

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ  
VIỆN TRÍ TUỆ NHÂN TẠO**

**-----\*\*\*-----**



**BÁO CÁO MÔN HỌC CƠ SỞ DỮ LIỆU  
ĐỀ TÀI:  
THIẾT KẾ CƠ SỞ DỮ LIỆU QUẢN LÝ  
KHÁCH HÀNG**

**Nhóm sinh viên thực hiện:**

- 1. Nguyễn Văn Linh – 23020395 (Nhóm trưởng)**
- 2. Trịnh Hoàng Đức – 23020359**
- 3. Phan Mạnh Quân – 23020419**
- 4. Đặng Đức Duy – 23020347**

**Giảng viên hướng dẫn: TS. Trần Hồng Việt**

**HÀ NỘI, 12/2024**

# MỞ ĐẦU

Trong thời đại công nghệ phát triển mạnh mẽ, việc quản lý thông tin hiệu quả là yếu tố then chốt giúp các doanh nghiệp và tổ chức tối ưu hóa hoạt động và nâng cao hiệu suất làm việc. Đối với ngành bất động sản, đặc biệt là lĩnh vực cho thuê nhà, quản lý thông tin khách hàng, hợp đồng và dịch vụ trở nên vô cùng quan trọng. Tuy nhiên, quá trình này thường gặp nhiều thách thức khi dữ liệu ngày càng phức tạp và khối lượng thông tin cần xử lý không ngừng tăng lên.

Qua đó, nhóm chúng em quyết định chọn đề tài “Quản lý hệ thống khách hàng” để làm báo cáo và xây dựng hệ thống Cơ sở dữ liệu quản lý khách hàng nhằm hỗ trợ việc lưu trữ, quản lý và xử lý thông tin khách hàng một cách khoa học và hiệu quả.

Báo cáo gồm 8 phần:

I. Phát biểu bài toán

II. Mô tả chi tiết hệ thống

III. Mô hình ER của hệ thống

IV. Mô hình quan hệ

V. Đặc tả yêu cầu dữ liệu

VI. Danh sách các ràng buộc dữ liệu

VII. Cài đặt vật lý.

# MỤC LỤC

<b>MỞ ĐẦU .....</b>	<b>1</b>
<b>I. Phát biểu bài toán.....</b>	<b>4</b>
<b>II. Mô tả chi tiết hệ thống .....</b>	<b>4</b>
<b>III. Mô hình ER của hệ thống .....</b>	<b>5</b>
<b>IV. Mô hình quan hệ (chuyển từ mô hình ER) .....</b>	<b>6</b>
<b>V. Đặc tả yêu cầu dữ liệu (Từ điển dữ liệu) .....</b>	<b>7</b>
<b>VI. Danh sách các ràng buộc dữ liệu .....</b>	<b>9</b>
<b>1. Ràng buộc UNIQUE CONSTRAINT .....</b>	<b>9</b>
<b>2. Ràng buộc check .....</b>	<b>9</b>
<b>3. Ràng buộc khóa ngoại .....</b>	<b>9</b>
<b>VII. Các bước cài đặt vật lý .....</b>	<b>11</b>
<b>1. Tạo database từ mô hình quan hệ tương ứng .....</b>	<b>11</b>
<b>2. Tạo constraint cho dữ liệu.....</b>	<b>13</b>
<b>3. Chèn dữ liệu vào Database.....</b>	<b>13</b>
<b>4. Tạo Trigger .....</b>	<b>15</b>
<b>5. Tạo Procedures .....</b>	<b>17</b>
<b>6. Tạo Transaction.....</b>	<b>19</b>
<b>7. Tạo Queries .....</b>	<b>20</b>
<b>VIII. Kết luận .....</b>	<b>22</b>

## I. Phát biểu bài toán

Hệ thống quản lý khách hàng thuê nhà được thiết kế để lưu trữ, quản lý và theo dõi thông tin liên quan đến khách hàng, nhà cho thuê, hợp đồng thuê, hóa đơn và các dịch vụ đi kèm. Mục tiêu của hệ thống là tạo ra một cơ sở dữ liệu tập trung, giúp cho việc quản lý quá trình cho thuê nhà trở nên hiệu quả và minh bạch hơn.

## II. Mô tả chi tiết hệ thống

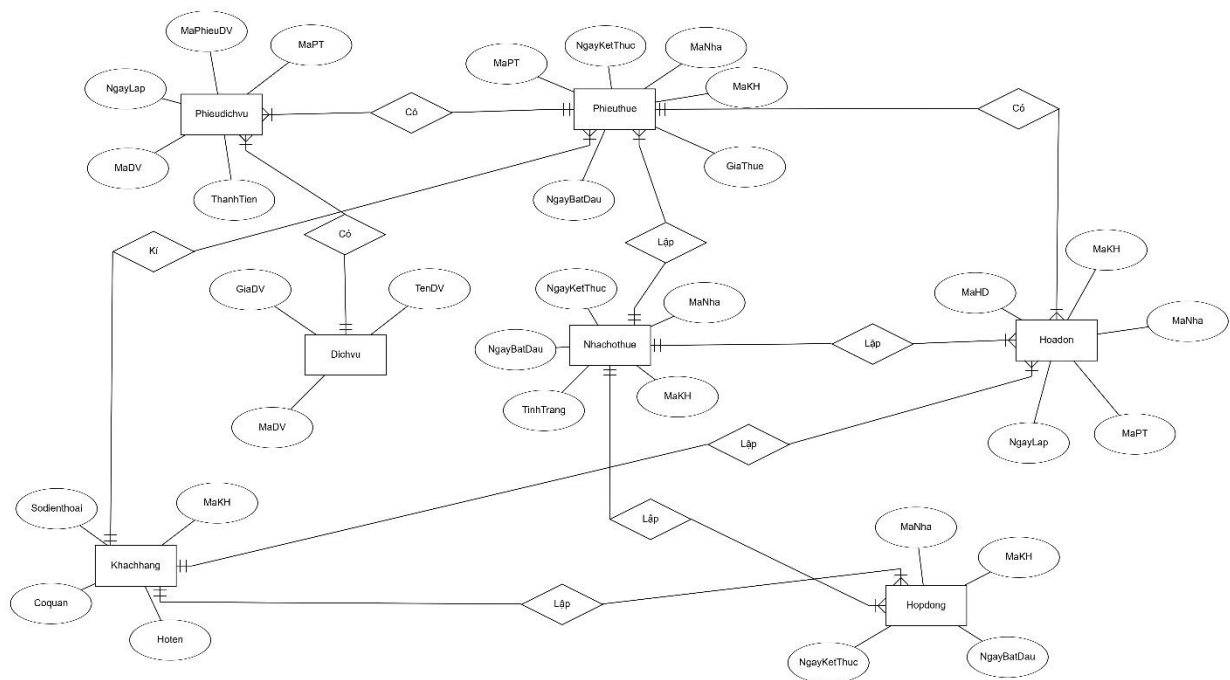
Hệ thống cơ sở dữ liệu quản lý khách hàng được xây dựng nhằm giúp giúp quản lý thông tin khách hàng, thông tin ngôi nhà, xây dựng hợp đồng và giới thiệu dịch vụ; giúp tối ưu hóa thời gian xử lý công việc.

