

**Nhóm: 18VP – 07**

**Thành viên:**

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

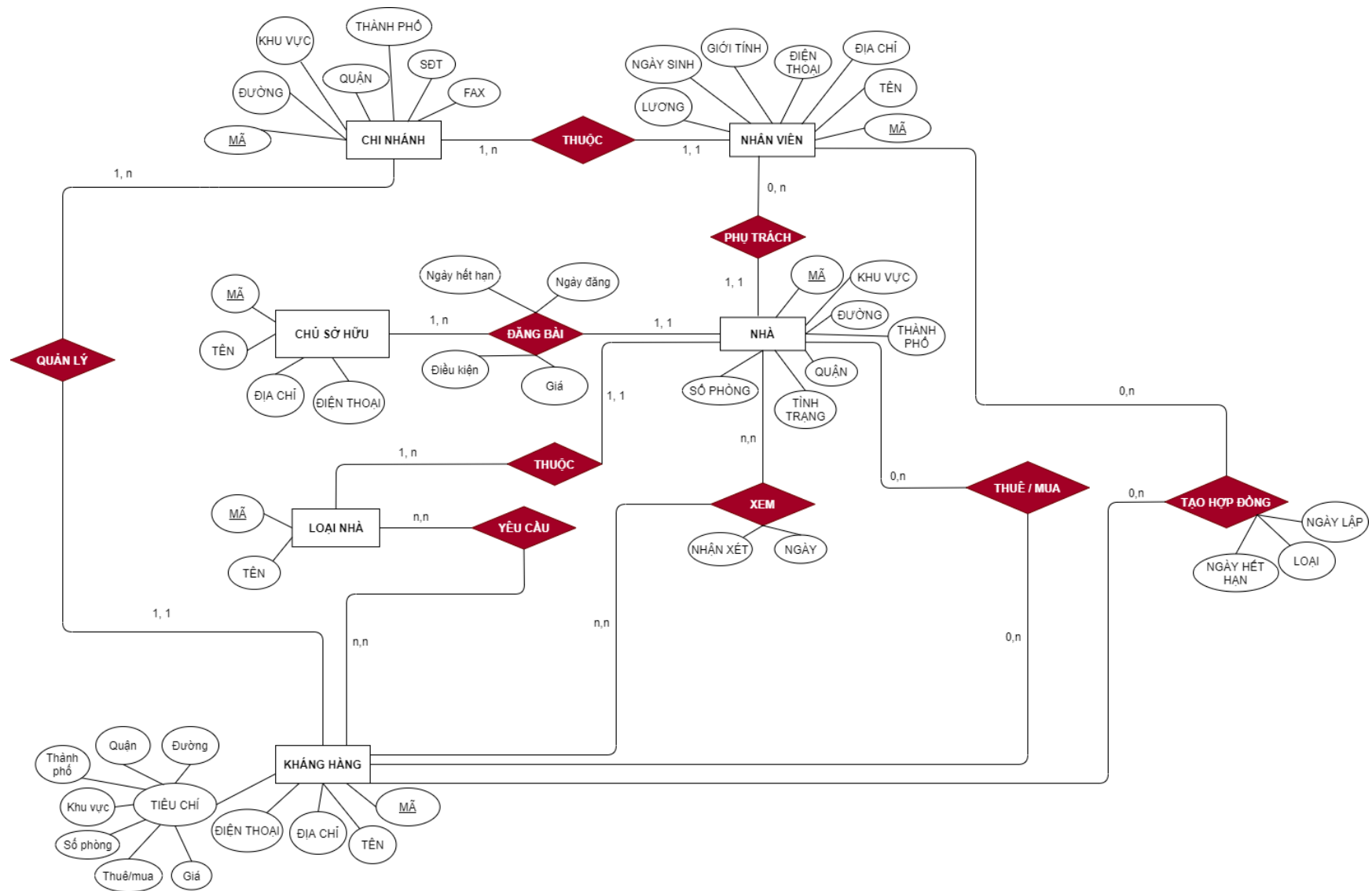
**BÁO CÁO ĐỒ ÁN MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

