

**TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỐ HỒ
CHÍ MINH**

KHOA CÔNG NGHỆ THÔNG TIN



**BÀI BÁO CÁO KẾT THÚC HỌC PHẦN
CƠ SỞ DỮ LIỆU NÂNG CAO**

Phân tích, thiết kế và cài đặt CSDL (Quản lý GARA Ô TÔ)

Giảng viên hướng dẫn: Lê Thị Minh Nguyễn

1. Phạm Duy Hiếu 22DH111081

Mục lục

Mục lục	1
Chương I. Giới thiệu đề tài	6
1.1. Giới thiệu	6
1.1.1. Mở đầu	6
1.1.2. Lý do	8
1.2. Khảo sát thực tế	9
1.2.1. Thông tin chung	9
1.2.2. Cơ cấu tổ chức	10
1.3. Mô tả yêu cầu lưu trữ:.....	10
1.3.1. Quản lý khách hàng	Error! Bookmark not defined.
1.3.2. Quản lý phương tiện	Error! Bookmark not defined.
1.3.3. Quản lý kho dữ trữ	Error! Bookmark not defined.
1.3.4. Quản lý dịch vụ sửa chữa	Error! Bookmark not defined.
1.3.5. Quản lý tài chính.....	Error! Bookmark not defined.
1.3.6. Yêu cầu chung	Error! Bookmark not defined.
1.4. Các yêu cầu khác.....	Error! Bookmark not defined.
Chương II: Cơ sở lý thuyết	13
2.1. Mô hình thực thể kết hợp (ERD)	13
2.1.1. Các thành phần chính trong ERD:	13
2.1.2. Biểu diễn đồ thị:.....	13
2.1.3. Các mối quan hệ:	14
2.1.4. Ưu điểm:	14

2.1.5. Hạn chế:	14
2.2. Phụ thuộc hàm.....	15
2.2.1. Khái niệm.....	15
2.2.2. Các loại phụ thuộc hàm:	15
2.2.3. Quy tắc:.....	15
2.2.4. Ứng dụng:	16
2.2.5. Hạn chế:	16
2.3. Các dạng chuẩn	16
2.3.1. Dạng chuẩn 1NF:	16
2.3.2. Dạng chuẩn 2NF:	17
2.3.3. Dạng chuẩn 3NF:	17
2.3.4. Dạng chuẩn boyce-codd (BCNF):	18
2.3.5. Dạng chuẩn 4NF:	18
2.3.6. Ưu điểm của chuẩn hóa:	18
2.4. Bảo toàn thông tin.....	19
2.4.1. Các thuộc tính bảo mật:	19
2.4.2. Các mối đe dọa:	19
2.4.3. Các kỹ thuật bảo mật:	19
2.4.4. Các tiêu chuẩn bảo mật:.....	20
2.5. Lập trình cơ sở dữ liệu	20
2.5.1. Các khái niệm cơ bản:.....	20
2.5.2. Các kỹ thuật lập trình cơ sở dữ liệu:	21
2.5.3. Các kỹ thuật nâng cao:.....	21

2.5.4. Ứng dụng:	22
2.6. Ràng buộc toàn vẹn cơ sở dữ liệu	22
2.6.1. Ràng buộc toàn vẹn cột:.....	22
2.6.2. Ràng buộc toàn vẹn bảng:.....	23
2.6.3. Ràng buộc toàn vẹn cơ sở dữ liệu:.....	23
2.6.4. Ưu điểm:	23
2.7. noSQL và newSQL	24
2.7.1. noSQL:.....	24
2.7.2. newSQL:	25
Chương III. Phân tích và thiết kế cơ sở dữ liệu	28
3.1. Mô hình thực thể kết hợp:.....	28
3.2. Mô tả thực thể	29
3.2.1. Thực thể: khách hàng.....	29
3.2.2. Thực thể: Xe.....	29
3.2.3. Thực thể: Loại khách hàng	29
3.2.4. Thực thể : Loại xe	29
3.2.5. Thực thể: Phiếu dịch vụ	29
3.2.6. Thực thể: Dịch vụ	30
3.2.7. Thực thể: Loại dịch vụ	30
3.2.8. Thực thể: phụ tùng	30
3.2.9. Thực thể: Loại phụ tùng.....	30
3.2.10. Thực thể: Kho	30
3.2.11. Thực thể: Phiếu nhập kho	31

3.2.12. Thực thể: Nhân viên.....	31
3.2.13. Thực thể: hóa đơn	31
3.3. Lược đồ cơ sở dữ liệu quan hệ.....	31
3.3.1. Chuyển thực thể mạnh và các thuộc tính đơn:.....	31
3.3.2. Chuyển các thuộc tính kết hợp.....	32
3.3.3. Chuyển các thuộc tính đa trị	32
3.3.4. Chuyển quan hệ (1 – n).....	32
3.3.5. Chuyển quan hệ n-n	33
3.3.6. kết quả.....	33
3.4. Phân tích chuẩn của lược đồ quan hệ.....	34
3.4.1. Lược đồ quan hệ KháchHang	34
3.4.2. Lược đồ quan hệ Xe.....	35
3.4.3. Lược đồ quan hệ DịchVụ	35
3.4.4. Lược đồ quan hệ PhụTung	36
3.5. Lược đồ quan hệ sau khi chuẩn hóa.....	37
3.6. Các ràng buộc toàn vẹn.....	37
3.6.1. Khách hàng, Nhân Viên	37
3.6.2. Xe	38
3.6.3. Dịch vụ.....	39
3.6.4. Phụ tùng	39
3.6.5. Phiếu dịch vụ	40
3.6.6. Phiếu nhập kho.....	40
3.6.7. Hóa đơn.....	41

3.6.8. Chi tiết phiếu nhập kho	41
--------------------------------------	----

Chương I. Giới thiệu đề tài

1.1. Giới thiệu

1.1.1. Mở đầu

- Mở đầu:

Trong bối cảnh nền kinh tế phát triển nhanh chóng và nhu cầu về phương tiện giao thông ngày càng tăng, các gara ô tô đóng vai trò quan trọng trong việc bảo trì và sửa chữa phương tiện. Tuy nhiên, việc quản lý các hoạt động này, từ theo dõi lịch bảo dưỡng, quản lý khách hàng, quản lý kho phụ tùng đến xử lý hóa đơn và thanh toán, đòi hỏi một hệ thống quản lý cơ sở dữ liệu (CSDL) hiện đại và hiệu quả.

- Giới thiệu tổng quan về nghiệp vụ doanh nghiệp:

Các gara ô tô hoạt động trong lĩnh vực dịch vụ bảo trì và sửa chữa ô tô. Nhiệm vụ chính của các gara này là đảm bảo các phương tiện giao thông hoạt động hiệu quả, an toàn và bền bỉ. Các nghiệp vụ cơ bản của gara ô tô bao gồm:

- + Bảo dưỡng định kỳ: Kiểm tra và thay thế các bộ phận, dung dịch cần thiết theo chu kỳ nhất định.

- + Sửa chữa: Khắc phục các hỏng hóc hoặc sự cố phát sinh trong quá trình sử dụng xe.

- + Nâng cấp: Cải thiện các tính năng của xe theo yêu cầu của khách hàng.

- + Quản lý kho phụ tùng: Đảm bảo kho phụ tùng luôn sẵn sàng đáp ứng nhu cầu sửa chữa và bảo dưỡng.

- + Quản lý khách hàng: Lưu trữ thông tin, theo dõi lịch sử dịch vụ và tương tác với khách hàng.

- + Quản lý tài chính: Xử lý hóa đơn, thanh toán, và các hoạt động tài chính khác liên quan.

- Lĩnh vực hoạt động:

Gara ô tô hoạt động chủ yếu trong lĩnh vực dịch vụ bảo trì và sửa chữa ô tô, một ngành công nghiệp đòi hỏi kiến thức chuyên môn cao và khả năng đáp ứng nhanh chóng các yêu cầu của khách hàng. Gara thường hợp tác với các nhà cung cấp phụ tùng, bảo hiểm, và các dịch vụ liên quan để mang lại dịch vụ toàn diện cho khách hàng.
- Quy trình nghiệp vụ liên quan đến đề tài:
 1. Tiếp nhận xe và yêu cầu dịch vụ
 - + Nhân viên tiếp nhận thông tin từ khách hàng về các vấn đề của xe.
 - + Lập phiếu tiếp nhận và nhập thông tin xe, yêu cầu dịch vụ vào hệ thống.
 2. Chuẩn đoán và kiểm tra
 - + Kỹ thuật viên tiến hành kiểm tra xe theo yêu cầu dịch vụ.
 - + Ghi nhận các vấn đề phát hiện được và báo cáo cho khách hàng.
 3. Lập kế hoạch sửa chữa và bảo dưỡng
 - + Đề xuất các giải pháp sửa chữa và bảo dưỡng.
 - + Lập kế hoạch thời gian và chi phí dự kiến, thông báo cho khách hàng.
 4. Thực hiện dịch vụ
 - + Kỹ thuật viên tiến hành các công việc sửa chữa và bảo dưỡng theo kế hoạch.
 - + Sử dụng phụ tùng và vật liệu từ kho để thực hiện công việc.
 5. Kiểm tra và giao xe
 - + Kiểm tra lại toàn bộ các công việc đã thực hiện.
 - + Lập báo cáo hoàn thành, thông báo cho khách hàng đến nhận xe.
 - + Xử lý các thủ tục thanh toán và ghi nhận thông tin vào hệ thống.

6. Quản lý dữ liệu

- + Lưu trữ thông tin khách hàng, xe, lịch sử dịch vụ.
 - + Theo dõi và quản lý kho phụ tùng, cập nhật số lượng và tình trạng phụ tùng.
 - + Quản lý tài chính, hóa đơn, và các báo cáo liên quan.
- Vấn đề liên quan đến dữ liệu:
- + Lưu trữ và truy xuất dữ liệu: Cần một hệ thống CSDL hiệu quả để lưu trữ và truy xuất thông tin nhanh chóng, chính xác.
 - + Tính nhất quán và toàn vẹn dữ liệu: Đảm bảo dữ liệu không bị sai sót hoặc mất mát trong quá trình nhập liệu và xử lý.
 - + Bảo mật dữ liệu: Bảo vệ dữ liệu khách hàng và thông tin kinh doanh khỏi các mối đe dọa bảo mật.
 - + Khả năng mở rộng: Hệ thống CSDL cần có khả năng mở rộng để đáp ứng nhu cầu phát triển của doanh nghiệp.
 - + Tích hợp dữ liệu: Dữ liệu cần được tích hợp và đồng bộ hóa giữa các bộ phận khác nhau để đảm bảo sự liên mạch trong hoạt động.

