**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: QUẢN LÝ CÔNG TY CHO THUÊ XE?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Lớp |
| 1 | 1771020705 | Phạm Đình Minh Trưởng | 30/04/2005 | CNTT 17-15 |
| 2 | 1771020717 | Vương Đức Tuấn | 02/11/2005 | CNTT 17-15 |

### 

**Hà Nội, năm 202**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: QUẢN LÝ CÔNG TY CHO THUÊ XE?**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020705 | Phạm Đình Minh Trưởng | 30/04/2005 |  |  |
| 2 | 1771020717 | Vương Đức Tuấn | 02/11/2005 |  |  |

### 

### CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2

**Hà Nội, năm 2022**

**LỜI NÓI ĐẦU**

Trong bối cảnh nền kinh tế phát triển mạnh mẽ, nhu cầu đi lại của con người ngày càng gia tăng. Các dịch vụ cho thuê xe ngày càng trở nên phổ biến, phục vụ nhiều mục đích như du lịch, công tác, vận chuyển hàng hóa và đưa đón khách hàng. Tuy nhiên, việc quản lý hoạt động cho thuê xe vẫn còn nhiều bất cập, đặc biệt là khi số lượng xe và khách hàng ngày càng lớn.

Hệ thống quản lý công ty cho thuê xe ra đời nhằm khắc phục những hạn chế trong việc lưu trữ, xử lý thông tin và theo dõi quá trình thuê xe. Hệ thống này giúp tự động hóa các công việc như quản lý xe, hợp đồng thuê xe, hóa đơn thanh toán, bảo trì xe và chăm sóc khách hàng, từ đó nâng cao hiệu suất làm việc và giảm thiểu sai sót trong quá trình quản lý.

Đề tài **"Xây dựng hệ thống quản lý công ty cho thuê xe"** được thực hiện nhằm thiết kế và triển khai một hệ thống giúp công ty dễ dàng kiểm soát các hoạt động kinh doanh, nâng cao chất lượng dịch vụ và tối ưu hóa việc sử dụng tài nguyên. Hệ thống này không chỉ giúp tiết kiệm thời gian mà còn hỗ trợ công ty ra quyết định dựa trên dữ liệu được thu thập và phân tích.

Trong quá trình nghiên cứu và thực hiện đề tài, chúng tôi đã tham khảo nhiều tài liệu, ứng dụng các công nghệ phù hợp và cố gắng hoàn thiện hệ thống một cách tốt nhất. Tuy nhiên, do giới hạn về thời gian và kiến thức, không thể tránh khỏi những thiếu sót. Chúng tôi rất mong nhận được sự góp ý từ thầy cô và các bạn để hoàn thiện hệ thống tốt hơn.

Xin chân thành cảm ơn!

**Mục Lục**

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI QUẢN LÝ CÔNG TY CHO THUÊ XE 5](#_Toc2042981354)

[1.1 Giới thiệu đề tài 5](#_Toc271584795)

[1.2 Lý do chọn đề tài 5](#_Toc1122899690)

[1.3 Mục tiêu nghiên cứu 5](#_Toc488799994)

[1.4 Phạm vi nghiên cứu 6](#_Toc1136549383)

[1.5 Phương pháp nghiên cứu 7](#_Toc118440016)

[CHƯƠNG 2: MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ 8](#_Toc2051491790)

[2.1. Xác định các thực thể, thuộc tính và ràng buộc 8](#_Toc1572561945)

[2.2. Xây dựng các bảng 13](#_Toc967365986)

[CHƯƠNG 3: TẠO CƠ SỞ DỮ LIỆU 19](#_Toc2068918912)

[3.1. Tạo Database 19](#_Toc264941760)

[3.2. Chèn Dữ Liệu 22](#_Toc1142560089)

[3.3. In bảng dữ liệu 25](#_Toc208644349)

[CHƯƠNG 5: XÂY DỰNG CÁC PROCEDURE 36](#_Toc1992261739)

*[5.1.1. Tổng quan về nội dung.](#_Toc378023349)* [36](#_Toc378023349)

[CHƯƠNG 6: XÂY DỰNG CÁC TRIGGER 46](#_Toc1181798817)

[6.1. Xây dựng các Trigger (Bộ kích hoạt) 46](#_Toc416466216)

[CHƯƠNG 7: PHÂN QUYỀN VÀ BẢO VỆ CƠ SỞ DỮ LIỆU 54](#_Toc898486739)

[7.1. Phân quyền trong SQL Server 54](#_Toc1909100316)

[7.2. Bảo vệ cơ sở dữ liệu 55](#_Toc695841185)

### 

# **CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI QUẢN LÝ CÔNG TY CHO THUÊ XE**

## **1.1 Giới thiệu đề tài**

Trong những năm gần đây, ngành dịch vụ cho thuê xe đã và đang phát triển mạnh mẽ do nhu cầu di chuyển ngày càng tăng. Các công ty cho thuê xe không chỉ phục vụ cá nhân mà còn cung cấp dịch vụ cho doanh nghiệp, tổ chức với nhiều loại hình xe khác nhau như xe du lịch, xe tự lái, xe hợp đồng dài hạn.

Tuy nhiên, việc quản lý dịch vụ cho thuê xe gặp nhiều khó khăn khi số lượng xe, khách hàng và hợp đồng ngày càng tăng. Các phương pháp quản lý truyền thống như sổ sách, file Excel không còn đáp ứng được yêu cầu của doanh nghiệp do dễ xảy ra sai sót, khó kiểm soát và không hiệu quả. Chính vì vậy, việc xây dựng một hệ thống quản lý công ty cho thuê xe giúp tối ưu hóa quy trình làm việc, giảm thiểu sai sót và nâng cao hiệu quả kinh doanh là điều cần thiết.

## **1.2 Lý do chọn đề tài**

Lựa chọn đề tài **"Xây dựng hệ thống quản lý công ty cho thuê xe"** xuất phát từ các lý do sau:

**+ Nhu cầu thực tế cao:** Hiện nay, nhiều doanh nghiệp kinh doanh dịch vụ cho thuê xe chưa có hệ thống quản lý chuyên nghiệp, dẫn đến khó khăn trong việc vận hành.

**+ Tối ưu hóa quy trình làm việc:** Hệ thống giúp tự động hóa nhiều công đoạn như quản lý khách hàng, hợp đồng, xe, hóa đơn thanh toán, bảo trì xe và chăm sóc khách hàng.

**+ Hạn chế rủi ro và sai sót:** Giảm thiểu các lỗi do con người gây ra trong quá trình nhập dữ liệu, theo dõi hợp đồng và thanh toán.

**+ Ứng dụng công nghệ vào thực tế:** Đề tài giúp áp dụng các công nghệ mới vào quản lý, từ đó nâng cao hiệu suất làm việc và khả năng cạnh tranh của doanh nghiệp.

## **1.3 Mục tiêu nghiên cứu**

Mục tiêu chính của đề tài là xây dựng một hệ thống quản lý cho công ty cho thuê xe nhằm:

**- Hỗ trợ quản lý danh sách xe:** Theo dõi thông tin chi tiết của từng xe như biển số, hãng xe, loại xe, tình trạng sử dụng.

**+ Quản lý khách hàng:** Lưu trữ thông tin khách hàng, lịch sử thuê xe và các giao dịch liên quan.

**+ Quản lý hợp đồng thuê xe:** Theo dõi hợp đồng thuê xe, thời gian thuê, giá thuê, các điều khoản hợp đồng.

**+ Quản lý hóa đơn và thanh toán:** Hỗ trợ tính toán chi phí, xuất hóa đơn và theo dõi thanh toán.

**+ Quản lý bảo trì xe:** Định kỳ bảo trì xe, thông báo lịch bảo trì và lưu trữ thông tin sửa chữa.

**+ Quản lý chăm sóc khách hàng:** Ghi nhận phản hồi của khách hàng, hỗ trợ giải quyết các vấn đề phát sinh.

## **1.4 Phạm vi nghiên cứu**

- Phạm vi nghiên cứu của đề tài tập trung vào:

**q+ Đối tượng nghiên cứu:** Hệ thống quản lý công ty cho thuê xe, bao gồm các nghiệp vụ liên quan đến khách hàng, xe, hợp đồng, hóa đơn và bảo trì.

**+ Phạm vi ứng dụng:** Ứng dụng trong các công ty cung cấp dịch vụ cho thuê xe du lịch, xe tự lái, xe hợp đồng ngắn hạn và dài hạn.

**+ Công nghệ sử dụng:** Đề tài áp dụng mô hình thực thể quan hệ (ER) để thiết kế cơ sở dữ liệu, sử dụng SQL Server để quản lý dữ liệu và có thể triển khai trên nền tảng web hoặc phần mềm desktop.

## **1.5 Phương pháp nghiên cứu**

Để thực hiện đề tài, các phương pháp nghiên cứu chính được sử dụng bao gồm:

**+ Phương pháp khảo sát:** Nghiên cứu thực tế hoạt động của các công ty cho thuê xe để xác định các nghiệp vụ quan trọng.

**+ Phương pháp phân tích và thiết kế hệ thống:** Sử dụng mô hình hóa dữ liệu (ERD), thiết kế cơ sở dữ liệu và xây dựng quy trình hoạt động của hệ thống.

**+ Phương pháp lập trình và thử nghiệm:** Triển khai hệ thống, kiểm tra tính ổn định và hiệu suất của hệ thống.

**+ Phương pháp đánh giá:** Thu thập ý kiến phản hồi từ người dùng để cải thiện và tối ưu hệ thống.

**CHƯƠNG 2: MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ**

**2.1. Xác định các thực thể, thuộc tính và ràng buộc**

## *****1. Bảng*** *KhachHang* ***(Quản lý khách hàng)*****

****Thuộc tính:****

MaKH (INT, PRIMARY KEY, NOT NULL) – Mã khách hàng.

HoTen (VARCHAR(50), NOT NULL) – Họ và tên khách hàng.

CMND (VARCHAR(12), UNIQUE, NOT NULL) – Số CMND/CCCD.

SDT (VARCHAR(15), NOT NULL) – Số điện thoại.

DiaChi (TEXT, NOT NULL) – Địa chỉ khách hàng.

Email (VARCHAR(50), UNIQUE, NULL) – Email khách hàng.

****Ràng buộc:****

MaKH là **khóa chính** (PRIMARY KEY).

CMND là **duy nhất** (UNIQUE), không được trùng.

Email là **duy nhất** (UNIQUE).

HoTen, SDT, DiaChi không được để trống (NOT NULL).

## *****2. Bảng*** *Xe* ***(Quản lý xe cho thuê)*****

**Thuộc tính:**

MaXe (INT, PRIMARY KEY, NOT NULL) – Mã xe.

BienSo (VARCHAR(15), UNIQUE, NOT NULL) – Biển số xe.

HangXe (VARCHAR(30), NOT NULL) – Hãng xe.

LoaiXe (VARCHAR(30), NOT NULL) – Loại xe (xe máy, ô tô, xe tải...).

NamSX (YEAR, NOT NULL) – Năm sản xuất.

TinhTrang (VARCHAR(20), NOT NULL) – Trạng thái xe (Sẵn sàng, Đang thuê, Bảo trì).

GiaThueNgay (DECIMAL(10,2), NOT NULL) – Giá thuê theo ngày.