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

## NỘI DUNG BÁO CÁO

<b>BẢNG THIẾT KẾ ER .....</b>	<b>3</b>
<b>LƯỢC ĐỒ QUAN HỆ .....</b>	<b>4</b>
<b>CÁC RÀNG BUỘC TOÀN VỆ.....</b>	<b>6</b>
<b>CÀI ĐẶT CƠ SỞ DỮ LIỆU TRONG SQL SERVER.....</b>	<b>7</b>
<b>XÁC ĐỊNH LOẠI NGƯỜI DÙNG.....</b>	<b>15</b>
<b>CÁC CHỨC NĂNG HỆ THỐNG .....</b>	<b>16</b>
<b>CÁC CHỨC NĂNG TRANH CHẤP ĐỒNG THỜI – LÝ THUYẾT.....</b>	<b>19</b>
<b>XỬ LÝ CÁC TÌNH HUỐNG TRANH CHẤP ĐỒNG THỜI – LÝ THUYẾT.....</b>	<b>27</b>

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



## LƯỢC ĐỒ QUAN HỆ

### CHI NHÁNH

<u>Mã</u>	Đường	Khu vực	Quận	Thành phố	SĐT	Fax
-----------	-------	---------	------	-----------	-----	-----

### NHÂN VIÊN

<u>Mã</u>	Tên	Giới tính	Địa chỉ	Điện thoại	Ngày sinh	Lương	Mã Chi nhánh
-----------	-----	-----------	---------	------------	-----------	-------	--------------

### CHỦ SỞ HỮU

<u>Mã</u>	Tên	Địa chỉ	Điện thoại
-----------	-----	---------	------------

### NHÀ

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

### LOẠI NHÀ

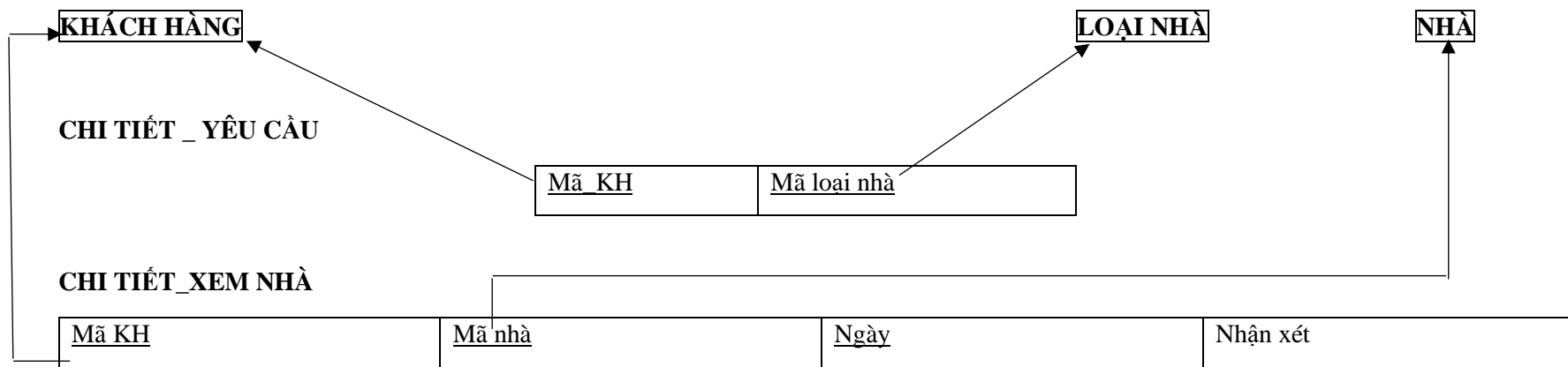
<u>Mã</u>	Tên
-----------	-----

### KHÁCH HÀNG

<u>Mã</u>	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	<u>Mã Chi nhánh</u>
-----------	-----	---------	------------	-------------------	------------------	------------------	------------------	----------------	---------------	--------------------	----------	------------	---------------------

