**Nhóm: 18VP – 07**

**Thành viên:**

1. **18126023 – Đặng Diễm Linh**
2. 18126019 – Nguyễn Trọng Khang
3. 18126038 – Dương Quang Vinh

**BÁO CÁO ĐỒ ÁN MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: QUẢN LÝ HỆ THỐNG CHO THUÊ / BÁN NHÀ**

NỘI DUNG BÁO CÁO

[**BẢNG THIẾT KẾ ER**  3](#_Toc68345808)

[**LƯỢC ĐỒ QUAN HỆ** 4](#_Toc68345809)

[**CÁC RÀNG BUỘC TOÀN VẸN** 6](#_Toc68345810)

[**CÀI ĐẶT CƠ SỞ DỮ LIỆU TRONG SQL SERVER** 7](#_Toc68345811)

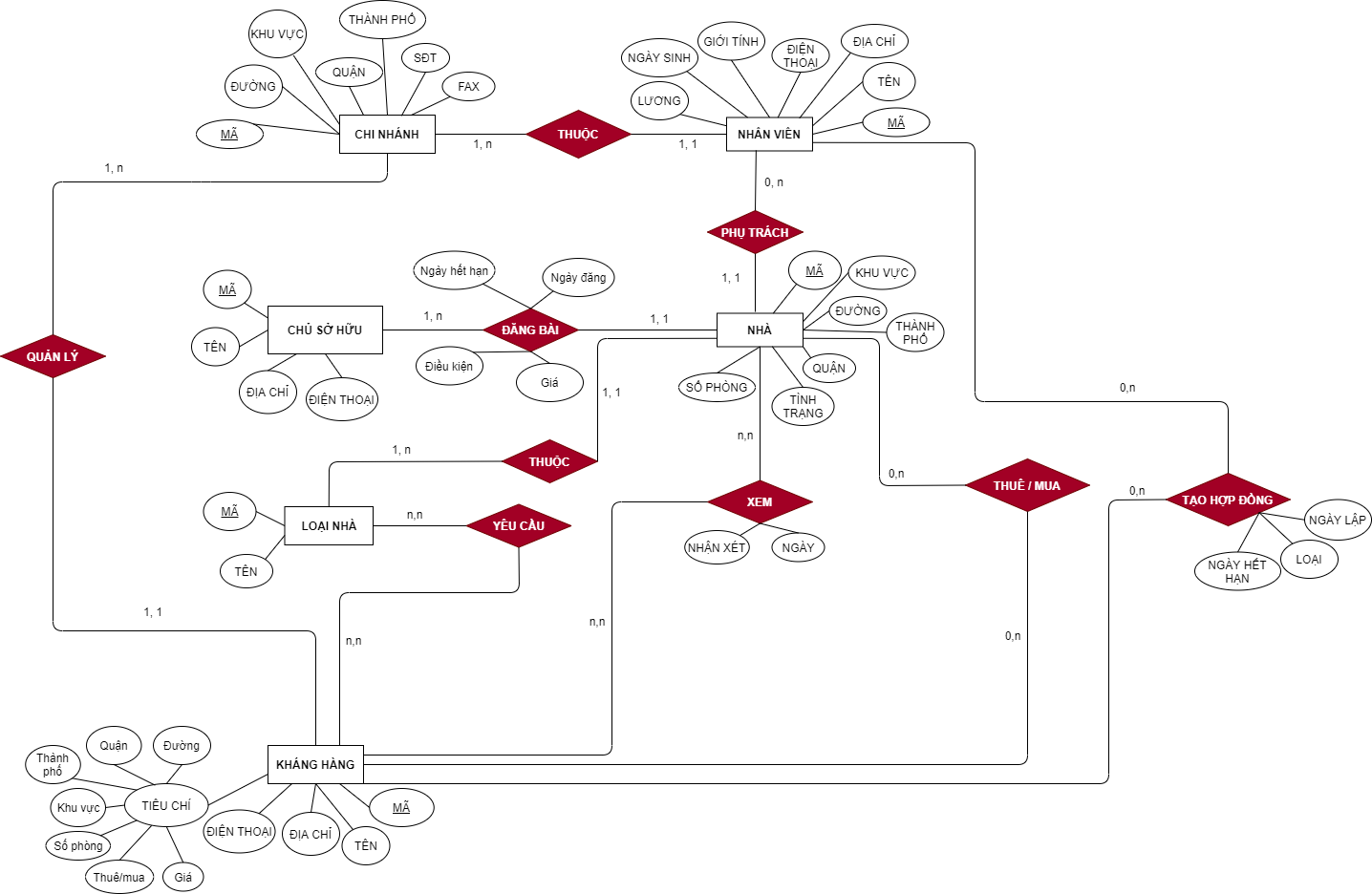
[**XÁC ĐỊNH LOẠI NGƯỜI DÙNG** 15](#_Toc68345812)

[**CÁC CHỨC NĂNG HỆ THỐNG** 16](#_Toc68345813)

[**CÁC CHỨC NĂNG TRANH CHẤP ĐỒNG THỜI – LÝ THUYẾT** 19](#_Toc68345814)

[**XỬ LÝ CÁC TÌNH HUỐNG TRANH CHẤP ĐỒNG THỜI – LÝ THUYẾT** 27](#_Toc68345815)

# **BẢNG THIẾT KẾ ER** *(xem rõ hơn ở file ER\_Diagramme.png đính kèm trong file zip****)***



# **LƯỢC ĐỒ QUAN HỆ**

**CHI NHÁNH**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mã | Đường | Khu vực | Quận | Thành phố | SĐT | Fax |

**NHÂN VIÊN**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Mã | Tên | Giới tính | Địa chỉ | Điện thoại | Ngày sinh | Lương | Mã Chi nhánh |

**CHỦ SỞ HỮU**

|  |  |  |  |
| --- | --- | --- | --- |
| Mã | Tên | Địa chỉ | Điện thoại |

**NHÀ**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mã | Khu vực | Đường | Quận | Thành phố | Số phòng | Mã loại | Mã NV | Mã chủ nhà | Ngày đăng | Ngày hết hạn | Điều kiện | Giá | Tình trạng | Lượt xem |

**LOẠI NHÀ**

|  |  |
| --- | --- |
| Mã | Tên |

**KHÁCH HÀNG**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mã | Tên | Địa chỉ | Điện thoại | Tiêu chí\_số phòng | Tiêu chí\_giá min | Tiêu chí\_giá max | Tiêu chí\_ khu vực | Tiêu chí \_ đường | Tiêu chí \_ quận | Tiêu chí\_ thành phố | Thuê/Mua | Đã liên hệ | Mã Chi nhánh |

**CHI\_TIẾT HỢP ĐỒNG**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mã Nhà | Mã NV | Mã KH | Loại hợp đồng | Ngày lập | Ngày hết hạn |

**KHÁCH HÀNG LOẠI NHÀ NHÀ**

**CHI TIẾT \_ YÊU CẦU**

|  |  |
| --- | --- |
| Mã\_KH | Mã loại nhà |

**CHI TIẾT\_XEM NHÀ**

|  |  |  |  |
| --- | --- | --- | --- |
| Mã KH | Mã nhà | Ngày | Nhận xét |

# **CÁC RÀNG BUỘC TOÀN VẸN**

1. **BẢNG NHÀ:** Một nhà có thể được cho thuê hoặc bán. Nếu là nhà cho thuê, phần **ĐIỀU KIỆN** để trống, giá thuê lớn hơn 0. Nếu là nhà bán, phần **Điều kiện** là điều kiện bán của chủ nhà, giá bán phải từ 500.000.000 trở lên.
2. **BẢNG NHÀ:** Ngày hết hạn bài đăng phải sau Ngày đăng bài.
3. **BẢNG NHÀ:** Lượt xem phải là số nguyên không âm.
4. **BẢNG NHÀ:** Tình trạng chỉ gồm: [ĐÃ CHO THUÊ; ĐÃ BÁN; CÒN TRỐNG]
5. **BẢNG NHÀ:** Số phòng phải là số nguyên dương.
6. **BẢNG NHÂN VIÊN:** Nhân viên phải đủ 18 tuổi.
7. **BẢNG NHÂN VIÊN:** Lương nhân viên phải từ 1.000.000 trở lên.
8. **BẢNG NHÂN VIÊN:** Giới tính nhận hai giá trị: 0 nếu là NAM và 1 nếu là NỮ
9. **BẢNG CHI\_TIẾT HỢP ĐỒNG:** Loại hợp đồng nhận 2 giá trị: 1 – Thuê, 0 - Bán
10. **BẢNG CHI\_TIẾT HỢP ĐỒNG:** Để trống **NGÀY HẾT HẠN HỢP ĐỒNG** nếu loại hợp đồng là **BÁN**.
11. **BẢNG KHÁCH HÀNG:** Thuê/Mua nhận giá trị tương ứng: 1 – Thuê, 0 - Mua
12. **BẢNG KHÁCH HÀNG:** 2 giá trị tương ứng: 1 – Đã liên hệ thành công và sẽ xem nhà ; 0 – Chưa liên hệ thành công
13. **BẢNG KHÁCH HÀNG:** tiêu chí giá cao nhất phải lớn hơn hoặc bằng tiêu chí giá thấp nhất.
14. **BẢNG CHI\_TIẾT HỢP ĐỒNG:** Ngày hết hạn hợp đồng phải sau Ngày lập hợp đồng
15. **BẢNG CHI\_TIẾT HỢP ĐỒNG:** Nhà đang cho thuê, chưa hết hạn hợp đồng thì không được thêm hợp đồng bán/cho thuê mới.
16. **BẢNG CHI\_TIẾT HỢP ĐỒNG và BẢNG NHÀ:** Ngày lập hợp đồng phải sau Ngày đăng bài.
17. **BẢNG NHÀ VÀ BẢNG CHI\_TIẾT XEM\_NHÀ:** Ngày xem nhà phải sau Ngày đăng bài.

# **CÀI ĐẶT CƠ SỞ DỮ LIỆU TRONG SQL SERVER**

*(các đoạn script chính đi kèm với các ràng buộc sẽ được giới thiệu trong report. Để xem đầy đủ script, hãy truy cập vào file* ***18VP\_07\_NHADAT\_Script*** *đính kèm trong file nén)*

1. **Tạo bảng CHI NHÁNH**

CREATE TABLE [dbo].[ChiNhanh](

[ID] [int] NOT NULL,

[Duong] [nvarchar](50) NULL,

[Khu\_Vuc] [nvarchar](50) NULL,

[Thanh\_Pho] [nvarchar](50) NULL,

[SDT] [char](8) NULL,

[Fax] [char](8) NULL,

CONSTRAINT [PK\_ChiNhanh] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

1. **Tạo bảng NHÀ**

CREATE TABLE [dbo].[Nha](

[ID] [int] NOT NULL,

[Khu\_Vuc] [nvarchar](50) NULL,

[Duong] [nvarchar](50) NULL,

[Quan] [nvarchar](50) NULL,

[Thanh\_Pho] [nvarchar](50) NULL,

[So\_Phong] [int] NULL,

[Ma\_Loai] [int] NULL,

[Ma\_NV] [int] NULL,

[Ma\_CSH] [int] NULL,

[Ngay\_Dang] [date] NULL,

[Ngay\_het\_han] [date] NULL,

[Dieu\_Kien] [nvarchar](max) NULL,

[Gia] [int] NULL,

[Tinh\_Trang] [nvarchar](20) NULL,

[Luot\_Xem] [int] NULL,

CONSTRAINT [PK\_Nha] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

1. **Tạo bảng LOẠI NHÀ**

CREATE TABLE [dbo].[LoaiNha](

[ID] [int] NOT NULL,

[Ten] [nvarchar](50) NULL,

CONSTRAINT [PK\_LoaiNha] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

1. **Tạo bảng NHÂN VIÊN**

CREATE TABLE [dbo].[NhanVien](

[ID] [int] NOT NULL,

[Ho\_Ten] [nvarchar](50) NULL,

[Gioi\_Tinh] [bit] NULL,

[Dia\_Chi] [nvarchar](50) NULL,

[SDT] [char](10) NULL,

[Ngay\_Sinh] [date] NULL,

[Luong] [int] NULL,

[Ma\_Chi\_Nhanh] [int] NULL,

CONSTRAINT [PK\_NhanVien] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

1. **Tạo bảng CHỦ SỞ HỮU**

CREATE TABLE [dbo].[ChuSoHuu](

[ID] [int] NOT NULL,

[Ho\_Ten] [nvarchar](50) NULL,

[Dia\_Chi] [nvarchar](50) NULL,

[SDT] [char](10) NULL,

CONSTRAINT [PK\_ChuSoHuu] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

1. **Tạo bảng KHÁCH HÀNG**

CREATE TABLE [dbo].[KhachHang](

[ID] [int] NOT NULL,

[Ho\_Ten] [nvarchar](50) NULL,

[Dia\_Chi] [nvarchar](50) NULL,

[Dien\_Thoai] [char](10) NULL,

[TieuChi\_SoPhong] [int] NULL,

[TieuChi\_Gia\_Min] [int] NULL,

[TieuChi\_Gia\_Max] [int] NULL,

[TieuChi\_KhuVuc] [nvarchar](50) NULL,

[TieuChi\_Duong] [nvarchar](50) NULL,

[TieuChi\_Quan] [nvarchar](50) NULL,

[TieuChi\_ThanhPhi] [nvarchar](50) NULL,

[Thue\_Mua] [bit] NULL,

[Da\_Lien\_He] [bit] NULL,

[Ma\_Chi\_Nhanh] [int] NULL,

CONSTRAINT [PK\_KhachHang] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

1. **Bảng CHI TIẾT YÊU CẦU**

CREATE TABLE [dbo].[ChiTiet\_YeuCau](

[Ma\_KH] [int] NOT NULL,

[Ma\_LoaiNha] [int] NOT NULL,

CONSTRAINT [PK\_ChiTiet\_YeuCau] PRIMARY KEY CLUSTERED

(

[Ma\_KH] ASC,

[Ma\_LoaiNha] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

1. **Bảng CHI TIẾT HỢP ĐỒNG**

CREATE TABLE [dbo].[ChiTiet\_HopDong](

[Ma\_Nha] [int] NOT NULL,

[Ma\_NhanVien] [int] NOT NULL,

[Ma\_KhachHang] [int] NOT NULL,

[Loai\_HopDong] [bit] NULL,

[NgayLap] [date] NOT NULL DEFAULT(GETDATE()),

[NgayHetHan] [date] NULL,

CONSTRAINT [PK\_ChiTiet\_HopDong] PRIMARY KEY CLUSTERED

(

[Ma\_Nha] ASC,

[Ma\_NhanVien] ASC,

[Ma\_KhachHang] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

1. **Bảng CHI TIẾT XEM NHÀ**

CREATE TABLE [dbo].[ChiTiet\_XemNha](

[Ma\_KH] [int] NOT NULL,

[Ma\_Nha] [int] NOT NULL,

[Ngay] [date] NULL,

[NhanXet] [text] NULL,

CONSTRAINT [PK\_ChiTiet\_XemNha] PRIMARY KEY CLUSTERED

(

[Ma\_KH] ASC,

[Ma\_Nha] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

1. **CÁC LOẠI RÀNG BUỘC**

* ***RÀNG BUỘC KHÓA NGOẠI***

1. ALTER TABLE [dbo].[ChiTiet\_HopDong] WITH CHECK ADD CONSTRAINT [FK\_ChiTiet\_HopDong\_KhachHang] FOREIGN KEY([Ma\_KhachHang]) REFERENCES [dbo].[KhachHang] ([ID])
2. ALTER TABLE [dbo].[ChiTiet\_HopDong] WITH CHECK ADD CONSTRAINT [FK\_ChiTiet\_HopDong\_Nha] FOREIGN KEY([Ma\_Nha]) REFERENCES [dbo].[Nha] ([ID])
3. ALTER TABLE [dbo].[ChiTiet\_HopDong] WITH CHECK ADD CONSTRAINT [FK\_ChiTiet\_HopDong\_NhanVien] FOREIGN KEY([Ma\_NhanVien]) REFERENCES [dbo].[NhanVien] ([ID])
4. ALTER TABLE [dbo].[ChiTiet\_XemNha] WITH CHECK ADD CONSTRAINT [FK\_ChiTiet\_XemNha\_KhachHang] FOREIGN KEY([Ma\_KH]) REFERENCES [dbo].[KhachHang] ([ID])
5. ALTER TABLE [dbo].[ChiTiet\_XemNha] WITH CHECK ADD CONSTRAINT [FK\_ChiTiet\_XemNha\_Nha] FOREIGN KEY([Ma\_Nha]) REFERENCES [dbo].[Nha] ([ID])
6. ALTER TABLE [dbo].[ChiTiet\_YeuCau] WITH CHECK ADD CONSTRAINT [FK\_ChiTiet\_YeuCau\_KhachHang] FOREIGN KEY([Ma\_KH]) REFERENCES [dbo].[KhachHang] ([ID])
7. ALTER TABLE [dbo].[ChiTiet\_YeuCau] WITH CHECK ADD CONSTRAINT [FK\_ChiTiet\_YeuCau\_LoaiNha] FOREIGN KEY([Ma\_LoaiNha]) REFERENCES [dbo].[LoaiNha] ([ID])
8. ALTER TABLE [dbo].[KhachHang] WITH CHECK ADD CONSTRAINT [FK\_KhachHang\_ChiNhanh] FOREIGN KEY([Ma\_Chi\_Nhanh]) REFERENCES [dbo].[ChiNhanh] ([ID])
9. ALTER TABLE [dbo].[Nha] WITH CHECK ADD CONSTRAINT [FK\_Nha\_ChuSoHuu] FOREIGN KEY([Ma\_NV]) REFERENCES [dbo].[ChuSoHuu] ([ID])
10. ALTER TABLE [dbo].[Nha] WITH CHECK ADD CONSTRAINT [FK\_Nha\_LoaiNha] FOREIGN KEY([Ma\_Loai]) REFERENCES [dbo].[LoaiNha] ([ID])
11. ALTER TABLE [dbo].[Nha] WITH CHECK ADD CONSTRAINT [FK\_Nha\_NhanVien] FOREIGN KEY([Ma\_NV]) REFERENCES [dbo].[NhanVien] ([ID])
12. ALTER TABLE [dbo].[NhanVien] WITH CHECK ADD CONSTRAINT [FK\_NhanVien\_ChiNhanh] FOREIGN KEY([Ma\_Chi\_Nhanh]) REFERENCES [dbo].[ChiNhanh] ([ID])

* ***RÀNG BUỘC TOÀN VẸN***

1) **BẢNG NHÀ:** Một nhà có thể được cho thuê hoặc bán. Nếu là nhà cho thuê, phần **ĐIỀU KIỆN** để trống, giá thuê lớn hơn 0. Nếu là nhà bán, phần **Điều kiện** là điều kiện bán của chủ nhà, giá bán phải từ 500.000.000 trở lên.

ALTER TABLE [dbo].[Nha] WITH CHECK ADD CONSTRAINT [CK\_Nha\_4] CHECK (([Dieu\_Kien] IS NULL AND [Gia]>(0) OR [Dieu\_Kien] IS NOT NULL AND [Gia]>(500000000)))

2) **BẢNG NHÀ:** Ngày hết hạn bài đăng phải sau Ngày đăng bài.