**Ràng buộc:**

MaXe là **khóa chính** (PRIMARY KEY).

BienSo là **duy nhất** (UNIQUE).

GiaThueNgay không được âm.

TinhTrang chỉ nhận giá trị hợp lệ (‘Sẵn sàng’, ‘Đang thuê’, ‘Bảo trì’).

## ***3. Bảng**HopDongThue (Quản lý hợp đồng thuê)***

**Thuộc tính:**

MaHD (INT, PRIMARY KEY, NOT NULL) – Mã hợp đồng.

MaKH (INT, FOREIGN KEY, NOT NULL) – Mã khách hàng.

MaXe (INT, FOREIGN KEY, NOT NULL) – Mã xe thuê.

NgayBatDau (DATE, NOT NULL) – Ngày bắt đầu thuê.

NgayKetThuc (DATE, NOT NULL) – Ngày kết thúc thuê.

TongTien (DECIMAL(10,2), NOT NULL) – Tổng tiền thuê.

**Ràng buộc:**

MaHD là **khóa chính** (PRIMARY KEY).

MaKH là **khóa ngoại** (FOREIGN KEY) tham chiếu KhachHang(MaKH).

MaXe là **khóa ngoại** (FOREIGN KEY) tham chiếu Xe(MaXe).

NgayBatDau phải nhỏ hơn NgayKetThuc.

TongTien phải lớn hơn 0.

## ***4. Bảng**NhanVien (Quản lý nhân viên)***

**Thuộc tính:**

MaNV (INT, PRIMARY KEY, NOT NULL) – Mã nhân viên.

HoTen (VARCHAR(50), NOT NULL) – Họ và tên nhân viên.

ChucVu (VARCHAR(30), NOT NULL) – Chức vụ.

SDT (VARCHAR(15), NOT NULL) – Số điện thoại nhân viên.

Email (VARCHAR(50), UNIQUE, NOT NULL) – Email nhân viên.

Luong (DECIMAL(10,2), NOT NULL) – Lương nhân viên.

**Ràng buộc:**

MaNV là **khóa chính** (PRIMARY KEY).

Email là **duy nhất** (UNIQUE).

Luong phải lớn hơn 0.

## ***5. Bảng PhieuThanhToan (Quản lý thanh toán)***

**Thuộc tính:**

MaPTT (INT, PRIMARY KEY, NOT NULL) – Mã phiếu thanh toán.

MaHD (INT, FOREIGN KEY, NOT NULL) – Mã hợp đồng.

NgayThanhToan (DATE, NOT NULL) – Ngày thanh toán.

SoTien (DECIMAL(10,2), NOT NULL) – Số tiền thanh toán.

PhuongThucTT (VARCHAR(20), NOT NULL) – Phương thức thanh toán (Tiền mặt, Chuyển khoản).

**Ràng buộc:**

MaPTT là **khóa chính** (PRIMARY KEY).

MaHD là **khóa ngoại** (FOREIGN KEY) tham chiếu HopDongThue(MaHD).

NgayThanhToan phải sau hoặc bằng NgayKetThuc của hợp đồng.

SoTien phải lớn hơn 0.

1. *****Bảng Hóa Đơn (HoaDon)*****

****Thuộc tính**:**

MaHD (INT, Khóa chính)

MaHopDong (INT, Khóa ngoại tham chiếu HopDongThue)

NgayThanhToan (DATE, Bắt buộc)

SoTien (DECIMAL(10,2), >= 0)

TrangThai (NVARCHAR(20), Chỉ nhận 'Đã thanh toán' hoặc 'Chưa thanh toán', mặc định 'Chưa thanh toán')

****Ràng buộc**:**

PRIMARY KEY (MaHD)

FOREIGN KEY (MaHopDong) REFERENCES HopDongThue(MaHD) ON DELETE CASCADE

CHECK (SoTien >= 0)

CHECK (TrangThai IN ('Đã thanh toán', 'Chưa thanh toán'))

1. *****Bảng Chăm Sóc Khách Hàng (ChamSocKH)*****

****Thuộc tính**:**

MaCSKH (INT, Khóa chính)

MaKH (INT, Khóa ngoại tham chiếu KhachHang)

MaNV (INT, Khóa ngoại tham chiếu NhanVien, có thể NULL)

NgayChamSoc (DATETIME, Mặc định GETDATE())

HinhThuc (NVARCHAR(50), Chỉ nhận 'Gọi điện', 'Email', 'Tin nhắn', 'Gặp mặt')

NoiDung (NVARCHAR(500), Bắt buộc)

GhiChu (NVARCHAR(255), Có thể NULL)

****Ràng buộc**:**

PRIMARY KEY (MaCSKH)

FOREIGN KEY (MaKH) REFERENCES KhachHang(MaKH) ON DELETE CASCADE

FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV) ON DELETE SET NULL

CHECK (HinhThuc IN ('Gọi điện', 'Email', 'Tin nhắn', 'Gặp mặt'))

1. *****Bảng Bảo Trì Xe (BaoTriXe)*****

****Thuộc tính**:**

MaBaoTri (INT, Khóa chính)

MaXe (INT, Khóa ngoại tham chiếu Xe)

NgayBaoTri (DATE, Mặc định GETDATE())

ChiPhi (DECIMAL(10,2), >= 0)

MoTa (NVARCHAR(500), Bắt buộc)

GhiChu (NVARCHAR(255), Có thể NULL)

****Ràng buộc**:**

PRIMARY KEY (MaBaoTri)

FOREIGN KEY (MaXe) REFERENCES Xe(MaXe) ON DELETE CASCADE

CHECK (ChiPhi >= 0)

**2.2. Xây dựng các bảng**

***1. Bảng Khách hàng (****KhachHang****)***

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MaKH | INT | PRIMARY KEY, IDENTITY(1,1) | Mã khách hàng |
| HoTen | NVARCHAR(100) | NOT NULL | Họ,tên khách hàng |
| CMND | VARCHAR(12) | UNIQUE, NOT NULL | CMND khách hàng |
| SDT | VARCHAR(15) | NULL | Số điện thoại khách hàng |
| DiaChi | NVARCHAR(255) | NULL | Địa chỉ khách hàng |
| Email | VARCHAR(100) | UNIQUE, NULL | Email khách hàng |

1. ***Bảng Xe (xe)***

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MaXe | INT | PRIMARY KEY, IDENTITY(1,1) | Mã xe (tự động tăng) |
| BienSo | VARCHAR(15) | UNIQUE, NOT NULL | Biển số xe duy nhất |
| HangXe | NVARCHAR(50) | NOT NULL | Hãng xe (Toyota, Honda,...) |
| Model | NVARCHAR(50) | NOT NULL | Dòng xe cụ thể |
| NamSX | INT | NOT NULL, CHECK (NamSX >= 1900 AND NamSX <= YEAR  (GETDATE())) | Năm sản xuất xe |
| GiaThue | DECIMAL(10,2) | NOT NULL, CHECK (GiaThue > 0) | Giá thuê xe mỗi ngày |
| TrangThai | NVARCHAR(20) | NOT NULL, DEFAULT 'Có sẵn', CHECK (TrangThai IN ('Có sẵn', 'Đang thuê', 'Bảo trì')) | Trạng thái hiện tại của xe |

1. ***Bảng Nhân Viên (NhanVien)***

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MaNV | INT | PRIMARY KEY, IDENTITY(1,1) | Mã nhân viên (tự động tăng) |
| HoTen | NVARCHAR(100) | NOT NULL | Họ và tên nhân viên |
| ChucVu | NVARCHAR(50) | NULL | Chức vụ của nhân viên |
| SDT | VARCHAR(15) | NULL | Số điện thoại nhân viên |
| Email | VARCHAR(100) | UNQUE, NULL | Email nhân viên |
| DiaChi | NVARCHAR(255) | NULL | Địa chỉ nhân viên |

1. ***Bảng Hợp Đồng Thuê Xe (HopDongThue)***

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MaHD | INT | PRIMARY KEY, IDENTITY(1,1) | Mã hợp đồng thuê xe |
| MaKH | INT | FOREIGN KEY REFERENCES KhachHang(MaKH), NOT NULL, ON DELETE CASCADE | Mã khách hàng |
| MaXe | INT | FOREIGN KEY REFERENCES Xe(MaXe), NOT NULL, ON DELETE CASCADE | Mã xe thuê |
| MaNV | INT | FOREIGN KEY REFERENCES NhanVien(MaNV), NULL, ON DELETE SET NULL | Nhân viên phụ trách |
| NgayBatDau | DATE | NOT NULL | Ngày bắt đầu thuê xe |
| NgayKetThuc | DATE | NOT NULL | Ngày kết thúc thuê xe |
| TongTien | DECIMAL(10,2) | NOT NULL, CHECK (TongTien >= 0) | Tổng tiền thuê xe |
| TrangThai | NVARCHAR(20) | NOT NULL, DEFAULT 'Đang hiệu lực', CHECK (TrangThai IN ('Đang hiệu lực', 'Hoàn thành', 'Hủy bỏ'))NOT NULL, DEFAULT 'Đang hiệu lực', CHECK (TrangThai IN ('Đang hiệu lực', 'Hoàn thành', 'Hủy bỏ')) | Trạng thái hợp đồng |

1. ***Bảng Hóa Đơn (HoaDon)***

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MaHD | INT | PRIMARY KEY, IDENTITY(1,1) | Mã hóa đơn |
| MaHopDong | INT | FOREIGN KEY REFERENCES HopDongThue(MaHD), NOT NULL, ON DELETE CASCADE | Mã hợp đồng liên quan |
| NgayThanhToan | DATE | NOT NULL, DEFAULT GETDATE() | Ngày thanh toán hóa đơn |
| SoTien | DECIMAL(10,2) | NOT NULL, CHECK (SoTien >= 0) | Số tiền thanh toán |
| TrangThai | NVARCHAR(20) | NOT NULL, DEFAULT 'Chưa thanh toán', CHECK (TrangThai IN ('Đã thanh toán', 'Chưa thanh toán')) | Trạng thái thanh toán |

1. ***Bảng Chăm Sóc Khách Hàng (ChamSocKH)***

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MaCSKH | INT | PRIMARY KEY, IDENTITY(1,1) | Mã chăm sóc khách hàng |
| MaKH | INT | FOREIGN KEY REFERENCE KhachHang (MaKH), NOT NULL, ON DELETE CASCADE | Mã khách hàng |
| MaNV | INT | FOREIGN KEY REFERENCES NhanVien(MaNV), NULL, ON DELETE SET NULL | Mã nhân viên chăm sóc |
| NgayChamSoc | DATETIME | NOT NULL, DEFAULT GETDATE() | Ngày chăm sóc khách hàng |
| HinhThuc | NVARCHAR(50) | NOT NULL, CHECK (HinhThuc IN ('Gọi điện', 'Email', 'Tin nhắn', 'Gặp mặt')) | Hình thức chăm sóc |
| NoiDung | NVARCHAR(500) | NOT NULL | Nội dung chăm sóc khách hàng |

