TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––



**BÁO CÁO BÀI TẬP NHÓM**

MÔN HỌC QUẢN TRỊ CƠ SỞ DỮ LIỆU

***Đề tài:***

**XÂY DỰNG HỆ THỐNG QUẢN LÍ NHẬP VÀ BÁN HÀNG CỦA CỬA HÀNG NT MART**

***Địa chỉ khảo sát: 17B đường Tạ Hiện, Cường Hòa Bắc, Hải Châu, Đà Nẵng***

Lớp : 49K14.1

Nhóm : 10

Thành viên : Nguyễn Đình Khoa

Lê Thị Tường Vy

Lê Thị Thanh

Phạm Thị Thanh Thảo

Trần Thị Thu Hà

**Đà Nẵng,** **04/2025**

**MỤC LỤC**

[1 R1. XÂY DỰNG CƠ SỞ DỮ LIỆU 3](#_Toc195304582)

[2 R2. TẠO MODULE VÀ DUMP DỮ LIỆU 3](#_Toc195304583)

[2.1 Phân chia công việc 3](#_Toc195304584)

[2.2 Chỉnh sửa Database 4](#_Toc195304585)

[2.3 Viết Module 6](#_Toc195304586)

[2.3.1 Bảng DanhMuc 6](#_Toc195304587)

[2.3.1.1 Thêm danh mục mới 6](#_Toc195304588)

[2.3.1.2 Cập nhập thông tin danh mục 7](#_Toc195304589)

[2.3.2 Bảng HangHoa 8](#_Toc195304590)

[2.3.2.1 Thêm hàng hóa mới 8](#_Toc195304591)

[2.3.2.2 Cập nhập thông tin hàng hóa 9](#_Toc195304592)

[2.3.3 Bảng GiaHangHoa 10](#_Toc195304593)

[2.3.3.1 Thêm giá hàng hóa mới 10](#_Toc195304594)

[2.3.4 Bảng NhaCungCap 11](#_Toc195304595)

[2.3.4.1 Thêm nhà cung cấp mới 11](#_Toc195304596)

[2.3.4.2 Cập nhập thông tin nhà cung cấp 12](#_Toc195304597)

[2.3.5 Bảng KhachHang 13](#_Toc195304598)

[2.3.5.1 Thêm khách hàng mới 13](#_Toc195304599)

[2.3.5.2 Cập nhập thông tin khách hàng 14](#_Toc195304600)

[2.3.6 Bảng NhanVien 15](#_Toc195304601)

[2.3.6.1 Thêm nhân viên mới 15](#_Toc195304602)

[2.3.6.2 Cập nhập thông tin nhân viên 16](#_Toc195304603)

[2.3.7 Bảng NhapHang 17](#_Toc195304604)

[2.3.7.1 Thêm phiếu nhập hàng 17](#_Toc195304605)

[2.3.8 Bảng CT\_NhapHang 18](#_Toc195304606)

[2.3.8.1 Thêm dữ liệu vào chi tiết nhập hàng 18](#_Toc195304607)

[2.3.9 Bảng ThanhToan 20](#_Toc195304608)

[2.3.9.1 Thêm mới dữ liệu thanh toán 20](#_Toc195304609)

[2.3.10 Bảng BanHang 21](#_Toc195304610)

[2.3.10.1 Thêm phiếu hóa đơn bán hàng 21](#_Toc195304611)

[2.3.11 Bảng CT\_BanHang 22](#_Toc195304612)

[2.3.11.1 Thêm chi tiết bán hàng 22](#_Toc195304613)

[2.4 Dump dữ liệu cho Database 24](#_Toc195304614)

[2.4.1 Bảng DanhMuc 24](#_Toc195304615)

[2.4.2 Bảng HangHoa 25](#_Toc195304616)

[2.4.3 Bảng NhaCungCap 27](#_Toc195304617)

[2.4.4 Bảng KhachHang 29](#_Toc195304618)

[2.4.5 Bảng NhanVien 32](#_Toc195304619)

[2.4.6 Bảng NhapHang, CT\_NhapHang, GiaHangHoa, ThanhToan 33](#_Toc195304620)

[2.4.7 Bảng BanHang, CT\_BanHang 37](#_Toc195304621)

# R1. XÂY DỰNG CƠ SỞ DỮ LIỆU

# R2. TẠO MODULE VÀ DUMP DỮ LIỆU

## Phân chia công việc



## Chỉnh sửa Database

\*Chú thích: Vì các loại mã (khóa chính) của các bạn có độ dài không phù hợp với dữ liệu lớn (1 triệu dòng) nên nhóm em xin được chỉnh sửa các kiểu dữ liệu của mã trong các bảng thành VARCHAR(11). Và bọn em có sửa một chút ở phần điều kiện Check số điện thoại của bảng KhachHang ạ.

CREATE TABLE DanhMuc

(

Ma\_DM VARCHAR(11) CONSTRAINT PK\_DanhMuc PRIMARY KEY,

Ten\_DM NVARCHAR(30) CONSTRAINT UQ\_Ten\_DM UNIQUE

)

CREATE TABLE HangHoa

(

Ma\_HH VARCHAR(11) CONSTRAINT PK\_HangHoa PRIMARY KEY,

Ten\_HH NVARCHAR(50) CONSTRAINT UQ\_Ten\_HH UNIQUE,

-- Bọn em có bổ sung thêm đơn vị tính cho hàng hóa để đa dạng hơn

Don\_Vi\_Tinh NVARCHAR(20) CONSTRAINT CK\_Don\_Vi\_Tinh CHECK(Don\_Vi\_Tinh IN (N'Gói', N'Chai', N'Lon', N'Hộp', N'Bịch', N'Cái', N'Kg', N'Lít', N'Tuýp', N'Lốc', N'Thùng')), SL\_Ton\_Kho INT CONSTRAINT CK\_SL\_Ton\_Kho CHECK(SL\_Ton\_Kho >= 0),

Ma\_DM VARCHAR(11) CONSTRAINT FK\_HangHoa\_DanhMuc FOREIGN KEY REFERENCES DanhMuc(Ma\_DM)

)

CREATE TABLE GiaHangHoa

(

Ma\_GHH VARCHAR(11) CONSTRAINT PK\_GiaHangHoa PRIMARY KEY,

Gia\_Nhap MONEY CONSTRAINT CK\_Gia\_Nhap CHECK(Gia\_Nhap >= 0),

Gia\_Ban MONEY CONSTRAINT CK\_Gia\_Ban CHECK(Gia\_Ban >= 0),

Ngay\_Bat\_Dau DATE CONSTRAINT CK\_Ngay\_Bat\_Dau CHECK(Ngay\_Bat\_Dau <= GETDATE()),

Ngay\_Ket\_Thuc DATE,

Ma\_HH VARCHAR(11) CONSTRAINT FK\_GiaHangHoa\_HangHoa FOREIGN KEY REFERENCES HangHoa(Ma\_HH)

)

CREATE TABLE NhaCungCap

(

Ma\_NCC VARCHAR(11) CONSTRAINT PK\_NhaCungCap PRIMARY KEY,

Ten\_NCC NVARCHAR(50) CONSTRAINT UQ\_Ten\_NCC UNIQUE,

Dia\_Chi\_NCC NVARCHAR(50),

SDT\_NCC VARCHAR(11) CONSTRAINT CK\_SDT\_NCC CHECK(LEN(SDT\_NCC) BETWEEN 10 AND 11 AND ISNUMERIC(SDT\_NCC) = 1),

Tong\_Tien\_No MONEY CONSTRAINT CK\_Tong\_Tien\_No CHECK(Tong\_Tien\_No >= 0)

)

CREATE TABLE KhachHang

(

Ma\_KH VARCHAR(11) CONSTRAINT PK\_KhachHang PRIMARY KEY,

Ten\_KH NVARCHAR(50) CONSTRAINT UQ\_Ten\_KH UNIQUE,

Dia\_Chi\_KH NVARCHAR(50),

-- Bọn em chuyển sang LEN(SDT\_KH) = 10 thay vì lúc trước là LEN(SDT\_KH) BETWEEN 10 AND 11

SDT\_KH VARCHAR(10) CONSTRAINT CK\_SDT\_KH CHECK(LEN(SDT\_KH) = 10 AND ISNUMERIC(SDT\_KH) = 1)

)

CREATE TABLE NhanVien

(

Ma\_NV VARCHAR(11) CONSTRAINT PK\_NhanVien PRIMARY KEY,

Ten\_NV NVARCHAR(50) CONSTRAINT UQ\_Ten\_NV UNIQUE,

SDT\_NV VARCHAR(10) CONSTRAINT CK\_SDT\_NV CHECK(LEN(SDT\_NV) = 10 AND ISNUMERIC(SDT\_NV) = 1)

)

--NHAP HANG

CREATE TABLE NhapHang

(

SoHD\_NhapHang VARCHAR(11) CONSTRAINT PK\_NhapHang PRIMARY KEY,

Ngay\_Nhap DATE CONSTRAINT CK\_Ngay\_Nhap CHECK(Ngay\_Nhap <= GETDATE()),

Tong\_Thanh\_Tien MONEY CONSTRAINT CK\_Tong\_Thanh\_Tien\_Nhap CHECK(Tong\_Thanh\_Tien >= 0),

Tien\_Da\_Tra MONEY CONSTRAINT CK\_Tien\_Da\_Tra CHECK(Tien\_Da\_Tra >= 0),

Ma\_NV VARCHAR(11) CONSTRAINT FK\_NhapHang\_NhanVien FOREIGN KEY REFERENCES NhanVien(Ma\_NV),

Ma\_NCC VARCHAR(11) CONSTRAINT FK\_NhapHang\_NhaCungCap FOREIGN KEY REFERENCES NhaCungCap(Ma\_NCC)

)

CREATE TABLE CT\_NhapHang

(

SoHD\_NhapHang VARCHAR(11) CONSTRAINT FK\_CT\_NhapHang\_NhapHang FOREIGN KEY REFERENCES NhapHang(SoHD\_NhapHang),

Ma\_HH VARCHAR(11) CONSTRAINT FK\_CT\_NhapHang\_HangHoa FOREIGN KEY REFERENCES HangHoa(Ma\_HH),

CONSTRAINT PK\_CT\_NhapHang PRIMARY KEY(SoHD\_NhapHang, Ma\_HH),

So\_Luong INT CONSTRAINT CK\_So\_Luong\_Nhap CHECK(So\_Luong >= 1),

-- Bọn em nhầm lẫn hạn sử dụng nên đã chỉnh sửa lại ở đây ạ

Han\_Su\_Dung DATE CONSTRAINT CK\_Han\_Su\_Dung CHECK(Han\_Su\_Dung > GETDATE()),

Thanh\_Tien MONEY CONSTRAINT CK\_Thanh\_Tien\_Nhap CHECK(Thanh\_Tien >= 0)

)

CREATE TABLE ThanhToan

(

Ma\_TT VARCHAR(11) CONSTRAINT PK\_ThanhToan PRIMARY KEY,

Ngay\_Thanh\_Toan DATE CONSTRAINT CK\_Ngay\_Thanh\_Toan CHECK(Ngay\_Thanh\_Toan <= GETDATE()),