ALTER TABLE [dbo].[Nha] WITH CHECK ADD CONSTRAINT [CK\_Nha] CHECK (([Ngay\_het\_han]>[Ngay\_Dang]))

3) **BẢNG NHÀ:** Lượt xem phải là số nguyên không âm.

ALTER TABLE [dbo].[Nha] WITH CHECK ADD CONSTRAINT [CK\_Nha\_2] CHECK (([Luot\_Xem]>=(0)))

4) **BẢNG NHÀ:** Tình trạng chỉ gồm: [ĐÃ CHO THUÊ; ĐÃ BÁN; CÒN TRỐNG]

ALTER TABLE [dbo].[Nha] WITH CHECK ADD CONSTRAINT [CK\_Nha\_3] CHECK (([Tinh\_Trang]=N'ĐÃ CHO THUÊ' OR [Tinh\_Trang]=N'ĐÃ BÁN' OR [Tinh\_Trang]=N'CÒN TRỐNG'))

5) **BẢNG NHÀ:** Số phòng phải là số nguyên dương.

ALTER TABLE [dbo].[Nha] WITH CHECK ADD CONSTRAINT [CK\_Nha\_1] CHECK (([So\_Phong]>(0)))

6) **BẢNG NHÂN VIÊN:** Nhân viên phải đủ 18 tuổi.

ALTER TABLE [dbo].[NhanVien] WITH CHECK ADD CONSTRAINT [CK\_NhanVien\_1] CHECK (((datepart(year,getdate())-datepart(year,[Ngay\_Sinh]))>=(18)))

7) **BẢNG NHÂN VIÊN:** Lương nhân viên phải từ 1.000.000 trở lên.

ALTER TABLE [dbo].[NhanVien] WITH CHECK ADD CONSTRAINT [CK\_NhanVien\_2] CHECK (([Luong]>=(1000000)))

8) **BẢNG NHÂN VIÊN:** Giới tính nhận hai giá trị: 0 nếu là NAM và 1 nếu là NỮ

ALTER TABLE [dbo].[NhanVien] WITH CHECK ADD CONSTRAINT [CK\_NhanVien] CHECK (([Gioi\_Tinh]=(0) OR [Gioi\_Tinh]=(1)))

9) **BẢNG CHI\_TIẾT HỢP ĐỒNG:** Loại hợp đồng nhận 2 giá trị: 1 – Thuê, 0 – Bán.

ALTER TABLE [dbo].[ChiTiet\_HopDong] WITH CHECK ADD CONSTRAINT [CK\_ChiTiet\_HopDong] CHECK (([Loai\_HopDong]=(1) OR [Loai\_HopDong]=(0)))

10) **BẢNG CHI\_TIẾT HỢP ĐỒNG:** Để trống **NGÀY HẾT HẠN HỢP ĐỒNG** nếu loại hợp đồng là **BÁN**.

ALTER TABLE [dbo].[ChiTiet\_HopDong] WITH CHECK ADD CONSTRAINT [CK\_ChiTiet\_HopDong\_1] CHECK (([Loai\_HopDong]=(0) AND [NgayHetHan] IS NULL))

11) **BẢNG KHÁCH HÀNG:** Thuê/Mua nhận giá trị tương ứng: 1 – Thuê, 0 - Mua

ALTER TABLE [dbo].[KhachHang] WITH CHECK ADD CONSTRAINT [CK\_KhachHang] CHECK (([Thue\_Mua]=(1) OR [Thue\_Mua]=(0)))

12) **BẢNG KHÁCH HÀNG:** Đã liên hệ nhận 2 giá trị tương ứng: 1 – Đã liên hệ, 0 – Chưa liên hệ

ALTER TABLE [dbo].[KhachHang] WITH CHECK ADD CONSTRAINT [CK\_KhachHang\_1] CHECK (([Da\_Lien\_He]=(1) OR [Da\_Lien\_He]=(0)))

13) **BẢNG KHÁCH HÀNG:** tiêu chí giá cao nhất phải lớn hơn hoặc bằng tiêu chí giá thấp nhất.

ALTER TABLE [dbo].[KhachHang] WITH CHECK ADD CONSTRAINT [CK\_KhachHang\_2] CHECK (([TieuChi\_Gia\_Max]>=[TieuChi\_Gia\_Min]))

14) **BẢNG CHI\_TIẾT HỢP ĐỒNG:** Ngày lập hợp đồng có giá trị mặc định là ngày hôm nay và ngày hết hạn phải sau ngày lập hợp đồng hoặc NULL.

[NgayLap] [date] NOT NULL DEFAULT(GETDATE())

…

ALTER TABLE [dbo].[ChiTiet\_HopDong] WITH CHECK ADD CONSTRAINT [CK\_ChiTiet\_HopDong\_2] CHECK (([NgayHetHan] IS NULL OR [NgayHetHan]>[NgayLap]))

15) **BẢNG CHI\_TIẾT HỢP ĐỒNG:** Nhà đang cho thuê, chưa hết hạn hợp đồng thì không được thêm hợp đồng bán/cho thuê mới.

CREATE TRIGGER rb\_HopDong ON ChiTiet\_HopDong FOR INSERT

AS

IF EXISTS ( SELECT \* FROM ChiTiet\_HopDong WHERE NgayHetHan IS NOT NULL AND NgayHetHan < getdate() )

BEGIN

Print 'Khong the them hop dong moi vi hop dong truoc chua het han'

RollBack Tran

END

GO

16) **BẢNG CHI\_TIẾT HỢP ĐỒNG và BẢNG NHÀ:** Ngày lập hợp đồng phải sau Ngày đăng bài.

CREATE TRIGGER rb\_HopDong\_2 ON ChiTiet\_HopDong FOR INSERT, UPDATE

AS

IF EXISTS (SELECT \* FROM (Nha JOIN ChiTiet\_HopDong ON Nha.ID = ChiTiet\_HopDong.Ma\_Nha) WHERE Nha.Ngay\_Dang > ChiTiet\_HopDong.NgayLap)

BEGIN

Print 'Ngay ky hop dong khong the truoc ngay dang bai'

RollBack Tran

END

GO

17) **BẢNG NHÀ và BẢNG CHI\_TIẾT XEM\_NHÀ:** Ngày xem nhà phải sau Ngày đăng bài.

CREATE TRIGGER rb\_XemNha On ChiTiet\_XemNha FOR INSERT, UPDATE

AS

IF EXISTS (SELECT \* FROM (Nha JOIN ChiTiet\_XemNha ON Nha.ID = ChiTiet\_XemNha.Ma\_Nha) WHERE Nha.Ngay\_Dang < ChiTiet\_XemNha.Ngay)

BEGIN

Print 'Ngay xem nha khong the truoc ngay dang bai'

RollBack Tran

END

GO

# **XÁC ĐỊNH LOẠI NGƯỜI DÙNG**

|  |  |
| --- | --- |
| **LOẠI NGƯỜI DÙNG** | **MỘT SỐ HOẠT ĐỘNG TƯƠNG ỨNG** |
| **CHỦ SỞ HỮU NHÀ** | * Quản lý thông tin nhà đã đăng: thêm, xóa, sửa thông tin. * Kiểm tra thông tin người muốn thuê (trong trường hợp là nhà bán) * Kiểm tra thông tin chi nhánh quản lý nhà. * Kiểm tra quá trình thuê / bán nhà và thông tin người thuê/mua nhà. |
| **NHÂN VIÊN** | * Kiểm tra sự trùng khớp giữa tiêu chí của người muốn thuê/mua nhà và người muốn đăng, thông báo cho khách hàng. * Lập hợp đồng khi khách muốn thuê / mua nhà. * Ghi chú các nhận xét vào bảng Chitiết\_xemnhà khi khách đến xem nhà và nhận xét. |
| **NGƯỜI TÌM THUÊ / MUA NHÀ** | * Kiểm tra thông tin chủ nhà. * Kiểm tra thông tin nhà. * Kiểm tra thông tin chi nhánh quản lý. * Kiểm tra thông tin nhân viên phụ trách nhà. * Kiểm tra thông tin hợp đồng. |
| **QUẢN TRỊ CHI NHÁNH** | * Quản lý thông tin khách hàng. |

# **CÁC CHỨC NĂNG HỆ THỐNG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TÊN CHỨC NĂNG** | **MÔ TẢ** | **INPUT** | **OUTPUT** | **ĐỐI TƯỢNG SỬ DỤNG** |
| **Hiển thị thông tin nhà** | Hiển thị thông tin chi tiết về các ngôi nhà đang được đăng tải trên hệ thống. |  | Danh sách nhà và thông tin chi tiết | Tất cả |
| **Lọc nhà theo các tiêu chí** | Hiển thị các ngôi nhà theo tiêu chí tìm kiếm như giá tiền, số phòng, vị trí… | Điều kiện, tiêu chí tìm nhà | Danh sách nhà phù hợp với tiêu chí đã chọn | Tất cả |
| **Sắp xếp thông tin nhà** | Sắp xếp danh sách thông tin nhà theo tiêu chí như”   * Giá từ cao đến thấp hoặc ngược lại. * Số phòng từ nhiều đến ít hoặc ngược lại | * Danh sách thông tin nhà cần sắp xếp. * Tiêu chí sắp xếp. | Danh sách nhà được sắp xếp theo tiêu chí input. | Tất cả |
| **Thêm một thông tin nhà** | Thêm một dòng về thông tin nhà cần bán hoặc cần thuê. | Tất cả thông tin liên quan đến ngôi nhà bao gồm vị trí, giá, số phòng, điều kiện của chủ nhà (nếu là nhà bán)… | Một dòng mới về thông tin nhà mà chưa có nhân viên phụ trách. | Chủ sở hữu |
| **Duyệt thông tin nhà** | Khi một ngôi nhà được thêm bởi chủ sở hữu, ngôi nhà vẫn chưa có nhân viên phụ trách. Quản trị viên của chi nhánh sẽ kiểm tra và chỉ định nhân viên phụ trách cho những ngôi nhà này. | * Danh sách thông tin các ngôi nhà chưa có nhân viên phụ trách. * Danh sách nhân viên. | Danh sách thông tin các ngôi nhà đã có nhân viên phụ trách. | Quản trị của chi nhánh |
| **Cập nhật thông tin ngôi nhà** | Thông tin của ngôi nhà có thể thay đổi và được cập nhật theo yêu cầu từ chủ nhà, do nhân viên thực hiện. | * Mã ngôi nhà. * Các thông tin cần cập nhật. | Thông tin ngôi nhà mới với thông tin cập nhật. | Nhân viên |
| **Xóa một ngôi nhà** | Chủ nhà có thể yêu cầu nhân viên xóa ngôi nhà đã đăng tải trước ngày hết hạn bài đăng | Mã ngôi nhà cần xóa. | Cơ sở dữ liệu không còn thông tin ngôi nhà với mã input. | Nhân viên |
| **Thêm nhận xét cho ngôi nhà** | Khi một người đi xem nhà sẽ có các nhận xét, nhân viên cần ghi chú lại. | * Mã ngôi nhà được nhận xét. * Mã khách hàng. * Ngày xem nhà. * Nhận xét của khách hàng. | Dòng dữ liệu mới trong bảng Chi tiết xem nhà. | Nhân viên |
| **Thêm một khách hàng** | Khi có khách hàng liên hệ chi nhánh, quản trị viên sẽ thêm thông tin khách hàng này vào cơ sở dữ liệu. | Thông tin của khách hàng | Một row khách hàng mới trong cơ sở dữ liệu | Quản trị chi nhánh |
| **Cập nhật một khách hàng** | Khách hàng có thể yêu cầu thay đổi thông tin cá nhân cũng như tiêu chí tìm nhà. | * Mã khách hàng. * Thông tin cần thay đổi. | Row khách hàng tương ứng với thông tin cập nhật. | Quản trị chi nhánh |
| **Xóa một khách hàng** | Khách hàng có thể yêu cầu xóa thông tin của mình trên app nếu như không muốn tiếp tục sử dụng dịch vụ. | Mã khách hàng cần xóa. | Xóa khách hàng khỏi CSDL | Quản trị chi nhánh |
| **Thêm hợp đồng** | Khi khách xem nhà đồng ý, nhân viên sẽ thêm hợp đồng để xác nhận quyền sử dụng / quyền sở hữu ngôi nhà của khách. | Mã khách hàng  Mã nhân viên  Mã chủ nhà  Thông tin hợp đồng | Một hợp đồng mới trong bảng hợp đồng.  Nếu là nhà bán, cần cập nhật lại Mã CSH của nhà trong bảng nhà. | Nhân viên |

# **CÁC CHỨC NĂNG TRANH CHẤP ĐỒNG THỜI và XỬ LÝ TÌNH HUỐNG TRANH CHẤP**

1. **DIRTY READ**
2. **Trường hợp 1 – Cập nhật giá thuê nhà mới nhưng người xem vẫn thấy giá cũ**

|  |  |  |
| --- | --- | --- |
|  | **T1 – Nhân viên thay đổi giá thuê nhà** | **T2 – Khách hàng xem thông tin nhà** |
| 1 | Nhân viên tiến hành kiểm tra mã số 4 có tồn tại không. |  |
| 2 | Sửa đổi giá của nhà 4 từ 4000 thành 3500 đồng. |  |
| 3 |  | Khách hàng đọc được nhà là giá thuê của nhà 4 = 3500. |
| 4 | Lỗi section. |  |
| 5 |  | **Commit** |
| 6 | **Roll back => giá nhà quay lại 4000**. |  |

Trong trường hợp trên, khách hàng đã đọc phải dữ liệu “rác” về giá của một căn nhà. Do đó, khi liên hệ nhân viên, khách hàng sẽ bị hoang mang vì giá thật sự và giá đọc được trên app khác nhau.

***Code mô tả T1:***

/\*T11\*/

USE QuanLyNha

GO

ALTER PROCEDURE T11(@id as int, @GiaMoi as int)

As

Begin Transaction

If Not Exists (Select \* From NHA Where ID = @id)

Begin

Print 'Nha khong ton tai'

RollBack Transaction

End

Update Nha Set Gia = @GiaMoi Where ID = @id

Waitfor Delay '00:00:05'

Rollback Transaction

Go

EXEC T11 4,3500

EXEC T12 4

***Code mô tả T2:***

/\*T12\*/

USE QuanLyNha

GO

Alter PROCEDURE T12(@id as int)

AS

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

Begin Transaction

Declare @gia int

Begin Try

Select @gia = Nha.Gia From Nha Where Nha.ID = @id

Select @gia As 'GiaNha04'

End Try

Begin Catch

Print 'Loi update gia nha trong T12'

Rollback Transaction

End Catch

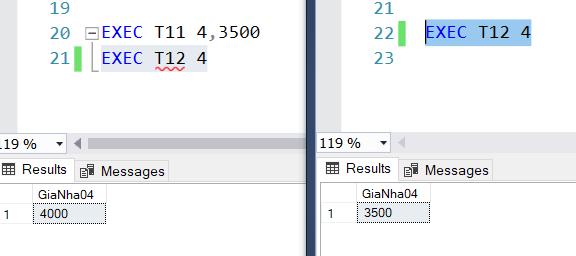
Commit

return @gia

Go

EXEC T12 4

***Kết quả trên SQL:***



Giá thuê đọc được của nhà **4** trong T2.

Giá thuê thật sự của nhà **4** trong database.

* **SỬA LỖI: dùng mức cô lập Read Committed (phát khóa X và giữ đến cuối giao tác)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Nhân viên thay đổi giá thuê nhà** | **Khóa** | **Khóa** | **T2 – Khách hàng xem thông tin nhà** |
| 1 | Nhân viên tiến hành kiểm tra mã số nhà **4** có tồn tại không. | S |  |  |
| 2 | Sửa đổi giá của nhà **4** từ 4000 thành 3500 đồng. | X | **S =>** không được vì T1 giữ khoá X | Khách hàng đọc giá của nhà **4** => giá = 4000, thống nhất với dữ liệu trong database. |
| 3 |  |  |  |
| 4 | Lỗi section | Đợi… |  |
| 5 | **Roll back** | Đợi… |  |
| 6 |  |  | **S** | Khách hàng đọc giá của nhà **4** => giá = 4000, thống nhất với dữ liệu trong database. |
| 7 |  |  |  | **Commit** |

***Code mô tả T1\_Fix:***

/\*T11\*/

USE QuanLyNha

GO

IF OBJECT\_ID(N'T11\_Fix') IS NOT NULL

DROP PROCEDURE T11\_Fix

GO

CREATE PROCEDURE T11\_Fix(@id as int, @GiaMoi as int)

As

Begin Transaction

If Not Exists (Select \* From NHA Where ID = @id)

Begin

Print 'Nha khong ton tai'

RollBack Transaction

End

Update Nha Set Gia = @GiaMoi Where ID = @id

Waitfor Delay '00:00:05'

Rollback Transaction

Go

EXEC T11\_Fix 4,3500

EXEC T12\_Fix 4

***Code mô tả T2\_Fix:***

/\*T12\*/

USE QuanLyNha

GO

IF OBJECT\_ID(N'T12\_Fix') IS NOT NULL

DROP PROCEDURE T12\_Fix

GO

CREATE PROCEDURE T12\_Fix(@id as int)

AS

Begin Transaction

Declare @gia int

Begin Try

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

Select @gia = Gia From Nha Where ID = @id

Select @gia As 'GiaNha04'

End Try

Begin Catch

Print 'Loi update gia nha trong T12\_Fix'

Rollback Transaction

End Catch

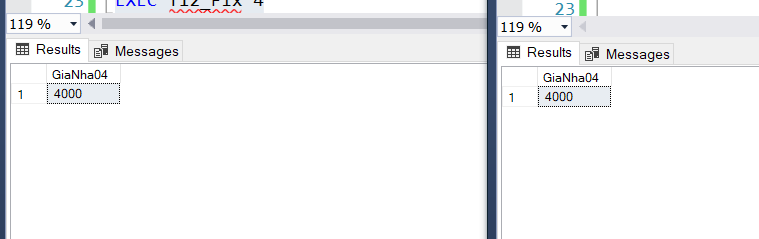
Commit Transaction

return @gia

Go

EXEC T12\_Fix 4

***Kết quả trong SQL:***



GIÁ HIỂN THỊ CỦA NHÀ **4** THỐNG NHẤT Ở 2 TRANSACTION.

1. **Trường hợp 2 – Hiển thị nhà đã bị xóa trên app nhưng sau đó nhà vẫn hiện thông tin bình thường**

|  |  |  |
| --- | --- | --- |
|  | **T1 – Nhân viên xóa nhà khỏi hệ thống** | **T2 – Khách hàng truy cập xem nhà** |
| 1 | Kiểm tra mã số nhà **6** có tồn tại không |  |
| 2 | Thao tác xóa nhà. |  |
| 3 |  | Khách hàng truy cập vào xem nhà => danh sách không hiển thị nhà 6. |
| 4 | Thao tác xóa nhà gặp lỗi => không xóa được |  |
| 5 |  | **Commit** |
| 6 | **Roll back** |  |

Nhân viên phụ trách sau khi load danh sách thấy nhà số **6** đã bị xóa, sẽ yên tâm không kiểm tra lại. Tuy nhiên, trên thực tế nhà vẫn còn trong hệ thống, dẫn đến khách hàng liên hệ xem nhà và ảnh hưởng thời gian hai bên.

