



TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN HỌC KỲ II
NĂM HỌC 2021-2022

MÔN: CHUYÊN ĐỀ ORACLE
QUẢN LÝ CHUỖI CỦA HÀNG
THỨC ĂN NHANH

Giảng viên hướng dẫn: Th.s Trần Thị Thanh Thảo

Sinh viên thực hiện:

Nguyễn Văn Bình Minh – 19DH110596

Phạm Gia Khương – 19DH110076

Trần Văn Minh – 19DH110060

Thành phố Hồ Chí Minh, tháng 6 năm 2022

MỤC LỤC

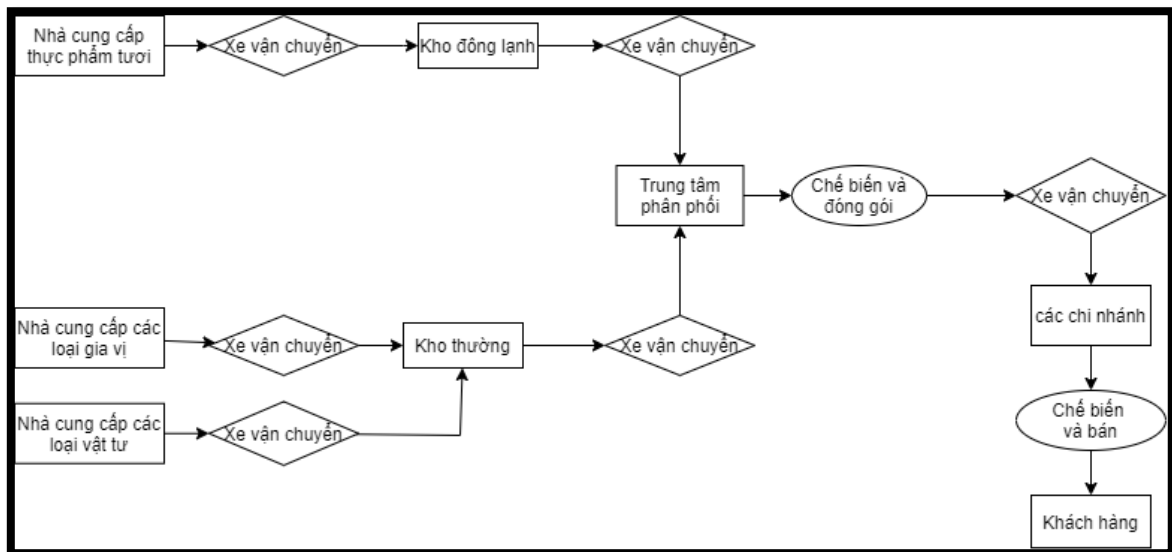
I. Tổng quan đề tài:	1
1. Mô hình chuỗi cung ứng của cửa hàng thức ăn nhanh:	1
2. Phân tích mô hình hoạt động chuỗi cung ứng thức ăn nhanh:	2
3. Thực thể cần quản lý:	4
4. Phân tích các thực thể thành bảng và chuyển về dạng chuẩn:	8
5. Biểu mẫu:	21
6. Các quy trình:	24
7. Thực thể cần quản lý:	28
II. Thiết kế cơ sở dữ liệu:	29
1. Sơ đồ ERD:	29
2. Mô tả bảng:	30
3. Quy định tạo mã:	42
III. Cài đặt:	44
1. Thiết lập dữ liệu và các ràng buộc:	44
2. Các câu truy vấn:	57
3. View hiện dữ liệu:	60
4. Stored Procedure	66
5. Hàm (function):	96
6. Trigger:	98
7. Tài khoản người dùng:	108
8. Cấp quyền sử dụng người dùng:	109
9. Cấm/thu hồi sử dụng:	110
10. Backup & Restore:	111
IV. Tổng kết:	114

MỤC LỤC HÌNH ẢNH

Hình 1: Mô hình chuỗi cung ứng	1
Hình 2: Biểu mẫu phiếu nhập kho.....	21
Hình 3: Biểu mẫu phiếu xuất kho.....	21
Hình 4: Hóa đơn thanh toán	22
Hình 5: Chi tiết lô hàng	22
Hình 6: Tồn kho.....	23
Hình 7: Chi tiết tồn kho	23
Hình 8: Quy trình cung cấp nguyên liệu	24
Hình 9: Quy trình phân phối nguyên liệu.....	24
Hình 10: Quy trình nhập hàng chi nhánh	25
Hình 11: Quy trình tạo hóa đơn.....	25
Hình 12: Quy trình nhập hàng	26
Hình 13: Quy trình xuất hàng	26
Hình 14: Quy trình kiểm tra tồn kho	27
Hình 15: Thực thể cần quản lý	28
Hình 16: Sơ đồ ERD.....	29

I. Tổng quan đề tài:

1. Mô hình chuỗi cung ứng của cửa hàng thức ăn nhanh:



Hình 1: Mô hình chuỗi cung ứng

2. Phân tích mô hình hoạt động chuỗi cung ứng thức ăn nhanh:

- Nhà cung cấp thực phẩm tươi:
 - Nhận yêu cầu từ kho
 - Sơ chế sản phẩm
 - Đóng gói sản phẩm và tạo mã sản phẩm
 - Đóng sản phẩm vào thùng và tạo mã cho thùng khi có yêu cầu nhập hàng
 - Mỗi thùng chứa 1 loại sản phẩm (VD: Thùng TH001 chứa gà loại A1)
 - Các thùng được giao cho xe vận chuyển theo lô đến các kho đông lạnh theo tuyến hàng.
 - Phải giao đủ số sản phẩm trong thời gian kho yêu cầu
- Nhà cung cấp các loại gia vị:
 - Nhận yêu cầu từ kho
 - Chuẩn bị các gia vị theo công thức
 - Đóng gói gia vị và tạo mã các gói
 - Đóng các gói gia vị vào thùng và tạo mã cho các thùng khi có yêu cầu nhập hàng
 - Mỗi thùng chứa 1 loại gia vị (VD: Thùng TH401 chứa gia vị gà chiên giòn C1)
 - Các thùng được giao cho xe vận chuyển theo lô đến các kho thường theo tuyến hàng.
 - Phải giao đủ số sản phẩm trong thời gian kho yêu cầu
- Nhà cung cấp các loại vật tư:
 - Nhận yêu cầu từ kho
 - Sản xuất hoặc nhập vật tư
 - Tạo mã cho mỗi vật tư
 - Chuyển các vật tư vào các xe đến kho khi có yêu cầu nhập hàng.
 - Phải giao đủ số sản phẩm trong thời gian kho yêu cầu
- Kho đông lạnh:
 - Theo dõi tình trạng hàng tồn
 - Nếu hàng tồn dưới mức quy định thì phải yêu cầu nhập hàng
 - Yêu cầu nhập hàng phải trong thời gian quy định
 - Kiểm tra hàng nhập, tình trạng các thùng hàng
 - Nếu thùng hàng có hư hỏng phải đổi trả với nhà cung cấp.
 - Lập phiếu nhập hàng
 - Xuất hàng nếu trung tâm phân phối yêu cầu
 - Lập các phiếu xuất hàng và kiểm kê lại kho
 - Phải xuất hàng trong thời gian trung tâm phân phối yêu cầu
 - Hàng trong kho chỉ được lưu kho trong thời gian quy định

- Kho thường:
 - Theo dõi tình trạng hàng tồn
 - Nếu hàng tồn dưới mức quy định thì phải yêu cầu nhập hàng
 - Yêu cầu nhập hàng phải trong thời gian quy định
 - Kiểm tra hàng nhập, tình trạng các thùng hàng
 - Nếu thùng hàng có hư hỏng phải đổi trả với nhà cung cấp.
 - Lập phiếu nhập hàng
 - Xuất hàng nếu trung tâm phân phối yêu cầu
 - Lập các phiếu xuất hàng và kiểm kê lại kho
 - Phải xuất hàng trong thời gian trung tâm phân phối yêu cầu
- Trung tâm phân phối:
 - Kiểm tra tình trạng hàng nhập từ các kho.
 - Nếu hàng nhập có hư hỏng phải yêu cầu đổi trả
 - Mở thùng và đóng gói các sản phẩm theo yêu cầu của các chi nhánh.
 - Các gói sản phẩm được đưa vào thùng và tạo mã cho từng thùng (VD: Thùng TH001 có 100 sản phẩm và TH401 có 100 sản phẩm được mở và phân phối vào 10 thùng mới mỗi thùng chứa 20 sản phẩm từ TH001 và TH401. Đóng gói 10 thùng và tạo mã cho 10 thùng đó.)
 - Phân phối các thùng sản phẩm theo yêu cầu các chi nhánh.
 - Phải phân phối các hàng hóa trong thời gian các chi nhánh yêu cầu
- Các chi nhánh:
 - Kiểm tra hàng nhập từ trung tâm phân phối
 - Nếu hàng nhập có hư hỏng phải yêu cầu đổi trả
 - Kiểm tra tình trạng hàng tồn theo danh mục sản phẩm
 - Gửi yêu cầu phân phối hàng nếu hàng hóa tồn dưới mức cho phép
 - Nhận yêu cầu sản phẩm từ khách hàng
 - Chuẩn bị sản phẩm
 - Đổi trả sản phẩm nếu có lỗi khi chuẩn bị
- Khách hàng:
 - Yêu cầu sản phẩm từ chi nhánh
 - Lưu thông tin khách hàng thân thiết
 - Ghi nhận điểm của khách hàng
 - Áp dụng ưu đãi cho khách hàng thân thiết
 - Lập hóa đơn giao dịch
 - Phản hồi về chi nhánh

3. Thực thể cần quản lý:

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Cơ sở	Mã cơ sở	
	Tên cơ sở	
	Địa chỉ	
	Tình trạng	“Hoạt động”, “Ngưng hoạt động”
	Loại cơ sở	“Kho”, “Kho lạnh”, “Kho trung tâm phân phối”, “Chi nhánh”

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Nhân viên	Mã nhân viên	
	Chứng minh nhân dân	
	Họ lót	
	Tên nhân viên	
	Tuổi	
	Giới tính	
	Số điện thoại	
	Chức vụ	“Nhân viên”, “Quản lý”

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Chi tiết nhân viên	Mã nhân viên	
	Ca	
	Mã nhà quản lý	
	Mã cơ sở	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Tài xế	Mã tài xế	
	Tên tài xế	
	Ca	
	Tuyến	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Xe vận chuyển	Số xe	
	Loại xe	
	Hàng	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Phân công	Số xe	
	Mã tài xe	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Nhà cung cấp	Mã nhà cung cấp	
	Tên nhà cung cấp	
	Địa chỉ	
	Số điện thoại	
	Tình trạng	“Hoạt động”, “Ngưng hoạt động”

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Lô hàng	Mã lô	
	Tổng số thùng	
	Số xe	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Thùng	Mã thùng	
	Tình trạng	“Nguyên kiện”, “Đã mở”, “Hỏng”

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Đơn vị tính	Mã đơn vị tính	
	Tên đơn vị tính	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Loại nguyên liệu	Mã loại nguyên liệu	
	Tên loại nguyên liệu	
	Mã nhà cung cấp	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Nguyên liệu	Mã nguyên liệu	
	Tên nguyên liệu	
	Đơn giá	
	Mã loại nguyên liệu	
	Mã đơn vị tính	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Loại sản phẩm	Mã loại sản phẩm	
	Tên loại sản phẩm	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Sản phẩm	Mã sản phẩm	
	Tên sản phẩm	
	Mã loại sản phẩm	
	Mã đơn vị tính	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Chi tiết sản phẩm	Mã sản phẩm	
	Mã nguyên liệu	
	Số lượng	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Biến động giá	Mã sản phẩm	
	Mã nhân viên	
	Đơn giá	
	Ngày cập nhật	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Khách hàng	Mã khách hàng	
	Số điện thoại	
	Họ lót	
	Tên khách hàng	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Chi tiết lô hàng	Mã lô	
	Mã thùng	
	Địa chỉ	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Chi tiết thùng	Mã thùng	
	Mã nguyên liệu	
	Tổng số nguyên liệu	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Phiếu nhập	Mã phiếu nhập	
	Ngày lập phiếu	
	Ngày nhập hàng	
	Mã nhân viên	
	Mã cơ sở nhập	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Phiếu xuất	Mã phiếu xuất	
	Ngày lập phiếu	
	Ngày xuất hàng	
	Mã nhân viên	
	Mã cơ sở nhập	
	Mã cơ sở xuất	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Chi tiết phiếu nhập	Mã phiếu nhập	
	Mã lô	
	Thời hạn	Nhỏ hơn hoặc bằng 5 ngày

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Chi tiết phiếu xuất	Mã phiếu xuất	
	Mã lô	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Hóa đơn	Số hóa đơn	
	Ngày lập hóa đơn	
	Mã nhân viên	
	Mã khách hàng	

Tên thực thể	Thông tin lưu trữ	Ràng buộc
Chi tiết hóa đơn	Số hóa đơn	
	Mã sản phẩm	
	Số lượng	Lớn hơn hoặc bằng 1 sản phẩm
	Thành tiền	Lớn hơn hoặc bằng 1000 đồng

4. Phân tích các thực thể thành bảng và chuyển về dạng chuẩn:

Phân tích các thực thể thành bảng và chuyển về dạng chuẩn:

❖ Nhà cung cấp

Mã nhà cung cấp	Tên nhà cung cấp	Địa chỉ	Số điện thoại	Tình trạng
NCC01	KMM Farm	83 Tô Hiến Thành	90002213	Hoạt động
NCC02	MKH Spice	182 Lê Lợi	94567812	Hoạt động
NCC03	TH Chair	892/1 Thái Bình	96782213	Ngưng hoạt động

- Nhà cung cấp(Mã nhà cung cấp, tên nhà cung cấp, địa chỉ, số điện thoại, tình trạng)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã nhà cung cấp].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa.

❖ Cơ sở

Mã cơ sở	Tên cơ sở	Địa chỉ	Tình trạng	Loại cơ sở
KH01	KMM ware	123 Lạc Long Quân	Hoạt động	Kho lạnh
KH02	MKH ware	16 Cộng Hòa	Hoạt động	Kho thương
PP03	TH Distribute	27/12 Tỉnh lộ 10	Ngừng	Phân phối
CN04	KMM Store	183 Tân Tạo	Hoạt động	Chi nhánh
KH05				

- Kho vận(Mã cơ sở, tên cơ sở, địa chỉ, tình trạng, loại cơ sở)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã cơ sở].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa.

❖ Xe vận chuyển

Số xe	Mã tài xế	Tên tài xế	Ca chạy	Tuyến chạy
XT0102 H3	NVTX011	Nguyễn Kim Anh	Ca 1	SG 001
XT0172 H3	NVTX012	Nguyễn Văn Ba	Ca 2	SG 003
XT0673 H3	NVTX013	Trần Thị Xu	Ca 1	SG 001
XT9721 H4	NVTX011	Nguyễn Kim Anh	Ca 3	SG 004
XT2132 H4	NVTX015	Lê Xu Ca	Ca 5	SG 002
XT9298 H8	NVTX016	Hoàng Văn Minh	Ca 1	SG 003
XT2224 H8	NVTX017	Thái Công Tiến	Ca 2	SG 001

- Xe vận chuyển(Số xe, Mã tài xế, tên tài xế, ca chạy, tuyến)
- Bảng đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đạt dạng chuẩn 2 do các thuộc tính không khóa phụ thuộc hàm đầy đủ vào khóa [Số xe]
- Bảng chưa đạt chuẩn 3 do có phụ thuộc hàm bất cầu [Mã tài xế] với [Tên tài xế], [Ca chạy] và [Tuyến chạy]
- Tách bảng thành 2 bảng là : Xe vận chuyển(Số xe, Mã tài xế) và Tài xế(Mã tài xế, tên tài xế, ca chạy, tuyến) với Mã tài xế là khóa chính tham chiếu khóa ngoại vào bảng Xe vận chuyển
- Xe vận chuyển:

Số xe	Mã tài xế
XT0102 H3	NVTX011
XT0172 H3	NVTX012
XT0673 H3	NVTX013
XT9721 H4	NVTX011
XT2132 H4	NVTX015
XT9298 H8	NVTX016
XT2224 H8	NVTX017

- Xe vận chuyển (Số xe, mã tài xế)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Số xe].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bất cầu vào các thuộc tính không khóa.

❖ Tài xế:

<u>Mã tài xế</u>	Tên tài xế	Ca chạy	Tuyến chạy
NVTX011	Nguyễn Kim Anh	Ca 1	SG 001
NVTX012	Nguyễn Văn Ba	Ca 2	SG 003
NVTX013	Trần Thị Xu	Ca 1	SG 001
NVTX011	Nguyễn Kim Anh	Ca 3	SG 004
NVTX015	Lê Xu Ca	Ca 5	SG 002
NVTX016	Hoàng Văn Minh	Ca 1	SG 003
NVTX017	Thái Công Tiến	Ca 2	SG 001

- Tài xế (Mã tài xế, Tên tài xế, Ca chạy, Tuyến chạy)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã tài xế].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa.

❖ Khách hàng

<u>Số điện thoại</u>	Họ	Tên	Tuổi	Giới tính	Điểm tích lũy	Xếp hạng
9213783210	Nguyễn Văn Bình	Minh	24	Nam	1000	Bạc
2137192831	Trần Văn	Minh	12	Nam	400	Đồng
2331331219	Phạm Gia	Khuong	31	Nam	200	Đồng
2313812919	Đỗ Thị Xuân	Mai	21	Nữ	3000	Bạc
3282193890	Cao Tuấn	Kiệt	42	Nam	5000	Vàng

- Khách hàng (Số điện thoại, Họ, Tên, Tuổi, Giới tính, Điểm tích lũy, Xếp hạng)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã tài xế].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa.

❖ Nhân viên

Số CMND	MãNV	Họ lót	Tên	Tuổi	Giới tính	Địa chỉ	Email	Số điện thoại	Ngày vào làm	Ca làm	Mã người quản lý	Chức vụ	Bộ phận	Trực thuộc
DX11043422	NV001	Nguyễn Thị	Tú	31	Nữ	194 Tô Hiến Thành	ntu@email.com	9213783210	23/9/2013	Ca 1	NULL	Quản lý	Nhập xuất	KH01
DX14672484	NV002	Trần Văn	Thịnh	23	Nam	18/2 Lê Lợi	nvt@email.com	2137192831	2/8/2011	Ca 2	NULL	Quản lý	Phân phối	PP02
DX21932782	NV003	Thái Trung	Sơn	21	Nam	89 Thái Bình	tts@email.com	2331331219	5/1/2021	Ca 1	NV001	Thư ký	Kiểm kê	KH01
DX02817282	NV004	Trần Văn	Tỉnh	20	Nam	183 Hương lộ 2	tvt@email.com	2313812919	16/12/2012	Ca 2	NV007	Nhân viên	Quầy	CN001

- Nhân viên(MãNV, Số CMND, Họ lót, Tên, Tuổi, Giới tính, Địa chỉ, Email, Số điện thoại, Ngày vào làm, Ca làm, Mã người quản lý, Chức vụ, Bộ Phận, Mã cơ sở)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng chưa đạt dạng chuẩn 2NF do các thuộc tính chưa phụ thuộc đầy đủ vào khóa [MãNV] cần tách thành 2 bảng.
- Bảng Nhân viên:

Số CMND	MãNV	Họ lót	Tên	Tuổi	Giới tính	Địa chỉ	Email	Số điện thoại
DX11043422	NV001	Nguyễn Thị	Tú	31	Nữ	194 Tô Hiến Thành	ntu@email.com	9213783210
DX14672484	NV002	Trần Văn	Thịnh	23	Nam	18/2 Lê Lợi	nvt@email.com	2137192831
DX21932782	NV003	Thái Trung	Sơn	21	Nam	89 Thái Bình	tts@email.com	2331331219
DX02817282	NV004	Trần Văn	Tỉnh	20	Nam	183 Hương lộ 2	tvt@email.com	2313812919

- Nhân viên(MãNV, Số CMND, Họ lót, Tên, Tuổi, Giới tính, Địa chỉ, Email, Số điện thoại, Ngày vào làm, Ca làm, Mã người quản lý, Chức vụ, Bộ Phận, Trục thuộc)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [MãNV].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa.

- Bảng Chi tiết nhân viên:

<u>MãNV</u>	Ngày vào làm	Ca làm	Mã người quản lý	Chức vụ	Bộ phận	Mã cơ sở
NV001	23/9/2013	Ca 1	NULL	Quản lý	Nhập xuất	KH01
NV002	2/8/2011	Ca 2	NULL	Quản lý	Phân phối	PP02
NV003	5/1/2021	Ca 1	NV001	Thư ký	Kiểm kê	KH01
NV004	16/12/2012	Ca 2	NV007	Nhân viên	Quầy	CN001

- Chi tiết nhân viên(MãNV, Ngày vào làm, Ca làm, Mã người quản lý, Chức vụ, Bộ Phận, Trục thuộc)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [MãNV].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa

❖ Tồn kho

<u>Mã phiếu nhập</u>	Tháng tồn	Năm tồn	Mã lô	Thời hạn
PN001	10	2021	LO001	1 tháng
PN002	11	2021	LO002	3 tháng
PN003	11	2021	LO003	4 tháng

- Tồn kho(Mã phiếu nhập, Tháng tồn, năm tồn, mã lô, thời hạn)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã phiếu nhập].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa.