### CHI TIẾT HỢP ĐỒNG

<u>Mã Nhà</u>	<u>Mã NV</u>	<u>Mã KH</u>	Loại hợp đồng	Ngày lập	Ngày hết hạn
---------------	--------------	--------------	---------------	----------	--------------



## CÁC RÀNG BUỘC TOÀN VỆN

- 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]
```

### 2) 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,
    [Ngày_Dang] [date] NULL,
    [Ngày_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]
```

### 3) 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]
```

### 4) 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]
```



## 5) 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]
```

## 6) 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]
```

## 7) 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]
```

## 8) 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]
```

## 9) 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]

```

## 10) 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 VỆN

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 truoac 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 truoac 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 truoac ngay dang bai'
    RollBack Tran
END
GO
```

## XÁC ĐỊNH LOẠI NGƯỜI DÙNG

LOẠI NGƯỜI DÙNG	MỘT SỐ HOẠT ĐỘNG TƯƠNG ỨNG
<b>CHỦ SỞ HỮU NHÀ</b>	<ul style="list-style-type: none"> <li>- Quản lý thông tin nhà đã đăng: thêm, xóa, sửa thông tin.</li> <li>- Kiểm tra thông tin người muốn thuê (trong trường hợp là nhà bán)</li> <li>- Kiểm tra thông tin chi nhánh quản lý nhà.</li> <li>- Kiểm tra quá trình thuê / bán nhà và thông tin người thuê/mua nhà.</li> </ul>
<b>NHÂN VIÊN</b>	<ul style="list-style-type: none"> <li>- Kiểm tra sự trùng khớp giữa tiêu chí của người muốn thuê/mua nhà và người muốn đăng, thông báo cho khách hàng.</li> <li>- Lập hợp đồng khi khách muốn thuê / mua nhà.</li> <li>- Ghi chú các nhận xét vào bảng Chitiết_xemnhà khi khách đến xem nhà và nhận xét.</li> </ul>
<b>NGƯỜI TÌM THUÊ / MUA NHÀ</b>	<ul style="list-style-type: none"> <li>- Kiểm tra thông tin chủ nhà.</li> <li>- Kiểm tra thông tin nhà.</li> <li>- Kiểm tra thông tin chi nhánh quản lý.</li> <li>- Kiểm tra thông tin nhân viên phụ trách nhà.</li> <li>- Kiểm tra thông tin hợp đồng.</li> </ul>
<b>QUẢN TRỊ CHI NHÁNH</b>	<ul style="list-style-type: none"> <li>- Quản lý thông tin khách hàng.</li> </ul>

## CÁC CHỨC NĂNG HỆ THỐNG

TÊN CHỨC NĂNG	MÔ TẢ	INPUT	OUTPUT	ĐỐI TƯỢNG SỬ DỤNG
<b>Hiển thị thông tin nhà</b>	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
<b>Lọc nhà theo các tiêu chí</b>	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
<b>Sắp xếp thông tin nhà</b>	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
<b>Thêm một thông tin nhà</b>	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



<b>Duyệt thông tin nhà</b>	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.	<ul style="list-style-type: none"> <li>- Danh sách thông tin các ngôi nhà chưa có nhân viên phụ trách.</li> <li>- Danh sách nhân viên.</li> </ul>	Danh sách thông tin các ngôi nhà đã có nhân viên phụ trách.	Quản trị của chi nhánh
<b>Cập nhật thông tin ngôi nhà</b>	Thông tin của ngôi nhà có thể thay đổi và được cập nhật theo yêu cầu từ chủ nhà, do nhân viên thực hiện.	<ul style="list-style-type: none"> <li>- Mã ngôi nhà.</li> <li>- Các thông tin cần cập nhật.</li> </ul>	Thông tin ngôi nhà mới với thông tin cập nhật.	Nhân viên
<b>Xóa một ngôi nhà</b>	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
<b>Thêm nhận xét cho ngôi nhà</b>	Khi một người đi xem nhà sẽ có các nhận xét, nhân viên cần ghi chú lại.	<ul style="list-style-type: none"> <li>- Mã ngôi nhà được nhận xét.</li> <li>- Mã khách hàng.</li> <li>- Ngày xem nhà.</li> <li>- Nhận xét của khách hàng.</li> </ul>	Dòng dữ liệu mới trong bảng Chi tiết xem nhà.	Nhân viên

