**Xây dựng cơ sở dữ liệu (R3)**

﻿﻿create database CTPhanPhoi

use CTPhanPhoi

CREATE TABLE NhaCungCap

(

MaNCC VARCHAR(10) NOT NULL,

TenNCC VARCHAR(30) NOT NULL,

TenNGH VARCHAR(30) NOT NULL,

MaThue VARCHAR(15) NOT NULL,

DiaChi VARCHAR(100) NOT NULL,

STK VARCHAR(15) NOT NULL,

PRIMARY KEY (MaNCC)

);

CREATE TABLE KhachHang

(

MaKH VARCHAR(10) NOT NULL,

TenKH VARCHAR(30) NOT NULL,

SDTKH VARCHAR(10) NOT NULL,

DiaChiKH VARCHAR(50) NOT NULL,

PRIMARY KEY (MaKH)

);

CREATE TABLE NhanVien

(

MaNV VARCHAR(10) NOT NULL,

TenNV VARCHAR(30) NOT NULL,

ViTri VARCHAR(30) NOT NULL,

PRIMARY KEY (MaNV)

);

CREATE TABLE DatHang

(

MaDH VARCHAR(10) NOT NULL,

NgayLDH DATE NOT NULL,

TongTien FLOAT NOT NULL,

MaKH VARCHAR(10) NOT NULL,

MaNV VARCHAR(10) NOT NULL,

PRIMARY KEY (MaDH),

FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV),

FOREIGN KEY (MaKH) REFERENCES KhachHang(MaKH)

);

CREATE TABLE HangHoaXuat

(

MaHang VARCHAR(10) NOT NULL,

TenHang VARCHAR(50) NOT NULL,

DVT VARCHAR(10) NOT NULL,

DonGiaXuat FLOAT NOT NULL,

PRIMARY KEY (MaHang)

);

CREATE TABLE DatHangChiTiet

(

MaDH VARCHAR(10) NOT NULL,

SoLuong INT NOT NULL,

ThanhTien FLOAT NOT NULL,

MaHang VARCHAR(10) NOT NULL,

PRIMARY KEY (MaDH),

FOREIGN KEY (MaHang) REFERENCES HangHoaXuat(MaHang),

FOREIGN KEY (MaDH) REFERENCES DatHang(MaDH)

);

CREATE TABLE ThanhToan

(

SoPTT VARCHAR(10) NOT NULL,

MaNV VARCHAR(10) NOT NULL,

NgayTT DATE NOT NULL,

Pmethod VARCHAR(20) NOT NULL,

VAT FLOAT NOT NULL,

TongTien FLOAT NOT NULL,

MaNCC VARCHAR(10) NOT NULL,

PRIMARY KEY (SoPTT),

FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV),

FOREIGN KEY (MaNCC) REFERENCES NhaCungCap(MaNCC)

);

CREATE TABLE Kho

(

MaKho VARCHAR(5) NOT NULL,

DiaChi VARCHAR(50) NOT NULL,

PRIMARY KEY (MaKho)

);

CREATE TABLE XuatKho

(

SoPXK VARCHAR(7) NOT NULL,

NgayXuat DATE NOT NULL,

LyDo VARCHAR(50) NOT NULL,

TongTien FLOAT NOT NULL,

MaKho VARCHAR(5) NOT NULL,

PRIMARY KEY (SoPXK),

FOREIGN KEY (MaKho) REFERENCES Kho(MaKho)

);

CREATE TABLE NhapKho

(

MaPNK VARCHAR(7) NOT NULL,

NgayNhap DATE NOT NULL,

TongTien FLOAT NOT NULL,

MaKho VARCHAR(5) NOT NULL,

MaNCC VARCHAR(10) NOT NULL,

PRIMARY KEY (MaPNK),

FOREIGN KEY (MaNCC) REFERENCES NhaCungCap(MaNCC),

FOREIGN KEY (MaKho) REFERENCES Kho(MaKho)

);

CREATE TABLE XuatKhoChiTiet

(

SoPXK VARCHAR(7) NOT NULL,

TheoYC INT NOT NULL,

ThucXuat INT NOT NULL,

ThanhTien FLOAT NOT NULL,

MaHang VARCHAR(10) NOT NULL,

PRIMARY KEY (SoPXK),

FOREIGN KEY (SoPXK) REFERENCES XuatKho(SoPXK),

FOREIGN KEY (MaHang) REFERENCES HangHoaXuat(MaHang)

);

CREATE TABLE HangHoaNhap

(

MaHang VARCHAR(10) NOT NULL,

TenHang VARCHAR(50) NOT NULL,

DVT VARCHAR(10) NOT NULL,

DonGiaNhap FLOAT NOT NULL,

PRIMARY KEY (MaHang)