❖ Phiếu nhập xuất

<u>Mã phiếu nhập</u>	Loại phiếu	Ngày lập phiếu	Mã nhân viên	Mã lô	Tổng số thùng	Cơ sở nhập	Cơ sở xuất
PN001	Nhập	20/10/2021	NV001	LO001	3	KH01	Null
PN002	Xuất	20/11/2021	NV003	LO002	2	CN002	PP02
PN003	Nhập	20/11/2021	NV001	LO003	3	CN001	PP03
PN001	Nhập	20/10/2021	NV001	LO004	4	KH01	Null

- Phiếu nhập (Mã phiếu nhập, Loại phiếu, Ngày lập phiếu, Mã nhân viên, Mã lô, Tổng số thùng, Cơ sở nhập, Cơ sở xuất)
- Bảng chưa đạt dạng chuẩn 1NF do có dữ liệu trùng lặp.
- Tách 2 bảng:
 - Phiếu nhập xuất:

<u>Mã phiếu nhập</u>	Loại phiếu	Ngày lập phiếu	Mã nhân viên	Cơ sở nhập	Cơ sở xuất
PN001	Nhập	20/10/2021	NV001	KH01	Null
PN002	Xuất	20/11/2021	NV003	CN002	PP02
PN003	Nhập	20/11/2021	NV001	CN001	PP03

- Phiếu nhập (Mã phiếu nhập, Loại phiếu, Ngày lập phiếu, Mã nhân viên, Cơ sở nhập, Cơ sở xuất)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã phiếu nhập].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bất cầu vào các thuộc tính không khóa
- Chi tiết phiếu:

<u>Mã phiếu nhập</u>	Mã lô	Tổng số thùng
PN001	LO001	3
PN002	LO002	2
PN003	LO003	3
PN001	LO004	4

- (Mã phiếu nhập, Mã lô, Tổng số thùng)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã phiếu nhập].

- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa

❖ Lô hàng

Mã lô	Mã thùng	Tổng số thùng	Số xe
LO001	TH001	3	XT0102 H3
LO001	TH002	3	XT0102 H3
LO001	TH003	3	XT0102 H3
LO002	TH004	2	XT0172 H3
LO002	TH005	2	XT0172 H3

- Lô hàng(Mã lô, Mã thùng, Tổng số thùng, số xe)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã phiếu nhập].
- Bảng chưa đạt dạng chuẩn 3NF do có phụ thuộc hàm [Mã thùng] phụ thuộc bắt cầu vào các thuộc tính không khóa [Số xe].
- Tách bảng để bảng không lặp giá trị

○ Lô hàng:

Mã lô	Tổng số thùng	Số xe
LO001	3	XT0102 H3
LO002	2	XT0172 H3

- Lô hàng(Mã lô, Tổng số thùng, số xe)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã lô].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa.

- Chi tiết lô hàng:

<u>Mã lô</u>	Mã thùng
LO001	TH001
LO001	TH002
LO001	TH003
LO002	TH004
LO002	TH005

- Chi tiết lô hàng(Mã lô, Mã thùng)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã lô].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bất cầu vào các thuộc tính không khóa.

❖ Thùng

<u>Mã thùng</u>	tình trạng	Mã nguyên liệu	Tổng số nguyên liệu
TH001	Nguyên	NL002	20
TH001	Nguyên	NL021	10
TH002	Nguyên	NL650	30
TH002	Mở	NL001	10
TH002	Mở	NL002	20

- Thùng(Mã thùng, tình trạng, mã nguyên liệu, tổng số nguyên liệu)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã thùng].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bất cầu vào các thuộc tính không khóa.
- Tách bảng để các giá trị không lặp lại
 - Thùng:

<u>Mã thùng</u>	tình trạng
TH001	Nguyên
TH002	Mở

- Thùng(Mã thùng, tình trạng)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã thùng].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bất cầu vào các thuộc tính không khóa.

- Chi tiết thùng:

<u>Mã thùng</u>	Mã nguyên liệu	Tổng số nguyên liệu
TH001	NL002	20
TH001	NL021	10
TH002	NL650	30
TH002	NL001	10
TH002	NL002	20

- Chi tiết thùng(Mã thùng, Mã nguyên liệu, tổng số nguyên liệu)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã thùng].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bất cầu vào các thuộc tính không khóa.

❖ Loại nguyên liệu

<u>Mã loại nguyên liệu</u>	Tên loại nguyên liệu	Mã nhà cung cấp	Đơn vị tính
LNL001	Gà nguyên con	KMM Farm	DVT02
LNL002	Ức gà	KHK Farm	DVT03
LNL003	Đùi gà	KMM Farm	DVT03
LNL004	Cánh gà	KMM Farm	DVT03

- Loại nguyên liệu(Mã loại nguyên liệu, Tên loại nguyên liệu, mã nhà cung cấp, đơn vị tính)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã loại nguyên liệu].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bất cầu vào các thuộc tính không khóa.

❖ Nguyên liệu

<u>Mã nguyên liệu</u>	Tên nguyên liệu	Mã loại nguyên liệu	Đơn giá
NL001	Gà nguyên con A1	LNL001	20.000
NL002	Gà nguyên con A2	LNL001	40.000
NL101	Ức gà A1	LNL002	50.000
NL102	Ức gà A2	LNL002	10.000

- Nguyên liệu(Mã nguyên liệu, Tên nguyên liệu, mã loại nguyên liệu, Đơn giá)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã nguyên liệu].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa.

❖ Loại sản phẩm

<u>Mã loại sản phẩm</u>	Tên loại sản phẩm	Đơn vị tính
LSP001	Gà chiên giòn	DVT01
LSP002	Gà Nướng	DVT01
LSP003	Burger	DVT01
LSP004	Salad	DVT01

- Loại sản phẩm(Mã loại sản phẩm, Tên loại sản phẩm, đơn vị tính)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã loại sản phẩm].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa.

❖ Sản phẩm

<u>Mã sản phẩm</u>	Tên sản phẩm	Mã loại sản phẩm	Mã nguyên liệu	Số lượng
SP001	Gà chiên giòn A1	LSP001	NL001	1
SP001	Gà Chiên giòn A1	LSP001	NL021	2
SP001	Gà chiên giòn A1	LSP001	NL035	1
SP002	Gà chiên giòn A2	LSP001	NL001	1

- Sản phẩm(Mã sản phẩm, Tên sản phẩm, Mã loại sản phẩm ,mã nguyên liệu, số lượng)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã sản phẩm].

- Bảng chưa đạt dạng chuẩn 3NF do có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa.
- Tách bảng để các giá trị không lặp lại
 - Sản phẩm:

<u>Mã sản phẩm</u>	Tên sản phẩm	Mã loại sản phẩm
SP001	Gà chiên giòn A1	LSP001
SP002	Gà chiên giòn A2	LSP001

- Sản phẩm(Mã sản phẩm, Tên sản phẩm, Mã loại sản phẩm)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã sản phẩm].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa.

- Chi tiết sản phẩm

<u>Mã sản phẩm</u>	Mã nguyên liệu	Số lượng
SP001	NL001	1
SP001	NL021	2
SP001	NL035	1
SP002	NL001	1

- Chi tiết sản phẩm(Mã sản phẩm, mã nguyên liệu, số lượng)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã sản phẩm].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bắt cầu vào các thuộc tính không khóa.

❖ Biến động giá

<u>Mã sản phẩm</u>	Đơn giá	Ngày cập nhật	Mã nhân viên
SP001	45000	24/12/2021	NV001
SP002	25000	22/3/2016	NV453
SP003	35000	5/4/2020	NV001
SP004	21000	6/9/2019	NV321

- Biến động giá(Mã sản phẩm, Đơn giá, Ngày cập nhật, Mã nhân viên)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã sản phẩm].

- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bất cầu vào các thuộc tính không khóa.

❖ Đơn vị tính

<u>Mã đơn vị tính</u>	Đơn vị tính
DVT01	Phân
DVT02	Con
DVT03	Cái

- Đơn vị tính(Mã đơn vị tính, Đơn vị tính)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Mã đơn vị tính].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bất cầu vào các thuộc tính không khóa.

❖ Hóa đơn

<u>Số hóa đơn</u>	Ngày lập	Nhân viên lập	Số điện thoại khách hàng	Mã sản phẩm	Số lượng	Thành tiền	Tổng tiền
HD001	24/12/2021	NV001	9213783210	SP001	1	45000	695000
HD001	24/12/2021	NV001	9213783210	SP002	2	50000	695000
HD001	24/12/2021	NV001	9213783210	SP043	12	600000	695000
HD002	1/2/2022	NV004	2313812919	SP002	3	75000	75000

- Hóa đơn(Số hóa đơn, Ngày lập, Nhân viên lập, Số điện thoại khách hàng, Mã sản phẩm, Số lượng, Thành tiền, Tổng tiền)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Số hóa đơn].
- Bảng chưa đạt dạng chuẩn 3NF do có phụ thuộc hàm [Thành tiền] phụ thuộc bất cầu vào các thuộc tính không khóa [Mã sản phẩm].
- Tách 2 bảng
 - Hóa đơn:

<u>Số hóa đơn</u>	Ngày lập	Nhân viên lập	Số điện thoại khách hàng	Tổng tiền
HD001	24/12/2021	NV001	9213783210	695000
HD002	1/2/2022	NV004	2313812919	75000

- Hóa đơn(Số hóa đơn, Ngày lập, Nhân viên lập, Số điện thoại khách hàng, Tổng tiền)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.

- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Số hóa đơn].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bất cầu vào các thuộc tính không khóa.

○ Chi tiết hóa đơn:

<u>Số hóa đơn</u>	Mã sản phẩm	Số lượng	Thành tiền
HD001	SP001	1	45000
HD001	SP002	2	50000
HD001	SP043	12	600000
HD002	SP002	3	75000

- Chi tiết hóa đơn(Số hóa đơn Mã sản phẩm, Số lượng, Thành tiền)
- Bảng đã đạt dạng chuẩn 1NF do không có dữ liệu trùng lặp, các giá trị đều là nguyên tử.
- Bảng đã đạt dạng chuẩn 2NF do các thuộc tính đều phụ thuộc đầy đủ vào khóa [Số hóa đơn].
- Bảng đã đạt dạng chuẩn 3NF do không có phụ thuộc hàm phụ thuộc bất cầu vào các thuộc tính không khóa.

5. Biểu mẫu:

PHIẾU NHẬP KHO				
Ngày.....tháng.....năm.....				
Số phiếu:.....				
- Họ tên người giao:.....- Ngày nhập hàng:.....				
- Xuất tại:.....				
- Nhập kho tại cơ sở:.....				
- Lô hàng:.....				
STT	Mã lô	Số lượng thùng	Trị giá lô	Thời hạn
Người lập phiếu (Ký, họ tên)		Người giao (Ký, họ tên)		Quản lý (Ký, họ tên)

Hình 2: Biểu mẫu phiếu nhập kho

PHIẾU XUẤT KHO				
Ngày.....tháng.....năm.....				
Số phiếu:.....				
- Họ tên người nhận:				
- Địa chỉ:				
- Xuất kho tại cơ sở:.....				
- Cơ sở yêu cầu nhập:.....				
STT	Mã lô	Số lượng thùng	Trị giá lô	Thời hạn
Người lập phiếu (Ký, họ tên)	Người nhận (Ký, họ tên)	Tài xế (Ký, họ tên)	Thủ kho (Ký, họ tên)	Quản lý (Ký, họ tên)

Hình 3: Biểu mẫu phiếu xuất kho

HÓA ĐƠN THANH TOÁN

SỐ HĐ:.....

Ngày in:.....

Nhân viên:.....

Tên khách hàng:....

SDT khách hàng:...

STT	Sản phẩm	SL	Đơn giá	Thành tiền
1				
2				

Tổng tiền:

Hình 4: Hóa đơn thanh toán

CHI TIẾT LÔ HÀNG

(Mã :)

- Tài xế:
- Ngày xuất:..... Ngày nhập:
- Thời hạn:.....
- Tổng số thùng:

Stt	Thùng	Nguyên liệu	Số lượng	Đơn vị tính	Giá trị

Người lập phiếu

(Ký, họ tên)

Thủ kho

(Ký, họ tên)

Quản lý

(Ký, họ tên)

Hình 5: Chi tiết lô hàng

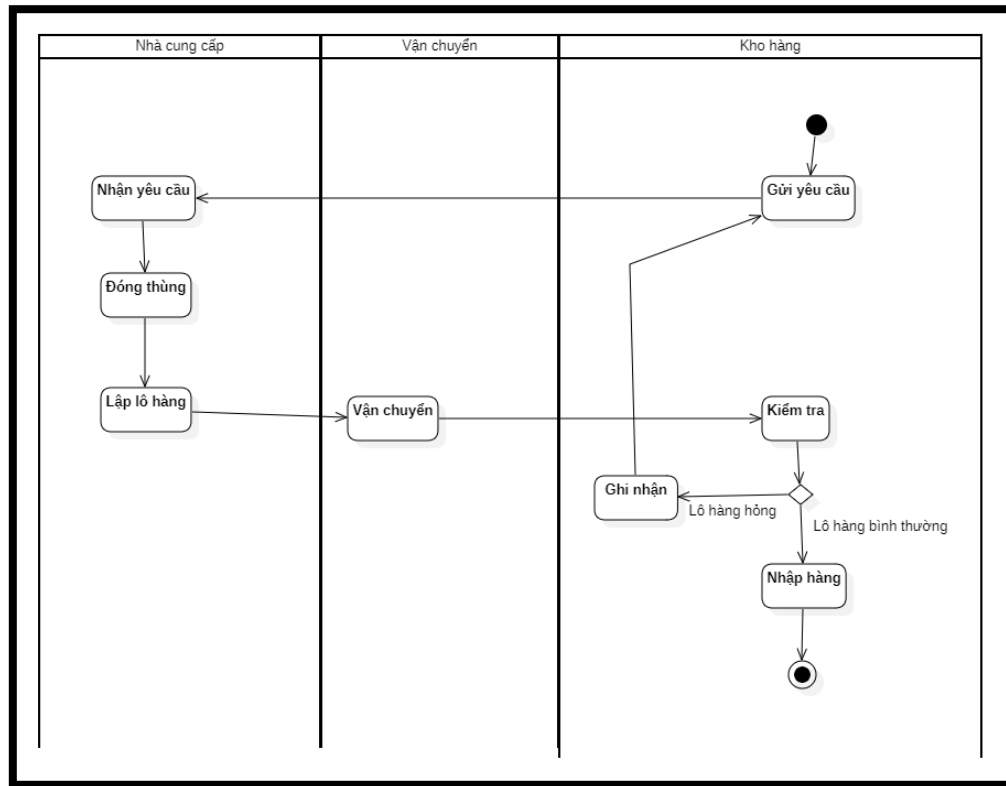
TỜ KHO					
- Cơ sở:.....					
- Ngày lập:.....					
Stt	Tháng	Năm	Nguyên liệu	Số lượng tồn	Tổng giá trị
Người lập phiếu <i>(Ký, họ tên)</i>			Thủ kho <i>(Ký, họ tên)</i>		Quản lý <i>(Ký, họ tên)</i>

Hình 6: Tờ kho

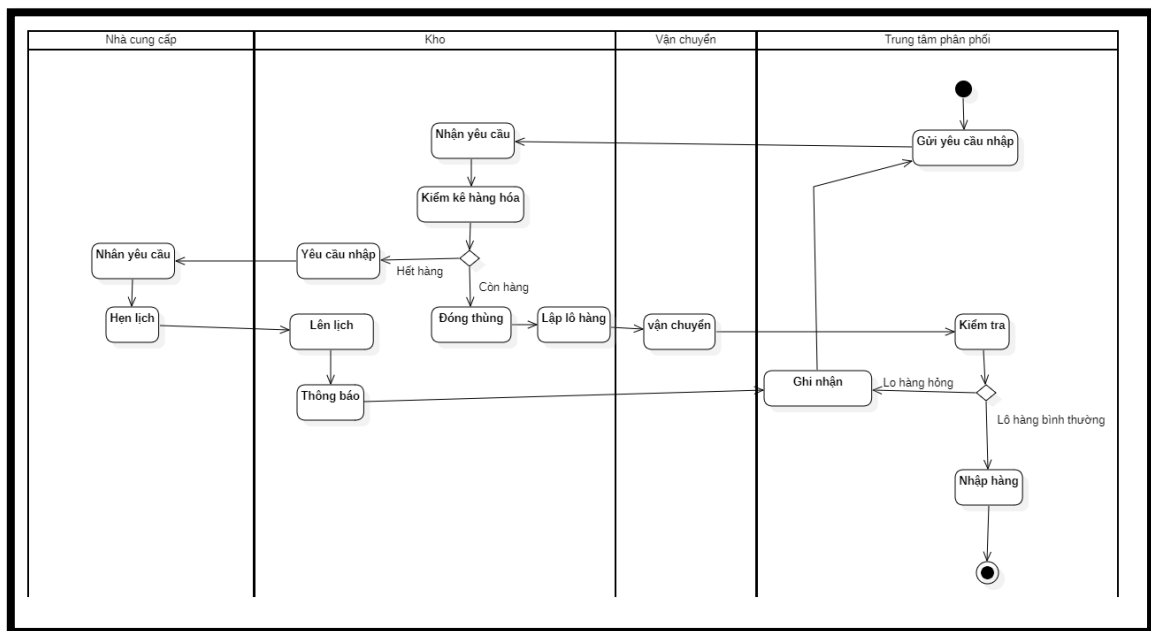
CHI TIẾT TỜ KHO					
- Cơ sở:.....					
- Ngày lập:.....					
- Tổng số lô tồn:.....					
Stt	Lô	Số lượng thùng	Thời hạn	Tình trạng	Tổng giá trị
Người lập phiếu <i>(Ký, họ tên)</i>			Thủ kho <i>(Ký, họ tên)</i>		Quản lý <i>(Ký, họ tên)</i>

Hình 7: Chi tiết tờ kho

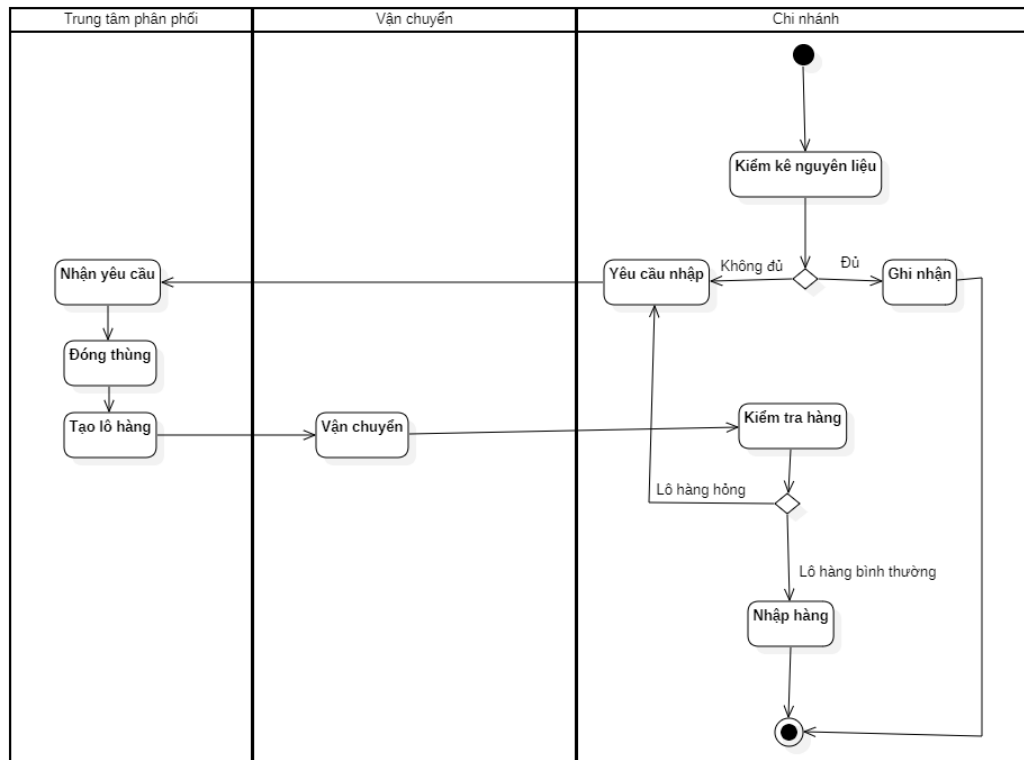
6. Các quy trình:



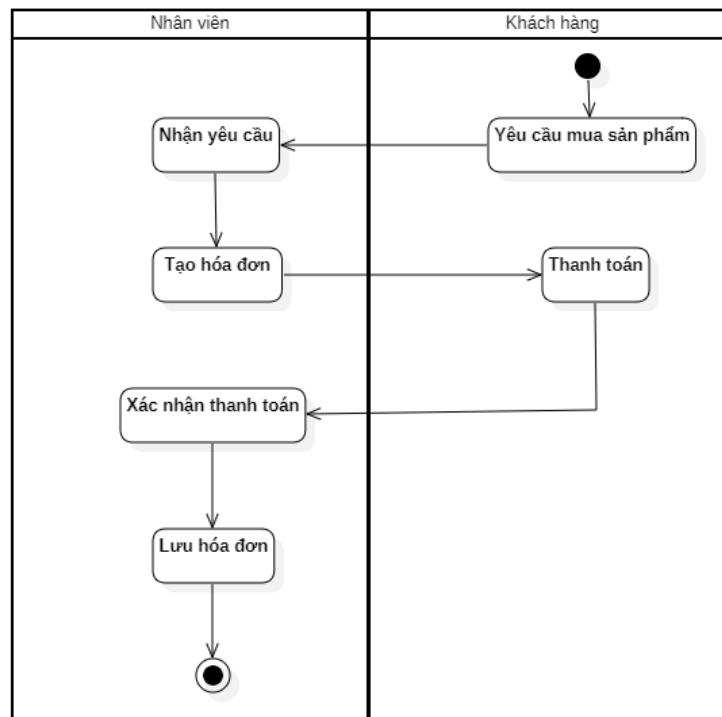
Hình 8: Quy trình cung cấp nguyên liệu



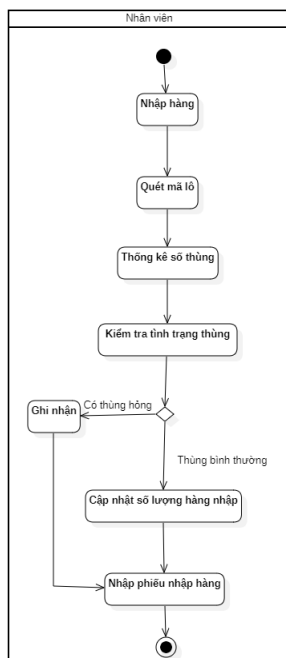
Hình 9: Quy trình phân phối nguyên liệu



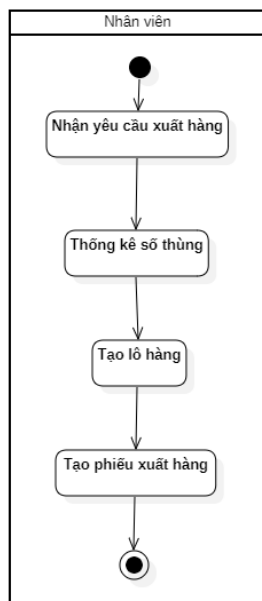
Hình 10: Quy trình nhập hàng chi nhánh



Hình 11: Quy trình tạo hóa đơn



Hình 12: Quy trình nhập hàng

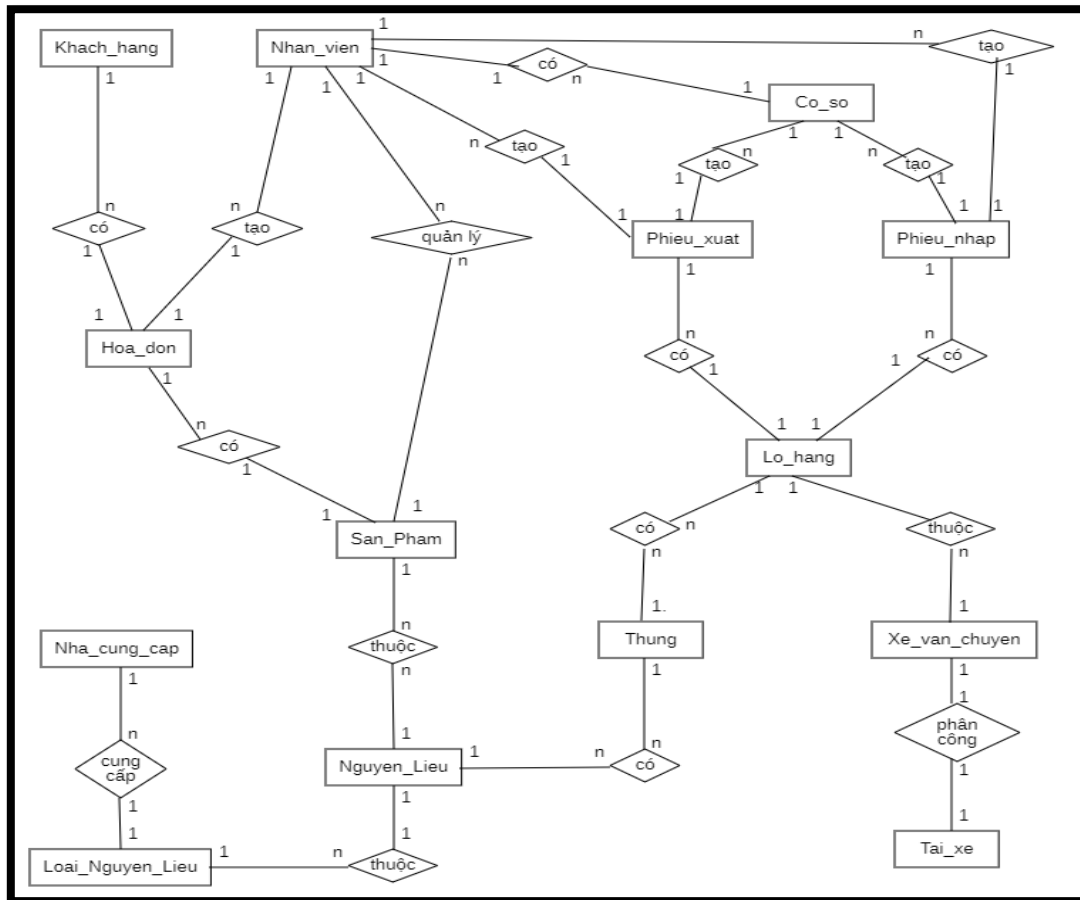


Hình 13: Quy trình xuất hàng



Hình 14: Quy trình kiểm tra tài khoản

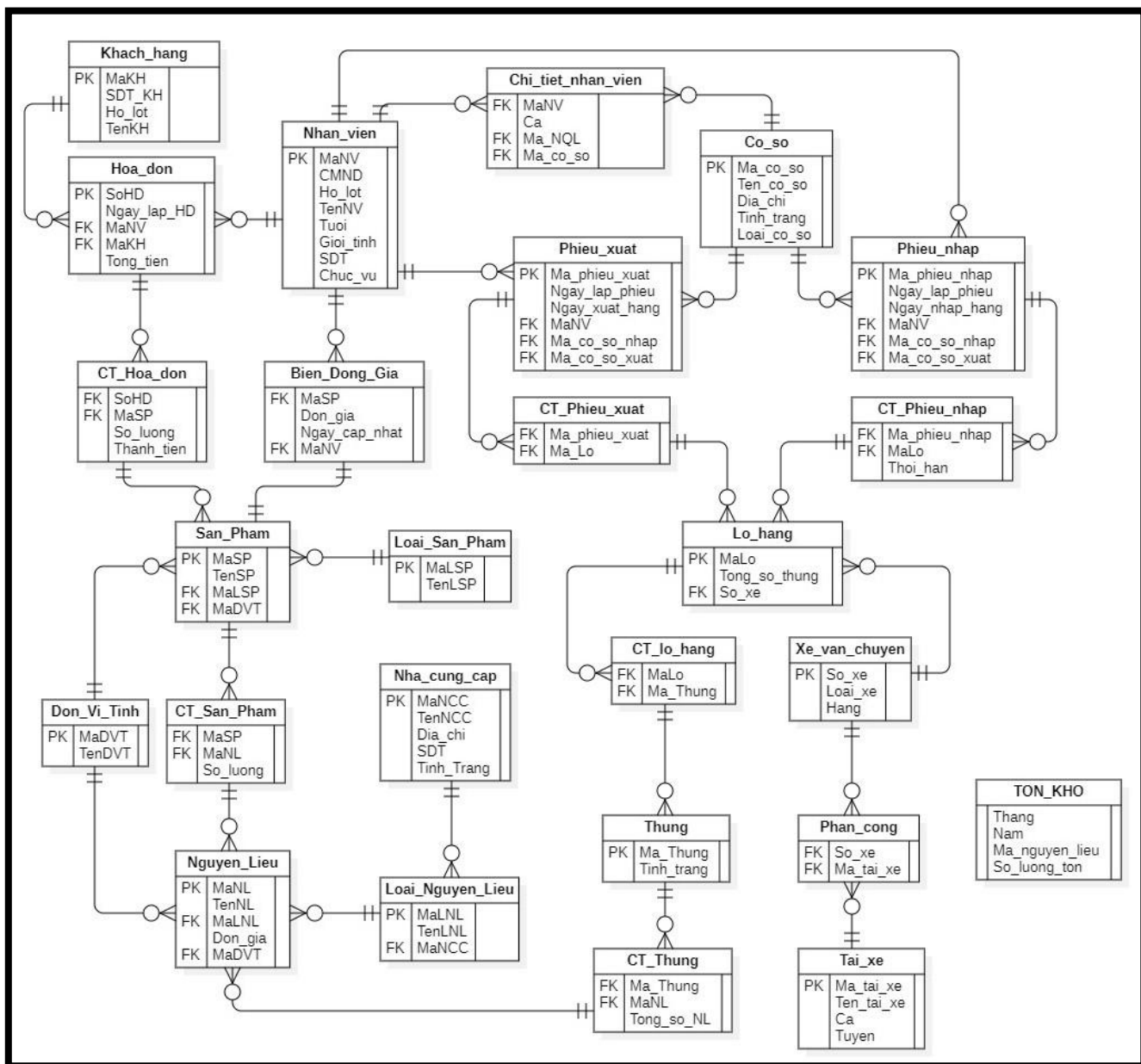
7. Thực thể cần quản lý:



Hình 15: Thực thể cần quản lý

II. Thiết kế cơ sở dữ liệu:

1. Sơ đồ ERD:



Hình 16: Sơ đồ ERD

2. Mô tả bảng:

Nhan_vien			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MaNV	Char (7)	PK
2	CMND	Number (12)	
3	Ho_lot	Nvarchar2 (30)	
4	TenNV	Nvarchar2 (20)	
5	Tuoi	Number (2)	
6	Gioi_tinh	Nvarchar2 (3)	
7	SDT	Number (12)	
8	Chuc_vu	Nvarchar2 (30)	

Chi_tiet_nhan_vien			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MaNV	Char (7)	FK
2	Ca	Number (1)	
3	Ma_NQL	Nvarchar2 (7)	
4	Chuc_vu	Nvarchar2 (30)	
5	Ma_co_so	Nvarchar2 (7)	FK

Phieu_nhap			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	Ma_phieu_nhap	Char (7)	PK
2	Ngay_lap_phieu	Date	
3	MaNV	Char (7)	FK
4	MaLo	Char (7)	
5	Ma_co_so_nhap	Char (7)	FK
6	Ma_co_so_xuat	Char (7)	FK

Phieu_xuat			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	Ma_phieu_xuat	Char (7)	PK
2	Ngay_lap_phieu	Date	
3	MaNV	Char (7)	FK
4	MaLo	Char (7)	
5	Ma_co_so_nhap	Char (7)	FK
6	Ma_co_so_xuat	Char (7)	FK

Co_so			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	Ma_co_so	Char (7)	PK
2	Ten_co_so	Nvarchar2 (30)	
3	Dia_chi	Nvarchar2 (60)	
4	Tinh_trang	Nvarchar2 (20)	
5	Loai_co_so	Nvarchar2 (25)	

Ton_kho			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	Thang	Date	
2	Nam	Date	
3	Ma_nguyen_lieu	Char (7)	
4	Ten_nguyen_lieu	Nvarchar2 (30)	
5	So_luong_ton	Number (10)	

Lo_hang			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MaLo	Char (7)	PK
2	Tong_so_thung	Number(38)	
3	So_xe	Varchar2 (12)	FK

Xe_van_chuyen			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	So_xe	Varchar2 (12)	PK
2	Loai_xe	Nvarchar2 (20)	
3	Hang	Nvarchar2 (20)	

Phan_cong			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	So_xe	Varchar2 (12)	FK
2	Ma_tai_xe	Char (7)	FK

Tai_xe			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	Ma_tai_xe	Char (7)	FK
2	Ten_tai_xe	Nvarchar2 (20)	
3	Ca	Number(1)	
4	Tuyen	Nvarchar2 (30)	

CT_lo_hang			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MaLo	Char (7)	FK
2	Ma_Thung	Char (7)	FK

Thung			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	Ma_Thung	Char (7)	PK
2	Tinh_trang	Nvarchar2 (12)	

CT_Thung			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	Ma_Thung	Char (7)	FK
2	MaNL	Char (7)	FK
3	Tong_so_NL	Number (38)	

Nguyen_Lieu			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MaNL	Char (7)	PK
2	TenNL	Nvarchar2 (30)	
3	Don_Gia	Float	
4	MaLNL	Char (7)	FK
5	MaDVT	Char (7)	FK

Loai_Nguyen_Lieu			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MaLNL	Char (7)	PK
2	TenLNL	Nvarchar2 (30)	
3	MaNCC	Char (7)	FK

Nha_cung_cap			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MaNCC	Char (7)	PK
2	TenNCC	Nvarchar2 (50)	
3	Dia_chi	Nvarchar2 (60)	
4	SĐT	Number (12)	
5	Tinh_Trang	Nvarchar2 (15)	

Don_Vi_Tinh			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MaDVT	Char (7)	PK
2	TenDVT	Nvarchar2 (20)	

Loai_San_Pham			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MaLSP	Char (7)	PK
2	TenLSP	Nvarchar2 (30)	

San_Pham			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MaSP	Char (7)	PK
2	TenSP	Nvarchar2 (30)	
3	MaLSP	Char (7)	FK
4	MaDVT	Char (7)	FK

CT_San_Pham			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MaSP	Char (7)	FK
2	MaNL	Char (7)	FK
3	So_luong	Number (2)	

Bien_Dong_Gia			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MaSP	Char (7)	FK
2	Don_gia	Float	
3	Ngay_cap_nhat	Date	
4	MaNV	Char (7)	FK

Hoa_don			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	SoHD	Char (7)	PK
2	Ngay_lap_HD	Date	
3	MaNV	Char (7)	FK
4	MaKH	Char (7)	FK

CT_Hoa_don			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	SoHD	Char (7)	FK
2	MaSP	Char (7)	FK
3	So_luong	Number (38)	
4	Thanh_tien	Float	

Khach_hang			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MaKH	Char (7)	PK
2	SDT_KH	Number (12)	
3	Ho_lot	Nvarchar2 (20)	
4	Ten_KH	Nvarchar2 (20)	

CT_Phieu_nhap			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	Ma_phieu_nhap	Char (12)	FK
2	MaLo	Char (12)	FK
3	Thoi_han	Number (1)	

CT_Phieu_xuat			
Stt	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	Ma_phieu_xuat	Char (12)	FK
2	MaLo	Char (12)	FK

3. Quy định tạo mã:

STT	Loại		Mã	Chú thích
1	Khách hàng		KH*****	2 kí tự chữ đầu, 5 kí tự số
2	Hóa Đơn		HD*****	2 kí tự chữ đầu, 5 kí tự số
3	Nhân Viên		NV*****	2 kí tự chữ đầu, 5 kí tự số
4	Sản Phẩm		SP*****	2 kí tự chữ đầu, 5 kí tự số
5	Loại Sản Phẩm		LSP*****	3 kí tự chữ đầu, 4 kí tự số
6	Đơn Vị Tính		DVT*****	3 kí tự chữ đầu, 4 kí tự số
7	Nguyên Liệu		NL*****	2 kí tự chữ đầu, 5 kí tự số
8	Loại Nguyên Liệu		LNL*****	3 kí tự chữ đầu, 4 kí tự số
9	Nhà Cung Cấp		NCC*****	3 kí tự chữ đầu, 4 kí tự số
10	Cơ Sở	Kho	KH*****	2 kí tự chữ đầu, 5 kí tự số
		Kho Lạnh	KL*****	
		Trung Tâm Phân Phối	PP*****	
		Chi Nhánh	CN*****	
11	Phiếu Xuất		PX*****	2 kí tự chữ đầu, 5 kí tự số
12	Phiếu Nhập		PN*****	2 kí tự chữ đầu, 5 kí tự số

13	Lô Hàng	LH*****	2 kí tự chữ đầu, 5 kí tự số
14	Thùng	TH*****	2 kí tự chữ đầu, 5 kí tự số
15	Tài Xé	TX*****	2 kí tự chữ đầu, 5 kí tự số

III. Cài đặt:

1. Thiết lập dữ liệu và các ràng buộc:

- Nguyễn Văn Bình Minh - Phạm Gia Khương: tạo bảng, thiết lập ràng buộc cho các bảng (PK, FK, Check, Default, Unique,...)
- Trần Văn Minh: nhập dữ liệu cho database (Insert into)

```
--CREATE TABLE
CREATE TABLE Nhan_vien
(
    MaNV char(7) PRIMARY KEY,
    CMND number(12) NOT NULL UNIQUE,
    Ho_lot nvarchar2(30) NOT NULL,
    TenNV nvarchar2(20) NOT NULL,
    Tuoi number(2) default 18,
    Gioi_tinh nvarchar2(3),
    SDT number(12),
    Chuc_vu nvarchar2(30) DEFAULT N'Nhân viên',
    CONSTRAINT CHK_Gioi_tinh CHECK (Gioi_tinh IN ('Nam', 'Nữ')),
    CONSTRAINT CHK_Chuc_vu CHECK (Chuc_vu IN ('Nhân viên', 'Quản lý'))
);

create table Co_so
(
    Ma_co_so char(7) PRIMARY KEY,
    Ten_co_so nvarchar2(30) NOT NULL,
    Dia_chi nvarchar2(60),
    Tinh_trang nvarchar2(20) default N'Hoạt động'
    CONSTRAINT CHK_Tinh_trang CHECK (Tinh_trang IN ('Hoạt động', 'Ngưng hoạt động')),
    Loai_co_so nvarchar2(25) CHECK (Loai_co_so IN ('Kho', 'Kho lạnh', 'Kho trung tâm phân phối', 'Chi nhánh'))
);
```

```
create table Chi_tiet_nhan_vien
(
    MaNV char(7) REFERENCES Nhan_vien(MaNV),
    Ca number(1),
    Ma_NQL char(7) CONSTRAINT FK_MaNQL REFERENCES Nhan_vien(MaNV) ,
    Ma_co_so char(7) REFERENCES Co_so(Ma_co_so)
);

create table Tai_xe
(
    Ma_tai_xe char(7) PRIMARY KEY,
    Ten_tai_xe nvarchar2(20),
    Ca number(1),
    Tuyen nvarchar2(30)
);

create table Xe_van_chuyen
(
    So_xe varchar2(12) PRIMARY KEY,
    Loai_xe nvarchar2(20),
    Hang nvarchar2(20)
);
```

```

create table Phan_cong
(
    So_xe varchar2(12) REFERENCES Xe_van_chuyen(So_xe),
    Ma_tai_xe char(7) REFERENCES Tai_xe(Ma_tai_xe)
);

create table Nha_cung_cap
(
    MaNCC char(7) PRIMARY KEY,
    TenNCC nvarchar2(30),
    Dia_chi nvarchar2(60),
    SDT number(12),
    Tinh_Trang nvarchar2(15) default N'Hoạt động'
    CONSTRAINT CHK_Tinh_trang_NCC
    CHECK (Tinh_trang IN ('Hoạt động', 'Ngưng hoạt động'))
);

```

```

create table Lo_hang
(
    MaLo char(7) PRIMARY KEY,
    Tong_so_thung integer,
    So_xe varchar2(12) REFERENCES Xe_van_chuyen(So_xe)
);

create table Thung
(
    Ma_Thung char(7) PRIMARY KEY,
    Tinh_trang nvarchar2(12) default N'Nguyên kiện'
    CONSTRAINT CHK_Tinh_trang_Thung
    CHECK (Tinh_trang IN ('Nguyên kiện', 'Đã mở', 'Hỏng'))
);

create table Don_Vi_Tinh
(
    MaDVT char(7) PRIMARY KEY,
    TenDVT nvarchar2(20)
);

```



```

create table Loai_Nguyen_Lieu
(
    MaLNL char(7) PRIMARY KEY,
    TenLNL nvarchar2(30),
    MaNCC char(7) REFERENCES Nha_cung_cap(MaNCC)
);

create table Nguyen_Lieu
(
    MaNL char(7) PRIMARY KEY,
    TenNL nvarchar2(30),
    Don_gia float,
    MaLNL char(7) REFERENCES Loai_Nguyen_Lieu(MaLNL),
    MaDVT char(7) REFERENCES Don_Vi_Tinh(MaDVT)
);

```

```

create table Loai_San_Pham
(
    MaLSP char(7) PRIMARY KEY,
    TenLSP nvarchar2(30)
);

create table San_Pham
(
    MaSP char(7) PRIMARY KEY,
    TenSP nvarchar2(30),
    MaLSP char(7) REFERENCES Loai_San_Pham(MaLSP),
    MaDVT char(7) REFERENCES Don_Vi_Tinh(MaDVT)
);

create table CT_San_Pham
(
    MaSP char(7) REFERENCES San_Pham(MaSP),
    MaNL char(7) REFERENCES Nguyen_Lieu(MaNL),
    So_luong number(2)
);

```

```

create table Bien_Dong_Gia
(
    MaSP char(7) REFERENCES San_Pham(MaSP),
    MaNV char(7) REFERENCES Nhan_vien(MaNV),
    Don_gia float default 1000,
    Ngay_cap_nhat date default SYSDATE
);

create table Khach_hang
(
    MaKH char(7) PRIMARY KEY,
    SDT_KH number(12),
    Ho_lot nvarchar2(20),
    Ten_KH nvarchar2(20)
);

```

```

create table CT_lo_hang
(
    MaLo char(7) REFERENCES Lo_hang(MaLo),
    Ma_Thung char(7) REFERENCES Thung(Ma_Thung)
);

create table CT_Thung
(
    Ma_Thung char(7) REFERENCES Thung(Ma_Thung),
    MaNL char(7) REFERENCES Nguyen_Lieu(MaNL),
    Tong_so_NL integer
);