<b>Thêm một khách hàng</b>	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
<b>Cập nhật một khách hàng</b>	Khách hàng có thể yêu cầu thay đổi thông tin cá nhân cũng như tiêu chí tìm nhà.	<ul style="list-style-type: none"> <li>- Mã khách hàng.</li> <li>- Thông tin cần thay đổi.</li> </ul>	Row khách hàng tương ứng với thông tin cập nhật.	Quản trị chi nhánh
<b>Xóa một khách hàng</b>	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
<b>Thêm hợp đồng</b>	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 – LÝ THUYẾT

### I. DIRTY READ

#### 1) 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 nhà có tồn tại không.	
2		Khách hàng truy cập vào bảng thông tin nhà.
3	Sửa đổi giá của nhà từ 6.000.000 thành 3.500.000 đồng	
4		Khách hàng tìm nhà giá 3.000.000 đến 3.500.000 và đọc được nhà là giá thuê nhà 3.500.000 đồng => liên hệ nhân viên.
5	Nhân viên hủy section vì tìm nhầm mã nhà, dữ liệu đổi thành 6.000.000 đồng	
6		Commit
7	Roll back	

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.

#### 2) Trường hợp 2 – Hiện 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 có tồn tại không	
2	Thao tác xóa nhà.	
3		Khách hàng truy cập vào xem nhà => nhà đã bị xóa

4	Thao tác xóa nhà gặp lỗi => không xóa được	
5		<b>Commit</b>
6	<b>Roll back</b>	

Chủ nhà sau khi yêu cầu xóa nhà, thấy thông tin nhà đã không còn trên website sẽ yên tâm. Nhân viên phụ trách sau khi thực hiện xóa nhà cũng không kiểm tra lại. Điều này có thể khiến khách hàng sau khi nhìn thấy thông tin nhà đã xóa (*không thành công*) và liên hệ với nhân viên sau đó => gây mất thời gian hai bên.

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

	<b>T1 – Nhân viên lập hợp đồng cho khách hàng</b>	<b>T2 – Khách hàng đọc thông tin của nhà</b>
1	Nhân viên tạo mã hợp đồng mới	
2	Nhân viên cập nhật thông tin hợp đồng	
3	Nhân viên chuyển trạng thái của nhà thành “Đã bán”	
4		<b>Khách hàng đọc tin, thấy trạng thái nhà “Đã bán”</b>
5	Hợp đồng bị hủy do hợp đồng cho thuê trước chưa hết hạn => nhà chuyển thành trạng thái “Cho thuê”	
6	<b>Roll back</b>	
7		<b>Commit</b>

Trong trường hợp này, trạng thái của nhà đã gây hiểu lầm, khiến khách hàng hiện tại không mua được nhà nhưng nghĩ mình đã mua thành công.

## II. UNREPEATABLE READ

- 1) 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	
2		Đọc danh sách tất cả các nhà
3		Update giá thuê mỗi nhà tăng thêm 100.000
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	
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.

- 2) Trường hợp 2 – Hiển thị danh sách nhà có giá từ 1.500.000 đến 2.000.000: 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 1.500.000 đến 2.000.000 => 5 nhà	
2		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 có giá từ 1.500.000 đến 2.000.000 và hiển thị => danh sách 6 nhà	
4	Commit	
5		Commit

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

- 3) Trường hợp 3 – Quản trị viên đổi tên đường đồng loạt từ Đình Tiên Hoàng thành Lê Văn Duyệt theo quy định, trong khi đó, một nhà ở Đình Tiên Hoàng được nhân viên thay đổi tên đường thành Phan Đăng Lưu => kết quả địa chỉ chứa quận 5, TP.THỦ ĐỨC => Dữ liệu sai