Tien\_Giao\_Dich MONEY CONSTRAINT CK\_Tien\_Giao\_Dich CHECK(Tien\_Giao\_Dich >= 0),

Tong\_No\_Con\_Lai MONEY CONSTRAINT CK\_Tong\_No\_Con\_Lai CHECK(Tong\_No\_Con\_Lai >= 0),

SoHD\_NhapHang VARCHAR(11) CONSTRAINT FK\_ThanhToan\_NhapHang FOREIGN KEY REFERENCES NhapHang(SoHD\_NhapHang)

)

--BAN HANG

CREATE TABLE BanHang

(

SoHD\_BanHang VARCHAR(11) CONSTRAINT PK\_BanHang PRIMARY KEY,

Ngay\_Ban DATE CONSTRAINT CK\_Ngay\_Ban CHECK(Ngay\_Ban <= GETDATE()),

Tong\_Thanh\_Tien MONEY CONSTRAINT CK\_Tong\_Thanh\_Tien\_Ban CHECK(Tong\_Thanh\_Tien >= 0),

Ma\_NV VARCHAR(11) CONSTRAINT FK\_BanHang\_NhanVien FOREIGN KEY REFERENCES NhanVien(Ma\_NV),

Ma\_KH VARCHAR(11) CONSTRAINT FK\_BanHang\_KhachHang FOREIGN KEY REFERENCES KhachHang(Ma\_KH)

)

CREATE TABLE CT\_BanHang

(

SoHD\_BanHang VARCHAR(11) CONSTRAINT FK\_CT\_BanHang\_BanHang FOREIGN KEY REFERENCES BanHang(SoHD\_BanHang),

Ma\_GHH VARCHAR(11) CONSTRAINT FK\_CT\_BanHang\_GiaHangHoa FOREIGN KEY REFERENCES GiaHangHoa(Ma\_GHH),

CONSTRAINT PK\_CT\_BanHang PRIMARY KEY(SoHD\_BanHang, Ma\_GHH),

So\_Luong INT CONSTRAINT CK\_So\_Luong\_Ban CHECK(So\_Luong >= 1),

Thanh\_Tien MONEY CONSTRAINT CK\_Thanh\_Tien\_Ban CHECK(Thanh\_Tien >= 0)

)

## Viết Module

### Bảng DanhMuc

#### Thêm danh mục mới

/\*

+ Input: @Ten\_DM

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Insert\_DanhMuc (@Ten\_DM NVARCHAR(30), @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem tên danh mục đã có trong dữ liệu chưa. Nếu đã có thì thông báo lỗi và dừng chạy thủ tục

IF EXISTS (SELECT 7 FROM DanhMuc WHERE Ten\_DM = @Ten\_DM)

BEGIN

PRINT N'---- Thông báo lỗi: Đã có tên danh mục này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo mã danh mục mới

DECLARE @Ma\_DM VARCHAR(11) = (SELECT 'DM' + FORMAT(ISNULL(RIGHT(MAX(Ma\_DM), 7), 0) + 1, '0000000') FROM DanhMuc)

-- Thêm dữ liệu vào bảng DanhMuc

INSERT INTO DanhMuc (Ma\_DM, Ten\_DM)

VALUES (@Ma\_DM, @Ten\_DM)

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã thêm thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Thêm không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

#### Cập nhập thông tin danh mục

/\*

+ Input: @Ma\_DM, @Ten\_DM\_Moi

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Update\_DanhMuc (@Ma\_DM VARCHAR(11), @Ten\_DM\_Moi NVARCHAR(30), @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem mã danh mục đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM DanhMuc WHERE Ma\_DM = @Ma\_DM)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã danh mục này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem tên danh mục đã có trong dữ liệu chưa. Nếu đã có thì thông báo lỗi và dừng chạy thủ tục

IF EXISTS (SELECT 7 FROM DanhMuc WHERE Ten\_DM = @Ten\_DM\_Moi)

BEGIN

PRINT N'---- Thông báo lỗi: Đã có tên danh mục này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

--Cập nhật lại thông tin ở bảng DanhMuc

UPDATE DanhMuc

SET Ten\_DM = @Ten\_DM\_Moi

WHERE Ma\_DM = @Ma\_DM

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã cập nhập thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Cập nhập không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

### Bảng HangHoa

#### Thêm hàng hóa mới

/\*

+ Input: @Ten\_HH, @Don\_Vi\_Tinh, @Ma\_DM

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Insert\_HangHoa (@Ten\_HH NVARCHAR(50), @Don\_Vi\_Tinh NVARCHAR(20), @Ma\_DM VARCHAR(11), @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem tên hàng hóa đã có trong dữ liệu chưa. Nếu đã có thì thông báo lỗi và dừng chạy thủ tục

IF EXISTS (SELECT 7 FROM HangHoa WHERE Ten\_HH = @Ten\_HH)

BEGIN

PRINT N'---- Thông báo lỗi: Đã có tên hàng hóa này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem đơn vị tính có nằm trong những mục hợp lệ không. Nếu không nằm trong đó thì thông báo lỗi và dừng chạy thủ tục

IF @Don\_Vi\_Tinh NOT IN (N'Gói', N'Chai', N'Lon', N'Hộp', N'Bịch', N'Cái', N'Kg', N'Lít', N'Tuýp', N'Lốc', N'Thùng', N'Thanh', N'Vỉ', N'Cuộn', N'Nải', N'Túi')

BEGIN

PRINT N'---- Thông báo lỗi: Đơn vị tính này không hợp lệ! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem mã danh mục đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM DanhMuc WHERE Ma\_DM = @Ma\_DM)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã danh mục này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo mã hàng hóa mới

DECLARE @Ma\_HH VARCHAR(11) = (SELECT 'HH' + FORMAT(ISNULL(RIGHT(MAX(Ma\_HH), 7), 0) + 1, '0000000') FROM HangHoa)

-- Thêm dữ liệu vào bảng HangHoa

INSERT INTO HangHoa (Ma\_HH, Ten\_HH, Don\_Vi\_Tinh, Ma\_DM)

VALUES (@Ma\_HH, @Ten\_HH, @Don\_Vi\_Tinh, @Ma\_DM)

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã thêm thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Thêm không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

#### Cập nhập thông tin hàng hóa

/\*

+ Input: @Ma\_HH, @Ten\_HH\_Moi, @Don\_Vi\_Tinh\_Moi, @Ma\_DM\_Moi

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Update\_HangHoa (@Ma\_HH VARCHAR(11), @Ten\_HH\_Moi NVARCHAR(50), @Don\_Vi\_Tinh\_Moi NVARCHAR(20), @Ma\_DM\_Moi VARCHAR(11), @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem mã hàng hóa đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM HangHoa WHERE Ma\_HH = @Ma\_HH)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã hàng hóa này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem tên danh mục đã có trong dữ liệu chưa. Nếu đã có thì thông báo lỗi và dừng chạy thủ tục

IF EXISTS (SELECT 7 FROM HangHoa WHERE Ten\_HH = @Ten\_HH\_Moi)

BEGIN

PRINT N'---- Thông báo lỗi: Đã có tên danh mục này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem mã danh mục đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM DanhMuc WHERE Ma\_DM = @Ma\_DM\_Moi)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã danh mục này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Cập nhật lại thông tin ở bảng HangHoa

UPDATE HangHoa

SET Ten\_HH = @Ten\_HH\_Moi, Don\_Vi\_Tinh = @Don\_Vi\_Tinh\_Moi, Ma\_DM = @Ma\_DM\_Moi

WHERE Ma\_HH = @Ma\_HH

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã cập nhập thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Cập nhập không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

### Bảng GiaHangHoa

#### Thêm giá hàng hóa mới

/\*

+ Input: @Ma\_HH, @Gia\_Nhap, @Ngay\_Bat\_Dau, @Ngay\_Ket\_Thuc

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Insert\_GiaHangHoa (@Ma\_HH VARCHAR(11), @Gia\_Nhap MONEY, @Ngay\_Bat\_Dau DATE, @Ngay\_Ket\_Thuc DATE, @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem giá nhập hàng có hợp lệ chưa (hợp lệ: > 0). Nếu không hợp lệ thì thông báo lỗi và dừng chạy thủ tục

IF @Gia\_Nhap <= 0

BEGIN

PRINT N'---- Thông báo lỗi: Giá nhập không hợp lệ! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tính giá bán bằng tổng của giá nhập và 20% của giá nhập

DECLARE @Gia\_Ban MONEY = @Gia\_Nhap + (@Gia\_Nhap \* 0.2)

-- Tạo mã giá hàng hoá mới

DECLARE @Ma\_GHH VARCHAR(11) = (SELECT 'GHH' + FORMAT(ISNULL(RIGHT(MAX(Ma\_GHH), 7), 0) + 1, '0000000') FROM GiaHangHoa)

-- Thêm dữ liệu vào bảng GiaHangHoa

INSERT INTO GiaHangHoa (Ma\_GHH, Gia\_Nhap, Gia\_Ban, Ngay\_Bat\_Dau, Ngay\_Ket\_Thuc, Ma\_HH)

VALUES(@Ma\_GHH, @Gia\_Nhap, @Gia\_Ban, @Ngay\_Bat\_Dau, @Ngay\_Ket\_Thuc, @Ma\_HH)

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã thêm thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Thêm không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

### Bảng NhaCungCap

#### Thêm nhà cung cấp mới

/\*

+ Input: @Ten\_NCC, @Dia\_Chi\_NCC, @SDT\_NCC

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Insert\_NhaCungCap (@Ten\_NCC NVARCHAR(50), @Dia\_Chi\_NCC NVARCHAR(100), @SDT\_NCC VARCHAR(11), @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem tên và địa chỉ nhà cung cấp đã có trong dữ liệu chưa. Nếu đã có thì thông báo lỗi và dừng chạy thủ tục

IF EXISTS (SELECT 7 FROM NhaCungCap WHERE Ten\_NCC = @Ten\_NCC AND Dia\_Chi\_NCC = @Dia\_Chi\_NCC)

BEGIN

PRINT N'---- Thông báo lỗi: Đã có tên và địa chỉ nhà cung cấp này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem số điện thoại nhà cung cấp có hợp lệ không (hợp lệ: toàn là chữ số và độ dài là 10 hoặc 11) hoặc kiểm tra số đó đã có trong dữ liệu chưa. Nếu sai hoặc đã có thì thông báo lỗi và dừng chạy thủ tục

IF (ISNUMERIC(@SDT\_NCC) = 0 OR LEN(@SDT\_NCC) NOT BETWEEN 10 AND 11) OR EXISTS (SELECT 7 FROM NhaCungCap WHERE SDT\_NCC = @SDT\_NCC)

BEGIN

PRINT N'---- Thông báo lỗi: Số điện thoại này không hợp lệ hoặc đã có! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo mã nhà cung cấp mới

DECLARE @Ma\_NCC VARCHAR(11) = (SELECT 'NCC' + FORMAT(ISNULL(RIGHT(MAX(Ma\_NCC), 7), 0) + 1, '0000000') FROM NhaCungCap)

