SoCNDangKy, TG\_HieuLucHD, PhanTramHoaHong)

Tân từ: Đối tác sau khi đã đăng ký thông tin thành công cần lập hợp đồng để được phục vụ. Hợp đồng gồm có: mã hợp đồng, số lượng chi nhánh đăng ký, thời gian hiệu lực, phần trăm hoa hồng theo quy định hiện tại của công ty

CT\_HOPDONG (**MaHD**, **MaCN**)

Tân từ: Chi tiết hợp đồng thể hiện hợp đồng được áp dụng với những chi nhánh nào. Mỗi chi tiết hợp đồng gồm: mã hợp đồng và mã chi nhánh

DON\_HANG (**MaDH**, MaDT, MaKH, HinhThuc\_ThanhToan, TenDuong, MaKV, TongPhiSP, PhiVanChuyen, TinhTrangDH)

Tân từ: Mỗi đơn hàng cần có một mã để phân biệt với các đơn hàng khác ngoài ra còn có: mã khách hàng, hình thức thanh toán, tên đường, mã khu vực, tổng phí sản phẩm, phí vận chuyển, tình trạng đơn hàng.

CT\_DONHANG (**MaDH**, **MaSP**, SoLuong)

Tân từ: Một đơn hàng có thể có nhiều loại sản phẩm, chi tiết đơn hàng cho biết những sản phẩm nào thuộc về đơn hàng nào. Mỗi chi tiết đơn hàng gồm: mã đơn hàng, mã sản phẩm, số lượng cho từng loại sản phẩm

KHACH\_HANG (**MaKH**, TenKH, SoDT)

Tân từ: Mỗi khách hàng cần một mã để phân biệt với các khách hàng khác, ngoài ra còn có: tên khách hàng, số điện thoại

KHU\_VUC (**MaKV**, Quan, Tinh)

Tân từ: Mỗi khu vực cần có một mã để phân biệt với các khu vực khác, ngoài ra còn

có các thông tin như: quận, tỉnh

SAN\_PHAM (**MaSP**, MaCN, TenSanPham, Loai, Gia)

Tân từ: Mỗi sản phẩm cần có một mã để phân biệt với các sản phẩm khác, ngoài ra còn có các thông tin như: mã chi nhánh cung cấp sản phẩm, tên sản phẩm, loại sản phẩm và giá của sản phẩm đó

TAI\_XE (**MaTX**, HoTen, Cmnd, SoDT, DiaChi, BienSo, MaKV, Email, TaiKhoanNH)

Tân từ: Mỗi tài xế cần có một mã để phân biệt với các tài xế khác, ngoài ra còn có: họ tên của tài xế, chứng minh nhân dân, số điện thoại, địa chỉ, biển số xe, mã khu vực, email, tài khoản ngân hàng

GIAO\_HANG(**MaTX**, **MaDH**, HoaHong)

Tân từ: Tài xế tiếp nhận đơn hàng và giao hàng, những đơn hàng mà tài xế tiếp nhận và tiền hoa hồng cho từng đơn hàng sẽ được lưu lại. Thủ tục giao hàng cần có mã tài xế, mã đơn hàng, tiền hoa hồng cho tài xế ứng với đơn hàng đó

## Mô tả thuộc tính

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thuộc tính | Diễn giải | Kiểu dữ liệu |
| DOI\_TAC | MaDT | Mã đối tác | varchar(20) |
| TenDT | Tên đối tác | nvarchar(50) |
| NguoiDaiDien | Người đại diện | nvarchar(50) |
| MaKV | Mã khu vực | varchar(20) |
| SoChiNhanh | Số chi nhánh | int |
| SLDH | Số lượng đơn hàng mỗi ngày | int |
| MaLoai | Mã loại | varchar(20) |
| DiaChiKD | Địa chỉ kinh doanh | nvarchar(50) |
| SoDT | Số điện thoại | varchar(15) |
| Email | Email | varchar(50) |
| MaSoThue | Mã số thuế | varchar(20) |
| CHI\_NHANH | MaCN | Mã chi nhánh | varchar(20) |
| MaDT | Mã đối tác | varchar(20) |
| MaKV | Mã khu vực | varchar(20) |
| DiaChiCuThe | Địa chỉ cụ thể | nvarchar(50) |
| LOAI\_HANG | MaLoai | Mã loại | varchar(20) |
| TenLoai | Tên loại | nvarchar(50) |
| HOP\_DONG | MaHD | Mã hợp đồng | varchar(20) |
| MaDT | Mã đối tác | varchar(20) |
| SoCNDangKy | Số chi nhánh đăng ký | varchar(15) |
| TG\_HieuLucHD | Thời gian hiệu lực của hợp đồng | nvarchar(10) |
| PhanTramHoaHong | Phầm trăm hoa hồng | float |
| CT\_HOPDONG | MaHD | Mã hợp đồng | varchar(20) |
| MaCN | Mã chi nhánh | varchar(20) |
| DON\_HANG | MaDH | Mã đơn hàng | varchar(20) |
| MaDT | Mã đối tác | varchar(20) |
| MaKH | Mã khách hàng | varchar(20) |
| HinhThuc\_ThanhToan | Hình thức thanh toán | nvarchar(50) |
| TenDuong | Tên Đường | nvarchar(50) |
| MaKV | Mã khu vực | varchar(20) |
| TongPhiSP | Tổng phí các sản phẩm đã mua | float |
| PhiVanChuyen | Phí vận chuyển | float |
| TinhTrangDH | Tình trạng đơn hàng | nvarchar(50) |
| CT\_DONHANG | MaDH | Mã đơn hàng | varchar(20) |
| MaSP | Mã sản phẩm | varchar(20) |
| SoLuong | Số lượng của mỗi loại sản phẩm | int |
| KHACH\_HANG | MaKH | Mã khách hàng | varchar(20) |
| TenKH | Tên khách hàng | nvarchar(50) |
| SoDT | Số điện thoại | varchar(15) |
| KHU\_VUC | MaKV | Mã khu vực | varchar(20) |
| Quan | Quận | nvarchar(50) |
| Tinh | Tỉnh | nvarchar(50) |
| SAN\_PHAM | MaSP | Mã sản phẩm | varchar(20) |
| MaCN | Mã chi nhánh | varchar(20) |
| TenSanPham | Tên sản phẩm | nvarchar(50) |
| Loai | Loại sản phẩm | varchar(20) |
| Gia | Giá sản phẩm | float |
| TAI\_XE | MaTX | Mã tài xế | varchar(20) |
| HoTen | Họ tên | nvarchar(50) |
| Cmnd | Chứng minh nhân dân | varchar(15) |
| SoDT | Số điện thoại | varchar(15) |
| DiaChi | Địa chỉ | nvarchar(100) |
| BienSo | Biển số xe | varchar(15) |
| MaKV | Mã khu vực | varchar(20) |
| Email | Email | varchar(50) |
| TaiKhoanNH | Tài khoản ngân hàng | nvarchar(100) |
| GIAO\_HANG | MaTX | Mã tài xế | varchar(20) |
| MaDH | Mã đơn hàng | varchar(20) |
| HoaHong | Hoa hồng | money |

# Phân quyền người dùng

## Phân quyền trên các role user:

* Role đối tác:
  + Có quyền xem, xóa sửa trên bảng KHU\_VUC
  + Có quyền xem bảng LOAI\_HANG
  + Có quyền xem hai thuộc tính MADT(mã đối tác) và TenDT(tên đối tác) trên bảng DOI\_TAC
  + Có quyền xem bảng DON\_HANG
  + Có quyền sửa thuộc tính TTDH (tình trạng đơn hàng) trên bảng DON\_HANG
  + Có quyền xem bảng CT\_HOPDONG
  + Có quyền xem bảng HOP\_DONG
* Role khách hàng:
  + Có quyền xem MADT(mã đối tác), TenDT(tên đối tác), NguoiDaiDien(Người đại diện), MaKV(mã khu vực), DiaChiKD(địa chỉ kinh doanh), SoDT(số điện thoại), Email trên bảng DOI\_TAC
  + Có quyền xem bảng DON\_HANG
  + Có quyền xem bảng CT\_DONHANG
  + Có quyền xem MaKH(mã khách hàng), TenKH(tên khách hàng) trên bảng KHACH\_HANG
  + Có quyền xem MaSP(mã sản phẩm), TenSanPham(TenSP), Gia(TenSP), Loai trên bảng SAN\_PHAM
  + Có quyền xem bảng LOAI\_HANG
* Role tài xế:
  + Có quyền xem bảng TAI\_XE
  + Có quyền xem bảng DON\_HANG
  + Có quyền xem bảng KHU\_VUC
  + Có quyền xem và thêm bảng GIAO\_HANG
  + Có quyền cập nhật bảng DON\_HANG
  + Có quyền xem MaKH(mã khách hàng), TenKH(tên khách hàng) trên bảng KHACH\_HANG
* Role nhân viên:
  + Có quyền xem bảng HOP\_DONG
  + Có quyền cập nhật TG\_HieuLucHD, PhanTramHoaHong trên bảng HOP\_DONG
  + Có quyền xem bảng DOI\_TAC
  + Có quyên xem bảng LOAI\_HANG
  + Có quyền thêm bảng HOP\_DONG
  + Có quyền xem, thêm, cập nhật bảng CT\_HOPDONG
  + Có quyền xem bảng CHI\_NHANH
  + Có quyền xem bảng KHU\_VUC
* Role admin:
  + System admin có các quyền sau:
    - Thêm/xóa/sửa tài khoản login SQL Server và tài khoản user của riêng database
    - Khóa/kích hoạt tài khoản login SQL Server và tài khoản user của riêng database
    - Thao tác trên database như db\_owner
  + Database admin có các quyền sau:
    - Thao tác trên database như db\_owner