	T1 – Chỉnh sửa đồng loạt tên đường Đình Tiên Hoàng thành đường Lê Văn Duyệt	T2 – Thay đổi tên đường của một nhà từ Đình Tiên Hoàng thành Phan Đăng Lưu
1	Nhân viên A chọn danh sách các Mã Nhà có tên đường là Đình Tiên Hoàng, thuộc quận Bình Thạnh	
2		Nhân viên B chọn và kiểm tra thông tin nhà 002 với đường Đình Tiên Hoàng, quận Bình Thạnh
3		Nhân viên B thay đổi tên đường Đình Tiên Hoàng thành đường Phan Đăng Lưu của mã nhà 002
4	Thay đổi đường của tất cả các nhà có mã trong danh sách nhà có tên đường Đình Tiên Hoàng thành đường Lê Văn Duyệt và hiển thị danh sách => không có nhà 002.	
5	Commit	
6		Hủy giao tác do thay đổi nhầm mã nhà
		Roll back

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 Đình Tiên Hoàng như cũ.

### III. PHANTOM

- 1) Trường hợp 1 – Thông báo số lượng nhà nhưng danh sách hiển thị dư số lượng

	T1 – Lọc danh sách nhà theo tiêu chí số lượng phòng	T2 – Thêm một nhà có số lượng phòng 5
--	---	---------------------------------------

1	Truy xuất thông tin đếm số lượng nhà có số phòng = 5 bằng hàm COUNT => kết quả là 5 nhà	
2		Thêm một nhà có số lượng phòng là 5
3	Truy xuất cơ sở dữ liệu hiển thị thông tin của tất cả nhà có số phòng bằng 5 => Hiển thị 6 nhà.	
4	<b>Commit</b>	
5		<b>Commit</b>

Trong trường hợp trên, một dòng dữ liệu mới khi hiển thị thông tin tất cả các nhà có số phòng là 5 chính là 1 PHANTOM bởi cùng thao tác truy xuất đến bộ dữ liệu nhà, ta có 2 danh sách khác nhau: 1 danh sách 5 kết quả và 1 danh sách 6 kết quả.

- 2) 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

	<b>T1 – Thêm nhân viên phụ trách cho nhà mới</b>	<b>T2 – Thêm nhà</b>
1	Duyệt danh sách các nhà chưa có nhân viên phụ trách, kiểm tra địa chỉ => quận 3, quận 5, quận 7	
2		Khách hàng thêm một nhà mới ở quận 4
3	Đếm số lượng nhà chưa có nhân viên phụ trách => 6	
4	Lấy danh sách 6 nhân viên phụ trách có địa chỉ quận 3, quận 5, quận 7	
5		<b>Commit</b>
6	Update danh sách 6 nhà mới với 6 nhân viên tìm được.	
7	<b>Commit</b>	

Trường hợp này không gây ra lỗi chương trình, tuy nhiên, do sự xuất hiện mới của một nhà, danh sách nhân viên được phân bổ theo địa chỉ sẽ không còn phù hợp theo đúng ý quản trị viên (ví dụ nhà ở quận 4 được phụ trách bởi các nhân viên ở quận 3, quận 5, quận 7), khiến việc xem nhà của khách hàng và nhân viên có thể gặp bất cập do khoảng cách địa lý.

**3) 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**

	<b>T1 – Đọc danh sách khách hàng theo tiêu chí</b>	<b>T2 – Thêm khách hàng</b>
1	Nhân viên đọc danh sách các khách hàng theo tiêu chí số lượng phòng là 5, quận Bình Thạnh	
2		Quản trị viên thêm khách hàng mới với tiêu chí số phòng = 5, quận Bình Thạnh
3	<b>Nhân viên update lại danh sách các khách hàng yêu cầu số lượng phòng = 5 và quận Bình Thạnh là 1</b>	
4	Thêm ngày hẹn cho n khách trong bảng Chi Tiết Xem nhà	
5		<b>Commit</b>
6	<b>Commit</b>	

Trong trường hợp trên, do lần đọc 1, nhân viên chỉ đọc được n khách hàng nên đã liên hệ với n khách. Nhưng ở lần đọc 2 để update danh sách lại có đến  $n + 1$  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.