```

```

create table Phieu_nhap
(
    Ma_phieu_nhap char(7) PRIMARY KEY,
    Ngay_lap_phieu date default SYSDATE,
    Ngay_nhap_hang date default SYSDATE,
    MaNV char(7) REFERENCES Nhan_vien(MaNV),
    Ma_co_so_nhap char(7) CONSTRAINT FK_Ma_co_so_nhap_PN REFERENCES Co_so(Ma_co_so),
    Ma_co_so_xuat char(7) CONSTRAINT FK_Ma_co_so_xuat_PN REFERENCES Co_so(Ma_co_so)
);

create table Phieu_xuat
(
    Ma_phieu_xuat char(7) PRIMARY KEY,
    Ngay_lap_phieu date default SYSDATE,
    Ngay_xuat_hang date default SYSDATE,
    MaNV char(7) REFERENCES Nhan_vien(MaNV),
    Ma_co_so_nhap char(7) CONSTRAINT FK_Ma_co_so_nhap_PX REFERENCES Co_so(Ma_co_so),
    Ma_co_so_xuat char(7) CONSTRAINT FK_Ma_co_so_xuat_PX REFERENCES Co_so(Ma_co_so)
);

```

```

create table CT_Phieu_xuat
(
    Ma_phieu_xuat char(7) REFERENCES Phieu_xuat(Ma_phieu_xuat),
    MaLo char(7) REFERENCES Lo_hang(MaLo)
);

create table CT_Phieu_nhap
(
    Ma_phieu_nhap char(7) REFERENCES Phieu_nhap(Ma_phieu_nhap),
    MaLo char(7) REFERENCES Lo_hang(MaLo),
    Thoi_han number(1) default 5,
    CONSTRAINT CHK_Thoi_han
    CHECK (Thoi_han <= 5)
);

```

```

create table Hoa_don
(
    SoHD char(7) PRIMARY KEY,
    Ngay_lap_HD date default SYSDATE,
    MaNV char(7) REFERENCES Nhan_vien(MaNV),
    MaKH char(7) REFERENCES Khach_hang(MaKH),
    CONSTRAINT CHK_Tong_tien
    CHECK (Tong_tien >= 1000)
);

create table CT_Hoa_don
(
    SoHD char(7) REFERENCES Hoa_don(SoHD),
    MaSP char(7) REFERENCES San_Pham(MaSP),
    So_luong integer default 1,
    Thanh_tien float default 1000,
    CONSTRAINT CHK_Thanh_tien
    CHECK (Thanh_tien >= 1000),
    CONSTRAINT CHK_So_luong
    CHECK (So_luong >= 1)
);

```

```

create table TON_KHO
(
    Thang VARCHAR2(5),
    Nam VARCHAR2(5),
    Ma_nguyen_lieu CHAR(7),
    Ten_nguyen_lieu NVARCHAR2(30),
    So_luong_ton NUMBER(10)
);

```

Create Table

Schema: DOAN

☐ Advanced

Name: TON_KHO

Table

DDL

Columns:

+

×

PK	Name	Data Type	Size	Not Null	Default	Comment
	THANG	VARCHAR2	5	<input type="checkbox"/>		
	NAM	VARCHAR2	5	<input type="checkbox"/>		
	MA_NGUYEN_LIEU	CHARACTER	7	<input type="checkbox"/>		
	TEN_NGUYEN_LIEU	NVARCHAR2	30	<input type="checkbox"/>		
	SO_LUONG_TON	NUMBER	10	<input type="checkbox"/>		

↑

↑

↓

↓

Help

OK

Cancel

```

ALTER TABLE NHAN_VIEN ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_NV_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE CO_SO ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_CS_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE NHA_CUNG_CAP ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_NCC_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE HOA_DON ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_HD_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE KHACH_HANG ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_KH_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE LOAI_SAN_PHAM ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_LSP_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE SAN_PHAM ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_SP_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE LOAI_NGUYEN_LIEU ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_LNL_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE NGUYEN_LIEU ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_NL_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE THUNG ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_TH_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE LO_HANG ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_LH_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE TAI_XE ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_TX_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE XE_VAN_CHUYEN ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_XVC_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE PHIEU_NHAP ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_PN_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE PHIEU_XUAT ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_PX_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));
ALTER TABLE DON_VI_TINH ADD DEL_STATUS VARCHAR2(7) DEFAULT 'ACTIVE' CONSTRAINT CHK_DVT_DEL_STATUS CHECK (DEL_STATUS IN('ACTIVE','DELETED'));

```

```

--Nhan_vien
insert into Nhan_vien ( cmnd, ho_lot, tennv, tuoi, gioi_tinh, sdt, chuc_vu)
values ( '0921858566',N'Nguyễn',N'Văn A','27','Nam','0929441668',N'Quản lý');
insert into Nhan_vien ( cmnd, ho_lot, tennv, tuoi, gioi_tinh, sdt, chuc_vu)
values ( '0939746452',N'Lê',N'Thị N','25','Nữ','0785975795',N'Quản lý');
insert into Nhan_vien ( cmnd, ho_lot, tennv, tuoi, gioi_tinh, sdt, chuc_vu)
values ( '0921336765',N'Nguyễn',N'Văn X','27','Nam','0908678264',N'Quản lý');
insert into Nhan_vien ( cmnd, ho_lot, tennv, tuoi, gioi_tinh, sdt, chuc_vu)
values ( '0939161662',N'Lê',N'Thành T','25','Nam','092673345',N'Quản lý');
insert into Nhan_vien ( cmnd, ho_lot, tennv, tuoi, gioi_tinh, sdt, chuc_vu)
values ( '6729279956',N'Nguyễn',N'Văn C','27','Nam','093925925',N'Nhân viên');
insert into Nhan_vien ( cmnd, ho_lot, tennv, tuoi, gioi_tinh, sdt, chuc_vu)
values ( '6657265746',N'Nguyễn',N'Văn D','26','Nam','095875644',N'Nhân viên');
insert into Nhan_vien ( cmnd, ho_lot, tennv, tuoi, gioi_tinh, sdt, chuc_vu)
values ( '0567527675',N'Nguyễn',N'Văn E','25','Nam','093944258',N'Nhân viên');
insert into Nhan_vien ( cmnd, ho_lot, tennv, tuoi, gioi_tinh, sdt, chuc_vu)
values ( '0697955722',N'Trần',N'Hải Đăng','24','Nam','0978417678',N'Nhân viên');

```

```

--Co_so
insert into Co_so ( ten_co_so, dia_chi, tinh_trang, loai_co_so )
values ('Kho 1',N'Quận 10',N'Hoạt động','Kho');
insert into Co_so ( ten_co_so, dia_chi, tinh_trang, loai_co_so )
values ('Kho 2',N'Quận 6',N'Hoạt động','Kho lạnh');
insert into Co_so ( ten_co_so, dia_chi, tinh_trang, loai_co_so )
values ('Kho 3',N'Quận 2',N'Hoạt động','Kho trung tâm phân phối');
insert into Co_so ( ten_co_so, dia_chi, tinh_trang, loai_co_so )
values ('Kho 4',N'Quận 3',N'Hoạt động','Chi nhánh');

--Chi_tiet_nhan_vien
insert into Chi_tiet_nhan_vien
values ('NV00003','1','NV00001','KH00001');
insert into Chi_tiet_nhan_vien
values ('NV00004','3','NV00002','KL00001');
insert into Chi_tiet_nhan_vien
values ('NV00005','2','NV00006','PP00001');
insert into Chi_tiet_nhan_vien
values ('NV00008','1','NV00007','CN00001');

```

```

--Tai_xe
insert into Tai_xe ( ten_tai_xe, ca, tuyen )
values (N'Thánh','1',N'Quận 10 - Quận 6');
insert into Tai_xe ( ten_tai_xe, ca, tuyen )
values (N'Tài','2',N'Quận 10 - Quận 2');
insert into Tai_xe ( ten_tai_xe, ca, tuyen )
values (N'Phúc','3',N'Quận 10 - Quận 3');

--Xe_van_chuyen
Insert into Xe_van_chuyen (so_xe, loai_xe, hang)
values (N'59-L2-88888',N'Xe tải',N'HYUNDAI');
Insert into Xe_van_chuyen (so_xe, loai_xe, hang)
values ('59-N1-66666',N'Xe tải', 'HYUNDAI');
Insert into Xe_van_chuyen (so_xe, loai_xe, hang)
values ('59-H2-56789',N'Xe tải', 'HYUNDAI');

```

```

--Phan_cong
insert into Phan_cong (so_xe, ma_tai_xe)
values ('59-L2-88888','TX00002');
insert into Phan_cong (so_xe, ma_tai_xe)
values ('59-N1-66666','TX00001');
insert into Phan_cong (so_xe, ma_tai_xe)
values ('59-H2-56789','TX00003');

--Nha_cung_cap
insert into Nha_cung_cap ( tenncc, dia_chi, sdt, tinh_trang )
values (N'NCC Chicken','Quận 1','0995556658',N'Hoạt động');
insert into Nha_cung_cap ( tenncc, dia_chi, sdt, tinh_trang )
values (N'NCC Vegetables','Quận 3','0987865556',N'Hoạt động');
insert into Nha_cung_cap ( tenncc, dia_chi, sdt, tinh_trang )
values (N'NCC Drink','Quận 5','0214548529',N'Hoạt động');

```

```

--Lo_hang
insert into Lo_hang ( tong_so_thung, so_xe )
values (30,'59-L2-88888');
insert into Lo_hang ( tong_so_thung, so_xe )
values (50,'59-N1-66666');
insert into Lo_hang ( tong_so_thung, so_xe )
values (20,'59-H2-56789');

--Thung
insert into Thung (tinh_trang)
values (N'Nguyên kiện');
insert into Thung (tinh_trang)
values (N'Nguyên kiện');
insert into Thung (tinh_trang)
values (N'Đã mở');

```

```

--Thung
insert into Thung (tinh_trang)
values (N'Nguyên kiện');
insert into Thung (tinh_trang)
values (N'Nguyên kiện');
insert into Thung (tinh_trang)
values (N'Đã mở');

--Don_vi_tinh
insert into Don_vi_tinh ( tendvt )
values (N'Thùng');
insert into Don_vi_tinh ( tendvt )
values (N'Con');
insert into Don_vi_tinh ( tendvt )
values (N'Cái');
insert into Don_vi_tinh ( tendvt )
values (N'kg');

```



```

--Loai_nguyen_lieu
insert into Loai_nguyen_lieu ( tenlnl, mancc )
values (N'Thịt','NCC0001');
insert into Loai_nguyen_lieu ( tenlnl, mancc )
values (N'Rau','NCC0002');
insert into Loai_nguyen_lieu ( tenlnl, mancc )
values (N'Nước','NCC0003');

--Nguyen_lieu
insert into Nguyen_lieu ( tennl, don_gia, mainl, madvt )
values (N'Cánh gà','20000','LNL0001','DVT0003');
insert into Nguyen_lieu ( tennl, don_gia, mainl, madvt )
values (N'Đùi gà','20000','LNL0001','DVT0003');
insert into Nguyen_lieu ( tennl, don_gia, mainl, madvt )
values (N'Ức gà','20000','LNL0001','DVT0003');

```

```

--Loai_san_pham
insert into Loai_san_pham ( tenlsp )
values ('Gà');
insert into Loai_san_pham ( tenlsp )
values ('Sandwich');

--San_pham
insert into San_pham ( tensp, malsp, madvt )
values (N'Cánh gà chiên muối tiêu chanh','LSP0001','DVT0003');
insert into San_pham ( tensp, malsp, madvt )
values (N'Cánh gà chiên kiểu Pháp','LSP0001','DVT0003');
insert into San_pham ( tensp, malsp, madvt )
values (N'Đùi gà chiên nước mắm','LSP0001','DVT0003');
insert into San_pham ( tensp, malsp, madvt )
values (N'Sandwich gà sốt phô mai','LSP0001','DVT0003');

--Ct_san_pham
insert into Ct_san_pham
values ('SP00001','NL00001',50);
insert into CT_san_pham
values ('SP00002','NL00002',50);
insert into CT_san_pham
values ('SP00003','NL00003',50);

```

```

--Bien_dong_gia
insert into Bien_dong_gia
values ('SP00001','NV00001','20000',TO_DATE('10/11/2021', 'dd/mm/yyyy'));
insert into Bien_dong_gia
values ('SP00002','NV00001','20000',TO_DATE('10/11/2021', 'dd/mm/yyyy'));
insert into Bien_dong_gia
values ('SP00003','NV00001','20000',TO_DATE('10/11/2021', 'dd/mm/yyyy'));
insert into Bien_dong_gia
values ('SP00004','NV00001','20000',TO_DATE('10/11/2021', 'dd/mm/yyyy'));

--Khach_hang
insert into Khach_hang ( sdt_kh, ho_lot, ten_kh )
values ('0929441668',N'Phạm ',N'Gia Khuong');
insert into Khach_hang ( sdt_kh, ho_lot, ten_kh )
values ('0939664785',N'Trần ',N'Văn Minh');
insert into Khach_hang ( sdt_kh, ho_lot, ten_kh )
values ('0924465758',N'Nguyễn Văn ',N'Bình Minh');

```

```

--Ct_lo_hang
insert into Ct_lo_hang
values ('LH00001','TH00001');
insert into Ct_lo_hang
values ('LH00002','TH00002');

--Ct_thung
insert into Ct_thung
values ('TH00001','NL00001',100);
insert into Ct_thung
values ('TH00002','NL00002',100);

```

```

--Phieu_nhap
insert into Phieu_nhap (ngay_lap_phieu, ngay_nhap_hang, manv, ma_co_so_nhap, ma_co_so_xuat)
values (TO_DATE('1/1/2022', 'dd/mm/yyyy'),TO_DATE('1/1/2022', 'dd/mm/yyyy'),'NV00003','PP00001','KH00001');
insert into Phieu_nhap (ngay_lap_phieu, ngay_nhap_hang, manv, ma_co_so_nhap, ma_co_so_xuat)
values (TO_DATE('10/1/2022', 'dd/mm/yyyy'),TO_DATE('15/1/2022', 'dd/mm/yyyy'),'NV00003','PP00001','KH00001');
insert into Phieu_nhap (ngay_lap_phieu, ngay_nhap_hang, manv, ma_co_so_nhap, ma_co_so_xuat)
values (TO_DATE('15/1/2022', 'dd/mm/yyyy'),TO_DATE('20/1/2022', 'dd/mm/yyyy'),'NV00003','PP00001','KH00001');

--Phieu_xuat
insert into Phieu_xuat (ngay_lap_phieu, ngay_xuat_hang, manv, ma_co_so_nhap, ma_co_so_xuat)
values (TO_DATE('5/1/2022', 'dd/mm/yyyy'),TO_DATE('7/1/2022', 'dd/mm/yyyy'),'NV00003','PP00001','KH00001');
insert into Phieu_xuat (ngay_lap_phieu, ngay_xuat_hang, manv, ma_co_so_nhap, ma_co_so_xuat)
values (TO_DATE('2/4/2022', 'dd/mm/yyyy'),TO_DATE('5/4/2022', 'dd/mm/yyyy'),'NV00003','PP00001','KH00001');
insert into Phieu_xuat (ngay_lap_phieu, ngay_xuat_hang, manv, ma_co_so_nhap, ma_co_so_xuat)
values (TO_DATE('12/5/2022', 'dd/mm/yyyy'),TO_DATE('16/5/2022', 'dd/mm/yyyy'),'NV00003','PP00001','KH00001');

```

```

--Ct_phieu_nhap
insert into Ct_phieu_nhap
values ('PN00001','LH00001',4);
insert into Ct_phieu_nhap
values ('PN00002','LH00002',5);
insert into Ct_phieu_nhap
values ('PN00003','LH00003',4);

--Ct_phieu_xuat
insert into Ct_phieu_xuat
values ('PX00001','LH00001');
insert into Ct_phieu_xuat
values ('PX00002','LH00002');
insert into Ct_phieu_xuat
values ('PX00003','LH00003');

```

```

--Hoa_don
insert into Hoa_don ( ngay_lap_hd, manv, makh )
values (TO_DATE('20/1/2022', 'dd/mm/yyyy'),'NV00003','KH00001');
insert into Hoa_don ( ngay_lap_hd, manv, makh )
values (TO_DATE('22/1/2022', 'dd/mm/yyyy'),'NV00003','KH00002');
insert into Hoa_don ( ngay_lap_hd, manv, makh )
values (TO_DATE('18/6/2022', 'dd/mm/yyyy'),'NV00003','KH00003');
insert into Hoa_don ( ngay_lap_hd, manv, makh )
values (TO_DATE('19/6/2022', 'dd/mm/yyyy'),'NV00003','KH00003');

--Ct_hoa_don
insert into Ct_hoa_don
values ('HD00001','SP00001','3');
insert into Ct_hoa_don (sohd, masp, so_luong)
values ('HD00001','SP00003','2');
insert into Ct_hoa_don
values ('HD00002','SP00002','3');
insert into Ct_hoa_don
values ('HD00002','SP00003','2');
insert into Ct_hoa_don
values ('HD00003','SP00001','3');
insert into Ct_hoa_don
values ('HD00003','SP00003','2');
insert into Ct_hoa_don (sohd, masp, so_luong)
values ('HD00003','SP00002','2');
insert into Ct_hoa_don (sohd, masp, so_luong)
values ('HD00021','SP00001','2');

```

2. Các câu truy vấn:

Trần Văn Minh – Truy Vấn Bảng Đơn

```
-- 10 BẢNG ĐƠN (Văn Minh)
--1 Xem thông tin khách hàng
SELECT * FROM KHACH_HANG;
--2 Xem thông tin khách hàng mã KH00001
SELECT * FROM KHACH_HANG WHERE MAKH = 'KH00001';
--3 Xem tất cả hóa đơn trong ngày
SELECT * FROM HOA_DON WHERE NGÀY_LAP_HD = SYSDATE;
--4 Xem tất cả các hóa đơn từ ngày hôm qua và có tổng tiền hơn 1k?
SELECT * FROM HOA_DON WHERE TONG_TIEN > 1000 AND NGÀY_LAP_HD < SYSDATE;
--5 Xem số lượng hóa đơn các khách hàng đã mua
SELECT MAKH, COUNT(SOHD) FROM HOA_DON GROUP BY MAKH;
--6 Xem mã phiếu nhập có ngày lập phiếu nhỏ hơn ngày nhập hàng
SELECT MA_PHIEU_NHAP FROM PHIEU_NHAP WHERE NGÀY_LAP_PHIEU < NGÀY_NHAP_HANG;
--7 Xem số lượng lô hàng nhập của mỗi phiếu nhập
SELECT MA_PHIEU_NHAP, COUNT(MALO) SO_LO_HANG FROM CT_PHIEU_NHAP GROUP BY ma_phieu_nhap;
--8 Xem số lượng lô hàng nhập lớn hơn 10 của mỗi phiếu nhập
SELECT MA_PHIEU_NHAP, COUNT(MALO) SO_LO_HANG FROM CT_PHIEU_NHAP GROUP BY ma_phieu_nhap HAVING COUNT(MALO)>=10;
--9 Xem số lượng lô hàng nhập lớn hơn 3 của mỗi phiếu nhập và sắp xếp tổng đơn theo mỗi phiếu nhập
SELECT MA_PHIEU_NHAP, COUNT(MALO) SO_LO_HANG FROM CT_PHIEU_NHAP GROUP BY ma_phieu_nhap HAVING COUNT(MALO)>=3 ORDER BY SO_LO_HANG;
--10 Xem phiếu nhập có số lượng lô hàng lớn nhất.
SELECT *
FROM (SELECT MA_PHIEU_NHAP, COUNT(MALO) SO_LO_HANG
      FROM CT_PHIEU_NHAP
      GROUP BY ma_phieu_nhap
      HAVING COUNT(MALO)>=1
      ORDER BY SO_LO_HANG
     )
WHERE ROWNUM = 1;
```