# Xác định tình huống tranh chấp

## 5. Mô tả chức năng:

|  |  |
| --- | --- |
| Đối tượng | Chức năng |
| Đối tác | - Cập nhật thời gian hiệu lực và phần trăm hoa hồng  - Thêm, xóa, sửa thông tin sản phẩm và chi nhánh có cung cấp sản phẩm này  - Xem thông tin đơn hàng và cập nhật tình trạng của đơn hàng |
| Khách hàng | - Xem danh sách đối tác  - Chọn sản phẩm, số lượng tương ứng, hình thức thanh toán và địa chỉ giao hàng  - Khi xác nhận đồng ý, đơn hàng sẽ được chuyển đến đối tác và tài xế  - Theo dõi quá trình vận chuyển đơn hàng |
| Tài xế | - Danh sách đơn hàng sẽ hiển thị theo khu vực  - Chọn đơn hàng sẽ phục vụ  - Cập nhật tình trạng đơn hàng  - Danh sách các đơn hàng tài xế đã nhận và phí vận chuyển tài xế |
| Nhân viên | - Xem danh sách hợp đồng của đối tác  - Xem danh sách hợp đồng đã lập của đối tác  - Duyệt hợp đồng |
| Admin | - Cập nhật thông tin tài khoản  - Thêm, xóa, sửa tài khoản admin, nhân viên |

## 6. Mô tả stored procerdure:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thủ tục | Mục đích | Tham số | Các bước thực hiện |
| UpdateContract | Đối tác cập nhật thời gian hiệu lực và phần trăm hoa hồng của hợp đồng | @masothue varchar(20), @tg\_hlhd date, @pthh float | Kiểm tra có phải là “Đối tác” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của hợp đồng:  - Nếu không tồn tại, rollback  - Nếu tồn tại, check thời gian định cập nhật xem có hợp lệ hay không  + Nếu không hợp lê, rollback  + Nếu hợp lệ, cập nhật thời gian hiệu lực và phần trăm hoa hồng của hợp đồng |
| AddProduct | Đối tác thêm thông tin sản phẩm và chi nhánh cung cấp sản phẩm này | @maSP varchar(20), @maCN varchar(20), @tensanpham nvarchar(50), @loai varchar(20), @gia float | Kiểm tra có phải là “Đối tác” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của sản phẩm:  - Nếu tồn tại, rollback  - Nếu không tồn tại, thêm thông tin sản phẩm và chi nhánh cung cấp sản phẩm này |
| UpdateProduct | Đối tác sửa thông tin sản phẩm và chi nhánh cung cấp sản phẩm này | @maSP varchar(20), @maCN varchar(20), @tensanpham nvarchar(50), @loai varchar(20), @gia float | Kiểm tra có phải là “Đối tác” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của sản phẩm:  - Nếu không tồn tại, rollback  - Nếu tồn tại, sửa thông tin sản phẩm và chi nhánh cung cấp sản phẩm này |
| DeleteProduct | Đối tác xóa thông tin sản phẩm và chi nhánh cung cấp sản phẩm này | @maSP varchar(20), @maCN varchar(20), @tensanpham nvarchar(50), @loai varchar(20), @gia float | Kiểm tra có phải là “Đối tác” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của sản phẩm:  - Nếu không tồn tại, rollback  - Nếu tồn tại, xóa thông tin sản phẩm và chi nhánh cung cấp sản phẩm này |
| ViewOrderInformation | Đối tác xem thông tin đơn hàng | @madh varchar(20) | Kiểm tra có phải là “Đối tác” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của đơn hàng:  - Nếu không tồn tại, rollback  - Nếu tồn tại, xem thông tin đơn hàng |
| UpdateOrderStatus | Đối tác cập nhật tình trạng đơn hàng | @madt varchar(20), @madh varchar(20), @ttdh nvarchar(50) | Kiểm tra có phải là “Đối tác” và “Tài xế” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của đơn hàng:  - Nếu không tồn tại, rollback  - Nếu tồn tại, cập nhật tình trạng đơn hàng |
| ViewPartnerList | Khách hàng xem danh sách đối tác |  | Kiểm tra có phải là “Khách hàng” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của đối tác:  - Nếu không tồn tại, rollback  - Nếu tồn tại, xem thông tin của đối tác (mã đối tác, người đại diện, mã khu vực, mã loại, địa chỉ kinh doanh, số điện thoại, email) |
| ViewProductListOfPartner | Khách hàng xem danh sách sản phẩm của đối tác | @madt varchar(20) | Kiểm tra có phải là “Khách hàng” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của sản phẩm:  - Nếu không phải sản phẩm của đối tác, rollback  - Nếu là sản phẩm của đối tác, xem thông tin sản phẩm của đối tác (mã sản phẩm, tên sản phẩm, giá, loại) |
| SelectOrderInformation | Khách hàng chọn sản phẩm, số lượng tương ứng, hình thức thanh toán và địa chỉ giao hàng | @masp varchar(20), @soluong int, @ht\_tt nvarchar(50), @tenduong nvarchar(50), @makv varchar(20) | Kiểm tra có phải là “Khách hàng” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của sản phẩm:  - Nếu không tồn tại, rollback  - Nếu tồn tại, chọn sản phẩm, số lượng tương ứng, hình thức thanh toán và địa chỉ giao hàng |
| CreateOrder | Khách hàng tạo đơn hàng | @madh varchar(20), @madt varchar(20), @makh varchar(20), @ht\_tt nvarchar(50), @tenduong nvarchar(50), @makv varchar(20), @masp varchar(20), @soluong int | Kiểm tra có phải là “Khách hàng” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của đơn hàng:  - Nếu tồn tại, rollback  - Nếu không tồn tại, thêm đơn hàng vào danh sách đơn hàng, chi tiết đơn hàng, cập nhật tổng tiền |
| UpdateOrder | Khách hàng cập nhật đơn hàng | @madh varchar(20), @madt varchar(20), @makh varchar(20), @ht\_tt nvarchar(50), @tenduong nvarchar(50), @makv varchar(20), @masp varchar(20), @soluong int | Kiểm tra có phải là “Khách hàng” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của đơn hàng:  - Nếu không tồn tại, rollback  - Nếu tồn tại:  + Nếu sản phẩm được cập nhật đã có trong đơn hàng, cập nhật số lượng, hình thức thanh toán, địa chỉ giao hàng, tổng tiền  + Nếu sản phẩm được cập nhật không có trong đơn hàng, thêm sản phẩm vào đơn hàng, cập nhật tổng tiền |
| ViewShippingProcess | Khách hàng theo dõi quá trình vận chuyển | @madh varchar(20) | Kiểm tra có phải là “Khách hàng” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của đơn hàng:  - Nếu không tồn tại, rollback  - Nếu tồn tại, xem tình trạng đơn hàng |
| ViewOrderList | Tài xế xem danh sách đơn hàng theo khu vực | @matx varchar(20) | Kiểm tra có phải là “Tài xế” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của tài xế:  - Nếu không tồn tại, rollback  - Nếu tồn tại, xem đơn hàng theo khu vực của tài xế |
| SelectOrder | Tài xế chọn đơn hàng phục vụ | @maTX varchar(20), @madh varchar(20) | Kiểm tra có phải là “Tài xế” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại:  - Nếu tài xế không tồn tại hoặc đơn hàng đang được giao hoặc đơn hàng không tồn tại hoặc đơn hàng không thuộc khu vực mà tài xế phục vụ, rollback  - Nếu tồn tại, xem đơn hàng theo khu vực của tài xế |
| UpdateOrderStatus | Tài xế cập nhật tình trạng đơn hàng | @matx varchar(20), @madh varchar(20), @ttdh nvarchar(50) | Kiểm tra có phải là “Tài xế” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra đơn hàng có đang được giao bởi tài xế:  - Nếu không tồn tại, rollback  - Nếu tồn tại, cập nhật tình trạng đơn hàng |
| ViewMyOrders | Xem danh sách đơn hàng mà tài xế đã nhận và phí vận chuyển của từng đơn hàng | @matx varchar(20) | Kiểm tra có phải là “Tài xế” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của tài xế:  - Nếu không tồn tại, rollback  - Nếu tồn tại, xem danh sách đơn hàng mà tài xế đã nhận và phí vận chuyển của từng đơn hàng |
| GetExpiredContract | Xem danh sách hợp đồng của đối tác | @madt varchar(20) | Kiểm tra có phải là “Nhân viên” và “Đối tác” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của đối tác:  - Nếu không tồn tại, rollback  - Nếu tồn tại, xem danh sách hợp đồng của đối tác |
| GetAllConstractList | Xem danh sách hợp đồng đã lập của đối tác | @madt varchar(20) | Kiểm tra có phải là “Nhân viên” và “Đối tác” và “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của đối tác:  - Nếu không tồn tại, rollback  - Nếu tồn tại, xem danh sách hợp đồng đã lập của đối tác |
| changeLoginName | Đổi tên đăng nhập | @oldName nvarchar(30), @newName nvarchar(30) | Kiểm tra tài khoản có là tài khoản hệ thống admin:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra tính hợp lệ của tên hiện tại và tên mới đăng nhập:  - Nếu không hợp lệ, rollback  - Nếu hợp lệ, đổi tên đăng nhập |
| changeLoginPassword | Đổi mật khẩu đăng nhập | @loginName nvarchar(30), @newPassword nvarchar(30) | Kiểm tra tài khoản có là tài khoản hệ thống admin:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra tính hợp lệ của tên hiện tại và mật khẩu mới đăng nhập:  - Nếu không hợp lệ, rollback  - Nếu hợp lệ, đổi mật khẩu đăng nhập |
| changeUsername | Đổi tên người dùng trong cơ sở dữ liệu | @oldUsername nvarchar(30), @newUsername nvarchar(30) | Kiểm tra tài khoản có là tài khoản hệ thống admin:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra tính hợp lệ của tên hiện tại và tên mới người dùng:  - Nếu không hợp lệ, rollback  - Nếu hợp lệ, đổi tên người dùng |
| addLoginAccount | Thêm tài khoản đăng nhập cho admin/ nhân viên | @loginName nvarchar(30), @loginPassword nvarchar(30) | Kiểm tra tài khoản có là tài khoản hệ thống admin:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra tính hợp lệ của tên và mặt khẩu tài khoản đăng nhập:  - Nếu không hợp lệ, rollback  - Nếu hợp lệ, thêm tài khoản đăng nhập |
| addUserForAdmin | Thêm tài khoản người dùng cho admin | @userName nvarchar(30), @loginName nvarchar(30) | Kiểm tra tài khoản có là tài khoản hệ thống admin:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra tính hợp lệ của tên người dùng và tên tài khoản đăng nhập:  - Nếu không hợp lệ, rollback  - Nếu hợp lệ, thêm người dùng và vai trò admin |
| addUserForEmployee | Thêm tài khoản người dùng cho nhân viên | @userName nvarchar(30), @loginName nvarchar(30) | Kiểm tra tài khoản có là tài khoản hệ thống admin:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra tính hợp lệ của tên người dùng và tên tài khoản đăng nhập:  - Nếu không hợp lệ, rollback  - Nếu hợp lệ, thêm người dùng và vai trò nhân viên |
| deleteUserAccount | Xóa tài khoản người dùng của admin/ nhân viên | @userName nvarchar(30) | Kiểm tra tài khoản có là tài khoản hệ thống admin:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra tính hợp lệ của tên người dùng:  - Nếu không hợp lệ, rollback  - Nếu hợp lệ, xóa tài khoản người dùng |
| lockLoginAccount | Khóa tài khoản đăng nhập của admin/ nhân viên | @loginName nvarchar(30) | Kiểm tra tài khoản có là tài khoản hệ thống admin:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra tính hợp lệ của tên đăng nhập:  - Nếu không hợp lệ, rollback  - Nếu hợp lệ, khóa tài khoản người dùng |
| unlockLoginAccount | Mở khóa tài khoản đăng nhập của admin/ nhân viên | @loginName nvarchar(30) | Kiểm tra tài khoản có là tài khoản hệ thống admin:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra tính hợp lệ của tên người dùng:  - Nếu không hợp lệ, rollback  - Nếu hợp lệ, mở khóa tài khoản người dùng của admin/ nhân viên |
| lockUserAccount | Khóa tài khoản người dùng của admin/ nhân viên | @userName nvarchar(30) | Kiểm tra tài khoản có là tài khoản hệ thống admin:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra tính hợp lệ của tên người dùng:  - Nếu không hợp lệ, rollback  - Nếu hợp lệ, khóa tài khoản người dùng của admin/ nhân viên |
| unlockUserAccount | Mở khóa tài khoản người dùng của admin/nhân viên | @userName nvarchar(30) | Kiểm tra tài khoản có là tài khoản hệ thống admin:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra tính hợp lệ của tên người dùng:  - Nếu không hợp lệ, rollback  - Nếu hợp lệ, mở khóa tài khoản người dùng của admin/ nhân viên |
| insertPartner | Thêm đối tác vào cơ sở dữ liệu | @maDT varchar(20),  @tenDT nvarchar(50),  @nguoiDaiDien nvarchar(50),  @maKV varchar(20),  @soChiNhanh int,  @soLuongDH int,  @maLoai varchar(20),  @diaChiKD nvarchar(50),  @soDT varchar(15),  @email varchar(50),  @maSoThue varchar(20),  @danhSachChiNhanh branchList READONLY | Kiểm tra có phải là “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của đối tác:  - Nếu tồn tại, rollback  - Nếu không tồn tại, thêm đối tác và chi nhánh của đối tác đó |
| insertCustomer | Thêm khách hàng vào cơ sở dữ liệu | @maKH varchar(20),  @tenKH nvarchar(50),  @soDT varchar(15) | Kiểm tra có phải là “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của khách hàng:  - Nếu tồn tại, rollback  - Nếu không tồn tại, thêm khách hàng |
| insertDriver | Thêm tài xế vào cơ sở dữ liệu | @maTX varchar(20),  @tenTX nvarchar(50),  @cmnd varchar(15),  @soDT varchar(15),  @diaChi nvarchar(100),  @bienSo varchar(15),  @maKV varchar(20),  @email varchar(50),  @taiKhoanNH nvarchar(100) | Kiểm tra có phải là “Chủ sở hữu cơ sở dữ liệu”:  ▪ Nếu không phải, rollback  ▪ Nếu phải, kiểm tra sự tồn tại của tài xế:  - Nếu tồn tại, rollback  - Nếu không tồn tại, tài xế |