1. ***Bảng Bảo Trì Xe (BaoTriXe)***

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MaBaoTri | INT | PRIMARY KEY, IDENTITY(1,1) | Mã bảo trì xe |
| MaXe | INT | FOREIGN KEY REFERENCES Xe(MaXe), NOT NULL, ON DELETE CASCADE | Mã xe cần bảo trì |
| NgayBaoTri | DATE | NOT NULL, DEFAULT GETDATE() | Ngày bảo trì xe |
| ChiPhi | DECIMAL(10,2) | NOT NULL, CHECK (ChiPhi >= 0) | Chi phí bảo trì |
| MoTa | NVARCHAR(500) | NOT NULL | Mô tả công việc bảo trì |

**CHƯƠNG 3: TẠO CƠ SỞ DỮ LIỆU**

## **3.1. Tạo Database**

### **Database Diagram**

**+ KhachHang** (MaKH) liên kết với **HopDongThue** (MaKH)

**+ HopDongThue** (MaHD) liên kết với **HoaDon** (MaHopDong)

**+ HopDongThue** (MaXe) liên kết với **Xe** (MaXe)

**+ HopDongThue** (MaNV) liên kết với **NhanVien** (MaNV)

**+ NhanVien** (MaNV) liên kết với **ChamSocKH** (MaNV)

**+ KhachHang** (MaKH) liên kết với **ChamSocKH** (MaKH)

**+ NhanVien** (MaNV) liên kết với **BaoTriXe** (MaXe)

### \* Mô tả chi tiết các bảng

**+ Bảng Khách Hàng**: Lưu thông tin khách hàng bao gồm họ tên, CMND, số điện thoại, địa chỉ và email.

**+ Bảng Xe**: Chứa thông tin về xe cho thuê như biển số, hãng xe, model, năm sản xuất, giá thuê và trạng thái hiện tại.

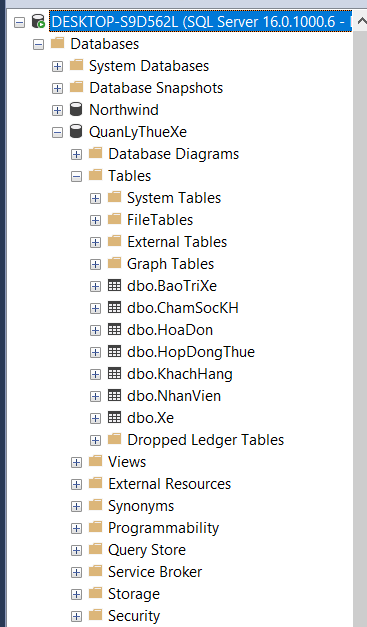
**+ Bảng Nhân Viên**: Lưu trữ thông tin nhân viên bao gồm họ tên, chức vụ, số điện thoại, email và địa chỉ.

**+ Bảng Hợp Đồng Thuê**: Quản lý hợp đồng giữa khách hàng và công ty, bao gồm thông tin xe thuê, nhân viên phụ trách, ngày bắt đầu, ngày kết thúc và tổng tiền.

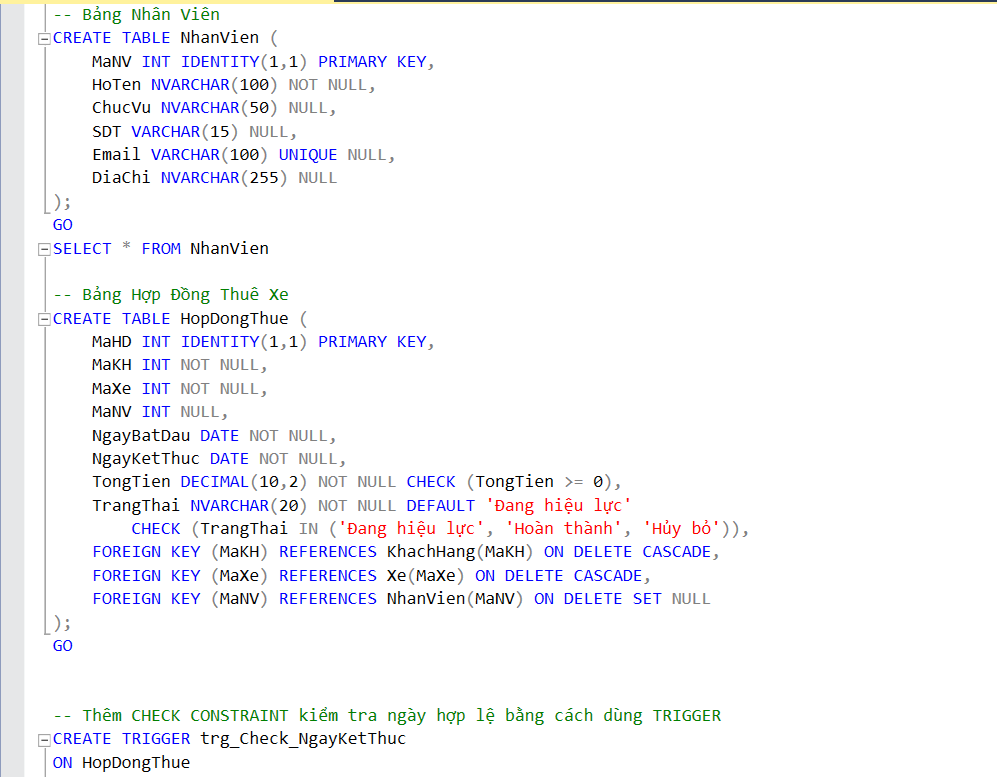
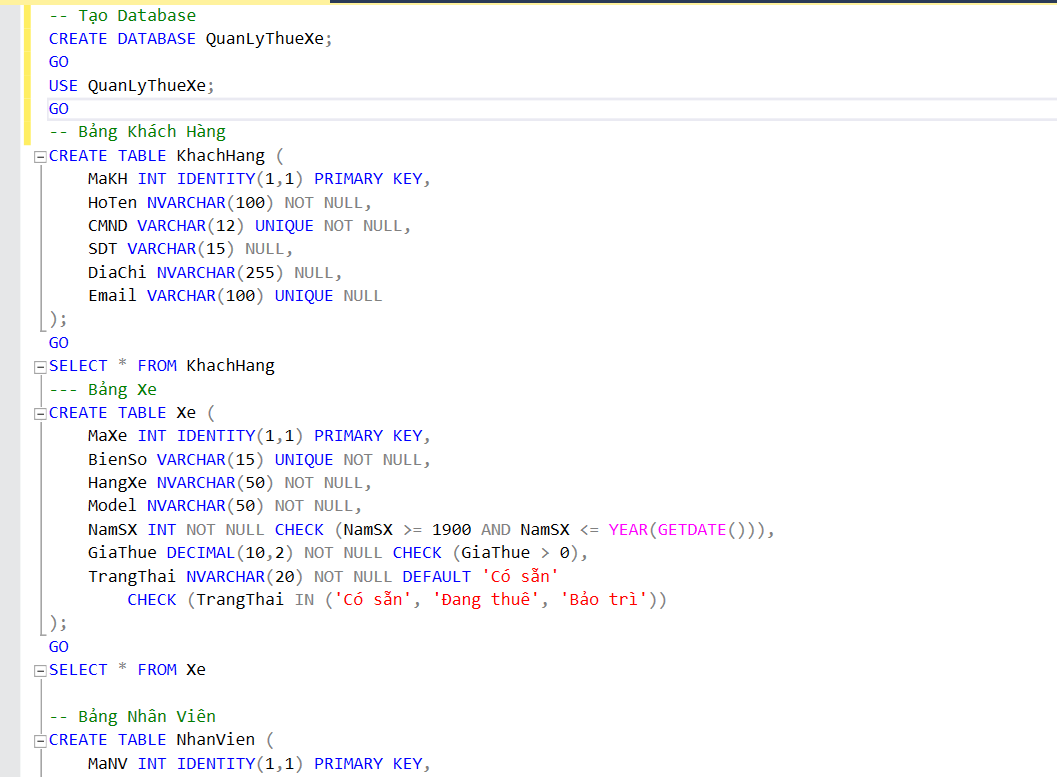
**+ Bảng Hóa Đơn**: Lưu trữ thông tin thanh toán cho hợp đồng.

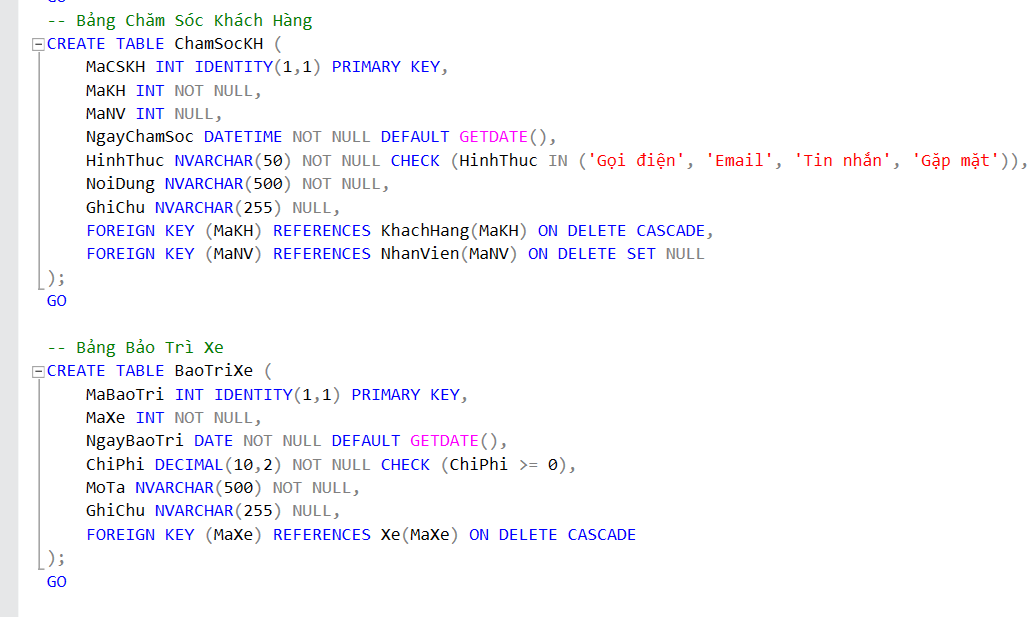
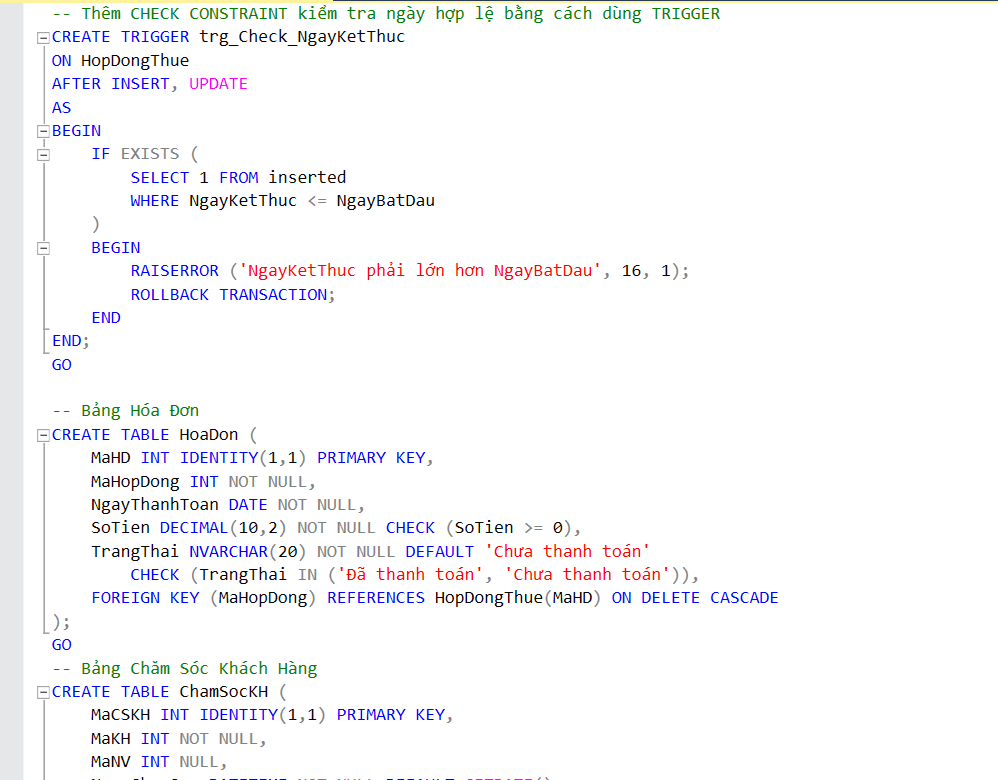
**+ Bảng hăm Sóc Khách Hàng**: Ghi lại các lần chăm sóc khách hàng, bao gồm ngày chăm sóc, hình thức, nội dung và nhân viên thực hiện.