- **Hệ quản trị cơ sở dữ liệu:**

- **Quản lý thông tin khách hàng:** Hệ thống lưu trữ thông tin chi tiết về khách hàng thuê nhà, bao gồm: họ tên, số điện thoại, cơ quan công tác. Thông tin khách hàng có thể được cập nhật, thêm mới hoặc xóa khi cần thiết.
- **Quản lý thông tin nhà cho thuê:** Hệ thống quản lý thông tin chi tiết về các căn nhà cho thuê, bao gồm: địa chỉ, giá thuê, tên chủ nhà, tình trạng nhà (có sẵn, đã thuê, đang bảo trì). Thông tin nhà có thể được cập nhật, thêm mới hoặc xóa khi cần thiết.
- **Quản lý hợp đồng thuê:** Hệ thống quản lý thông tin về các hợp đồng thuê, bao gồm: khách hàng thuê, nhà cho thuê, ngày bắt đầu, ngày kết thúc, các điều khoản hợp đồng. Thông tin hợp đồng có thể được tìm kiếm, thêm mới, chỉnh sửa hoặc xóa khi cần thiết.
- **Quản lý phiếu thuê:** Hệ thống quản lý thông tin về các phiếu thuê, bao gồm: khách hàng thuê, nhà cho thuê, ngày bắt đầu thuê theo phiếu thuê, ngày kết thúc thuê theo phiếu thuê, giá thuê theo phiếu thuê. Thông tin phiếu thuê có thể được tìm kiếm, thêm mới, chỉnh sửa hoặc xóa khi cần thiết.
- **Quản lý hóa đơn:** Hệ thống quản lý thông tin về các hóa đơn, bao gồm: khách hàng thuê, nhà cho thuê, phiếu thuê liên quan, ngày lập hóa đơn, tổng số tiền phải thanh toán. Thông tin hóa đơn có thể được tìm kiếm, thêm mới, chỉnh sửa hoặc xóa khi cần thiết.

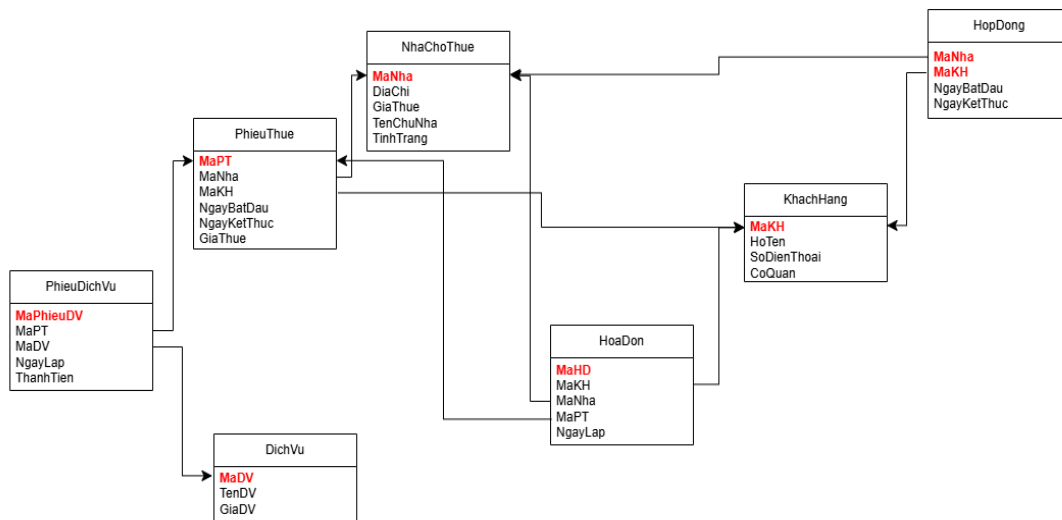
- **Quản lý dịch vụ:** Hệ thống quản lý thông tin về các dịch vụ đi kèm, bao gồm: tên dịch vụ, giá dịch vụ. Thông tin dịch vụ có thể được tìm kiếm, thêm mới, chỉnh sửa hoặc xóa khi cần thiết.
- **Quản lý phiếu dịch vụ:** Hệ thống quản lý thông tin về các phiếu dịch vụ, bao gồm: phiếu thuế liên quan, dịch vụ, ngày lập phiếu dịch vụ, thành tiền. Thông tin phiếu dịch vụ có thể được tìm kiếm, thêm mới, chỉnh sửa hoặc xóa khi cần thiết.