***Code demo T1:***

USE QuanLyNha

GO

IF OBJECT\_ID(N'T21') IS NOT NULL

DROP PROCEDURE T21

GO

CREATE PROCEDURE T21(@id as int)

AS

Begin Transaction

If Not Exists (Select \* From Nha Where ID = @id)

Begin

Print 'Nha khong ton tai'

RollBack Transaction

End

DELETE FROM Nha WHERE ID = @id

Waitfor Delay '00:00:05'

Rollback Transaction

GO

EXEC T21 6

EXEC T22

***Code demo T2:***

USE QuanLyNha

GO

IF OBJECT\_ID(N'T22') IS NOT NULL

DROP PROCEDURE T22

GO

CREATE PROCEDURE T22

AS

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

BEGIN TRANSACTION

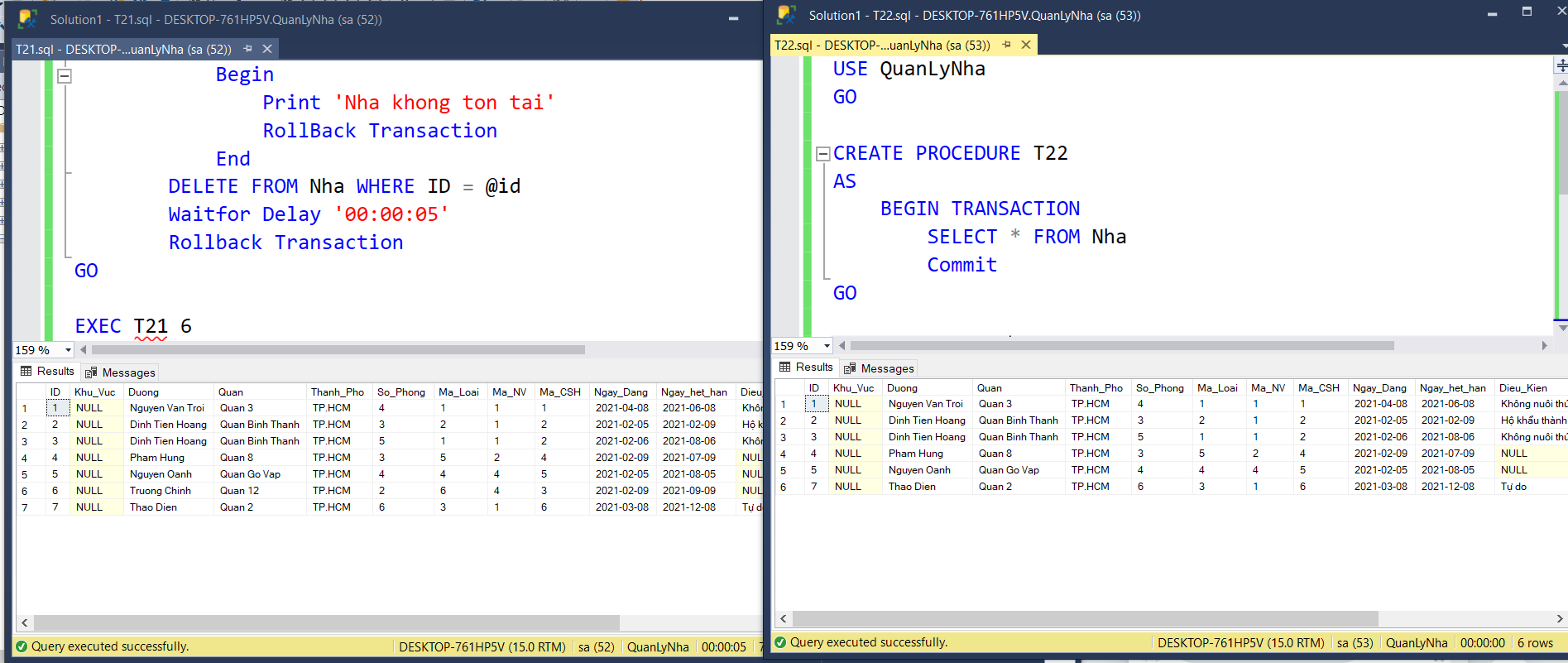
SELECT \* FROM Nha

Commit

GO

EXEC T22

***Kết quả trên SQL:***

******

* **SỬA LỖI:** SỬ DỤNG MỨC CÔ LẬP **READ COMMITTED** (không đọc data khi chưa commit)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Nhân viên xóa nhà khỏi hệ thống** | **Khoá** | **Khoá** | **T2 – Khách hàng truy cập xem nhà** |
| 1 | Kiểm tra mã số nhà **6** có tồn tại không | **S** |  |  |
| 2 | Thao tác xóa nhà. | X |  |  |
| 3 |  | Xin khoá S => không được vì T2 Ggiữ khoá X | Khách hàng truy cập vào xem nhà => danh sách không hiển thị nhà 6. |
| 4 | Thao tác xóa nhà gặp lỗi => không xóa được |  |  |
| 5 | **Roll back** |  |  |
| 6 |  |  | S | Truy cập xem nhà, nhà 6 vẫn hiển thị trong database. |
| 7 |  |  |  | **Commit** |

Như vậy, khi nhân viên duyệt lại nhà một lần nữa, sẽ thấy nhà chưa bị xoá và biết thao tác của mình đã gặp lỗi, nhờ đó thực thi lại kịp thời.

***Code demo T1\_Fix:***

USE QuanLyNha

GO

IF OBJECT\_ID('T21\_Fix') IS NOT NULL

DROP PROCEDURE T21\_Fix

GO

CREATE PROCEDURE T21\_Fix(@id as int)

AS

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

Begin Transaction

If Not Exists (Select \* From Nha Where ID = @id)

Begin

Print 'Nha khong ton tai'

RollBack Transaction

End

DELETE FROM Nha WHERE ID = @id

Waitfor Delay '00:00:05'

Rollback Transaction

GO

EXEC T21\_Fix 6

EXEC T22\_Fix

***Code demo T2\_Fix:***

USE QuanLyNha

GO

IF OBJECT\_ID('T22\_Fix') IS NOT NULL

DROP PROCEDURE T22\_Fix

GO

CREATE PROCEDURE T22\_Fix

AS

BEGIN TRANSACTION

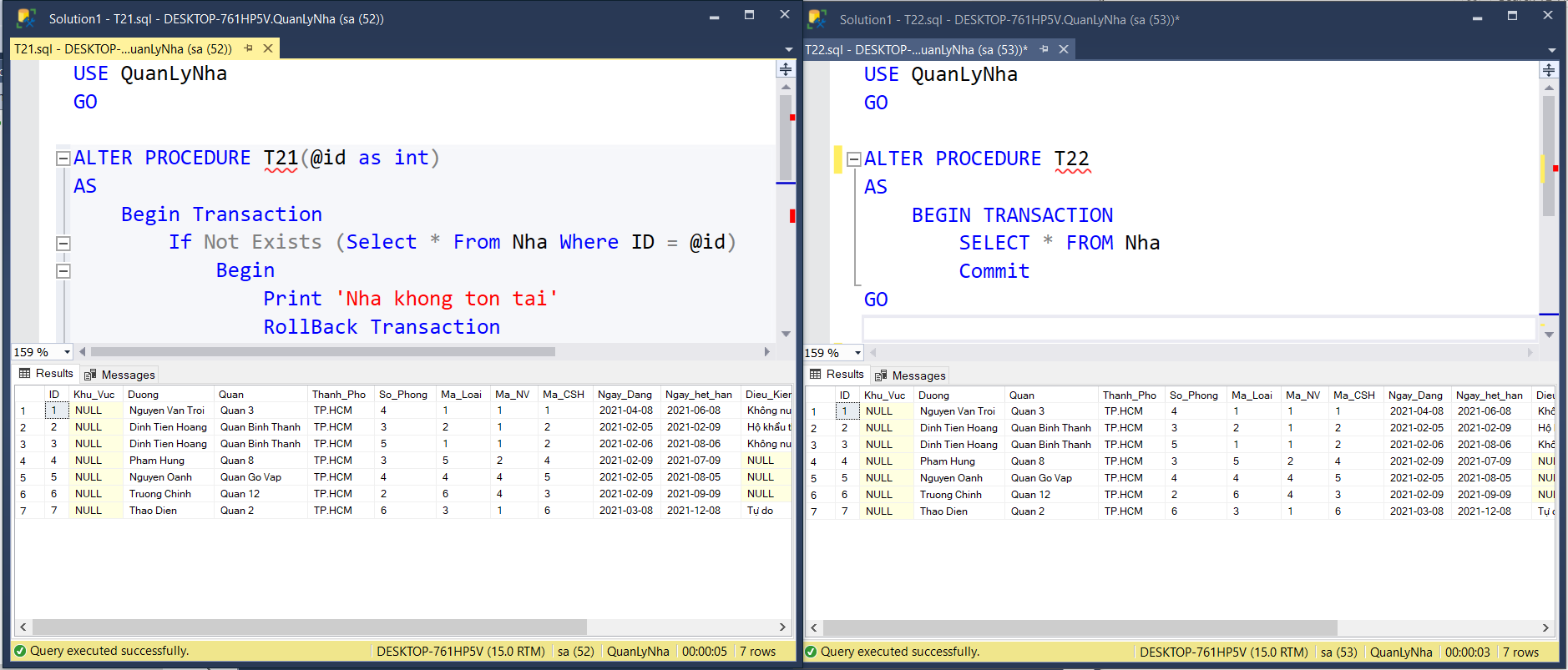
SELECT \* FROM Nha

Commit

GO

EXEC T22\_Fix

***Kết quả trên SQL:***

******

1. **Trường hợp 3 – Hợp đồng mua bán bị hủy do có hợp đồng cho thuê trước chưa hết hạn, nhưng khách hàng vẫn đọc thấy trạng thái “đã bán” trên ứng dụng**

|  |  |  |
| --- | --- | --- |
|  | **T1 – Nhân viên lập hợp đồng cho khách hàng** | **T2 – Khách hàng đọc thông tin của nhà** |
| 1 | Nhân viên chuyển trạng thái của nhà thành “Đã bán” |  |
| 2 | Nhân viên cập nhật mã Chủ sở hữu là mã Khách hàng |  |
| 3 |  | Khách hàng đọc tin, thấy trạng thái nhà “Đã bán” |
| 4 | Hợp đồng bị hủy vì nhân viên nhầm hợp đồng cho thuê trước chưa hết hạn. |  |
| 5 | **Roll back** |  |
| 6 |  | **Commit** |

Trong trường hợp này, trạng thái của nhà đã gây hiểu lầm, khiến khách hàng hiện tại không mua được nhà nhưng thực tế hợp đồng không được cập nhật trong hệ thống.

***Code demo T1:***

/\*T31\*/

USE QuanLyNha

GO

IF OBJECT\_ID('T31') IS NOT NULL

DROP PROCEDURE T31

GO

CREATE PROCEDURE T31(@MaNha as int, @MaKH as int, @MaNV as int, @loai as int)

AS

BEGIN

BEGIN TRANSACTION

UPDATE Nha SET Tinh\_Trang = N'ĐÃ BÁN' WHERE ID = @MaNha

UPDATE Nha SET Ma\_CSH = @MaKH WHERE ID = @MaNha

Waitfor Delay '00:00:05'

IF NOT EXISTS(SELECT \* FROM Nha Where ID = @MaNha)

BEGIN

Print 'Nha khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM KhachHang Where ID = @MaKH)

BEGIN

Print 'Khach hang khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM NhanVien Where ID = @MaNV)

BEGIN

Print 'Nhan vien khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF EXISTS(SELECT \* From ChiTiet\_HopDong WHERE Ma\_Nha = @MaNha AND Loai\_HopDong = 1 AND NgayHetHan > GETDATE())

BEGIN

PRINT 'Hop dong cho thue truoc chua het han'

ROLLBACK TRANSACTION

RETURN

END

BEGIN TRY

INSERT INTO ChiTiet\_HopDong(Ma\_Nha, Ma\_NhanVien, Ma\_KhachHang, Loai\_HopDong, NgayLap) VALUES(@MaNha, @MaNV, @MaKH, 0, GETDATE())

END TRY

BEGIN CATCH

SELECT ERROR\_MESSAGE() AS ErrorMessage

ROLLBACK TRANSACTION

RETURN

END CATCH

COMMIT TRANSACTION

END

GO

EXEC T31 1,4,1,0

SELECT \* FROM ChiTiet\_HopDong

EXEC T32 1

***Code demo T2:***

/\*T31\*/

USE QuanLyNha

GO

IF OBJECT\_ID('T32') IS NOT NULL

DROP PROCEDURE T32

GO

CREATE PROCEDURE T32(@MaNha as int)

AS

BEGIN

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

BEGIN TRANSACTION

SELECT \* FROM Nha Where ID = @MaNha

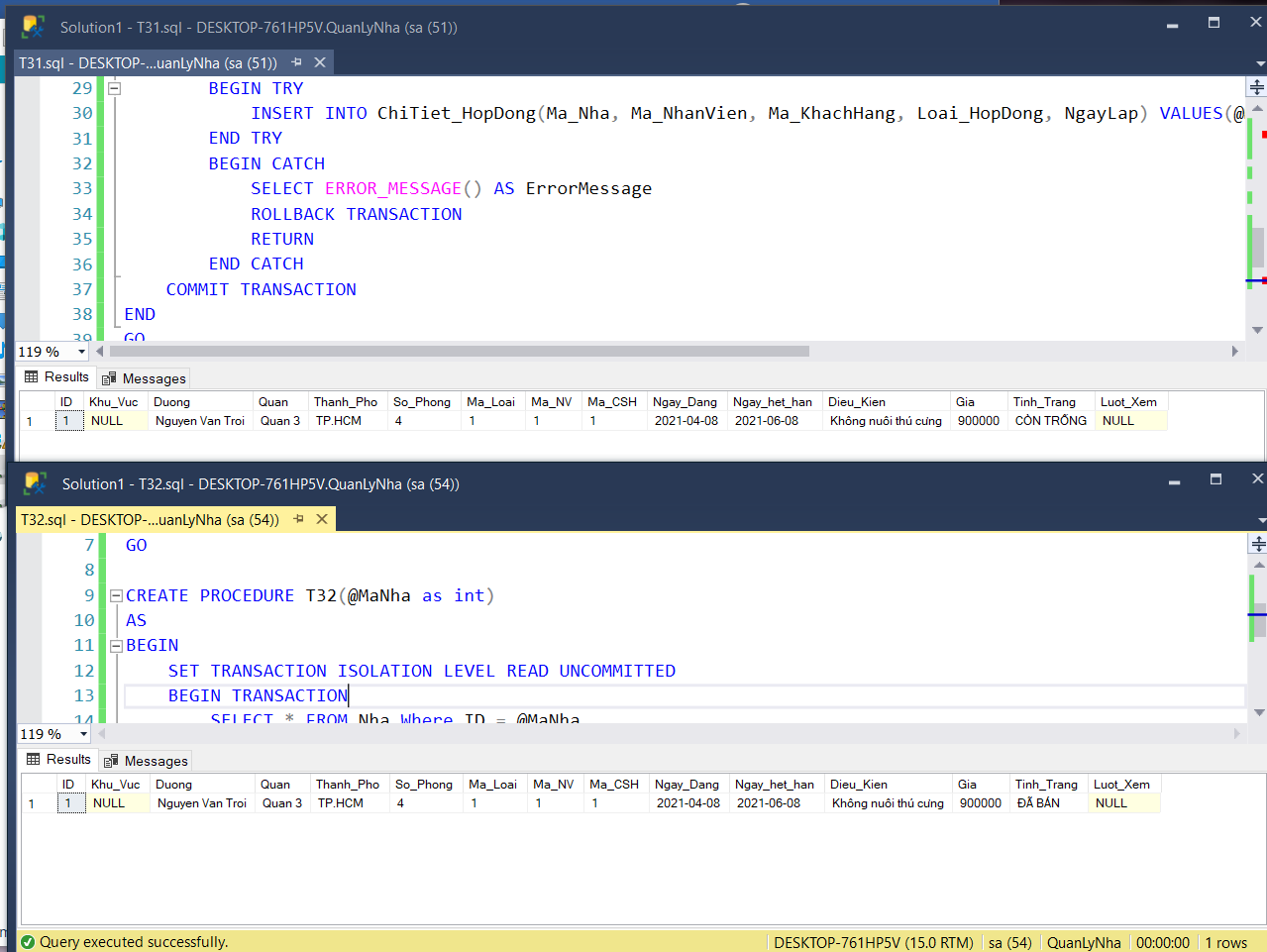
COMMIT TRANSACTION

END

GO

EXEC T32 1

***Kết quả trên SQL:***

******

* **SỬA LỖI:** DÙNG MỨC CÔ LẬP **READ COMMITTED** (không đọc thông tin đang được điều chỉnh)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Nhân viên lập hợp đồng cho khách hàng** | **Khoá** | **Khoá** | **T2 – Khách hàng đọc thông tin của nhà** |
| 1 | Nhân viên chuyển trạng thái của nhà thành “Đã bán” | **Khoá X** |  |  |
| 2 | Nhân viên cập nhật mã Chủ sở hữu là mã Khách hàng |  |  |
| 3 |  | Xin khoá S => không được vì T1 giữ khoá X | Khách hàng đọc tin, thấy trạng thái nhà “Đã bán” |
| 4 | Hợp đồng bị hủy vì nhân viên nhầm hợp đồng cho thuê trước chưa hết hạn. |  |  |
| 5 | **Roll back** |  |  |
| 6 |  |  | **S** | Đọc tin, trạng thái nhà còn trống => liên hệ nhân viên |

Với cách giải quyết này, khách hàng sẽ không còn đọc phải thông tin rác trên hệ thống (nếu có).