**+ Bảng Bảo Trì Xe**: Ghi nhận các lần bảo trì xe, chi phí và mô tả chi tiết.



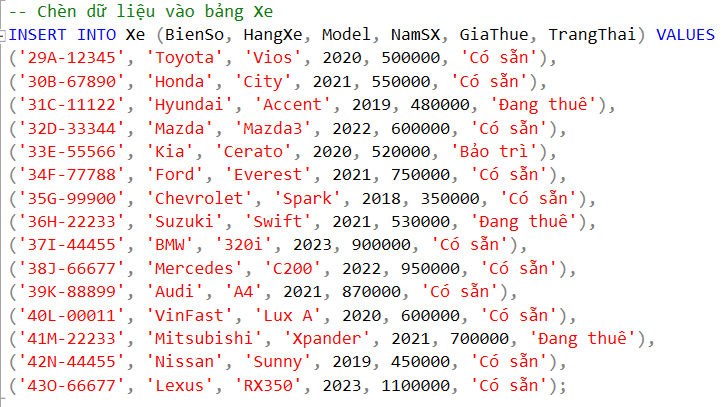
### Câu lệnh SQL

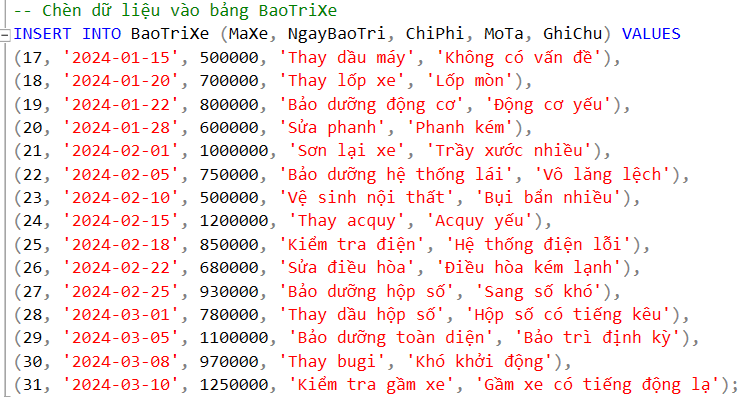
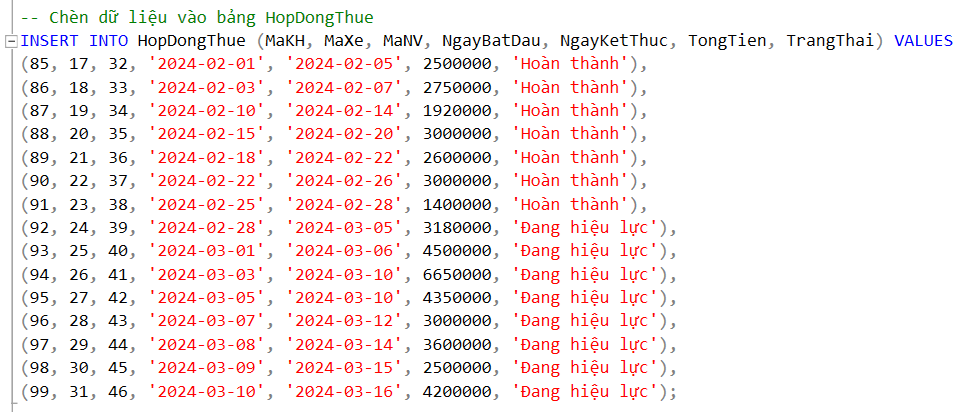
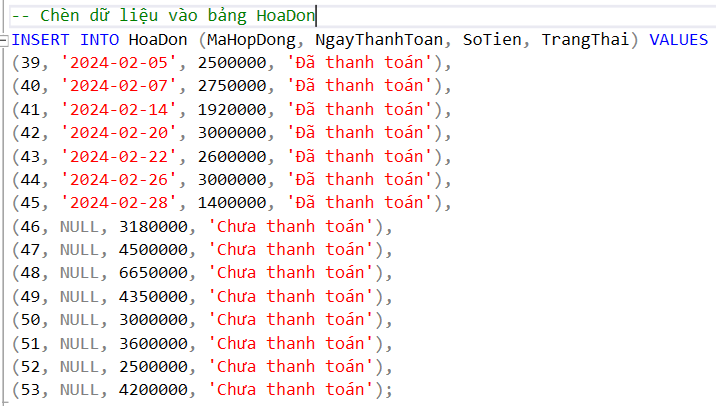


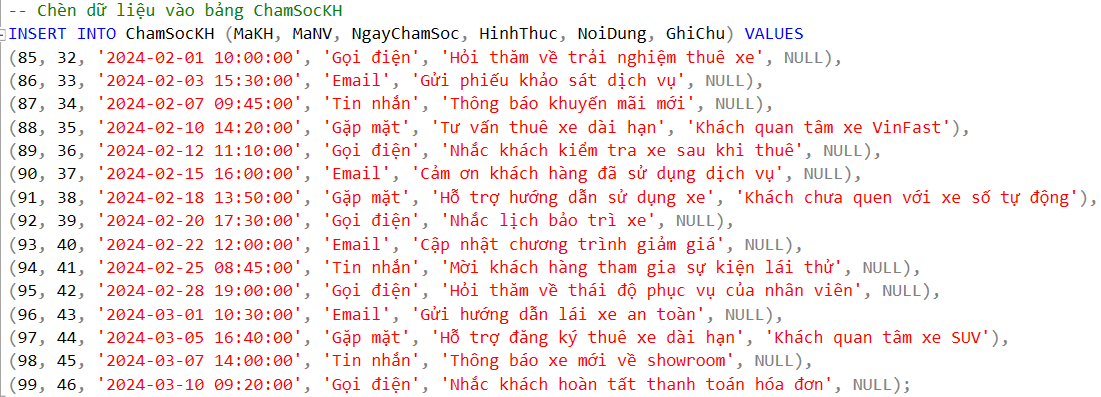


## **3.2. Chèn Dữ Liệu**

### Câu lệnh SQL:





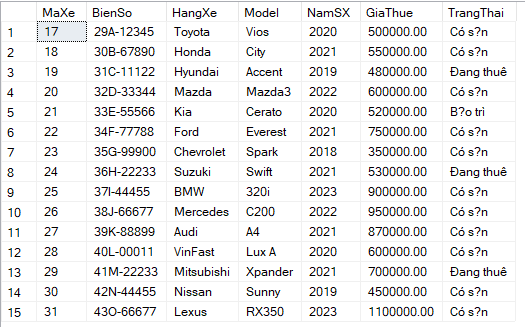


**3.3. In bảng dữ liệu**

**1. Bảng Khách Hàng**



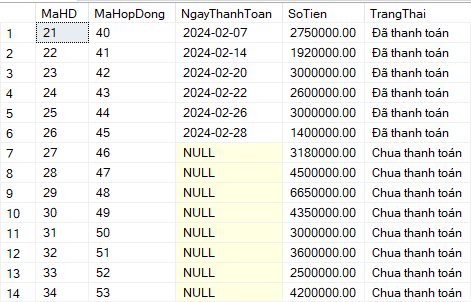
1. Bảng Xe.



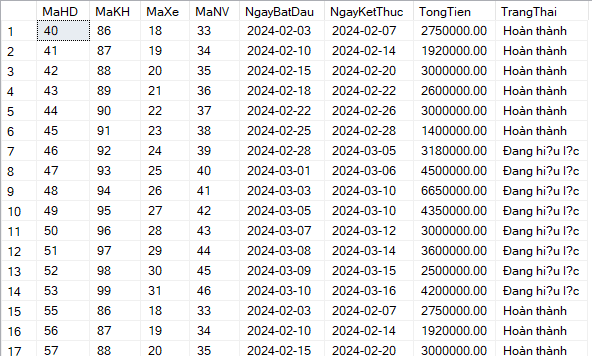
1. Bảng Nhân Viên



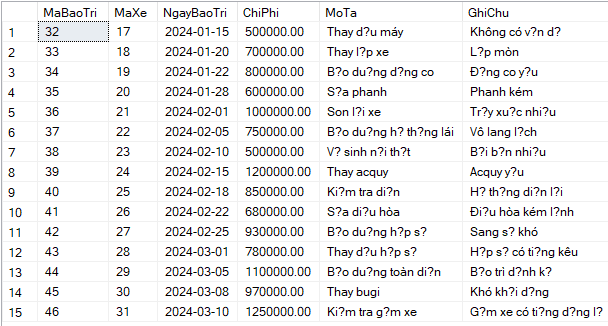
1. Bảng Hợp Đồng



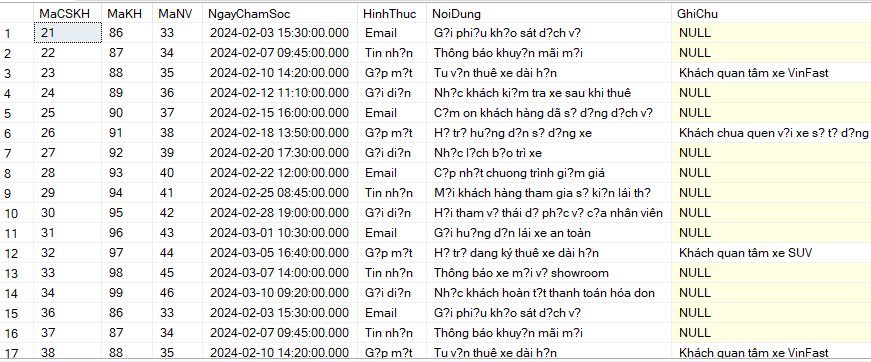
1. Bảng Hợp Đồng Thuê



1. Bảng Bảo Trì Xe



1. Bảng Chăm Sóc Khách Hàng



**CHƯƠNG 4: XÂY DỰNG CÁC VIEW .**

**4.1. Xây dựng các View**

***4.1.1. Tổng quan về nội dung***

**View** là một bảng ảo được định nghĩa bằng một câu lệnh SELECT. Khi người dùng truy vấn View, hệ quản trị cơ sở dữ liệu sẽ thực thi câu lệnh đó và trả về kết quả tương ứng.