**4) LOST UPDATE**

**1) Trường hợp 1 – Lỗi cập nhật nhân viên phụ trách không đồng nhất**

	<b>T1 – Cập nhật nhân viên phụ trách A =&gt; B</b>	<b>T2 – Cập nhật nhân viên phụ trách A =&gt; C</b>
1	Kiểm tra mã nhân viên phụ trách nhà 001 => mã A	



2		Kiểm tra mã nhân viên phụ trách nhà 001 => <b>mã A</b>
3	Thay đổi mã A thành mã B	
4		Thay đổi mã A thành mã C
5		Kiểm tra bảng chi tiết xem nhà, nếu nhà 001 có lượt xem vào ngày sau ngày hiện tại => update mã nhân viên thành C.
6	Kiểm tra bảng chi tiết xem nhà, nếu nhà 001 có lượt xem vào ngày sau ngày hiện tại => update mã nhân viên thành B.	
7	<b>Commit</b>	
8		<b>Commit</b>

Trong trường hợp trên, các transactions đã tạo nên sự bất đồng nhất về dữ liệu trên database, khi mã nhân viên phụ trách sau cùng của nhà 001 là C nhưng mã nhân viên phụ trách xem nhà của cùng nhà 001 là B.

- 2) 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 1	
2	Tăng 20% giá nhà $A = 1.2A$	
3		Tăng giá trị nhà: $A = A + 1.000.000$
4	Nếu $A > 3000$ : $A = A * 1.05$	
5		<b>Commit</b>
6	<b>Commit</b>	

Như vậy, giả sử giá nhà là 2.700.000, sau khi hoàn tất 2 giao tác này sẽ là 4.452.000 đồng. Tuy nhiên, nếu như thực hiện đúng và tuần tự, ví dụ T1 trước T2, thì giá nhà phải là 4.402.000 đồng hoặc nếu T2 trước T1, giá nhà phải là 4.662.000. Điều này sẽ khiến tính đồng nhất về dữ liệu giá nhà bị ảnh hưởng.

**3) Trường hợp 3 – cập nhật thành phố cùng lúc của một nhà trong hai tiến trình gây ra sai lầm trong lưu trữ dữ liệu**

	<b>T1 – Thay đổi đồng loạt thành phố của các nhà ở quận 9 và quận Thủ Đức thành thành phố Thủ Đức</b>	<b>T2 – Thay đổi địa chỉ một nhà ở quận 9 thành quận 5, TP.HCM</b>
1		Kiểm tra sự tồn tại của mã nhà 005
2		Kiểm tra thành phố của mã nhà 005
3	Duyệt các nhà trong hệ thống	
4		Nếu là TP khác, đổi thành TP.HCM.
5	Nếu là quận 9 hoặc quận Thủ Đức, cập nhật thành phố HCM thành TP.Thủ Đức	
6		Kiểm tra quận của mã nhà 005, cập nhật quận 9 thành quận 5.
7		<b>Commit</b>
8	<b>Commit</b>	

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”**.

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

### I. DIRTY READ => SỬ DỤNG MỨC CÔ LẬP **READ COMMITTED** (MẶC ĐỊNH)

#### 1. Trường hợp 1 – **Sửa lỗi** Cập nhật giá thuê nhà mới nhưng người xem vẫn thấy giá cũ

	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 có tồn tại không.	S		
2			S	Khách hàng truy cập vào bảng thông tin nhà.
3	Sửa đổi giá của nhà từ 6.000.000 thành 3.500.000 đồng	X		
4			S => không được	Khách hàng tìm nhà giá 3.000.000 đến 3.500.000
5	Nhân viên hủy section vì tìm nhầm mã nhà, dữ liệu đổi thành 6.000.000 đồng		Đợi...	
6	<b>Commit</b>		Đợi...	
7			S	Khách hàng tìm nhà giá 3.000.000 đến 3.500.000 => dữ liệu đã được commit đúng đắn => các nhà đều có giá đúng từ 3.000.000 đến 3.500.000
8				<b>Commit</b>

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.

## 2. 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	<b>Commit</b>		Đợi...	
6			S	Chủ nhà truy cập vào xem nhà => nhà vẫn chưa bị xóa.
7				<b>Commit</b>

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.