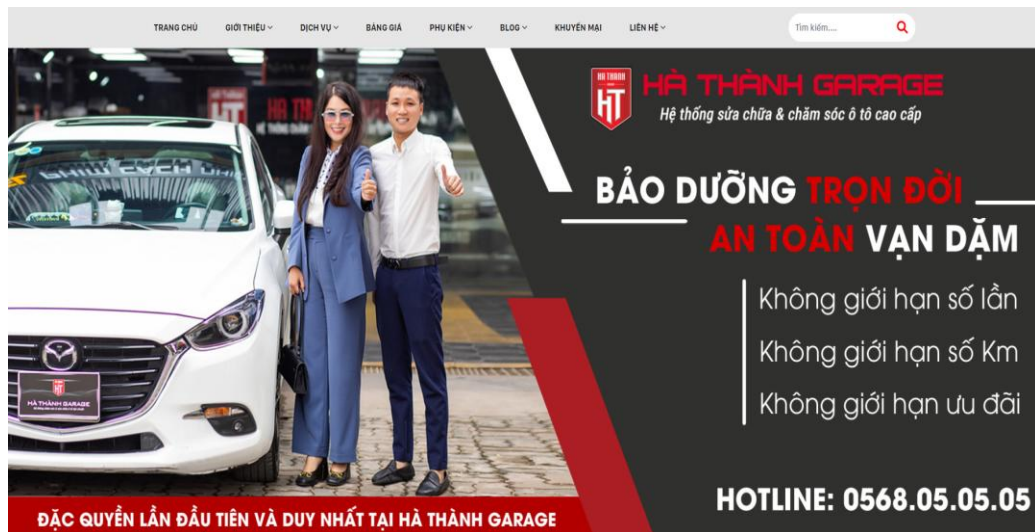
1.1.2. Lý do

- Quản lý thủ công hoặc sử dụng các phương pháp truyền thống đã bộc lộ nhiều hạn chế như:
 - + Khó khăn trong việc theo dõi và lưu trữ thông tin khách hàng, xe và dịch vụ.
 - + Tốn nhiều thời gian và công sức trong việc tra cứu và cập nhật thông tin.
 - + Nguy cơ mất mát dữ liệu cao và khó kiểm soát chất lượng dịch vụ.

1.2. Khảo sát thực tế

1.2.1. Thông tin chung

- Giao diện bảo dưỡng ô tô:



- Thông tin lưu trữ:

+ Hotline liên lạc

+ Dịch vụ bảo dưỡng không giới hạn số lần, số Km và có nhiều ưu đãi hấp dẫn

+ Chăm sóc và làm đẹp ô tô

+ Cứu hộ ô tô

+ Bảo dưỡng trọn đời, an toàn vạn dặm

+ Sơn và phục hồi thân xe

1.2.2. Cơ cấu tổ chức



1.3. Mô tả yêu cầu lưu trữ:

1.3.1. Bảng Lưu trữ của khách hàng:

Stt	Dữ liệu	Kiểu	Hình thức nhập	Ràng buộc
1	MaKH	Text	Nhập từ bàn phím	Không trùng nhau
2	TenKH	Text	Nhập từ bàn phím	
3	SoDT	Text	Nhập từ bàn phím	
4	Phai	Text	Nhập từ bàn phím	
5	NgaySinh	Date	Nhập từ bàn phím	
6	DiaChi	Text	Nhập từ bàn phím	
7	Email	Text	Nhập từ bàn phím	

1.3.2. Bảng lưu trữ của Xe:

Stt	Dữ liệu	Kiểu	Hình thức nhập	Ràng buộc
1	MaXe	Text	Nhập từ bàn phím	Không trùng nhau
2	BienSoXe	Number	Nhập từ bàn phím	
3	NamSX	Number	Nhập từ bàn phím	
4	HangXe	Text	Nhập từ bàn phím	

1.3.3. Bảng lưu trữ của loại khách hàng

Stt	Dữ liệu	Kiểu	Hình thức nhập	Ràng buộc
1	MaLoaiKH	Text	Nhập từ bàn phím	Không trùng nhau
2	TenLoaiKH	Text	Nhập từ bàn phím	

1.3.4. Bảng lưu trữ của Loại xe

Stt	Dữ liệu	Kiểu	Hình thức nhập	Ràng buộc
1	MaLoaiXe	Text	Nhập từ bàn phím	Không trùng nhau
2	TenLoaiXe	Text	Nhập từ bàn phím	

1.3.5. Bảng lưu trữ của Phiếu dịch vụ

Stt	Dữ liệu	Kiểu	Hình thức nhập	Ràng buộc
1	MaPhieuDV	Text	Nhập từ bàn phím	Không trùng nhau
2	NgayLapPhieu	Date	Nhập từ bàn phím	
3	TongTienDV	Number	Bảng tổng tiền trong chi tiết dịch vụ	Tổng tiền tính được trong chi tiết dịch vụ
4	TongTienPT	Number	Bảng tổng tiền trong chi tiết phụ tung	Tổng tiền tính được trong chi tiết phụ tung
5	TongTien	Date	Bảng tổng tiền chi tiết + tổng tiền dịch vụ	

1.3.6. Bảng lưu trữ của dịch vụ

Stt	Dữ liệu	Kiểu	Hình thức nhập	Ràng buộc
1	MaDV	Text	Nhập từ bàn phím	Không trùng nhau
2	TenDV	Date	Nhập từ bàn phím	
3	GiaDV	Number	Nhập từ bàn phím	

1.3.7. Bảng lưu trữ của Loại dịch vụ

Stt	Dữ liệu	Kiểu	Hình thức nhập	Ràng buộc
1	MaLoaiDV	Text	Nhập từ bàn phím	Không trùng nhau
2	TenLoaiDV	Text	Nhập từ bàn phím	

1.3.8. Bảng lưu trữ của Phụ tùng

Stt	Dữ liệu	Kiểu	Hình thức nhập	Ràng buộc
1	MaPT	Text	Nhập từ bàn phím	Không trùng nhau
2	TenPT	Text	Nhập từ bàn phím	
3	GiaPT	Number	Nhập từ bàn phím	
4	SoLuongTK	Number	Nhập từ bàn phím	
5	DonViTinh	Date	Nhập từ bàn phím	

1.3.9. Bảng lưu trữ của loại phụ tùng:

Stt	Dữ liệu	Kiểu	Hình thức nhập	Ràng buộc
1	MaLoaiPT	Text	Nhập từ bàn phím	Không trùng nhau
2	TenLoaiPT	Text	Nhập từ bàn phím	

1.3.10. Bảng lưu trữ của kho:

Stt	Dữ liệu	Kiểu	Hình thức nhập	Ràng buộc
1	MaKho	Text	Nhập từ bàn phím	Không trùng nhau
2	TenKho	Text	Nhập từ bàn phím	

Chương II: Cơ sở lý thuyết

2.1. Mô hình thực thể kết hợp (ERD)

- Mô hình thực thể kết hợp (ERD) là một sơ đồ biểu diễn dữ liệu trong cơ sở dữ liệu quan hệ. Nó sử dụng các biểu tượng đồ họa để thể hiện các thực thể (thực thể là đối tượng cần lưu trữ dữ liệu) và mối quan hệ giữa chúng. ERD giúp:

- + ERD cung cấp một cách trực quan để nhìn thấy các thực thể và cách chúng liên kết với nhau.
- + ERD giúp thiết kế cơ sở dữ liệu hiệu quả, đảm bảo dữ liệu được tổ chức hợp lý và dễ dàng truy cập.
- + ERD giúp các nhà phát triển phần mềm và người dùng cuối cùng hiểu rõ cách dữ liệu được tổ chức và sử dụng.

2.1.1. Các thành phần chính trong ERD:

- Thực thể (Entity): Đối tượng cần lưu trữ dữ liệu, ví dụ: Khách hàng, Xe, Kho,...
- Thuộc tính (Attribute): Đặc điểm của 1 thực thể, ví dụ: Biển số, mã xe,...
- Mối quan hệ (Relationship): Mối liên kết giữa các thực thể, ví dụ: Khách hàng “sở hữu” xe, Dịch vụ “bao gồm” phụ tùng,...

2.1.2. Biểu diễn đồ thị:

- Hình chữ nhật: Biểu diễn thực thể (Entity).
- Hình tròn: Biểu diễn thuộc tính (Attribute).
- Hình thoi: Biểu diễn mối quan hệ (Relationship).
- Đường nối: Kết nối thực thể với mối quan hệ (Connect).

2.1.3. Các mối quan hệ:

- Một-một (1-1): Một thực thể liên kết với duy nhất một thực thể khác. Ví dụ: Một người chỉ có thể có một chứng minh thư.
- Một-nhiều (1-n): Một thực thể liên kết với nhiều thực thể khác. Ví dụ: Một tác giả có thể viết nhiều cuốn sách, nhưng một cuốn sách chỉ có thể có một tác giả.
- Nhiều-nhiều (n-n): Nhiều thực thể liên kết với nhiều thực thể khác. Ví dụ: Một sinh viên có thể học nhiều môn học, và một môn học có thể được học bởi nhiều sinh viên.

2.1.4. Ưu điểm:

- ERD sử dụng biểu diễn đồ họa đơn giản, dễ hiểu và dễ dàng truyền tải thông tin.
- ERD cho phép người dùng nhìn thấy toàn bộ cấu trúc dữ liệu, bao gồm các thực thể, thuộc tính và mối quan hệ giữa chúng.
- ERD là công cụ quan trọng trong thiết kế cơ sở dữ liệu, đảm bảo tính nhất quán và hiệu quả của dữ liệu.

2.1.5. Hạn chế:

- ERD có thể khó sử dụng để mô hình hóa dữ liệu phức tạp, có nhiều mối quan hệ và quy tắc phức tạp.
- ERD chỉ mô tả cấu trúc tĩnh của dữ liệu, không thể mô tả hành vi của dữ liệu, như các quy tắc kinh doanh hoặc luồng công việc.

2.2. Phụ thuộc hàm

2.2.1. Khái niệm

- Phụ thuộc hàm là một khái niệm cơ bản trong lý thuyết cơ sở dữ liệu, đặc biệt trong thiết kế cơ sở dữ liệu quan hệ.
- Nó thể hiện mối quan hệ giữa các thuộc tính trong một bảng.
- Nói cách khác, nó cho biết một thuộc tính (hay tập hợp thuộc tính) có thể được xác định duy nhất bởi một thuộc tính (hay tập hợp thuộc tính) khác.
- Ví dụ (Bảng khách hàng): Mã khách hàng(MaKH) \rightarrow Tên khách hàng, số điện thoại.
- \rightarrow Nếu biết mã khách hàng ta có thể biết tên và số điện thoại của sinh viên đó.

2.2.2. Các loại phụ thuộc hàm:

- Phụ thuộc hàm đầy đủ: $X \rightarrow Y$ nếu và chỉ nếu $X \rightarrow Y$ và không có tập con thực sự của $X \rightarrow Y$.
- Phụ thuộc hàm một phần: $X \rightarrow Y$ nếu $X \rightarrow Y$ và có tập con thực sự của $X \rightarrow Y$.
- Phụ thuộc hàm chuyển tiếp: $X \rightarrow Y, Y \rightarrow Z$ thì $X \rightarrow Z$.

2.2.3. Quy tắc:

- Quy tắc phản xạ: $X \rightarrow X$ (một thuộc tính luôn phụ thuộc hàm vào chính nó).
- Quy tắc tăng cường: Nếu $X \rightarrow Y$ thì $XZ \rightarrow YZ$ (có thể thêm thuộc tính vào cả hai vế của phụ thuộc hàm).
- Quy tắc chuyển tiếp: Nếu $X \rightarrow Y$ và $Y \rightarrow Z$ thì $X \rightarrow Z$ (nếu X xác định Y và Y xác định Z thì X xác định Z).

- Quy tắc hợp nhất (Union): Nếu $X \rightarrow Y$ và $X \rightarrow Z$ thì $X \rightarrow YZ$ (nếu X xác định Y và X xác định Z thì X xác định cả Y và Z).
- Quy tắc phân chia (Decomposition): Nếu $X \rightarrow YZ$ thì $X \rightarrow Y$ và $X \rightarrow Z$ (nếu X xác định cả Y và Z thì X cũng xác định Y và X cũng xác định Z).

2.2.4. Ứng dụng:

- Phụ thuộc hàm giúp xác định khóa chính, khóa ngoại và thiết kế bảng hiệu quả.
- Phụ thuộc hàm là cơ sở cho việc bình thường hóa cơ sở dữ liệu, nhằm loại bỏ sự trùng lặp dữ liệu và đảm bảo tính nhất quán của dữ liệu.
- Phụ thuộc hàm giúp xác định các ràng buộc dữ liệu và kiểm tra tính nhất quán của dữ liệu trong bảng.