### ****\* Ưu điểm của View****

**Giúp ẩn đi sự phức tạp của truy vấn SQL**: Thay vì viết một truy vấn phức tạp nhiều lần, View giúp đơn giản hóa bằng cách tạo một bảng ảo để sử dụng lại.

**Cải thiện bảo mật dữ liệu**: Người dùng chỉ có thể xem dữ liệu theo View mà không thể truy cập toàn bộ bảng gốc.

**Tăng tính nhất quán dữ liệu**: Nếu nhiều ứng dụng sử dụng cùng một truy vấn, View đảm bảo rằng tất cả đều truy xuất dữ liệu giống nhau.

### ****\* Nhược điểm của View****

**Giảm hiệu suất trong một số trường hợp**: Mỗi khi truy vấn View, hệ thống phải thực thi lại câu lệnh SELECT.

**Không thể cập nhật trực tiếp trong một số View phức tạp**: Nếu View chứa JOIN, GROUP BY, hoặc DISTINCT, việc cập nhật dữ liệu có thể không thực hiện được.

## **\* Các Đặc điểm**

+ View **không lưu trữ dữ liệu thực tế**, chỉ lưu câu lệnh truy vấn.

+ Khi dữ liệu trong bảng gốc thay đổi, dữ liệu trong View cũng thay đổi tương ứng.

+ Một số View có thể cập nhật được (Updatable View), nhưng chỉ trong các điều kiện nhất định.

### ****\* Ứng dụng của View****

**+ Đơn giản hóa truy vấn phức tạp**, giúp lập trình viên dễ dàng truy xuất dữ liệu.

**+ Tăng cường bảo mật bằng cách ẩn đi các cột hoặc hàng nhạy cảm**.

**+ Tạo giao diện ổn định cho ứng dụng**, ngay cả khi cấu trúc bảng thay đổi.

***4.1.2. Câu lệnh SQL.***

**1. v\_DanhSachKhachHang**

**\* Tác dụng:**

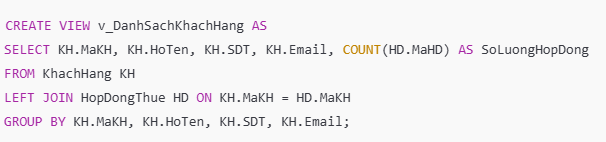
Hiển thị danh sách khách hàng cùng với số lượng hợp đồng đã ký.

**\* Giải thích:**

+ Lấy danh sách khách hàng (KhachHang), kết hợp với bảng HopDongThue bằng LEFT JOIN.

+ Dùng COUNT(HD.MaHD) để đếm số hợp đồng mà mỗi khách hàng đã ký.

+ Nếu khách hàng chưa có hợp đồng nào, số lượng hợp đồng sẽ là 0.



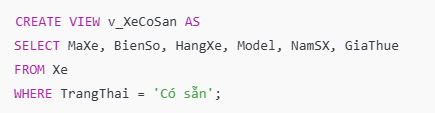
1. **v\_XeCoSan**

**\* Tác dụng:**

Liệt kê danh sách các xe có thể thuê.

**\* Giải thích:**

Chỉ hiển thị các xe đang ở trạng thái "Có sẵn" để khách hàng có thể thuê ngay.



1. **v\_TongTienHoaDon**

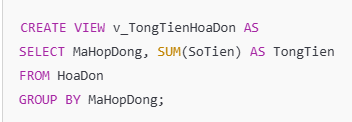
**\* Tác dụng:**

Tính tổng số tiền của từng hợp đồng từ các hóa đơn.

**\* Giải thích:**

Gom nhóm (GROUP BY) theo MaHopDong để tính tổng số tiền (SUM(SoTien)) mà mỗi

hợp đồng đã thanh toán.



1. **v\_NhanVienHopDong**

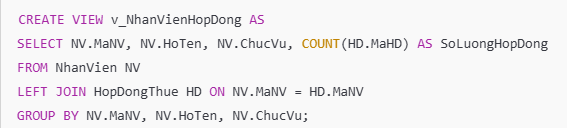
### **\* Tác dụng**

Hiển thị danh sách nhân viên cùng số lượng hợp đồng họ đã quản lý.

### **\* Giải thích**

+ Lấy MaNV, HoTen, ChucVu từ bảng NhanVien.

+ Đếm số hợp đồng (SoLuongHopDong) do nhân viên đó xử lý từ HopDongThue.



1. **v\_HopDongHieuLuc**

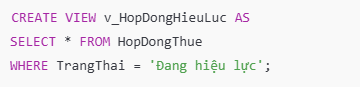
### **\* Tác dụng**

Lọc ra các hợp đồng đang có hiệu lực (chưa kết thúc).

### **\* Giải thích**

+ Lấy tất cả thông tin từ HopDongThue.

+ Chỉ lấy các hợp đồng có trạng thái "Đang hiệu lực".



1. **v\_ChiPhiBaoTriNam**

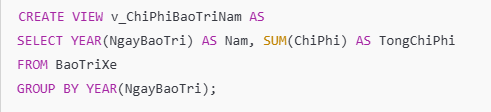
### **\* Tác dụng**

Thống kê tổng chi phí bảo trì xe theo từng năm.

### **\* Giải thích**

+ Lấy năm (YEAR(NgayBaoTri)) từ bảng BaoTriXe.

+ Tính tổng chi phí bảo trì của từng năm.



1. **v\_XeDangThue**

### **\* Tác dụng**

Liệt kê các xe hiện đang được thuê cùng thông tin khách hàng và hợp đồng.

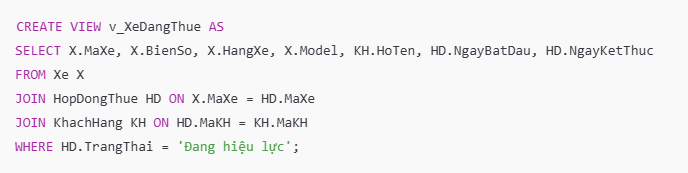
### **\* Giải thích**

+ Lấy thông tin xe (MaXe, BienSo, HangXe, Model).

+ Lấy thông tin khách hàng (HoTen).

+ Lấy ngày thuê (NgayBatDau, NgayKetThuc).

+ Chỉ lấy các hợp đồng đang có hiệu lực (HD.TrangThai = 'Đang hiệu lực').



1. **v\_TongXeTheoHang**

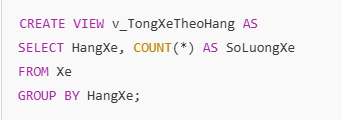
### **\* Tác dụng**

Thống kê tổng số xe theo từng hãng xe.

### **\* Giải thích**

+ Nhóm theo HangXe.

+ Đếm số lượng xe thuộc từng hãng (SoLuongXe).



1. **v\_HoaDonChuaThanhToan**

### **\* Tác dụng**

Lọc ra các hóa đơn chưa được thanh toán.

### **\* Giải thích**

+ Lấy MaHD, MaHopDong, NgayThanhToan, SoTien.

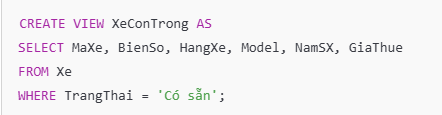
+ Chỉ lấy các hóa đơn có trạng thái "Chưa thanh toán".



1. **XeConTrong (Danh sách xe còn trống để thuê)**

### ****Chức năng:****

Lọc danh sách xe chưa được thuê.



**CHƯƠNG 5: XÂY DỰNG CÁC PROCEDURE**

**5.1. Xây dựng các Stored Procedure (Thủ tục lưu trữ).**

***5.1.1. Tổng quan về nội dung.***

## **\* Khái niệm**

Stored Procedure là một tập hợp câu lệnh SQL được lưu trữ sẵn và có thể thực thi khi cần.

Nó giúp tăng hiệu suất và tự động hóa các thao tác trên cơ sở dữ liệu.

## ****\* Ưu điểm của Stored Procedure****

**+ Tăng hiệu suất và giảm thời gian thực thi:** Stored Procedure được **biên dịch một lần** và có thể được sử dụng nhiều lần, giúp giảm thời gian thực thi so với chạy các truy vấn riêng lẻ.

**+ Giảm thiểu lỗi và tăng tính nhất quán:** Do được lập trình và lưu sẵn trong cơ sở dữ liệu, Stored Procedure giúp **giảm thiểu lỗi do nhập sai SQL**.Khi cần sửa đổi logic, chỉ cần cập nhật Stored Procedure thay vì thay đổi nhiều dòng code trong ứng dụng.

**+ Bảo mật tốt hơn:** Người dùng có thể được cấp quyền **chạy Stored Procedure** mà không cần quyền truy cập trực tiếp vào bảng dữ liệu.Dữ liệu nhạy cảm có thể được xử lý trong Stored Procedure mà không cần hiển thị toàn bộ bảng cho người dùng.

**+ Hỗ trợ tham số linh hoạt:** Stored Procedure có thể nhận tham số đầu vào (IN), đầu ra (OUT), hoặc cả hai (INOUT), giúp linh hoạt hơn trong xử lý dữ liệu.

**+ Tối ưu hóa xử lý logic trong cơ sở dữ liệu:** Giúp **giảm tải công việc cho ứng dụng**, vì nhiều xử lý dữ liệu có thể được thực hiện trực tiếp trên server SQL thay vì trên ứng dụng.

## ****\* Nhược điểm của Stored Procedure****

**+ Khó bảo trì và debug:** Khi Stored Procedure trở nên quá phức tạp, việc bảo trì, debug và sửa lỗi có thể khó khăn hơn so với code ứng dụng thông thường.Các công cụ debug SQL không mạnh mẽ như trong các ngôn ngữ lập trình như Python, Java,...

**+ Không hỗ trợ lập trình hướng đối tượng (OOP):** Không thể sử dụng các tính năng như kế thừa (inheritance) hay đa hình (polymorphism) như trong các ngôn ngữ lập trình thông thường.

**+ Không dễ di chuyển giữa các hệ quản trị cơ sở dữ liệu khác nhau:** Stored Procedure có cú pháp và cách triển khai khác nhau trên các hệ quản trị như SQL Server, MySQL, Oracle, PostgreSQL, gây khó khăn khi chuyển đổi hệ thống.

**+ Tiêu tốn tài nguyên nếu không được tối ưu hóa tốt:** Nếu không được thiết kế hiệu quả, Stored Procedure có thể **gây tải nặng cho database server**, đặc biệt khi xử lý khối lượng dữ liệu lớn.