## 7. Mô tả kịch bản của các tình huống lỗi tranh chấp đồng thời:

### 7.1. Lost Update

* TEST CASE 1:

Mô tả tình huống tranh chấp:

Giả sử sếp giao việc cho nhân viên A là xem và tái ký hợp đồng (cập nhật lại thời gian hiệu lực và phần trăm hợp đồng) cho đối tác M (**gia hạn thêm 3 tháng so với thời hạn cũ**). Tuy nhiên, nhân viên A chưa làm ngay, lúc này đối tác liên lạc lại với sếp và đổi lại thành gia hạn thêm **2 tháng** (**tức là giảm 1 tháng so với giao kèo trước đó**), sếp quên mất đã giao cho nhân viên A nên giao lại việc này cho nhân viên B mà không thông báo lại với nhân viên A. Giả sử lúc này cả hai nhân viên A và B cùng lúc truy cập vào hệ thống và thực hiện việc tái ký hợp đồng. Lost update xảy ra.

|  |  |  |
| --- | --- | --- |
| T1 - đối tác (nhân viên A) cập nhật thời gian hiệu lực hợp đồng | T2 - đối tác (nhân viên B) cập nhật thời gian hiệu lực hợp đồng | GHI CHÚ |
| SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON (d.MaDT = h.MaDT) WHERE d.MaSoThue = @masothue |  | Nhân viên A kiểm tra tính hợp lệ của thời gian truyền vào |
| UPDATE HOP\_DONG  SET TG\_HieuLucHD = @tg\_hlhd, PhanTramHoaHong = @pthh where MaDT IN (SELECT MaDT FROM DOI\_TAC WHERE MaSoThue = @masothue) |  | Nhân viên A cập nhật thời gian hiệu lực và phần trăm hoa hồng của hợp đồng đối tác (gia hạn thêm 3 tháng) |
|  | SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON (d.MaDT = h.MaDT) WHERE d.MaSoThue = @masothue | Nhân viên B kiểm tra tính hợp lệ của thời gian truyền vào |
|  | UPDATE HOP\_DONG SET TG\_HieuLucHD =@tg\_hlhd, PhanTramHoaHong = @pthh where MaDT IN (SELECT MaDT FROM DOI\_TAC WHERE MaSoThue = @masothue) | Nhân viên B cập nhật thời gian hiệu lực và phần trăm hoa hồng của hợp đồng đối tác (gia hạn thêm 2 tháng) |
| T1 commit |  | Nhân viên A commit |
|  | T2 commit | Nhân viên B commit  Nhân viên B ghi đè lên dữ liệu cập nhật của Nhân viên A. Kết quả cập nhật của nhân viên A sẽ không bao giờ xảy ra  => Lost update |