Phạm Gia Khuong

```
--1 Xem toàn bộ chi tiết của nhân viên. (Gia Khuong)
SELECT* FROM NHAN_VIEN JOIN CHI_TIET_NHAN_VIEN ON NHAN_VIEN.MANV = CHI_TIET_NHAN_VIEN.MANV;
--2 Xem nhân viên và cơ sở mà nhân viên làm việc
SELECT NHAN_VIEN.MANV, TEN_CO_SO
FROM NHAN_VIEN JOIN CHI_TIET_NHAN_VIEN ON NHAN_VIEN.MANV = CHI_TIET_NHAN_VIEN.MANV
      JOIN CO_SO ON CHI_TIET_NHAN_VIEN.MA_CO_SO = CO_SO.MA_CO_SO;
--3 Xem phiếu xuất do nhân viên nào xuất ở cơ sở nào
SELECT NHAN_VIEN.MANV, TEN_CO_SO, MA_PHIEU_XUAT
FROM NHAN_VIEN JOIN CHI_TIET_NHAN_VIEN ON NHAN_VIEN.MANV = CHI_TIET_NHAN_VIEN.MANV
      JOIN CO_SO ON CHI_TIET_NHAN_VIEN.MA_CO_SO = CO_SO.MA_CO_SO
      JOIN PHIEU_XUAT ON PHIEU_XUAT.MANV = NHAN_VIEN.MANV;
--4 Xem số lượng cơ sở mà nhân viên làm việc
SELECT NHAN_VIEN.MANV, COUNT(CO_SO.MA_CO_SO)
FROM NHAN_VIEN JOIN CHI_TIET_NHAN_VIEN ON NHAN_VIEN.MANV = CHI_TIET_NHAN_VIEN.MANV
      JOIN CO_SO ON CHI_TIET_NHAN_VIEN.MA_CO_SO = CO_SO.MA_CO_SO
GROUP BY NHAN_VIEN.MANV;
--5 Xem số lượng phiếu xuất >1 mà nhân viên đã lập
SELECT NHAN_VIEN.MANV, COUNT(MA_PHIEU_XUAT)
FROM NHAN_VIEN JOIN CHI_TIET_NHAN_VIEN ON NHAN_VIEN.MANV = CHI_TIET_NHAN_VIEN.MANV
      JOIN PHIEU_XUAT ON PHIEU_XUAT.MANV = NHAN_VIEN.MANV
GROUP BY NHAN_VIEN.MANV
HAVING COUNT(MA_PHIEU_XUAT)>1;
--6 Xem số lượng phiếu nhập từ ba trở lên mà nhân viên NV0003 đã lập
SELECT NHAN_VIEN.MANV, COUNT(MA_PHIEU_NHAP)
FROM NHAN_VIEN JOIN CHI_TIET_NHAN_VIEN ON NHAN_VIEN.MANV = CHI_TIET_NHAN_VIEN.MANV
      JOIN PHIEU_NHAP ON PHIEU_NHAP.MANV = NHAN_VIEN.MANV
      JOIN CO_SO ON CHI_TIET_NHAN_VIEN.MA_CO_SO = CO_SO.MA_CO_SO
GROUP BY NHAN_VIEN.MANV
HAVING COUNT(MA_PHIEU_NHAP)>=3 OR NHAN_VIEN.MANV LIKE 'NV00003' ;
```

```
--7 Xem số lượng nguyên liệu nhập của mỗi phiếu nhập (Bình Minh)
SELECT PN.MA_PHIEU_NHAP, COUNT(MANL) SO_LUONG_NL
FROM PHIEU_NHAP PN
    JOIN CT_PHIEU_NHAP CTPN ON PN.MA_PHIEU_NHAP = CTPN.MA_PHIEU_NHAP
    JOIN LO_HANG LH ON CTPN.MALO = LH.MALO
    JOIN CT_LO_HANG CTLH ON LH.MALO = CTLH.MALO
    JOIN THUNG ON THUNG.MA_THUNG = CTLH.MA_THUNG
    JOIN CT_THUNG ON THUNG.MA_THUNG = CT_THUNG.MA_THUNG
GROUP BY PN.MA_PHIEU_NHAP;

--8 Xem số lượng nguyên liệu nhập >2 của mỗi phiếu nhập
SELECT PN.MA_PHIEU_NHAP, COUNT(MANL) SO_LUONG_NL
FROM PHIEU_NHAP PN
    JOIN CT_PHIEU_NHAP CTPN ON PN.MA_PHIEU_NHAP = CTPN.MA_PHIEU_NHAP
    JOIN LO_HANG LH ON CTPN.MALO = LH.MALO
    JOIN CT_LO_HANG CTLH ON LH.MALO = CTLH.MALO
    JOIN THUNG ON THUNG.MA_THUNG = CTLH.MA_THUNG
    JOIN CT_THUNG ON THUNG.MA_THUNG = CT_THUNG.MA_THUNG
GROUP BY PN.MA_PHIEU_NHAP
HAVING COUNT(MANL) > 2;
```

```
--9 Xem nhập xuất của mỗi tháng
SELECT NHAP_KHO.THANG THANG, NHAP_KHO.NAM NAM, SO_LUONG_NHAP, SO_LUONG_XUAT,
    CASE
        WHEN SO_LUONG_NHAP IS NULL AND SO_LUONG_XUAT IS NULL
            THEN 0
        WHEN SO_LUONG_NHAP IS NOT NULL AND SO_LUONG_XUAT IS NOT NULL
            THEN SO_LUONG_NHAP - SO_LUONG_XUAT
        WHEN SO_LUONG_NHAP IS NOT NULL AND SO_LUONG_XUAT IS NULL
            THEN SO_LUONG_NHAP
        ELSE SO_LUONG_XUAT
    END TON_THANG
FROM NHAP_KHO FULL OUTER JOIN XUAT_KHO
    ON NHAP_KHO.THANG = XUAT_KHO.THANG
    AND NHAP_KHO.NAM = XUAT_KHO.NAM AND NHAP_KHO.MANL = XUAT_KHO.MANL
ORDER BY NAM;
```

```

--10 Xem tên kho của mỗi tháng
SELECT
NHAP_KHO.THANG THANG, NHAP_KHO.NAM NAM, NHAP_KHO.TENNL
,CASE
  --000
  WHEN SO_LUONG_XUAT IS NULL AND SO_LUONG_NHAP_PREV IS NULL AND NHAP_KHO.SO_LUONG_NHAP IS NULL
  THEN 0
  --001
  WHEN SO_LUONG_XUAT IS NULL AND SO_LUONG_NHAP_PREV IS NULL AND NHAP_KHO.SO_LUONG_NHAP IS NOT NULL
  THEN NHAP_KHO.SO_LUONG_NHAP
  --010
  WHEN SO_LUONG_XUAT IS NULL AND SO_LUONG_NHAP_PREV IS NOT NULL AND NHAP_KHO.SO_LUONG_NHAP IS NULL
  THEN NHAP_KHO.SO_LUONG_NHAP - SO_LUONG_XUAT
  --011
  WHEN SO_LUONG_XUAT IS NULL AND SO_LUONG_NHAP_PREV IS NOT NULL AND NHAP_KHO.SO_LUONG_NHAP IS NOT NULL
  THEN NHAP_KHO.SO_LUONG_NHAP + SO_LUONG_NHAP_PREV
  --111
  WHEN SO_LUONG_XUAT IS NOT NULL AND SO_LUONG_NHAP_PREV IS NOT NULL AND NHAP_KHO.SO_LUONG_NHAP IS NOT NULL
  THEN (NHAP_KHO.SO_LUONG_NHAP + SO_LUONG_NHAP_PREV) - SO_LUONG_XUAT
  --110
  WHEN SO_LUONG_XUAT IS NOT NULL AND SO_LUONG_NHAP_PREV IS NOT NULL AND NHAP_KHO.SO_LUONG_NHAP IS NULL
  THEN SO_LUONG_NHAP_PREV - SO_LUONG_XUAT
  --100
  WHEN SO_LUONG_XUAT IS NOT NULL AND SO_LUONG_NHAP_PREV IS NULL AND NHAP_KHO.SO_LUONG_NHAP IS NULL
  THEN 0 - SO_LUONG_XUAT
  ELSE (NHAP_KHO.SO_LUONG_NHAP - SO_LUONG_XUAT)
END SO_LUONG_TON

```

```

FROM
(SELECT NHAP_KHO.THANG THANG, NHAP_KHO.NAM NAM, CASE
  WHEN SO_LUONG_NHAP IS NULL AND SO_LUONG_XUAT IS NULL
  THEN 0
  WHEN SO_LUONG_NHAP IS NOT NULL AND SO_LUONG_XUAT IS NOT NULL
  THEN SO_LUONG_NHAP - SO_LUONG_XUAT
  WHEN SO_LUONG_NHAP IS NOT NULL AND SO_LUONG_XUAT IS NULL
  THEN SO_LUONG_NHAP
  ELSE SO_LUONG_XUAT
END SO_LUONG_NHAP_PREV FROM NHAP_KHO FULL OUTER JOIN XUAT_KHO
ON NHAP_KHO.THANG = XUAT_KHO.THANG
AND NHAP_KHO.NAM = XUAT_KHO.NAM AND NHAP_KHO.MANL = XUAT_KHO.MANL
ORDER BY NAM) A
JOIN NHAP_KHO
ON A.THANG = NHAP_KHO.THANG AND A.NAM=NHAP_KHO.NAM
FULL OUTER JOIN XUAT_KHO
ON NHAP_KHO.THANG = XUAT_KHO.THANG
AND NHAP_KHO.NAM = XUAT_KHO.NAM AND NHAP_KHO.MANL = XUAT_KHO.MANL
ORDER BY NAM;

```

3. View hiện dữ liệu:

Phạm Gia Khương – View Thông tin chi tiết nhân viên

View Thông tin chi tiết nhân viên Admin

```
--1
CREATE OR REPLACE VIEW THONG TIN_CHI TIET_NHAN_VIEN
AS
SELECT NHAN_VIEN.MANV, CONCAT(CONCAT(HO_LOT, ' '), TENNV) AS HO_TEN, GIOI_TINH, SDT, CHUC_VU, CA, MA_NQL, MA_CO_SO
FROM NHAN_VIEN FULL OUTER JOIN CHI_TIET_NHAN_VIEN
ON NHAN_VIEN.MANV = CHI_TIET_NHAN_VIEN.MANV
WHERE DEL_STATUS NOT LIKE 'DELETED';
--1.1
CREATE OR REPLACE VIEW THONG TIN_CHI TIET_NHAN_VIEN_ADMIN
AS
SELECT NHAN_VIEN.MANV, CONCAT(CONCAT(HO_LOT, ' '), TENNV) AS HO_TEN,
GIOI_TINH, SDT, CHUC_VU, CA, MA_NQL, MA_CO_SO, NHAN_VIEN.DEL_STATUS AS TINH_TRANG
FROM NHAN_VIEN FULL OUTER JOIN CHI_TIET_NHAN_VIEN
ON NHAN_VIEN.MANV = CHI_TIET_NHAN_VIEN.MANV;
```

Phạm Gia Khương - View Danh Sách Nhân Viên

View Danh Sách Nhân Viên Admin

```
--2
CREATE OR REPLACE VIEW DANH_SACH_NHAN_VIEN
AS
SELECT MANV, CONCAT(CONCAT(HO_LOT, ' '), TENNV) AS HO_TEN, GIOI_TINH, SDT, CHUC_VU
FROM NHAN_VIEN
WHERE DEL_STATUS NOT LIKE 'DELETED';
--2.1
CREATE OR REPLACE VIEW DANH_SACH_NHAN_VIEN_ADMIN
AS
SELECT MANV, CONCAT(CONCAT(HO_LOT, ' '), TENNV) AS HO_TEN, GIOI_TINH, SDT, CHUC_VU, NHAN_VIEN.DEL_STATUS AS TINH_TRANG
FROM NHAN_VIEN;
```

Nguyễn Văn Bình Minh - View Danh Sách Sản Phẩm

View Danh Sách Sản Phẩm Admin

```
--3
CREATE OR REPLACE VIEW DANH_SACH_SAN_PHAM
AS
SELECT TENS_P, TENLSP, TENDVT
FROM SAN_PHAM JOIN loai_san_pham ON SAN_PHAM.MALSP = LOAI_SAN_PHAM.MALSP
JOIN DON_VI_TINH ON SAN_PHAM.MADVT = don_vi_tinh.madv_t
WHERE SAN_PHAM.DEL_STATUS NOT LIKE 'DELETED';
--3.1
CREATE OR REPLACE VIEW DANH_SACH_SAN_PHAM_ADMIN
AS
SELECT TENS_P, TENLSP, TENDVT, SAN_PHAM.DEL_STATUS AS TINH_TRANG
FROM SAN_PHAM JOIN loai_san_pham ON SAN_PHAM.MALSP = LOAI_SAN_PHAM.MALSP
JOIN DON_VI_TINH ON SAN_PHAM.MADVT = don_vi_tinh.madv_t;
```

Nguyễn Văn Bình Minh - View Công Thức Sản Phẩm

View Công Thức Sản Phẩm Admin

```
--4
CREATE OR REPLACE VIEW CONG_THUC_SAN_PHAM
AS
SELECT TENS_P, TENNL, SO_LUONG
FROM CT_SAN_PHAM JOIN NGUYEN_LIEU ON CT_SAN_PHAM.MANL = NGUYEN_LIEU.MANL
JOIN SAN_PHAM ON SAN_PHAM.MASP = CT_SAN_PHAM.MASP
WHERE SAN_PHAM.DEL_STATUS NOT LIKE 'DELETED';
--4.1
CREATE OR REPLACE VIEW CONG_THUC_SAN_PHAM_ADMIN
AS
SELECT TENS_P, TENNL, SO_LUONG, SAN_PHAM.DEL_STATUS AS TINH_TRANG
FROM CT_SAN_PHAM JOIN NGUYEN_LIEU ON CT_SAN_PHAM.MANL = NGUYEN_LIEU.MANL
JOIN SAN_PHAM ON SAN_PHAM.MASP = CT_SAN_PHAM.MASP;
```


Trần Văn Minh – View Hóa Đơn

View Danh Sách Hóa Đơn

View Danh Sách Hóa Đơn Admin

```
--5
CREATE OR REPLACE VIEW V_HOA_DON
AS
SELECT HOA_DON.SOHD,NGAY_LAP_HD,MANV,MAKH,SUM(THANH_TIEN) TONG_TIEN, DEL_STATUS
FROM HOA_DON JOIN CT_HOA_DON ON HOA_DON.SOHD = CT_HOA_DON.SOHD
GROUP BY HOA_DON.SOHD,NGAY_LAP_HD,MANV,MAKH,DEL_STATUS;
--5.1
CREATE OR REPLACE VIEW DANH_SACH_HOA_DON
AS
SELECT SOHD, NGAY_LAP_HD, TONG_TIEN,
CONCAT(CONCAT(NHAN_VIEN.HO_LOT,' '), TENNV) AS NHAN_VIEN_LAP_PHIEU, CONCAT(CONCAT(KHACH_HANG.HO_LOT,' '), TEN_KH)AS TEN_KHACH_HANG
FROM V_HOA_DON JOIN KHACH_HANG ON V_HOA_DON.MAKH = KHACH_HANG.MAKH
JOIN NHAN_VIEN ON V_HOA_DON.MANV=NHAN_VIEN.MANV
WHERE V_HOA_DON.DEL_STATUS NOT LIKE 'DELETED';
--5.2
CREATE OR REPLACE VIEW DANH_SACH_HOA_DON_ADMIN
AS
SELECT SOHD, NGAY_LAP_HD, TONG_TIEN,
CONCAT(CONCAT(NHAN_VIEN.HO_LOT,' '), TENNV) AS NHAN_VIEN_LAP_PHIEU,
CONCAT(CONCAT(KHACH_HANG.HO_LOT,' '), TEN_KH)AS TEN_KHACH_HANG, V_HOA_DON.DEL_STATUS AS TINH_TRANG
FROM V_HOA_DON JOIN KHACH_HANG ON V_HOA_DON.MAKH = KHACH_HANG.MAKH
JOIN NHAN_VIEN ON V_HOA_DON.MANV=NHAN_VIEN.MANV;
```

Trần Văn Minh – View Thông Tin Hóa Đơn

View Thông Tin Hóa Đơn Admin

```
--6
CREATE OR REPLACE VIEW THONG TIN_HOA_DON
AS
SELECT HOA_DON.SOHD, TENSF, SO_LUONG, THANH_TIEN
FROM HOA_DON JOIN CT_HOA_DON ON HOA_DON.SOHD = CT_HOA_DON.SOHD
JOIN SAN_PHAM ON CT_HOA_DON.MASP = SAN_PHAM.MASP
WHERE HOA_DON.DEL_STATUS NOT LIKE 'DELETED';
--6.1
CREATE OR REPLACE VIEW THONG TIN_HOA_DON_ADMIN
AS
SELECT HOA_DON.SOHD, TENSF, SO_LUONG, THANH_TIEN,HOA_DON.DEL_STATUS AS TINH_TRANG
FROM HOA_DON JOIN CT_HOA_DON ON HOA_DON.SOHD = CT_HOA_DON.SOHD
JOIN SAN_PHAM ON CT_HOA_DON.MASP = SAN_PHAM.MASP;
```

Nguyễn Văn Bình Minh - View Nhập Kho

```
--7
CREATE OR REPLACE VIEW NHAP_KHO
AS
SELECT TO_CHAR(NGAY_NHAP_HANG, 'MM') AS THANG, TO_CHAR(NGAY_NHAP_HANG, 'YYYY') AS NAM, NL.MANL,
TENNL, SUM(TONG_SO_NL) AS SO_LUONG_NHAP
FROM PHIEU_NHAP PN JOIN CT_PHIEU_NHAP CTPN ON PN.MA_PHIEU_NHAP = CTPN.MA_PHIEU_NHAP
JOIN LO_HANG LH ON LH.MALO = CTPN.MALO
JOIN CT_LO_HANG CTLH ON LH.MALO=CTLH.MALO
JOIN THUNG TH ON TH.MA_THUNG = CTLH.MA_THUNG
JOIN CT_THUNG CTTH ON TH.MA_THUNG = CTTH.MA_THUNG
JOIN NGUYEN_LIEU NL ON NL.MANL = CTTH.MANL
JOIN CO_SO CS ON PN.MA_CO_SO_NHAP = CS.MA_CO_SO
WHERE LOAI_CO_SO LIKE N'Kho' OR LOAI_CO_SO LIKE N'Kho lạnh'
GROUP BY TO_CHAR(NGAY_NHAP_HANG, 'MM'), TO_CHAR(NGAY_NHAP_HANG, 'YYYY'), NL.MANL, TENNL;
```

Nguyễn Văn Bình Minh - View Xuất Kho

```
--8
CREATE OR REPLACE VIEW XUAT_KHO
AS
SELECT TO_CHAR(NGAY_XUAT_HANG, 'MM') AS THANG, TO_CHAR(NGAY_XUAT_HANG, 'YYYY') AS NAM, NL.MANL,
TENNL, SUM(TONG_SO_NL) AS SO_LUONG_XUAT
FROM PHIEU_XUAT PX JOIN CT_PHIEU_XUAT CTPX ON PX.MA_PHIEU_XUAT = CTPX.MA_PHIEU_XUAT
JOIN LO_HANG LH ON LH.MALO = CTPX.MALO
JOIN CT_LO_HANG CTLH ON LH.MALO=CTLH.MALO
JOIN THUNG TH ON TH.MA_THUNG = CTLH.MA_THUNG
JOIN CT_THUNG CTTH ON TH.MA_THUNG = CTTH.MA_THUNG
JOIN NGUYEN_LIEU NL ON NL.MANL = CTTH.MANL
JOIN CO_SO CS ON PX.MA_CO_SO_XUAT = CS.MA_CO_SO
WHERE LOAI_CO_SO LIKE N'Kho' OR LOAI_CO_SO LIKE N'Kho lạnh'
GROUP BY TO_CHAR(NGAY_XUAT_HANG, 'MM'), TO_CHAR(NGAY_XUAT_HANG, 'YYYY'), NL.MANL, TENNL;
```

Phạm Gia Khương – View Xem Tồn Kho

```
--9
CREATE OR REPLACE VIEW XEM_TON_KHO
AS
SELECT * FROM TON_KHO;
```

Trần Văn Minh – View Chi Phí

View Tổng Thu

View Lợi Nhuận

```
--11
CREATE OR REPLACE VIEW CHI_PHI
AS
SELECT THANG, NAM, NGUYEN_LIEU.TENNL, DON_GIA,SO_LUONG_NHAP, (DON_GIA * SO_LUONG_NHAP) TONG_CHI_PHI
FROM NHAP_KHO JOIN NGUYEN_LIEU ON NHAP_KHO.MANL = NGUYEN_LIEU.MANL;

--12
CREATE OR REPLACE VIEW TONG_THU
AS
SELECT TO_CHAR(NGAY_LAP_HD,'MM') AS THANG,TO_CHAR(NGAY_LAP_HD,'YYYY') AS NAM, SUM(TONG_TIEN) TONG_THU
FROM V_HOA_DON
GROUP BY TO_CHAR(NGAY_LAP_HD,'MM'),TO_CHAR(NGAY_LAP_HD,'YYYY');

--13
CREATE OR REPLACE VIEW LOI_NHUAN
AS
SELECT TONG_THU.THANG, TONG_THU.NAM, (TONG_THU - TONG_CHI_PHI) LOI_NHUAN
FROM CHI_PHI FULL OUTER JOIN TONG_THU ON CHI_PHI.THANG = TONG_THU.THANG AND CHI_PHI.NAM = TONG_THU.NAM;
```

Phạm Gia Khương – View Thông Tin Phiếu Nhập

```
--14
CREATE OR REPLACE VIEW THONG TIN_PHIEU_NHAP
AS
SELECT PN.MA_PHIEU_NHAP, NGAY_LAP_PHIEU, CONCAT(CONCAT(HO_LOT,' '), TENNV) AS HO_TEN ,NHAP_TU,XUAT_DEN, NGAY_NHAP_HANG
FROM PHIEU_NHAP PN JOIN CT_PHIEU_NHAP CTPN ON PN.MA_PHIEU_NHAP = CTPN.MA_PHIEU_NHAP
JOIN NHAN_VIEN NV ON PN.MANV = NV.MANV
JOIN (SELECT (PHIEU_NHAP.MA_CO_SO_NHAP) AS MA_NHAP, TEN_CO_SO_NHAP_TU FROM CO_SO JOIN PHIEU_NHAP ON CO_SO.MA_CO_SO = PHIEU_NHAP.MA_CO_SO_NHAP)
ON PN.MA_CO_SO_NHAP = A.MA_NHAP
FULL OUTER JOIN (SELECT (PHIEU_NHAP.MA_CO_SO_XUAT) AS MA_XUAT, TEN_CO_SO_XUAT_DEN FROM CO_SO JOIN PHIEU_NHAP
ON CO_SO.MA_CO_SO = PHIEU_NHAP.MA_CO_SO_XUAT) B
ON PN.MA_CO_SO_XUAT = B.MA_XUAT;
```