);

CREATE TABLE ThanhToanChiTiet

(

SoPTT VARCHAR(10) NOT NULL,

MaHang VARCHAR(10) NOT NULL,

SoLuong INT NOT NULL,

ThanhTien FLOAT NOT NULL,

PRIMARY KEY (SoPTT),

FOREIGN KEY (SoPTT) REFERENCES ThanhToan(SoPTT),

FOREIGN KEY (MaHang) REFERENCES HangHoaNhap(MaHang)

);

CREATE TABLE NhapKhoChiTiet

(

MaPNK VARCHAR(7) NOT NULL,

TheoCT INT NOT NULL,

ThucNhap INT NOT NULL,

ThanhTien FLOAT NOT NULL,

MaHang VARCHAR(10) NOT NULL,

PRIMARY KEY (MaPNK),

FOREIGN KEY (MaHang) REFERENCES HangHoaNhap(MaHang),

FOREIGN KEY (MaPNK) REFERENCES NhapKho(MaPNK)

);

GO;

-- Nhap 1001 du lieu : Bang Kh

CREATE OR ALTER PROCEDURE K1

AS

BEGIN

DECLARE @stt INT = 1,

@doi INT = 1000000,

@sdt int = 905000000;

WHILE @stt <= 1001 -- Chèn 1001 bản ghi

BEGIN

INSERT INTO KhachHang (MaKH, TenKH, SDTKH, DiaChiKH)

VALUES (CAST(@doi AS VARCHAR(10)),

'Khach Hang ' + CAST(@stt AS VARCHAR(10)),

'0' + CAST(@sdt AS VARCHAR(10)),

'Dia Chi ' + CAST(@stt AS VARCHAR(10)));

SET @stt = @stt + 1;

set @doi = @doi +1

set @sdt = @sdt + 1

END

END

GO;

EXEC K1

SELECT \* FROM KHACHHANG;

GO;

-- Bang Nv

CREATE OR ALTER PROCEDURE K2

AS

BEGIN

DECLARE @stt INT = 1,

@doi INT = 1000000

WHILE @stt <= 1001 -- Chèn 1001 bản ghi

BEGIN

INSERT INTO NhanVien(MaNV,TenNV,ViTri)

VALUES ('NV\_' + CAST(@doi AS VARCHAR(10)),

'Nhan vien ' + CAST(@stt AS VARCHAR(10)),

'Vi Tri ' + CAST(@stt AS VARCHAR(10)));

SET @stt = @stt + 1;

set @doi = @doi +1

END

END;

GO;

EXEC K2

SELECT \* FROM NHANVIEN;

GO;

-- Bang DatHang

Create or alter proc K3

as

Begin

declare @tt int= 1000000,

@ma int = 1

while @ma < 1001

begin

insert into DatHang (MaDH,NgayLDH,TongTien,MaKH,MaNV)

values (

'DH\_' + cast(@tt as varchar(10)),

getdate(),

cast(@tt as varchar(10)),

cast(@tt as varchar(10)),

'NV\_' + cast(@tt as varchar(10))

)

set @tt = @tt + 1

set @ma = @ma + 1

end

end

GO;

EXEC K3

SELECT \* FROM DATHANG;

GO;

-- K4 Bang HangHoaXuat

Create or alter proc K4

as

Begin

declare @tt int= 1000000,

@dvt varchar(50) = 'Goi',

@ma int = 1

while @ma < 1001

begin

insert into HangHoaXuat(MaHang,TenHang,DVT,DonGiaXuat)

values (

'H\_' + cast(@tt as varchar(10)),

'Hang Hoa '+ cast(@ma as varchar(10)),

cast(@dvt as varchar(50)),

cast(@tt as varchar(10))

)

if @ma > 500

set @dvt = 'Cay'

set @tt = @tt + 1

set @ma = @ma + 1

end

end

GO;

EXEC K4

SELECT \* FROM HANGHOAXUAT;

GO;

-- K5 DatHangChiTiet

Create or alter proc K5

as

Begin

declare @tt int= 1000000,

@sl int = 10,

@ma int = 1

while @ma < 1001

begin

insert into DatHangChiTiet(MaDH,SoLuong,ThanhTien,MaHang)

values (

'DH\_' + cast(@tt as varchar(10)),

cast(@sl as varchar(10)),

cast(@tt as varchar(50)),

'H\_'+ cast(@tt as varchar(10))

)

if @sl > 50

set @sl = 10

set @sl= @sl + 1

set @tt = @tt + 1

set @ma = @ma + 1

end

end

GO;