***Code demo T1\_Fix:***

/\*T31\*/

USE QuanLyNha

GO

IF OBJECT\_ID('T31\_Fix') IS NOT NULL

DROP PROCEDURE T31\_Fix

GO

CREATE PROCEDURE T31\_Fix(@MaNha as int, @MaKH as int, @MaNV as int, @loai as int)

AS

BEGIN

BEGIN TRANSACTION

UPDATE Nha SET Tinh\_Trang = N'ĐÃ BÁN' WHERE ID = @MaNha

UPDATE Nha SET Ma\_CSH = @MaKH WHERE ID = @MaNha

Waitfor Delay '00:00:05'

IF NOT EXISTS(SELECT \* FROM Nha Where ID = @MaNha)

BEGIN

Print 'Nha khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM KhachHang Where ID = @MaKH)

BEGIN

Print 'Khach hang khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM NhanVien Where ID = @MaNV)

BEGIN

Print 'Nhan vien khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF EXISTS(SELECT \* From ChiTiet\_HopDong WHERE Ma\_Nha = @MaNha AND Loai\_HopDong = 1 AND NgayHetHan > GETDATE())

BEGIN

PRINT 'Hop dong cho thue truoc chua het han'

ROLLBACK TRANSACTION

RETURN

END

BEGIN TRY

INSERT INTO ChiTiet\_HopDong(Ma\_Nha, Ma\_NhanVien, Ma\_KhachHang, Loai\_HopDong, NgayLap) VALUES(@MaNha, @MaNV, @MaKH, 0, GETDATE())

END TRY

BEGIN CATCH

SELECT ERROR\_MESSAGE() AS ErrorMessage

ROLLBACK TRANSACTION

RETURN

END CATCH

COMMIT TRANSACTION

END

GO

EXEC T31\_Fix 1,4,1,0

SELECT \* FROM ChiTiet\_HopDong

EXEC T32\_Fix 1

***Code demo T2\_Fix:***

/\*T31\*/

USE QuanLyNha

GO

IF OBJECT\_ID('T32\_Fix') IS NOT NULL

DROP PROCEDURE T32\_Fix

GO

CREATE PROCEDURE T32\_Fix(@MaNha as int)

AS

BEGIN

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

BEGIN TRANSACTION

SELECT \* FROM Nha Where ID = @MaNha

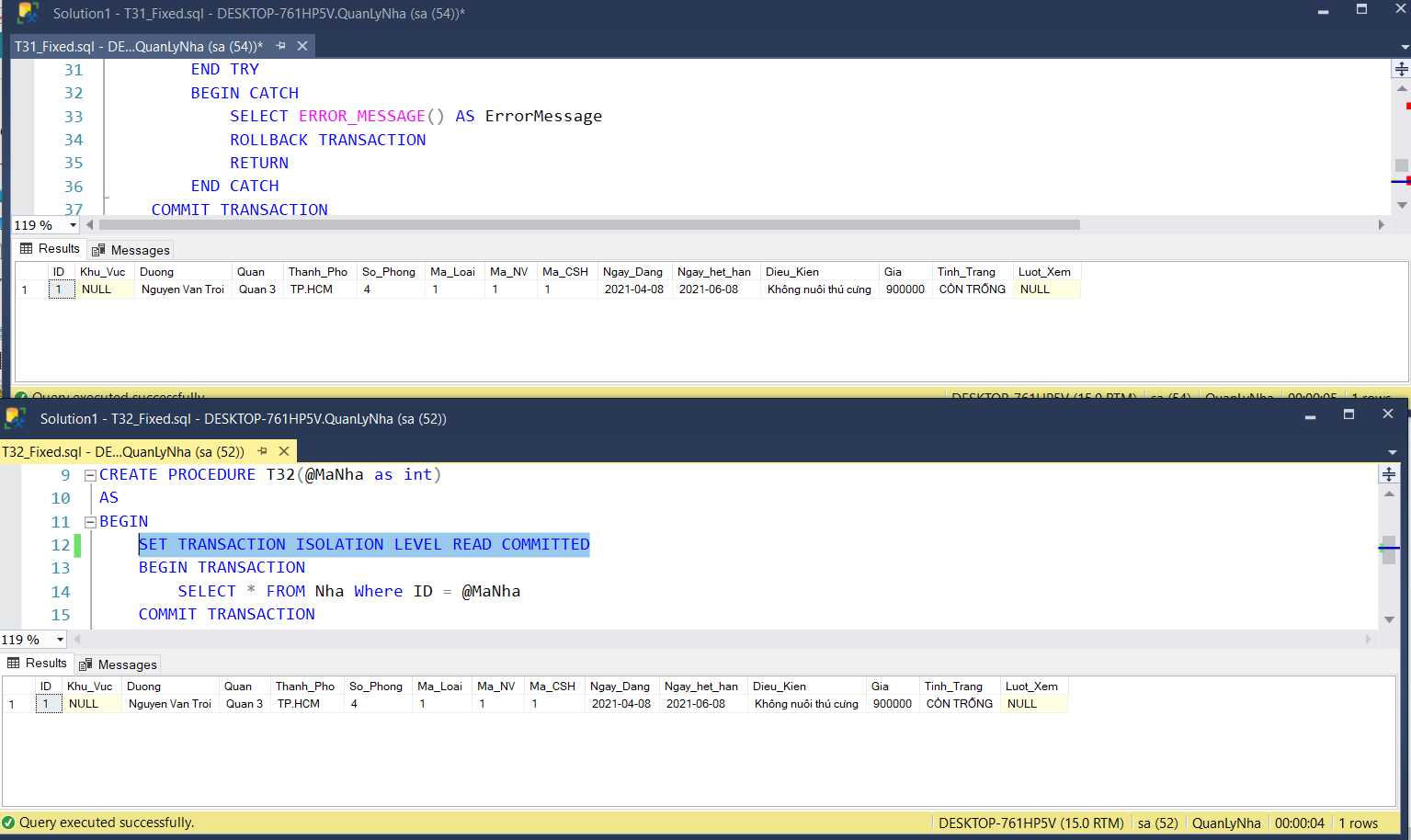
COMMIT TRANSACTION

END

GO

EXEC T32\_Fix 1

***Kết quả trên SQL:***

******

1. **UNREPEATABLE READ**
2. **Trường hợp 1 – Hiển thị tất cả các nhà có giá thuê thấp nhất nhưng không hiển thị được vì giá nhà tăng thêm một khoảng 100.000 phí dịch vụ ngày Lễ**

|  |  |  |
| --- | --- | --- |
|  | **T1 – Tìm nhà có giá thuê thấp nhất** | **T2 – Thay đổi giá thuê nhà** |
| 1 | Tìm giá thuê nhà thấp nhất hiện tại trong cơ sở dữ liệu => min = 2000 |  |
| 2 |  | Đọc danh sách tất cả các nhà |
| 3 |  | Update giá thuê mỗi nhà tăng thêm 100 |
| 4 | Lọc danh sách tất cả các nhà có giá thuê = giá thuê thấp nhất ở t1 => không tìm thấy nhà nào vì giá thuê đã tăng => min lúc này là 2100 khác 2000 |  |
| 5 | **Commit** |  |
| 6 |  | **Commit** |

Trong trường hợp này, khách hàng không thể lọc được danh sách nhà có giá thuê thấp nhất nếu như tại thời điểm đó giá trị nhà bị thay đổi đồng loạt.

***Code demo T1:***

USE QuanLyNha

GO

IF OBJECT\_ID('U11') IS NOT NULL

DROP PROCEDURE U11

GO

CREATE PROCEDURE U11

AS

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

BEGIN TRANSACTION

BEGIN TRY

DECLARE @min as int

SELECT @min = MIN(Gia) FROM Nha WHERE Nha.Dieu\_Kien IS NULL

SELECT @min AS GiaThapNhat

WAITFOR DELAY '00:00:05'

SELECT \* FROM Nha Where Gia = @min AND Nha.Dieu\_Kien IS NULL

END TRY

BEGIN CATCH

print 'Loi Tim Kiem Nha Thue Re Nhat'

Rollback Transaction

END CATCH

COMMIT TRANSACTION

GO

EXEC U11

***Code demo T2:***

/\*U12\*/

Use QuanLyNha

go

IF OBJECT\_ID('U12') IS NOT NULL

DROP PROCEDURE U12

GO

CREATE PROCEDURE U12(@gia as int)

As

Begin Transaction

Begin Try

Update Nha Set Gia = Gia + @gia

End Try

Begin Catch

Print 'Loi update gia nha trong U12'

Rollback Transaction

RETURN

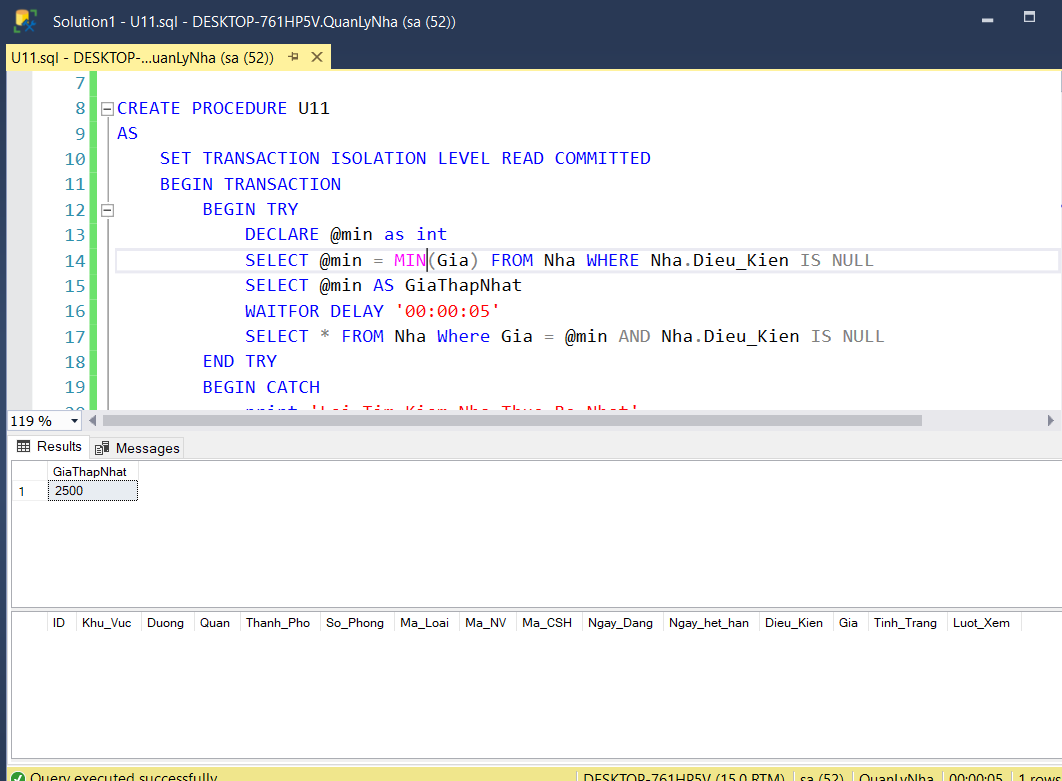
End Catch

Commit Transaction

Go

EXEC U12 -100

***Kết quả trên SQL:***

******

* **SỬA LỖI:** Sử dụng mức cô lập **REPEATABLE READ** (không cho thay đổi khi đang đọc)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Tìm nhà có giá thuê thấp nhất** | **Khóa** | **Khóa** | **T2 – Thay đổi giá thuê nhà** |
| 1 | Tìm giá thuê nhà thấp nhất hiện tại trong cơ sở dữ liệu | S |  |  |
| 2 |  | S | Đọc danh sách tất cả các nhà |
| 3 |  | X => không được vì còn khóa S ở T1 | Update giá thuê mỗi nhà tăng thêm 100 |
| 4 | Lọc danh sách tất cả các nhà có giá thuê = giá thuê thấp nhất ở t1 | Đợi… |  |
| 5 | **Commit** | Đợi… |  |
| 6 |  |  | X | Update giá thuê mỗi nhà tăng thêm 100 |
| 7 |  |  |  | **Commit** |

Như vậy, T1 sẽ thực hiện trước T2 và in ra trước giá nhà thấp nhất cũng như danh sách nhà trong cơ sở dữ liệu, đảm bảo tính thống nhất của dữ liệu.

***Code demo T1\_Fix:***

USE QuanLyNha

GO

IF OBJECT\_ID('U11\_Fix') IS NOT NULL

DROP PROCEDURE U11\_Fix

GO

CREATE PROCEDURE U11\_Fix

AS

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

DECLARE @min as int

SELECT @min = MIN(Gia) FROM Nha WHERE Nha.Dieu\_Kien IS NULL

SELECT @min AS GiaThapNhat

WAITFOR DELAY '00:00:05'

SELECT \* FROM Nha Where Gia = @min AND Nha.Dieu\_Kien IS NULL

END TRY

BEGIN CATCH

print 'Loi Tim Kiem Nha Thue Re Nhat'

Rollback Transaction

END CATCH

COMMIT TRANSACTION

GO

EXEC U11\_Fix

***Code demo T2\_Fix:***

/\*U12\*/

Use QuanLyNha

go

IF OBJECT\_ID('U12\_Fix') IS NOT NULL

DROP PROCEDURE U12

GO

CREATE PROCEDURE U12\_Fix(@gia as int)

As

Begin Transaction

Begin Try

Update Nha Set Gia = Gia + @gia

End Try

Begin Catch

Print 'Loi update gia nha trong tangGiaNha\_1'

Rollback Transaction

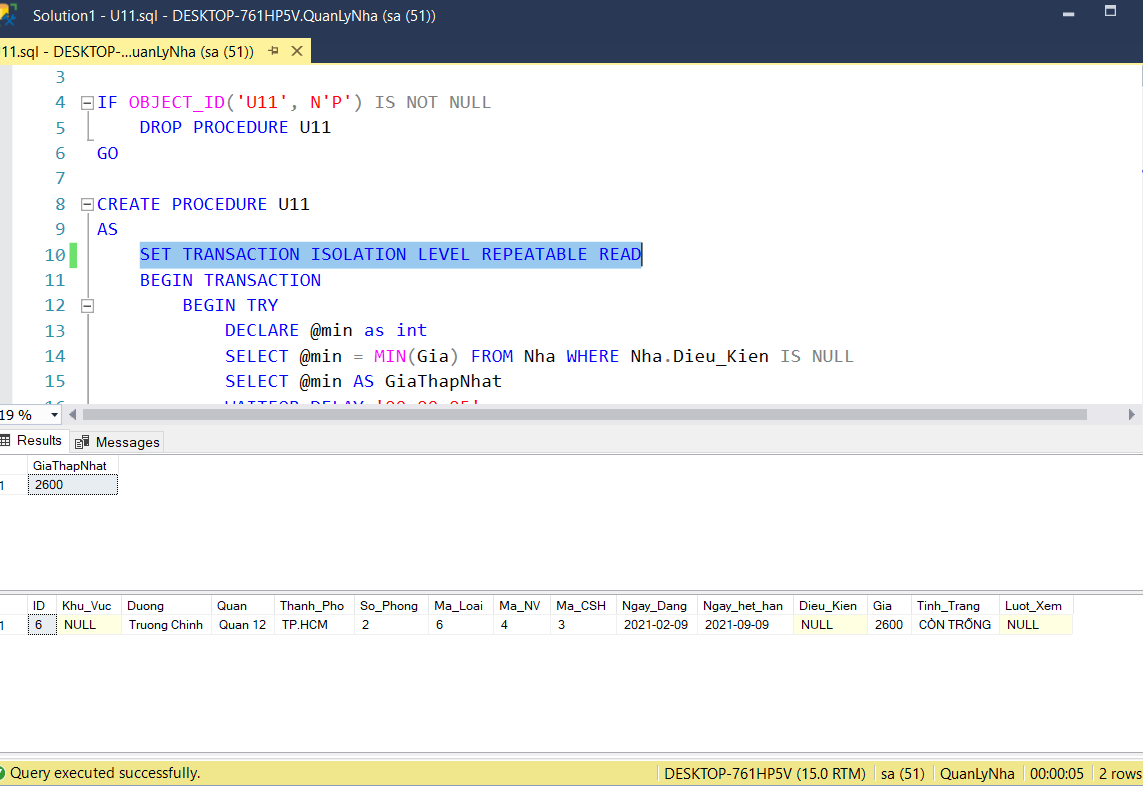
End Catch

Commit Transaction

Go

EXEC U12\_Fix 100

***Kết quả trên SQL:***

******

1. **Trường hợp 2 – Hiển thị danh sách nhà có có giá từ 2000 đến 4000: số lượng nhà và danh sách thật sự khác nhau**

|  |  |  |
| --- | --- | --- |
|  | **T1 – Hiển thị nhà theo tiêu chí yêu cầu** | **T2 – Thay đổi giá thuê của nhà** |
| 1 | Dùng hàm COUNT đếm tất cả các nhà có giá từ 2000 đến 4000 => 2 nhà |  |
| 2 |  | Thay đổi giá của nhà 6 từ 2000 còn 1500 |
| 3 | Dùng lệnh SELECT để lấy ra tất cả các nhà có giá từ 2000 đến 4000 và hiển thị => danh sách 1 nhà |  |
| 4 | **Commit** |  |
| 5 |  | **Commit** |

Trong trường hợp này, thông tin về số lượng nhà và danh sách nhà không đồng nhất trong T1.

***Code demo T1:***

USE QuanLyNha

GO

IF OBJECT\_ID('U21') IS NOT NULL

DROP PROCEDURE U21

GO

CREATE PROCEDURE U21(@from as int, @to as int)

AS

Update Nha Set Gia = 2000 Where ID = 6

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

BEGIN TRANSACTION

BEGIN TRY

SELECT COUNT(\*) AS So\_luong FROM Nha Where Gia >= @from AND Gia <= @to

WAITFOR DELAY '00:00:05'

SELECT \* FROM Nha Where Gia >= @from AND Gia <= @to

END TRY

BEGIN CATCH

print 'Loi tim kiem nha'

Rollback Transaction

END CATCH

COMMIT TRANSACTION

GO

EXEC U21 2000, 4000

***Code demo T2:***

/\*U12\*/

Use QuanLyNha

go

IF OBJECT\_ID('U22') IS NOT NULL

DROP PROCEDURE U22

GO

CREATE PROCEDURE U22(@id as int, @gia as int)

As

Begin Transaction

If Not Exists (Select \* From NHA Where ID = @id)

Begin

Print 'Nha khong ton tai'

RollBack Transaction

End

Begin Try

Update Nha Set Gia = @gia Where ID = @id

End Try

Begin Catch

Print 'Loi update gia nha'

Rollback Transaction

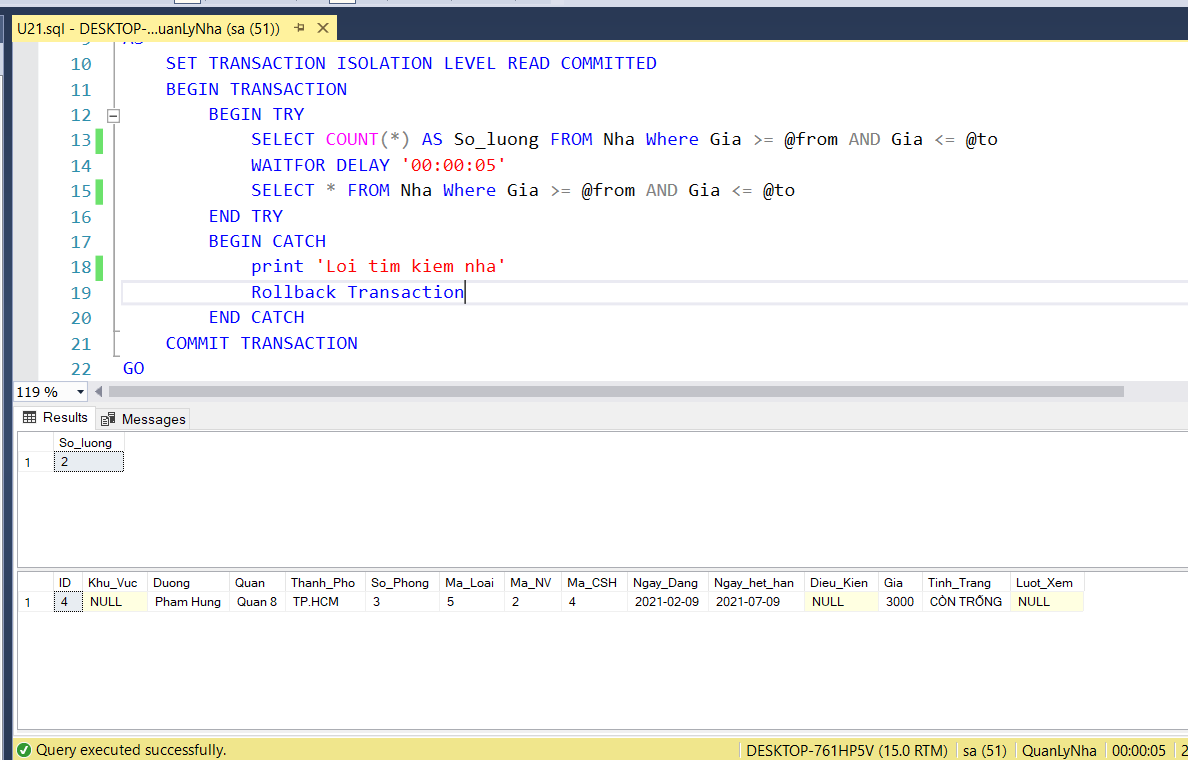
End Catch

Commit Transaction

Go

EXEC U22 6,1500

***Kết quả trên SQL:***

******

* **SỬA LỖI:** SỬ DỤNG MỨC CÔ LẬP **REPEATABLE READ** (ngăn sự thay đổi khi đang đọc)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Hiển thị nhà theo tiêu chí yêu cầu** | **Khóa** | **Khóa** | **T2 – Thay đổi giá thuê của nhà** |
| 1 | Dùng hàm COUNT đếm tất cả các nhà có giá từ 2000 đến 4000 => 2 nhà | **S** |  |  |
|  |  | **S** | Kiểm tra mã 003 có tồn tại không |
| 2 |  | X => không được vì còn khóa S | Thay đổi giá của nhà 003 từ 2.500.000 còn 2.000.000 |
| 3 | Dùng lệnh SELECT để lấy ra tất cả các nhà ccó giá từ 2000 đến 4000 và hiển thị => danh sách 2 nhà. | Đợi… |  |
| 4 | **Commit** | Đợi… |  |
| 5 |  |  | X | Thay đổi giá của nhà 003 từ 2000 còn 1500 |
| 6 |  |  |  | **Commit** |

Như vậy, vấn đề số lượng và danh sách bất đồng bộ bởi sự xuất hiện dữ liệu mới đã được giải quyết.