**Hướng giải quyết lỗi:** Thiết lập updatelock thay cho shared lock, updatelock được giữ đến hết giao tác => Thêm updatelock vào lúc select của hai giao tác

|  |  |  |
| --- | --- | --- |
| T1 - Nhân viên A | T2 - Nhân viên B | GHI CHÚ |
| SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h **WITH(UPDLOCK)** ON (d.MaDT = h.MaDT) WHERE d.MaSoThue = @masothue |  | Nhân viên A xin khóa updlock  => thành công  Nhân viên A kiểm tra tính hợp lệ của thời gian truyền vào |
| UPDATE HOP\_DONG  SET TG\_HieuLucHD = @tg\_hlhd, PhanTramHoaHong = @pthh where MaDT IN (SELECT MaDT FROM DOI\_TAC WHERE MaSoThue = @masothue) |  | Lúc này khóa updlock sẽ tăng lên thành khóa độc quyền  Nhân viên A cập nhật thời gian hiệu lực và phần trăm hoa hồng của hợp đồng đối tác (gia hạn thêm 3 tháng) |
|  | SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h **WITH (UPDLOCK)** ON (d.MaDT = h.MaDT) WHERE d.MaSoThue = @masothue  => CHỜ | Nhân viên B xin khóa updlock  => chờ do nhân viên A còn đang giữ khóa |
| T1 commit |  | Nhân viên A commit |
|  | SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h **WITH (UPDLOCK)** ON (d.MaDT = h.MaDT) WHERE d.MaSoThue = @masothue | Nhân viên B xin khóa updlock  => thành công  Nhân viên B kiểm tra tính hợp lệ của thời gian truyền vào thì thấy không hợp lệ do A đã gia hạn thêm 3 tháng trước đó mà B chỉ gia hạn thêm 2 tháng |
|  | T2 rollback |  |

* TEST CASE 2:

Mô tả tình huống tranh chấp:

Giả sử 2 nhân viên của đối tác đang cập nhật lại giá của sản phẩm. Nhưng do có sai sót về dữ liệu cung cấp cho 2 nhân viên đó, cả 2 đều cập nhật cùng một sản phẩm có mã sản phẩm là X với 2 đơn giá khác nhau. Lost update xảy ra.

|  |  |  |
| --- | --- | --- |
| T1 - nhân viên A của đối tác cập nhật giá sản phẩm | T2 - nhân viên B của đối tác cập nhật giá sản phẩm | GHI CHÚ |
|  | Read  SELECT \* FROM SAN\_PHAM WHERE MaSP = @masp | Nhân viên B xem thành công giá của sản phẩm |
| Read  SELECT \* FROM SAN\_PHAM WHERE MaSP = @masp |  | Nhân viên A xem thành công giá của sản phẩm |
| Update  UPDATE SAN\_PHAM  SET Gia = @giasp  WHERE MaSP = @masp |  | Nhân viên A cập nhật thành công giá sản phẩm |
|  | Update  UPDATE SAN\_PHAM  SET Gia = @giasp  WHERE MaSP = @masp | Nhân viên B cập nhật thành công giá sản phẩm |
| Commit  COMMIT TRANSACTION |  |  |
|  | Commit  COMMIT TRANSACTION | Nhân viên B thực hiện cập nhật sau nên thay thế cho dữ liệu cập nhật của Nhân viên A của đối tác . Kết quả cập nhật của nhân viên A của đối tác sẽ không bao giờ xảy ra  => Lost update |

**Hướng giải quyết lỗi:** Thiết lập updatelock thay cho shared lock, uplock được giữ đến hết giao tác => Thêm updatelock vào lúc select của hai giao tác

|  |  |  |
| --- | --- | --- |
| T1 - nhân viên A của đối tác cập nhật giá sản phẩm | T2 - nhân viên B của đối tác cập nhật giá sản phẩm | GHI CHÚ |
|  | Read  SELECT \* FROM SAN\_PHAM with(updlock) WHERE MaSP = @masp | Nhân viên B xem thành công giá của sản phẩm |
|  | Update  UPDATE SAN\_PHAM  SET Gia = @giasp  WHERE MaSP = @masp | Nhân viên B cập nhật thành công giá sản phẩm |
|  | Commit  COMMIT TRANSACTION | Nhân viên B commit thành công |
| Read  SELECT \* FROM SAN\_PHAM with(updlock) WHERE MaSP = @masp |  | Nhân viên A xem thành công giá của sản phẩm |
| Update  UPDATE SAN\_PHAM  SET Gia = @giasp  WHERE MaSP = @masp |  | Nhân viên A cập nhật thành công giá sản phẩm |
| Commit  COMMIT TRANSACTION |  | Nhân viên A commit thành công |

* TEST CASE 3:

Mô tả tình huống tranh chấp:

Giả sử đối tác và tài xế đều cập nhật trạng thái đơn hàng cùng một thời điểm. Đối tác thì cập nhật trạng thái đơn hàng là “Chờ xử lý giao hang”, còn tài xế do bị bể bánh xe nên cập nhật đơn hàng là “Đã hủy giao hàng”. Lost update xảy ra.

|  |  |  |
| --- | --- | --- |
| T1 - Đối tác cập nhật tình trạng đơn hàng | T2 – Tài xế cập nhật tình trạng đơn hàng | GHI CHÚ |
| Read  SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = @madh |  | Đối tác xem thành công tình trạng đơn hàng của đơn hàng |
|  | Read  SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = @madh | Tài xế xem thành công tình trạng đơn hàng của đơn hàng |
|  | Update  UPDATE DON\_HANG  SET TinhTrangDH = @ttdh  WHERE MaDH = @madh | Tài xế cập nhật thành công tình trạng đơn hàng của đơn hàng là “Đã hủy giao hàng” |
| Update  UPDATE DON\_HANG  SET TinhTrangDH = @ttdh  WHERE MaDH = @madh |  | Đối tác cập nhật thành công tình trạng đơn hàng của đơn hàng là “Chờ xử lý giao hàng” |
|  | Commit  COMMIT TRANSACTION |  |
| Commit  COMMIT TRANSACTION |  | Đối tác thực hiện cập nhật sau nên thay thế cho dữ liệu cập nhật của Tài xế. Kết quả cập nhật của Tài xế sẽ không bao giờ xảy ra  => Lost update |

**Hướng giải quyết lỗi**: Thiết lập updatelock thay cho shared lock, uplock được giữ đến hết giao tác => Thêm updatelock vào lúc select của hai giao tác

|  |  |  |
| --- | --- | --- |
| T1 - Đối tác cập nhật tình trạng đơn hàng | T2 – Tài xế cập nhật tình trạng đơn hàng | GHI CHÚ |
| Read  SELECT TinhTrangDH FROM DON\_HANG with(updlock) WHERE MaDH = @madh |  | Đối tác xem thành công tình trạng đơn hàng của đơn hàng |
| Update  UPDATE DON\_HANG  SET TinhTrangDH = @ttdh  WHERE MaDH = @madh |  | Đối tác cập nhật thành công tình trạng đơn hàng của đơn hàng là “Chờ xử lý giao hàng” |
| Commit  COMMIT TRANSACTION |  | Đối tác commit thành công |
|  | Read  SELECT TinhTrangDH FROM DON\_HANG with(updlock) WHERE MaDH = @madh | Tài xế xem thành công tình trạng đơn hàng của đơn hàng |
|  | Update  UPDATE DON\_HANG  SET TinhTrangDH = @ttdh  WHERE MaDH = @madh | Tài xế cập nhật thành công tình trạng đơn hàng của đơn hàng là “Đã hủy giao hàng” |
|  | Commit  COMMIT TRANSACTION | Tài xế commit thành công |

### 7.2. Dirty Read

### TEST CASE 1:

Mô tả tình huống tranh chấp:

Giả sử đối tác cập nhật tình trạng đơn hàng là “Đã xử lý đơn hàng và giao hàng cho tài xế”, nhưng do bị mất kết nối mạng nên cập nhật không được thực hiện (hủy cập nhật). Trong lúc đó, khách hàng vào xem tình trạng đơn hàng của mình thì đọc được cập nhật bị hủy. Dirty read xảy ra.

|  |  |  |
| --- | --- | --- |
| T1 – Đối tác cập nhật tình trạng đơn hàng | T2 – Khách hàng xem tình trạng đơn hàng | GHI CHÚ |
| SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED |  | Gán mức cô lập cho giao tác T1 |
| Read  SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = @madh |  | Đối tác xem thành công tình trạng đơn hàng của đơn hàng |
| Update  UPDATE DON\_HANG SET TinhTrangDH = @ttdh WHERE MaDH = @madh |  | Đối tác cập nhật tình trạng đơn hàng của đơn hàng là “Đã xử lý đơn hàng và giao hàng cho tài xế” |
|  | SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED | Gán mức cô lập cho giao tác T2 |
|  | Read  SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = @madh | Khách hàng xem thành công tình trạng đơn hàng |
| Rollback  ROLLBACK TRANSACTION |  | Do gặp trục trặc về mạng nên hủy cập nhật trước đó của đối tác. Khách hàng đọc dữ liệu đã hủy cập nhật của đối tác  => Dirty Read |

**Hướng giải quyết lỗi:** Thiết lập Slock khi đọc và trả khóa ngay sau khi đọc xong ở T2: Bỏ SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

|  |  |  |
| --- | --- | --- |
| T1 – Đối tác cập nhật tình trạng đơn hàng | T2 – Khách hàng xem tình trạng đơn hàng | GHI CHÚ |
| Read  SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = @madh |  | Đối tác xem thành công tình trạng đơn hàng của đơn hàng |
| Update  UPDATE DON\_HANG SET TinhTrangDH = @ttdh WHERE MaDH = @madh |  | Đối tác cập nhật tình trạng đơn hàng của đơn hàng là “Đã xử lý đơn hàng và giao hàng cho tài xế” |
|  | Read  SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = @madh | Khách hàng xem thành công tình trạng đơn hàng |
| Rollback  ROLLBACK TRANSACTION |  |  |

### TEST CASE 2:

Mô tả tình huống tranh chấp:

Giả sử đối tác cập nhật thời gian hiệu lực của hợp đồng của mình, nhưng trong đó lại gặp sự cố mất điện, cập nhật đó bị hủy. Trong thời gian đó thì nhân viên vào xem danh sách hợp đồng của đối tác của mình thì đọc được cập nhật trước đó. Dirty read xảy ra.

|  |  |  |
| --- | --- | --- |
| T1 – Nhân viên xem hợp đồng | T2 – Đối tác cập nhật thời gian hiệu lực của hợp đồng | GHI CHÚ |
|  | SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED | Gán mức cô lập cho T2 |
|  | Read  SELECT TG\_HieuLucHD FROM HOP\_DONG WHERE MaDT = @madt and MaHD = @mahd | Đối tác xem thành công thời gian hiệu lực của hợp đồng |
|  | Update  UPDATE HOP\_DONG SET TG\_HieuLucHD = @tghlhd WHERE MaDT = @madt and MaHD = @mahd | Đối tác cập nhật thành công thời gian hiệu lực của hợp đồng |
| SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED |  | Gán mức cô lập cho T1 |
| Read  SELECT \* FROM HOP\_DONG WHERE MaDT = @madt |  | Nhân viên xem danh sách hợp đồng của đối tác của mình |
|  | Rollback  ROLLBACK TRANSACTION | Do bị mất kết nối mạng nên đối tác hủy cập nhật. Nhân viên đọc dữ liệu đã hủy cập nhật của đối tác  => Dirty Read |

**Hướng giải quyết lỗi:** Thiết lập Slock khi đọc và trả khóa ngay sau khi đọc xong ở T1: Bỏ SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

|  |  |  |
| --- | --- | --- |
| T1 – Đối tác xem hợp đồng | T2 – Nhân viên cập nhật thời gian hiệu lực và phần trăm hoa hồng của hợp đồng đối tác | GHI CHÚ |
|  | Read  SELECT TG\_HieuLucHD, PhanTramHoaHong FROM HOP\_DONG WHERE MaDT = @madt and MaHD = @mahd | Nhân viên xem thành công thời gian hiệu lực và phần trăm hoa hồng của hợp đồng đối tác |
|  | Update  SET @doanhsoban = (SELECT SUM(d.TongPhiSP) FROM DON\_HANG d WHERE d.MaDT = @madt)  UPDATE HOP\_DONG  SET TG\_HieuLucHD = @tghlhd, PhanTramHoaHong = (@pthh \* @doanhsoban) / 100  where MaDT = @madt | Nhân viên cập nhật thành công thời gian hiệu lực và phần trăm hoa hồng của hợp đồng đối tác |
| Read  SELECT \* FROM HOP\_DONG WHERE MaDT = @madt |  | Đối tác xem thành công danh sách hợp đồng của mình |
|  | Rollback  ROLLBACK TRANSACTION |  |

### TEST CASE 3:

Mô tả tình huống tranh chấp:

Giả sử đối tác điều chỉnh giá của một sản phẩm của đối tác đó cung cấp, nhưng sau đó phải hủy cập nhật do mạng gặp trục trặc và bị mất kết nối. Trong lúc đó, khách khàng vào xem chi tiết các sản phẩm của đối tác thì đọc được cập nhật đã hủy. Dirty read xảy ra.

|  |  |  |
| --- | --- | --- |
| T1 – Đối tác cập nhật giá sản phẩm | T2 – Khách hàng xem danh sách các sản phẩm của đối tác | GHI CHÚ |
| SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED |  | Gán mức cô lập cho T1 |
| Read  SELECT \* FROM SAN\_PHAM WHERE MaSP = @masp |  | Đối tác xem thành công sản phẩm dự tính cập nhật |
| Update  UPDATE SAN\_PHAM SET GIA = @gia WHERE MaSP = @masp |  | Đối tác cập nhật thành công giá sản phẩm |
|  | SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED | Gán mức cô lập cho T2 |
|  | Read  SELECT \* FROM SAN\_PHAM sp JOIN CHI\_NHANH cn on sp.MaCN = cn.MaCN and cn.MaDT = @madt | Khách hàng xem thành công danh sách các sản phẩm của đối tác |
| Rollback  ROLLBACK TRANSACTION |  | Đối tác hủy cập nhật. Khách hàng đọc dữ liệu đã hủy của đối tác  => Dirty read |

**Hướng giải quyết lỗi:** Thiết lập Slock khi đọc và trả khóa ngay sau khi đọc xong ở T2: Bỏ SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

|  |  |  |
| --- | --- | --- |
| T1 – Đối tác cập nhật giá sản phẩm | T2 – Khách hàng xem danh sách các sản phẩm của đối tác | GHI CHÚ |
| Read  SELECT \* FROM SAN\_PHAM WHERE MaSP = @masp |  | Đối tác xem thành công sản phẩm dự tính cập nhật |
| Update  UPDATE SAN\_PHAM SET GIA = @gia WHERE MaSP = @masp |  | Đối tác cập nhật thành công giá sản phẩm |
|  | Read  Xin shared lock trên bảng SAN\_PHAM => Thất bại | Khách hàng xem sản phẩm không thành công vì bây giờ mức cô lập đang là read committed. Bảng sản phẩm đang được T1 giữ khóa. |
| Rollback  ROLLBACK TRANSACTION |  |  |
| Unlock bảng SAN\_PHAM |  | Trả khóa cho bản sản phẩm |
|  | SELECT \* FROM SAN\_PHAM sp JOIN CHI\_NHANH cn on sp.MaCN = cn.MaCN and cn.MaDT = @madt | Bây giờ khách hàng xem sản phẩm thành công vì đã xin được khóa đọc trên bảng SAN\_PHAM. |

### 7.3. Unrepeatable Read

### TEST CASE 1:

Mô tả tình huống tranh chấp:

* + Một tài xế đang giao đơn hàng ABC cho khách => Tình trạng đơn hàng hiện tại là “Đang giao hàng”.
  + Nửa tiếng sau, tài xế hoàn thành việc giao hàng đến cho khách hàng nhưng **quên** cập nhật tình trạng đơn hàng là “Đã giao hàng”.
  + Khách hàng sau khi xem hàng thì không vừa ý nên liên hệ đối tác để hoàn trả hàng. Lúc này tình trạng đơn hàng đang là “Đang giao hàng”, đối tác cần cập nhật lại tình trạng đơn hàng thành “Đã hoàn trả hàng” theo yêu cầu của khách. **Giả sử đối tác sơ xuất không để ý rằng tình trạng đơn hàng đang là “Đang giao hàng” nhưng khách lại báo là nhận hàng rồi và muốn hoàn trả.**
  + Khi tài xế nhớ ra mình đã giao hàng rồi nhưng chưa cập nhật là “Đã giao hàng”, vì thế tài xế cần cập nhật từ “Đang giao hàng” thành “Đã giao hàng”.
  + Giả sử trình tự thực hiện thao tác cập nhật của đối tác và tài xế như sau:
    - Tài xế bắt đầu cập nhật
    - Đối tác bắt đầu cập nhật
    - Đối tác hoàn tất cập nhật
    - Tài xế hoàn tất cập nhật
* Việc đối tác hoàn tất cập nhật trong quá trình tài xế đang cập nhật dẫn đến trạng thái đơn hàng lúc này không còn đúng như tài xế đã thấy trước đó.
  + **Giả sử theo quy định:** 
    - Khi tình trạng đơn hàng là “Đã hoàn trả hàng” thì đối tác và tài xế không được cập nhật nữa.
    - Khi tình trạng đơn hàng là “Đã giao hàng” thì đối tác và tài xế chỉ được cập nhật tiếp là “Đã hoàn trả hàng”, ngoài ra không được cập nhật thành bất kỳ trạng thái nào khác.

|  |  |  |
| --- | --- | --- |
| **T1 – Tài xế** | **T2 – Đối tác** | **Ghi Chú** |
| SELECT \* FROM GIAO\_HANG WHERE MaDH = ‘123’ AND MaDT = ‘xyz’ |  | T1 kiểm tra xem mã đơn hàng có khớp với mã đối tác không |
| SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = ‘123’ |  | T1 kiểm tra tình trạng đơn hàng xem có phải là “Đã giao hàng” không, nếu phải thì chỉ được cập nhật là “Đã hoàn trả hàng”  **Tình trạng đơn hàng được select lúc này là: “Đang giao hàng”** |
|  | SELECT \* FROM DON\_HANG WHERE MaDH = ‘123’ AND MaDT = ‘xyz’ | T2 kiểm tra xem mã đơn hàng có khớp với mã đối tác không |
|  | SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = ‘123’ | T2 kiểm tra tình trạng đơn hàng xem có phải là “Đã giao hàng” không, nếu phải thì chỉ được cập nhật là “Đã hoàn trả hàng”  **Tình trạng đơn hàng được select lúc này là: “Đang giao hàng”** |
|  | SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = ‘123’ | T2 kiểm tra tình trạng đơn hàng xem có phải là “Đã hoàn trả hàng” không, nếu phải thì không cập nhật nữa.  **Tình trạng đơn hàng được select lúc này là: “Đang giao hàng”** |
|  | UPDATE DON\_HANG  SET TinhTrangDH = ‘Đã hoàn trả hàng’  WHERE MaDH = ‘123’ | T2 cập nhật tình trạng đơn hàng thành “Đã hoàn trả hàng” |
|  | COMMIT | T2 commit  **Tình trạng đơn hàng sau khi T2 commit là: “Đã hoàn trả hàng”** |
| SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = ‘123’ |  | T1 kiểm tra tình trạng đơn hàng xem có phải là “Đã hoàn trả hàng” không, nếu phải thì không cập nhật nữa  **Tình trạng đơn hàng được select lúc này là: “Đã hoàn trả hàng” (Hoàn toàn khác với câu select trước đó của T1) => ROLLBACK** |
| ROLLBACK |  |  |