2.2.5. Hạn chế:

- Phụ thuộc hàm chỉ phản ánh mối quan hệ giữa các thuộc tính trong một bảng, không thể mô tả hành vi của dữ liệu.

2.3. Các dạng chuẩn

2.3.1. Dạng chuẩn 1NF:

- Điều kiện: Mỗi ô trong bảng chỉ chứa một giá trị đơn lẻ, không chứa tập hợp các giá trị hoặc các cấu trúc dữ liệu phức tạp.
- Ví dụ: Bảng "Sinh viên" với các cột: Mã SV, Tên, Địa chỉ, Danh sách môn học. Không hợp lệ vì cột "Danh sách môn học" chứa tập hợp các giá trị.
- Cách sửa: Tách cột "Danh sách môn học" thành một bảng riêng "Môn học", mỗi hàng trong bảng "Môn học" tương ứng với một môn học, và

thêm cột "Mã SV" vào bảng "Môn học" để tạo mối liên hệ với bảng "Sinh viên".

2.3.2. Dạng chuẩn 2NF:

- Điều kiện: Phải thỏa mãn 1NF và tất cả các thuộc tính phụ thuộc hàm vào khóa chính, không phụ thuộc hàm vào một phần của khóa chính.
- Ví dụ: Bảng "Sản phẩm" với các cột: Mã SP, Tên SP, Giá, Mã Loại SP, Tên Loại SP. Không hợp lệ vì cột "Tên Loại SP" phụ thuộc hàm vào cột "Mã Loại SP", mà không phụ thuộc vào toàn bộ khóa chính "Mã SP".
- Cách sửa: Tách cột "Tên Loại SP" thành một bảng riêng "Loại SP", mỗi hàng trong bảng "Loại SP" tương ứng với một loại sản phẩm, và thêm cột "Mã Loại SP" vào bảng "Sản phẩm" để tạo mối liên hệ với bảng "Loại SP".

2.3.3. Dạng chuẩn 3NF:

- Điều kiện: Phải thỏa mãn 2NF và tất cả các thuộc tính không phụ thuộc hàm vào khóa chính, phải phụ thuộc hàm vào khóa chính.
- Ví dụ: Bảng "Nhân viên" với các cột: Mã NV, Tên NV, Lương, Mã Phòng Ban, Tên Phòng Ban. Không hợp lệ vì cột "Tên Phòng Ban" phụ thuộc hàm vào cột "Mã Phòng Ban", mà không phụ thuộc vào khóa chính "Mã NV".
- Cách sửa: Tách cột "Tên Phòng Ban" thành một bảng riêng "Phòng Ban", mỗi hàng trong bảng "Phòng Ban" tương ứng với một phòng ban, và thêm cột "Mã Phòng Ban" vào bảng "Nhân viên" để tạo mối liên hệ với bảng "Phòng Ban".

2.3.4. Dạng chuẩn boyce-codd (BCNF):

- Điều kiện: Phải thỏa mãn 3NF và mọi thuộc tính phải phụ thuộc hàm vào khóa chính (không có thuộc tính nào phụ thuộc hàm vào một phần của khóa chính).
- Ví dụ: Bảng "Học sinh" với các cột: Mã HS, Tên HS, Mã Lớp, Tên Lớp. Không hợp lệ vì cột "Tên Lớp" phụ thuộc hàm vào cột "Mã Lớp", mà không phụ thuộc vào khóa chính "Mã HS".
- Cách sửa: Tách cột "Tên Lớp" thành một bảng riêng "Lớp", mỗi hàng trong bảng "Lớp" tương ứng với một lớp học, và thêm cột "Mã Lớp" vào bảng "Học sinh" để tạo mối liên hệ với bảng "Lớp".

2.3.5. Dạng chuẩn 4NF:

- Điều kiện: Phải thỏa mãn BCNF và không có sự phụ thuộc đa trị (multi-valued dependency).
- Ví dụ: Bảng "Sinh viên" với các cột: Mã SV, Tên SV, Danh sách môn học, Danh sách giáo viên. Không hợp lệ vì có sự phụ thuộc đa trị giữa "Mã SV" và "Danh sách môn học", "Danh sách giáo viên".
- Cách sửa: Tách "Danh sách môn học" và "Danh sách giáo viên" thành các bảng riêng biệt, mỗi hàng tương ứng với một môn học hoặc giáo viên, và thêm cột "Mã SV" để tạo mối liên hệ với bảng "Sinh viên".

2.3.6. Ưu điểm của chuẩn hóa:

- Loại bỏ sự trùng lặp dữ liệu: Giảm thiểu không gian lưu trữ.
- Nâng cao tính nhất quán dữ liệu: Đảm bảo tính chính xác và nhất quán dữ liệu trong hệ thống.
- Dễ dàng cập nhật dữ liệu: Cập nhật dữ liệu ở một nơi duy nhất.

- Cải thiện hiệu suất truy vấn: Truy vấn dữ liệu nhanh hơn và hiệu quả hơn.

2.4. Bảo toàn thông tin

2.4.1. Các thuộc tính bảo mật:

- Bí mật: Chỉ những người được phép mới có thể truy cập thông tin.
- Toàn vẹn: Thông tin được bảo vệ khỏi bị sửa đổi hoặc xóa trái phép.
- Khả dụng: Thông tin có sẵn cho những người được phép truy cập khi họ cần.

2.4.2. Các mối đe dọa:

- Xâm nhập trái phép: Người dùng trái phép truy cập vào hệ thống hoặc thông tin.
- Sửa đổi thông tin: Người dùng trái phép sửa đổi thông tin.
- Xóa thông tin: Người dùng trái phép xóa thông tin.
- Phủ nhận: Người dùng phủ nhận đã thực hiện hành động nào đó.
- Tiết lộ thông tin: Thông tin được tiết lộ cho những người không được phép.
- Chối bỏ dịch vụ: Hệ thống bị tấn công hoặc bị lỗi khiến người dùng không thể truy cập thông tin.

2.4.3. Các kỹ thuật bảo mật:

- Mã hóa: Chuyển đổi thông tin thành dạng khó đọc, chỉ có người có khóa giải mã mới có thể đọc được.
- Kiểm soát truy cập: Thiết lập các quy tắc để xác định người nào có thể truy cập thông tin nào.

- Kiểm tra xác thực: Xác minh danh tính của người dùng trước khi họ được phép truy cập vào hệ thống.
- Xác minh toàn vẹn: Kiểm tra xem thông tin đã bị sửa đổi hay chưa.
- Phát hiện xâm nhập: Phát hiện các hoạt động bất thường trong hệ thống để ngăn chặn tấn công.
- Sao lưu và phục hồi: Sao lưu thông tin để có thể phục hồi trong trường hợp dữ liệu bị mất.

2.4.4. Các tiêu chuẩn bảo mật:

- ISO 27001: Tiêu chuẩn về hệ thống quản lý bảo mật thông tin.
- NIST Cybersecurity Framework: Khung khổ bảo mật cho các tổ chức.
- GDPR (General Data Protection Regulation): Quy định về bảo mật dữ liệu cá nhân.

Kết luận:

- Bảo toàn thông tin là một lĩnh vực phức tạp và liên tục phát triển. Hiểu rõ cơ sở lý thuyết bảo mật giúp chúng ta đưa ra các giải pháp hiệu quả để bảo vệ thông tin, đồng thời đảm bảo tính toàn vẹn và khả dụng của dữ liệu.

2.5. Lập trình cơ sở dữ liệu

2.5.1. Các khái niệm cơ bản:

- Cơ sở dữ liệu: Là một tập hợp có tổ chức các dữ liệu, được lưu trữ và quản lý bởi một hệ quản trị cơ sở dữ liệu.
- Hệ quản trị cơ sở dữ liệu: Là một phần mềm giúp quản lý cơ sở dữ liệu, bao gồm việc tạo, lưu trữ, truy vấn và cập nhật dữ liệu.
- Ngôn ngữ truy vấn có cấu trúc (SQL): Là ngôn ngữ tiêu chuẩn để truy vấn, thao tác và quản lý dữ liệu trong cơ sở dữ liệu quan hệ.

- Ngôn ngữ lập trình: Là công cụ để viết các lệnh và thuật toán để tương tác với cơ sở dữ liệu.

2.5.2. Các kỹ thuật lập trình cơ sở dữ liệu:

- Truy vấn dữ liệu (Data Querying): Sử dụng SQL để truy xuất dữ liệu từ cơ sở dữ liệu.
- Thao tác dữ liệu (Data Manipulation): Sử dụng SQL để chèn, cập nhật, xóa và sửa đổi dữ liệu trong cơ sở dữ liệu.
- Quản lý dữ liệu (Data Administration): Sử dụng SQL và các công cụ của DBMS để quản lý cấu trúc bảng, khóa chính, khóa ngoại, chỉ mục và các đối tượng khác trong cơ sở dữ liệu.
- Xây dựng ứng dụng (Application Development): Sử dụng ngôn ngữ lập trình (như Java, Python, C#, PHP) để kết nối với cơ sở dữ liệu, truy vấn dữ liệu và xây dựng các ứng dụng sử dụng dữ liệu từ cơ sở dữ liệu.

2.5.3. Các kỹ thuật nâng cao:

- Stored Procedure: Các đoạn mã SQL được lưu trữ trong cơ sở dữ liệu, có thể được gọi lại nhiều lần.
- Trigger: Các đoạn mã SQL tự động chạy khi có sự kiện xảy ra trong cơ sở dữ liệu.
- View: Là một dạng truy vấn SQL lưu trữ, cho phép truy cập dữ liệu từ nhiều bảng một cách dễ dàng.
- Transaction: Là một tập hợp các thao tác dữ liệu, được xử lý như một đơn vị duy nhất để đảm bảo tính nhất quán của dữ liệu.
- Concurrency Control: Các kỹ thuật để kiểm soát việc truy cập đồng thời vào cơ sở dữ liệu, tránh xung đột dữ liệu.

2.5.4. Ứng dụng:

- Xây dựng các trang web: Lưu trữ và quản lý dữ liệu người dùng, sản phẩm, bài viết, ...
- Phát triển ứng dụng di động: Lưu trữ và quản lý dữ liệu người dùng, vị trí, thông tin, ...
- Quản lý thông tin doanh nghiệp: Lưu trữ và quản lý dữ liệu khách hàng, nhân viên, sản phẩm, tài chính, ...
- Phân tích dữ liệu: Truy xuất và xử lý dữ liệu để tạo ra các báo cáo và phân tích.
- Học máy và trí tuệ nhân tạo: Lưu trữ và xử lý dữ liệu để đào tạo các mô hình học máy.

Kết luận:

- Lập trình cơ sở dữ liệu là một kỹ năng quan trọng trong phát triển phần mềm. Hiểu rõ các khái niệm và kỹ thuật lập trình cơ sở dữ liệu sẽ giúp bạn xây dựng các ứng dụng hiệu quả và đáng tin cậy.

2.6. Ràng buộc toàn vẹn cơ sở dữ liệu

2.6.1. Ràng buộc toàn vẹn cột:

- Ràng buộc không rỗng (NOT NULL): Cột này không được phép chứa giá trị NULL. Điều này đảm bảo rằng cột luôn có giá trị.
- Ràng buộc duy nhất (UNIQUE): Cột này chỉ được phép chứa các giá trị duy nhất, không được phép trùng lặp.
- Ràng buộc khóa chính (PRIMARY KEY): Cột này là khóa chính của bảng, đảm bảo tính duy nhất cho mỗi hàng.