## **\* Đặc điểm**

**+ Tăng tốc độ thực thi**: Được biên dịch một lần và có thể gọi nhiều lần.

**+ Hỗ trợ tham số đầu vào (**IN**), đầu ra (**OUT**), và cả hai (**INOUT**)**.

**+ Giảm tải cho ứng dụng**, giúp xử lý dữ liệu trực tiếp trong database thay vì trên ứng dụng.

### ****\* Ứng dụng của Stored Procedure****

**+ Tự động hóa các thao tác CRUD** (Create, Read, Update, Delete).

**+ Giảm thiểu lỗi do nhập sai SQL** bằng cách đóng gói logic xử lý.

**+ Cải thiện bảo mật dữ liệu** bằng cách kiểm soát quyền truy cập vào dữ liệu nhạy cảm.

***5.2.1. Chi tiết về Code***

### 1. sp\_ThemKhachHang ****- Thêm khách hàng mới****

#### \* Chức năng:

Thêm một khách hàng mới vào bảng KhachHang, nhưng chỉ khi khách hàng chưa tồn tại dựa trên CMND hoặc Email.

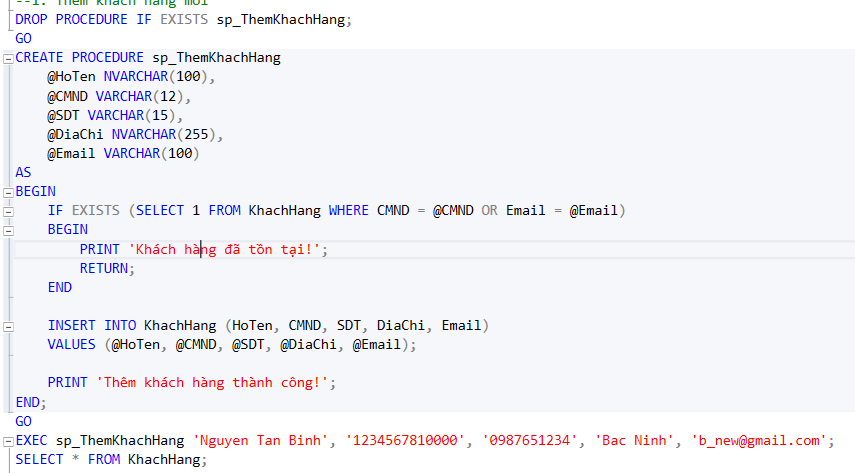
#### \* Giải thích từng bước:

Kiểm tra xem khách hàng đã tồn tại chưa (CMND hoặc Email trùng).

Nếu khách hàng đã tồn tại → In thông báo 'Khách hàng đã tồn tại!' và kết thúc.

Nếu chưa tồn tại → Chèn dữ liệu khách hàng vào bảng KhachHang.

In thông báo 'Thêm khách hàng thành công!'.



### 2. DeleteCustomerByPhone ****- Xóa khách hàng theo số điện thoại****

#### \* Chức năng:

Xóa một khách hàng khỏi bảng KhachHang dựa trên số điện thoại, nhưng chỉ khi khách hàng không có hợp đồng thuê xe.

#### \* Giải thích từng bước:

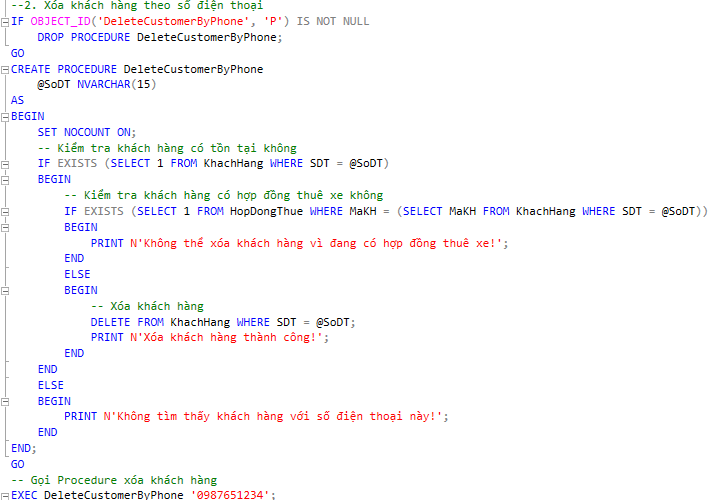
Kiểm tra xem khách hàng có tồn tại không (SDT trùng).

Nếu không tồn tại → In thông báo 'Không tìm thấy khách hàng với số điện thoại này!'.

Nếu tồn tại, kiểm tra xem khách hàng có hợp đồng thuê xe không:

Nếu có → In thông báo 'Không thể xóa khách hàng vì đang có hợp đồng thuê xe!'.

Nếu không có → Xóa khách hàng và in thông báo 'Xóa khách hàng thành công!'.



### 3. FindCustomerByPhone ****- Tìm kiếm khách hàng theo số điện thoại****

#### \* Chức năng:

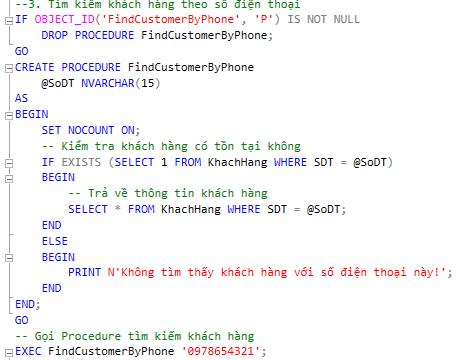
Trả về thông tin của khách hàng dựa trên số điện thoại.

#### \* Giải thích từng bước:

Kiểm tra xem khách hàng có tồn tại không (SDT trùng).

Nếu có → Trả về thông tin khách hàng.

Nếu không → In thông báo 'Không tìm thấy khách hàng với số điện thoại này!'.



### 4. AddCar ****- Thêm xe mới****

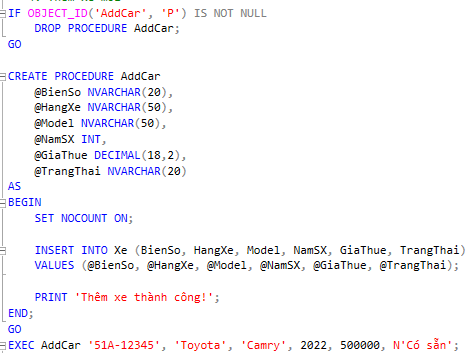
#### \* Chức năng:

Thêm một chiếc xe mới vào bảng Xe.

#### \* Giải thích từng bước:

Thêm xe mới với các thông tin (BienSo, HangXe, Model, NamSX, GiaThue, TrangThai).

In thông báo 'Thêm xe thành công!'



### 5. DeleteCarByBienSo ****- Xóa xe theo biển số****

#### \* Chức năng:

Xóa xe khỏi bảng Xe, nhưng chỉ khi xe không có hợp đồng thuê.

#### \* Giải thích từng bước:

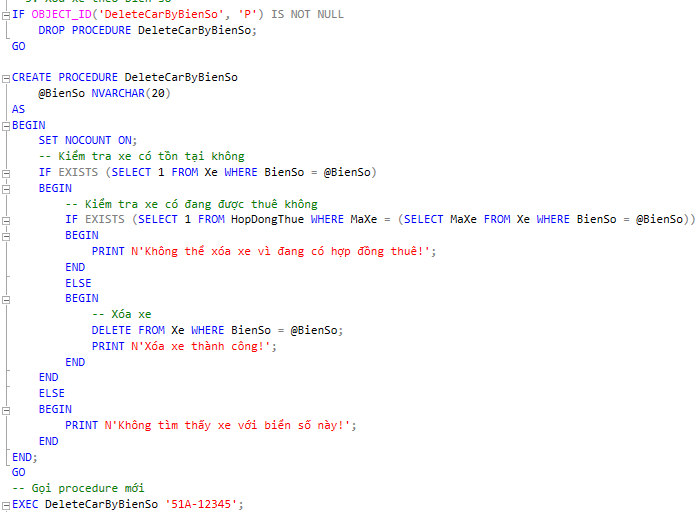
Kiểm tra xem xe có tồn tại không (BienSo trùng).

Nếu không tồn tại → In thông báo 'Không tìm thấy xe với biển số này!'.

Nếu tồn tại, kiểm tra xem xe có hợp đồng thuê không:

Nếu có → In thông báo 'Không thể xóa xe vì đang có hợp đồng thuê!'.

Nếu không có → Xóa xe và in thông báo 'Xóa xe thành công!'.



### 6. FindCarByBienSo ****- Tìm kiếm xe theo biển số****

#### \* Chức năng:

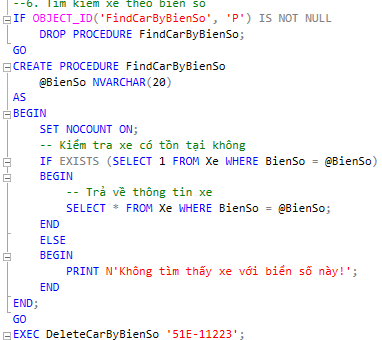
Trả về thông tin của xe dựa trên biển số.

#### \* Giải thích từng bước:

Kiểm tra xem xe có tồn tại không (BienSo trùng).

Nếu có → Trả về thông tin xe.

Nếu không → In thông báo 'Không tìm thấy xe với biển số này!'.



### 7. ThongKeXeThueTheoThang ****- Thống kê số lượng xe thuê theo tháng****

#### \* Chức năng:

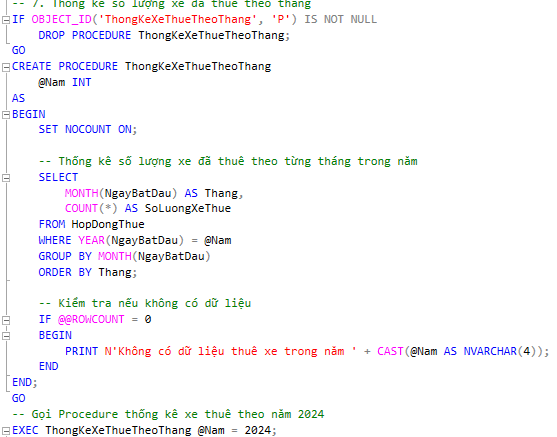
Thống kê số lượng xe được thuê trong từng tháng của một năm cụ thể.

#### \* Giải thích từng bước:

Lọc các hợp đồng thuê trong năm được cung cấp (@Nam).

Đếm số lượng hợp đồng thuê theo tháng (MONTH(NgayBatDau)).

Nếu không có dữ liệu → In thông báo 'Không có dữ liệu thuê xe trong năm'.



### 8. sp\_TimKiemXe ****- Tìm kiếm xe theo trạng thái****

#### \* Chức năng:

Trả về danh sách xe dựa trên trạng thái (Có sẵn, Đang thuê...).

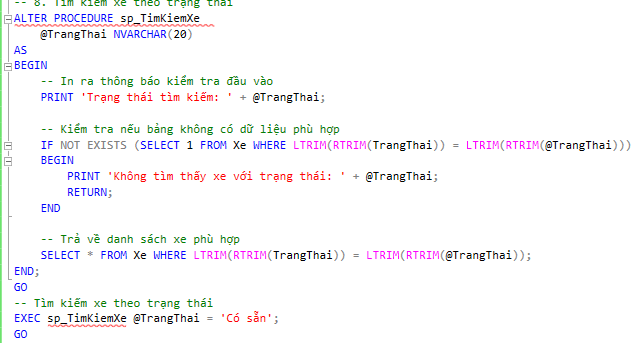
#### \* Giải thích từng bước:

Chuẩn hóa đầu vào bằng LTRIM(RTRIM(@TrangThai)) để loại bỏ khoảng trắng.