**Hướng giải quyết lỗi:** ta đặt khóa UPDLOCK khi đọc dữ liệu trong bảng DON\_HANG lần đầu tiên cho cả tài xế và đối tác để ngăn không cho các thao tác khác can thiệp vào giữa giai đoạn cập nhật làm thay đổi dữ liệu ở bảng DON\_HANG(cụ thể ở đây là tình trạng đơn hàng).

|  |  |  |
| --- | --- | --- |
| **T1 – Tài xế** | **T2 – Đối tác** | **Ghi Chú** |
| SELECT \* FROM GIAO\_HANG WHERE MaDH = ‘123’ AND MaDT = ‘xyz’ |  | T1 kiểm tra xem mã đơn hàng có khớp với mã đối tác không |
| SELECT TinhTrangDH FROM DON\_HANG **WITH(UPDLOCK)** WHERE MaDH = ‘123’ |  | **T1 yêu cầu khóa updlock trên bảng DON\_HANG**  T1 kiểm tra tình trạng đơn hàng xem có phải là “Đã giao hàng” không, nếu phải thì chỉ được cập nhật là “Đã hoàn trả hàng”  **Tình trạng đơn hàng được select lúc này là: “Đang giao hàng”** |
|  | SELECT \* FROM DON\_HANG **WITH(UPDLOCK)** WHERE MaDH = ‘123’ AND MaDT = ‘xyz’  **=> CHỜ** | **T2 yêu cầu khóa updlock trên bảng DON\_HANG nhưng do T1 đang giữ khóa nên T2 phải chờ** |
| SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = ‘123’ |  | T1 kiểm tra tình trạng đơn hàng xem có phải là “Đã hoàn trả hàng” không, nếu phải thì không cập nhật nữa  **Tình trạng đơn hàng được select lúc này vẫn là: “Đang giao hàng”** |
| UPDATE DON\_HANG  SET TinhTrangDH = ‘Đã giao hàng’  WHERE MaDH = ‘123’ |  | T1 cập nhật tình trạng đơn hàng thành “Đã giao hàng” |
| COMMIT |  | **T1 commit và nhả khóa**  **Tình trạng đơn hàng sau khi T1 commit là: “Đã giao hàng”** |
|  | SELECT \* FROM DON\_HANG **WITH(UPDLOCK)** WHERE MaDH = ‘123’ AND MaDT = ‘xyz’ | **T2 yêu cầu khóa updlock trên bảng DON\_HANG => lấy được khóa do T1 đã nhả khóa**  T2 kiểm tra tình trạng đơn hàng xem có phải là “Đã hoàn trả hàng” không, nếu phải thì không cập nhật nữa |
|  | SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = ‘123’ | T2 kiểm tra tình trạng đơn hàng xem có phải là “Đã giao hàng” không, nếu phải thì chỉ được cập nhật là “Đã hoàn trả hàng”  **Tình trạng đơn hàng được select lúc này là: “Đã giao hàng”** |
|  | SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = ‘123’ | T2 kiểm tra tình trạng đơn hàng xem có phải là “Đã hoàn trả hàng” không, nếu phải thì không cập nhật nữa.  **Tình trạng đơn hàng được select lúc này là: “Đã giao hàng”** |
|  | UPDATE DON\_HANG  SET TinhTrangDH = ‘Đã hoàn trả hàng’  WHERE MaDH = ‘123’ | T2 cập nhật tình trạng đơn hàng thành “Đã hoàn trả hàng” |
|  | COMMIT | T2 commit  **Tình trạng đơn hàng sau khi T2 commit là: “Đã hoàn trả hàng”** |

### TEST CASE 2:

Mô tả tình huống tranh chấp:

* + Giám của đối tác ‘abc’ bàn với người đại diện của mình về việc gia hạn hợp đồng, giám đốc nói rằng sẽ tự liên hệ với bên hệ thống đặt và chuyển hàng online để gia hạn hợp đồng thêm 3 tháng. Tuy nhiên do người đại diện không tập trung nên đã nghe nhầm rằng giám đốc yêu cầu mình đi gia hạn hợp đồng thêm 2 tháng.
  + Lúc này giám đốc gọi cho bên hệ thống đặt và chuyển hàng online **yêu cầu nhân viên A gia hạn hợp đồng thêm 3 tháng**. Cùng lúc đó, người đại diện cũng gọi cho bên hệ thống đặt và chuyển hàng online **yêu cầu nhân viên B gia hạn hợp đồng thêm 2 tháng.**

|  |  |  |
| --- | --- | --- |
| **T1 – Nhân viên A** | **T2 – Nhân viên B** | **Ghi Chú** |
| SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON (d.MaDT = h.MaHD) WHERE d.MaSoThue = ‘123’ |  | Kiểm tra xem hợp đồng có tồn tại hay không, nếu không thì rollback  T1 kiểm tra thì thấy select ra được 1 dòng dữ liệu => hợp đồng có tồn tại |
|  | SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON (d.MaDT = h.MaHD) WHERE d.MaSoThue = ‘123’ | Kiểm tra xem hợp đồng có tồn tại hay không, nếu không thì rollback  T2 kiểm tra thì thấy select ra được 1 dòng dữ liệu => hợp đồng có tồn tại |
|  | SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON (d.MaDT = h.MaHD) WHERE d.MaSoThue = ‘123’ | T2 so sánh xem thời gian hiệu lực truyền vào từ input có hợp lệ không bằng cách so với thời gian hiệu lực hiện tại trong cơ sở dữ liệu |
|  | SET @doanhsoban = (SELECT SUM(dh.TongPhiSP) FROM DON\_HANG dh JOIN DOI\_TAC dt ON (dh.MaDT = dt.MaDT) WHERE dt.MaSoThue = ‘123’) | T2 tính toán doanh số bán để cập nhật phần trăm hoa hồng |
|  | UPDATE HOP\_DONG  SET TG\_HieuLucHD = @tg\_hlhd, PhanTramHoaHong = (@pthh \* @doanhsoban) / 100  WHERE MaDT IN (SELECT MaDT FROM DOI\_TAC WHERE MaSoThue = ‘123’) | T2 cập nhật thời gian hiệu lực và phần trăm hoa hồng |
|  | COMMIT | T2 commit |
| SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON (d.MaDT = h.MaHD) WHERE d.MaSoThue = ‘123’ |  | T1 so sánh xem thời gian hiệu lực truyền vào từ input có hợp lệ không bằng cách so với thời gian hiệu lực hiện tại trong cơ sở dữ liệu  **Lúc này, thời gian hiệu lực được select từ cơ sở dữ liệu đã không còn giống với lần select trước đó của T1 do T2 vừa cập nhật thời gian hiệu lực trên cùng một hợp đồng** |
| SET @doanhsoban = (SELECT SUM(dh.TongPhiSP) FROM DON\_HANG dh JOIN DOI\_TAC dt ON (dh.MaDT = dt.MaDT) WHERE dt.MaSoThue = ‘123’) |  | T1 tính toán doanh số bán để cập nhật phần trăm hoa hồng |
| UPDATE HOP\_DONG  SET TG\_HieuLucHD = @tg\_hlhd, PhanTramHoaHong = (@pthh \* @doanhsoban) / 100  WHERE MaDT IN (SELECT MaDT FROM DOI\_TAC WHERE MaSoThue = ‘123’) |  | T1 cập nhật thời gian hiệu lực và phần trăm hoa hồng |
| COMMIT |  | T1 COMMIT |

**Hướng giải quyết lỗi:** ta đặt khóa UPDLOCK khi đọc dữ liệu trong bảng HOP\_DONG lần đầu tiên để ngăn không cho các thao tác khác can thiệp vào giữa giai đoạn cập nhật làm thay đổi dữ liệu ở bảng HOP\_DONG(cụ thể ở đây là thời gian hiệu lực hợp đồng).