- Ràng buộc kiểm tra (CHECK): Cột này phải thỏa mãn một điều kiện nhất định, ví dụ: giá trị phải nằm trong một phạm vi nhất định hoặc phải tuân theo một mẫu cụ thể.

2.6.2. Ràng buộc toàn vẹn bảng:

- Ràng buộc khóa ngoại (FOREIGN KEY): Cột này tham chiếu đến khóa chính của một bảng khác, đảm bảo mối quan hệ giữa các bảng.
- Ràng buộc duy nhất (UNIQUE): Một nhóm cột trong bảng chỉ được phép chứa các giá trị duy nhất, không được phép trùng lặp.

2.6.3. Ràng buộc toàn vẹn cơ sở dữ liệu:

- Ràng buộc toàn vẹn tham chiếu: Duy trì mối quan hệ giữa các bảng bằng cách đảm bảo rằng khóa ngoại trong một bảng luôn tham chiếu đến khóa chính hợp lệ trong bảng khác.
- Ràng buộc toàn vẹn domain: Đảm bảo rằng dữ liệu trong một cột tuân theo một tập hợp giá trị cho phép.

2.6.4. Ưu điểm:

- Đảm bảo tính chính xác của dữ liệu: Hạn chế dữ liệu sai lệch và không hợp lệ.
- Nâng cao tính nhất quán của dữ liệu: Giúp duy trì mối quan hệ giữa các bảng và đảm bảo tính nhất quán của dữ liệu.
- Cải thiện hiệu suất truy vấn: Giúp truy vấn dữ liệu nhanh hơn và hiệu quả hơn.
- Giảm thiểu lỗi: Hạn chế lỗi do người dùng nhập liệu sai hoặc thao tác dữ liệu không đúng.
- Tăng cường bảo mật dữ liệu: Hỗ trợ việc kiểm soát truy cập và bảo vệ dữ liệu khỏi bị sửa đổi trái phép.

- Ví dụ:

- Bảng "Sinh viên" với khóa chính "Mã SV":

NOT NULL cho cột "Tên SV" để đảm bảo mỗi sinh viên phải có tên.

UNIQUE cho cột "Mã SV" để đảm bảo mỗi sinh viên có mã SV duy nhất.

- Bảng "Môn học" với khóa chính "Mã MH":

FOREIGN KEY cho cột "Mã GV" tham chiếu đến khóa chính "Mã GV" trong bảng "Giảng viên" để đảm bảo mỗi môn học phải có giáo viên phụ trách.

- Kết luận:

Ràng buộc toàn vẹn là một công cụ quan trọng trong thiết kế và quản lý cơ sở dữ liệu. Nó giúp đảm bảo tính chính xác, nhất quán và bảo mật của dữ liệu, đồng thời nâng cao hiệu suất truy vấn và giảm thiểu lỗi.

2.7. noSQL và newSQL

2.7.1. noSQL:

- Khái niệm: NoSQL là một loại cơ sở dữ liệu không tuân theo mô hình quan hệ truyền thống. Nó cung cấp cách thức lưu trữ và truy xuất dữ liệu linh hoạt hơn, phù hợp với các nhu cầu dữ liệu phi cấu trúc và nửa cấu trúc.

- Đặc điểm:

+ Dễ dàng mở rộng: Dễ dàng thêm nhiều máy chủ vào hệ thống để xử lý khối lượng dữ liệu lớn.

+ Linh hoạt về dữ liệu: Hỗ trợ nhiều loại dữ liệu, bao gồm dữ liệu phi cấu trúc, nửa cấu trúc và dữ liệu có cấu trúc.

- + Hiệu suất cao: Thường có hiệu suất cao hơn RDBMS trong các hoạt động đọc/ghi dữ liệu.

- + Không cần thiết kế bảng: Không cần thiết kế bảng theo kiểu quan hệ truyền thống.

- Các loại noSQL:

- + Cơ sở dữ liệu tài liệu (Document databases): Lưu trữ dữ liệu dưới dạng các tài liệu JSON hoặc XML, ví dụ: MongoDB, Couchbase.

- + Cơ sở dữ liệu cột (Column-family databases): Lưu trữ dữ liệu theo cột, ví dụ: Cassandra, HBase.

- + Cơ sở dữ liệu đồ thị (Graph databases): Lưu trữ dữ liệu theo cấu trúc đồ thị, ví dụ: Neo4j, OrientDB.

- + Cơ sở dữ liệu kho lưu trữ giá trị (Key-value stores): Lưu trữ dữ liệu dưới dạng các cặp khóa-giá trị, ví dụ: Redis, Memcached.

- Ứng dụng:

- + Các ứng dụng truyền thông xã hội

- + Các ứng dụng phân tích dữ liệu lớn

- + Các ứng dụng IoT (Internet of Things)

- + Các ứng dụng cần khả năng mở rộng và linh hoạt cao

2.7.2. newSQL:

- Khái niệm: NewSQL là một loại cơ sở dữ liệu kết hợp những ưu điểm của cả NoSQL và RDBMS, nhằm cung cấp khả năng mở rộng và hiệu suất cao như NoSQL nhưng vẫn giữ được tính nhất quán và độ tin cậy của RDBMS.

- Đặc điểm:

- + Mở rộng quy mô: Có thể xử lý khối lượng dữ liệu lớn.

- + Hiệu suất cao: Có thể xử lý các hoạt động đọc/ghi dữ liệu với tốc độ cao.

- + Tính nhất quán cao: Đảm bảo tính nhất quán của dữ liệu, giống như RDBMS.

- + Hỗ trợ ACID: Hỗ trợ các thuộc tính ACID (Atomicity, Consistency, Isolation, Durability) để đảm bảo tính toàn vẹn của dữ liệu.

- Các loại newSQL:

- + VoltDB

- + MemSQL

- + NuoDB

- Ứng dụng:

- + Các ứng dụng giao dịch trực tuyến (OLTP)

- + Các ứng dụng cần khả năng mở rộng và độ tin cậy cao, đồng thời đảm bảo tính nhất quán của dữ liệu.

2.7.3. Lựa chọn cơ sở dữ liệu:

- Chọn NoSQL khi:

- + Cần khả năng mở rộng và linh hoạt cao.

- + Xử lý dữ liệu phi cấu trúc hoặc nửa cấu trúc.

- + Không yêu cầu tính nhất quán cao.

- Chọn NewSQL khi:

- + Cần khả năng mở rộng và hiệu suất cao.

- + Cần đảm bảo tính nhất quán của dữ liệu.

- + Sử dụng trong các ứng dụng giao dịch trực tuyến.

- Kết luận:

- NoSQL và NewSQL là những lựa chọn thay thế cho RDBMS, cung cấp giải pháp cho các nhu cầu dữ liệu phức tạp và ngày càng tăng. Việc lựa chọn loại cơ sở dữ liệu phù hợp phụ thuộc vào các yêu cầu cụ thể của ứng dụng và dữ liệu.

3.2. Mô tả thực thể

3.2.1. Thực thể: khách hàng

- Thông tin thuộc tính:

Stt	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Ghi chú
1	MaKH	Char	5	Primary key	Duy nhất
2	TenKH	Nvarchar	50	Not null	
3	SoDT	Varchar	20	Not null	
4	Phai	Nvarchar	10	Not null	
5	NgaySinh	Date		Not null	
6	DiaChi	Nvarchar	250	Nullable	
7	Email	Varchar	250	Nullable	

3.2.2. Thực thể: Xe

- Thông tin thuộc tính:

Stt	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Ghi chú
1	MaXe	Char	5	Primary key	Duy nhất
2	BienSoXe	Varchar	50	Not null	
3	NamSX	Int		Not null	
4	HangXe	Varchar	50	Not null	

3.2.3. Thực thể: Loại khách hàng

- Thông tin thuộc tính:

Stt	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Ghi chú
1	MaLoaiKH	Char	5	Primary key	Duy nhất
2	TenLoaiKH	Nvarchar	50	Not null	

3.2.4. Thực thể : Loại xe

Stt	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Ghi chú
1	MaLoaiXe	Char	5	Primary key	Duy nhất
2	TenLoaiXe	Nvarchar	50	Not null	

3.2.5. Thực thể: Phiếu dịch vụ

Stt	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Ghi chú
1	MaPhieuDV	Char	5	Primary key	Duy nhất
2	NgayLapPhieu	Date		Not null	
3	TongTienDV	Int		Nullable	
4	TongTienPT	Int		Nullable	

5	TongTien	Int		Nullable	
----------	----------	-----	--	----------	--

3.2.6. Thực thể: Dịch vụ

- Thông tin thuộc tính:

Stt	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Ghi chú
1	MaDV	Char	5	Primary key	Duy nhất
2	TenDV	Nvarchar	50	Not null	
3	GiaDV	Int		Not null	

3.2.7. Thực thể: Loại dịch vụ

- Thông tin thuộc tính:

Stt	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Ghi chú
1	MaLoaiDV	Char	5	Primary key	Duy nhất
2	TenLoaiDV	Nvarchar	50	Not null	

3.2.8. Thực thể: phụ tùng

- Thông tin thuộc tính:

Stt	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Ghi chú
1	MaPT	Char	5	Primary key	Duy nhất
2	TenPT	Nvarchar	50	Not null	
3	DonViTinh	Nvarchar	10	Not null	
4	GiaPT	Int		Not null	
5	SoLuongTK	Int		Not null	

3.2.9. Thực thể: Loại phụ tùng

- Thông tin thuộc tính:

Stt	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Ghi chú
1	MaLoaiPT	Char	5	Primary key	Duy nhất
2	TenLoaiPT	Int		Not null	

3.2.10. Thực thể: Kho

- Thông tin thuộc tính:

Stt	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Ghi chú
1	MaKho	Char	5	Primary key	Duy nhất
2	TenKho	Nvarchar	50	Not null	

3.2.11. Thực thể: Phiếu nhập kho

- Thông tin thuộc tính:

Stt	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Ghi chú
1	MaPhieuNK	Char	5	Primary key	Duy nhất
2	NgayLapPhieu	Date		Not null	
4	TongTien	Int		Not null	

3.2.12. Thực thể: Nhân viên

- Thông tin thuộc tính:

Stt	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Ghi chú
1	MaNV	Char	5	Primary key	Duy nhất
2	HoTenNV	Nvarchar	50	Not null	
3	SoDT	Varchar	20	Not null	
4	Phai	Nvarchar	10	Not null	
5	NgaySinh	Date		Not null	
6	Email	Varchar	50	Nullable	
7	DiaChi	Nvarchar	250	Nullable	
8	ChucVu	Nvarchar	20	Not null	

3.2.13. Thực thể: hóa đơn

Stt	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Ghi chú
1	MaHD	Char	5	Primary key	Duy nhất
2	NgayLapHD	Date		Not null	
3	TongTien	Int		Not Null	

3.3. Lược đồ cơ sở dữ liệu quan hệ

3.3.1. Chuyển thực thể mạnh và các thuộc tính đơn:

- LoaiKH(MaLoaiKH, TenLoaiKH)
- LoaiXe(MaLoaiXe, TenLoaiXe)
- LoaiDV(MaLoaiDV, TenLoaiDV)
- Kho(MaKho, TenKho)
- NhanVien(MaNV, TenNV, Phai, NgaySinh, Email, ChucVu)

3.3.2. Chuyển các thuộc tính kết hợp

- KháchHang(MaKH, TenKH, Phai, NgaySinh Email, DiaChi)
- NhanVien(MaNV, TenNV, Phai, NgaySinh Email, ChucVu, DiaChi)