Phạm Gia Khương – View Thông Tin Phiếu Xuất

```
--15
CREATE OR REPLACE VIEW THONG TIN_PHIEU_XUAT
AS
SELECT PX.MA_PHIEU_XUAT, NGAY_LAP_PHIEU, CONCAT(CONCAT(HO_LOT,' '), TENNV) AS HO_TEN ,NHAP_DEN, XUAT_TU, NGAY_XUAT_HANG
FROM PHIEU_XUAT PX JOIN CT_PHIEU_XUAT CTPX ON PX.MA_PHIEU_XUAT = CTPX.MA_PHIEU_XUAT
JOIN NHAN_VIEN NV ON PX.MANV = NV.MANV
FULL OUTER JOIN (SELECT (PHIEU_XUAT.MA_CO_SO_NHAP) AS MA_NHAP, TEN_CO_SO_NHAP_DEN FROM CO_SO JOIN PHIEU_XUAT
ON CO_SO.MA_CO_SO = PHIEU_XUAT.MA_CO_SO_NHAP) A
ON PX.MA_CO_SO_NHAP = A.MA_NHAP
FULL OUTER JOIN (SELECT (PHIEU_XUAT.MA_CO_SO_XUAT) AS MA_XUAT, TEN_CO_SO_XUAT_TU FROM CO_SO JOIN PHIEU_XUAT
ON CO_SO.MA_CO_SO = PHIEU_XUAT.MA_CO_SO_XUAT) B
ON PX.MA_CO_SO_XUAT = B.MA_XUAT;
```

Nguyễn Văn Bình Minh - View V_tồn_kho

```
CREATE OR REPLACE VIEW V_TON_KHO
AS
SELECT
NHAP_KHO.THANG THANG, NHAP_KHO.NAM NAM, NHAP_KHO.MANL MANL, NHAP_KHO.TENNL TENNL,
CASE
    WHEN SO_LUONG_XUAT IS NULL
        THEN SO_LUONG_NHAP - 0
    ELSE (SO_LUONG_NHAP - SO_LUONG_XUAT)
END SO_LUONG_TON
FROM NHAP_KHO FULL OUTER JOIN XUAT_KHO ON NHAP_KHO.THANG = XUAT_KHO.THANG
AND NHAP_KHO.NAM = XUAT_KHO.NAM AND NHAP_KHO.MANL = XUAT_KHO.MANL
WHERE NHAP_KHO.THANG IS NOT NULL AND NHAP_KHO.NAM IS NOT NULL AND NHAP_KHO.MANL
IS NOT NULL AND NHAP_KHO.TENNL IS NOT NULL AND
CASE
    WHEN SO_LUONG_XUAT IS NULL
        THEN SO_LUONG_NHAP - 0
    ELSE (SO_LUONG_NHAP - SO_LUONG_XUAT)
END IS NOT NULL
ORDER BY NHAP_KHO.NAM;
```

4. Stored Procedure

Phạm Gia Khương – Thủ tục Insert Nhân Viên

```
CREATE or replace PROCEDURE PRC_INSERT_NV (  
    input_cmnd number,  
    input_holot nvarchar2,  
    input_tennv nvarchar2,  
    input_tuoi number,  
    input_gioi_tinh nvarchar2,  
    input_sdt char,  
    input_chucvu nvarchar2  
)  
  
AS  
BEGIN  
    IF input_chucvu IS NULL  
    THEN  
        INSERT INTO NHAN_VIEN (CMND,HO_LOT,TENNV,TUOI,GIOI_TINH,SDT)  
        VALUES(input_cmnd,input_holot,input_tennv,input_tuoi,input_gioi_tinh,input_sdt);  
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
  
    ELSIF input_chucvu IS NOT NULL  
    THEN  
        INSERT INTO NHAN_VIEN (CMND,HO_LOT,TENNV,TUOI,GIOI_TINH,SDT,CHUC_VU)  
        VALUES(input_cmnd,input_holot,input_tennv,input_tuoi,input_gioi_tinh,input_sdt,input_chucvu);  
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
  
    ELSE  
        DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');  
    END IF;  
END;
```

Phạm Gia Khương – Thủ tục Insert Chi Tiết Nhân Viên

```
CREATE or replace PROCEDURE PRC_INSERT_CT_NV (  
    input_MaNV char,  
    input_Ca number,  
    input_MaSQL char,  
    input_MaCoSo char  
)  
  
AS  
BEGIN  
    INSERT INTO chi_tiet_nhan_vien  
    VALUES(input_MaNV,input_Ca,input_MaSQL,input_MaCoSo);  
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
    EXCEPTION WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');  
END;
```

Nguyễn Văn Bình Minh – Thủ tục Insert Cơ Sở

```
CREATE or replace PROCEDURE PRC_INSERT_COSO (
    input_ten nvarchar2,
    input_diachi nvarchar2,
    input_tinhtrang nvarchar2,
    input_loai nvarchar2
)
AS
BEGIN
    IF input_loai LIKE N'Kho'
    THEN
        INSERT INTO CO_SO(TEN_CO_SO,DIA_CHI,TINH_TRANG,LOAI_CO_SO)
        VALUES(input_ten,input_diachi,input_tinhtrang,input_loai);
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');

    ELSIF input_loai LIKE N'Kho lạnh'
    THEN
        INSERT INTO CO_SO(TEN_CO_SO,DIA_CHI,TINH_TRANG,LOAI_CO_SO)
        VALUES(input_ten,input_diachi,input_tinhtrang,input_loai);
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS KHO_LANH');

    ELSIF input_loai LIKE N'Kho trung tâm phân phối'
    THEN
        INSERT INTO CO_SO(TEN_CO_SO,DIA_CHI,TINH_TRANG,LOAI_CO_SO)
        VALUES(input_ten,input_diachi,input_tinhtrang,input_loai);
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS PHAN_PHOI');

    ELSIF input_loai LIKE N'Chi nhánh'
    THEN
        INSERT INTO CO_SO(TEN_CO_SO,DIA_CHI,TINH_TRANG,LOAI_CO_SO)
        VALUES(input_ten,input_diachi,input_tinhtrang,input_loai);
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS CHI_NHANH');

    ELSE
        INSERT INTO CO_SO(TEN_CO_SO,DIA_CHI,TINH_TRANG)
        VALUES(input_ten,input_diachi,input_tinhtrang);
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS DEFAULT FACILITY');
    END IF;
END;
```

Trần Văn Minh – Thủ tục Insert Khách Hàng

```
CREATE or replace PROCEDURE PRC_INSERT_KH (
    input_sdt number,
    input_holot nvarchar2,
    input_tenk nvarchar2
)
AS
BEGIN
    INSERT INTO KHACH_HANG (SDT_KH, HO_LOT, TEN_KH)
    VALUES (input_sdt, input_holot, input_tenk);
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');
    EXCEPTION WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');
END;
```

Nguyễn Văn Bình Minh – Thủ tục Insert Hóa Đơn

```
create or replace PROCEDURE PRC_INSERT_HD (
    input_ngaylap date,
    input_manv char,
    input_makh char
)
AS
BEGIN
    IF input_ngaylap IS NULL
    THEN
        INSERT INTO HOA_DON (MANV, MAKH)
        VALUES (input_manv, input_makh);
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');

    ELSIF input_ngaylap IS NOT NULL
    THEN
        INSERT INTO HOA_DON (NGAY_LAP_HD, MANV, MAKH)
        VALUES (input_ngaylap, input_manv, input_makh);
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');
    -- EXCEPTION WHEN OTHERS THEN
    ELSE
        DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');

    END IF;
END;
```

Phạm Gia Khương – Thủ tục Insert Đơn Vị Tính

```
CREATE or replace PROCEDURE PRC_INSERT_DVT (  
                                input_tendvt nvarchar2  
                                )  
AS  
BEGIN  
    INSERT INTO DON_VI_TINH(TENDVT)  
    VALUES(input_tendvt);  
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
    EXCEPTION WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');  
END;
```

Phạm Gia Khương – Thủ tục Insert Loại Sản Phẩm

```
CREATE or replace PROCEDURE PRC_INSERT_LSP (  
                                input_tenlsp nvarchar2  
                                )  
AS  
BEGIN  
    INSERT INTO LOAI_SAN_PHAM (TENLSP)  
    VALUES(input_tenlsp);  
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
    EXCEPTION WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');  
END;
```


Trần Văn Minh – Thủ tục Insert Sản Phẩm

```
CREATE or replace PROCEDURE PRC_INSERT_SP (
    input_tensp nvarchar2,
    input_maloaisp char,
    input_madvtn char
)
AS
BEGIN
    INSERT INTO SAN_PHAM (TENSP,MALSP,MADVT)
    VALUES(input_tensp,input_maloaisp,input_madvtn);
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');
    EXCEPTION WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');
END;
```

Trần Văn Minh – Thủ tục Insert Chi Tiết Hóa Đơn

```
CREATE or replace PROCEDURE PRC_INSERT_CTHD (
    input_sohd CHAR,
    input_masp char,
    input_soluong NUMBER
)
AS
BEGIN
    INSERT INTO CT_HOA_DON (sohd,masp,so_luong)
    VALUES(input_sohd,input_masp,input_soluong);
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');
    EXCEPTION WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');
END;
```

Phạm Gia Khương – Thủ tục Insert Biến Động Giá

```
CREATE or replace PROCEDURE PRC_INSERT_BDG (  
    input_masp CHAR,  
    input_manv char,  
    input_dongia float,  
    input_ngaycapnhat date  
)  
AS  
BEGIN  
    IF input_ngaycapnhat IS NULL  
    THEN  
        INSERT INTO BIEN_DONG_GIA (MASP,MANV,DON_GIA)  
        VALUES(input_masp,input_manv,input_dongia);  
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
    ELSIF input_ngaycapnhat IS NOT NULL  
    THEN  
        INSERT INTO BIEN_DONG_GIA  
        VALUES(input_masp,input_manv,input_dongia,input_ngaycapnhat);  
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
    ELSE  
        DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');  
    END IF;  
END;
```

Nguyễn Văn Bình Minh – Thủ tục Insert Phiếu Xuất

```
CREATE or replace PROCEDURE PRC_INSERT_PX (
    input_ngaylappphieu date,
    input_manv char,
    input_cosonhap char,
    input_cosoxuat char,
    input_ngayxuathang date
)
AS
BEGIN
    IF input_ngaylappphieu IS NULL AND input_ngayxuathang IS NULL
    THEN

        INSERT INTO PHIEU_XUAT (MANV,MA_CO_SO_NHAP,MA_CO_SO_XUAT)
        VALUES(input_manv,input_cosonhap,input_cosoxuat);
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');

    ELSIF input_ngaylappphieu IS NOT NULL AND input_ngayxuathang IS NOT NULL
    THEN

        INSERT INTO PHIEU_XUAT (NGAY_LAP_PHIEU,MANV,MA_CO_SO_NHAP,MA_CO_SO_XUAT,NGAY_XUAT_HANG)
        VALUES(input_ngaylappphieu,input_manv,input_cosonhap,input_cosoxuat,input_ngayxuathang);
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');

    ELSIF input_ngaylappphieu IS NULL AND input_ngayxuathang IS NOT NULL
    THEN

        INSERT INTO PHIEU_XUAT (MANV,MA_CO_SO_NHAP,MA_CO_SO_XUAT,NGAY_XUAT_HANG)
        VALUES(input_manv,input_cosonhap,input_cosoxuat,input_ngayxuathang);
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');

    ELSIF input_ngaylappphieu IS NOT NULL AND input_ngayxuathang IS NULL
    THEN

        INSERT INTO PHIEU_XUAT (NGAY_LAP_PHIEU,MANV,MA_CO_SO_NHAP,MA_CO_SO_XUAT)
        VALUES(input_ngaylappphieu,input_manv,input_cosonhap,input_cosoxuat);
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');

    ELSE
        DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');
    END IF;
END;
```

Nguyễn Văn Bình Minh – Thủ tục Insert Phiếu Xuất

```
CREATE or replace PROCEDURE PRC_INSERT_FN (  
    input_ngaylapphieu date,  
    input_manv char,  
    input_cosonhap char,  
    input_cosoxuat char,  
    input_ngaynhaphang date  
)  
  
AS  
BEGIN  
    IF input_ngaylapphieu IS NULL AND input_ngaynhaphang IS NULL  
    THEN  
  
        INSERT INTO PHIEU_NHAP (MANV,MA_CO_SO_NHAP,MA_CO_SO_XUAT)  
        VALUES(input_manv,input_cosonhap,input_cosoxuat);  
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
  
    ELSIF input_ngaylapphieu IS NOT NULL AND input_ngaynhaphang IS NOT NULL  
    THEN  
  
        INSERT INTO PHIEU_NHAP(NGAY_LAP_PHIEU,MANV,MA_CO_SO_NHAP,MA_CO_SO_XUAT,NGAY_NHAP_HANG)  
        VALUES(input_ngaylapphieu,input_manv,input_cosonhap,input_cosoxuat,input_ngaynhaphang);  
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
  
    ELSIF input_ngaylapphieu IS NULL AND input_ngaynhaphang IS NOT NULL  
    THEN  
  
        INSERT INTO PHIEU_NHAP (MANV,MA_CO_SO_NHAP,MA_CO_SO_XUAT,NGAY_NHAP_HANG)  
        VALUES(input_manv,input_cosonhap,input_cosoxuat,input_ngaynhaphang);  
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
  
    ELSIF input_ngaylapphieu IS NOT NULL AND input_ngaynhaphang IS NULL  
    THEN  
  
        INSERT INTO PHIEU_NHAP (NGAY_LAP_PHIEU,MANV,MA_CO_SO_NHAP,MA_CO_SO_XUAT)  
        VALUES(input_ngaylapphieu,input_manv,input_cosonhap,input_cosoxuat);  
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
  
    ELSE  
        DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');  
    END IF;  
END;
```

Trần Văn Minh – Thủ tục Insert Tài Xế

```
CREATE or replace PROCEDURE PRC_INSERT_TX (  
    input_tentx nvarchar2,  
    input_ca number,  
    input_tuyen nvarchar2  
)  
  
AS  
BEGIN  
    INSERT INTO TAI_XE (TEN_TAI_XE,CA,TUYEN)  
VALUES(input_tentx,input_ca,input_tuyen);  
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
    EXCEPTION WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');  
END;
```

Trần Văn Minh – Thủ tục Insert Xe Vận Chuyển

```
CREATE or replace PROCEDURE PRC_INSERT_XVC (  
    input_soxe varchar2,  
    input_loaixे nvarchar2,  
    input_hang nvarchar2  
)  
  
AS  
BEGIN  
    INSERT INTO XE_VAN_CHUYEN (SO_XE,LOAI_XE,HANG)  
VALUES(input_soxe,input_loaixे,input_hang);  
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
    EXCEPTION WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');  
END;
```

Trần Văn Minh – Thủ tục Insert Phân Công

```
CREATE or replace PROCEDURE PRC_INSERT_PC (
                                input_matx char,
                                input_soxe varchar2
                                )
AS
BEGIN
    INSERT INTO PHAN_CONG
    VALUES(input_matx,input_soxe);
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');
    EXCEPTION WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');
END;
```

Phạm Gia Khương – Thủ tục Insert Lô Hàng

```
CREATE or replace PROCEDURE PRC_INSERT_LH (
                                input_tongthung number,
                                input_soxe varchar2
                                )
AS
BEGIN
    INSERT INTO LO_HANG (TONG_SO_THUNG,SO_XE)
    VALUES(input_tongthung,input_soxe);
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');
    EXCEPTION WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');
END;
```

Phạm Gia Khương – Thủ tục Insert Thùng

```
CREATE or replace PROCEDURE PRC_INSERT_TH (
                                input_tinhtrang varchar2
                                )
AS
BEGIN
    INSERT INTO THUNG (TINH_TRANG)
    VALUES(input_tinhtrang);
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');
    EXCEPTION WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');
END;
```

Phạm Gia Khương – Thủ tục Insert Chi Tiết Lô Hàng

```
CREATE or replace PROCEDURE PRC_INSERT_CTLH (  
                                input_malo char,  
                                input_mathung char  
                                )  
AS  
BEGIN  
    INSERT INTO CT_LO_HANG  
    VALUES(input_malo,input_mathung);  
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
    EXCEPTION WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');  
END;
```

Nguyễn Văn Bình Minh – Thủ tục Insert Chi tiết phiếu nhập

```
create or replace PROCEDURE PRC_INSERT_CTPN (  
                                input_mapn char,  
                                input_malo char,  
                                input_thoihan number  
                                )  
AS  
BEGIN  
    IF input_thoihan > 5  
    THEN  
    DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');  
    ELSE  
    INSERT INTO CT_PHIEU_NHAP  
    VALUES(input_mapn,input_malo,input_thoihan);  
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
    PRC_GET_TON_KHO;  
    END IF;  
END;
```

Nguyễn Văn Bình Minh – Thủ tục Insert Chi Tiết Phiếu Xuất

```
CREATE or replace PROCEDURE PRC_INSERT_CTPX (  
                                input_mapx char,  
                                input_malo char  
                                )  
AS  
BEGIN  
    INSERT INTO CT_PHIEU_XUAT  
VALUES(input_mapx,input_malo);  
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
    EXCEPTION WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');  
END;
```

Phạm Gia Khương – Thủ tục Insert Nhà Cung Cấp

```
CREATE or replace PROCEDURE PRC_INSERT_NCC (  
                                input_tenncc nvarchar2,  
                                input_diachi nvarchar2,  
                                input_sdt number,  
                                input_tinhtrang nvarchar2  
                                )  
AS  
BEGIN  
    INSERT INTO NHA_CUNG_CAP(TENNCC,DIA_CHI,SDT,TINH_TRANG)  
VALUES(input_tenncc,input_diachi,input_sdt,input_tinhtrang);  
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
    EXCEPTION WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');  
END;
```


Phạm Gia Khương – Thủ tục Insert Loại Nguyên Liệu

```
CREATE or replace PROCEDURE PRC_INSERT_LNL (
    input_tenlnl nvarchar2,
    input_mancc char
)
AS
BEGIN
    INSERT INTO LOAI_NGUYEN_LIEU(TENLNL,MANCC)
    VALUES(input_tenlnl,input_mancc);
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');
    EXCEPTION WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');
END;
```

Trần Văn Minh – Thủ tục Insert Nguyên Liệu

```
CREATE or replace PROCEDURE PRC_INSERT_NL (
    input_tenlnl nvarchar2,
    input_dongia float,
    input_malnl char,
    input_madvn char
)
AS
BEGIN
    INSERT INTO NGUYEN_LIEU(TENNL,DON_GIA,MALNL,MADVT)
    VALUES(input_tenlnl,input_dongia,input_malnl,input_madvn);
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');
    EXCEPTION WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');
END;
```

Trần Văn Minh – Thủ tục Insert Chi Tiết Sản Phẩm

```
CREATE or replace PROCEDURE PRC_INSERT_CTSP (
    input_masp char,
    input_manl char,
    input_soluong number
)
AS
BEGIN
    INSERT INTO CT_SAN_PHAM
    VALUES(input_masp,input_manl,input_soluong);
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');
    EXCEPTION WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');
END;
```

```
CREATE or replace PROCEDURE PRC_INSERT_CTTHUNG (  
    input_mathung char,  
    input_manl char,  
    input_soluong number  
)  
AS  
BEGIN  
    INSERT INTO CT_THUNG  
    VALUES(input_mathung,input_manl,input_soluong);  
    DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
    EXCEPTION WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('INSERT UNSUCCESSFULLY');  
END;
```

```

-- CALL INSERT NHANVIEN
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_NV(&CMND,&HOLOT,&TENNV,&TUOI,&GIOI_TINH,&SDT,&CHUC_VU);
END;

-- CALL INSERT CT_NHANVIEN
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_CT_NV(&MaNV,&Ca,&MaNQL,&MaCoSo);
END;

-- CALL INSERT CO_SO
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_COSO(&TenCoSo,&DiaChi,&TinhTrang,&Loai);
END;

-- CALL INSERT KHACH_HANG
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_KH(&SDT,&HOLOT,&TENKH);
END;

-- CALL INSERT HOA_DON
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_HD(&NgayLapPhieu,&TongTien,&MaNV,&MaKH);
END;

```

```

-- CALL INSERT DVT
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_DVT (&TenDVT);
END;

-- CALL INSERT LOAI_SP
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_LSP (&TenLSP);
END;

-- CALL INSERT SAN_PHAM
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_SP (&TENS, &MALOAI, &MADVT);
END;

-- CALL INSERT CT_HOA_DON
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_CTHD (&SOHD, &MASP, &SOLUONG);
END;

-- CALL INSERT BIEN_DONG_GIA
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_BDG (&MASP, &MANV, &DONGIA, &NGAYCAPNHAT);
END;

```

```

-- CALL INSERT PHIEU_XUAT
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_PX (&NGAY_LAP_PHIEU, &MANV, &CO_SO_NHAP, &CO_SO_XUAT, &NGAY_XUAT_HANG);
END;

-- CALL INSERT PHIEU_NHAP
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_PN (&NGAY_LAP_PHIEU, &MANV, &CO_SO_NHAP, &CO_SO_XUAT, &NGAY_NHAP_HANG);
END;

-- CALL INSERT TAI_XE
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_TX (&TENTX, &CA, &TUYEN);
END;

-- CALL INSERT XE_VAN_CHUYEN
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_XVC (&SOXE, &LOAIXE, &HANG);
END;

-- CALL INSERT PHAN_CONG
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_PC (&MATX, &SOXE);
END;

```

```

-- CALL INSERT LO_HANG
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_LH(&tongSoThung, &SoXe);
END;

-- CALL INSERT THUNG
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_TH(&Tinhtrang);
END;

-- CALL INSERT CT_LO_HANG
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_CTLH(&MALO, &MATHUNG);
END;

-- CALL INSERT CT_PHIEU_NHAP
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_CTPN(&MAPHIEUNHAP, &MALO, &THOIHAN);
END;

-- CALL INSERT CT_PHIEU_XUAT
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_CTPX(&MAPHIEUXUAT, &MALO);
END;