|  |  |  |
| --- | --- | --- |
| **T1 – Nhân viên A** | **T2 – Nhân viên B** | **Ghi Chú** |
| SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON **WITH(UPDLOCK)** (d.MaDT = h.MaDT) WHERE d.MaSoThue = ‘123’ |  | **T1 yêu cầu khóa updlock trên bảng HOP\_DONG**  Kiểm tra xem hợp đồng có tồn tại hay không, nếu không thì rollback  T1 kiểm tra thì thấy select ra được 1 dòng dữ liệu => hợp đồng có tồn tại |
|  | SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON **WITH(UPDLOCK)** (d.MaDT = h.MaDT) WHERE d.MaSoThue = ‘123’  **=> CHỜ** | **T2 yêu cầu khóa updlock trên bảng HOP\_DONG nhưng phải chờ do T1 đang giữ khóa** |
| SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON (d.MaDT = h.MaDT) WHERE d.MaSoThue = ‘123’ |  | T1 so sánh xem thời gian hiệu lực truyền vào từ input có hợp lệ không bằng cách so với thời gian hiệu lực hiện tại trong cơ sở dữ liệu  **Lúc này, thời gian hiệu lực được select từ cơ sở dữ liệu vẫn giống với lần select trước đó của T1** |
| SET @doanhsoban = (SELECT SUM(dh.TongPhiSP) FROM DON\_HANG dh JOIN DOI\_TAC dt ON (dh.MaDT = dt.MaDT) WHERE dt.MaSoThue = ‘123’) |  | T1 tính toán doanh số bán để cập nhật phần trăm hoa hồng |
| UPDATE HOP\_DONG  SET TG\_HieuLucHD = @tg\_hlhd, PhanTramHoaHong = (@pthh \* @doanhsoban) / 100  WHERE MaDT IN (SELECT MaDT FROM DOI\_TAC WHERE MaSoThue = ‘123’) |  | T1 cập nhật thời gian hiệu lực và phần trăm hoa hồng |
| COMMIT |  | **T1 commit và nhả khóa** |
|  | SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON **WITH(UPDLOCK)** (d.MaDT = h.MaDT) WHERE d.MaSoThue = ‘123’ | **T2 yêu cầu khóa updlock trên bảng HOP\_DONG => lấy được khóa do T1 vừa nhả khóa**  Kiểm tra xem hợp đồng có tồn tại hay không, nếu không thì rollback  T2 kiểm tra thì thấy select ra được 1 dòng dữ liệu => hợp đồng có tồn tại |
|  | SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON (d.MaDT = h.MaDT) WHERE d.MaSoThue = ‘123’ | T2 so sánh xem thời gian hiệu lực truyền vào từ input có hợp lệ không bằng cách so với thời gian hiệu lực hiện tại trong cơ sở dữ liệu  **=> T2 rollback do T1 vừa cập nhật lại thời gian hiệu lực hợp đồng thêm 3 tháng nhưng T2 lại muốn cập nhật chỉ thêm 2 tháng** |
|  | T2 ROLLBACK |  |

### 7.4. Phantom

### TEST CASE 1:

Mô tả tình huống tranh chấp:

* + Sếp yêu cầu nhân viên A xóa hợp đồng của đối tác “abc” do đã hết hạn quá lâu (>= 3 năm). Tuy nhiên nhân viên A chưa làm ngay.
  + Sau đó khoảng 1 tiếng, nhân viên A lúc này bắt đầu xóa hợp đồng mà sếp giao khi nãy. Đồng thời, nhân viên B khi truy cập vào cơ sở dữ liệu thì thấy hợp đồng của đối tác “abc” quá hạn đã hơn 3 năm nên nhân viên B tự ý xóa hợp đồng của đối tác “abc” mà không báo cho sếp.

|  |  |  |
| --- | --- | --- |
| **T1 – Nhân viên A** | **T2 – Nhân viên B** | **Ghi Chú** |
| SELECT h.MaHD FROM HOP\_DONG h WHERE MaDT = ‘abc’ |  | Kiểm tra xem hợp đồng có tồn tại hay không, nếu không thì rollback  T1 kiểm tra thì thấy select ra được 1 dòng dữ liệu => hợp đồng có tồn tại |
|  | SELECT h.MaHD FROM HOP\_DONG h WHERE MaDT = ‘abc’ | Kiểm tra xem hợp đồng có tồn tại hay không, nếu không thì rollback  T2 kiểm tra thì thấy select ra được 1 dòng dữ liệu => hợp đồng có tồn tại |
|  | SET @mahd = (SELECT h.MaHD FROM HOP\_DONG h WHERE MaDT = ‘abc’) | T2 select để lấy mã hợp đồng ứng với mã đối tác được truyền vào |
|  | DELETE FROM CT\_HOPDONG  WHERE MaHD = @mahd | T2 xóa các dòng dữ liệu ứng với mã hợp đồng vừa select ở trên trong bảng chi tiết hợp đồng |
|  | DELETE FROM HOP\_DONG  WHERE MaHD = @mahd | T2 xóa các dòng dữ liệu ứng với mã hợp đồng vừa select ở trên trong bảng hợp đồng |
|  | COMMIT | T2 commit |
| SET @mahd = (SELECT h.MaHD FROM HOP\_DONG h WHERE MaDT = ‘abc’) |  | T1 select để lấy mã hợp đồng ứng với mã đối tác được truyền vào **nhưng không thấy một dòng dữ liệu nào cả do T2 vừa xóa hợp đồng trước đó** |
| DELETE FROM CT\_HOPDONG  WHERE MaHD = @mahd |  | T1 xóa các dòng dữ liệu ứng với mã hợp đồng vừa select ở trên trong bảng chi tiết hợp đồng **nhưng không có tác dụng gì do hợp đồng đã được xóa bởi T2 trước đó** |
| DELETE FROM HOP\_DONG  WHERE MaHD = @mahd |  | T1 xóa các dòng dữ liệu ứng với mã hợp đồng vừa select ở trên trong bảng hợp đồng **nhưng không có tác dụng gì do hợp đồng đã được xóa bởi T2 trước đó** |
| COMMIT |  | T1 commit |

**Hướng giải quyết lỗi:** ta đặt khóa UPDLOCK khi đọc dữ liệu trong bảng HOP\_DONG lần đầu tiên để ngăn không cho các thao tác khác can thiệp vào giữa chừng làm thay đổi dữ liệu ở bảng HOP\_DONG (cụ thể ở đây là ngăn thao tác xóa của T2 can thiệp vào giữa giai đoạn xóa của T1 làm cho 2 lần SELECT ra số dòng dữ liệu không giống nhau).

|  |  |  |
| --- | --- | --- |
| **T1 – Nhân viên A** | **T2 – Nhân viên B** | **Ghi Chú** |
| SELECT h.MaHD FROM HOP\_DONG h **WITH(UPDLOCK)** WHERE MaDT = ‘abc’ |  | **T1 yêu cầu khóa updlock trên bảng HOP\_DONG**  Kiểm tra xem hợp đồng có tồn tại hay không, nếu không thì rollback  T1 kiểm tra thì thấy select ra được 1 dòng dữ liệu => hợp đồng có tồn tại |
|  | SELECT h.MaHD FROM HOP\_DONG h **WITH(UPDLOCK)** WHERE MaDT = ‘abc’  **=> CHỜ** | **T2 yêu cầu khóa updlock trên bảng HOP\_DONG nhưng phải chờ do T1 đang giữ khóa** |
| SET @mahd = (SELECT h.MaHD FROM HOP\_DONG h WHERE MaDT = ‘abc’) |  | T1 select để lấy mã hợp đồng ứng với mã đối tác được truyền vào |
| DELETE FROM CT\_HOPDONG  WHERE MaHD = @mahd |  | T1 xóa các dòng dữ liệu ứng với mã hợp đồng vừa select ở trên trong bảng chi tiết hợp đồng |
| DELETE FROM HOP\_DONG  WHERE MaHD = @mahd |  | T1 xóa các dòng dữ liệu ứng với mã hợp đồng vừa select ở trên trong bảng hợp đồng |
| COMMIT |  | **T1 commit và nhả khóa** |
|  | SELECT h.MaHD FROM HOP\_DONG h **WITH(UPDLOCK)** WHERE MaDT = ‘abc’ | **T2 yêu cầu khóa updlock trên bảng HOP\_DONG thành công do T1 vừa nhả khóa trước đó**  Kiểm tra xem hợp đồng có tồn tại hay không, nếu không thì rollback  T2 kiểm tra thì thấy select ra được số dòng dữ liệu là 0 => hợp đồng không tồn tại do T1 đã xóa trước đó => T2 rollback |
|  | T2 ROLLBACK |  |

### 7.5. Deadlock

### 7.5.1. Converson Deadlock:

### TEST CASE 1:

Mô tả tình huống tranh chấp:

* Giả sử tài xế A hoạt động ở khu vực quận 5, TPHCM.
* Giả sử tình trạng đơn hàng X đang là “Nhập kho quận 5, TPHCM”, đối tác lên kế hoạch sẽ giao đơn hàng này cho tài xế A, tuy nhiên đối tác quên cập nhật tình trạng đơn hàng thành “Đã bàn giao cho tài xế”. Sau 1 tiếng, đối tác nhớ ra và bắt đầu cập nhật.
* Tuy nhiên cùng lúc đó, tài xế A truy cập vào và vẫn thấy đơn hàng X trong danh sách đơn hàng theo khu vực quận 5 do đối tác chưa cập nhật xong. Lúc này tài xế nghĩ rằng đối tác đã quên cập nhật nên bấm nút chấp nhận đơn, đơn hàng sau khi được chấp nhận sẽ tự động cập nhật thành “Đang giao hàng”.