3.3.3. Chuyển các thuộc tính đa trị

- KHDT(MaKH, SoDT)
- NVDT(MaNV, SoDT)

3.3.4. Chuyển quan hệ (1 – n)

(1 loại khách hàng có thể có nhiều khách hàng)

- KháchHang(MaKH, TenKH, Email, Phai, NgaySinh DiaChi, **MaLoaiKH**)

(1 loại xe có thể có nhiều xe)

- Xe(MaXe, BienSoXe, HangXe, NamSX, **MaLoaiXe**)

(1 khách hàng có thể sở hữu nhiều xe)

- Xe(MaXe, BienSoXe, HangXe, NamSX, **MaLoaiXe**, **MaKH**)

(1 xe có thể có nhiều phiếu dịch vụ)

- PhieuDV(MaPhieuDV, NgayLapPhieu, TongTienDV, TongTienPT, TongTien, **MaXe**)

(1 loại dịch vụ có thể có nhiều dịch vụ)

- DichVu(MaDV, TenDV, GiaDV, **MaLoaiDV**)

(1 loại phụ tùng có thể có nhiều phụ tùng)

- PhuTung(MaPT, TenPT, GiaPT, DonViTinh, SoLuongTL, **MaLoaiPT**)

(1 kho có thể có nhiều phụ tùng)

- Kho(MaKho, TenKho, **MaPT**)

(1 Nhân viên có thể thực hiện nhiều phiếu dịch vụ)

- PhieuDV(MaPhieuDV, NgayLapPhieu, TongTien, **MaXe**, **MaNV**)

(1 nhân viên có thể thực hiện nhiều phiếu nhập kho)

- PhieuNK(MaPhieuNK, NgayLapPhieu, TongTienDV, TongTienPT, Tongtien, **MaNV**)

(1 Phiếu dịch vụ có thể tạo nhiều hóa đơn)

- HoaDon(MaHD, NgayLapHD, TongTien, **MaPhieuDV**)

3.3.5. Chuyên quan hệ n-n

- ChiTietDV(MaPhieuDV, MaDV)

- ChiTietPT(MaPhieuDV, MaPT, SoLuong)

- ChiTietPhieuNK(MaPhieuNK, MaPT, NgayNhapKho, SoLuongNhap)

3.3.6. kết quả

- KhachHang(MaKH, TenKH, Phai, NgaySinh, Email, DiaChi **MaLoaiKH**)

- Xe(MaXe, BienSoXe, HangXe, NamSX, **MaLoaiXe**, **MaKH**)

- PhieuDV(MaPhieuDV, NgayLapPhieu, TongTienDV, TongTienPT, TongTien, **MaXe**, **MaNV**)

- DichVu(MaDV, TenDV, GiaDV, **MaLoaiDV**)

- PhuTung(MaPT, TenPT, GiaPT, DonViTinh, SoLuongTL, **MaLoaiPT**)

- Kho(MaKho, TenKho, **MaPT**)

- PhieuNK(MaPhieuNK, NgayLapPhieu, TongTien, **MaNV**)

- NhanVien(MaNV, TenNV, Phai, NgaySinh, Email, ChucVu, DiaChi)

- LoaiKH(MaLoaiKH, TenLoaiKH)

- LoaiXe(MaLoaiXe, TenLoaiXe)
- LoaiDV(MaLoaiDV, TenLoaiDV)
- KHDT(MaKH, SoDT)
- NVDT(MaNV, SoDT)
- ChiTietDV(MaPhieuDV, MaDV)
- ChiTietPT(MaPhieuDV, MaPT, SoLuong)
- ChiTietPhieuNK(MaPhieuNK, MaPT, NgayNhapKho, SoLuongNhap)
- HoaDon(MaHD, NgayLapHD, TongTien, **MaPhieuDV**)

3.4. Phân tích chuẩn của lược đồ quan hệ

3.4.1. Lược đồ quan hệ KháchHang

1. Chuẩn hóa 1NF

- Lược đồ KháchHang(MaKH, TenKH, SoDT, Phai, Email, DiaChi, MaLoaiKH, TenLoaiKH) với SoDT là thuộc tính đa trị
- + Loại SoDT ra khỏi KháchHang: KháchHang(MaKH, TenKH, Phai, Email, DiaChi, MaLoaiKH, TenLoaiKH)
- + Tạo lược đồ quan hệ mới DTKH(MaKH, SoDT)

2. Chuẩn hóa 2NF

- Lược đồ quan hệ ở dạng chuẩn 1NF:
- KháchHang(MaKH, TenKH, Phai, Email, DiaChi, MaLoaiKH, TenLoaiKH) với các phụ thuộc hàm:
 - + {MaKH, MaLoaiKH} → {TenKH, Phai, Email, TenLoaiKH, DiaChi}
 - + MaKH → {TenKH, Phai, Email, DiaChi}
 - + MaLoaiKH → TenLoaiKH

- Chuẩn hóa về 2NF:
- Loại TenLoaiKH ra khỏi KhachHang:
- + KhachHang(MaKH, TenKH, Phai, Email, DiaChi, MaLoaiKH)
- + Tạo lược đồ quan hệ mới LoaiKH(MaLoaiKH, TenLoaiKH)

3.4.2. Lược đồ quan hệ Xe

- Lược đồ đang ở dạng chuẩn 1NF:
- Xe(MaXe, HangXe, NamSX, MaLoaiXe, TenLoaiXe, MaKH) với các phụ thuộc hàm:
- + MaXe, MaLoaiXe \rightarrow (HangXe, NamSX, TenLoaiXe, MaKH)
- + MaXe \rightarrow (HangXe, NamSX, MaKH)
- + MaLoaiXe \rightarrow (TenLoaiXe)
- Để chuẩn hóa 2NF:
- + Loại TenLoaiXe ra khỏi lược đồ Xe: Xe(MaXe, HangXe, NamSX, MaLoaiXe, MaKH)
- + Tạo lược đồ quan hệ mới LoaiXe(MaLoaiXe, TenLoaiXe)

3.4.3. Lược đồ quan hệ DichVu

- Lược đồ đang ở dạng chuẩn 1NF:
- DichVu(MaDV, TenDV, GiaDV, MaLoaiDV, TenLoaiDV) với các phụ thuộc hàm:
- + MaDV, MaLoaiDV \rightarrow (TenDV, GiaDV, TenLoaiDV)
- + MaDV \rightarrow (TenDV, GiaDV)
- + MaLoaiDV \rightarrow (TenLoaiDV)
- Để chuẩn hóa 2NF:

- + Loại TenLoaiDV ra khỏi lược đồ DịchVu: DịchVu(MaDV, TenDV, GiaDV, MaLoaiDV)
- + Tạo lược đồ quan hệ mới LoaiDV(MaLoaiDV, TenLoaiDV)

3.4.4. Lược đồ quan hệ PhuTung

1. Chuẩn hóa 2NF

- Lược đồ đang ở dạng chuẩn 1NF
- PhuTung(MaPT, TenPT, DonViTinh, GiaPT, MaKho, TenKho, SoLuongTK, MaLoaiPT, TenLoaiPT) với các phụ thuộc hàm:
 - + MaPT, MaLoaiPT \rightarrow (TenPT, DonViTinh, GiaPT, MaKho, TenKho, SoLuongTK, TenLoaiPT)
 - + MaPT \rightarrow (TenPT, DonViTinh, GiaPT, MaKho, TenKho, SoLuongTK)
 - + MaLoaiPT \rightarrow (TenLoaiPT)
- Để chuẩn hóa 2NF:
 - + Loại TenLoaiPT ra khỏi lược đồ PhuTung(MaPT, TenPT, DonViTinh, GiaPT, MaKho, TenKho, SoLuongTK, MaLoaiPT)
 - + Tạo lược đồ quan hệ mới LoaiPT(MaLoaiPT, TenLoaiPT)

2. Chuẩn hóa 3NF

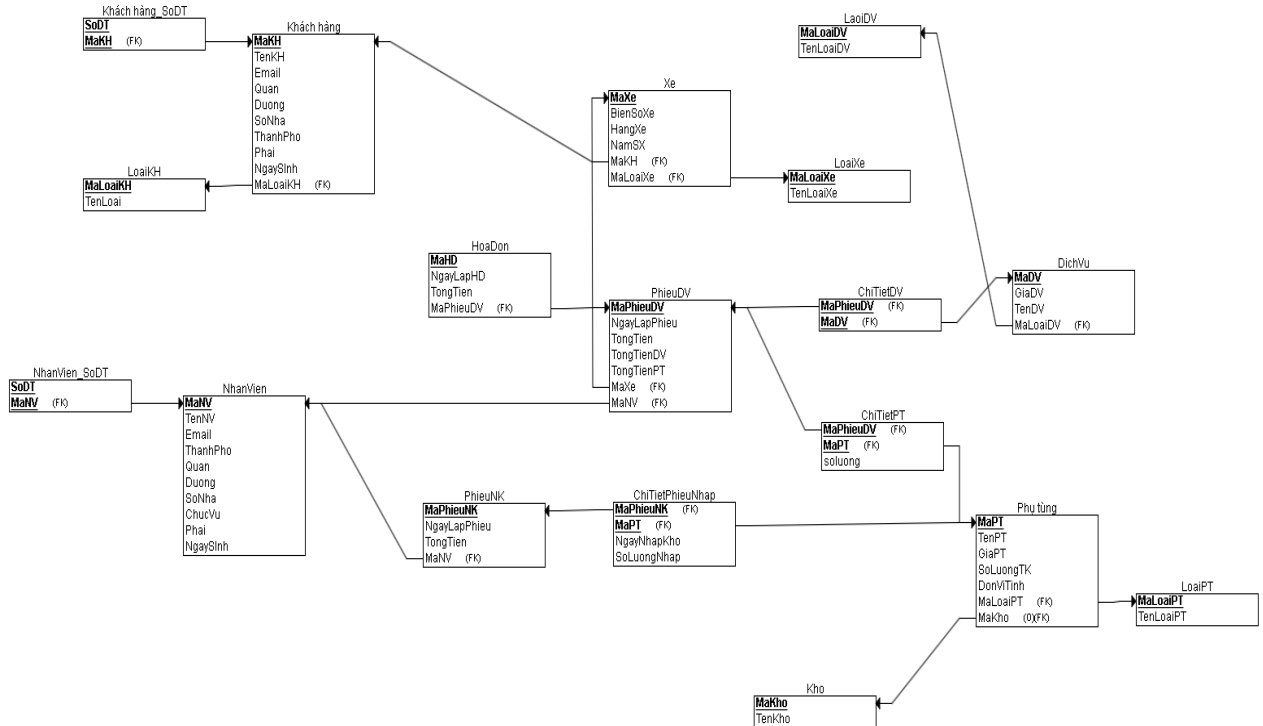
- Lược đồ PhuTung(MaPT, TenPT, DonViTinh, GiaPT, MaKho, TenKho, SoLuongTK, MaLoaiPT) với các phụ thuộc hàm:
 - MaPT \rightarrow (TenPT, DonViTinh, GiaPT, MaKho, TenKho, SoLuongTK, MaLoaiPT)
 - MaKho \rightarrow (TenKho)
- Lược đồ phụ tùng không ở dạng chuẩn 3NF vì phụ thuộc hàm bắc cầu MaPT \rightarrow MaKho \rightarrow TenKho

- Để chuẩn hóa PhụTung về 3NF:

+ Loại TenKho ra khỏi lược đồ phụ tùng: PhụTung(MaPT, TenPT, DonViTinh, GiaPT, MaKho, SoLuongTK, MaLoaiPT)

+ Tạo lược đồ quan hệ mới Kho(MaKho, TenKho)