### III. Mô hình ER của hệ thống



Ảnh 1: Mô hình ER cho hệ thống quản lý khách hàng

## IV. Mô hình quan hệ (chuyển từ mô hình ER)



Ảnh 2: Mô hình quan hệ

Các thành phần của cơ sở dữ liệu:

- **KhachHang** (MaKH, HoTen, SoDienThoai, CoQuan)
- **NhaChoThue** (MaNha, DiaChi, GiaThue, TenChuNha, TinhTrang)
- **HopDong** (MaNha, MaKH, NgayBatDau, NgayKetThuc)
- **PhieuThue** (MaPT, MaNha, MaKH, NgayBatDau, NgayKetThuc, GiaThue)
- **HoaDon** (MaHD, MaKH, MaNha, MaPT, NgayLap)
- **DichVu** (MaDV, TenDV, GiaDV)
- **PhieuDichVu** (MaPhieuDV, MaPT, MaDV, NgayLap, ThanhTien)

## V. Đặc tả yêu cầu dữ liệu (Từ điển dữ liệu)

TỪ ĐIỂN DỮ LIỆU					
Tên thực thể	Tên thuộc tính	Kiểu dữ liệu	Độ dài	Mô tả	Giá trị
KhachHang	<u>MaKH</u>	INT	Variable	Mã định danh duy nhất của khách hàng	001, 002, ...
	Hoten	VARCHAR	100	Tên khách hàng	Nguyễn Văn Linh, ...
	Sodienthoai	VARCHAR	15	Số điện thoại của khách hàng	0981915756, ...
	Coquan	VARCHAR	100	Cơ quan của khách hàng	Trường DH FPT, ...
NhaChoThue	<u>MaNha</u>	INT	Variable	Mã định danh duy nhất của ngôi nhà	001, 002, ...
	DiaChi	VARCHAR	255	Địa chỉ nhà	123 Lê Lợi, Q1; ...
	GiaThue	DECIMAL	Variable	Giá thuê của ngôi nhà	10000000.00, ...
	TenChuNha	VARCHAR	100	Tên của chủ nhà	Đỗ Việt Trung, ...
	TinhTrang			Tình trạng của ngôi nhà khách muốn thuê	'Có sẵn', 'Đã thuê', ...
HopDong	<u>MaNha</u>	INT	Variable	Mã định danh duy nhất của ngôi nhà	001, 002, ...
	<u>MaKH</u>	INT	Variable	Mã định danh duy nhất của khách hàng	001, 002, ...
	NgayBatDau	DATE	Variable	Ngày bắt đầu của hợp đồng thuê	2023-01-01, ...
	NgayKetThuc	DATE	Variable	Ngày kết thúc hợp đồng thuê	2023-12-31, ...
PhieuThue	<u>MaPT</u>	INT	Variable	Mã định danh duy nhất của phiếu thuê	001, 002, ...
	MaNha	INT	Variable	Mã định danh duy nhất của ngôi nhà	001, 002, ...
	MaKH	INT	Variable	Mã định danh duy nhất của khách hàng	001, 002, ...

	NgayBatDau	DATE	Variable	Ngày bắt đầu của hợp đồng thuê	2023-01-01
	NgayKetThuc	DATE	Variable	Ngày kết thúc hợp đồng thuê	2023-12-31
	GiaThue	DECIMAL	Variable	Giá thuê 1 tháng của ngôi nhà khách muốn thuê	10000000.00
HoaDon	<u>MaHD</u>	INT	Variable	Mã định danh duy nhất của hoá đơn	001, 002, ...
	MaKH	INT	Variable	Mã định danh duy nhất của khách hàng	001, 002, ...
	MaNha	INT	Variable	Mã định danh duy nhất của ngôi nhà	001, 002, ...
	MaPT	INT	Variable	Mã định danh duy nhất của phiếu thuê	001, 002, ...
	NgayLap	DATE	Variable	Ngày lập hóa đơn	2023-01-10
DichVu	<u>MaDV</u>	INT	Variable	Mã định danh duy nhất của dịch vụ	001, 002, ...
	TenDV	VARCHAR	100	Tên của dịch vụ	'Dọn dẹp', 'Bảo trì', ...
	GiaDV	DECIMAL	Variable	Giá của dịch vụ khách hàng muốn sử dụng	500000.00
PhieuDichVu	<u>MaPhieuDV</u>	INT	Variable	Mã định danh duy nhất của phiếu dịch vụ	001, 002, ...
	MaPT	INT	Variable	Mã định danh duy nhất của phiếu thuê	001, 002, ...
	MaDV	INT	Variable	Mã định danh duy nhất của dịch vụ	001, 002, ...
	NgayLap	DATE	Variable	Ngày lập hóa đơn	2023-01-10
	ThanhTien	DECIMAL	Variable	Tổng tiền của một phiếu dịch vụ	500000.00



## VI. Danh sách các ràng buộc dữ liệu

### 1. Ràng buộc UNIQUE CONSTRAINT:

- **UC\_SoDienThoai:** Đây là ràng buộc duy nhất để đảm bảo rằng không có hai khách hàng nào có cùng số điện thoại.

### 2. Ràng buộc check:

- **CK\_TinhTrang:** Đảm bảo rằng cột TinhTrang chỉ chứa các giá trị 'Có sẵn', 'Đã thuê', hoặc 'Đang bảo trì'.
- **CK\_PhuongThuc:** Ràng buộc kiểm tra đảm bảo rằng NgayBatDau phải nhỏ hơn NgayKetThuc.
- **CK\_GiaDV:** Ràng buộc kiểm tra đảm bảo rằng giá trị của cột GiaDV phải lớn hơn 0.