***Code demo T1\_Fix:***

USE QuanLyNha

GO

IF OBJECT\_ID('U21\_Fix') IS NOT NULL

DROP PROCEDURE U21\_Fix

GO

CREATE PROCEDURE U21\_Fix(@from as int, @to as int)

AS

Update Nha Set Gia = 2000 Where ID = 6

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

SELECT COUNT(\*) AS So\_luong FROM Nha Where Gia >= @from AND Gia <= @to

WAITFOR DELAY '00:00:05'

SELECT \* FROM Nha Where Gia >= @from AND Gia <= @to

END TRY

BEGIN CATCH

print 'Loi tim kiem nha'

Rollback Transaction

END CATCH

COMMIT TRANSACTION

GO

EXEC U21\_Fix 2000, 4000

***Code demo T2\_Fix:***

/\*U12\*/

Use QuanLyNha

go

IF OBJECT\_ID('U22\_Fix') IS NOT NULL

DROP PROCEDURE U22\_Fix

GO

CREATE PROCEDURE U22\_Fix(@id as int, @gia as int)

As

Begin Transaction

If Not Exists (Select \* From NHA Where ID = @id)

Begin

Print 'Nha khong ton tai'

RollBack Transaction

End

Begin Try

Update Nha Set Gia = @gia Where ID = @id

End Try

Begin Catch

Print 'Loi update gia nha'

Rollback Transaction

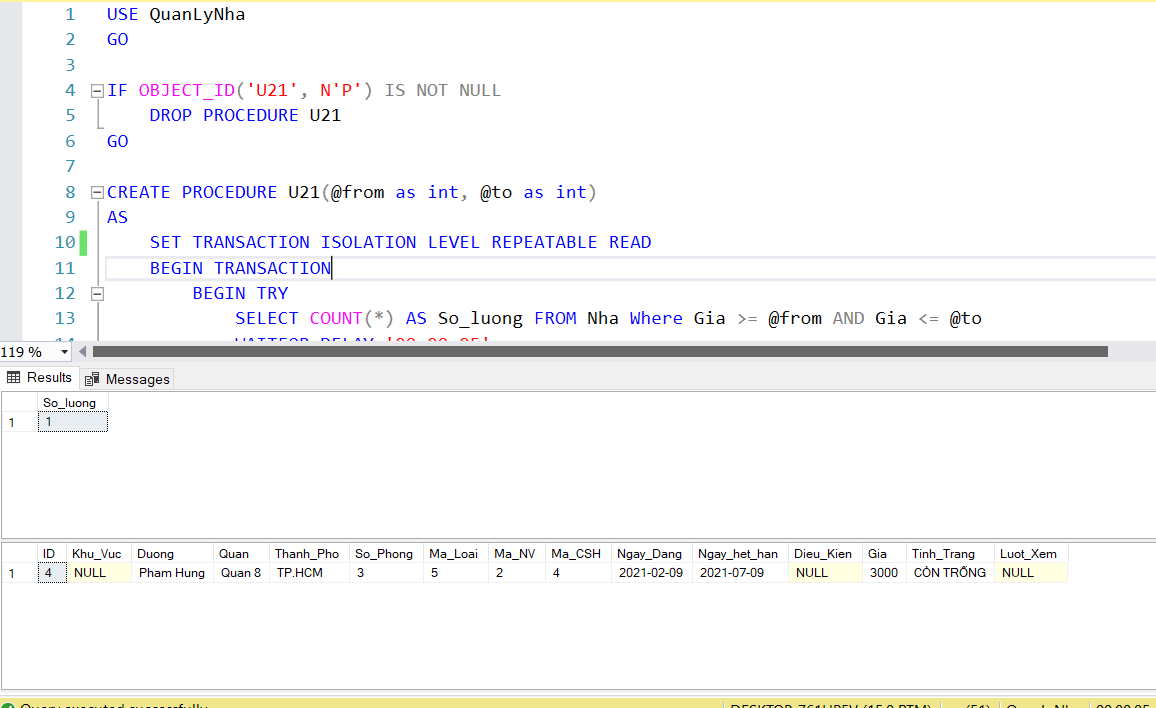
End Catch

Commit Transaction

Go

EXEC U22\_Fix 6,1500

***Kết quả trên SQL:***

******

1. **Trường hợp 3 – Nhân viên tìm các nhà ở đường Đinh Tiên Hoàng, quận Bình Thạnh. Trong lúc thực hiện tìm kiếm, quản trị viên thay đổi một số nhà trên đường Đinh Tiên Hoàng, quận Bình Thạnh thành Đinh Tiên Hoàng, quận 1.**

|  |  |  |
| --- | --- | --- |
|  | **T1 – Tìm kiếm nhà ở đường Đinh Tiên Hoàng, quận Bình Thạnh** | **T2 – Thay đổi quận của mã nhà 3 thành quận 1** |
| 1 | Nhân viên A đếm số lượng các nhà ở đường là Đinh Tiên Hoàng, thuộc quận Bình Thạnh => số lượng = 2 |  |
| 2 |  | Nhân viên B chọn và kiểm tra thông tin nhà 3 tồn tại. |
| 3 |  | Nhân viên B thay đổi quận Bình Thạnh thành Quận 1 |
| 4 | Thay đổi đường của tất cả các nhà có mã trong danh sách nhà có tên đường Đinh Tiên Hoàng thành đường Lê Văn Duyệt => chỉ có 1 sự thay đổi. |  |
| 5 | **Commit** |  |
| 6 |  | **Commit** |

Trong trường hợp này, T1 đã không lấy lại được danh sách cũ để thay đổi toàn bộ. Do đó, thay vì nhà 002 đã có thể đổi tên thành đường Lê Văn Duyệt, nhưng vì sự nhầm lẫn của nhân viên B, đến cuối cùng nhà 002 vẫn giữ đường Đinh Tiên Hoàng như cũ.

***Code demo T1:***

USE QuanLyNha

GO

IF OBJECT\_ID('U31') IS NOT NULL

DROP PROCEDURE U31

GO

CREATE PROCEDURE U31

AS

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

Update Nha Set Duong = N'Dinh Tien Hoang' Where ID = 2 OR ID = 3

Update Nha Set Quan = N'Quan Binh Thanh' Where ID = 2 OR ID = 3

BEGIN TRANSACTION

BEGIN TRY

SELECT \* FROM Nha Where Duong = N'Dinh Tien Hoang' AND Quan = N'Quan Binh Thanh'

WAITFOR DELAY '00:00:10'

UPDATE Nha SET Duong = N'Le Van Duyet' Where Duong = N'Dinh Tien Hoang' AND Quan = N'Quan Binh Thanh'

SELECT \* FROM Nha WHERE Duong = N'Le Van Duyet' AND Quan = N'Quan Binh Thanh'

END TRY

BEGIN CATCH

print 'Loi tim kiem nha'

Rollback Transaction

END CATCH

COMMIT TRANSACTION

GO

EXEC U31

***Code demo T2:***

/\*U12\*/

Use QuanLyNha

go

IF OBJECT\_ID('U32') IS NOT NULL

DROP PROCEDURE U32

GO

CREATE PROCEDURE U32(@id as int)

As

Begin Transaction

If Not Exists (Select \* From NHA Where ID = @id)

Begin

Print 'Nha khong ton tai'

RollBack Transaction

End

Update Nha Set Duong = N'Dinh Tien Hoang' Where ID = @id

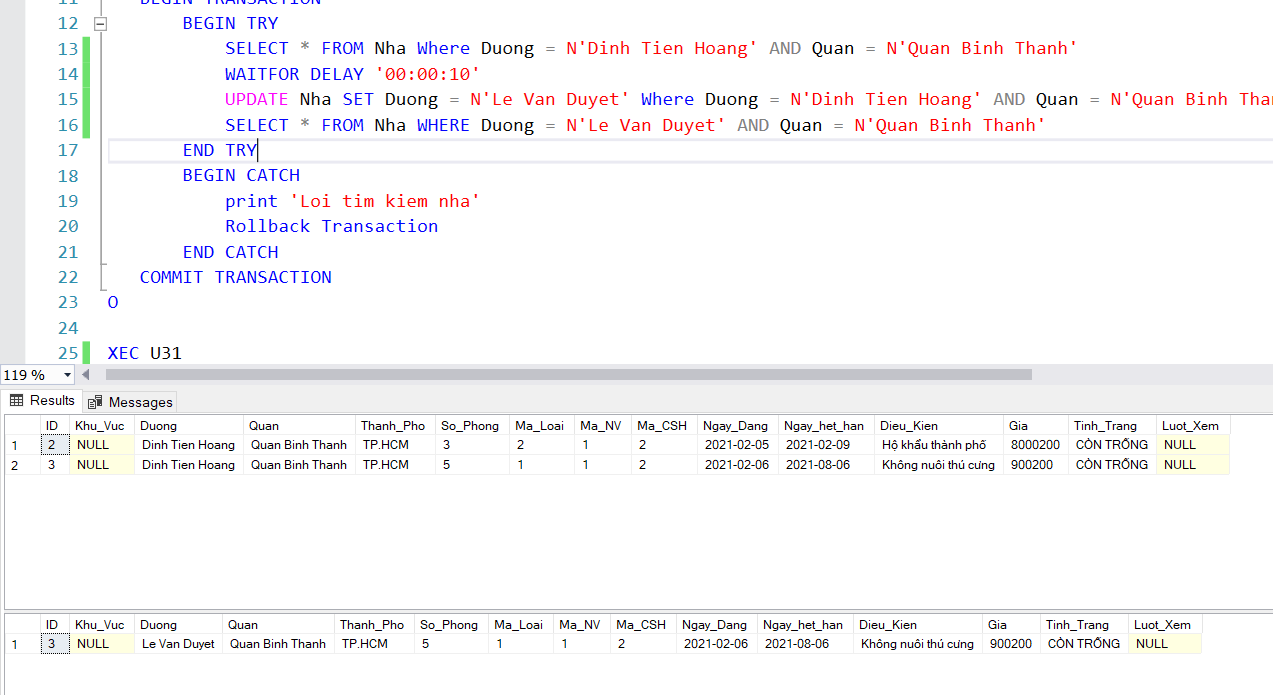
Update Nha Set Quan = N'Quan 1' Where ID = @id

Commit Transaction

Go

EXEC U32 2

***Kết quả trên SQL:***

******

* **SỬA LỖI:** DÙNG MỨC CÔ LẬP **REPEATABLE READ** (Không đọc dữ liệu khi đang sửa đổi)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Tìm kiếm nhà ở đường Đinh Tiên Hoàng, quận Bình Thạnh** | **Khoá** | **Khoá** | **T2 – Thay đổi quận của mã nhà 3 thành quận 1** |
| 1 | Nhân viên A đếm số lượng các nhà ở đường là Đinh Tiên Hoàng, thuộc quận Bình Thạnh => số lượng = 2 | S |  |  |
| 2 |  |  | Xin khoá X => Không được vì T1 đang giữ khoá S | Nhân viên B thay đổi quận Bình Thạnh thành Quận 1 |
| 3 | Thay đổi đường của tất cả các nhà có mã trong danh sách nhà có tên đường Đinh Tiên Hoàng thành đường Lê Văn Duyệt => thay đổi danh sách 2 nhà. |  | Đợi… |  |
| 4 | **Commit** |  | Khoá X | Nhân viên B thay đổi quận Bình Thạnh thành quận 1 |
| 5 |  |  | Khoá X | Nhân viên B thay đổi đường thành đường Đinh Tiên Hoàng |
| 6 |  |  |  | **Commit** |

Dữ liệu của nhà thống nhất về đường và quận. Ở T1, số lượng nhà đọc được và số lượng thay đổi là như nhau.

***Code demo T1\_Fix:***

USE QuanLyNha

GO

IF OBJECT\_ID('U31\_Fix') IS NOT NULL

DROP PROCEDURE U31\_Fix

GO

CREATE PROCEDURE U31\_Fix

AS

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

Update Nha Set Duong = N'Dinh Tien Hoang' Where ID = 2 OR ID = 3

Update Nha Set Quan = N'Quan Binh Thanh' Where ID = 2 OR ID = 3

BEGIN TRANSACTION

BEGIN TRY

SELECT \* FROM Nha Where Duong = N'Dinh Tien Hoang' AND Quan = N'Quan Binh Thanh'

WAITFOR DELAY '00:00:10'

UPDATE Nha SET Duong = N'Le Van Duyet' Where Duong = N'Dinh Tien Hoang' AND Quan = N'Quan Binh Thanh'

SELECT \* FROM Nha WHERE Duong = N'Le Van Duyet' AND Quan = N'Quan Binh Thanh'

END TRY

BEGIN CATCH

print 'Loi tim kiem nha'

Rollback Transaction

END CATCH

COMMIT TRANSACTION

GO

EXEC U31\_Fix

***Code demo T2\_Fix:***

/\*U12\*/

Use QuanLyNha

go

IF OBJECT\_ID('U32\_Fix') IS NOT NULL

DROP PROCEDURE U32\_Fix

GO

CREATE PROCEDURE U32\_Fix(@id as int)

As

Begin Transaction

If Not Exists (Select \* From NHA Where ID = @id)

Begin

Print 'Nha khong ton tai'

RollBack Transaction

End

Update Nha Set Duong = N'Dinh Tien Hoang' Where ID = @id

Update Nha Set Quan = N'Quan 1' Where ID = @id

Commit Transaction

Go

EXEC U32\_Fix 2

SELECT \* FROM Nha Where ID = 2

***Kết quả trên SQL:***

******

1. **PHANTOM**
2. **Trường hợp 1 – Thông báo số lượng nhà theo tiêu chí giá nhưng danh sách hiển thị dư số lượng**

|  |  |  |
| --- | --- | --- |
|  | **T1 – Lọc danh sách nhà theo tiêu chí giá** | **T2 – Thêm một nhà** |
| 1 | Truy xuất thông tin đếm số lượng nhà có giá nhà từ 500000 đến 1000000 bằng hàm COUNT => Kết quả 2 nhà |  |
| 2 |  | Thêm một nhà có giá 550000 vào cơ sở dữ liệu |
| 3 | Truy xuất cơ sở dữ liệu hiện thị thông tin của tất cả nhà trong khoảng giá 500000 đến 1000000 => Hiển thị 3 nhà. |  |
| 4 | **Commit** |  |
| 5 |  | **Commit** |

Trong trường hợp trên, một dòng dữ liệu mới khi hiển thị là 1 PHANTOM bởi cùng thao tác truy xuất đến bộ dữ liệu nhà, ta có 2 danh sách khác nhau: 1 danh sách 2 kết quả và 1 danh sách 3 kết quả.

***Code demo T1:***

Use QuanLyNha

go

IF OBJECT\_ID('P11') IS NOT NULL

DROP PROCEDURE P11

GO

IF EXISTS(SELECT \* FROM Nha WHERE ID = 10)

BEGIN

DELETE FROM Nha WHERE ID = 10

END

GO

CREATE PROCEDURE P11(@from as int, @to as int)

As

Begin Transaction

Begin Try

SELECT COUNT(\*) AS SoLuongNha From Nha WHERE Gia >= @from AND Gia <= @to

WAITFOR DELAY '00:00:05'

SELECT \* FROM Nha WHERE Gia >= @from AND Gia <= @to

End Try

Begin Catch

Print 'Loi tim nha trong P12'

Rollback Transaction

RETURN

End Catch

Commit Transaction

Go

EXEC P11 500000, 1000000

***Code demo T2:***

Use QuanLyNha

go

IF OBJECT\_ID('P12') IS NOT NULL

DROP PROCEDURE P12

GO

CREATE PROCEDURE P12(@id as int, @Duong as nvarchar, @Quan as nvarchar, @Thanh\_Pho as nvarchar, @So\_Phong as int, @Ma\_Loai as int, @Ma\_CSH as int, @Ngay\_Het\_Han as date, @Dieu\_Kien as nvarchar, @Gia as int)

As

IF EXISTS(SELECT \* FROM Nha WHERE ID = @id)

BEGIN

DELETE FROM Nha WHERE ID = @id

END

Begin Transaction

Begin Try

INSERT INTO Nha (ID, Duong, Quan, Thanh\_Pho, So\_Phong, Ma\_Loai, Ma\_CSH, Ngay\_Dang, Ngay\_het\_han, Dieu\_Kien, Gia, Tinh\_Trang)

VALUES(@id, @Duong, @Quan, @Thanh\_Pho, @So\_Phong, @Ma\_Loai, @Ma\_CSH, GETDATE(), @Ngay\_Het\_Han, @Dieu\_Kien, @Gia, N'CÒN TRỐNG')

End Try

Begin Catch

SELECT ERROR\_MESSAGE() As ErrorMessage

Rollback Transaction

RETURN

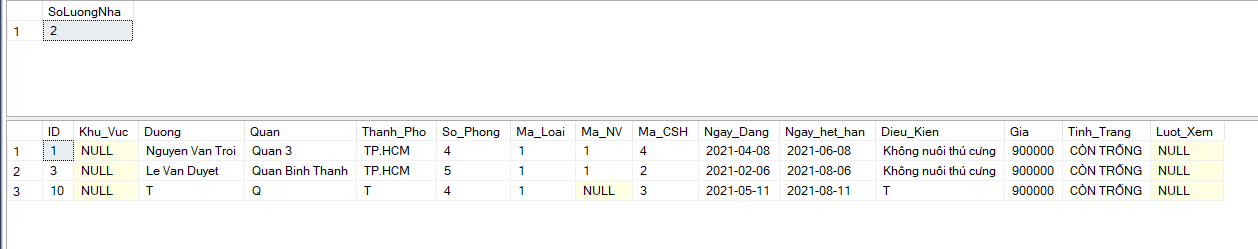
End Catch

Commit Transaction

Go

EXEC P12 10, N'Ton Dan', N'Quan 4', N'TP.HCM', 4, 1, 3, '08-11-2021', N'Tự do', 900000

***Kết quả trên SQL:***

******

* **SỬA LỖI:** Dùng mức cô lập **SERIALIZABLE** (ngăn không cho insert khi đang đọc)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Lọc danh sách nhà theo tiêu chí số lượng phòng** | **Khóa** | **Khóa** | **T2 – Xóa một nhà** |
| 1 | Truy xuất thông tin đếm số lượng nhà có giá nhà từ 500000 đến 1000000 bằng hàm COUNT => Kết quả 2 nhà | **S** |  |  |
| 2 |  | X: không được vì SQL ngăn chặn thêm dữ liệu | Thêm một nhà có giá 550000 vào cơ sở dữ liệu |
| 3 | Truy xuất cơ sở dữ liệu hiện thị thông tin của tất cả nhà có số phòng giá từ 500000 đến 1000000 => Hiển thị 2 nhà. | Đợi… |  |
| 4 | **Commit** | Đợi… |  |
| 5 |  |  | **X** | Thêm một nhà có giá 550000 vào cơ sở dữ liệu |
| 6 |  |  |  | **Commit** |

Như vậy, danh sách và số lượng nhà tại thời điểm thực hiện T1 sẽ đồng nhất.