3.5. Lược đồ quan hệ sau khi chuẩn hóa



3.6. Các ràng buộc toàn vẹn

3.6.1. Khách hàng, Nhân Viên

“Giới tính của khách hàng, nhân viên chỉ có thể là nam hoặc nữ”

- Bối cảnh: Khách hàng, nhân viên

- Bảng tầm ảnh hưởng :

Bối cảnh	Thêm	Xóa	Sửa
Khách hàng	+	-	+(Phai)

Bối cảnh	Thêm	Xóa	Sửa
Nhân viên	+	-	+(Phai)

“ Tuổi của khách hàng và nhân viên phải trên 18 tuổi”

- Bối cảnh: KháchHang, NhanVien

- Bảng ảnh hưởng:

Bối cảnh	Thêm	Xóa	Sửa
KhachHang	+	-	+(NgaySinh)

Bối cảnh	Thêm	Xóa	Sửa
NhanVien	+	-	+(NgaySinh)

“Email của khách hàng và nhân viên không được trùng nhau”

- Bối cảnh: NhanVien, KhachHang

- Bảng ảnh hưởng:

Bối cảnh	Thêm	Xóa	Sửa
NhanVien	+	-	+(Email)
KhachHang	+	-	+(Email)

3.6.2. Xe

“Chỉ được xóa xe khi không có khách hàng nào sở hữu”

- Bối cảnh: Xe

- Bảng ảnh hưởng:

Bối cảnh	Thêm	Xóa	Sửa
Xe	-	+	-

“Mỗi xe phải thuộc 1 thể loại xe trong gara”

- Bối cảnh: Xe,LoaiXe

- Bảng tầm ảnh hưởng

Bối cảnh	Thêm	Xóa	Sửa
LoaiXe	-	+	-
Xe	+	-	+(MaLoaiXe)

3.6.3. Dịch vụ

“Mỗi dịch vụ phải thuộc 1 loại dịch vụ trong gara”

- Bối cảnh: DichVu, LoaiDV

- Bảng ảnh hưởng:

Bối cảnh	Thêm	Xóa	Sửa
LoaiDV	-	+	-
DichVu	+	-	+(MaLoaiDV)

“Khi thêm dịch vụ tên dịch vụ không được trùng tên”

- Bối cảnh: DichVu

- Bảng ảnh hưởng:

Bối cảnh	Thêm	Xóa	Sửa
DichVu	+	-	+(TenDV)

“Khi thêm hoặc sửa dịch vụ giá dịch vụ phải lớn hơn 150000”

Bối cảnh	Thêm	Xóa	Sửa
DichVu	+	-	+(GiaDV)

3.6.4. Phụ tùng

“Mỗi phụ tùng phải thuộc 1 loại phụ tùng trong gara”

- Bối cảnh: PhuTung, LoaiPT

- Bảng ảnh hưởng

Bối cảnh	Thêm	Xóa	Sửa
LoaiPT	-	+	-
PhuTung	+	-	+(MaLoaiPT)

“Khi thêm phụ tùng tên phụ tùng không được trùng nhau”

- Bối cảnh: PhuTung

- Bảng ảnh hưởng

Bối cảnh	Thêm	Xóa	Sửa
PhuTung	+	-	+(TenPT)

“ Khi thêm hoặc sửa dịch phụ tùng thì giá phụ tùng phải lớn hơn 100000”

- Bối cảnh: PhuTung

- Bảng ảnh hưởng:

Bối cảnh	Thêm	Xóa	Sửa
PhuTung	+	-	+(GiaPT)

3.6.5. Phiếu dịch vụ

“Ngày lập phiếu dịch vụ phải là ngày hiện tại hoặc trước ngày hiện tại 7 ngày”

- Bối cảnh: Phiếu dịch vụ

- Bảng ảnh hưởng:

Bối cảnh	Thêm	Xóa	Sửa
PhuDV	+	-	+(PhieuDV)

3.6.6. Phiếu nhập kho

“Ngày lập phiếu phải là ngày hiện tại”

- Bối cảnh: Phiếu nhập kho

- Bảng ảnh hưởng:

Bối cảnh	Thêm	Xóa	Sửa
PhieuNK	+	-	+(NgayLapPhieu)

3.6.7. Hóa đơn

“Ngày lập hóa đơn phải là ngày hiện tại”

- Bối cảnh: HoaDon

- Bảng ảnh hưởng:

Bối cảnh	Thêm	Xóa	Sửa
HoaDon	+	-	+(NgayLapHD)

3.6.8. Chi tiết phiếu nhập kho

“Ngày nhập kho phải lớn hơn 4 ngày và bé hơn 7 ngày”

- Bối cảnh: ChiTietPhieuNhap

- Bảng ảnh hưởng:

Bối cảnh	Thêm	Xóa	Sửa
ChiTietPhieuNhap	+	-	+(NgayNhapKho)

Chương IV. Triển khai cơ sở dữ liệu trên MS SQL SERVER

4.1. Tạo cơ sở dữ liệu

```
create database PhamDuyHieu
```

```
use PhamDuyHieu
```

```
CREATE TABLE LoaiKH  
(  
    MaLoaiKH CHAR(5) NOT NULL,  
    TenLoai VARCHAR(50) NOT NULL,  
    PRIMARY KEY (MaLoaiKH)  
);
```

```
CREATE TABLE LoaiXe  
(  
    MaLoaiXe CHAR(5) NOT NULL,  
    TenLoaiXe VARCHAR(50) NOT NULL,  
    PRIMARY KEY (MaLoaiXe)  
);
```

```
CREATE TABLE NhanVien  
(  
    MaNV CHAR(5) NOT NULL,  
    TenNV VARCHAR(50) NOT NULL,  
    Email VARCHAR(100) NOT NULL,  
    ThanhPho VARCHAR(50) NOT NULL,  
    Quan VARCHAR(10) NOT NULL,  
    Duong VARCHAR(50) NOT NULL,  
    SoNha VARCHAR(50) NOT NULL,  
    ChucVu VARCHAR(50) NOT NULL,  
    Phai VARCHAR(10) NOT NULL,
```

```
    NgaySinh DATE NOT NULL,  
    PRIMARY KEY (MaNV)  
);
```

```
CREATE TABLE LaoiDV  
(  
    TenLoaiDV VARCHAR(50) NOT NULL,  
    MaLoaiDV CHAR(5) NOT NULL,  
    PRIMARY KEY (MaLoaiDV)  
);
```

```
CREATE TABLE LoaiPT  
(  
    TenLoaiPT VARCHAR(50) NOT NULL,  
    MaLoaiPT CHAR(5) NOT NULL,  
    PRIMARY KEY (MaLoaiPT)  
);
```

```
CREATE TABLE Kho  
(  
    MaKho CHAR(5) NOT NULL,  
    TenKho VARCHAR(50) NOT NULL,  
    PRIMARY KEY (MaKho)  
);
```

```
CREATE TABLE PhieuNK  
(  
    MaPhieuNK CHAR(5) NOT NULL,  
    NgayLapPhieu DATE NOT NULL,  
    TongTien INT,  
    MaNV CHAR(5) NOT NULL,  
    PRIMARY KEY (MaPhieuNK),  
    FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)
```

);

CREATE TABLE NhanVien_SoDT

(

SoDT VARCHAR(20) NOT NULL,

MaNV CHAR(5) NOT NULL,

PRIMARY KEY (SoDT, MaNV),

FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)

);

CREATE TABLE KhachHang

(

MaKH CHAR(5) NOT NULL,

TenKH VARCHAR(50) NOT NULL,

Email VARCHAR(100) NOT NULL,

Quan VARCHAR(10) NOT NULL,

Duong VARCHAR(50) NOT NULL,

SoNha VARCHAR(50) NOT NULL,

ThanhPho VARCHAR(50) NOT NULL,

Phai VARCHAR(10) NOT NULL,

NgaySinh DATE NOT NULL,

MaLoaiKH CHAR(5) NOT NULL,

PRIMARY KEY (MaKH),

FOREIGN KEY (MaLoaiKH) REFERENCES LoaiKH(MaLoaiKH)

);

CREATE TABLE Xe

(

MaXe CHAR(5) NOT NULL,

BienSoXe VARCHAR(50) NOT NULL,

HangXe VARCHAR(50) NOT NULL,

NamSX INT NOT NULL,

MaKH CHAR(5) NOT NULL,

```

MaLoaiXe CHAR(5) NOT NULL,
PRIMARY KEY (MaXe),
FOREIGN KEY (MaKH) REFERENCES KhachHang(MaKH),
FOREIGN KEY (MaLoaiXe) REFERENCES LoaiXe(MaLoaiXe)
);

```

```

CREATE TABLE Phụ_tùng
(
    MaPT CHAR(5) NOT NULL,
    TenPT VARCHAR(50) NOT NULL,
    GiaPT INT NOT NULL,
    SoLuongTK INT NOT NULL,
    DonViTinh VARCHAR(10) NOT NULL,
    MaLoaiPT CHAR(5) NOT NULL,
    MaKho CHAR(5),
    PRIMARY KEY (MaPT),
    FOREIGN KEY (MaLoaiPT) REFERENCES LoaiPT(MaLoaiPT),
    FOREIGN KEY (MaKho) REFERENCES Kho(MaKho)
);

```

```

CREATE TABLE PhieuDV
(
    MaPhieuDV CHAR(5) NOT NULL,
    NgayLapPhieu DATE NOT NULL,
    TongTien INT NOT NULL,
    TongTienDV INT ,
    TongTienPT INT ,
    MaXe CHAR(5) NOT NULL,
    MaNV CHAR(5) NOT NULL,
    PRIMARY KEY (MaPhieuDV),
    FOREIGN KEY (MaXe) REFERENCES Xe(MaXe),
    FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)
);

```

```

CREATE TABLE DichVu
(
    GiaDV INT NOT NULL,
    TenDV VARCHAR(50) NOT NULL,
    MaDV CHAR(5) NOT NULL,
    MaLoaiDV CHAR(5) NOT NULL,
    PRIMARY KEY (MaDV),
    FOREIGN KEY (MaLoaiDV) REFERENCES LaoiDV(MaLoaiDV)
);

```

```

CREATE TABLE HoaDon
(
    MaHD CHAR(5) NOT NULL,
    NgayLapHD DATE NOT NULL,
    TongTien INT,
    MaPhieuDV CHAR(5) NOT NULL,
    PRIMARY KEY (MaHD),
    FOREIGN KEY (MaPhieuDV) REFERENCES PhieuDV(MaPhieuDV)
);

```

```

CREATE TABLE ChiTietDV
(
    MaPhieuDV CHAR(5) NOT NULL,
    MaDV CHAR(5) NOT NULL,
    PRIMARY KEY (MaPhieuDV, MaDV),
    FOREIGN KEY (MaPhieuDV) REFERENCES PhieuDV(MaPhieuDV),
    FOREIGN KEY (MaDV) REFERENCES DichVu(MaDV)
);

```

```

CREATE TABLE ChiTietPhieuNhap
(
    NgayNhapKho DATE NOT NULL,

```

```

SoLuongNhap INT NOT NULL,
MaPhieuNK CHAR(5) NOT NULL,
MaPT CHAR(5) NOT NULL,
PRIMARY KEY (MaPhieuNK, MaPT),
FOREIGN KEY (MaPhieuNK) REFERENCES PhieuNK(MaPhieuNK),
FOREIGN KEY (MaPT) REFERENCES Phụ_tùng(MaPT)
);