```

```

-- CALL INSERT NHA_CUNG_CAP
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_NCC (&TEN_NCC, &DIACHI, &SDT, &TINH_TRANG);
END;

-- CALL INSERT LOAI_NL
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_LNL (&TEN_LNL, &MANCC);
END;

-- CALL INSERT NGUYEN_LIEU
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_NL (&TEN_NL, &DON_GIA, &MALNL, &MADVT);
END;

-- CALL INSERT CT_SAN_PHAM
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_CTSP (&MASP, &MANL, &SOLUONG);
END;

-- CALL INSERT CT_THUNG
SET SERVEROUTPUT ON;
BEGIN
    PRC_INSERT_CTTHUNG (&MATHUNG, &MANL, &SOLUONG);
END;

```

UPDATE

Phạm Gia Khương – thủ tục update Cơ Sở

```
CREATE OR REPLACE PROCEDURE PRC_UPDATE_COSO (input_macs CHAR, input_tinhtrang NVARCHAR2)
AS
BEGIN
    UPDATE CO_SO
    SET TINH_TRANG = input_tinhtrang
    WHERE MA_CO_SO = input_macs;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_UPDATE_COSO (&MA_CO_SO, &TINH_TRANG);
END;
```

Trần Văn Minh – thủ tục update Phiếu Nhập

```
CREATE OR REPLACE PROCEDURE PRC_UPDATE_PHIEUNHAP (input_mapn CHAR, input_ngaynhap DATE)
AS
BEGIN
    IF input_ngaynhap IS NULL THEN
        UPDATE PHIEU_NHAP
        SET NGAY_NHAP_HANG = SYSDATE
        WHERE MA_PHIEU_NHAP = input_mapn;
    ELSE
        UPDATE PHIEU_NHAP
        SET NGAY_NHAP_HANG = input_ngaynhap
        WHERE MA_PHIEU_NHAP = input_mapn;
    END IF;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_UPDATE_PHIEUNHAP (&MA_PHIEU_NHAP, &NGAY_NHAP_HANG);
END;
```


Nguyễn Văn Bình Minh – thủ tục update Nhân Viên

```
CREATE OR REPLACE PROCEDURE PRC_UPDATE_NV (input_manv CHAR, input_tuoi NUMBER, input_phone NUMBER, input_chucvu NVARCHAR2)
AS
BEGIN
--000
IF input_tuoi IS NOT NULL AND input_phone IS NOT NULL AND input_chucvu IS NOT NULL
THEN
    UPDATE NHAN_VIEN
    SET TUOI=input_tuoi,
    SDT=input_phone,
    CHUC_VU= input_chucvu
    WHERE MANV=input_manv;
--001
ELSIF input_tuoi IS NOT NULL AND input_phone IS NOT NULL AND input_chucvu IS NULL
THEN
    UPDATE NHAN_VIEN
    SET TUOI=input_tuoi,
    SDT=input_phone
    WHERE MANV=input_manv;
--011
ELSIF input_tuoi IS NOT NULL AND input_phone IS NULL AND input_chucvu IS NULL
THEN
    UPDATE NHAN_VIEN
    SET TUOI=input_tuoi
    WHERE MANV=input_manv;
--111
ELSIF input_tuoi IS NULL AND input_phone IS NULL AND input_chucvu IS NULL
THEN
    raise_application_error (-20001,'Dữ liệu chưa thay đổi');
--110
ELSIF input_tuoi IS NULL AND input_phone IS NULL AND input_chucvu IS NOT NULL
THEN
    UPDATE NHAN_VIEN
    SET
    CHUC_VU= input_chucvu
    WHERE MANV=input_manv;
--100
ELSIF input_tuoi IS NULL AND input_phone IS NOT NULL AND input_chucvu IS NOT NULL
THEN
    UPDATE NHAN_VIEN
    SET
    SDT=input_phone,
    CHUC_VU= input_chucvu
    WHERE MANV=input_manv;
--101
ELSIF input_tuoi IS NULL AND input_phone IS NOT NULL AND input_chucvu IS NULL
THEN
    UPDATE NHAN_VIEN
    SET
    SDT=input_phone
    WHERE MANV=input_manv;
--010
ELSIF input_tuoi IS NOT NULL AND input_phone IS NULL AND input_chucvu IS NOT NULL
THEN
    UPDATE NHAN_VIEN
    SET TUOI=input_tuoi,
    CHUC_VU= input_chucvu
    WHERE MANV=input_manv;
END IF;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_UPDATE_NV (&MANV, &TUOI, &SDT, &CHUC_VU );
END;
```

Nguyễn Văn Bình Minh – thủ tục update Chi Tiết Nhân Viên

```
create or replace PROCEDURE PRC_UPDATE_CTNV (input_manv CHAR, input_ca NUMBER, input_nql CHAR, input_macs CHAR)
AS
BEGIN
--000
IF input_ca IS NOT NULL AND input_nql IS NOT NULL AND input_macs IS NOT NULL
THEN
    UPDATE CHI_TIET_NHAN_VIEN
    SET CA = input_ca,
        MA_NQL = input_nql,
        MA_CO_SO = input_macs
    WHERE MANV=input_manv;
--001
ELSIF input_ca IS NOT NULL AND input_nql IS NOT NULL AND input_macs IS NULL
THEN
    UPDATE CHI_TIET_NHAN_VIEN
    SET CA = input_ca,
        MA_NQL = input_nql
    WHERE MANV=input_manv;
--011
ELSIF input_ca IS NOT NULL AND input_nql IS NULL AND input_macs IS NULL
THEN
    UPDATE CHI_TIET_NHAN_VIEN
    SET CA = input_ca
    WHERE MANV=input_manv;
--111
ELSIF input_ca IS NULL AND input_nql IS NULL AND input_macs IS NULL
THEN
    raise_application_error (-20001,'Dữ liệu chưa thay đổi');

--110
ELSIF input_ca IS NULL AND input_nql IS NULL AND input_macs IS NOT NULL
THEN
    UPDATE CHI_TIET_NHAN_VIEN
    SET
        MA_CO_SO = input_macs
    WHERE MANV=input_manv;
--100
ELSIF input_ca IS NULL AND input_nql IS NOT NULL AND input_macs IS NOT NULL
THEN
    UPDATE CHI_TIET_NHAN_VIEN
    SET
        MA_NQL = input_nql,
        MA_CO_SO = input_macs
    WHERE MANV=input_manv;
--101
ELSIF input_ca IS NULL AND input_nql IS NOT NULL AND input_macs IS NULL
THEN
    UPDATE CHI_TIET_NHAN_VIEN
    SET
        MA_NQL = input_nql
    WHERE MANV=input_manv;

--010
ELSIF input_ca IS NOT NULL AND input_nql IS NULL AND input_macs IS NOT NULL
THEN
    UPDATE CHI_TIET_NHAN_VIEN
    SET CA = input_ca,
        MA_CO_SO = input_macs
    WHERE MANV=input_manv;
END IF;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_UPDATE_CTNV (input_manv, input_ca, input_nql, input_macs );
END;
```

Trần Văn Minh – thủ tục update Phiếu Xuất

```
CREATE OR REPLACE PROCEDURE PRC_UPDATE_PHIEUXUAT (input_mapx CHAR, input_ngayxuat DATE)
AS
BEGIN
IF input_ngayxuat IS NULL THEN
    UPDATE PHIEU_XUAT
    SET  NGAY_XUAT_HANG = SYSDATE
    WHERE MA_PHIEU_XUAT = input_mapx;
ELSE
    UPDATE PHIEU_XUAT
    SET  NGAY_XUAT_HANG = input_ngayxuat
    WHERE MA_PHIEU_XUAT = input_mapx;
END IF;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_UPDATE_PHIEUXUAT (&MA_PHIEU_XUAT,&NGAY_XUAT_HANG);
END;
```

Phạm Gia Khương – thủ tục update Chi Tiết Hóa Đơn

```
CREATE OR REPLACE PROCEDURE PRC_UPDATE_CTHOADON (input_sohd CHAR, input_masp CHAR, input_soluong NUMBER)
AS
BEGIN
    UPDATE CT_HOA_DON
    SET  MASP = input_masp,
        SO_LUONG = input_soluong
    WHERE SOHD = input_sohd;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_UPDATE_CTHOADON (&SOHD,&MASP,&SO_LUONG);
END;
```

Phạm Gia Khương – thủ tục update Thùng

```
CREATE OR REPLACE PROCEDURE PRC_UPDATE_THUNG (input_mathung CHAR, input_tinhtrang VARCHAR2)
AS
BEGIN
    UPDATE THUNG
    SET
        TINH_TRANG = input_tinhtrang
    WHERE MA_THUNG = input_mathung;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_UPDATE_THUNG (&MA_THUNG,&TINH_TRANG);
END;
```

Nguyễn Văn Bình Minh – thủ tục Get Tồn Kho

```
CREATE TYPE TYPE_TON_KHO IS OBJECT (Thang VARCHAR2(5), Nam VARCHAR2(5), Ma_nguyen_lieu CHAR(7),  
Ten_nguyen_lieu NVARCHAR2(30), So_luong_ton NUMBER(10));  
  
CREATE TYPE TABLE_TON_KHO IS TABLE OF TYPE_TON_KHO;
```

```
create or replace PROCEDURE PRC_GET_TON_KHO  
AS  
    RETURN_TON_KHO TABLE_TON_KHO;  
BEGIN  
    SELECT TYPE_TON_KHO (THANG, NAM, MANL, TENNL, SO_LUONG_TON) BULK COLLECT  
    INTO RETURN_TON_KHO  
    FROM V_TON_KHO;  
    FOR X IN 1..RETURN_TON_KHO.COUNT  
    LOOP  
        declare  
            c integer;  
        begin  
            C :=0;  
            select count(*) into c from TON_KHO  
            where (CONCAT(RETURN_TON_KHO(X).THANG, CONCAT(RETURN_TON_KHO(X).NAM, RETURN_TON_KHO(X).Ma_nguyen_lieu)))  
            LIKE (CONCAT(TON_KHO.THANG, CONCAT(TON_KHO.NAM, TON_KHO.Ma_nguyen_lieu)));  
            if c = 0 then  
                INSERT INTO TON_KHO  
                VALUES (RETURN_TON_KHO(X).THANG, RETURN_TON_KHO(X).NAM, RETURN_TON_KHO(X).Ma_nguyen_lieu,  
                    RETURN_TON_KHO(X).Ten_nguyen_lieu, RETURN_TON_KHO(X).So_luong_ton);  
            end if;  
        end;  
    END LOOP;  
END;
```

Nguyễn Văn Bình Minh – thủ tục Update thời hạn tồn

```
create or replace PROCEDURE PRC_UPDATE_THOI_HAN_CTPN (  
    input_mapn char,  
    input_malo char,  
    input_thoihan number  
)  
AS  
BEGIN  
    IF input_thoihan > 5  
    THEN  
        DBMS_OUTPUT.PUT_LINE('UPDATE UNSUCCESSFULLY, thoi han phai <= 5');  
    ELSE  
        UPDATE CT_PHIEU_NHAP  
        SET THOI_HAN = input_thoihan  
        WHERE MA_PHIEU_NHAP LIKE input_mapn AND MALO LIKE 'input_malo';  
        DBMS_OUTPUT.PUT_LINE('INSERT SUCCESS');  
    END IF;  
END;
```

DELETE

Trần Văn Minh – thủ tục delete Nhân Viên

```
--DELETE NHAN_VIEN
CREATE OR REPLACE PROCEDURE PRC_DEL_NHAN_VIEN (input_manv CHAR)
AS
BEGIN
UPDATE NHAN_VIEN
SET DEL_STATUS = 'DELETED'
WHERE MANV LIKE input_manv;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_NHAN_VIEN (&MANV);
END;
```

Trần Văn Minh – thủ tục delete Cơ Sở

```
--DELETE CO_SO
CREATE OR REPLACE PROCEDURE PRC_DEL_CO_SO (input_macs CHAR)
AS
BEGIN
UPDATE CO_SO
SET DEL_STATUS = 'DELETED'
WHERE MA_CO_SO LIKE input_macs;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_CO_SO (&MA_CO_SO);
END;
```

Phạm Gia Khương – Thủ tục Delete Nhà Cung Cấp

```
--DELETE NCC
CREATE OR REPLACE PROCEDURE PRC_DEL_NCC (input_mancc CHAR)
AS
BEGIN
UPDATE NHA_CUNG_CAP
SET DEL_STATUS = 'DELETED'
WHERE MANCC LIKE input_mancc;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_NCC (&MANCC);
END;
```

Phạm Gia Khương – Thủ tục Delete Hóa Đơn

```
--DELETE HOA DON
CREATE OR REPLACE PROCEDURE PRC_DEL_HD (input_mahd CHAR)
AS
BEGIN
UPDATE HOA_DON
SET DEL_STATUS = 'DELETED'
WHERE SOHD LIKE input_mahd;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_HD (&SOHD);
END;
```

Trần Văn Minh – Thủ tục delete Khách Hàng

```
--DELETE KHACH_HANG
CREATE OR REPLACE PROCEDURE PRC_DEL_KH (input_makh CHAR)
AS
BEGIN
UPDATE KHACH_HANG
SET DEL_STATUS = 'DELETED'
WHERE MAKH LIKE input_makh;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_KH (&MAKH);
END;
```

Nguyễn Văn Bình Minh – thủ tục delete Loại Sản Phẩm

```
--DELETE LOAI_SAN_PHAM
CREATE OR REPLACE PROCEDURE PRC_DEL_LSP (input_malsp CHAR)
AS
BEGIN
UPDATE LOAI_SAN_PHAM
SET DEL_STATUS = 'DELETED'
WHERE MALSP LIKE input_malsp;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_LSP (&MALSP);
END;
```

Nguyễn Văn Bình Minh – thủ tục delete Sản Phẩm

```
--DELETE SAN_PHAM
CREATE OR REPLACE PROCEDURE PRC_DEL_SP (input_masp CHAR)
AS
BEGIN
UPDATE SAN_PHAM
SET DEL_STATUS = 'DELETED'
WHERE MASP LIKE input_masp;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_SP (&MASP);
END;
```

Nguyễn Văn Bình Minh – thủ tục delete Loại Nguyên Liệu

```
--DELETE LOAI_NGUYEN_LIEU
CREATE OR REPLACE PROCEDURE PRC_DEL_LNL (input_malnl CHAR)
AS
BEGIN
UPDATE LOAI_NGUYEN_LIEU
SET DEL_STATUS = 'DELETED'
WHERE MALNL LIKE input_malnl;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_LNL (&MALNL);
END;
```

Phạm Gia Khương - thủ tục delete Nguyên Liệu

```
--DELETE NGUYEN_LIEU
CREATE OR REPLACE PROCEDURE PRC_DEL_NL (input_manl CHAR)
AS
BEGIN
UPDATE NGUYEN_LIEU
SET DEL_STATUS = 'DELETED'
WHERE MANL LIKE input_manl;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_NL (&MANL);
END;
```

Phạm Gia Khương – thủ tục delete Thùng

```
--DELETE THUNG
CREATE OR REPLACE PROCEDURE PRC_DEL_TH (input_mathung CHAR)
AS
BEGIN
UPDATE THUNG
SET DEL_STATUS = 'DELETED'
WHERE MA_THUNG LIKE input_mathung;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_TH (&MA_THUNG);
END;
```


Phạm Gia Khương – thủ tục delete Lô Hàng

```
--DELETE LO_HANG
CREATE OR REPLACE PROCEDURE PRC_DEL_LH (input_malo CHAR)
AS
BEGIN
UPDATE LO_HANG
SET DEL_STATUS = 'DELETED'
WHERE MALO LIKE input_malo;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_LH (&MALO);
END;
```

Trần Văn Minh – thủ tục delete Tài Xế

```
--DELETE TAI_XE
CREATE OR REPLACE PROCEDURE PRC_DEL_TX (input_matx CHAR)
AS
BEGIN
UPDATE TAI_XE
SET DEL_STATUS = 'DELETED'
WHERE MA_TAI_XE LIKE input_matx;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_TX (&MA_TAI_XE);
END;
```

Phạm Gia Khương – thủ tục delete Xe Vận Chuyển

```
--DELETE XE_VAN_CHUYEN
CREATE OR REPLACE PROCEDURE PRC_DEL_XVC (input_maxvc CHAR)
AS
BEGIN
UPDATE XE_VAN_CHUYEN
SET DEL_STATUS = 'DELETED'
WHERE SO_XE LIKE input_maxvc;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_XVC (&SO_XE);
END;
```

Trần Văn Minh - thủ tục delete Phiếu Nhập

```
--DELETE PHIEU_NHAP
CREATE OR REPLACE PROCEDURE PRC_DEL_PN (input_mapn CHAR)
AS
BEGIN
UPDATE PHIEU_NHAP
SET DEL_STATUS = 'DELETED'
WHERE MA_PHIEU_NHAP LIKE input_mapn;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_PN (&MA_PHIEU_NHAP);
END;
```

Trần Văn Minh – thủ tục delete Phiếu Xuất

```
--DELETE PHIEU_XUAT
CREATE OR REPLACE PROCEDURE PRC_DEL_PX (input_mapx CHAR)
AS
BEGIN
UPDATE PHIEU_XUAT
SET DEL_STATUS = 'DELETED'
WHERE MA_PHIEU_XUAT LIKE input_mapx;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_PX (&MA_PHIEU_XUAT);
END;
```

Phạm Gia Khương - thủ tục delete Đơn Vị Tính

```
--DELETE DON_VI_TINH
CREATE OR REPLACE PROCEDURE PRC_DEL_DVT (input_madvn CHAR)
AS
BEGIN
UPDATE DON_VI_TINH
SET DEL_STATUS = 'DELETED'
WHERE MADVT LIKE input_madvn;
END;
SET SERVEROUTPUT ON;
BEGIN
    PRC_DEL_DVT (&MADVT);
END;
```

5. Hàm (function):

Nguyễn Văn Bình Minh – Hàm tạo mã 5 số 0

```
--5 NUMBERS
CREATE OR REPLACE FUNCTION numberZerosString(
  n IN NUMBER
) RETURN VARCHAR2
AS
BEGIN
  RETURN LPAD( n, 5, '0' );
END;
```

Nguyễn Văn Bình Minh – Hàm tạo mã 4 số 0

```
--4 NUMBERS
CREATE OR REPLACE FUNCTION numberFourZerosString(
  n IN NUMBER
) RETURN VARCHAR2
AS
BEGIN
  RETURN LPAD( n, 4, '0' );
END;
```

Phạm Gia Khương - Hàm CTHOADON

```
--CTHOADON
CREATE OR REPLACE FUNCTION GET_LATEST_PRICE (N IN CHAR, M IN DATE) RETURN FLOAT
AS
R FLOAT;
BEGIN
  SELECT *
  INTO R
  FROM
    (SELECT DON_GIA FROM Bien_Dong_Gia
     WHERE N LIKE Bien_Dong_Gia.MASP AND M >= NGAY_CAP_NHAT
     ORDER BY Ngay_cap_nhat desc)
  WHERE ROWNUM = 1;
RETURN R;
END;
```

Trần Văn Minh – Hàm Ngày Lập Hóa Đơn

```
CREATE OR REPLACE FUNCTION GET_NGAY_LAP_HD (N IN CHAR) RETURN DATE
AS
R DATE;
BEGIN
    SELECT NGAY_LAP_HD
    INTO R
    FROM HOA_DON
    WHERE N LIKE SOHD;
    RETURN R;
END;
```

6. Trigger:

Phạm Gia Khương – trigger tạo Mã Nhân Viên tự động

```
--1 NHAN VIEN-----
CREATE SEQUENCE SEQ_NV_ID MAXVALUE 99999 START WITH 1;

--TRIGGER NV
CREATE OR REPLACE TRIGGER TRIG_INSERT_NV
BEFORE INSERT
ON NHAN_VIEN
FOR EACH ROW
DECLARE
    input_manv CHAR(7);
BEGIN
    :NEW.MANV := TO_CHAR(concat('NV', numberZeroString(SEQ_NV_ID.NEXTVAL)));
END;
```

Phạm Gia Khương – trigger tạo Mã Cơ Sở tự động

```
--3 CO SO
CREATE SEQUENCE SEQ_KHO_ID MAXVALUE 99999 START WITH 00001;
CREATE SEQUENCE SEQ_KHOLANH_ID MAXVALUE 99999 START WITH 00001;
CREATE SEQUENCE SEQ_KHOPHANPHOI_ID MAXVALUE 99999 START WITH 00001;
CREATE SEQUENCE SEQ_KHOCHINHANH_ID MAXVALUE 99999 START WITH 00001;

--TRIGGER
CREATE OR REPLACE TRIGGER TRIG_INSERT_CS
BEFORE INSERT
ON CO_SO
FOR EACH ROW
DECLARE
    input CHAR(7);
BEGIN
    IF :NEW.LOAI_CO_SO LIKE N'Kho'
    THEN
        :NEW.MA_CO_SO := TO_CHAR(concat('KH', numberZeroString(SEQ_KHO_ID.NEXTVAL)));

    ELSIF :NEW.LOAI_CO_SO LIKE N'Kho lạnh'
    THEN
        :NEW.MA_CO_SO := TO_CHAR(concat('KL', numberZeroString(SEQ_KHOLANH_ID.NEXTVAL)));

    ELSIF :NEW.LOAI_CO_SO LIKE N'Kho trung tâm phân phối'
    THEN
        :NEW.MA_CO_SO := TO_CHAR(concat('PP', numberZeroString(SEQ_KHOPHANPHOI_ID.NEXTVAL)));

    ELSIF :NEW.LOAI_CO_SO LIKE N'Chi nhánh'
    THEN
        :NEW.MA_CO_SO := TO_CHAR(concat('CN', numberZeroString(SEQ_KHOCHINHANH_ID.NEXTVAL)));

    ELSE
        :NEW.MA_CO_SO := TO_CHAR(concat('KH', numberZeroString(SEQ_KHO_ID.NEXTVAL)));
    END IF;
END;
```

```
--4 KHACH HANG
CREATE SEQUENCE SEQ_KH_ID MAXVALUE 99999 START WITH 1;

--trigger
CREATE OR REPLACE TRIGGER TRIG_INSERT_KH
BEFORE INSERT
ON KHACH_HANG
FOR EACH ROW
DECLARE
    input CHAR(7);
BEGIN
    :NEW.MAKH := TO_CHAR(concat('KH', numberZerosString(SEQ_KH_ID.NEXTVAL)));
END;
```