-- Thêm dữ liệu vào bảng NhaCungCap

INSERT INTO NhaCungCap (Ma\_NCC, Ten\_NCC, Dia\_Chi\_NCC, SDT\_NCC)

VALUES (@Ma\_NCC, @Ten\_NCC, @Dia\_Chi\_NCC, @SDT\_NCC)

IF @@ROWCOUNT>0

BEGIN

PRINT N'---- Thông báo kết quả: Đã thêm thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Thêm không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

#### Cập nhập thông tin nhà cung cấp

/\*

+ Input: @Ma\_NCC, @Ten\_NCC\_Moi, @Dia\_Chi\_NCC\_Moi, @SDT\_NCC\_Moi

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Update\_NhaCungCap (@Ma\_NCC VARCHAR(11), @Ten\_NCC\_Moi NVARCHAR(50), @Dia\_Chi\_NCC\_Moi NVARCHAR(50), @SDT\_NCC\_Moi VARCHAR(11), @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem mã nhà cung cấp đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM NhaCungCap WHERE Ma\_NCC = @Ma\_NCC)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã nhà cung cấp này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem tên và địa chỉ nhà cung cấp đã có trong dữ liệu chưa. Nếu có rồi thì thông báo lỗi và dừng chạy thủ tục

IF EXISTS (SELECT 7 FROM NhaCungCap WHERE Ten\_NCC = @Ten\_NCC\_Moi AND Dia\_Chi\_NCC = @Dia\_Chi\_NCC\_Moi)

BEGIN

PRINT N'---- Thông báo lỗi: Đã có tên và địa chỉ nhà cung cấp này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem số điện thoại nhà cung cấp có hợp lệ không (hợp lệ: toàn là chữ số và độ dài là 10 hoặc 11) hoặc kiểm tra số đó đã có trong dữ liệu chưa. Nếu sai hoặc đã có thì thông báo lỗi và dừng chạy thủ tục

IF (ISNUMERIC(@SDT\_NCC\_Moi) = 0 OR LEN(@SDT\_NCC\_Moi) NOT BETWEEN 10 AND 11) OR EXISTS (SELECT 7 FROM NhaCungCap WHERE SDT\_NCC = @SDT\_NCC\_Moi)

BEGIN

PRINT N'---- Thông báo lỗi: Số điện thoại không hợp lệ hoặc đã tồn tại! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Cập nhập lại thông tin ở bảng NhaCungCap

UPDATE NhaCungCap

SET Ten\_NCC = @Ten\_NCC\_Moi, Dia\_Chi\_NCC = @Dia\_Chi\_NCC\_Moi, SDT\_NCC = @SDT\_NCC\_Moi

WHERE Ma\_NCC = @Ma\_NCC

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã cập nhập thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Cập nhập không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

### Bảng KhachHang

#### Thêm khách hàng mới

/\*

+ Input: @Ten\_KH, @Dia\_Chi\_KH, @SDT\_KH

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Insert\_KhachHang (@Ten\_KH NVARCHAR(50), @Dia\_Chi\_KH NVARCHAR(50), @SDT\_KH VARCHAR(10), @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem tên và địa chỉ khách hàng đã có trong dữ liệu chưa. Nếu đã có thì thông báo lỗi và dừng chạy thủ tục

IF EXISTS (SELECT 7 FROM KhachHang WHERE Ten\_KH = @Ten\_KH AND Dia\_Chi\_KH = @Dia\_Chi\_KH)

BEGIN

PRINT N'---- Thông báo lỗi: Đã có tên và địa chỉ khách hàng này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem số điện thoại khách hàng có hợp lệ không (hợp lệ: toàn là chữ số và độ dài là 10) hoặc kiểm tra số đó đã có trong dữ liệu chưa. Nếu sai hoặc đã có thì thông báo lỗi và dừng chạy thủ tục

IF (ISNUMERIC(@SDT\_KH) = 0 OR LEN(@SDT\_KH) != 10) OR EXISTS (SELECT 7 FROM KhachHang WHERE SDT\_KH = @SDT\_KH)

BEGIN

PRINT N'---- Thông báo lỗi: Số điện thoại không hợp lệ hoặc đã tồn tại! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo mã khách hàng mới

DECLARE @Ma\_KH VARCHAR(11) = (SELECT 'KH' + FORMAT(ISNULL(RIGHT(MAX(Ma\_KH), 7), 0) + 1, '0000000') FROM KhachHang)

-- Thêm dữ liệu vào bảng KhachHang

INSERT INTO KhachHang (Ma\_KH, Ten\_KH, Dia\_Chi\_KH, SDT\_KH)

VALUES (@Ma\_KH, @Ten\_KH, @Dia\_Chi\_KH, @SDT\_KH)

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã thêm thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Thêm không thành công'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

#### Cập nhập thông tin khách hàng

CREATE OR ALTER PROCEDURE sp\_Update\_KhachHang (@Ma\_KH VARCHAR(11), @Ten\_KH\_Moi NVARCHAR(50), @Dia\_Chi\_KH\_Moi NVARCHAR(50), @SDT\_KH\_Moi VARCHAR(10), @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem mã khách hàng đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM KhachHang WHERE Ma\_KH = @Ma\_KH)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã khách hàng này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem tên và địa chỉ khách hàng đã có trong dữ liệu chưa. Nếu có rồi thì thông báo lỗi và dừng chạy thủ tục

IF EXISTS (SELECT 7 FROM KhachHang WHERE Ten\_KH = @Ten\_KH\_Moi AND Dia\_Chi\_KH = @Dia\_Chi\_KH\_Moi)

BEGIN

PRINT N'---- Thông báo lỗi: Đã có tên và địa chỉ khách hàng này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem số điện thoại khách hàng có hợp lệ không (hợp lệ: toàn là chữ số và độ dài là 10) hoặc kiểm tra số đó đã có trong dữ liệu chưa. Nếu sai hoặc đã có thì thông báo lỗi và dừng chạy thủ tục

IF (ISNUMERIC(@SDT\_KH\_Moi) = 0 OR LEN(@SDT\_KH\_Moi) != 10) OR EXISTS (SELECT 7 FROM KhachHang WHERE SDT\_KH = @SDT\_KH\_Moi)

BEGIN

PRINT N'---- Thông báo lỗi: Số điện thoại không hợp lệ hoặc đã tồn tại! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Cập nhập lại thông tin ở bảng KhachHang

UPDATE KhachHang

SET Ten\_KH = @Ten\_KH\_Moi, Dia\_Chi\_KH = @Dia\_Chi\_KH\_Moi, SDT\_KH = @SDT\_KH\_Moi

WHERE Ma\_KH = @Ma\_KH

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã cập nhập thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Cập nhập không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

### Bảng NhanVien

#### Thêm nhân viên mới

/\*

+ Input: @Ten\_NV, @SDT\_NV

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Insert\_NhanVien (@Ten\_NV NVARCHAR(50), @SDT\_NV VARCHAR(10), @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem số điện thoại có hợp lệ không (hợp lệ: toàn là chữ số và độ dài là 10) hoặc kiểm tra số đó đã có trong dữ liệu chưa. Nếu sai hoặc đã có thì thông báo lỗi và dừng chạy thủ tục

IF (ISNUMERIC(@SDT\_NV) = 0 OR LEN(@SDT\_NV) != 10) OR EXISTS (SELECT 7 FROM NhanVien WHERE SDT\_NV = @SDT\_NV)

BEGIN

PRINT N'---- Thông báo lỗi: Số điện thoại không hợp lệ hoặc đã tồn tại! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo mã nhân viên mới

DECLARE @Ma\_NV VARCHAR(11) = (SELECT 'NV' + FORMAT(ISNULL(RIGHT(MAX(Ma\_NV), 7), 0) + 1, '0000000') FROM NhanVien)

-- Thêm nhân dữ liệu vào bảng NhanVien

INSERT INTO NhanVien

VALUES(@Ma\_NV, @Ten\_NV, @SDT\_NV)

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã thêm thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Thêm không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

#### Cập nhập thông tin nhân viên

/\*

+ Input: @Ma\_NV, @Ten\_NV\_Moi, @SDT\_NV\_Moi

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Update\_NhanVien (@Ma\_NV VARCHAR(11), @Ten\_NV\_Moi NVARCHAR(50), @SDT\_NV\_Moi VARCHAR(10), @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem số điện thoại có hợp lệ không (hợp lệ: toàn là chữ số và độ dài là 10) hoặc kiểm tra số đó đã có trong dữ liệu chưa. Nếu sai hoặc đã có thì thông báo lỗi và dừng chạy thủ tục

IF (ISNUMERIC(@SDT\_NV\_Moi) = 0 OR LEN(@SDT\_NV\_Moi) != 10) OR EXISTS (SELECT 7 FROM NhanVien WHERE SDT\_NV = @SDT\_NV\_Moi)

BEGIN

PRINT N'---- Thông báo lỗi: Số điện thoại không hợp lệ hoặc đã tồn tại! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Cập nhập lại thông tin ở bảng NhanVien

UPDATE NhanVien

SET Ten\_NV = @Ten\_NV\_Moi, SDT\_NV = @SDT\_NV\_Moi

WHERE Ma\_NV = @Ma\_NV

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã cập nhập thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Cập nhập không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

### Bảng NhapHang

#### Thêm phiếu nhập hàng

CREATE OR ALTER PROCEDURE sp\_Insert\_NhapHang (@Ma\_NV VARCHAR(11), @Ma\_NCC VARCHAR(11), @Ngay\_Nhap DATE, @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem mã nhân viên đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM NhanVien WHERE Ma\_NV = @Ma\_NV)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã nhân viên này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem mã nhà cung cấp đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM NhaCungCap WHERE Ma\_NCC = @Ma\_NCC)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã nhà cung cấp này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem ngày nhập có hợp lệ không (hợp lệ: <= ngày hiện tại). Nếu sai thì thông báo lỗi và dừng chạy thủ tục

IF @Ngay\_Nhap > GETDATE()

BEGIN

PRINT N'---- Thông báo lỗi: Ngày nhập không hợp lệ! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo mã số hóa đơn nhập hàng mới

DECLARE @SoHD\_NhapHang VARCHAR(11) = (SELECT 'NH' + FORMAT(ISNULL(RIGHT(MAX(SoHD\_NhapHang), 7), 0) + 1, '0000000') FROM NhapHang)

-- Thêm dữ liệu vào bảng NhapHang

INSERT INTO NhapHang (SoHD\_NhapHang, Ngay\_Nhap, Ma\_NV, Ma\_NCC)

VALUES (@SoHD\_NhapHang, @Ngay\_Nhap, @Ma\_NV, @Ma\_NCC)

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã thêm thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Thêm không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