```

```

CREATE TABLE ChiTietPT
(
    soluong INT NOT NULL,
    MaPhieuDV CHAR(5) NOT NULL,
    MaPT CHAR(5) NOT NULL,
    PRIMARY KEY (MaPhieuDV, MaPT),
    FOREIGN KEY (MaPhieuDV) REFERENCES PhieuDV(MaPhieuDV),
    FOREIGN KEY (MaPT) REFERENCES Phụ_tùng(MaPT)
);

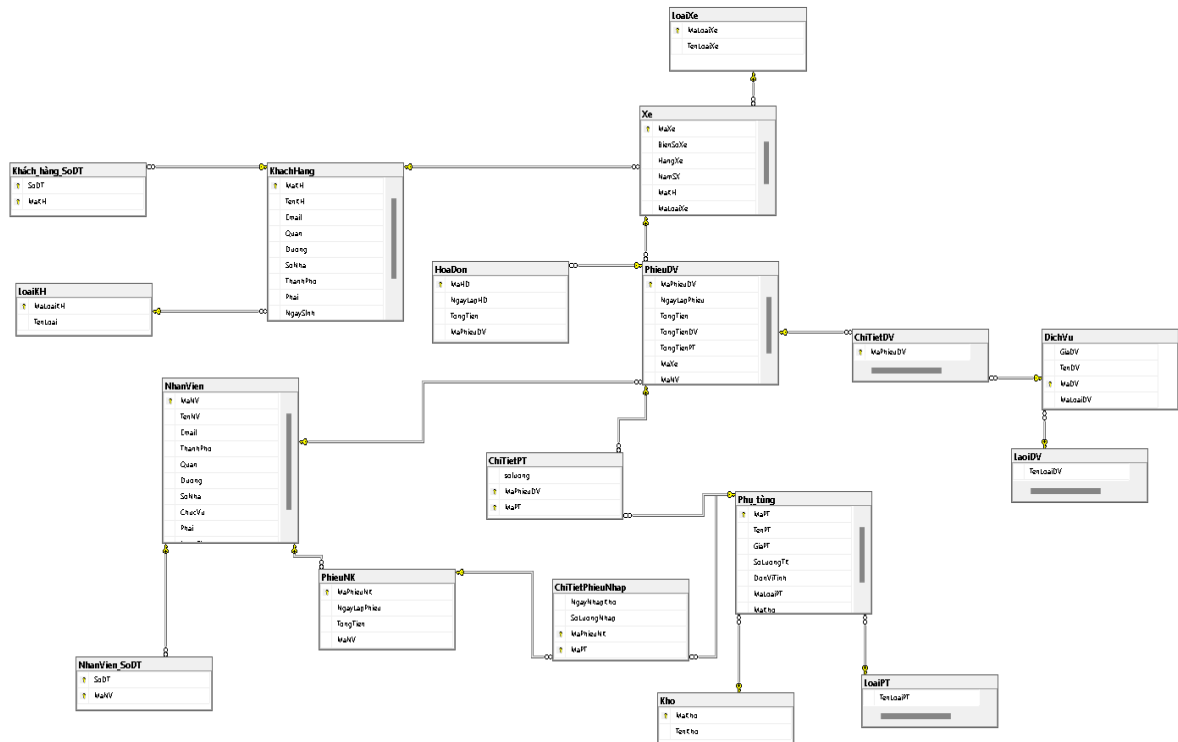
```

```

CREATE TABLE Khách_hàng_SoDT
(
    SoDT VARCHAR(20) NOT NULL,
    MaKH CHAR(5) NOT NULL,
    PRIMARY KEY (SoDT, MaKH),
    FOREIGN KEY (MaKH) REFERENCES KhachHang(MaKH)
);

```


- Sơ đồ cơ sở dữ liệu:



4.2. Nhập liệu mẫu

INSERT INTO LoaiKH (MaLoaiKH, TenLoai) VALUES

('KH01', N'Khách hàng thường xuyên'),

('KH02', N'Khách hàng VIP'),

('KH03', N'Khách hàng tiềm năng'),

('KH04', N'Khách hàng đã sử dụng dịch vụ một lần'),

('KH05', N'Khách hàng giới thiệu');

INSERT INTO Khách_hàng_SoDT (SoDT, MaKH) VALUES

('0901234567', 'KH001'),

('0912345678', 'KH001'),

('0987654321', 'KH002'),
('0976543210', 'KH003'),
('0965432109', 'KH004'),
('0954321098', 'KH005');

INSERT INTO LoaiXe (MaLoaiXe, TenLoaiXe) VALUES

('LX01', N'Ô tô con'),
('LX02', N'Ô tô tải'),
('LX05', N'Xe tải nhỏ'),
('LX06', N'Xe bán tải');

INSERT INTO NhanVien (MaNV, TenNV, Email, ThanhPho, Quan,
Duong, SoNha, ChucVu, Phai, NgaySinh) VALUES

('NV01', N'Nguyễn Văn A', 'nguyenvana@gara.com', N'Hồ Chí Minh',
N'Quận 1', N'Đường Lê Lợi', '123', N'Kỹ thuật viên', N'Nam', '1990-01-
01'),

('NV02', N'Trần Thị B', 'tranthib@gara.com', N'Hồ Chí Minh', N'Quận
2', N'Đường Nguyễn Văn Linh', '456', N'Lễ tân', N'Nữ', '1992-03-15'),

('NV03', N'Lê Văn C', 'levanc@gara.com', N'Hồ Chí Minh', N'Quận 3',
N'Đường Võ Thị Sáu', '789', N'Giám đốc', N'Nam', '1985-07-20'),

('NV04', N'Phạm Thị D', 'phamthid@gara.com', N'Hồ Chí Minh',
N'Quận 4', N'Đường Đồng Khởi', '101', N'Kế toán', N'Nữ', '1988-05-
10'),

('NV05', N'Võ Văn E', 'vovane@gara.com', N'Hồ Chí Minh', N'Quận 5', N'Đường Trần Hưng Đạo', '111', N'Thợ sửa chữa', N'Nam', '1995-09-25');

INSERT INTO LaoiDV (MaLoaiDV, TenLoaiDV)VALUES

('DV01', N'Bao duong dinh ky'),
('DV02', N'Sửa chữa động cơ'),
('DV03', N'Sửa chữa hệ thống điện'),
('DV04', N'Sửa chữa hệ thống phanh'),
('DV05', N'Sửa chữa hệ thống lái'),
('DV06', N'Sửa chữa gầm xe'),
('DV07', N'Vệ sinh nội thất'),
('DV08', N' Sơn sửa'),
('DV09', N'Thay thế phụ tùng'),
('DV10', N'Kiểm tra khí thải');

INSERT INTO LoaiPT (MaLoaiPT, TenLoaiPT) VALUES

('PT01', N'Lốp xe'),
('PT02', N'Ắc quy'),
('PT03', N'Bugì'),
('PT04', N'Dầu nhớt'),
('PT05', N'Bộ lọc gió'),
('PT06', N'Bộ lọc dầu'),

```
('PT07', N'Cùm thắng'),  
('PT08', N'Đĩa thắng'),  
('PT09', N'Bô pô'),  
('PT10', N'Gương chiếu hậu');
```

```
INSERT INTO Kho (MaKho, TenKho) VALUES
```

```
('KH01', N'Kho chính'),  
('KH02', N'Kho phụ tùng'),  
('KH03', N'Kho phụ kiện');
```

```
INSERT INTO NhanVien_SoDT (MaNV, SoDT) VALUES
```

```
('NV01', '0912345678'),  
('NV01', '0912345699'),  
('NV02', '0934567890'),  
('NV03', '0987654321'),  
('NV04', '0901234567'),  
('NV04', '0901239876'),  
('NV05', '0967890123');
```

```
INSERT INTO KháchHang (MaKH, TenKH, Email, Quan, Duong,  
SoNha, ThanhPho, Phai, NgaySinh, MaLoaiKH) VALUES
```

```
('KH001', N'Nguyễn Văn A', 'nguyenvana@example.com', N'Quận 1',  
N'Lê Lợi', '123', N'Hồ Chí Minh', N'Nam', '1980-01-01', 'KH01'),
```

('KH002', N'Trần Thị B', 'tranthib@example.com', N'Quận 3', N'Nguyễn Đình Chiêu', '456', N'Hồ Chí Minh', N'Nữ', '1990-02-02', 'KH02'),
 ('KH003', N'Lê Văn C', 'levanc@example.com', N'Quận 5', N'Trần Hưng Đạo', '789', N'Hồ Chí Minh', 'Nam', N'1985-03-03', 'KH03'),
 ('KH004', N'Phạm Thị D', 'phamthid@example.com', N'Quận 7', N'Nguyễn Văn Linh', '321', N'Hồ Chí Minh', N'Nữ', '1995-04-04', 'KH04'),
 ('KH005', N'Hoàng Văn E', 'hoangvane@example.com', N'Quận 10', N'Cách Mạng Tháng 8', '654', N'Hồ Chí Minh', N'Nam', '1982-05-05', 'KH05');

INSERT INTO Xe (MaXe, BienSoXe, HangXe, NamSX, MaKH, MaLoaiXe) VALUES

('XE001', '51A-12345', 'Toyota', 2010, 'KH001', 'LX01'),
 ('XE002', '51B-67890', 'Ford', 2015, 'KH002', 'LX02'),
 ('XE003', '51C-11122', 'Hyundai', 2018, 'KH003', 'LX05'),
 ('XE004', '51D-33344', 'Chevrolet', 2020, 'KH004', 'LX06'),
 ('XE005', '51E-55566', 'Honda', 2012, 'KH005', 'LX01');

INSERT INTO PhieuDV (MaPhieuDV, NgayLapPhieu, TongTien, TongTienDV, TongTienPT, MaXe, MaNV) VALUES

('PDV01', '2023-01-01', 3000000, 2000000, 1000000, 'XE001', 'NV01'),
 ('PDV02', '2023-02-15', 2500000, 1500000, 1000000, 'XE002', 'NV02'),
 ('PDV03', '2023-03-10', 4000000, 2500000, 1500000, 'XE003', 'NV03'),

```
('PDV04', '2023-04-20', 3500000, 2000000, 1500000, 'XE004', 'NV04'),  
('PDV05', '2023-05-25', 4500000, 3000000, 1500000, 'XE005', 'NV05');
```

```
INSERT INTO DichVu (MaDV, TenDV, GiaDV, MaLoaiDV) VALUES  
('DV001', N'Bảo dưỡng định kỳ 1', 1000000, 'DV01'),  
('DV002', N'Sửa chữa động cơ 1', 2000000, 'DV02'),  
('DV003', N'Sửa chữa hệ thống điện 1', 1500000, 'DV03'),  
('DV004', N'Sửa chữa hệ thống phanh 1', 1200000, 'DV04'),  
('DV005', N'Sửa chữa hệ thống lái 1', 1300000, 'DV05'),  
('DV006', N'Sửa chữa gầm xe 1', 1600000, 'DV06'),  
('DV007', N'Vệ sinh nội thất 1', 800000, 'DV07'),  
('DV008', N'Sơn sửa 1', 1800000, 'DV08'),  
('DV009', N'Thay thế phụ tùng 1', 500000, 'DV09'),  
('DV010', N'Kiểm tra khí thải 1', 700000, 'DV10');
```

```
INSERT INTO HoaDon (MaHD, NgayLapHD, TongTien,  
MaPhieuDV) VALUES  
('HD001', '2023-01-02', 3000000, 'PDV01'),  
('HD002', '2023-02-16', 2500000, 'PDV02'),  
('HD003', '2023-03-11', 4000000, 'PDV03'),  
('HD004', '2023-04-21', 3500000, 'PDV04'),  
('HD005', '2023-05-26', 4500000, 'PDV05');
```