***Code demo T1\_Fix:***

Use QuanLyNha

go

IF OBJECT\_ID('P11\_Fix') IS NOT NULL

DROP PROCEDURE P11\_Fix

GO

IF EXISTS(SELECT \* FROM Nha WHERE ID = 10)

BEGIN

DELETE FROM Nha WHERE ID = 10

END

GO

CREATE PROCEDURE P11\_Fix(@from as int, @to as int)

As

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE

Begin Transaction

Begin Try

SELECT COUNT(\*) AS SoLuongNha From Nha WHERE Gia >= @from AND Gia <= @to

WAITFOR DELAY '00:00:05'

SELECT \* FROM Nha WHERE Gia >= @from AND Gia <= @to

End Try

Begin Catch

Print 'Loi tim nha trong P12'

Rollback Transaction

RETURN

End Catch

Commit Transaction

Go

EXEC P11\_Fix 500000, 1000000

***Code demo T2\_Fix:***

Use QuanLyNha

go

IF OBJECT\_ID('P12\_Fix') IS NOT NULL

DROP PROCEDURE P12\_Fix

GO

CREATE PROCEDURE P12\_Fix(@id as int, @Duong as nvarchar, @Quan as nvarchar, @Thanh\_Pho as nvarchar, @So\_Phong as int, @Ma\_Loai as int, @Ma\_CSH as int, @Ngay\_Het\_Han as date, @Dieu\_Kien as nvarchar, @Gia as int)

As

IF EXISTS(SELECT \* FROM Nha WHERE ID = @id)

BEGIN

DELETE FROM Nha WHERE ID = @id

END

Begin Transaction

Begin Try

INSERT INTO Nha (ID, Duong, Quan, Thanh\_Pho, So\_Phong, Ma\_Loai, Ma\_CSH, Ngay\_Dang, Ngay\_het\_han, Dieu\_Kien, Gia, Tinh\_Trang)

VALUES(@id, @Duong, @Quan, @Thanh\_Pho, @So\_Phong, @Ma\_Loai, @Ma\_CSH, GETDATE(), @Ngay\_Het\_Han, @Dieu\_Kien, @Gia, N'CÒN TRỐNG')

End Try

Begin Catch

SELECT ERROR\_MESSAGE() As ErrorMessage

Rollback Transaction

RETURN

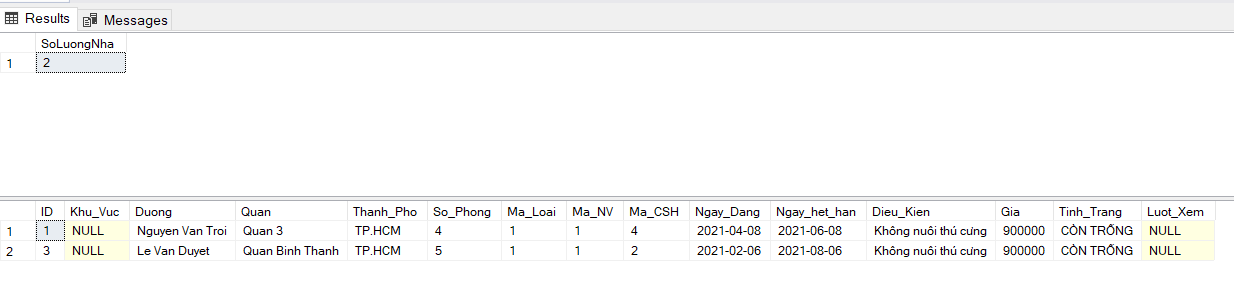
End Catch

Commit Transaction

Go

EXEC P12\_Fix 10, N'Ton Dan', N'Quan 4', N'TP.HCM', 4, 1, 3, '08-11-2021', N'Tự do', 900000

***Kết quả trên SQL:***

******

1. **Trường hợp 2 – Cập nhật thông tin nhân viên phụ trách cho các ngôi nhà mới theo địa chỉ để thuận tiện cho việc hướng dẫn khách xem nhà, trong khi đó, một nhà mới được khách hàng thêm vào => nhà mới cũng được update nhân viên phụ trách nhưng địa chỉ không phù hợp**

|  |  |  |
| --- | --- | --- |
|  | **T1 – Thêm nhân viên phụ trách cho nhà mới** | **T2 – Thêm nhà** |
| 1 | Duyệt danh sách các nhà chưa có nhân viên phụ trách, kiểm tra địa chỉ => 2 nhà quận 3 |  |
| 2 |  | Khách hàng thêm một nhà mới ở quận 4 |
| 3 | Chọn một nhân viên có địa chỉ quận 3 |  |
| 4 |  | **Commit** |
| 5 | Cập nhật nhân viên phụ trách cho **các nhà chưa có nhân viên phụ trách** => có **3 sự thay đổi**. |  |
| 6 | **Commit** |  |

Trường hợp này không gây ra lỗi chương trình, tuy nhiên, ban đầu quản trị viên mong muốn xếp nhân viên **5** ở quận 3 phụ trách nhà ở quận 3 để thuận tiện về địa lý.

Tuy nhiên, sự thêm một nhà ở quận 4 đã khiến quản trị viên này phân công nhân viên **5** phụ trách luôn nhà mới => bất cập về khoảng cách và vấn đề xem nhà.

***Code demo T1:***

Use QuanLyNha

go

IF OBJECT\_ID('P21') IS NOT NULL

DROP PROCEDURE P21

GO

IF EXISTS(SELECT \* FROM Nha WHERE ID = 10)

BEGIN

DELETE FROM Nha WHERE ID = 10

END

GO

UPDATE Nha SET Ma\_NV = NULL WHERE ID = 8 OR ID = 9

GO

CREATE PROCEDURE P21

As

Begin Transaction

Begin Try

DECLARE @MaNV as int

SELECT \* From Nha

SELECT \* From Nha WHERE Ma\_NV IS NULL

SELECT TOP 1 @MaNV = ID FROM NhanVien WHERE Dia\_Chi = N'Quan 3'

WAITFOR DELAY '00:00:05'

UPDATE Nha SET Ma\_NV = @MaNV WHERE Ma\_NV IS NULL

SELECT \* FROM Nha

End Try

Begin Catch

Print 'Loi tim nha trong P21'

Rollback Transaction

RETURN

End Catch

Commit Transaction

Go

EXEC P21

***Code demo T2:***

Use QuanLyNha

go

IF OBJECT\_ID('P22') IS NOT NULL

DROP PROCEDURE P22

GO

CREATE PROCEDURE P22(@id as int, @Duong as nvarchar(50), @Quan as nvarchar(50), @Thanh\_Pho as nvarchar(50), @So\_Phong as int, @Ma\_Loai as int, @Ma\_CSH as int, @Ngay\_Het\_Han as date, @Dieu\_Kien nvarchar(50), @Gia as int)

As

IF EXISTS(SELECT \* FROM Nha WHERE ID = @id)

BEGIN

DELETE FROM Nha WHERE ID = @id

END

Begin Transaction

Begin Try

INSERT INTO Nha (ID, Duong, Quan, Thanh\_Pho, So\_Phong, Ma\_Loai, Ma\_CSH, Ngay\_Dang, Ngay\_het\_han, Dieu\_Kien, Gia, Tinh\_Trang)

VALUES(@id, @Duong, @Quan, @Thanh\_Pho, @So\_Phong, @Ma\_Loai, @Ma\_CSH, GETDATE(), @Ngay\_Het\_Han, @Dieu\_Kien, @Gia, N'CÒN TRỐNG')

End Try

Begin Catch

SELECT ERROR\_MESSAGE() As ErrorMessage

Rollback Transaction

RETURN

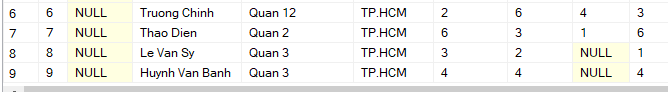
End Catch

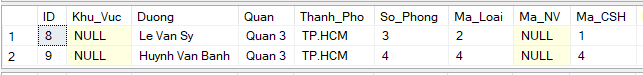
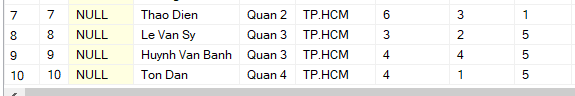
Commit Transaction

Go

EXEC P22 10, N'Ton Dan', N'Quan 4', N'TP.HCM', 4, 1, 3, '08-11-2021', N'Tự do', 900000

***Kết quả trên SQL:***

******

******

* **SỬA LỖI:** SỬ DỤNG MỨC CÔ LẬP **SERIALIZABLE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Thêm nhân viên phụ trách cho nhà mới** | **Khoá** | **Khoá** | **T2 – Thêm nhà** |
| 1 | Duyệt danh sách các nhà chưa có nhân viên phụ trách, kiểm tra địa chỉ => 2 nhà quận 3 | **S** |  |  |
| 2 |  |  | Xin khoá X => không được vì T1 giữ khoá S | Khách hàng thêm một nhà mới ở quận 4 |
| 3 | Chọn một nhân viên có địa chỉ quận 3 |  | Đợi |  |
| 4 |  |  |  |  |
| 5 | Cập nhật nhân viên phụ trách cho **các nhà chưa có nhân viên phụ trách** => có **2 sự thay đổi**. |  |  |  |
| 6 | **Commit** |  | X | Khách hàng thêm một nhà mới ở quận 4 |
| 7 |  |  |  | **Commit** |

Như vậy, danh sách nhà được cập nhật nhân viên phụ trách sẽ không còn bị sót nhà mới thêm vào nữa.

***Code demo T1\_Fix:***

Use QuanLyNha

go

IF OBJECT\_ID('P21\_Fix') IS NOT NULL

DROP PROCEDURE P21\_Fix

GO

IF EXISTS(SELECT \* FROM Nha WHERE ID = 10)

BEGIN

DELETE FROM Nha WHERE ID = 10

END

GO

UPDATE Nha SET Ma\_NV = NULL WHERE ID = 8 OR ID = 9

GO

CREATE PROCEDURE P21\_Fix

As

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE

Begin Transaction

Begin Try

DECLARE @MaNV as int

SELECT \* From Nha

SELECT \* From Nha WHERE Ma\_NV IS NULL

SELECT TOP 1 @MaNV = ID FROM NhanVien WHERE Dia\_Chi = N'Quan 3'

WAITFOR DELAY '00:00:05'

UPDATE Nha SET Ma\_NV = @MaNV WHERE Ma\_NV IS NULL

SELECT \* FROM Nha

End Try

Begin Catch

Print 'Loi tim nha trong P21'

Rollback Transaction

RETURN

End Catch

Commit Transaction

Go

EXEC P21\_Fix

***Code demo T2\_Fix:***

Use QuanLyNha

go

IF OBJECT\_ID('P22\_Fix') IS NOT NULL

DROP PROCEDURE P22\_Fix

GO

CREATE PROCEDURE P22\_Fix(@id as int, @Duong as nvarchar(50), @Quan as nvarchar(50), @Thanh\_Pho as nvarchar(50), @So\_Phong as int, @Ma\_Loai as int, @Ma\_CSH as int, @Ngay\_Het\_Han as date, @Dieu\_Kien nvarchar(50), @Gia as int)

As

IF EXISTS(SELECT \* FROM Nha WHERE ID = @id)

BEGIN

DELETE FROM Nha WHERE ID = @id

END

Begin Transaction

Begin Try

INSERT INTO Nha (ID, Duong, Quan, Thanh\_Pho, So\_Phong, Ma\_Loai, Ma\_CSH, Ngay\_Dang, Ngay\_het\_han, Dieu\_Kien, Gia, Tinh\_Trang)

VALUES(@id, @Duong, @Quan, @Thanh\_Pho, @So\_Phong, @Ma\_Loai, @Ma\_CSH, GETDATE(), @Ngay\_Het\_Han, @Dieu\_Kien, @Gia, N'CÒN TRỐNG')

End Try

Begin Catch

SELECT ERROR\_MESSAGE() As ErrorMessage

Rollback Transaction

RETURN

End Catch

Commit Transaction

Go

EXEC P22\_Fix 10, N'Ton Dan', N'Quan 4', N'TP.HCM', 4, 1, 3, '08-11-2021', N'Tự do', 900000

***Kết quả trên SQL\_Fix:***

******

1. **Trường hợp 3 – Nhân viên đọc danh sách các khách hàng theo tiêu chí xác định và liên hệ, sau đó một khách hàng khác được thêm vào với cùng tiêu chí khiến nhân viên bỏ sót khách hàng mới**

|  |  |  |
| --- | --- | --- |
|  | **T1 – Đọc danh sách khách hàng theo tiêu chí** | **T2 – Thêm khách hàng** |
| 1 | Nhân viên đọc danh sách các khách hàng theo tiêu chí số lượng phòng là 3, quận 1 => danh sách 2 người. |  |
| 2 |  | Quản trị viên thêm khách hàng mới với tiêu chí số phòng = 3, quận 1 |
| 3 | Nhân viên update lại danh sách các khách hàng yêu cầu số lượng phòng = 1và quận 1 là Đã liên hệ |  |
| 4 |  | **Commit** |
| 5 | **Commit** |  |

Trong trường hợp trên, do lần đọc 1, nhân viên chỉ đọc được 2 khách hàng nên đã liên hệ với 2khách. Nhưng ở lần đọc 2, để update danh sách lại có đến **3** khách hàng cùng tiêu chí, do đó việc đánh dấu “**ĐÃ LIÊN HỆ (1)**” sẽ chèn lên ***khách hàng mới*** vừa được thêm vào, dẫn đến bỏ sót khách hàng này.

***Code demo T1:***

Use QuanLyNha

go

IF OBJECT\_ID('P31') IS NOT NULL

DROP PROCEDURE P31

GO

IF EXISTS(SELECT \* FROM KhachHang WHERE ID = 5)

BEGIN

DELETE FROM KhachHang WHERE ID = 5

END

GO

UPDATE KhachHang SET Da\_Lien\_He = 0 WHERE ID = 1 OR ID = 3

GO

CREATE PROCEDURE P31

As

Begin Transaction

Begin Try

DECLARE @MaNV as int

SELECT ID, Ho\_Ten, Dien\_Thoai, Da\_Lien\_He From KhachHang WHERE TieuChi\_SoPhong = 3 AND TieuChi\_Quan = N'Quan 1'

WAITFOR DELAY '00:00:05'

UPDATE KhachHang SET Da\_Lien\_He = 1 WHERE TieuChi\_SoPhong = 3 AND TieuChi\_Quan = N'Quan 1'

SELECT ID, Ho\_Ten, Dien\_Thoai, Da\_Lien\_He FROM KhachHang WHERE TieuChi\_SoPhong = 3 AND TieuChi\_Quan = N'Quan 1'

End Try

Begin Catch

Print 'Loi tim nha trong P31'

Rollback Transaction

RETURN

End Catch

Commit Transaction

Go

EXEC P31

***Code demo T2:***

Use QuanLyNha

go

IF OBJECT\_ID('P32') IS NOT NULL

DROP PROCEDURE P32

GO

CREATE PROCEDURE P32(@id as int, @HoTen as nvarchar(50), @DiaChi as nvarchar(50), @DienThoai as char(10), @So\_Phong as int, @Duong as nvarchar(50), @Quan as nvarchar(50), @ThanhPho as nvarchar(50), @nhuCau as bit)

As

IF EXISTS(SELECT \* FROM KhachHang WHERE ID = @id)

BEGIN

DELETE FROM KhachHang WHERE ID = @id

END

Begin Transaction

Begin Try

INSERT INTO KhachHang(ID, Ho\_Ten, Dia\_Chi, Dien\_Thoai, TieuChi\_SoPhong, TieuChi\_Duong, TieuChi\_Quan, TieuChi\_ThanhPhi, Thue\_Mua, Da\_Lien\_He)

VALUES(@id, @HoTen, @DiaChi, @DienThoai, @So\_Phong, @Duong, @Quan, @ThanhPho, @nhuCau, 0)

End Try

Begin Catch

SELECT ERROR\_MESSAGE() As ErrorMessage

Rollback Transaction

RETURN

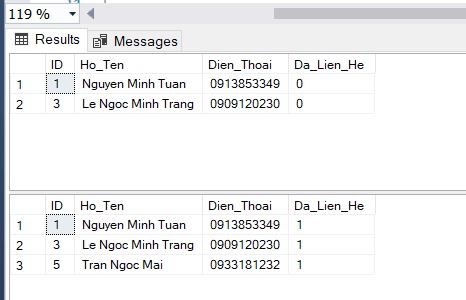
End Catch

Commit Transaction

Go

EXEC P32 5, N'Tran Ngoc Mai', N'Quan 7', '0933181232', 3, N'Hai Ba Trung', N'Quan 1', N'TP.HCM', 1

***Kết quả trên SQL:***

******

* **SỬA LỖI:** SỬ DỤNG MỨC CÔ LẬP **SERIALIZABLE** (ngăn không cho Insert khi đang đọc)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Đọc danh sách khách hàng theo tiêu chí** | **Khoá** | **Khoá** | **T2 – Thêm khách hàng** |
| 1 | Nhân viên đọc danh sách các khách hàng theo tiêu chí số lượng phòng là 3, quận 1 => danh sách 2 người. | Khoá S |  |  |
| 2 |  |  | Xin Khoá X => Không được vì T1 đang giữ khoá S | Quản trị viên thêm khách hàng mới với tiêu chí số phòng = 3, quận 1 |
| 3 | Nhân viên update lại danh sách các khách hàng yêu cầu số lượng phòng = 1 và quận 1 là Đã liên hệ => chỉ update 2 người | S |  |  |
| 4 | **Commit** |  |  |  |
| 5 |  |  | X | Quản trị viên thêm khách hàng mới với tiêu chí số phòng = 3, quận 1 |
| 6 |  |  |  | **Commit** |

Như vậy, nhân viên chỉ update đúng số lượng người đã liên hệ, nhà mới thêm vào sẽ vẫn ở tình trạng chưa có nhân viên phụ trách và có thể thêm ở lần sau.