### 3. Ràng buộc khóa ngoại:

- **FK\_HopDong\_MaNha:** Ràng buộc khóa ngoại đảm bảo rằng MaNha trong bảng HopDong phải tồn tại trong cột MaNha của bảng NhaChoThue. Khi một hàng trong NhaChoThue bị xóa, các hàng tương ứng trong HopDong cũng bị xóa.
- **FK\_HopDong\_MaKH:** Ràng buộc khóa ngoại đảm bảo rằng MaKH trong bảng HopDong phải tồn tại trong cột MaKH của bảng KhachHang. Khi một hàng trong KhachHang bị xóa, các hàng tương ứng trong HopDong cũng bị xóa.
- **FK\_HoaDon\_MaKH:** Ràng buộc khóa ngoại đảm bảo rằng MaKH trong bảng HoaDon phải tồn tại trong cột MaKH của bảng KhachHang. Khi một hàng trong KhachHang bị xóa, các hàng tương ứng trong HoaDon cũng bị xóa.
- **FK\_HoaDon\_MaNha:** Ràng buộc khóa ngoại đảm bảo rằng MaNha trong bảng HoaDon phải tồn tại trong cột MaNha của bảng NhaChoThue. Khi một hàng trong NhaChoThue bị xóa, các hàng tương ứng trong HoaDon cũng bị xóa.
- **FK\_HoaDon\_MaPT:** Ràng buộc khóa ngoại đảm bảo rằng MaPT trong bảng HoaDon phải tồn tại trong cột MaPT của bảng PhieuThue. Khi một hàng trong PhieuThue bị xóa, các hàng tương ứng trong HoaDon cũng bị xóa.
- **FK\_PhieuDichVu\_MaPT:** Ràng buộc khóa ngoại đảm bảo rằng MaPT trong bảng PhieuDichVu phải tồn tại trong cột MaPT của bảng PhieuThue. Khi một hàng trong PhieuThue bị xóa, các hàng tương ứng trong PhieuDichVu cũng bị xóa.

- **FK\_PhieuDichVu\_MaDV:** Ràng buộc khóa ngoại đảm bảo rằng MaDV trong bảng PhieuDichVu phải tồn tại trong cột MaDV của bảng DichVu. Khi một hàng trong DichVu bị xóa, các hàng tương ứng trong PhieuDichVu cũng bị xóa
- **FK\_PhieuThue\_MaNha:** Ràng buộc khóa ngoại đảm bảo rằng MaNha trong bảng PhieuThue phải tồn tại trong cột MaNha của bảng NhaChoThue. Khi một hàng trong NhaChoThue bị xóa, các hàng tương ứng trong PhieuThue cũng bị xóa.
- **FK\_PhieuThue\_MaKH:** Ràng buộc khóa ngoại đảm bảo rằng MaKH trong bảng PhieuThue phải tồn tại trong cột MaKH của bảng KhachHang. Khi một hàng trong KhachHang bị xóa, các hàng tương ứng trong PhieuThue cũng bị xóa.

## VII. Các bước cài đặt vật lý

### 1. Tạo database từ mô hình quan hệ tương ứng

```
DROP DATABASE IF EXISTS DB04;  
CREATE DATABASE DB04;  
USE DB04;
```

```
DROP TABLE IF EXISTS PhieuDichVu;  
DROP TABLE IF EXISTS PhieuThue;  
DROP TABLE IF EXISTS HoaDon;  
DROP TABLE IF EXISTS HopDong;  
DROP TABLE IF EXISTS NhaChoThue;  
DROP TABLE IF EXISTS KhachHang;  
DROP TABLE IF EXISTS DichVu;
```

```
CREATE TABLE KhachHang (  
    MaKH INT PRIMARY KEY AUTO_INCREMENT,  
    HoTen VARCHAR(100) NOT NULL,  
    SoDienThoai VARCHAR(15) NOT NULL UNIQUE,  
    CoQuan VARCHAR(100)  
);
```

```
CREATE TABLE NhaChoThue (  
    MaNha INT PRIMARY KEY AUTO_INCREMENT,  
    DiaChi VARCHAR(255),  
    GiaThue DECIMAL(10, 2),  
    TenChuNha VARCHAR(100),  
    TinhTrang ENUM('Có sẵn', 'Đã thuê', 'Đang bảo trì') NOT NULL  
);
```

```

CREATE TABLE HopDong (
    MaNha INT,
    MaKH INT,
    NgayBatDau DATE,
    NgayKetThuc DATE,
    PRIMARY KEY (MaNha, MaKH),
    FOREIGN KEY (MaNha) REFERENCES NhaChoThue(MaNha) ON DELETE CASCADE,
    FOREIGN KEY (MaKH) REFERENCES KhachHang(MaKH) ON DELETE CASCADE,
    CHECK (NgayBatDau < NgayKetThuc)
);

CREATE TABLE PhieuThue (
    MaPT INT PRIMARY KEY AUTO_INCREMENT,
    MaNha INT,
    MaKH INT,
    NgayBatDau DATE,
    NgayKetThuc DATE,
    GiaThue DECIMAL(10, 2),
    FOREIGN KEY (MaNha) REFERENCES NhaChoThue(MaNha) ON DELETE CASCADE,
    FOREIGN KEY (MaKH) REFERENCES KhachHang(MaKH) ON DELETE CASCADE,
    CHECK (NgayBatDau < NgayKetThuc)
);

CREATE TABLE HoaDon (
    MaHD INT PRIMARY KEY AUTO_INCREMENT,
    MaKH INT,
    MaNha INT,
    MaPT INT,
    NgayLap DATE,
    FOREIGN KEY (MaKH) REFERENCES KhachHang(MaKH) ON DELETE CASCADE,
    FOREIGN KEY (MaNha) REFERENCES NhaChoThue(MaNha) ON DELETE CASCADE,
    FOREIGN KEY (MaPT) REFERENCES PhieuThue(MaPT) ON DELETE CASCADE
);

CREATE TABLE DichVu (
    MaDV INT PRIMARY KEY AUTO_INCREMENT,
    TenDV VARCHAR(100) NOT NULL,
    GiaDV DECIMAL(10, 2) NOT NULL
);

CREATE TABLE PhieuDichVu (
    MaPhieuDV INT PRIMARY KEY AUTO_INCREMENT,
    MaPT INT,
    MaDV INT,
    NgayLap DATE,
    ThanhTien DECIMAL(15, 2),
    FOREIGN KEY (MaPT) REFERENCES PhieuThue(MaPT) ON DELETE CASCADE,
    FOREIGN KEY (MaDV) REFERENCES DichVu(MaDV) ON DELETE CASCADE
);