INSERT INTO ChiTietDV (MaPhieuDV, MaDV) VALUES

('PDV01', 'DV001'),

('PDV01', 'DV002'),

('PDV02', 'DV003'),

('PDV02', 'DV004'),

('PDV03', 'DV005'),

('PDV03', 'DV006'),

('PDV04', 'DV007'),

('PDV04', 'DV008'),

('PDV05', 'DV009'),

('PDV05', 'DV010');

INSERT INTO PhieuNK (MaPhieuNK, NgayLapPhieu, TongTien,
MaNV) VALUES

('PNK01', '2023-01-15', 5000000, 'NV01'),

('PNK02', '2023-02-20', 3500000, 'NV02'),

('PNK03', '2023-03-18', 4200000, 'NV03'),

('PNK04', '2023-04-10', 2700000, 'NV04'),

('PNK05', '2023-05-30', 6000000, 'NV05');

INSERT INTO ChiTietPhieuNhap (NgayNhapKho, SoLuongNhap,
MaPhieuNK, MaPT) VALUES

('2023-01-15', 50, 'PNK01', 'PT001'),

('2023-01-15', 30, 'PNK01', 'PT002'),
 ('2023-02-20', 20, 'PNK02', 'PT003'),
 ('2023-02-20', 15, 'PNK02', 'PT004'),
 ('2023-03-18', 10, 'PNK03', 'PT005'),
 ('2023-03-18', 20, 'PNK03', 'PT001'),
 ('2023-04-10', 25, 'PNK04', 'PT002'),
 ('2023-04-10', 30, 'PNK04', 'PT003'),
 ('2023-05-30', 40, 'PNK05', 'PT004'),
 ('2023-05-30', 25, 'PNK05', 'PT005');

INSERT INTO Phụ_tùng (MaPT, TenPT, GiaPT, SoLuongTK,
 DonViTinh, MaLoaiPT, MaKho) VALUES
 ('PT001', 'Lốp xe ô tô', 2000000, 30, 'Cái', 'PT01', 'KH01'),
 ('PT002', 'Ắc quy ô tô', 1200000, 20, 'Cái', 'PT02', 'KH02'),
 ('PT003', 'Bugi xe máy', 50000, 100, 'Cái', 'PT03', 'KH03'),
 ('PT004', 'Dầu nhớt động cơ', 150000, 50, 'Lít', 'PT04', 'KH01'),
 ('PT005', 'Bộ lọc gió xe', 80000, 40, 'Cái', 'PT05', 'KH02'),
 ('PT006', 'Bộ lọc dầu xe', 100000, 60, 'Cái', 'PT06', 'KH03'),
 ('PT007', 'Cùm thắng xe', 400000, 15, 'Cái', 'PT07', 'KH01'),
 ('PT008', 'Đĩa thắng xe', 600000, 10, 'Cái', 'PT08', 'KH02'),
 ('PT009', 'Bô pô xe', 2000000, 25, 'Cái', 'PT09', 'KH03'),
 ('PT010', 'Gương chiếu hậu xe', 150000, 30, 'Cái', 'PT10', 'KH01');


```

INSERT INTO ChiTietPT (soluong, MaPhieuDV, MaPT) VALUES

(2, 'PDV01', 'PT001'),

(1, 'PDV01', 'PT002'),

(3, 'PDV02', 'PT003'),

(4, 'PDV02', 'PT004'),

(2, 'PDV03', 'PT005'),

(1, 'PDV03', 'PT006'),

(1, 'PDV04', 'PT007'),

(3, 'PDV04', 'PT008'),

(2, 'PDV05', 'PT009'),

(1, 'PDV05', 'PT010');

```

4.3. Cài đặt các ràng buộc toàn vẹn

4.3.1. Ràng buộc toàn vẹn với bảng khách hàng và bảng xe

“Giới tính của khách hàng, nhân viên chỉ có thể là nam hoặc nữ”

- Ví dụ vi phạm:

The screenshot shows a SQL Server window with the following SQL code for a trigger named `trg_PhaiNV` on the `NhanVien` table:

```

create trigger trg_PhaiNV
on NhanVien
for insert, update
as
begin
    declare @Phai varchar(10)
    select @Phai = Phai from inserted
    if @Phai <> N'Nam' and @Phai <> N'Nữ'
    begin
        raiserror('Giới tính của nhân viên phải là Nam hoặc Nữ',16,1)
        rollback tran
    end
end
insert into NhanVien
values('NV06', N'Phạm Duy Hiếu', 'duyhiu@gara.com', N'Hồ Chí Minh',
N'Quận 1', N'Đường Thống Nhất', N'672', N'Kế toán viên', N'Chưa Chắc', '1990-01-01')

```

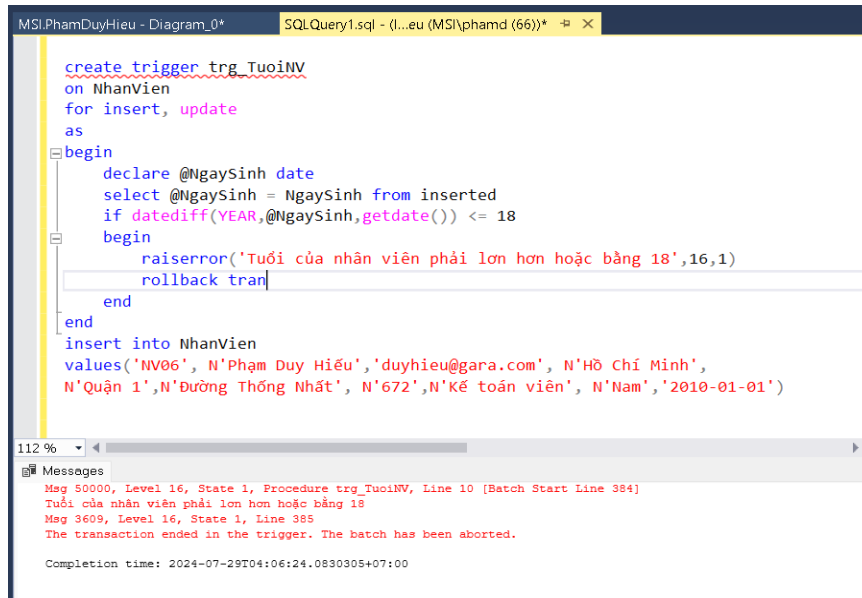
Below the code, the Messages pane displays the following error message:

```

Msg 50000, Level 16, State 1, Procedure trg_PhaiNV, Line 10 [Batch Start Line 382]
Giới tính của nhân viên phải là Nam hoặc Nữ
Msg 3609, Level 16, State 1, Line 383
The transaction ended in the trigger. The batch has been aborted.
Completion time: 2024-07-29T03:53:54.8731741+07:00

```

“ Tuổi của khách hàng và nhân viên phải ≥ 18 tuổi”



```
create trigger trg_TuoiNV
on NhanVien
for insert, update
as
begin
    declare @NgaySinh date
    select @NgaySinh = NgaySinh from inserted
    if datediff(YEAR,@NgaySinh,getdate()) <= 18
    begin
        raiserror('Tuổi của nhân viên phải lớn hơn hoặc bằng 18',16,1)
        rollback tran
    end
end
insert into NhanVien
values('NV06', 'Phạm Duy Hiếu', 'duyhieu@gara.com', 'Hồ Chí Minh',
      'Quận 1', 'Đường Thống Nhất', '672', 'Kế toán viên', 'Nam', '2010-01-01')
```

Messages

Msg 50000, Level 16, State 1, Procedure trg_TuoiNV, Line 10 [Batch Start Line 384]
Tuổi của nhân viên phải lớn hơn hoặc bằng 18
Msg 3609, Level 16, State 1, Line 385
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2024-07-29T04:06:24.0830305+07:00

- Dữ liệu bảng Nhân viên:

	MaNV	TenNV	Email	ThanhPho	Quan	Duong	SoNha	ChucVu	Phai	NgaySinh
1	NV01	Nguyễn Văn A	nguyenvana@gara.com	H? Chí Minh	Qu?n 1	Đu?ng Lê L?i	123	K? thu? viên	Nam	1990-01-01
2	NV02	Tr?n Th? B	tranthib@gara.com	H? Chí Minh	Qu?n 2	Đu?ng Nguyễn Văn Linh	456	L? t?n	N?	1992-03-15
3	NV03	Lê Văn C	levanc@gara.com	H? Chí Minh	Qu?n 3	Đu?ng Võ Th? S?u	789	Giám d?c	Nam	1985-07-20
4	NV04	Ph?m Th? D	phamthid@gara.com	H? Chí Minh	Qu?n 4	Đu?ng Đ?ng Kh?i	101	K? toán	N?	1988-05-10
5	NV05	Võ Văn E	vovane@gara.com	H? Chí Minh	Qu?n 5	Đu?ng Tr?n Hưng Đ?o	111	Th? s?a ch?a	Nam	1995-09-25

“Email của khách hàng và nhân viên không được trùng nhau”

- Ví dụ vi phạm:

```

create trigger trg_EmailNV
on NhanVien
for insert, update
as
begin
    declare @Email varchar(100), @Count int
    select @Email = Email from inserted
    select @Count = count(Email) from NhanVien where Email = @Email
    if @Count > 1
    begin
        raiserror('Email của nhân viên không được trùng nhau',16,1)
        rollback tran
    end
end
insert into NhanVien
values('NV06', N'Phạm Duy Hiếu', 'nguyenvana@gara.com', N'Hồ Chí Minh',
N'Quận 1', N'Đường Thống Nhất', N'672', N'Kế toán viên', N'Nam', '2004-01-01')
select * from NhanVien

```

Messages

sg 50000, Level 16, State 1, Procedure trg_EmailNV, Line 11 [Batch Start Line 385]
mail của nhân viên không được trùng nhau
sg 3609, Level 16, State 1, Line 386
the transaction ended in the trigger. The batch has been aborted.

Completion time: 2024-07-29T04:15:15.7866358+07:00

4.3.2. Ràng buộc toàn vẹn với bảng Xe, LoaiXe

“Khi thêm hoặc sửa dữ liệu trong bảng Xe phải kiểm tra xem có MaLoaixe trong bảng LoaiXe hay không ”

```

Create trigger trg_ThemSuaXe
on Xe
for insert, update
as
begin
    declare @MaLoaiXe char(5), @Count int
    select @MaLoaiXe = MaLoaiXe from inserted
    select @Count = count(*) from LoaiXe where MaLoaiXe = @MaLoaiXe
    if @Count < 1
    begin
        raiserror('Dữ liệu sai',16,1)
        rollback
    end
end
Insert into Xe
values('XE006', '51B-67890', 'BMW', '2020', 'Kh001', 'LX01')
select * from LoaiXe
select * from Xe

```

“khi xóa dữ liệu trong bảng loại xe phải kiểm tra có mã loại xe trong bảng xe hay không”

```

Create trigger trg_XoaloaiXe
on LoaiXe
for delete
as
begin
    declare @MaLoaiXe char(5), @Count int
    select @MaLoaiXe = MaLoaiXe from deleted
    select @Count = count(*) from Xe where MaLoaiXe = @MaLoaiXe
    if @Count < 1
    begin
        raiserror('Dữ liệu sai',16,1)
        rollback
    end
end
end

```

4.3.3. Ràng buộc toàn vẹn với bảng DichVu, LoaiDV

“Khi thêm dịch vụ tên dịch vụ không được trùng”

```

Create trigger trg_ThemSuaDV
on DichVu
for insert, update
as
begin
    declare @TenDV char(5), @Count int
    select @TenDV = TenDV from inserted
    select @Count = Count(*) from DichVu where TenDV = @TenDV
    if @Count > 1
    begin
        raiserror('Tên dịch vụ không được trùng nhau',16,1)
        rollback
    end
end
end

```