EXEC K5

SELECT \* FROM DATHANGCHITIET;

GO;

--K6 Kho

Create or alter proc K6

as

Begin

declare @tt int= 1000,

@ma int = 1

while @ma < 1001

begin

insert into Kho(MaKho,DiaChi)

values (

'K' + cast(@tt as varchar(10)),

'Dia Chi Kho '+ cast(@ma as varchar(10))

)

set @tt = @tt + 1

set @ma = @ma + 1

end

end

GO;

EXEC K6

SELECT \* FROM KHO;

GO;

-- K7 XuatKho

Create or alter proc K7

as

Begin

declare @tt int= 1000000,

@stt int = 1000,

@LyDo varchar(50) = 'Ban Hang',

@ma int = 1

while @ma < 1001

begin

insert into XuatKho(SoPXK,NgayXuat,LyDo,TongTien,MaKho)

values (

'PXK' + cast(@stt as varchar(10)),

getdate(),

cast(@LyDo as varchar(50)),

cast(@tt as varchar(10)),

'K' + cast(@stt as varchar(10))

)

if @ma > 500

set @LyDo = 'Bo Sung'

set @tt = @tt + 1

set @ma = @ma + 1

set @stt = @stt +1

end

end

GO;

EXEC K7

SELECT \* FROM XUATKHO;

GO;

-- K8 Bang XuatKhoChiTiet

Create or alter proc K8

as

Begin

declare @tt int= 1000000,

@stt int = 1000,

@yc int = 4,

@ma int = 1

while @ma < 1001

begin

insert into XuatKhoChiTiet(SoPXK,TheoYC,ThucXuat,ThanhTien,MaHang)

values (

'PXK' + cast(@stt as varchar(10)),

cast(@yc + 1 as varchar(10)),

cast(@yc as varchar(10)),

cast(@tt as varchar(10)),

'H\_' + cast(@tt as varchar(10))

)

if @ma = 500

set @yc = 1

set @yc = @yc + 1

set @tt = @tt + 1

set @ma = @ma + 1

set @stt = @stt +1

end

end

GO;

EXEC K8

SELECT \* FROM XUATKHOCHITIET

GO;

-- K9 Bang NCC

Create or alter proc K9

as

Begin

declare @stt int = 1000,

@stk int = 12345678,

@ma int = 1

while @ma < 1001

begin

insert into NhaCungCap(MaNCC,TenNCC,TenNGH,MaThue,DiaChi,STK)

values (

'MC\_' + cast(@stt as varchar(10)),

'Nha Cung Cap ' + cast(@ma as varchar(10)),

'Nguoi Giao Hang '+ cast(@ma as varchar(10)),

'MT\_' + cast(@stt as varchar(10)),

'Dia Chi NCC '+ cast(@ma as varchar(10)),

cast(@stk as varchar(10))

)

set @stk = @stk + 1

set @ma = @ma + 1

set @stt = @stt +1

end

end

GO;

EXEC K9

SELECT \* FROM NHACUNGCAP

GO;

-- K10 Bang NhapKho

Create or alter proc K10

as

Begin

declare @tt int= 1000000,

@stt int = 1000,

@ma int = 1

while @ma < 1001

begin

insert into NhapKho(MaPNK,NgayNhap,TongTien,MaKho,MaNCC)

values (

'PNK' + cast(@stt as varchar(10)),

getdate(),

cast(@tt as varchar(10)),

'K' + cast(@stt as varchar(10)),

'MC\_' + cast(@stt as varchar(10))

)

set @tt = @tt + 1

set @ma = @ma + 1

set @stt = @stt +1

end

end

GO;

EXEC K10

SELECT \* FROM NHAPKHO;

GO;

-- K11 Bang HangHoaNhap

Create or alter proc K11

as

Begin

declare @tt int= 1000000,

@dvt varchar(50) = 'Goi',

@ma int = 1

while @ma < 1001

begin

insert into HangHoaNhap(MaHang,TenHang,DVT,DonGiaNhap)

values (

'H\_' + cast(@tt as varchar(10)),

'Hang Hoa '+ cast(@ma as varchar(10)),

cast(@dvt as varchar(50)),

cast(@tt as varchar(10))

)

if @ma > 500

set @dvt = 'Cay'

set @tt = @tt + 1

set @ma = @ma + 1

end

end

go;

EXEC K11

SELECT \* FROM HANGHOANHAP;

GO;