```

## 2. Tạo constraint cho dữ liệu

```
ALTER TABLE KhachHang
ADD CONSTRAINT UC_SoDienThoai UNIQUE (SoDienThoai);

ALTER TABLE NhaChoThue
ADD CONSTRAINT CK_TinhTrang CHECK (TinhTrang IN ('Có sẵn', 'Đã thuê', 'Đang bảo trì'));

ALTER TABLE HopDong
ADD CONSTRAINT FK_HopDong_MaNha FOREIGN KEY (MaNha) REFERENCES NhaChoThue(MaNha) ON DELETE CASCADE,
ADD CONSTRAINT FK_HopDong_MaKH FOREIGN KEY (MaKH) REFERENCES KhachHang(MaKH) ON DELETE CASCADE;

ALTER TABLE HoaDon
ADD CONSTRAINT FK_HoaDon_MaKH FOREIGN KEY (MaKH) REFERENCES KhachHang(MaKH) ON DELETE CASCADE,
ADD CONSTRAINT FK_HoaDon_MaNha FOREIGN KEY (MaNha) REFERENCES NhaChoThue(MaNha) ON DELETE CASCADE,
ADD CONSTRAINT FK_HoaDon_MaPT FOREIGN KEY (MaPT) REFERENCES PhieuThue(MaPT) ON DELETE CASCADE;

ALTER TABLE PhieuDichVu
ADD CONSTRAINT FK_PhieuDichVu_MaPT FOREIGN KEY (MaPT) REFERENCES PhieuThue(MaPT) ON DELETE CASCADE,
ADD CONSTRAINT FK_PhieuDichVu_MaDV FOREIGN KEY (MaDV) REFERENCES DichVu(MaDV) ON DELETE CASCADE;

ALTER TABLE PhieuThue
ADD CONSTRAINT CK_PhuongThuc CHECK (NgayBatDau < NgayKetThuc),
ADD CONSTRAINT FK_PhieuThue_MaNha FOREIGN KEY (MaNha) REFERENCES NhaChoThue(MaNha) ON DELETE CASCADE,
ADD CONSTRAINT FK_PhieuThue_MaKH FOREIGN KEY (MaKH) REFERENCES KhachHang(MaKH) ON DELETE CASCADE;

ALTER TABLE DichVu
ADD CONSTRAINT CK_GiaDV CHECK (GiaDV > 0);
```

## 3. Chèn dữ liệu vào Database

```
INSERT INTO KhachHang (MaKH, HoTen, SoDienThoai, CoQuan) VALUES
(1, 'Nguyễn Văn Linh', '0123456789', 'Bệnh viện Bạch Mai'),
(2, 'Trần Thị Xuyên Chi', '0987654321', 'Công ty Chứng khoán ACB'),
(3, 'Lê Hoàng Nguyễn', '0912345678', 'Trường Đại học FPT'),
(4, 'Phạm Văn Lập', '0908765432', 'Ngân hàng Vietcombank'),
(5, 'Nguyễn Thị Thanh Huyền', '0934567890', 'Bệnh viện Đa khoa Quốc tế Vinmec'),
(6, 'Đoàn Văn Hậu', '0923456789', 'Công ty Samsung Việt Nam'),
(7, 'Võ Thị Quế', '0945678901', 'Truyền hình VTV'),
(8, 'Phạm Minh Hiếu', '0967890123', 'Công ty Google Việt Nam'),
(9, 'Lê Thanh Huyền', '0954321789', 'Công ty Microsoft Việt Nam'),
(10, 'Nguyễn Hoàng Anh', '0976543210', 'Công ty VinGroup');

INSERT INTO NhaChoThue (MaNha, DiaChi, GiaThue, TenChuNha, TinhTrang) VALUES
(1, '123 Lê Lợi, Q1', 10000000.00, 'Đỗ Việt Trung', 'Có sẵn'),
(2, '456 Nguyễn Trãi, Q5', 8000000.00, 'Nguyễn Ngọc Bảo Long', 'Đã thuê'),
(3, '789 Trần Hưng Đạo, Q1', 15000000.00, 'Đỗ Quyên', 'Đang bảo trì'),
(4, '321 Cách Mạng Tháng 8, Q10', 7000000.00, 'Trần Đức Anh', 'Có sẵn'),
(5, '654 Điện Biên Phủ, Q3', 12000000.00, 'Đỗ Nhật Minh', 'Có sẵn'),
(6, '987 Lý Tự Trọng, Q1', 20000000.00, 'Vũ Trung Kiên', 'Đã thuê'),
(7, '213 Pasteur, Q3', 11000000.00, 'Đào Duy Huỳnh', 'Có sẵn'),
(8, '435 Hai Bà Trưng, Q1', 9000000.00, 'Nguyễn Hà Phương', 'Đã thuê'),
(9, '678 Võ Văn Tần, Q3', 17000000.00, 'Đỗ Hoàng Tú', 'Đang bảo trì'),
(10, '890 Hoàng Diệu, Q4', 6000000.00, 'Phạm Quốc Đạt', 'Có sẵn');
```

**INSERT INTO** HopDong (MaNha, MaKH, NgayBatDau, NgayKetThuc) **VALUES**

(1, 1, '2023-01-01', '2023-12-31'),  
(1, 2, '2023-02-01', '2023-12-31'),  
(2, 3, '2023-03-01', '2023-11-30'),  
(3, 4, '2023-01-15', '2023-10-15'),  
(4, 5, '2023-05-01', '2023-09-30'),  
(5, 6, '2023-04-01', '2023-12-31'),  
(6, 7, '2023-06-01', '2023-12-31'),  
(7, 8, '2023-07-01', '2023-12-31'),  
(8, 9, '2023-08-01', '2023-12-31'),  
(10, 10, '2023-09-01', '2023-12-31');

**INSERT INTO** PhieuThue (MaPT, MaNha, MaKH, NgayBatDau, NgayKetThuc, GiaThue) **VALUES**

(1, 1, 1, '2023-01-01', '2023-12-31', 10000000.00),  
(2, 1, 2, '2023-02-01', '2023-12-31', 10000000.00),  
(3, 2, 3, '2023-03-01', '2023-11-30', 15000000.00),  
(4, 3, 4, '2023-01-15', '2023-10-15', 7000000.00),  
(5, 4, 5, '2023-05-01', '2023-09-30', 12000000.00),  
(6, 5, 6, '2023-04-01', '2023-12-31', 20000000.00),  
(7, 6, 7, '2023-06-01', '2023-12-31', 11000000.00),  
(8, 7, 8, '2023-07-01', '2023-12-31', 9000000.00),  
(9, 8, 9, '2023-08-01', '2023-12-31', 17000000.00),  
(10, 10, 10, '2023-09-01', '2023-12-31', 6000000.00);