### Bảng CT\_NhapHang

#### Thêm dữ liệu vào chi tiết nhập hàng

/\*

+ Input: @SoHD\_NhapHang, @Ma\_HH, @So\_Luong, @Han\_Su\_Dung, @Thanh\_Tien

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Insert\_CT\_NhapHang (@SoHD\_NhapHang VARCHAR(11), @Ma\_HH VARCHAR(11), @So\_Luong INT, @Han\_Su\_Dung DATE, @Thanh\_Tien MONEY, @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem mã số hóa đơn nhập hàng đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM NhapHang WHERE SoHD\_NhapHang = @SoHD\_NhapHang)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã số hóa đơn nhập hàng này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem mã hàng hóa đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM HangHoa WHERE Ma\_HH = @Ma\_HH)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã hàng hóa này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem cặp khóa đã có trong dữ liệu chưa. Nếu có rồi thì thông báo lỗi và dừng chạy thủ tục

IF EXISTS (SELECT 7 FROM CT\_NhapHang WHERE SoHD\_NhapHang = @SoHD\_NhapHang AND Ma\_HH = @Ma\_HH)

BEGIN

PRINT N'---- Thông báo lỗi: Đã có cặp khóa này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem số lượng có hợp lệ không (hợp lệ: > 0). Nếu sai thì thông báo lỗi và dừng chạy thủ tục

IF @So\_Luong <= 0

BEGIN

PRINT N'---- Thông báo lỗi: Số lượng nhập hàng không hợp lệ! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem hạn sử dụng có hợp lệ không (hợp lệ: > ngày hiện tại). Nếu sai thì thông báo lỗi và dừng chạy thủ tục

IF @Han\_Su\_Dung <= GETDATE()

BEGIN

PRINT N'---- Thông báo lỗi: Hạn sử dụng không hợp lệ! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem thành tiền có hợp lệ không (hợp lệ: > 0). Nếu sai thì thông báo lỗi và dừng chạy thủ tục

IF @Thanh\_Tien <= 0

BEGIN

PRINT N'---- Thông báo lỗi: Thành tiền không hợp lệ! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Thêm dữ liệu vào bảng CT\_NhapHang

INSERT INTO CT\_NhapHang (SoHD\_NhapHang, Ma\_HH, So\_Luong, Han\_Su\_Dung, Thanh\_Tien)

VALUES (@SoHD\_NhapHang, @Ma\_HH, @So\_Luong, @Han\_Su\_Dung, @Thanh\_Tien)

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã thêm thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Thêm không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

### Bảng ThanhToan

#### Thêm mới dữ liệu thanh toán

/\*

+ Input: @SoHD\_NhapHang, @Ngay\_Thanh\_Toan, @Tien\_Giao\_Dich

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Insert\_ThanhToan (@SoHD\_NhapHang VARCHAR(11), @Ngay\_Thanh\_Toan DATE, @Tien\_Giao\_Dich MONEY, @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem mã số hóa đơn nhập hàng đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM NhapHang WHERE SoHD\_NhapHang = @SoHD\_NhapHang)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã số hóa đơn nhập hàng này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem ngày thanh toán có hợp lệ không (hợp lệ: <= ngày hiện tại). Nếu sai thì thông báo lỗi và dừng chạy thủ tục

IF @Ngay\_Thanh\_Toan > GETDATE()

BEGIN

PRINT N'---- Thông báo lỗi: Ngày thanh toán không hợp lệ! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem tiền giao dịch có hợp lệ không (hợp lệ: > 0). Nếu sai thì thông báo lỗi và dừng chạy thủ tục

IF @Tien\_Giao\_Dich <= 0

BEGIN

PRINT N'---- Thông báo lỗi: Tiền giao dịch không hợp lệ! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tính tổng nợ còn lại

DECLARE @Tong\_Thanh\_Tien MONEY = (SELECT Tong\_Thanh\_Tien FROM NhapHang WHERE SoHD\_NhapHang = @SoHD\_NhapHang)

DECLARE @Tong\_No\_Con\_Lai MONEY = @Tong\_Thanh\_Tien - @Tien\_Giao\_Dich

-- Tạo mã thanh toán mới

DECLARE @Ma\_TT VARCHAR(11) = (SELECT 'TT' + FORMAT(ISNULL(RIGHT(MAX(Ma\_TT), 7), 0) + 1, '0000000') FROM ThanhToan)

-- Thêm dữ liệu vào bảng ThanhToan

INSERT INTO ThanhToan (Ma\_TT, Ngay\_Thanh\_Toan, Tien\_Giao\_Dich, Tong\_No\_Con\_Lai, SoHD\_NhapHang)

VALUES (@Ma\_TT, @Ngay\_Thanh\_Toan, @Tien\_Giao\_Dich, @Tong\_No\_Con\_Lai, @SoHD\_NhapHang)

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã thêm thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Thêm không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

### Bảng BanHang

#### Thêm phiếu hóa đơn bán hàng

/\*

+ Input: @Ma\_NV, @Ma\_KH, @Ngay\_Ban

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Insert\_BanHang (@Ma\_NV VARCHAR(11), @Ma\_KH VARCHAR(11), @Ngay\_Ban DATE, @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem mã nhân viên đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM NhanVien WHERE Ma\_NV = @Ma\_NV)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã nhân viên này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem mã khách hàng đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM KhachHang WHERE Ma\_KH = @Ma\_KH)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã khách hàng này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem ngày nhập có hợp lệ không (hợp lệ: <= ngày hiện tại). Nếu sai thì thông báo lỗi và dừng chạy thủ tục

IF @Ngay\_Ban > GETDATE()

BEGIN

PRINT N'---- Thông báo lỗi: Ngày bán không hợp lệ! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo mã số hóa đơn nhập hàng mới

DECLARE @SoHD\_BanHang VARCHAR(11) = (SELECT 'BH' + FORMAT(ISNULL(RIGHT(MAX(SoHD\_BanHang), 7), 0) + 1, '0000000') FROM BanHang)

-- Thêm dữ liệu vào bảng BanHang

INSERT INTO BanHang (SoHD\_BanHang, Ngay\_Ban, Ma\_NV, Ma\_KH)

VALUES (@SoHD\_BanHang, @Ngay\_Ban, @Ma\_NV, @Ma\_KH)

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã thêm thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Thêm không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

### Bảng CT\_BanHang

#### Thêm chi tiết bán hàng

/\*

+ Input: @SoHD\_BanHang, @Ma\_GHH, @So\_Luong

+ Output: @ketQuaTraVe (1: Thành công, 0: Không thành công)

+ Process:

\*/

CREATE OR ALTER PROCEDURE sp\_Insert\_CT\_BanHang (@SoHD\_BanHang VARCHAR(11), @Ma\_GHH VARCHAR(11), @So\_Luong INT, @ketQuaTraVe BIT OUT)

AS

BEGIN

-- Cài đặt chế độ cô lập giao tác và tương tranh

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

-- Tạo điều kiện kiểm tra xem mã số hóa đơn bán hàng đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM BanHang WHERE SoHD\_BanHang = @SoHD\_BanHang)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã số hóa đơn bán hàng này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem mã giá hàng hóa đã có trong dữ liệu chưa. Nếu chưa có thì thông báo lỗi và dừng chạy thủ tục

IF NOT EXISTS (SELECT 7 FROM GiaHangHoa WHERE Ma\_GHH = @Ma\_GHH)

BEGIN

PRINT N'---- Thông báo lỗi: Chưa có mã giá hàng hóa này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tạo điều kiện kiểm tra xem cặp khóa đã có trong dữ liệu chưa. Nếu có rồi thì thông báo lỗi và dừng chạy thủ tục

IF EXISTS (SELECT 7 FROM CT\_BanHang WHERE SoHD\_BanHang = @SoHD\_BanHang AND Ma\_GHH = @Ma\_GHH)

BEGIN

PRINT N'---- Thông báo lỗi: Đã có cặp khóa này! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Lấy mã hàng hóa và giá bán

DECLARE @Ma\_HH VARCHAR(11), @Gia\_Ban MONEY

SELECT @Gia\_Ban = Gia\_Ban, @Ma\_HH = Ma\_HH FROM GiaHangHoa WHERE Ma\_GHH = @Ma\_GHH

-- Lấy số lượng tồn kho

DECLARE @SL\_Ton\_Kho INT = (SELECT SL\_Ton\_Kho FROM HangHoa WHERE Ma\_HH = @Ma\_HH)

-- Tạo điều kiện kiểm tra xem số lượng tồn kho có đủ không (đủ: <= số lượng tồn kho). Nếu không đủ thì thông báo lỗi và dừng chạy thủ tục

IF @So\_Luong > @SL\_Ton\_Kho

BEGIN

PRINT N'---- Thông báo lỗi: Số lượng tồn kho không đủ để bán! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

RETURN

END

-- Tính thành tiền

DECLARE @Thanh\_Tien MONEY = @So\_Luong \* @Gia\_Ban

-- Thêm dữ liệu vào bảng CT\_BanHang

INSERT INTO CT\_BanHang (SoHD\_BanHang, Ma\_GHH, So\_Luong, Thanh\_Tien)

VALUES (@SoHD\_BanHang, @Ma\_GHH, @So\_Luong, @Thanh\_Tien)

IF @@ROWCOUNT > 0

BEGIN

PRINT N'---- Thông báo kết quả: Đã thêm thành công! ----'

SET @ketQuaTraVe = 1

COMMIT TRANSACTION

END

ELSE

BEGIN

PRINT N'---- Thông báo kết quả: Thêm không thành công! ----'

SET @ketQuaTraVe = 0

ROLLBACK TRANSACTION

END

END

## Dump dữ liệu cho Database

### Bảng DanhMuc

-- DanhMuc --> Master

/\*Lý do:

+ Vì bảng lưu thông tin phân loại hàng hóa theo từng nhóm.

+ Dữ liệu trong bảng này không thay đổi thường xuyên vì dùng để tra cứu.

+ Các bảng khác như HangHoa cần tham chiếu đến DanhMuc thông qua khóa ngoại đảm bảo hàng hóa thuộc danh mục hợp lệ.

=> Kết luận: Bảng DanhMuc được chọn làm bảng master vì chứa dữ liệu ít thay đổi, là bảng dùng để tra cứu và phân nhóm hàng hóa.

\*/

DECLARE @noti\_DM BIT

EXEC sp\_Insert\_DanhMuc N'Bánh kẹo và đồ ngọt', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Cà phê và trà', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Chất tẩy rửa và vệ sinh nhà cửa', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Đặc sản vùng miền', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Đồ ăn nhanh', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Đồ đông lạnh', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Đồ dùng trẻ em', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Đồ hộp và thực phẩm chế biến sẵn', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Gia vị và nước chấm', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Giấy và bao bì', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Hạt dinh dưỡng và trái cây sấy', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Mì gói và cháo ăn liền', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Nước giải khát', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Nước uống đóng chai', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Rau củ và trái cây', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Sữa và sản phẩm từ sữa', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Thẻ cào điện thoại và dịch vụ tiện ích', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Thực phẩm ăn kiêng và hữu cơ', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Thực phẩm chay và vegan', @noti\_DM OUT

EXEC sp\_Insert\_DanhMuc N'Vệ sinh cá nhân', @noti\_DM OUT

-----------------------------------------------------

SELECT COUNT(\*) 'DanhMuc' FROM DanhMuc

SELECT TOP(5) \* FROM DanhMuc ORDER BY NEWID()

-----------------------------------------------------

A screenshot of a computer

AI-generated content may be incorrect.