|  |  |  |
| --- | --- | --- |
| T1 \_ Đối tác | T2 \_ Tài xế | Ghi Chú |
| Shared\_Lock(DON\_HANG) |  | Đối tác thành công xin khóa đọc trên bảng đơn hàng vì bảng đó chưa có khóa nào chiếm. |
| IF EXISTS (Select \* From DON\_HANG Where MaDT = @madt AND MaDH = @madh); |  | Đối tác đọc trên bảng đơn hàng thành công vì đã có đã chiếm được shared lock trước đó. |
|  | Shared\_lock(DON\_HANG) | Tài xế thành công xin khóa đọc trên bảng đơn hàng. Vì bảng đơn hàng đã bị chiếm với khóa shared lock nên có thể chiếm giữ khóa S cùng lúc được. |
|  | Select TenDuong From DON\_HANG Where MaDH = @madh | Tài xế đọc đơn hàng thành công vì đã chiếm khóa đọc trước đó. |
|  | Exclusive\_Lock(DON\_HANG)  UPDATE DON\_HANG (TinhTrangDH)  SET TinhTrangDH = “Tài xế đã nhận đơn hàng”  WHERE MaDH = @madh | Tài xế không thể xin được khóa ghi vì bảng đơn hàng đã bị chiếm khóa đọc trước bởi T1 và cả T2 trước đó. Mà do 2 transaction đều đang ở mức cô lập Serializable nên hai khóa đều được giữ đến cuối giao tác. Lúc này giao tác 2 chưa hoàn thành nên chưa trả khóa đọc. |
| Exclusive\_Lock(DON\_HANG)  UPDATE DON\_HANG (TinhTrangDH)  SET TinhTrangDH = “Đã xuất kho Mê Linh Hà Nội”  WHERE MaDH = @madh |  | Đối tác cũng không thể cập nhật được tình trạng đơn hàng vì không xin được khóa ghi do bảng đơn hàng đã bị chiếm bởi khóa đọc. |

**Hướng giải quyết lỗi**: Thay vì chiếm giữ share lock trước khi đọc của T1 và T2 ta sẽ nâng cấp khóa thành chiếm giữ update lock trước khi đọc T1 và T2 vì biết trước sẽ ghi trong tương lai.

|  |  |  |
| --- | --- | --- |
| T1 \_ Đối tác | T2 \_ Tài xế | Ghi Chú |
| Update\_Lock(DON\_HANG) |  | Đối tác thành công xin khóa update trên bảng đơn hàng vì bảng đó chưa có khóa nào chiếm. |
| IF EXISTS (Select \* From DON\_HANG Where MaDT = @madt AND MaDH = @madh); |  | Đối tác đọc trên bảng đơn hàng thành công vì đã có đã chiếm được update lock trước đó. |
|  | Update\_Lock(DON\_HANG) | Tài xế xin khóa update trên Đơn hàng không thành công vì T1 đã giữ khóa update trước đó, nên T2 sẽ chờ cho đến khi xin được |
| Exclusive\_Lock(DON\_HANG)  UPDATE DON\_HANG (TinhTrangDH)  SET TinhTrangDH = “Đã xuất kho Mê Linh Hà Nội”  WHERE MaDH = @madh |  | Đối tác có thể cập nhật được tình trạng đơn hàng vì đã nâng cấp từ khóa update lên khóa ghi nên có thể cập nhật được tình trạng đơn hàng. |
| UNLOCK (DON\_HANG) |  | Đối tác kết thúc tác vụ và trả lại khóa trên bảng đơn hàng cho hệ thống |
|  | Update\_Lock(DON\_HANG)  Select TenDuong From DON\_HANG Where MaDH = @madh | Tài xế xin thành công khóa update trên bảng đơn hàng vì lúc này bảng đơn hàng chưa bị giao tác nào chiếm khóa nên tài xế đã đọc đơn hàng thành công. |
|  | Exclusive\_Lock(DON\_HANG)  UPDATE DON\_HANG (TinhTrangDH)  SET TinhTrangDH = “Tài xế đã nhận đơn hàng”  WHERE MaDH = @madh | Lúc này tài xế nâng cấp từ khóa update lên khóa độc quyền. Do đã nâng cấp thành công lên khóa độc quyền nên tài xế đã cập nhật tình trạng đơn hàng thành công. |
|  | UNLOCK(DON\_HANG) | Tài xế hoàn thành tác vụ của mình vả trả lại khóa cho bảng đơn hàng. |

=> Ở đây T1 và T2 đã được quy về chạy tuần tự, tức là T1 hoàn thành, trả khóa rồi T2 mới chạy nên đã giải quyết được vấn đề conversation deadlock.

### 7.5.2. Cycle Deadlock:

### TEST CASE 2:

Mô tả tình huống tranh chấp:

Giả sử khi tài xế chọn một đơn hàng để giao, tài xế sẽ thêm đơn hàng đó vào bảng giao hàng đồng thời tài xế cũng sẽ cập nhật tình trạng đơn hàng là “Đã nhận hàng”. Cùng lúc đó thì đối tác cũng cập nhật tình trạng đơn hàng là “Đã xuất kho” đồng thời cũng đọc bảng giao hàng để biết được tài xế nào đang giao đơn hàng đó.

|  |  |  |
| --- | --- | --- |
| T1\_ tài xế | T2\_đối tác | Ghi chú |
| Exclusive\_Lock(GIAO\_HANG) |  | Tài xế xin khóa ghi thành công vì bảng giao hàng chưa bị chiếm bởi khóa nào. |
| INSERT INTO GIAO\_HANG (MaDH, MaTX)  VALUES (@madh,@matx) |  | Tài xế thêm thành công đơn hàng mình chọn vào bảng giao hàng vì đã chiếm được khóa ghi trước đó |
|  | Exclusive\_Lock(DON\_HANG) | Đối tác chiếm thành công khóa đọc trên bảng đơn hàng vì bảng đơn hàng chưa bị chiếm bởi khóa nào. |
|  | UPDATE DON\_HANG  SET TinhTrangDH = ‘Đã xuất kho’  WHERE MaDH = @madh | Đối tác cập nhật thành công tình trạng đơn hàng vì đã chiếm được khóa độc quyền trước đó. |
|  | Shared\_Lock(GIAO\_HANG)  SELECT MaTX  FROM GIAO\_HANG  WHERE MaDH = @madh | Đối tác xin khóa chia sẽ trên bảng Giao\_Hàng không thành công vì bảng giao hàng đã bị T1 chiếm khóa độc quyền trước đó và T1 vẫn chưa trả khóa do giao tác T1 vẫn chưa hoàn thành xong |
| Exclusive\_Lock(DON\_HANG)  UPDATE DON\_HANG  SET TinhTrangDH = “Đã nhận đơn”  WHERE MaDH = @madh |  | Tài xế xin khóa độc quyền trên bảng đơn hàng thất bại vì bảng đơn hàng đã bị chiếm khóa đọc quyền bởi T2 và do T2 chưa hoàn thành nên T2 vẫn chưa trả khóa độc quyền trên bàng đơn hàng |

**Hướng giải quyết lỗi:** Để giải quyết vấn đề cycle deadlock trong thực tế là một vấn đề vô cùng khó khăn và tiêu tốn rất nhiều chi phí. Nên khi gặp cycle deadlock thì một trong những phương pháp được áp dụng phổ biến mà tiết kiệm chi phí là ta sẽ cho một giao tác chờ một khoản thời gian trước khi bắt đầu để đợi các giao tác khác hoàn thành xong và trả về hết các khóa cần thiết.

|  |  |  |
| --- | --- | --- |
| T1\_ tài xế | T2\_đối tác | Ghi chú |
| Wait for delay ‘0:00:10’ |  | Cho giao tác của tài xế đợi 10 giây trước khi bắt đầu thực hiện. |
|  | Exclusive\_Lock(DON\_HANG) | Đối tác chiếm thành công khóa đọc trên bảng đơn hàng vì bảng đơn hàng chưa bị chiếm bởi khóa nào. |
|  | UPDATE DON\_HANG  SET TinhTrangDH = ‘Đã xuất kho’  WHERE MaDH = @madh | Đối tác cập nhật thành công tình trạng đơn hàng vì đã chiếm được khóa độc quyền trước đó. |
|  | Shared\_Lock(GIAO\_HANG)  SELECT MaTX  FROM GIAO\_HANG  WHERE MaDH = @madh | Đối tác xin khóa chia sẽ trên bảng Giao\_Hàng thành công vì bảng giao hàng chưa bị chiếm khóa gì trước đó. |
|  | UnLock (ĐON\_HANG)  UnLock (GIAO\_HANG) | Trước khi kết thúc giao tác của đối tác thì nó sẽ trả lại khóa trên bảng đơn hàng và giao hàng đã chiếm giữ trước đó. |
| Exclusive\_Lock(GIAO\_HANG) |  | Sau khi hết thời gian 10 giây thì T1 bắt đầu chạy.  Tài xế xin khóa ghi thành công vì bảng giao hàng đã được trả khóa trước đó. |
| INSERT INTO GIAO\_HANG (MaDH, MaTX)  VALUES (@madh,@matx) |  | Tài xế thêm thành công đơn hàng mình chọn vào bảng giao hàng vì đã chiếm được khóa ghi trước đó |
| Exclusive\_Lock(DON\_HANG)  UPDATE DON\_HANG  SET TinhTrangDH = “Đã nhận đơn”  WHERE MaDH = @madh |  | Tài xế xin khóa độc quyền trên bảng đơn hàng thành công vì bảng giao hàng đã được T2 trả khóa. Cập nhật đơn hàng thành công. |
| Unlock GIAO\_HANG  Unlock DON\_HANG |  | Sau khi kết thúc giao tác, T1 sẽ trao trả lại tất cả khóa nó chiếm giữ cho hệ thống. |