---

**INSERT INTO** HoaDon (MaHD, MaKH, MaNha, MaPT, NgayLap) **VALUES**

(1, 1, 1, 1, '2023-01-10'),  
(2, 2, 1, 2, '2023-02-10'),  
(3, 3, 2, 3, '2023-03-10'),  
(4, 4, 3, 4, '2023-01-20'),  
(5, 5, 4, 5, '2023-05-10'),  
(6, 6, 5, 6, '2023-04-15'),  
(7, 7, 6, 7, '2023-06-10'),  
(8, 8, 7, 8, '2023-07-10'),  
(9, 9, 8, 9, '2023-08-10'),  
(10, 10, 10, 10, '2023-09-15');

```
INSERT INTO DichVu (MaDV, TenDV, GiaDV) VALUES
```

```
(1, 'Dọn dẹp', 500000.00),  
(2, 'Bảo trì', 700000.00),  
(3, 'Internet', 300000.00),  
(4, 'Điện nước', 600000.00),  
(5, 'Vệ sinh công cộng', 450000.00);
```

```
INSERT INTO PhieuDichVu (MaPhieuDV, MaPT, MaDV, NgayLap, ThanhTien) VALUES
```

```
(1, 1, 1, '2023-01-10', 500000.00),  
(2, 1, 3, '2023-01-10', 300000.00),  
(3, 2, 2, '2023-02-10', 700000.00),  
(4, 2, 4, '2023-02-10', 600000.00),  
(5, 3, 1, '2023-03-10', 500000.00),  
(6, 3, 5, '2023-03-10', 450000.00),  
(7, 4, 3, '2023-01-20', 300000.00),  
(8, 4, 2, '2023-01-20', 700000.00),  
(9, 5, 4, '2023-05-10', 600000.00),  
(10, 5, 5, '2023-05-10', 450000.00),  
(11, 6, 1, '2023-04-15', 500000.00),  
(12, 6, 3, '2023-04-15', 300000.00),  
(13, 7, 2, '2023-06-10', 700000.00),  
(14, 7, 4, '2023-06-10', 600000.00),  
(15, 8, 3, '2023-07-10', 300000.00),  
(16, 8, 5, '2023-07-10', 450000.00),  
(17, 9, 4, '2023-08-10', 600000.00),  
(18, 9, 5, '2023-08-10', 450000.00),  
(19, 10, 4, '2023-09-15', 600000.00),  
(20, 10, 5, '2023-09-15', 450000.00);
```

## 4. Tạo Trigger

```
-- Trigger cập nhật giá thuê khi giá thuê của một nhà thay đổi  
DELIMITER //  
CREATE TRIGGER CapNhatGiaThueHopDong  
AFTER UPDATE ON NhaChoThue  
FOR EACH ROW  
BEGIN  
    IF OLD.GiaThue <> NEW.GiaThue THEN  
        UPDATE HopDong  
        SET GiaThue = NEW.GiaThue  
        WHERE MaNha = NEW.MaNha;  
    END IF;  
END; //  
DELIMITER ;
```

- **Trigger cập nhật giá thuê khi giá thuê của một nhà thay đổi:**

**Mục đích:** Cập nhật giá thuê trong bảng [HopDong](#) khi giá thuê của một nhà trong bảng [NhaChoThue](#) thay đổi.

**Hoạt động:** Sau khi một bản ghi trong [NhaChoThue](#) được cập nhật, nếu giá thuê thay đổi ([OLD.GiaThue](#) <> [NEW.GiaThue](#)), nó sẽ cập nhật tất cả các bản ghi trong [HopDong](#) có cùng [MaNha](#) với giá thuê mới.

```
-- Trigger kiểm tra xem ngày kết thúc hợp đồng có hợp lý không
DELIMITER //
CREATE TRIGGER KiemTraNgayKetThucHopDong
BEFORE INSERT ON HopDong
FOR EACH ROW
BEGIN
    IF NEW.NgayKetThuc <= NEW.NgayBatDau THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Ngày kết thúc hợp đồng phải sau ngày bắt đầu';
    END IF;
END; //
DELIMITER ;
```

- **Trigger kiểm tra xem ngày kết thúc hợp đồng có hợp lý không:**

**Mục đích:** Đảm bảo rằng ngày kết thúc hợp đồng luôn sau ngày bắt đầu.

**Hoạt động:** Trước khi một bản ghi mới được chèn vào bảng [HopDong](#), nếu [NgayKetThuc](#) <= [NgayBatDau](#), nó sẽ phát sinh lỗi với thông báo rằng ngày kết thúc phải sau ngày bắt đầu.

```
-- Trigger tự động cập nhật trạng thái nhà cho thuê khi có khách thuê
DELIMITER //
CREATE TRIGGER CapNhatTinhTrangNhaKhiThue
AFTER INSERT ON HopDong
FOR EACH ROW
BEGIN
    -- Cập nhật trạng thái nhà cho thuê khi có hợp đồng
    UPDATE NhaChoThue
    SET TinhTrang = 'Đã thuê'
    WHERE MaNha = NEW.MaNha;
END; //
DELIMITER ;
```

- **Trigger tự động cập nhật trạng thái nhà cho thuê khi có khách thuê:**

**Mục đích:** Tự động cập nhật trạng thái nhà cho thuê khi có hợp đồng thuê mới.

**Hoạt động:** Sau khi một bản ghi mới được chèn vào bảng [HopDong](#), nó sẽ cập nhật trạng thái của nhà trong bảng [NhaChoThue](#) thành 'Đã thuê'.