### Bảng HangHoa

-- HangHoa --> Master

/\*Lý do:

+ Bảng chứa thông tin mô tả chi tiết của từng mặt hàng: mã hàng, tên hàng, đơn vị tính, số lượng tồn kho, mã danh mục.

+ Không phát sinh dữ liệu theo từng giao dịch.

+ Dữ liệu ít thay đổi, chỉ cập nhật khi thêm, sửa thông tin hàng hóa.

+ Là bảng nền cho các nghiệp vụ nhập hàng, bán hàng, kiểm tồn kho,..

=> Kết luận: Bảng HangHoa được chọn làm bảng master vì là bảng định danh hàng hóa, ít thay đổi, không phát sinh theo thời gian.

\*/

DECLARE @noti\_HH BIT

EXEC sp\_Insert\_HangHoa N'Bánh Oreo vị vani 133g', N'Gói', 'DM0000001', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Kẹo Alpenliebe mềm sữa 90g', N'Gói', 'DM0000001', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Bánh LU Pháp bơ sữa 200g', N'Hộp', 'DM0000001', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Kẹo Halls bạc hà 33g', N'Vỉ', 'DM0000001', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Bánh KitKat thanh nhỏ 10g', N'Thanh', 'DM0000001', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Cà phê G7 3in1 16g x 20 gói', N'Hộp', 'DM0000002', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Cà phê sữa nâu đá Highlands lon 235ml', N'Lon', 'DM0000002', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Trà Cozy hương nhài 25 túi lọc', N'Hộp', 'DM0000002', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Trà Ô Long Tea+ Plus 455ml', N'Chai', 'DM0000002', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Cà phê rang xay Trung Nguyên 500g', N'Gói', 'DM0000002', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Chả bò Đà Nẵng gói 500g', N'Gói', 'DM0000003', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Nem chua Thanh Hóa gói 10 cái', N'Gói', 'DM0000003', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Bánh cốm Nguyên Ninh 300g', N'Hộp', 'DM0000003', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Mắm nêm cá cơm Quảng Nam 250ml', N'Chai', 'DM0000003', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Khô cá chỉ vàng Phan Thiết 250g', N'Gói', 'DM0000003', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Xúc xích CP Đức gói 500g', N'Gói', 'DM0000004', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Pizza đông lạnh Đà Lạt Milk 200g', N'Cái', 'DM0000004', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Bánh mì sandwich Tài Ký 500g', N'Bịch', 'DM0000004', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Bánh bao nhân thịt Vissan 150g', N'Cái', 'DM0000004', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Chả giò rế CP 500g', N'Gói', 'DM0000005', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Tôm thẻ đông lạnh Minh Phú 300g', N'Gói', 'DM0000005', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Cá viên Cầu Tre 500g', N'Gói', 'DM0000005', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Gà cánh giữa đông lạnh CP 1kg', N'Kg', 'DM0000005', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Bánh xếp Ajinomoto Nhật 240g', N'Gói', 'DM0000005', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Tã quần Huggies size M 56 miếng', N'Bịch', 'DM0000006', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Khăn ướt Bobby không mùi 100 tờ', N'Gói', 'DM0000006', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Dầu gội Johnson’s Baby 200ml', N'Chai', 'DM0000006', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Bình sữa Pigeon 160ml cổ rộng', N'Cái', 'DM0000006', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Kem chống hăm Bepanthen 30g', N'Tuýp', 'DM0000006', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Thịt hộp Tulip Đan Mạch 340g', N'Hộp', 'DM0000007', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Cá ngừ ngâm dầu Hạ Long 170g', N'Lon', 'DM0000007', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Xúc xích tiệt trùng Đức Việt 200g', N'Gói', 'DM0000007', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Pate Gan Vissan hộp 170g', N'Hộp', 'DM0000007', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Cà ri bò hộp Vissan 150g', N'Hộp', 'DM0000007', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Nước mắm Nam Ngư 3 trong 1 500ml', N'Chai', 'DM0000008', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Dầu ăn Tường An 1L', N'Chai', 'DM0000008', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Nước tương Maggi 700ml', N'Chai', 'DM0000008', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Bột ngọt Ajinomoto 454g', N'Gói', 'DM0000008', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Hạt nêm Knorr nấm hương 400g', N'Gói', 'DM0000008', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Khăn giấy Bless You hộp 180 tờ', N'Hộp', 'DM0000009', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Giấy vệ sinh Pulppy 10 cuộn', N'Bịch', 'DM0000009', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Túi rác đen tự hủy 78x100cm', N'Cuộn', 'DM0000009', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Giấy nến nướng thực phẩm 5m', N'Cuộn', 'DM0000009', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Mì Hảo Hảo tôm chua cay 75g', N'Gói', 'DM0000010', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Mì Omachi sốt spaghetti 80g', N'Gói', 'DM0000010', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Cháo Vifon gà nấm 50g', N'Gói', 'DM0000010', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Mì 3 Miền bò hầm 75g', N'Gói', 'DM0000010', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Cháo tươi SG Food gà ác 270g', N'Túi', 'DM0000010', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Trà chanh C2 360ml', N'Chai', 'DM0000011', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Nước ngọt Coca-Cola 390ml', N'Lon', 'DM0000011', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Nước tăng lực Sting dâu 330ml', N'Lon', 'DM0000011', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Nước ép cam Twister 1L', N'Chai', 'DM0000011', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Nước suối Lavie 500ml', N'Chai', 'DM0000012', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Nước khoáng Vĩnh Hảo 1.5L', N'Chai', 'DM0000012', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Nước tinh khiết Aquafina 1.5L', N'Chai', 'DM0000012', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Nước ion Life 500ml', N'Chai', 'DM0000012', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Rau muống sạch Gói 300g', N'Gói', 'DM0000013', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Dưa leo baby 500g', N'Gói', 'DM0000013', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Chuối già hương 1 nải', N'Nải', 'DM0000013', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Cà chua bi Đà Lạt 500g', N'Gói', 'DM0000013', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Cam sành Vĩnh Long 1kg', N'Kg', 'DM0000013', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Sữa tươi Vinamilk 100% 1L', N'Hộp', 'DM0000014', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Sữa chua uống Probi dâu 80ml x4', N'Lốc', 'DM0000014', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Phô mai Con Bò Cười 120g', N'Hộp', 'DM0000014', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Sữa đặc Ngôi Sao Phương Nam 380g', N'Lon', 'DM0000014', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Ngũ cốc ăn kiêng Nestlé Fitnesse 375g', N'Hộp', 'DM0000015', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Sữa hạt óc chó Vegemil 190ml', N'Hộp', 'DM0000015', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Mì gạo lứt Hapi 500g', N'Gói', 'DM0000015', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Bánh gạo lứt Nissin 150g', N'Gói', 'DM0000015', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Kem đánh răng PS Trà xanh 180g', N'Tuýp', 'DM0000016', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Sữa tắm Lifebuoy 850g', N'Chai', 'DM0000016', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Dầu gội Clear sạch gàu 650g', N'Chai', 'DM0000016', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Khăn giấy ướt diệt khuẩn 100 tờ', N'Gói', 'DM0000016', @noti\_HH OUT

EXEC sp\_Insert\_HangHoa N'Bàn chải đánh răng Colgate Slim Soft', N'Cái', 'DM0000016', @noti\_HH OUT

-----------------------------------------------------

SELECT COUNT(\*) 'HangHoa' FROM HangHoa

SELECT TOP(5) \* FROM HangHoa ORDER BY NEWID()

-----------------------------------------------------

A screenshot of a menu

AI-generated content may be incorrect.

### Bảng NhaCungCap

-- NhaCungCap --> Master

/\*Lý do:

+ Chứa thông tin cơ bản và ổn định về các nhà cung cấp.

+ Dữ liệu trong bảng ít thay đổi, chủ yếu chỉ cập nhật khi có thông tin mới hoặc chỉnh sửa.

+ Được dùng để tham chiếu trong nhiều bảng khác như bảng HangHoa, bảng NhapHang…

+ Giúp quản lý dữ liệu tập trung, tránh trùng lặp và đảm bảo tính nhất quán trong toàn bộ hệ thống dữ liệu.

=> Kết luận: Bảng NhaCungCap được chọn làm bảng Master vì nó lưu trữ thông tin nền tảng, mang tính ổn định và phục vụ như một nguồn tham chiếu chung cho các bảng khác trong hệ thống. Việc tách riêng bảng này giúp chuẩn hóa cơ sở dữ liệu, giảm trùng lặp, tăng tính toàn vẹn và dễ dàng quản lý thông tin nhà cung cấp một cách tập trung.

\*/

CREATE OR ALTER PROCEDURE dump\_Insert\_NhaCungCap (@number INT)

AS

BEGIN

DECLARE @TrachNhiem TABLE (TrachNhiem NVARCHAR(50))

INSERT INTO @TrachNhiem

VALUES

(N'TNHH'), (N'CP'), (N'Đầu tư'), (N'Thực phẩm'), (N'Phân phối'), (N'TNHH'), (N'TNHH'), (N'TNHH')

DECLARE @Ten\_CTY TABLE (Ten\_CTY NVARCHAR(50))

INSERT INTO @Ten\_CTY

VALUES

(N'Đà Nẵng'), (N'Sông Hàn'), (N'Minh Quang'), (N'Ngọc Mai'), (N'Ngọc Mai'),

(N'Trường Sơn'), (N'Hoài Ngọc'), (N'Thịnh Phát'), (N'An Gia Khang'), (N'An Gia Khánh'),

(N'Gia Trường Phúc'), (N'Ngọc Bích'), (N'An Lộc')

DECLARE @TenDuong\_NCC TABLE (TenDuong\_NCC NVARCHAR(50))

INSERT INTO @TenDuong\_NCC

VALUES

(N'Hoàng Diệu'), (N'Đào Tấn'), (N'Tố Hữu'), (N'Trưng Nữ Vương'), (N'Đỗ Quang'),

(N'Nguyễn Hữu Thọ'), (N'Nguyễn Như Hạnh'), (N'Nguyễn Văn Thoại'), (N'Chu Văn An'), (N'Nguyễn Tất Thành'),

(N'Trần Cao Vân'), (N'Nguyễn Phước Nguyên'), (N'Phan Thanh'), (N'Tôn Thất Tùng'), (N'Nguyễn Đình Tứ'),

(N'Ngô Thì Nhậm'), (N'Nguyễn Chí Thanh'), (N'Phan Huy Chú'), (N'Lương Nhữ Hộc'), (N'Âu Cơ'), (N'Hải Phòng'),

(N'Lê Duẩn'), (N'Hà Huy Tập')

DECLARE @Quan\_NCC TABLE (Quan\_NCC NVARCHAR(50))

INSERT INTO @Quan\_NCC

VALUES

(N'Hải Châu'), (N'Ngũ Hành Sơn'), (N'Liên Chiểu'), (N'Thanh Khê'), (N'Cẩm Lệ'), (N'Sơn Trà')