***Code demo T1\_Fix:***

Use QuanLyNha

go

IF OBJECT\_ID('P31\_Fix') IS NOT NULL

DROP PROCEDURE P31\_Fix

GO

IF EXISTS(SELECT \* FROM KhachHang WHERE ID = 5)

BEGIN

DELETE FROM KhachHang WHERE ID = 5

END

GO

UPDATE KhachHang SET Da\_Lien\_He = 0 WHERE ID = 1 OR ID = 3

GO

CREATE PROCEDURE P31\_Fix

As

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE

Begin Transaction

Begin Try

DECLARE @MaNV as int

SELECT ID, Ho\_Ten, Dien\_Thoai, Da\_Lien\_He From KhachHang WHERE TieuChi\_SoPhong = 3 AND TieuChi\_Quan = N'Quan 1'

WAITFOR DELAY '00:00:05'

UPDATE KhachHang SET Da\_Lien\_He = 1 WHERE TieuChi\_SoPhong = 3 AND TieuChi\_Quan = N'Quan 1'

SELECT ID, Ho\_Ten, Dien\_Thoai, Da\_Lien\_He FROM KhachHang WHERE TieuChi\_SoPhong = 3 AND TieuChi\_Quan = N'Quan 1'

End Try

Begin Catch

Print 'Loi tim nha trong P31'

Rollback Transaction

RETURN

End Catch

Commit Transaction

Go

EXEC P31\_Fix

***Code demo T2\_Fix:***

Use QuanLyNha

go

IF OBJECT\_ID('P32\_Fix') IS NOT NULL

DROP PROCEDURE P32\_Fix

GO

CREATE PROCEDURE P32\_Fix(@id as int, @HoTen as nvarchar(50), @DiaChi as nvarchar(50), @DienThoai as char(10), @So\_Phong as int, @Duong as nvarchar(50), @Quan as nvarchar(50), @ThanhPho as nvarchar(50), @nhuCau as bit)

As

IF EXISTS(SELECT \* FROM KhachHang WHERE ID = @id)

BEGIN

DELETE FROM KhachHang WHERE ID = @id

END

Begin Transaction

Begin Try

INSERT INTO KhachHang(ID, Ho\_Ten, Dia\_Chi, Dien\_Thoai, TieuChi\_SoPhong, TieuChi\_Duong, TieuChi\_Quan, TieuChi\_ThanhPhi, Thue\_Mua, Da\_Lien\_He)

VALUES(@id, @HoTen, @DiaChi, @DienThoai, @So\_Phong, @Duong, @Quan, @ThanhPho, @nhuCau, 0)

End Try

Begin Catch

SELECT ERROR\_MESSAGE() As ErrorMessage

Rollback Transaction

RETURN

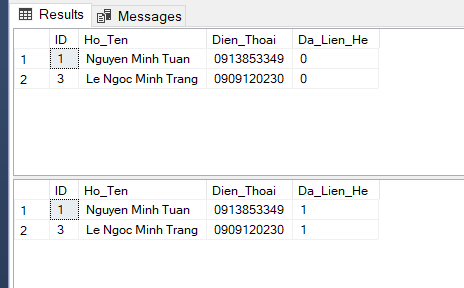
End Catch

Commit Transaction

Go

EXEC P32\_Fix 5, N'Tran Ngoc Mai', N'Quan 7', '0933181232', 3, N'Hai Ba Trung', N'Quan 1', N'TP.HCM', 1

***Kết quả trên SQL:***

******

1. **LOST UPDATE**
2. **Trường hợp 1 – Lỗi cập nhật lượt xem**

|  |  |  |
| --- | --- | --- |
|  | **T1 – Tạo ngày xem nhà cho nhà 2** | **T2 – Tạo ngày xem nhà cho nhà 2** |
| 1 | Tạo một ngày hẹn trong bảng Chi Tiết Xem Nhà cho nhà **2** |  |
| 2 | Đọc số lượt xem hiện tại của nhà 2 => A = 0 |  |
| 3 |  | Tạo một ngày hẹn trong bảng Chi Tiết Xem Nhà cho nhà **2** |
| 4 |  | Đọc số lượt xem hiện tại của nhà 2 => A = 0 |
| 5 |  | Cập nhật A = A + 1 |
| 6 | Cập nhật A = A + 1 |  |
| 7 | Update số lượt xem của nhà 2 là A = 1 |  |
| 8 |  | Update số lượt xem của nhà 2 là A = 1 |
| 9 | **Commit** | **Commit** |

Trong trường hợp trên, số lượt xem của nhà 2 phải tăng thêm 2 (do tạo 2 lượt xem nhà), tuy nhiên, do tình huống ghi đè của T2 đã dẫn đến việc nhà **2** chỉ tăng một lượt xem.

***Code demo T1:***

Use QuanLyNha

go

IF OBJECT\_ID('L11') IS NOT NULL

DROP PROCEDURE L11

GO

CREATE PROCEDURE L11(@Ma\_Nha as int, @Ma\_KH as int, @Ma\_NV as int, @ngay AS DATE)

As

Begin Transaction

IF NOT EXISTS(SELECT \* FROM Nha WHERE ID = @Ma\_Nha)

BEGIN

PRINT 'Nha khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM NhanVien WHERE ID = @Ma\_NV)

BEGIN

PRINT 'Nhan vien khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM KhachHang WHERE ID = @Ma\_KH)

BEGIN

PRINT 'Khach hang khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

Begin Try

INSERT INTO ChiTiet\_XemNha (Ma\_KH, Ma\_Nha, MaNV, Ngay) VALUES(@Ma\_KH, @Ma\_Nha, @Ma\_NV, @ngay)

DECLARE @luotxem as int

SELECT @luotxem = Luot\_Xem FROM Nha WHERE ID = @Ma\_Nha

WAITFOR DELAY '00:00:05'

SET @luotxem = @luotxem + 1

UPDATE Nha Set Luot\_Xem = @luotxem WHERE ID = @Ma\_Nha

End Try

Begin Catch

Print 'Loi THEM LUOT XEM trong L11'

Rollback Transaction

RETURN

End Catch

Commit Transaction

Go

DELETE FROM ChiTiet\_XemNha WHERE Ma\_Nha = 2

UPDATE Nha SET Luot\_Xem = 0 WHERE ID = 2

EXEC L11 2, 1, 1, '06/08/2021'

SELECT ID,Luot\_Xem FROM Nha WHERE ID = 2

SELECT \* FROM ChiTiet\_XemNha WHERE Ma\_Nha = 2

***Code demo T2:***

Use QuanLyNha

go

IF OBJECT\_ID('L12') IS NOT NULL

DROP PROCEDURE L12

GO

CREATE PROCEDURE L12(@Ma\_Nha as int, @Ma\_KH as int, @Ma\_NV as int, @ngay AS DATE)

As

Begin Transaction

IF NOT EXISTS(SELECT \* FROM Nha WHERE ID = @Ma\_Nha)

BEGIN

PRINT 'Nha khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM NhanVien WHERE ID = @Ma\_NV)

BEGIN

PRINT 'Nhan vien khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM KhachHang WHERE ID = @Ma\_KH)

BEGIN

PRINT 'Khach hang khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

Begin Try

INSERT INTO ChiTiet\_XemNha (Ma\_KH, Ma\_Nha, MaNV, Ngay) VALUES(@Ma\_KH, @Ma\_Nha, @Ma\_NV, @ngay)

DECLARE @luotxem as int

SELECT @luotxem = Luot\_Xem FROM Nha WHERE ID = @Ma\_Nha

SET @luotxem = @luotxem + 1

UPDATE Nha Set Luot\_Xem = @luotxem WHERE ID = @Ma\_Nha

End Try

Begin Catch

Print 'Loi THEM LUOT XEM trong L11'

Rollback Transaction

RETURN

End Catch

Commit Transaction

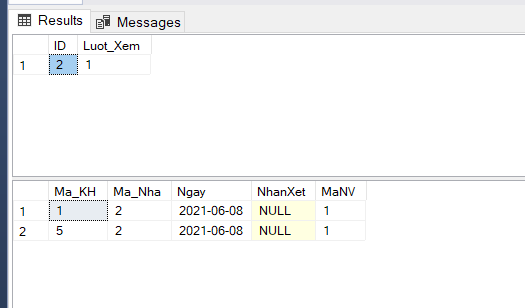
Go

EXEC L12 2, 5, 1, '06/08/2021'

SELECT ID,Luot\_Xem FROM Nha WHERE ID = 2

SELECT \* FROM ChiTiet\_XemNha WHERE Ma\_Nha = 2

***Kết quả trên SQL:***

******

* **SỬA LỖI:** Sử dụng mức cô lập **REPEATABLE READ** (không cho thay đổi khi đang đọc)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Tạo ngày xem nhà cho nhà 2** | **Khoá** | **Khoá** | **T2 – Tạo ngày xem nhà cho nhà 2** |
| 1 | Tạo một ngày hẹn trong bảng Chi Tiết Xem Nhà cho nhà **2** | *Đơn vị ChiTiet\_XemNha => không xét* |  |  |
| 2 | Đọc số lượt xem hiện tại của nhà 2 => A = 0 | **Xin khoá S trên đơn vị Nhà** |  |  |
| 3 |  |  | *Đơn vị ChiTiet\_XemNha => không xét* | Tạo một ngày hẹn trong bảng Chi Tiết Xem Nhà cho nhà **2** |
| 4 |  |  | **Xin khoá S trên đơn vị Nhà** | Đọc số lượt xem hiện tại của nhà 2 => A = 0 |
| 5 |  | **DEADLOCK => Huỷ giao tác 2** |  | Cập nhật A = A + 1 |
| 6 | Cập nhật A = A + 1 |  |  |  |
| 7 | Update số lượt xem của nhà 2 là A = 1 | Xin khoá X => không được vì T2 đang giữ khoá S |  |  |
| 8 | **Update số lượt xem của nhà 2 là A = 1** | **Xin khoá X** | Xin khoá X => không được vì T1 đang giữ khoá S | Update số lượt xem của nhà 2 là A = 1 |
| 9 | **Commit** |  |  | **Commit** |

Như vậy, do cơ chế khoá, DEADLOCK đã xuất hiện và hệ quản trị sẽ huỷ T2 (hoặc T1) để có thể tiếp tục thực thi, đồng thời đảm bảo tính thống nhất dữ liệu.

***Code demo T1\_Fix***

Use QuanLyNha

go

IF OBJECT\_ID('L11\_Fix') IS NOT NULL

DROP PROCEDURE L11\_Fix

GO

CREATE PROCEDURE L11\_Fix(@Ma\_Nha as int, @Ma\_KH as int, @Ma\_NV as int, @ngay AS DATE)

As

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

Begin Transaction

IF NOT EXISTS(SELECT \* FROM Nha WHERE ID = @Ma\_Nha)

BEGIN

PRINT 'Nha khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM NhanVien WHERE ID = @Ma\_NV)

BEGIN

PRINT 'Nhan vien khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM KhachHang WHERE ID = @Ma\_KH)

BEGIN

PRINT 'Khach hang khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

Begin Try

INSERT INTO ChiTiet\_XemNha (Ma\_KH, Ma\_Nha, MaNV, Ngay) VALUES(@Ma\_KH, @Ma\_Nha, @Ma\_NV, @ngay)

DECLARE @luotxem as int

SELECT @luotxem = Luot\_Xem FROM Nha WHERE ID = @Ma\_Nha

WAITFOR DELAY '00:00:05'

SET @luotxem = @luotxem + 1

UPDATE Nha Set Luot\_Xem = @luotxem WHERE ID = @Ma\_Nha

End Try

Begin Catch

Print 'Loi THEM LUOT XEM trong L11'

Rollback Transaction

RETURN

End Catch

Commit Transaction

Go

DELETE FROM ChiTiet\_XemNha WHERE Ma\_Nha = 2

UPDATE Nha SET Luot\_Xem = 0 WHERE ID = 2

EXEC L11\_Fix 2, 1, 1, '06/08/2021'

SELECT ID,Luot\_Xem FROM Nha WHERE ID = 2

SELECT \* FROM ChiTiet\_XemNha WHERE Ma\_Nha = 2

***Code demo T2\_Fix***

Use QuanLyNha

go

IF OBJECT\_ID('L12\_Fix') IS NOT NULL

DROP PROCEDURE L12\_Fix

GO

CREATE PROCEDURE L12\_Fix(@Ma\_Nha as int, @Ma\_KH as int, @Ma\_NV as int, @ngay AS DATE)

As

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

Begin Transaction

IF NOT EXISTS(SELECT \* FROM Nha WHERE ID = @Ma\_Nha)

BEGIN

PRINT 'Nha khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM NhanVien WHERE ID = @Ma\_NV)

BEGIN

PRINT 'Nhan vien khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM KhachHang WHERE ID = @Ma\_KH)

BEGIN

PRINT 'Khach hang khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

Begin Try

INSERT INTO ChiTiet\_XemNha (Ma\_KH, Ma\_Nha, MaNV, Ngay) VALUES(@Ma\_KH, @Ma\_Nha, @Ma\_NV, @ngay)

DECLARE @luotxem as int

SELECT @luotxem = Luot\_Xem FROM Nha WHERE ID = @Ma\_Nha

SET @luotxem = @luotxem + 1

UPDATE Nha Set Luot\_Xem = @luotxem WHERE ID = @Ma\_Nha

End Try

Begin Catch

Print 'Loi THEM LUOT XEM trong L11'

Rollback Transaction

RETURN

End Catch

Commit Transaction

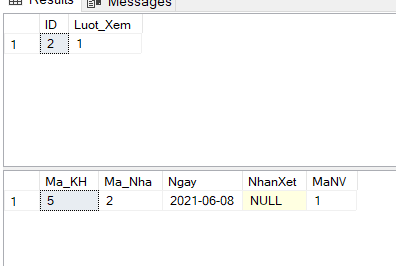
Go

EXEC L12\_Fix 2, 5, 1, '06/08/2021'

SELECT ID,Luot\_Xem FROM Nha WHERE ID = 2

SELECT \* FROM ChiTiet\_XemNha WHERE Ma\_Nha = 2

***Kết quả trên SQL***

******

1. **Trường hợp 2 – Tăng giá trị các nhà ở quận 1 cộng với 20% giá trị hiện tại, nếu giá thuê sau khi tăng trên 3.000.000 đồng thì cộng thêm 5% phí dịch vụ. Trong khi đó, giá trị của một nhà ở quận 1 được tăng thêm 1.000.000 đồng => giá thuê sau cập nhật mất tính đúng đắn**

|  |  |  |
| --- | --- | --- |
|  | **T1 – Tăng giá trị thuê của mỗi nhà quận 1 thêm 20%** | **T2 – Thay đổi giá trị của một nhà ở quận 1** |
| 1 | Tìm tất cả các nhà ở quận 3 |  |
| 2 | Tăng 20% giá nhà  A = 1.2A |  |
| 3 |  | Tăng giá trị nhà:  A = A + 1000 |
| 4 | Nếu A > 3000: A = A\*1.05 |  |
| 5 |  | **Commit** |
| 6 | **Commit** |  |

Như vậy, giả sử giá nhà là 8500, nếu như thực hiện đúng và tuần tự, ví dụ T1 trước T2, thì giá nhà phải là 11710 đồng hoặc nếu T2 trước T1, giá nhà phải là 11970. Nhưng kết quả hiển thị khác so với 2 thứ tự này. Điều này sẽ khiến tính đồng nhất về dữ liệu giá nhà bị ảnh hưởng.

***Code demo T1:***

Use QuanLyNha

go

IF OBJECT\_ID('L21') IS NOT NULL

DROP PROCEDURE L21

GO

CREATE PROCEDURE L21(@QUAN as nvarchar(50))

As

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

Begin Transaction

Begin Try

UPDATE Nha SET Gia = 1.2\*Gia WHERE Quan = @QUAN AND Dieu\_Kien IS NULL

WAITFOR DELAY '00:00:15'

UPDATE Nha SET Gia = 1.05\*Gia WHERE Gia > 3000 AND Dieu\_Kien IS NULL

End Try

Begin Catch

Print 'Loi CAP NHAT GIA trong L21'

Rollback Transaction

RETURN

End Catch

Commit Transaction

Go

UPDATE Nha SET Gia = 8500 WHERE ID = 9

EXEC L21 N'Quan 3'

***Code demo T2:***

Use QuanLyNha

go

IF OBJECT\_ID('L22') IS NOT NULL

DROP PROCEDURE L22

GO

CREATE PROCEDURE L22(@Ma\_Nha as int, @gia as int)

As

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

Begin Transaction

If Not Exists (Select \* From NHA Where ID = @Ma\_Nha)

Begin

Print 'Nha khong ton tai'

RollBack Transaction

End

Begin Try

Declare @GiaCapNhat int

Select @GiaCapNhat = Gia From Nha Where ID = @Ma\_Nha

PRINT @GiaCapNhat

Update Nha Set Gia = @GiaCapNhat + @gia Where ID = @Ma\_Nha

End Try

Begin Catch

Print 'Loi update gia nha trong L22'

Rollback Transaction

End Catch

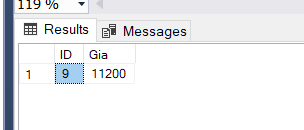
Commit Transaction

Go

EXEC L22 9, 1000

SELECT ID,Gia FROM Nha WHERE ID = 9

***Kết quả trên SQL:***

******

* **SỬA LỖI: Sử dụng mức cô lập Read Committed (phát khoá X và giữ đến cuối giao tác).**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Tăng giá trị thuê của mỗi nhà quận 1 thêm 20%** | **Khóa** | **Khóa** | **T2 – Thay đổi giá trị của một nhà ở quận 1** |
| 1 | Tìm tất cả các nhà ở quận 1 | **X** |  |  |
| 2 | Tăng 20% giá nhà  A = 1.2A |  |  |
| 3 |  | S => không xin được vì còn khóa X | Tìm nhà có mã 008 ở quận 1 |
| 4 | Nếu A > 3000: A = A\*1.05 | Đợi… |  |
| 5 | **Commit** | Đợi… |  |
| 6 |  |  | **S** | Tìm nhà có mã 008 ở quận 1 |
| 7 |  |  | Tăng giá nhà  A = A + 1.000.000 |
| 8 |  |  | **Commit** |

Như vậy, giả sử giá nhà là 8500, sau khi hoàn tất 2 giao tác này sẽ là 11710, tương tự như việc thực hiện tuần tự T1 => T2.