Kiểm tra xem có xe nào phù hợp không.

Nếu có → Trả về danh sách xe.

Nếu không → In thông báo 'Không tìm thấy xe với trạng thái này'.



### 9. sp\_CapNhatTrangThaiXe ****- Cập nhật trạng thái xe sau khi trả****

#### \* Chức năng:

Cập nhật trạng thái xe về 'Có sẵn' khi xe được trả.

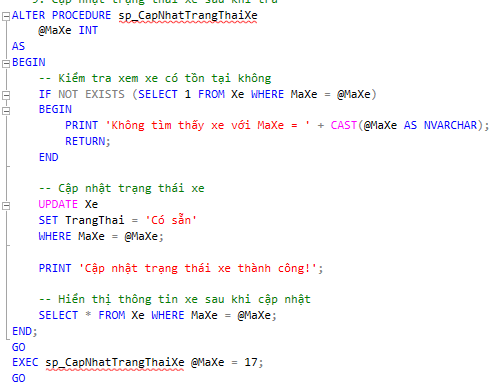
\* Giải thích từng bước:

Kiểm tra xem xe có tồn tại không (MaXe trùng).

Nếu không tồn tại → In thông báo 'Không tìm thấy xe với MaXe này'.

Nếu có → Cập nhật trạng thái thành 'Có sẵn'.

Trả về thông tin xe sau khi cập nhật.



### 10. sp\_ThongKeXe ****- Thống kê số lượng xe theo trạng thái****

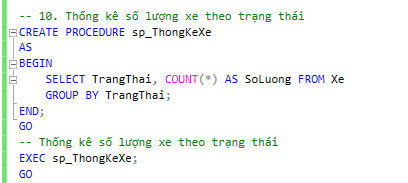
#### \* Chức năng:

Thống kê số lượng xe theo từng trạng thái (Có sẵn, Đang thuê...).

#### \* Giải thích từng bước:

Đếm số lượng xe theo từng giá trị TrangThai.

Trả về kết quả dưới dạng bảng.



**CHƯƠNG 6: XÂY DỰNG CÁC TRIGGER**

**6.1. Xây dựng các Trigger (Bộ kích hoạt)**

***6.1.1. Tổng quan về nội dung.***

## ****\* Khái niệm****

Trigger là một tập hợp các câu lệnh SQL tự động thực thi khi có sự kiện như INSERT, UPDATE, hoặc DELETE.

## ****\* Ưu điểm của Trigger****

**+ Tự động hóa quy trình kiểm tra và cập nhật dữ liệu:** Trigger giúp **đảm bảo dữ liệu luôn tuân theo các ràng buộc và quy tắc nghiệp vụ** mà không cần lập trình thủ công trong ứng dụng.

**+ Cải thiện bảo mật và tính toàn vẹn dữ liệu:** Trigger có thể **ngăn chặn các thao tác không hợp lệ**, ví dụ: không cho phép xóa sản phẩm nếu nó còn tồn tại trong hóa đơn.

**+ Ghi log lịch sử thay đổi dữ liệu:** Trigger giúp ghi lại **lịch sử cập nhật, xóa dữ liệu**, hỗ trợ kiểm tra và khôi phục khi cần thiết.

**+ Hỗ trợ đồng bộ hóa dữ liệu giữa các bảng:** Khi một bảng thay đổi, Trigger có thể **cập nhật dữ liệu liên quan trong các bảng khác** một cách tự động.

## ****\* Nhược điểm của Trigger****

## + **Khó kiểm soát và debug:** Vì Trigger chạy tự động trong nền khi có sự kiện, nên **khó theo dõi và sửa lỗi** khi có vấn đề xảy ra.Không thể gọi Trigger trực tiếp để kiểm tra như Stored Procedure.

**+ Gây ảnh hưởng đến hiệu suất:** Nếu Trigger chứa quá nhiều logic phức tạp, nó có thể **gây chậm hệ thống** vì mỗi khi một thao tác (INSERT, UPDATE, DELETE) xảy ra, Trigger sẽ tự động thực thi.

**+ Có thể gây vòng lặp Trigger (Trigger Recursion):** Nếu Trigger cập nhật dữ liệu trên một bảng mà chính bảng đó có Trigger khác, nó có thể dẫn đến vòng lặp vô hạn.

**+ Khó di chuyển giữa các hệ quản trị cơ sở dữ liệu khác nhau:** Cú pháp và cách triển khai Trigger khác nhau giữa SQL Server, MySQL, PostgreSQL, Oracle, gây khó khăn khi chuyển đổi hệ thống.

## ****\* Đặc điểm****

**+ Tự động thực thi khi có thay đổi trên bảng dữ liệu**.

**+ Không thể gọi trực tiếp như Stored Procedure**.

**+ Có thể thực thi trước (**BEFORE**) hoặc sau (**AFTER**) khi một thao tác xảy ra**.

## ****\* Ứng dụng của Trigger****

**+ Tự động kiểm tra và bảo vệ tính toàn vẹn dữ liệu**.

**+ Ghi lại lịch sử thay đổi dữ liệu** để kiểm soát hoạt động của hệ thống.

**+ Đồng bộ hóa dữ liệu giữa các bảng**.

***6.2.1. Chi tiết Code***

## **1. **Trigger:** trg\_Check\_NamSX **- Kiểm tra tuổi của xe khi thêm vào hệ thống****

### **\* Mục đích:**

Đảm bảo rằng năm sản xuất (NamSX) của xe hợp lệ khi thêm vào bảng Xe.

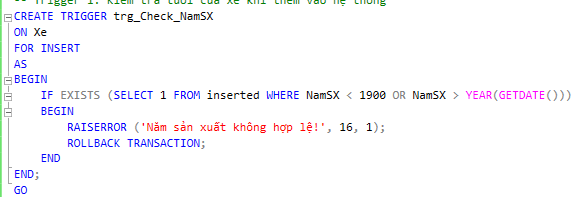
### **\* Cách hoạt động:**

Khi có một bản ghi mới được chèn (INSERT) vào bảng Xe, trigger này sẽ kiểm tra:

Nếu NamSX < 1900 hoặc NamSX > Năm hiện tại (YEAR(GETDATE())) thì:

Gây lỗi 'Năm sản xuất không hợp lệ!'.

**Hủy giao dịch (**ROLLBACK TRANSACTION**)**, không cho phép chèn dữ liệu.



## **2.**Trigger:** trg\_Update\_TrangThai\_Xe **- Cập nhật trạng thái xe khi có hợp đồng thuê****

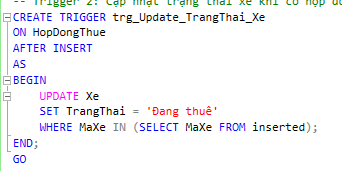
### **\* Mục đích:**

Tự động cập nhật trạng thái của xe thành **"Đang thuê"** khi có hợp đồng thuê mới.

### **\* Cách hoạt động:**

Khi có **hợp đồng thuê mới (**INSERT**)** vào bảng HopDongThue, trigger này sẽ:

Cập nhật trạng thái xe (Xe.TrangThai) thành **'Đang thuê'** nếu MaXe xuất hiện trong hợp đồng mới.



## **3.**Trigger:** trg\_Check\_Duplicate\_Rental **- Không cho phép thuê cùng một xe hai lần chưa trả****

### **\* Mục đích:**

Không cho phép một khách hàng thuê lại **cùng một xe** khi **hợp đồng cũ chưa kết thúc**.

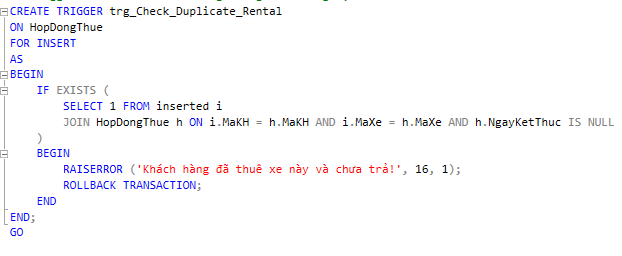
### **\* Cách hoạt động:**

Khi chèn một hợp đồng mới (INSERT vào HopDongThue), trigger sẽ kiểm tra:

Nếu khách hàng (MaKH) đã có một hợp đồng thuê cùng một xe (MaXe) mà chưa trả (NgayKetThuc IS NULL), thì:

Gây lỗi 'Khách hàng đã thuê xe này và chưa trả!'.

**Hủy giao dịch (**ROLLBACK TRANSACTION**)**.



## **4.**Trigger:** trg\_Return\_Xe **- Cập nhật trạng thái xe khi xe được trả****

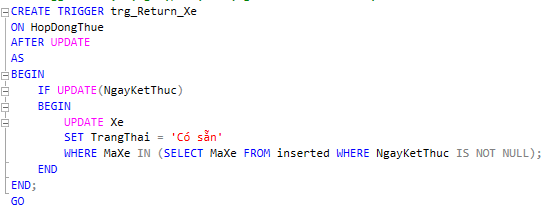
### **\* Mục đích:**

Tự động đổi trạng thái xe về **'Có sẵn'** khi hợp đồng thuê xe kết thúc.

### **\* Cách hoạt động:**

Khi một hợp đồng được **cập nhật (**UPDATE**)**, nếu cột NgayKetThuc thay đổi từ NULL → có giá trị, trigger sẽ:

Cập nhật trạng thái xe (Xe.TrangThai) thành **'Có sẵn'**.



## **5.**Trigger:** trg\_Check\_SDT\_KhachHang **- Kiểm tra số điện thoại khách hàng****

### **\* Mục đích:**

Đảm bảo số điện thoại (SDT) chỉ chứa số và không bị nhập sai định dạng.

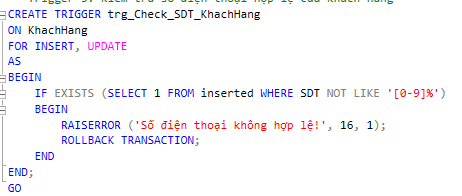
### **\* Cách hoạt động:**

Khi chèn (INSERT) hoặc cập nhật (UPDATE) khách hàng (KhachHang), trigger kiểm tra:

Nếu SDT không bắt đầu bằng số ('[0-9]%'), thì:

Gây lỗi 'Số điện thoại không hợp lệ!'.

**Hủy giao dịch (**ROLLBACK TRANSACTION**)**.



## **6.**Trigger:** trg\_Check\_Email\_NhanVien **- Kiểm tra định dạng email của nhân viên****

### **\* Mục đích:**

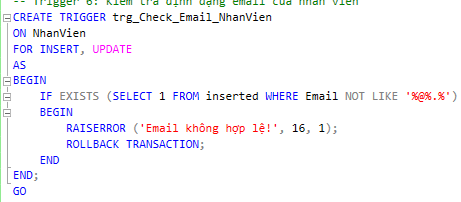
Đảm bảo email có định dạng hợp lệ (...@...).

### **\* Cách hoạt động:**

Nếu Email không chứa '@' và '.', trigger sẽ:

Gây lỗi 'Email không hợp lệ!'.