DECLARE @num INT = 1

WHILE @num <= @number

BEGIN

-- Tạo tên nhà cung cấp ngẫu nhiên

DECLARE @Ten\_NCC NVARCHAR(50) = (SELECT TOP(1) 'Công ty' + ' ' + TrachNhiem + ' ' + Ten\_CTY FROM @TrachNhiem, @Ten\_CTY ORDER BY NEWID())

-- Tạo địa chỉ nhà cung cấp ngẫu nhiên

DECLARE @So\_NCC NVARCHAR(50), @rd BIT

SELECT @rd = CAST(ROUND(1 \* RAND(), 0) AS BIT)

IF @rd = 1

SET @So\_NCC = CAST(CAST(RAND() \* 179 AS INT) + 1 AS VARCHAR) + CHAR(RAND() \* 7 + 65)

ELSE

SET @So\_NCC = CAST(CAST(RAND() \* 179 AS INT) + 1 AS VARCHAR)

DECLARE @DiaChi\_NCC NVARCHAR(50) = (SELECT TOP(1) @So\_NCC + ', ' + TenDuong\_NCC + ', ' + Quan\_NCC + N', Đà Nẵng' FROM @TenDuong\_NCC, @Quan\_NCC ORDER BY NEWID())

-- Tạo số điện thoại ngẫu nhiên

DECLARE @SDT\_NCC VARCHAR(11) = '0'

DECLARE @digit INT = 1

SELECT @rd = CAST(ROUND(1 \* RAND(), 0) AS BIT)

IF @rd = 1

BEGIN

WHILE @digit <= 9

BEGIN

SET @SDT\_NCC = @SDT\_NCC + CAST(CAST(RAND() \* 10 AS INT) AS VARCHAR)

SET @digit = @digit + 1

END

END

ELSE

BEGIN

WHILE @digit <= 10

BEGIN

SET @SDT\_NCC = @SDT\_NCC + CAST(CAST(RAND() \* 10 AS INT) AS VARCHAR)

SET @digit = @digit + 1

END

END

-- Thêm

DECLARE @noti\_NV BIT

EXEC sp\_Insert\_NhaCungCap @Ten\_NCC, @DiaChi\_NCC, @SDT\_NCC, @noti\_NV OUT

SET @num = @num + 1

END

END

-----------------------------------------------------

EXEC dump\_Insert\_NhaCungCap 37

SELECT COUNT(\*) 'NhaCungCap' FROM NhaCungCap

SELECT TOP(5) \* FROM NhaCungCap ORDER BY NEWID()

-----------------------------------------------------

A screenshot of a computer

AI-generated content may be incorrect.

### Bảng KhachHang

-- KhachHang --> Master

/\*Lý do:

+ Vì bảng KhachHang là gốc để tạo các giao dịch.

+ Dữ liệu khách hàng không thay đổi thường xuyên.

+ Các bảng khác trong nghiệp vụ bán hàng đều phụ thuộc vào KhachHang.

+ Backup hay xuất dữ liệu dễ hơn.

=> Kết luận: Bảng KhachHang được chọn làm master vì nó là trung tâm của nghiệp vụ bán hàng, ít thay đổi, và các bảng khác của nghiệp vụ bán hàng đều phụ thuộc vào nó.

\*/

CREATE OR ALTER PROCEDURE dump\_Insert\_KhachHang (@number INT)

AS

BEGIN

-- Họ, tên đệm, tên

DECLARE @Ho\_KH TABLE (Ho\_KH NVARCHAR(15))

INSERT INTO @Ho\_KH

VALUES

(N'Châu'), (N'Triệu'), (N'Hoàng'), (N'Bùi'), (N'Tăng'), (N'Ngô'), (N'Hồ'), (N'Phan'), (N'Đoàn'), (N'Đỗ'),

(N'Trần'), (N'Thạch'), (N'Quách'), (N'Hà'), (N'Vương'), (N'Trịnh'), (N'Lê'), (N'Lâm'), (N'Tô'), (N'Đặng'),

(N'Lý'), (N'Huỳnh'), (N'Vũ'), (N'La'), (N'Dương'), (N'Kiều'), (N'Cao'), (N'Nguyễn')

DECLARE @TenDem\_KH TABLE (TenDem\_KH NVARCHAR(15))

INSERT INTO @TenDem\_KH

VALUES

(N'Nam'), (N'Thị'), (N'Hải'), (N'Ngọc'), (N'Văn'), (N'Trúc'), (N'Phương'), (N'Bảo'), (N'Trí'), (N'Minh Nhật'),

(N'Thế'), (N'Thanh'), (N'Khánh'), (N'Thảo'), (N'Tuệ'), (N'Quốc'), (N'Minh'), (N'Lê'), (N'Tấn'), (N'Thái'),

(N'Tú'), (N'Hồng'), (N'Bích'), (N'Thuỳ'), (N'Khôi'), (N'Gia'), (N'Thiên'), (N'Nhật')

DECLARE @Ten\_KH TABLE (Ten\_KH NVARCHAR(15))

INSERT INTO @Ten\_KH

VALUES

(N'Nam'), (N'Kim'), (N'Khang'), (N'Giang'), (N'Hưng'), (N'Tuấn'), (N'Bảo'), (N'Đạt'), (N'Trân'), (N'Phong'),

(N'Tâm'), (N'Lan'), (N'Phúc'), (N'Tài'), (N'Thư'), (N'Vỹ'), (N'Long'), (N'Quốc'), (N'Hà'), (N'Chi'),

(N'Tiên'), (N'Nhi'), (N'Đăng'), (N'Vy'), (N'Duyên'), (N'Nguyên'), (N'Thiên')

-- Tên đường, phường, quận, thành phố

DECLARE @TenDuong\_KH TABLE (TenDuong\_KH NVARCHAR(50))

INSERT INTO @TenDuong\_KH

VALUES

(N'Hoàng Diệu'), (N'Đào Tấn'), (N'Tố Hữu'), (N'Trưng Nữ Vương'), (N'Đỗ Quang'),

(N'Nguyễn Hữu Thọ'), (N'Nguyễn Như Hạnh'), (N'Nguyễn Văn Thoại'), (N'Chu Văn An'), (N'Nguyễn Tất Thành'),

(N'Trần Cao Vân'), (N'Nguyễn Phước Nguyên'), (N'Phan Thanh'), (N'Tôn Thất Tùng'), (N'Nguyễn Đình Tứ'),

(N'Ngô Thì Nhậm'), (N'Nguyễn Chí Thanh'), (N'Phan Huy Chú'), (N'Lương Nhữ Hộc')

DECLARE @Phuong\_KH TABLE (Phuong\_KH NVARCHAR(50))

INSERT INTO @Phuong\_KH

VALUES

(N'Hòa Xuân'), (N'Hòa Hiệp Nam'), (N'Hòa Minh'), (N'Phước Ninh'), (N'Khuê Trung'),

(N'Thuận Phước'), (N'Hòa Cường Nam'), (N'Xuân Hà'), (N'An Khê'), (N'Bình Hiên'),

(N'Tam Thuận'), (N'Nam Dương'), (N'Hòa Khê'), (N'Hòa An'), (N'Mỹ An'), (N'Hải Châu'), (N'Hòa Cường Bắc'), (N'Thạc Gián')

DECLARE @Quan\_KH TABLE (Quan\_KH NVARCHAR(50))

INSERT INTO @Quan\_KH

VALUES

(N'Hải Châu'), (N'Ngũ Hành Sơn'), (N'Liên Chiểu'), (N'Thanh Khê'), (N'Cẩm Lệ')

DECLARE @ThanhPho\_KH TABLE (ThanhPho\_KH NVARCHAR(50))

INSERT INTO @ThanhPho\_KH

VALUES

(N'Đà Nẵng')

DECLARE @num INT = 1

WHILE @num <= @number

BEGIN

-- Tạo tên khách hàng ngẫu nhiên

DECLARE @HoTen\_KH NVARCHAR(50) = (SELECT TOP(1) Ho\_KH + ' ' + TenDem\_KH + ' ' + Ten\_KH FROM @Ho\_KH, @TenDem\_KH, @Ten\_KH ORDER BY NEWID())

-- Tạo địa chỉ khách hàng ngẫu nhiên

DECLARE @SoNha\_KH NVARCHAR(50), @rd BIT

SELECT @rd = CAST(ROUND(1 \* RAND(), 0) AS BIT)

IF @rd = 1

SET @SoNha\_KH = CAST(CAST(RAND() \* 179 AS INT) + 1 AS VARCHAR) + CHAR(RAND() \* 7 + 65)

ELSE

SET @SoNha\_KH = CAST(CAST(RAND() \* 179 AS INT) + 1 AS VARCHAR)

DECLARE @DiaChi\_KH NVARCHAR(50) = (SELECT TOP(1) @SoNha\_KH + ', ' + TenDuong\_KH + ', ' + Phuong\_KH + ', ' + Quan\_KH + ', ' + ThanhPho\_KH FROM @TenDuong\_KH, @Phuong\_KH, @Quan\_KH, @ThanhPho\_KH ORDER BY NEWID())

-- Tạo số điện thoại ngẫu nhiên

DECLARE @SDT\_KH VARCHAR(10) = '09'

DECLARE @digit INT = 1

WHILE @digit <= 8

BEGIN

SET @SDT\_KH = @SDT\_KH + CAST(CAST(RAND() \* 10 AS INT) AS VARCHAR)

SET @digit = @digit + 1

END

-- Thêm

DECLARE @noti\_NV BIT

EXEC sp\_Insert\_KhachHang @HoTen\_KH, @DiaChi\_KH, @SDT\_KH, @noti\_NV OUT

SET @num = @num + 1

END

END

-----------------------------------------------------

EXEC dump\_Insert\_KhachHang 84

SELECT COUNT(\*) 'KhachHang' FROM KhachHang

SELECT TOP(5) \* FROM KhachHang ORDER BY NEWID()

-----------------------------------------------------

A screenshot of a computer

AI-generated content may be incorrect.

### Bảng NhanVien

-- NhanVien --> Master

/\*Lý do:

+ Mỗi dòng trong bảng NhanVien mô tả một thực thể nhân viên cụ thể (mã, tên, số điện thoại).

+ Dữ liệu của bảng này không mang tính thời điểm hoặc không phản ánh một sự kiện nghiệp vụ nào.

+ Nhân viên có thể được thêm mới, chỉnh sửa thông tin, nhưng không thường xuyên và không có tính chất phát sinh liên tục như dữ liệu giao dịch.

=> Kết luận: Bảng NhanVien được chọn làm bảng Master vì nó lưu thông tin tĩnh, dùng để mô tả nhân viên như một thực thể chính trong hệ thống.