***Code demo T1\_Fix:***

Use QuanLyNha

go

IF OBJECT\_ID('L21') IS NOT NULL

DROP PROCEDURE L21

GO

CREATE PROCEDURE L21(@QUAN as nvarchar(50))

As

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

Begin Transaction

Begin Try

UPDATE Nha SET Gia = 1.2\*Gia WHERE Quan = @QUAN AND Dieu\_Kien IS NULL

WAITFOR DELAY '00:00:15'

UPDATE Nha SET Gia = 1.05\*Gia WHERE Gia > 3000 AND Dieu\_Kien IS NULL

End Try

Begin Catch

Print 'Loi CAP NHAT GIA trong L21'

Rollback Transaction

RETURN

End Catch

Commit Transaction

Go

UPDATE Nha SET Gia = 8500 WHERE ID = 9

EXEC L21 N'Quan 3'

***Code demo T2\_Fix:***

Use QuanLyNha

go

IF OBJECT\_ID('L22') IS NOT NULL

DROP PROCEDURE L22

GO

CREATE PROCEDURE L22(@Ma\_Nha as int, @gia as int)

As

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

Begin Transaction

If Not Exists (Select \* From NHA Where ID = @Ma\_Nha)

Begin

Print 'Nha khong ton tai'

RollBack Transaction

End

Begin Try

Declare @GiaCapNhat int

Select @GiaCapNhat = Gia From Nha Where ID = @Ma\_Nha

PRINT @GiaCapNhat

Update Nha Set Gia = @GiaCapNhat + @gia Where ID = @Ma\_Nha

End Try

Begin Catch

Print 'Loi update gia nha trong L22'

Rollback Transaction

End Catch

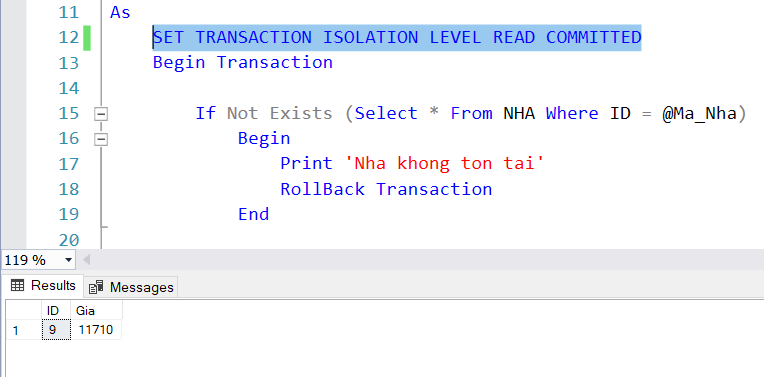
Commit Transaction

Go

EXEC L22 9, 1000

SELECT ID,Gia FROM Nha WHERE ID = 9

***Kết quả trên SQL:***

******

1. **Trường hợp 3 – Cập nhật lượt xem đồng thời, dẫn đến một khách xem cùng nhà trong một ngày hai lần, với hai nhân**

|  |  |  |
| --- | --- | --- |
|  | **T1 – Nhân viên 4 thêm lượt xem cho khách 1, nhà 3** | **T2 – Nhân viên 5 thêm lượt xem cho khách 1, nhà 3** |
| 1 | Kiểm tra Mã nhà, Mã khách và Ngày đã tồn tại chưa |  |
| 2 |  | Kiểm tra Mã nhà, Mã khách và Ngày đã tồn tại chưa |
| 3 |  | Đọc số lượt xem hiện tại của nhà 2 => A = 0 |
| 4 | Thêm lượt xem với thông tin NV 4, Khách 1, nhà 3 |  |
| 5 |  | Thêm lượt xem với thông tin NV 4, Khách 1, nhà 3 |
| 6 | Cập nhật số lượt xem + 1 cho nhà 3 trong bảng NHÀ |  |
| 7 |  | Cập nhật số lượt xem + 1 cho nhà 3 trong bảng NHÀ |
| 8 | **Commit** | **Commit** |

Trong trường hợp trên, T2 đã thực hiện kiểm tra thành phố và quận nhưng do sự xen giữa của T1, thành phố của nhà 005 đã được cập nhật thành TP.Thủ Đức trước khi thay đổi quận. Do đó, dữ liệu của nhà sẽ là **“Quận 5, thành phố Thủ Đức”** thay vì dữ liệu đúng là **“Quận 5, thành phố HCM”**.

***Code demo T1:***

Use QuanLyNha

go

IF OBJECT\_ID('L31') IS NOT NULL

DROP PROCEDURE L31

GO

CREATE PROCEDURE L31(@Ma\_Nha as int, @Ma\_KH as int, @Ma\_NV as int, @ngay AS DATE)

As

Begin Transaction

IF NOT EXISTS(SELECT \* FROM Nha WHERE ID = @Ma\_Nha)

BEGIN

PRINT 'Nha khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM NhanVien WHERE ID = @Ma\_NV)

BEGIN

PRINT 'Nhan vien khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM KhachHang WHERE ID = @Ma\_KH)

BEGIN

PRINT 'Khach hang khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

Begin Try

IF NOT EXISTS (SELECT \* FROM ChiTiet\_XemNha WHERE Ma\_KH = @Ma\_KH AND Ma\_Nha = @Ma\_Nha AND Ngay = @ngay)

BEGIN

WAITFOR DELAY '00:00:05'

INSERT INTO ChiTiet\_XemNha (Ma\_KH, Ma\_Nha, MaNV, Ngay) VALUES(@Ma\_KH, @Ma\_Nha, @Ma\_NV, @ngay)

UPDATE Nha Set Luot\_Xem = Luot\_Xem + 1 WHERE ID = @Ma\_Nha

END

End Try

Begin Catch

Print 'Loi THEM LUOT XEM trong L31'

SELECT ERROR\_MESSAGE()

Rollback Transaction

RETURN

End Catch

Commit Transaction

Go

DELETE FROM ChiTiet\_XemNha WHERE Ma\_Nha = 3

UPDATE Nha SET Luot\_Xem = 0 WHERE ID = 3

EXEC L31 3, 1, 4, '06/08/2021'

SELECT ID,Luot\_Xem FROM Nha WHERE ID = 3

SELECT \* FROM ChiTiet\_XemNha WHERE Ma\_Nha = 3

***Code demo T2:***

Use QuanLyNha

go

IF OBJECT\_ID('L32\_Fix') IS NOT NULL

DROP PROCEDURE L32\_Fix

GO

CREATE PROCEDURE L32\_Fix(@Ma\_Nha as int, @Ma\_KH as int, @Ma\_NV as int, @ngay AS DATE)

As

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

Begin Transaction

IF NOT EXISTS(SELECT \* FROM Nha WHERE ID = @Ma\_Nha)

BEGIN

PRINT 'Nha khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM NhanVien WHERE ID = @Ma\_NV)

BEGIN

PRINT 'Nhan vien khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM KhachHang WHERE ID = @Ma\_KH)

BEGIN

PRINT 'Khach hang khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

Begin Try

IF NOT EXISTS (SELECT \* FROM ChiTiet\_XemNha WHERE Ma\_KH = @Ma\_KH AND Ma\_Nha = @Ma\_Nha AND Ngay = @ngay)

BEGIN

INSERT INTO ChiTiet\_XemNha (Ma\_KH, Ma\_Nha, MaNV, Ngay) VALUES(@Ma\_KH, @Ma\_Nha, @Ma\_NV, @ngay)

UPDATE Nha Set Luot\_Xem = Luot\_Xem + 1 WHERE ID = @Ma\_Nha

END

End Try

Begin Catch

Print 'Loi THEM LUOT XEM trong L31'

Rollback Transaction

RETURN

End Catch

Commit Transaction

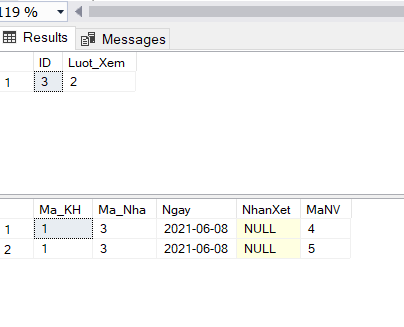
Go

EXEC L32\_Fix 3, 1, 5, '06/08/2021'

SELECT ID,Luot\_Xem FROM Nha WHERE ID = 3

SELECT \* FROM ChiTiet\_XemNha WHERE Ma\_Nha = 3

***Kết quả trên SQL:***

******

* **SỬA LỖI: Sử dụng mức cô lập REPEATABLE READ (ngăn không cho Update trên một đối tượng khi đang có đối tượng khác đọc**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Nhân viên 4 thêm lượt xem cho khách 1, nhà 3** | **Khoá** | **Khoá** | **T2 – Nhân viên 5 thêm lượt xem cho khách 1, nhà 3** |
| 1 | Kiểm tra Mã nhà, Mã khách và Ngày đã tồn tại chưa | Khoá S |  |  |
| 2 |  |  | Khoá S | Kiểm tra Mã nhà, Mã khách và Ngày đã tồn tại chưa |
| 3 |  |  | Xin khoá X => không được vì T1 đang đọc | Đọc số lượt xem hiện tại của nhà 2 => A = 0 |
| 4 | Thêm lượt xem với thông tin NV 4, Khách 1, nhà 3 | Xin khoá X => không được vì T2 đang đọc | **DEADLOCK** |  |
| 5 |  | **HUỶ T2** |  | Thêm lượt xem với thông tin NV 4, Khách 1, nhà 3 |
| 6 | Thêm lượt xem với thông tin NV 4, Khách 1, nhà 3 | Xin khoá X |  |  |
| 7 | Cập nhật số lượt xem + 1 cho nhà 3 trong bảng NHÀ | X |  | Cập nhật số lượt xem + 1 cho nhà 3 trong bảng NHÀ |
| 8 | **Commit** |  |  | **Commit** |

Như vậy, bằng cách trên, một giao tác sẽ bị huỷ (ở đây là T2) để đảm bảo tính thống nhất của dữ liệu.

***Code demo T1\_Fix:***

Use QuanLyNha

go

IF OBJECT\_ID('L31\_Fix') IS NOT NULL

DROP PROCEDURE L31\_Fix

GO

CREATE PROCEDURE L31\_Fix(@Ma\_Nha as int, @Ma\_KH as int, @Ma\_NV as int, @ngay AS DATE)

As

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

Begin Transaction

IF NOT EXISTS(SELECT \* FROM Nha WHERE ID = @Ma\_Nha)

BEGIN

PRINT 'Nha khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM NhanVien WHERE ID = @Ma\_NV)

BEGIN

PRINT 'Nhan vien khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM KhachHang WHERE ID = @Ma\_KH)

BEGIN

PRINT 'Khach hang khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

Begin Try

IF NOT EXISTS (SELECT \* FROM ChiTiet\_XemNha WHERE Ma\_KH = @Ma\_KH AND Ma\_Nha = @Ma\_Nha AND Ngay = @ngay)

BEGIN

WAITFOR DELAY '00:00:05'

INSERT INTO ChiTiet\_XemNha (Ma\_KH, Ma\_Nha, MaNV, Ngay) VALUES(@Ma\_KH, @Ma\_Nha, @Ma\_NV, @ngay)

UPDATE Nha Set Luot\_Xem = Luot\_Xem + 1 WHERE ID = @Ma\_Nha

END

End Try

Begin Catch

Print 'Loi THEM LUOT XEM trong L31'

SELECT ERROR\_MESSAGE()

Rollback Transaction

RETURN

End Catch

Commit Transaction

Go

DELETE FROM ChiTiet\_XemNha WHERE Ma\_Nha = 3

UPDATE Nha SET Luot\_Xem = 0 WHERE ID = 3

EXEC L31\_Fix 3, 1, 4, '06/08/2021'

SELECT ID,Luot\_Xem FROM Nha WHERE ID = 3

SELECT \* FROM ChiTiet\_XemNha WHERE Ma\_Nha = 3

***Code demo T2\_Fix:***

Use QuanLyNha

go

IF OBJECT\_ID('L32') IS NOT NULL

DROP PROCEDURE L32

GO

CREATE PROCEDURE L32(@Ma\_Nha as int, @Ma\_KH as int, @Ma\_NV as int, @ngay AS DATE)

As

Begin Transaction

IF NOT EXISTS(SELECT \* FROM Nha WHERE ID = @Ma\_Nha)

BEGIN

PRINT 'Nha khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM NhanVien WHERE ID = @Ma\_NV)

BEGIN

PRINT 'Nhan vien khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS(SELECT \* FROM KhachHang WHERE ID = @Ma\_KH)

BEGIN

PRINT 'Khach hang khong ton tai'

ROLLBACK TRANSACTION

RETURN

END

Begin Try

IF NOT EXISTS (SELECT \* FROM ChiTiet\_XemNha WHERE Ma\_KH = @Ma\_KH AND Ma\_Nha = @Ma\_Nha AND Ngay = @ngay)

BEGIN

INSERT INTO ChiTiet\_XemNha (Ma\_KH, Ma\_Nha, MaNV, Ngay) VALUES(@Ma\_KH, @Ma\_Nha, @Ma\_NV, @ngay)

UPDATE Nha Set Luot\_Xem = Luot\_Xem + 1 WHERE ID = @Ma\_Nha

END

End Try

Begin Catch

Print 'Loi THEM LUOT XEM trong L31'

Rollback Transaction

RETURN

End Catch

Commit Transaction

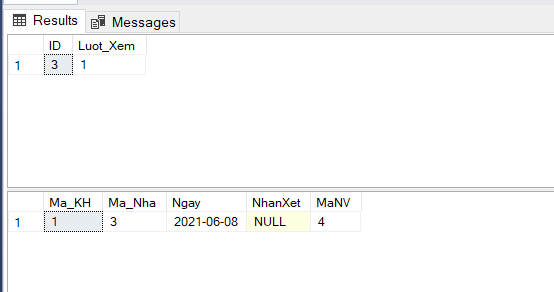
Go

EXEC L32 3, 1, 5, '06/08/2021'

SELECT ID,Luot\_Xem FROM Nha WHERE ID = 3

SELECT \* FROM ChiTiet\_XemNha WHERE Ma\_Nha = 3

***Kết quả trên SQL:***

******

# **XỬ LÝ CÁC TÌNH HUỐNG TRANH CHẤP ĐỒNG THỜI – LÝ THUYẾT**

1. **DIRTY READ => SỬ DỤNG MỨC CÔ LẬP READ COMITTED (MẶC ĐỊNH)**
2. **Trường hợp 1 – Sửa lỗi Cập nhật giá thuê nhà mới nhưng người xem vẫn thấy giá cũ**

Như vậy, khách hàng sẽ có thể yên tâm thông tin về giá nhà mình đọc được trên trang là giá nhà chính xác, không bị sửa đổi hay hủy bỏ trong lúc đọc => việc liên hệ nhân viên sẽ trở nên dễ dàng hơn.

1. **Trường hợp 2 – Sửa lỗi Hiển thị nhà đã bị xóa trên app nhưng sau đó nhà vẫn hiện thông tin bình thường**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Nhân viên xóa nhà khỏi hệ thống** | **Khóa** | **Khóa** | **T2 – Khách hàng truy cập xem nhà** |
| 1 | Kiểm tra mã số nhà có tồn tại không | S |  |  |
| 2 | Thao tác xóa nhà. | X |  |  |
| 3 |  | S => không được vì còn khóa X | Khách hàng truy cập vào xem nhà => nhà đã bị xóa |
| 4 | Thao tác xóa nhà gặp lỗi => không xóa được | Đợi… |  |
| 5 | **Commit** | Đợi… |  |
| 6 |  |  | S | Chủ nhà truy cập vào xem nhà => nhà vẫn chưa bị xóa. |
| 7 |  |  |  | **Commit** |

Như vậy, chủ nhà sẽ xem được chính xác liệu nhà đã xóa thành công trên hệ thống hay chưa. Nếu chưa, chủ nhà có thể liên hệ nhân viên để yêu cầu xóa lại.

1. **Trường hợp 3 – Sửa lỗi Hợp đồng bị hủy do vi phạm điều kiện, nhưng khách hàng vẫn đọc thấy trạng thái “đã bán” trên ứng dụng**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Nhân viên lập hợp đồng cho khách hàng** | **Khóa** | **Khóa** | **T2 – Khách hàng đọc thông tin của nhà** |
| 1 | Nhân viên cập nhật thông tin hợp đồng | **X** |  |  |
| 2 | Nhân viên chuyển trạng thái của nhà thành “Đã bán” |  |  |
| 3 |  | S => không được vì còn khóa X | Khách hàng đọc tin, thấy trạng thái nhà “Đã bán” |
| 4 | Hợp đồng bị hủy do hợp đồng cho thuê trước chưa hết hạn => nhà chuyển về trạng thái “Cho Thuê” | Đợi… |  |
| 5 | **Commit** | Đợi… |  |
| 6 |  |  | S | Khách hàng thấy tin “**Cho thuê**” |
| 7 |  |  |  | **Commit** |

Như vậy, khách hàng mua nhà sẽ thấy thông tin hợp đồng chưa chuyển sang trạng thái **“Đã bán”**, sẽ liên hệ để hỏi nhân viên về hợp đồng Mua – Bán, không gặp phải tình trạng hợp đồng chưa tạo thành công nhưng nhà vẫn có thông tin “Đã bán” trên trang.

1. **PHANTOM => Dùng mức cô lập Serializable**
2. **Trường hợp 1 – Sửa lỗi Thông báo số lượng nhà nhưng danh sách hiển thị không đủ số lượng**
3. **Trường hợp 2 – Sửa lỗi Cập nhật thông tin nhân viên phụ trách cho các ngôi nhà mới theo địa chỉ để thuận tiện cho việc hướng dẫn khách xem nhà, trong khi đó, một nhà mới được khách hàng thêm vào => nhà mới cũng được update nhân viên phụ trách nhưng địa chỉ không phù hợp**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Thêm nhân viên phụ trách cho nhà mới** | **Khóa** | **Khóa** | **T2 – Thêm nhà** |
| 1 | Duyệt danh sách các nhà chưa có nhân viên phụ trách, kiểm tra địa chỉ => quận 3, quận 5, quận 7 | S |  |  |
| 2 |  | X => không được vì chặn Insert | Khách hàng thêm một nhà mới ở quận 4 |
| 3 | Đếm số lượng nhà chưa có nhân viên phụ trách => 5 |  |  |
| 4 | Lấy danh sách 5 nhân viên phụ trách có địa chỉ quận 3, quận 5, quận 7 |  |  |
| 5 | Update danh sách 5 nhà mới với 5 nhân viên tìm được. |  |  |
| 6 | **Commit** |  |  |
| 7 |  |  | X | Khách hàng thêm một nhà mới ở quận 4 |
| 8 |  |  |  | **Commit** |

Như vậy, nhà được thêm sau giai đoạn kiểm tra sẽ được cập nhật nhân viên phụ trách sau hoặc do quản trị viên khác thêm, không xảy ra trường hợp cập nhật nhân viên ở xa nhà phụ trách nữa.

1. **Trường hợp 3 – Sửa lỗi Nhân viên đọc danh sách các khách hàng theo tiêu chí xác định và liên hệ, sau đó một khách hàng khác được thêm vào với cùng tiêu chí khiến nhân viên bỏ sót khách hàng mới**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **T1 – Đọc danh sách khách hàng theo tiêu chí** | **Khóa** | **Khóa** | **T2 – Thêm khách hàng** |
| 1 | Nhân viên đọc danh sách các khách hàng theo tiêu chí số lượng phòng là 5, quận Bình Thạnh | **S** |  |  |
| 2 |  | **X => không được vì chặn INSERT** | Quản trị viên thêm khách hàng mới với tiêu chí số phòng = 5, quận Bình Thạnh |
| 3 | Nhân viên update lại danh sách các khách hàng yêu cầu số lượng phòng = 5 và quận Bình Thạnh là “**Đã liên hệ”** |  |  |
| 4 | Thêm ngày hẹn cho n khách trong bảng Chi Tiết Xem nhà |  |  |
| 5 | **Commit** |  |  |
| 6 |  |  | **X** | Quản trị viên thêm khách hàng mới với tiêu chí số phòng = 5, quận Bình Thạnh |
| 7 |  |  |  | **Commit** |

Như vậy, khách hàng mới sẽ được tạo ra sau khi thao tác liên hệ và thêm chi tiết xem nhà của nhân viên kết thúc => không gây ra hiện tượng ghi nhầm dữ liệu “**Đã liên hệ”** lên khách hàng mới dẫn đến bỏ sót khách hàng này.