Trần Văn Minh – trigger Hóa Đơn

```
--5 HOA DON
CREATE SEQUENCE SEQ_HD_ID MAXVALUE 99999 START WITH 1;

--trigger
CREATE OR REPLACE TRIGGER TRIG_INSERT_HD
BEFORE INSERT
ON HOA_DON
FOR EACH ROW
DECLARE

BEGIN
    :NEW.SOHD := TO_CHAR(concat('HD', numberZerosString(SEQ_HD_ID.NEXTVAL)));
END;

--TRIGGER Cam chinh sua HD
CREATE OR REPLACE TRIGGER TRIG_ALTER_HD
BEFORE INSERT OR UPDATE
ON HOA_DON
FOR EACH ROW
BEGIN
    IF (TRUNC(SYSDATE) > TRUNC(:OLD.NGAY_LAP_HD))
    THEN RAISE_APPLICATION_ERROR(-20011, 'You can not alter this form anymore');
    end if;
END;
```

```
--TRIGGER Cam chinh sua CT_HD
CREATE OR REPLACE TRIGGER TRIG_ALTER_CT_HD
BEFORE INSERT OR UPDATE
ON CT_HOA_DON
FOR EACH ROW
BEGIN
    IF (TRUNC(SYSDATE) > TRUNC(GET_NGAY_LAP_HD(:NEW.SOHD)))
    THEN RAISE_APPLICATION_ERROR(-20011, 'You can not alter this form anymore');
    end if;
END;
```

Phạm Gia Khương – trigger Mã Đơn Vị Tính tự động

```

--6 DON VI TINH
CREATE SEQUENCE SEQ_DVT_ID MAXVALUE 99999 START WITH 1;

--TRIGGER
CREATE OR REPLACE TRIGGER TRIG_INSERT_DVT
BEFORE INSERT
ON DON_VI_TINH
FOR EACH ROW
BEGIN
    :NEW.MADVT := TO_CHAR(concat('DVT', numberFourZerosString(SEQ_DVT_ID.NEXTVAL)));
END;

```

Phạm Gia Khương – trigger tạo Mã Loại Sản Phẩm tự động

```

--7 LOAI SAN PHAM
CREATE SEQUENCE SEQ_LSP_ID MAXVALUE 99999 START WITH 1;

--TRIGGER
CREATE OR REPLACE TRIGGER TRIG_INSERT_LSP
BEFORE INSERT
ON LOAI_SAN_PHAM
FOR EACH ROW
DECLARE

BEGIN
    :NEW.MALSP := TO_CHAR(concat('LSP', numberFourZerosString(SEQ_LSP_ID.NEXTVAL)));
END;

```


Phạm Gia Khương – trigger tạo Mã Sản Phẩm tự động

```
--8 SAN PHAM
CREATE SEQUENCE SEQ_SP_ID MAXVALUE 99999 START WITH 1;

--TRIGGER insert
CREATE OR REPLACE TRIGGER TRIG_INSERT_SP
BEFORE INSERT
ON SAN_PHAM
FOR EACH ROW
DECLARE

BEGIN
    :NEW.MASP := TO_CHAR(concat('SP', numberZerosString(SEQ_SP_ID.NEXTVAL)));
END;
```

Nguyễn Văn Bình Minh – trigger Chi Tiết Hóa Đơn: tự động tính Thành Tiền

```
--9 CHITIETHOADON
--TRIGGER
CREATE OR REPLACE TRIGGER TRIG_CTHD_THANH_TIEN
BEFORE INSERT OR UPDATE ON CT_HOA_DON
FOR EACH ROW
BEGIN
    :NEW.THANH_TIEN := (:NEW.SO_LUONG * GET_LATEST_PRICE(:NEW.MASP, GET_NGAY_LAP_HD(:NEW.SOHD)));
END;
```

Trần Văn Minh – trigger tạo Mã Phiếu Xuất tự động

```
--11 PHIEU XUAT
CREATE SEQUENCE SEQ_PX_ID MAXVALUE 99999 START WITH 1;

--TRIGGER
CREATE OR REPLACE TRIGGER TRIG_INSERT_PX
BEFORE INSERT
ON PHIEU_XUAT
FOR EACH ROW
DECLARE

BEGIN
    :NEW.MA_PHIEU_XUAT := TO_CHAR(concat('PX', numberZerosString(SEQ_PX_ID.NEXTVAL)));
END;
```

Trần Văn Minh – trigger tạo Mã Phiếu Nhập tự động

```
--12 PHIEU NHAP
CREATE SEQUENCE SEQ_PN_ID MAXVALUE 99999 START WITH 1;

--TRIGGER
CREATE OR REPLACE TRIGGER TRIG_INSERT_PN
BEFORE INSERT
ON PHIEU_NHAP
FOR EACH ROW
DECLARE

BEGIN
    :NEW.MA_PHIEU_NHAP := TO_CHAR(concat('PN', numberZerosString(SEQ_PN_ID.NEXTVAL)));
END;
```

Nguyễn Văn Bình Minh – trigger tạo Mã Tài Xế tự động

```
--13 TAI XE
CREATE SEQUENCE SEQ_TX_ID MAXVALUE 99999 START WITH 1;

--TRIGGER
CREATE OR REPLACE TRIGGER TRIG_INSERT_TX
BEFORE INSERT
ON TAI_XE
FOR EACH ROW
DECLARE

BEGIN
    :NEW.MA_TAI_XE := TO_CHAR(concat('TX', numberZerosString(SEQ_TX_ID.NEXTVAL)));
END;
```

Nguyễn Văn Bình Minh – trigger tạo Mã Lô Hàng tự động

```
--16 LO HANG
CREATE SEQUENCE SEQ_LH_ID MAXVALUE 99999 START WITH 1;

--TRIGGER
CREATE OR REPLACE TRIGGER TRIG_INSERT_LH
BEFORE INSERT
ON LO_HANG
FOR EACH ROW
DECLARE

BEGIN
    :NEW.MALO := TO_CHAR(concat('LH', numberZeroString(SEQ_LH_ID.NEXTVAL)));
END;
```

Phạm Gia Khương – trigger tạo Mã Thùng tự động

```
--17 THUNG
CREATE SEQUENCE SEQ_TH_ID MAXVALUE 99999 START WITH 1;

--TRIGGER
CREATE OR REPLACE TRIGGER TRIG_INSERT_TH
BEFORE INSERT
ON THUNG
FOR EACH ROW
DECLARE

BEGIN
    :NEW.MA_THUNG := TO_CHAR(concat('TH', numberZeroString(SEQ_TH_ID.NEXTVAL)));
END;
```

Phạm Gia Khương – trigger tạo Mã Nhà Cung Cấp tự động

```
--21 NHA CUNG CAP
CREATE SEQUENCE SEQ_NCC_ID MAXVALUE 99999 START WITH 1;

--TRIGGER
CREATE OR REPLACE TRIGGER TRIG_INSERT_NCC
BEFORE INSERT
ON NHA_CUNG_CAP
FOR EACH ROW
DECLARE

BEGIN
    :NEW.MANCC := TO_CHAR(concat('NCC', numberFourZerosString(SEQ_NCC_ID.NEXTVAL)));
END;
```

Trần Văn Minh – trigger tạo Mã Loại Nguyên Liệu tự động

```
--22 LOAI NGUYEN LIEU
CREATE SEQUENCE SEQ_LNL_ID MAXVALUE 99999 START WITH 1;

--TRIGGER
CREATE OR REPLACE TRIGGER TRIG_INSERT_LNL
BEFORE INSERT
ON LOAI_NGUYEN_LIEU
FOR EACH ROW
DECLARE

BEGIN
    :NEW.MALNL := TO_CHAR(concat('LNL', numberFourZerosString(SEQ_LNL_ID.NEXTVAL)));
END;
```

Trần Văn Minh – trigger tạo Mã Nguyên Liệu tự động

```
--23 NGUYEN LIEU
CREATE SEQUENCE SEQ_NL_ID MAXVALUE 99999 START WITH 1;

--TRIGGER
CREATE OR REPLACE TRIGGER TRIG_INSERT_NL
BEFORE INSERT
ON NGUYEN_LIEU
FOR EACH ROW
DECLARE

BEGIN
    :NEW.MANL := TO_CHAR(concat('NL', numberZeroString(SEQ_NL_ID.NEXTVAL)));
END;
```

Phạm Gia Khương – trigger cập nhật Chức Vụ Nhân Viên

```
--TRIGGER UPDATE NHAN_VIEN
CREATE OR REPLACE TRIGGER TRIG_UPDATE_NV
BEFORE UPDATE
ON NHAN_VIEN
FOR EACH ROW
BEGIN
    IF :NEW.CHUC_VU NOT LIKE 'Nhân viên' AND :NEW.CHUC_VU NOT LIKE 'Quản lý'
    THEN raise_application_error (-20001, 'Chức vụ không tồn tại');
    END IF;
END;
```

Phạm Gia Khương – trigger cập nhật Tình Trạng Cơ Sở

```
--TRIGGER UPDATE CO_SO
CREATE OR REPLACE TRIGGER TRIG_UPDATE_CS
BEFORE UPDATE
ON CO_SO
FOR EACH ROW
BEGIN
    IF :NEW.TINH_TRANG NOT LIKE 'Hoạt động' AND :NEW.TINH_TRANG NOT LIKE 'Ngưng hoạt động'
    THEN raise_application_error (-20001, 'Tình trạng không tồn tại');
    END IF;
END;
```

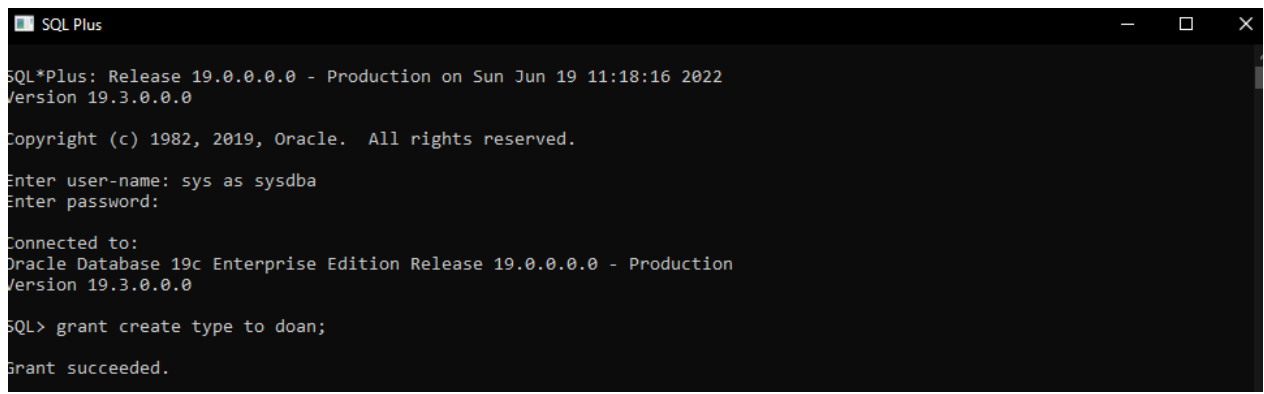
Nguyễn Văn Bình Minh - trigger cập nhật Tình Trạng Thùng

```
--TRIGGER UPDATE THUNG
CREATE OR REPLACE TRIGGER TRIG_UPDATE_THUNG
BEFORE UPDATE
ON THUNG
FOR EACH ROW
BEGIN
    IF :NEW.TINH_TRANG LIKE 'Hỏng'
    THEN
        UPDATE CT_THUNG
        SET TONG_SO_NL = 0
        WHERE MA_THUNG LIKE :NEW.MA_THUNG;
    END IF;
END;
```

Nguyễn Văn Bình Minh - trigger cập nhật tồn kho

```
create or replace TRIGGER TRIG_UPDATE_TON_KHO
AFTER INSERT OR UPDATE ON CT_THUNG
FOR EACH ROW
BEGIN
    PRC_GET_TON_KHO;
END;
```

7. Tài khoản người dùng:



```
SQL*Plus
SQL*Plus: Release 19.0.0.0.0 - Production on Sun Jun 19 11:18:16 2022
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Enter user-name: sys as sysdba
Enter password:

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> grant create type to doan;

Grant succeeded.
```

```
create user BinhMinh identified by BinhMinh1999;
create user VanMinh identified by VanMinh2001;
create user GiaKhuong identified by khuong2001;
```

8. Cấp quyền sử dụng người dùng:

```
grant select on Doan.NHAN_VIEN to BinhMinh;  
grant update, delete on Doan.SAN_PHAM to BinhMinh;  
grant insert on Doan.BIEN_DONG_GIA to BinhMinh;  
grant select, update on Doan.CO_SO to BinhMinh;  
grant select on Doan.V_HOA_DON to BinhMinh;  
grant insert, delete on Doan.CHI_PHI to BinhMinh;  
grant insert, update, delete on Doan.TONG_THU to BinhMinh;  
grant select, insert, update, delete on Doan.V_HD_TONG_TIEN to BinhMinh;
```

```
grant select, insert on Doan.HOA_DON to VanMinh;  
grant update on Doan.CT_HOA_DON to VanMinh;  
grant delete on Doan.XE_VAN_CHUYEN to VanMinh;  
grant select, insert, update on Doan.NHA_CUNG_CAP to VanMinh;  
grant update, delete on Doan.CONG_THUC_SAN_PHAM to VanMinh;  
grant select, delete on Doan.DANH_SACH_HOA_DON to VanMinh;  
grant select, insert, delete on Doan.XEM_TON_KHO to VanMinh;  
grant insert, update on Doan.DANH_SACH_NHAN_VIEN to VanMinh;
```

```
grant select, insert, update, delete on Doan.KHACH_HANG to GiaKhuong;  
grant update on Doan.PHIEU_NHAP to GiaKhuong;  
grant select, insert, update on Doan.NGUYEN_LIEU to GiaKhuong;  
grant select, delete on Doan.PHIEU_XUAT to GiaKhuong;  
grant insert, update on Doan.DANH_SACH_SAN_PHAM to GiaKhuong;  
grant select, update, delete on Doan.LOI_NHUAN to GiaKhuong;  
grant update, delete on Doan.THONG_TIN_PHIEU_NHAP to GiaKhuong;  
grant select, insert, update on Doan.THONG_TIN_CHI_TIET_NHAN_VIEN to GiaKhuong;
```


9. Cấm/thu hồi sử dụng:

```
revoke delete on Doan.SAN_PHAM from BinhMinh;  
revoke update on Doan.CO_SO from BinhMinh;  
revoke delete on Doan.CHI_PHI from BinhMinh;  
revoke select, update on Doan.V_HD_TONG_TIEN from BinhMinh;
```

```
revoke insert on Doan.HOA_DON from VanMinh;  
revoke update on Doan.NHA_CUNG_CAP from VanMinh;  
revoke delete on Doan.DANH_SACH_HOA_DON from VanMinh;  
revoke insert, update on Doan.DANH_SACH_NHAN_VIEN from VanMinh;
```

```
revoke update on Doan.KHACH_HANG from GiaKhuong;  
revoke delete on Doan.PHIEU_XUAT from GiaKhuong;  
revoke delete on Doan.LOI_NHUAN from GiaKhuong;  
revoke insert, update on Doan.THONG_TIN_CHI_TIET_NHAN_VIEN from GiaKhuong;
```

10. Transaction & Backup, Restore :

Transaction kiểm tra nhập mã nhân viên hợp lệ:

```
--TRANSACTION
CREATE OR REPLACE PROCEDURE PRC_CHECK_UPDATE_NV_NHAP_HD(input_sohd char)
AS
COUNT_V NUMBER(1);
BEGIN
SET TRANSACTION READ WRITE NAME 'TRANS_CHECK_UPDATE_NV_NHAP_HD';
UPDATE HOA_DON
SET MANV = input_sohd;
EXCEPTION WHEN OTHERS THEN ROLLBACK;
END;
```

Shutdown và Startup

```
Administrator: Command Prompt - RMAN
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\kenje>RMAN

Recovery Manager: Release 19.0.0.0.0 - Production on Sun Jun 19 17:38:45 2022
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

RMAN> CONNECT TARGET/;

connected to target database: ORCL (DBID=1627962842)

RMAN> SHUTDOWN

using target database control file instead of recovery catalog
database closed
database dismounted
Oracle instance shut down

RMAN>

RMAN> STARTUP MOUNT;

connected to target database (not started)
Oracle instance started
database mounted

Total System Global Area 3204446680 bytes
Fixed Size 9271768 bytes
Variable Size 687865856 bytes
Database Buffers 2499805184 bytes
Redo Buffers 7503872 bytes
```

Backup full database

```

RMAN> BACKUP FULL DATABASE;

Starting backup at 19-JUN-22
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=389 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=D:\ORACLEDB\ORADATA\ORCL\SYSTEM01.DBF
input datafile file number=00003 name=D:\ORACLEDB\ORADATA\ORCL\SYSAUX01.DBF
input datafile file number=00004 name=D:\ORACLEDB\ORADATA\ORCL\UNDOTBS01.DBF
input datafile file number=00007 name=D:\ORACLEDB\ORADATA\ORCL\USERS01.DBF
channel ORA_DISK_1: starting piece 1 at 19-JUN-22
channel ORA_DISK_1: finished piece 1 at 19-JUN-22
piece handle=D:\ORACLEDB\WINDOWS.X64_193000_DB_HOME\DATABASE\0F10FA96_1_1 tag=TAG20220619T174510 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:05
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00010 name=D:\ORACLEDB\ORADATA\ORCL\ORCLPDB\SYSAUX01.DBF
input datafile file number=00009 name=D:\ORACLEDB\ORADATA\ORCL\ORCLPDB\SYSTEM01.DBF
input datafile file number=00011 name=D:\ORACLEDB\ORADATA\ORCL\ORCLPDB\UNDOTBS01.DBF
input datafile file number=00012 name=D:\ORACLEDB\ORADATA\ORCL\ORCLPDB\USERS01.DBF
channel ORA_DISK_1: starting piece 1 at 19-JUN-22
channel ORA_DISK_1: finished piece 1 at 19-JUN-22
piece handle=D:\ORACLEDB\WINDOWS.X64_193000_DB_HOME\DATABASE\0G10FAB8_1_1 tag=TAG20220619T174510 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:26
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00006 name=D:\ORACLEDB\ORADATA\ORCL\PDSEED\SYSAUX01.DBF
input datafile file number=00005 name=D:\ORACLEDB\ORADATA\ORCL\PDSEED\SYSTEM01.DBF
input datafile file number=00008 name=D:\ORACLEDB\ORADATA\ORCL\PDSEED\UNDOTBS01.DBF
channel ORA_DISK_1: starting piece 1 at 19-JUN-22
channel ORA_DISK_1: finished piece 1 at 19-JUN-22
piece handle=D:\ORACLEDB\WINDOWS.X64_193000_DB_HOME\DATABASE\0H10FAC2_1_1 tag=TAG20220619T174510 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:25
Finished backup at 19-JUN-22

Starting Control File and SPFILE Autobackup at 19-JUN-22
piece handle=D:\ORACLEDB\WINDOWS.X64_193000_DB_HOME\DATABASE\C-1627962842-20220619-00 comment=NONE
Finished Control File and SPFILE Autobackup at 19-JUN-22

RMAN>

```

Restore database

```

RMAN> RESTORE DATABASE;

Starting restore at 19-JUN-22
using channel ORA_DISK_1

RMAN> backup database;
Quan sát và ghi nhận thông số của kết quả.

skipping datafile 1; already restored to file D:\ORACLEDB\ORADATA\ORCL\SYSTEM01.DBF
skipping datafile 3; already restored to file D:\ORACLEDB\ORADATA\ORCL\SYSAUX01.DBF
skipping datafile 4; already restored to file D:\ORACLEDB\ORADATA\ORCL\UNDOTBS01.DBFg trong database. Vì dụ
skipping datafile 7; already restored to file D:\ORACLEDB\ORADATA\ORCL\USERS01.DBF
skipping datafile 5; already restored to file D:\ORACLEDB\ORADATA\ORCL\PDSEED\SYSTEM01.DBF
skipping datafile 6; already restored to file D:\ORACLEDB\ORADATA\ORCL\PDSEED\SYSAUX01.DBF
skipping datafile 8; already restored to file D:\ORACLEDB\ORADATA\ORCL\PDSEED\UNDOTBS01.DBF
skipping datafile 9; already restored to file D:\ORACLEDB\ORADATA\ORCL\ORCLPDB\SYSTEM01.DBF các lệnh sau
skipping datafile 10; already restored to file D:\ORACLEDB\ORADATA\ORCL\ORCLPDB\SYSAUX01.DBF
skipping datafile 11; already restored to file D:\ORACLEDB\ORADATA\ORCL\ORCLPDB\UNDOTBS01.DBF
skipping datafile 12; already restored to file D:\ORACLEDB\ORADATA\ORCL\ORCLPDB\USERS01.DBF
restore not done; all files read only, offline, excluded, or already restored
Finished restore at 19-JUN-22

RMAN> restore database;
RMAN> recover database until time "to

```

IV. Tổng kết:

- ❖ Các chức năng chưa thực hiện được:
 - Trigger thời hạn tồn sản phẩm
 - Thủ tục thống kê công nợ
 - Tạo mã khuyến mãi