\*/

CREATE OR ALTER PROCEDURE dump\_Insert\_NhanVien (@number INT)

AS

BEGIN

-- Họ, tên đệm, tên

DECLARE @Ho\_NV TABLE (Ho\_NV NVARCHAR(15))

INSERT INTO @Ho\_NV

VALUES

(N'Huỳnh'), (N'La'), (N'Tô'), (N'Hồ'), (N'Lê'), (N'Phạm'), (N'Phan'), (N'Quách'), (N'Triệu'), (N'Tạ'),

(N'Đỗ'), (N'Nguyễn'), (N'Châu'), (N'Võ'), (N'Dương'), (N'Hà'), (N'Hoàng'), (N'Lý'), (N'Trần'), (N'Tăng'),

(N'Thạch'), (N'Ngô'), (N'Bùi'), (N'Kiều'), (N'Lâm'), (N'Cao'), (N'Đặng'), (N'Mai')

DECLARE @TenDem\_NV TABLE (TenDem\_NV NVARCHAR(15))

INSERT INTO @TenDem\_NV

VALUES

(N'Tấn'), (N'Anh'), (N'Thị Bích'), (N'Kim'), (N'Quốc'), (N'Hữu'), (N'Đình'), (N'Hoàng Việt'), (N'Tú'), (N'Vĩnh'),

(N'Tường Vy'), (N'Gia'), (N'Bảo'), (N'Tường'), (N'Ngọc'), (N'Thái'), (N'Nguyễn Nhật'), (N'Thanh'), (N'Minh'), (N'Văn Minh'),

(N'Thành'), (N'Hoàng'), (N'Thị Kim'), (N'Thái Thanh'), (N'Đức'), (N'Hoàng Gia'), (N'Văn'), (N'Quang'), (N'Khánh'), (N'Thiện'),

(N'Hoài'), (N'Phương'), (N'Quốc Anh'), (N'Hải'), (N'Nhật')

DECLARE @Ten\_NV TABLE (Ten\_NV NVARCHAR(15))

INSERT INTO @Ten\_NV

VALUES

(N'Thắng'), (N'Vy'), (N'Hào'), (N'Sơn'), (N'Khoa'), (N'Thảo'), (N'Phương'), (N'Phúc'), (N'Duy'), (N'Lộc'),

(N'Cẩm'), (N'Hải'), (N'Duyên'), (N'Quỳnh'), (N'Ngân'), (N'Tùng'), (N'Phát'), (N'Châu'), (N'Kiệt'), (N'Xuân'),

(N'Ngọc'), (N'Hoa'), (N'Kim'), (N'Long'), (N'Bảo'), (N'Hương'), (N'Nhân'), (N'Tài'), (N'Trâm'), (N'Hạnh'),

(N'Ly'), (N'Tâm'), (N'Yến'), (N'Hồng'), (N'Minh'), (N'Toàn'), (N'Thái'), (N'Hiếu'), (N'Nam'), (N'Linh'),

(N'Thư'), (N'Huy'), (N'Hân'), (N'Dương'), (N'Tuyết'), (N'Thịnh'), (N'Tú'), (N'Mai')

DECLARE @num INT = 1

WHILE @num <= @number

BEGIN

-- Tạo tên nhân viên ngẫu nhiên

DECLARE @HoTen\_NV NVARCHAR(50) = (SELECT TOP(1) Ho\_NV + ' ' + TenDem\_NV + ' ' + Ten\_NV FROM @Ho\_NV, @TenDem\_NV, @Ten\_NV ORDER BY NEWID())

-- Tạo số điện thoại ngẫu nhiên

DECLARE @SDT\_NV VARCHAR(10) = '09'

DECLARE @digit INT = 1

WHILE @digit <= 8

BEGIN

SET @SDT\_NV = @SDT\_NV + CAST(CAST(RAND() \* 10 AS INT) AS VARCHAR)

SET @digit = @digit + 1

END

-- Thêm

DECLARE @noti\_NV BIT

EXEC sp\_Insert\_NhanVien @HoTen\_NV, @SDT\_NV, @noti\_NV OUT

SET @num = @num + 1

END

END

-----------------------------------------------------

EXEC dump\_Insert\_NhanVien 10

SELECT COUNT(\*) 'NhanVien' FROM NhanVien

SELECT TOP(5) \* FROM NhanVien ORDER BY NEWID()

-----------------------------------------------------

A screenshot of a computer

AI-generated content may be incorrect.

### Bảng NhapHang, CT\_NhapHang, GiaHangHoa, ThanhToan

-- NhapHang --> Transaction

/\*Lý do:

+ Vì dữ liệu từ bảng này có tính thời điểm (ví dụ Ngay\_Nhap ghi nhận ở thời điểm phát sinh giao dịch).

+ Dữ liệu này gắn liền với từng hoạt động nhập hàng cụ thể – không lặp lại.

+ Bao gồm thông tin thanh toán: Tong\_Thanh\_Tien, Tien\_Da\_Tra, nhân viên thực hiện (Ma\_NV), nhà cung cấp (Ma\_NCC) => tất cả đều là yếu tố nghiệp vụ.

+ Mỗi dòng dữ liệu lưu trữ các giao dịch thực tế với NCC thế nên cần phải lưu giữ nguyên vẹn để phục vụ kiểm toán hay đối chiếu công nợ, thống kê.

=> Kết luận: Bảng NhapHang được chọn làm bảng Transaction vì nó lưu trữ thông tin tổng quát của mỗi giao dịch nhập hàng phát sinh trong thực tế. Mỗi dòng trong bảng phản ánh một sự kiện nhập hàng cụ thể tại một thời điểm cụ thể, không có tính trùng lặp và không nên thay đổi tùy tiện sau khi đã ghi nhận.

\*/

-- CT\_NhapHang --> Transaction

/\*Lý do:

+ Bảng này là phần mở rộng chi tiết của bảng NhapHang.

+ Cho biết mỗi hóa đơn nhập bao gồm những mặt hàng nào, số lượng, hạn sử dụng, thành tiền.

+ Được xác định qua khóa ngoại SoHD\_NhapHang => không thể tồn tại độc lập.

+ Giao dịch nào có hàng gì thì chi tiết được lưu tại đây.

=> Kết luận: Bảng CT\_NhapHang được chọn làm bảng Transaction vì nó chứa chi tiết từng mặt hàng được nhập trong mỗi hóa đơn nhập hàng. Mỗi dòng thể hiện một sự kiện nhập cụ thể của từng loại hàng, có ảnh hưởng trực tiếp đến tồn kho và theo dõi hạn dùng.

\*/

-- GiaHangHoa --> Transaction

/\*Lý do:

+ Bảng này lưu trữ các mức giá nhập và bán của từng hàng hóa trong từng thời điểm cụ thể.

+ Mỗi dòng trong bảng phản ánh một lần thay đổi giá của hàng hóa, thường được cập nhật khi có biến động thị trường hoặc chính sách bán hàng.

+ Dữ liệu này không phải là thông tin nền tảng tĩnh như tên hàng hóa, đơn vị tính (đã có trong HangHoa), mà là thông tin biến động mang tính thời điểm, có thể thay đổi theo từng đợt.

=> Kết luận: Bảng GiaHangHoa được chọn làm bảng Transaction vì nó phản ánh các sự kiện điều chỉnh giá bán và giá nhập hàng hóa theo thời gian, có tính thời điểm rõ ràng.

\*/

-- ThanhToan --> Transaction

/\*Lý do:

+ Bảng này ghi lại các giao dịch thanh toán với nhà cung cấp, gắn với từng hóa đơn nhập hàng cụ thể.

+ Các giao dịch thanh toán xảy ra thường xuyên.

+ Mỗi dòng dữ liệu là một lần thanh toán riêng biệt, có mã thanh toán, ngày thanh toán, số tiền, nợ còn lại.

+ Bảng quan trọng cho việc theo dõi công nợ, quản lý dòng tiền.

=> Kết luận: Bảng ThanhToan được chọn làm bảng transaction vì dữ liệu trong bảng này phát sinh thường xuyên, phụ thuộc vào các bảng khác. Ghi nhận chi tiết hoạt động thanh toán giúp quản lý tình hình tài chính của của hàng.

\*/

CREATE OR ALTER TRIGGER trigger\_afterInsert\_CT\_NhapHang ON CT\_NhapHang

AFTER INSERT

AS

BEGIN

DECLARE @SoHD\_NhapHang VARCHAR(11), @Ma\_HH VARCHAR(11), @So\_Luong INT, @Thanh\_Tien MONEY

SELECT @SoHD\_NhapHang = SoHD\_NhapHang, @Ma\_HH = Ma\_HH, @So\_Luong = So\_Luong, @Thanh\_Tien = Thanh\_Tien FROM inserted

UPDATE HangHoa

SET SL\_Ton\_Kho = ISNULL(SL\_Ton\_Kho, 0) + @So\_Luong

WHERE Ma\_HH = @Ma\_HH

UPDATE NhapHang

SET Tong\_Thanh\_Tien = ISNULL(Tong\_Thanh\_Tien, 0) + (@Thanh\_Tien \* @So\_Luong)

WHERE SoHD\_NhapHang = @SoHD\_NhapHang

END

CREATE OR ALTER TRIGGER trigger\_afterDelete\_CT\_NhapHang ON CT\_NhapHang

AFTER DELETE

AS

BEGIN

DECLARE @SoHD\_NhapHang VARCHAR(11), @Ma\_HH VARCHAR(11), @So\_Luong INT, @Thanh\_Tien MONEY

SELECT @SoHD\_NhapHang = SoHD\_NhapHang, @Ma\_HH = Ma\_HH, @So\_Luong = So\_Luong, @Thanh\_Tien = Thanh\_Tien FROM deleted

UPDATE HangHoa

SET SL\_Ton\_Kho = SL\_Ton\_Kho - @So\_Luong

WHERE Ma\_HH = @Ma\_HH

UPDATE NhapHang

SET Tong\_Thanh\_Tien = Tong\_Thanh\_Tien - (@Thanh\_Tien \* @So\_Luong)

WHERE SoHD\_NhapHang = @SoHD\_NhapHang

END

CREATE OR ALTER TRIGGER trigger\_afterInsert\_ThanhToan ON ThanhToan

AFTER INSERT

AS

BEGIN

DECLARE @Tong\_Tien MONEY, @Tong\_Tien\_Giao\_Dich MONEY

DECLARE @Ma\_NCC VARCHAR(11) = (SELECT NH.Ma\_NCC FROM inserted i JOIN NhapHang NH ON i.SoHD\_NhapHang = NH.SoHD\_NhapHang)

SELECT @Tong\_Tien = SUM(Tong\_Thanh\_Tien) FROM NhapHang WHERE Ma\_NCC = @Ma\_NCC

SELECT @Tong\_Tien\_Giao\_Dich = SUM(Tien\_Giao\_Dich) FROM ThanhToan TT JOIN NhapHang NH ON TT.SoHD\_NhapHang = NH.SoHD\_NhapHang WHERE NH.Ma\_NCC = @Ma\_NCC

UPDATE NhaCungCap

SET Tong\_Tien\_No = @Tong\_Tien - @Tong\_Tien\_Giao\_Dich

WHERE Ma\_NCC = @Ma\_NCC

END

CREATE OR ALTER PROC dump\_Insert\_NhapHang\_Ct\_NhapHang\_GiaHangHoa\_ThanhToan (@number INT)