## 5. Tạo Procedures

```
-- Tính tổng tiền phải thanh toán của một phiếu thuê
DROP PROCEDURE IF EXISTS TinhTongTienPhieuThue;
DELIMITER $$

CREATE PROCEDURE TinhTongTienPhieuThue(IN MaPhieuThue INT)
BEGIN
    DECLARE SoThang INT;
    DECLARE TongTienThue DECIMAL(15,2);
    DECLARE TongTienDichVu DECIMAL(15, 2);
    DECLARE TongTien DECIMAL(15, 2);
    -- Tính số tháng
    SELECT
        (DATEDIFF(NgayKetThuc, NgayBatDau)/30)
    INTO SoThang
    FROM PhieuThue
    WHERE MaPT = MaPhieuThue;
    -- Tính tổng tiền dịch vụ
    SELECT COALESCE(SUM(ThanhTien), 0)
    INTO TongTienDichVu
    FROM PhieuDichVu
    WHERE MaPT = MaPhieuThue;
    -- Tính tổng tiền thuê trực tiếp (Giá thuê * Số tháng)
    SELECT GiaThue * SoThang
    INTO TongTienThue
    FROM PhieuThue
    WHERE MaPT = MaPhieuThue;
    -- Cộng tổng tiền thuê và tổng tiền dịch vụ để ra tổng tiền
    SET TongTien = TongTienThue + TongTienDichVu;
    -- Trả về kết quả
    SELECT TongTienThue AS TongTienThue, TongTienDichVu AS TongTienDichVu, TongTien AS TongTienPhiThue;
END $$

DELIMITER ;
```

- **Tính tổng tiền phải thanh toán của một phiếu thuê:**

**Mục đích:** Tính tổng tiền phải thanh toán cho một phiếu thuê.

**Hoạt động:**

- Tính số tháng thuê dựa trên **NgayBatDau** và **NgayKetThuc**.
- Tính tổng tiền dịch vụ bằng cách tính tổng thành tiền của các dịch vụ sử dụng.
- Tính tổng tiền thuê nhà bằng cách nhân Giá thuê với số tháng.
- Cộng tổng tiền dịch vụ và tổng tiền thuê để ra tổng tiền phải thanh toán.

```

-- Thêm thông tin khách hàng và hợp đồng
DROP PROCEDURE IF EXISTS ThemKhachHangVaHopDong;
DELIMITER $$
CREATE PROCEDURE ThemKhachHangVaHopDong(
    IN p_HoTen VARCHAR(100),
    IN p_SoDienThoai VARCHAR(15),
    IN p_CoQuan VARCHAR(100),
    IN p_MaNha INT,
    IN p_NgayBatDau DATE,
    IN p_NgayKetThuc DATE
)
BEGIN
    DECLARE newMaKH INT;
    INSERT INTO KhachHang (HoTen, SoDienThoai, CoQuan)
    VALUES (p_HoTen, p_SoDienThoai, p_CoQuan);
    SET newMaKH = LAST_INSERT_ID();
    INSERT INTO HopDong (MaNha, MaKH, NgayBatDau, NgayKetThuc)
    VALUES (p_MaNha, newMaKH, p_NgayBatDau, p_NgayKetThuc);
END $$
DELIMITER ;

```

- **Thêm thông tin khách hàng và hợp đồng:**

**Mục đích:** Thêm một khách hàng mới và hợp đồng thuê của khách hàng đó vào cơ sở dữ liệu.

**Hoạt động:**

- Chèn thông tin khách hàng mới vào bảng **KhachHang** và lấy **ID** vừa được tạo.
- Dùng **ID** này để chèn thông tin hợp đồng thuê vào bảng **HopDong**.

```

-- Cập nhật giá thuê nhà
DELIMITER $$
CREATE PROCEDURE CapNhatGiaThue(
    IN p_MaNha INT,
    IN p_GiaThue DECIMAL(10, 2))
BEGIN
    UPDATE NhaChoThue
    SET GiaThue = p_GiaThue
    WHERE MaNha = p_MaNha;
END$$
DELIMITER ;

-- Xóa hợp đồng thuê của một khách hàng
DELIMITER $$
CREATE PROCEDURE XoaHopDong(
    IN p_MaKH INT,
    IN p_MaNha INT)
BEGIN
    DELETE FROM HopDong
    WHERE MaKH = p_MaKH AND MaNha = p_MaNha;
END$$
DELIMITER ;

```

- **Cập nhật giá thuê nhà:**

**Mục đích:** Cập nhật giá thuê của một nhà trong bảng [NhaChoThue](#).

**Hoạt động:** Cập nhật cột [GiaThue](#) của nhà có [MaNha](#) tương ứng với giá thuê mới.

- **Xóa hợp đồng thuê của một khách hàng:**

**Mục đích:** Xóa hợp đồng thuê của một khách hàng đối với một nhà cụ thể.

**Hoạt động:** Xóa bản ghi trong bảng [HopDong](#) với [MaKH](#) và [MaNha](#) tương ứng.

## 6. Tạo Transaction

- **Thêm khách hàng và hợp đồng, sau đó hủy:**

**Mục đích:** Thêm mới một khách hàng vào bảng [KhachHang](#) và sau đó thêm hợp đồng thuê cho khách hàng này vào bảng [HopDong](#). Cuối cùng, hủy các thay đổi bằng cách sử dụng ROLLBACK.

- **Cập nhật tên chủ nhà:**

**Mục đích:** Cập nhật tên chủ nhà của một nhà cụ thể trong bảng [NhaChoThue](#). Cuối cùng, hủy các thay đổi bằng cách sử dụng ROLLBACK.

- **Xóa khách hàng và các thông tin liên quan:**

**Mục đích:** Xóa một khách hàng và tất cả các thông tin liên quan từ các bảng [HoaDon](#), [PhieuThue](#), [HopDong](#), và [KhachHang](#). Cuối cùng, hủy các thay đổi bằng cách sử dụng ROLLBACK.

- **Cập nhật trạng thái nhà và thêm phiếu thuê mới:**

**Mục đích:** Cập nhật trạng thái của một nhà trong bảng [NhaChoThue](#) và thêm một phiếu thuê mới vào bảng [PhieuThue](#). Cuối cùng, hủy các thay đổi bằng cách sử dụng ROLLBACK.