## 3. 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	<b>Commit</b>		Đợi...	
6			S	Khách hàng thấy tin “Cho thuê”
7				<b>Commit</b>

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.

## II. UNREPEATABLE READ => SỬ DỤNG MỨC CÔ LẬP REPEATABLE READ

1. Trường hợp 1 – **Sửa lỗi** Hiện 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	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	Update giá thuê mỗi nhà tăng thêm 100.000
4	Lọc danh sách tất cả các nhà có giá thuê = giá thuê thấp nhất ở t1		Đợi...	
5	<b>Commit</b>		Đợi...	

6			X	Update giá thuê mỗi nhà tăng thêm 100.000
7				<b>Commit</b>

Như vậy, sau khi lọc danh sách các nhà thấp nhất thì giá nhà mới tăng thêm 100.000. Do đó, danh sách giá nhà thấp nhất vẫn được giữ nguyên mà không bị thay đổi so với giá thấp nhất ban đầu.

## 2. Trường hợp 2 – **Sửa lỗi** Hiển thị danh sách nhà có giá từ 1.500.000 đến 2.000.000: số lượng nhà và danh sách thật sự khác nhau

	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 1.500.000 đến 2.000.000 => 5 nhà	S		
			<b>S</b>	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 có giá từ 1.500.000 đến 2.000.000 và hiển thị => danh sách 5 nhà.		Đợi...	
4	<b>Commit</b>		Đợi...	
5			X	Thay đổi giá của nhà 003 từ 2.500.000 còn 2.000.000
6				<b>Commit</b>

Như vậy, T1 đã có thể hiển thị đúng số lượng và danh sách nhà có giá từ 1.500.000 đến 2.000.000 ngay tại thời điểm kiểm tra. Nhà 003 dù đã cập nhật nhưng do cập nhật sau giai đoạn kiểm tra nên không nằm trong danh sách. Đây là điều hợp lý.

**3. Trường hợp 3 – Sửa lỗi** Quản trị viên đổi tên đường đồng loạt từ Đình Tiên Hoàng thành Lê Văn Duyệt theo quy định, trong khi đó, một nhà ở Đình Tiên Hoàng được nhân viên thay đổi tên đường thành Phan Đăng Lưu => kết quả địa chỉ chứa quận 5, TP.THỦ ĐỨC => Dữ liệu sai

	T1 – Chỉnh sửa đồng loạt tên đường Đình Tiên Hoàng thành đường Lê Văn Duyệt	Khóa	Khóa	T2 – Thay đổi tên đường của một nhà từ Đình Tiên Hoàng thành Phan Đăng Lưu
1	Nhân viên A chọn danh sách các Mã Nhà có tên đường là Đình Tiên Hoàng, thuộc quận Bình Thạnh	X		
2			S => không được vì còn khóa X	Nhân viên B chọn và kiểm tra thông tin nhà 002 với đường Đình Tiên Hoàng, quận Bình Thạnh
3			Đợi...	Nhân viên B thay đổi tên đường Đình Tiên Hoàng thành đường Phan Đăng Lưu của mã nhà 002
4	Thay đổi đường của tất cả các nhà có mã trong danh sách nhà có tên đường Đình Tiên Hoàng thành đường Lê Văn Duyệt và hiển thị danh sách.		Đợi...	
5	Commit		Đợi...	
6			S	Nhân viên B chọn và kiểm tra thông tin nhà 002 với đường Lê Văn Duyệt, quận Bình Thạnh

7				thay đổi tên thành đường Phan Đăng Lưu của mã nhà 002
8				Hủy transaction do nhầm mã nhà.
9				<b>Commit</b>

Như vậy, tên đường của nhà 002 sẽ không bị mất đồng bộ, được thay đổi thành Lê Văn Duyệt theo đúng mong muốn của quản trị viên.