AS

BEGIN

DECLARE @out BIT, @DateBatDau DATE = '2012-01-01',

@Ma\_NV VARCHAR(11), @Ma\_NCC VARCHAR(11), @SoHD\_NhapHang VARCHAR(11),

@Ma\_HH VARCHAR(11), @So\_Luong INT, @Han\_Su\_Dung DATE, @Thanh\_Tien MONEY

DECLARE @num INT = 1

WHILE @num <= @number

BEGIN

DECLARE @day INT = CAST(RAND() \* 7 AS INT) + 1

DECLARE @i INT = 1

WHILE @i <= @day

BEGIN

SELECT @Ma\_NV = Ma\_NV, @Ma\_NCC = Ma\_NCC FROM NhanVien, NhaCungCap ORDER BY NEWID()

EXEC sp\_Insert\_NhapHang @Ma\_NV, @Ma\_NCC, @DateBatDau, @out OUT

SELECT @SoHD\_NhapHang = MAX(SoHD\_NhapHang) FROM NhapHang

DECLARE @soCT INT= CAST(RAND() \* 5 AS INT) + 1

DECLARE @j INT = 1

WHILE @j <= @soCT

BEGIN

SELECT @Ma\_HH = Ma\_HH FROM HangHoa ORDER BY NEWID()

SELECT @So\_Luong = CAST(RAND() \* 10 AS INT) + 1

SELECT @Han\_Su\_Dung = DATEADD(MONTH, CAST(RAND() \* 8 AS INT) + 1, @DateBatDau)

SELECT @Thanh\_Tien = CAST((FLOOR(RAND() \* 150) \* 2 + 2) \* 1000 AS MONEY)

EXEC sp\_Insert\_CT\_NhapHang @SoHD\_NhapHang, @Ma\_HH, @So\_Luong, @Han\_Su\_Dung, @Thanh\_Tien, @out OUT

SET @j = @j + 1

END

DECLARE @Ngay\_Bat\_Dau DATE = DATEADD(DAY, 1, @DateBatDau), @Ngay\_Ket\_Thuc DATE = DATEADD(DAY, -5, @Han\_Su\_Dung)

EXEC sp\_Insert\_GiaHangHoa @Ma\_HH, @Thanh\_Tien, @Ngay\_Bat\_Dau, @Ngay\_Ket\_Thuc, @out OUT

DECLARE @Tong\_Thanh\_Tien MONEY = (SELECT Tong\_Thanh\_Tien FROM NhapHang WHERE SoHD\_NhapHang = @SoHD\_NhapHang)

DECLARE @Tong\_Tien\_Giao\_Dich MONEY = 0, @Tien\_Giao\_Dich MONEY, @Ngay\_Thanh\_Toan DATE = @DateBatDau

WHILE @Tong\_Tien\_Giao\_Dich < @Tong\_Thanh\_Tien

BEGIN

SET @Tien\_Giao\_Dich = CAST((FLOOR(RAND() \* (@Tong\_Thanh\_Tien - @Tong\_Tien\_Giao\_Dich)) + 1) AS MONEY)

SET @Tong\_Tien\_Giao\_Dich = @Tong\_Tien\_Giao\_Dich + @Tien\_Giao\_Dich

EXEC sp\_Insert\_ThanhToan @SoHD\_NhapHang, @Ngay\_Thanh\_Toan, @Tien\_Giao\_Dich, @out OUT

SET @Ngay\_Thanh\_Toan = DATEADD(DAY, CAST(RAND() \* 10 AS INT), @Ngay\_Thanh\_Toan)

END

UPDATE NhapHang

SET Tien\_Da\_Tra = @Tong\_Thanh\_Tien

WHERE SoHD\_NhapHang = @SoHD\_NhapHang

SET @i = @i + 1

END

SET @DateBatDau = DATEADD(DAY, 1, @DateBatDau)

SET @num = @num + 1

END

END

-----------------------------------------------------

EXEC dump\_Insert\_NhapHang\_Ct\_NhapHang 1000000

SELECT COUNT(\*) 'NhapHang' FROM NhapHang

SELECT TOP(5) \* FROM NhapHang ORDER BY NEWID()

SELECT COUNT(\*) 'CT\_NhapHang' FROM CT\_NhapHang

SELECT TOP(5) \* FROM CT\_NhapHang ORDER BY NEWID()

SELECT COUNT(\*) 'GiaHangHoa' FROM GiaHangHoa

SELECT TOP(5) \* FROM GiaHangHoa ORDER BY NEWID()

SELECT COUNT(\*) 'ThanhToan' FROM ThanhToan

SELECT TOP(5) \* FROM ThanhToan ORDER BY NEWID()

-----------------------------------------------------

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

### Bảng BanHang, CT\_BanHang

-- BanHang --> Transaction

/\*Lý do:

+ Bảng BanHang ghi nhận các giao dịch bán hàng diễn ra hàng ngày.

+ Dữ liệu trong bảng thường xuyên thay đổi và cập nhật liên tục.

+ Mỗi bản ghi đại diện cho một hoạt động kinh doanh cụ thể (như đơn hàng, hóa đơn…).

+ Có quan hệ mật thiết với nhiều bảng khác như khách hàng, hàng hóa, nhân viên, nhà cung cấp…

+ Là trung tâm của các phân tích dữ liệu trong hệ thống bán hàng

=> Kết luận: Bảng BanHang được chọn làm bảng Transaction vì nó phản ánh các nghiệp vụ phát sinh liên tục trong hoạt động bán hàng. Tính chất động, thay đổi thường xuyên, cùng vai trò trung tâm trong việc kết nối và phân tích dữ liệu kinh doanh khiến bảng này phù hợp với đặc điểm của một bảng giao dịch trong thiết kế cơ sở dữ liệu chuẩn hóa.

\*/

-- CT\_BanHang --> Transaction

/\*Lý do:

+ Là bảng trung tâm ghi nhận các giao dịch bán hàng chi tiết, nằm ở cuối chuỗi nghiệp vụ.

+ Giao dịch xảy ra thường xuyên.

+ Nó không tồn tại độc lập, luôn cần thông tin từ các bảng "master".

+ Dữ liệu có tính biến động cao, đúng với đặc điểm của bảng transaction

+ Là bảng rất quan trọng trong nghiệp vụ bán hàng.

=> Kết luận: Bảng CT\_BanHang được chọn làm bảng transaction vì nó phản ánh dữ liệu giao dịch phát sinh trong quá trình bán hàng, phụ thuộc vào các bảng khác, và có tần suất thay đổi cao. Đây là bảng chi tiết hóa nghiệp vụ và là nơi lưu dấu các hoạt động kinh doanh thực tế.

\*/

CREATE OR ALTER TRIGGER trigger\_afterInsert\_CT\_BanHang ON CT\_BanHang

AFTER INSERT

AS

BEGIN

DECLARE @SoHD\_BanHang VARCHAR(11), @Ma\_GHH VARCHAR(11), @So\_Luong INT, @Thanh\_Tien MONEY

SELECT @SoHD\_BanHang = SoHD\_BanHang, @Ma\_GHH = Ma\_GHH, @So\_Luong = So\_Luong, @Thanh\_Tien = Thanh\_Tien FROM inserted

DECLARE @Ma\_HH VARCHAR(11) = (SELECT Ma\_HH FROM GiaHangHoa WHERE Ma\_GHH = @Ma\_GHH)

UPDATE HangHoa

SET SL\_Ton\_Kho = ISNULL(SL\_Ton\_Kho, 0) - @So\_Luong

WHERE Ma\_HH = @Ma\_HH

UPDATE BanHang

SET Tong\_Thanh\_Tien = ISNULL(Tong\_Thanh\_Tien, 0) + @Thanh\_Tien

WHERE SoHD\_BanHang = @SoHD\_BanHang

END

CREATE OR ALTER TRIGGER trigger\_afterDelete\_CT\_BanHang ON CT\_BanHang

AFTER DELETE

AS

BEGIN

DECLARE @SoHD\_BanHang VARCHAR(11), @Ma\_GHH VARCHAR(11), @So\_Luong INT, @Thanh\_Tien MONEY

SELECT @SoHD\_BanHang = SoHD\_BanHang, @Ma\_GHH = Ma\_GHH, @So\_Luong = So\_Luong, @Thanh\_Tien = Thanh\_Tien FROM deleted

DECLARE @Ma\_HH VARCHAR(11) = (SELECT Ma\_HH FROM GiaHangHoa WHERE Ma\_GHH = @Ma\_GHH)

UPDATE HangHoa

SET SL\_Ton\_Kho = SL\_Ton\_Kho + @So\_Luong

WHERE Ma\_HH = @Ma\_HH

UPDATE BanHang

SET Tong\_Thanh\_Tien = Tong\_Thanh\_Tien - @Thanh\_Tien

WHERE SoHD\_BanHang = @SoHD\_BanHang

END

CREATE OR ALTER PROC dump\_Insert\_BanHang\_Ct\_BanHang (@number INT)

AS

BEGIN

DECLARE @out BIT, @DateBatDau DATE = '2012-01-01',

@Ma\_NV VARCHAR(11), @Ma\_KH VARCHAR(11), @SoHD\_BanHang VARCHAR(11),

@Ma\_GHH VARCHAR(11), @So\_Luong INT, @Thanh\_Tien MONEY

DECLARE @num INT = 1

WHILE @num <= @number

BEGIN

DECLARE @day INT = CAST(RAND() \* 7 AS INT) + 1

DECLARE @i INT = 1

WHILE @i <= @day

BEGIN

SELECT @Ma\_NV = Ma\_NV, @Ma\_KH = Ma\_KH FROM NhanVien, KhachHang ORDER BY NEWID()

EXEC sp\_Insert\_BanHang @Ma\_NV, @Ma\_KH, @DateBatDau, @out OUT

SELECT @SoHD\_BanHang = MAX(SoHD\_BanHang) FROM BanHang

DECLARE @soCT INT= CAST(RAND() \* 5 AS INT) + 1

DECLARE @j INT = 1

WHILE @j <= @soCT

BEGIN

SELECT @Ma\_GHH = Ma\_GHH FROM GiaHangHoa ORDER BY NEWID()

SELECT @So\_Luong = CAST(RAND() \* 10 AS INT) + 1

SELECT @Thanh\_Tien = Gia\_Ban FROM GiaHangHoa WHERE Ma\_GHH = @Ma\_GHH

EXEC sp\_Insert\_CT\_BanHang @SoHD\_BanHang, @Ma\_GHH, @So\_Luong, @out OUT

SET @j = @j + 1

END

UPDATE BanHang

SET Tong\_Thanh\_Tien = (SELECT SUM(Thanh\_Tien) FROM CT\_BanHang WHERE SoHD\_BanHang = @SoHD\_BanHang)

WHERE SoHD\_BanHang = @SoHD\_BanHang

SET @i = @i + 1

END

SET @DateBatDau = DATEADD(DAY, 1, @DateBatDau)

SET @num = @num + 1

END

END

-----------------------------------------------------

EXEC dump\_Insert\_BanHang\_Ct\_BanHang 1000000

SELECT COUNT(\*) 'BanHang' FROM BanHang

SELECT TOP(5) \* FROM BanHang ORDER BY NEWID()

SELECT COUNT(\*) 'CT\_BanHang' FROM CT\_BanHang

SELECT TOP(5) \* FROM CT\_BanHang ORDER BY NEWID()

-----------------------------------------------------

A screenshot of a computer

AI-generated content may be incorrect.