**Hủy giao dịch (**ROLLBACK TRANSACTION**)**.



## **7.**Trigger:** trg\_Prevent\_Delete\_NhanVien **- Không cho phép xóa nhân viên có hợp đồng chưa kết thúc****

### **\* Mục đích:**

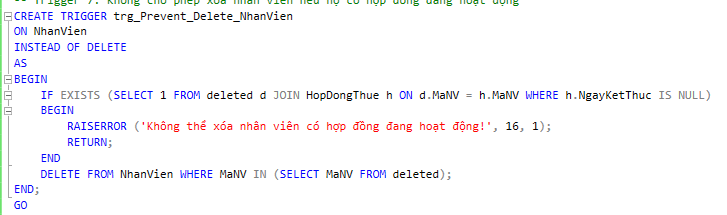
Không cho phép xóa nhân viên nếu họ còn hợp đồng chưa hoàn thành.

### **\* Cách hoạt động:**

Nếu nhân viên (MaNV) có hợp đồng chưa kết thúc (NgayKetThuc IS NULL), trigger sẽ:

Gây lỗi 'Không thể xóa nhân viên có hợp đồng đang hoạt động!'.

**Ngăn chặn thao tác xóa (**INSTEAD OF DELETE**)**.



## **8.**Trigger:** trg\_Prevent\_Delete\_KhachHang **- Không cho phép xóa khách hàng nếu có hợp đồng chưa trả****

### **\* Mục đích:**

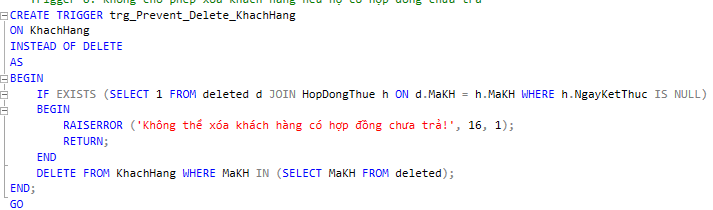
Ngăn xóa khách hàng nếu họ có hợp đồng chưa hoàn tất.

### **\* Cách hoạt động:**

Nếu khách hàng có hợp đồng chưa kết thúc, trigger sẽ:

Gây lỗi 'Không thể xóa khách hàng có hợp đồng chưa trả!'.

**Ngăn chặn thao tác xóa.**



## **9.**Trigger:** trg\_Check\_GiaThue **- Kiểm tra giá thuê xe không âm****

### **\* Mục đích:**

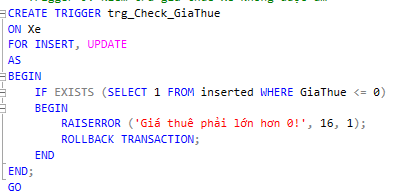
Đảm bảo GiaThue > 0.

### **\* Cách hoạt động:**

Nếu GiaThue <= 0, trigger sẽ:

Gây lỗi 'Giá thuê phải lớn hơn 0!'.

**Hủy giao dịch.**



## **10.**Trigger:** trg\_Log\_New\_HopDong **- Ghi log khi thêm hợp đồng mới****

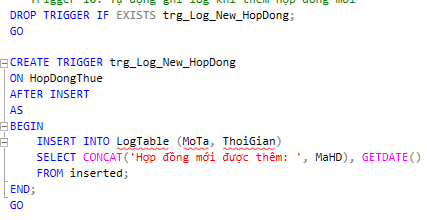
### **\* Mục đích:**

Tự động ghi lại lịch sử thêm hợp đồng vào LogTable.

### **\* Cách hoạt động:**

Khi thêm hợp đồng mới (INSERT), trigger sẽ:

Ghi vào LogTable với thời gian GETDATE().



**CHƯƠNG 7: PHÂN QUYỀN VÀ BẢO VỆ CƠ SỞ DỮ LIỆU**

## **7.1. Phân quyền trong SQL Server**

### *****7.1.1. Mô hình bảo mật trong SQL Server*****

\* SQL Server áp dụng mô hình bảo mật theo nhiều cấp độ:

**+ Server-Level**: Kiểm soát truy cập vào toàn bộ SQL Server.

**+ Database-Level**: Kiểm soát truy cập vào từng cơ sở dữ liệu.

**+ Schema-Level**: Quản lý quyền trong từng schema (bộ sưu tập của bảng, stored procedure, view,...).

**+ Object-Level**: Cấp quyền chi tiết cho từng bảng, cột, view, stored procedure,...

### *****7.1.2. Các loại tài khoản và vai trò*****

#### **\* Tài khoản người dùng trong SQL Server**

**Login**: Tài khoản đăng nhập vào SQL Server (tạo ở cấp độ server).

**User**: Tài khoản của một người dùng trong từng cơ sở dữ liệu cụ thể.

#### **\* Các vai trò (Roles) trong SQL Server**

#### **- Server Roles**:

**+ sysadmin**: Toàn quyền quản trị SQL Server.

**+ dbcreator**: Tạo, thay đổi, xóa cơ sở dữ liệu.

**+ securityadmin**: Quản lý quyền của user và role.

**- Database Roles**:

**+ db\_owner**: Toàn quyền trên cơ sở dữ liệu.

**+ db\_datareader**: Chỉ có quyền đọc dữ liệu.

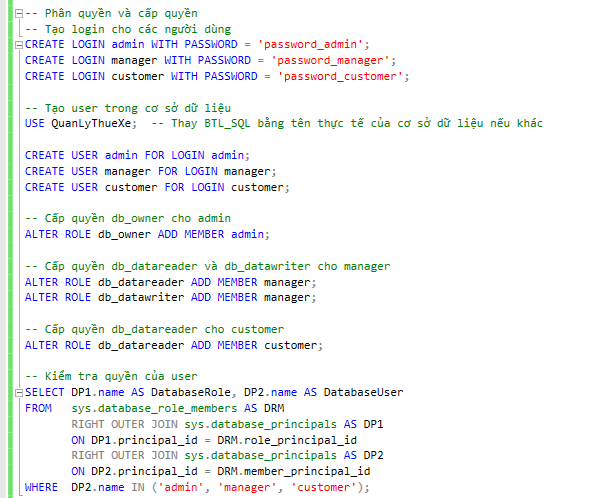
**+ db\_datawriter**: Chỉ có quyền ghi dữ liệu.

**+ db\_securityadmin**: Quản lý quyền truy cập trong database.

### *****7.1.3. Cấp quyền và thu hồi quyền*****

### **Cấp quyền truy cập cho người dùng**

**Tạo tài khoản đăng nhập và gán quyền**



## **7.2. Bảo vệ cơ sở dữ liệu**

### *****7.2.1. Kiểm soát quyền truy cập*****

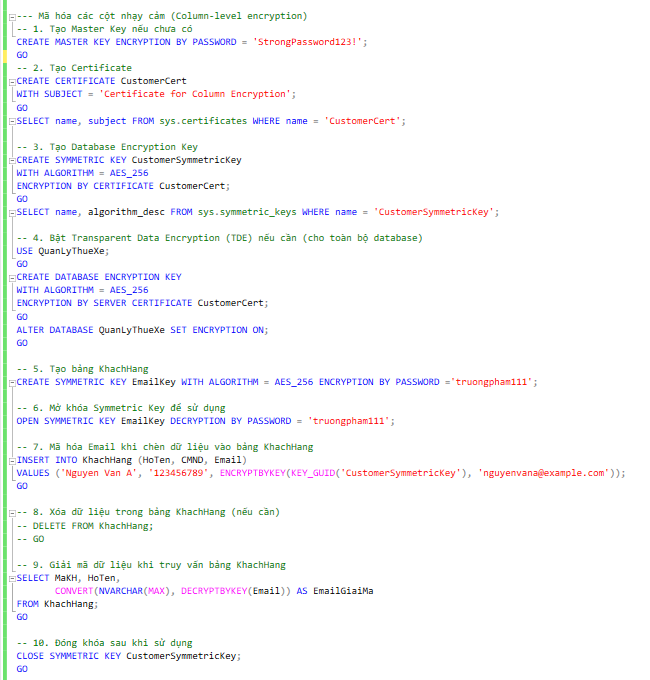
Chỉ cấp quyền cần thiết cho từng tài khoản.

Sử dụng **role-based access control (RBAC)** thay vì cấp quyền trực tiếp cho từng user.

### *****7.2.2. Mã hóa dữ liệu (Encryption)*****

Mã hóa giúp bảo vệ dữ liệu quan trọng khỏi bị lộ hoặc đánh cắp.

#### **\*** Mã hóa dữ liệu ở cấp độ cột (Column-Level Encryption)



**KẾT LUẬN**

Sau quá trình nghiên cứu và triển khai, hệ thống **Quản lý công ty cho thuê xe** đã được thiết kế và xây dựng với đầy đủ các chức năng cốt lõi, bao gồm quản lý khách hàng, xe, hợp đồng, thanh toán, bảo trì và chăm sóc khách hàng. Hệ thống sử dụng **SQL Server** làm nền tảng quản trị cơ sở dữ liệu, đảm bảo tính toàn vẹn dữ liệu, khả năng mở rộng và hiệu suất cao.

### **\* Những kết quả đạt được**

Hệ thống đã hoàn thiện các tính năng chính:

**Quản lý khách hàng**: Lưu trữ thông tin khách hàng, lịch sử thuê xe và giao dịch.

**Quản lý xe**: Theo dõi tình trạng xe (có sẵn, đang thuê, bảo trì), giá thuê và lịch sử sử dụng.

**Quản lý hợp đồng**: Hỗ trợ tạo hợp đồng, theo dõi thời gian thuê, tính toán chi phí và quản lý trạng thái hợp đồng.

**Quản lý thanh toán**: Kiểm soát hóa đơn, xác định trạng thái thanh toán (đã thanh toán, chưa thanh toán).

**Quản lý bảo trì xe**: Lập lịch bảo trì, ghi nhận chi phí và theo dõi lịch sử sửa chữa xe.

**Quản lý chăm sóc khách hàng**: Theo dõi phản hồi của khách hàng, hỗ trợ giải quyết khiếu nại.

\* Bên cạnh đó, hệ thống còn áp dụng các kỹ thuật nâng cao:

**Stored Procedures** giúp tối ưu hóa truy vấn và giảm thời gian xử lý.

**Triggers** đảm bảo tính toàn vẹn dữ liệu và tự động cập nhật trạng thái xe, hợp đồng.

**Views** giúp đơn giản hóa truy vấn dữ liệu, nâng cao bảo mật và tối ưu hiệu suất.

**Phân quyền người dùng** bảo vệ dữ liệu, chỉ cho phép các nhóm người dùng truy cập thông tin phù hợp.

### **\* Hạn chế và khó khăn**

Mặc dù hệ thống đã đáp ứng được các yêu cầu cơ bản, vẫn còn một số hạn chế cần khắc phục:

**Chưa có giao diện trực quan**: Hệ thống mới chỉ tập trung vào cơ sở dữ liệu, chưa tích hợp với giao diện người dùng trên web hoặc ứng dụng di động.

**Chưa tối ưu hiệu suất cho dữ liệu lớn**: Khi số lượng xe, khách hàng và hợp đồng tăng cao, cần tối ưu truy vấn SQL và hạ tầng cơ sở dữ liệu để đảm bảo tốc độ xử lý.

**Chưa có tính năng dự báo**: Hiện tại, hệ thống chưa tích hợp các công cụ phân tích để dự báo nhu cầu thuê xe theo mùa, khu vực hoặc xu hướng thị trường.