### III. PHANTOM => Dùng mức cô lập Serializable

#### 1. Trường hợp 1 – **Sửa lỗi** Thông báo số lượng nhà nhưng danh sách hiển thị không đủ số lượng

	<b>T1 – Lọc danh sách nhà theo tiêu chí số lượng phòng</b>	<b>Khóa</b>	<b>Khóa</b>	<b>T2 – Xóa một nhà</b>
1	Truy xuất thông tin đếm số lượng nhà có số phòng = 5 bằng hàm COUNT => kết quả là 5 nhà	<b>S</b>		
2			<b>X: không được vì SQL ngăn chặn thêm dữ liệu</b>	
3	Truy xuất cơ sở dữ liệu hiển thị thông tin của tất cả nhà có số phòng bằng 5 => Hiển thị 5 nhà.		Đợi...	
4	<b>Commit</b>		Đợi...	
5			<b>X</b>	Thêm một nhà có số lượng phòng là 5
6				<b>Commit</b>



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

**2. 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	<b>Commit</b>			
7			X	Khách hàng thêm một nhà mới ở quận 4
8				<b>Commit</b>

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.

**3. 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

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

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.

#### IV. LOST UPDATE => SỬ DỤNG MỨC CÔ LẬP REPEATABLE READ

##### 1. Trường hợp 1 – **Sửa lỗi** Lỗi cập nhật nhân viên phụ trách không đồng nhất

	T1 – Cập nhật nhân viên phụ trách A => B	Khóa	Khóa	T2 – Cập nhật nhân viên phụ trách A => C
1	Kiểm tra mã nhân viên phụ trách nhà 001 => <b>mã A</b>	U		
2			U => <b>không xin được vì có khóa U khác</b>	Kiểm tra mã nhân viên phụ trách nhà 001 => <b>mã A</b>
3	Thay đổi mã A thành mã B		Đợi...	
4			Đợi...	
5			Đợi...	
6	Kiểm tra bảng chi tiết xem nhà, nếu nhà 001 có lượt xem vào ngày sau ngày hiện tại => update mã nhân viên thành B.		Đợi...	
7	<b>Commit</b>		Đợi...	
8			U	Kiểm tra mã nhân viên phụ trách nhà 001 => <b>mã B # A</b>
				<b>Commit</b>

Như vậy, sau khi kiểm tra, quản trị ở T2 thấy nhân viên phụ trách không còn là A nên ngưng thay đổi => mã nhân viên phụ trách mới là B thống nhất ở cả hai bảng Nhà và ChiTiết – XemNhà.

##### 2. Trường hợp 2 – **Sửa lỗi** 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.

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

Như vậy, giả sử giá nhà là 2.700.000, sau khi hoàn tất 2 giao tác này sẽ là 4.402.000 đồng, tương tự như việc thực hiện tuần từ T1 => T2.

### 3. Trường hợp 3 – **Sửa lỗi** cập nhật thành phố cùng lúc của một nhà trong hai tiến trình gây ra sai lầm trong lưu trữ dữ liệu

	<b>T1 – Thay đổi đồng loạt thành phố của các nhà ở quận 9 và quận Thủ Đức thành thành phố Thủ Đức</b>	<b>Khóa</b>	<b>Khóa</b>	<b>T2 – Thay đổi địa chỉ một nhà ở quận 9 thành quận 5, TP.HCM</b>
1			<b>S</b>	Kiểm tra sự tồn tại của mã nhà 005
2				Kiểm tra thành phố của mã nhà 005
3	Duyệt các nhà trong hệ thống	<b>X =&gt; không xin được vì còn khóa S</b>		
4		Đợi...		Nếu là TP khác, đổi thành TP.HCM.
5		Đợi...		Kiểm tra quận của mã nhà 005, cập nhật quận 9 thành quận 5.
6		Đợi...		<b>Commit</b>
7	Duyệt các nhà trong hệ thống			
8	Nếu là quận 9 hoặc quận Thủ Đức, cập nhật thành phố HCM thành TP.Thủ Đức			
9	<b>Commit</b>			

Như vậy, do T1 đợi đến khi T2 hoàn tất, mã nhà 005 lúc này sẽ là quận 5, do đó khi duyệt ở T1 sẽ không truy lại mã nhà 005 và không thực hiện thay đổi thành thành phố Thủ Đức => đồng bộ dữ liệu.