- **Đổi trạng thái nhà cho thuê và thêm dịch vụ mới:**

**Mục đích:** Cập nhật trạng thái của một nhà trong bảng [NhaChoThue](#), thêm một dịch vụ mới vào bảng [DichVu](#), và thêm phiếu dịch vụ mới vào bảng [PhieuDichVu](#). Cuối cùng, hủy các thay đổi bằng cách sử dụng ROLLBACK.

## 7. Tạo Queries

```
-- 1. Query using INNER JOIN
-- Lấy thông tin hợp đồng, khách hàng, và nhà cho thuê
SELECT KhachHang.HoTen AS TenKH, NhaChoThue.DiaChi AS DiaChiNha, HopDong.NgayBatDau, HopDong.NgayKetThuc
FROM HopDong
INNER JOIN KhachHang ON HopDong.MaKH = KhachHang.MaKH
INNER JOIN NhaChoThue ON HopDong.MaNha = NhaChoThue.MaNha;
```

- **Sử dụng INNER JOIN:**

**Mục đích:** Lấy thông tin hợp đồng, tên khách hàng và địa chỉ nhà cho thuê.

**Hoạt động:**

- INNER JOIN giữa HopDong và KhachHang dựa trên MaKH để lấy tên khách hàng.
- INNER JOIN giữa HopDong và NhaChoThue dựa trên MaNha để lấy địa chỉ nhà cho thuê.

```
-- 2. Query using OUTER JOIN
-- Lấy thông tin khách hàng và hợp đồng thuê
SELECT KhachHang.HoTen AS TenKH, NhaChoThue.DiaChi AS DiaChiNha, HopDong.NgayBatDau, HopDong.NgayKetThuc
FROM KhachHang
LEFT OUTER JOIN HopDong ON KhachHang.MaKH = HopDong.MaKH
LEFT OUTER JOIN NhaChoThue ON HopDong.MaNha = NhaChoThue.MaNha;
```

- **Sử dụng OUTER JOIN:**

**Mục đích:** Lấy thông tin khách hàng và hợp đồng thuê

**Hoạt động:**

- LEFT OUTER JOIN giữa KhachHang và HopDong dựa trên MaKH để lấy tên khách hàng.
- LEFT OUTER JOIN giữa HopDong và NhaChoThue dựa trên MaNha để lấy địa chỉ nhà cho thuê.

```
-- 3. Using Subquery in WHERE
-- Tìm khách hàng có hợp đồng thuê với giá thuê lớn hơn giá thuê trung bình
SELECT KhachHang.HoTen AS TenKH, PhieuThue.GiaThue
FROM KhachHang
INNER JOIN PhieuThue ON KhachHang.MaKH = PhieuThue.MaKH
WHERE PhieuThue.GiaThue > (SELECT AVG(GiaThue) FROM PhieuThue);
```

- **Sử dụng Subquery trong WHERE:**

**Mục đích:** Tìm các khách hàng có hợp đồng thuê với giá thuê lớn hơn giá thuê trung bình.

**Hoạt động:**

- INNER JOIN giữa KhachHang và PhieuThue dựa trên MaKH.
- Sử dụng Subquery trong WHERE để so sánh giá thuê với giá thuê trung bình.

```
-- 4. Using Subquery in FROM
-- Lấy danh sách khách hàng có tổng tiền thuê lớn hơn 100 triệu
SELECT KH.MaKH, KH.HoTen, KH.SoDienThoai, KH.CoQuan, TotalRent.TongTienThue
FROM (
    SELECT PT.MaKH, SUM(PT.GiaThue * (DATEDIFF(PT.NgayKetThuc, PT.NgayBatDau)/30)) AS TongTienThue
    FROM PhieuThue PT
    GROUP BY PT.MaKH
) AS TotalRent
JOIN KhachHang KH ON KH.MaKH = TotalRent.MaKH
WHERE TotalRent.TongTienThue > 100000000;
```

- **Sử dụng Subquery trong FROM:**

**Mục đích:** Lấy danh sách khách hàng có tổng tiền thuê lớn hơn 100 triệu.

**Hoạt động:**

- Subquery để tính tổng tiền thuê của mỗi khách hàng.
- Lọc các khách hàng có tổng tiền thuê lớn hơn 100 triệu.

```
-- 5. Query using GROUP BY and Aggregate Functions
-- In ra tiền thuê của các khách hàng
SELECT KhachHang.HoTen AS TenKH, SUM(PhieuThue.GiaThue) AS TongTienThue
FROM KhachHang
INNER JOIN PhieuThue ON KhachHang.MaKH = PhieuThue.MaKH
GROUP BY KhachHang.MaKH;
```

- **Sử dụng GROUP BY và Aggregate Functions:**

**Mục đích:** Tính tổng tiền thuê của mỗi khách hàng.

**Hoạt động:**

- INNER JOIN giữa KhachHang và PhieuThue dựa trên MaKH.
- Sử dụng GROUP BY và SUM để tính tổng tiền thuê cho mỗi khách hàng.

## VIII. Kết luận

Hệ thống cơ sở dữ liệu quản lý khách hàng trên được thiết kế để tự động hóa và tối ưu hóa các quy trình quản lý, từ việc lưu trữ thông tin khách hàng, nhà cho thuê đến việc quản lý hợp đồng, hóa đơn và các dịch vụ đi kèm. Hệ thống cung cấp một nền tảng quản lý chuyên nghiệp, giúp:

- **Tăng hiệu quả làm việc:** Tự động hóa các tác vụ thủ công, giảm thiểu sai sót và tiết kiệm thời gian.
- **Cải thiện chất lượng dịch vụ:** Đảm bảo thông tin luôn chính xác, cập nhật, giúp phục vụ khách hàng tốt hơn.
- **Tăng tính minh bạch:** Tất cả các giao dịch đều được ghi nhận và lưu trữ, tạo sự minh bạch và tin cậy.
- **Hỗ trợ ra quyết định:** Cung cấp các báo cáo thống kê chi tiết, giúp người dùng đưa ra các quyết định kinh doanh sáng suốt.
- **Giảm thiểu rủi ro:** Giảm thiểu các rủi ro phát sinh do quản lý thủ công.