-- K12 Bang NhapKhoChiTiet

Create or alter proc K12

as

Begin

declare @tt int= 1000000,

@stt int = 1000,

@yc int = 4,

@ma int = 1

while @ma < 1001

begin

insert into NhapKhoChiTiet(MaPNK,TheoCT,ThucNhap,ThanhTien,MaHang)

values (

'PNK' + cast(@stt as varchar(10)),

cast(@yc + 1 as varchar(10)),

cast(@yc as varchar(10)),

cast(@tt as varchar(10)),

'H\_' + cast(@tt as varchar(10))

)

set @yc = @yc + 1

set @tt = @tt + 1

set @ma = @ma + 1

set @stt = @stt +1

end

end

GO;

EXEC K12

SELECT \* FROM NHAPKHOCHITIET;

GO;

-- K13 Bang ThanhToan

Create or alter proc K13

as

Begin

declare @tt int= 1000000,

@stt int = 1000,

@Met varchar(50) = 'Tien Mat',

@tax int = 1,

@ma int = 1

while @ma < 1001

begin

insert into ThanhToan(SoPTT,MaNV,NgayTT,Pmethod,VAT,TongTien,MaNCC)

values (

'PTT\_' + cast(@stt as varchar(10)),

'NV\_'+ cast(@tt as varchar(10)),

getdate(),

@met,

cast(@tax as varchar(10)),

cast(@tt as varchar(50)),

'MC\_' + cast(@stt as varchar(10))

)

if @ma = 500

set @Met = 'Chuyen Khoan'

set @tax = 1

set @tax = @tax +1

set @tt = @tt + 1

set @ma = @ma + 1

set @stt = @stt +1

end

end

GO;

EXEC K13

SELECT \* FROM THANHTOAN;

GO;

-- K14 ThanhToanChiTiet

Create or alter proc K14

as

Begin

declare @tt int= 1000000,

@stt int = 1000,

@ma int = 1

while @ma < 1001

begin

insert into ThanhToanChiTiet(SoPTT,MaHang,SoLuong,ThanhTien)

values (

'PTT\_' + cast(@stt as varchar(10)),

'H\_'+ cast(@tt as varchar(10)),

cast(@ma as varchar(10)),

cast(@tt as varchar(50))

)

set @tt = @tt + 1

set @ma = @ma + 1

set @stt = @stt +1

end

end

go;

EXEC K14

SELECT \* FROM THANHTOANCHITIET;

GO;

**Xây dựng module tạo dữ liệu dump cho các bảng trong cơ sở dữ liệu (R4)**

1. **Tổng hợp các module nhằm tối ưu dữ liệu**
   1. Module tính toán và kiểm tra tổng tiền của các hàng hóa trong một đơn hàng có chính xác và hợp lệ (tổng tiền không âm)
   2. Module kiểm tra xem một loại hàng hóa đã có trong kho hay chưa, nếu có thì thông báo “Đã có”, ngược lại thì thêm hàng hóa mới vào kho. Yêu cầu đối với hàng hóa mới: đơn giá phải > 0 (MaHang, TenHang, DVT, DonGiaNhap)
   3. Module tính tổng doanh thu của toàn bộ đơn hàng theo tháng/quý/năm.
   4. Module tính toán lợi nhuận của tháng/quý/năm.
   5. Module kiểm tra số lượng của mỗi loại hàng hóa còn trong kho tại thời điểm kiểm tra.
   6. Module tính tổng chi phí hàng hóa nhập theo tháng/quý/năm.
   7. Module thêm khách hàng, nếu trong csdl Mã khách hàng đã tồn tại thì thông báo “Đã có”, ngược lại thì thêm thông tin của khách hàng mới (Mã khách hàng, tên khách hàng, số điện thoại, địa chỉ)
   8. Module thêm thông tin của một nhân viên mới, nếu trong csdl đã có mã nhân viên thì thông báo “Đã có”, ngược lại thì thêm thông tin của nhân viên mới (Mã nhân viên, tên nhân viên, vị trí).
   9. Module thêm nhà cung cấp, nếu trong csdl Mã nhà cung cấp đã tồn tại thì thông báo “Đã có”, ngược lại thì thêm thông tin của nhà cung cấp (Mã nhà cung cấp, tên nhà cung cấp, tên người giao hàng, mã thuế, địa chỉ, stk)
   10. Module tìm tháng có doanh thu cao nhất và thấp nhất trong năm.
   11. Module tìm ra top 5 khách hàng đặt nhiều đơn hàng nhất trong năm
   12. Module tìm ra top 10 sản phẩm được bán nhiều nhất trong tháng với điều kiện giá của sản phẩm trên 50.000 đồng.
   13. Module tìm ra top 3 nhân viên bán được nhiều đơn hàng nhất trong năm
2. **Viết code cho từng module**

/\* M1: tính toán và kiểm tra tổng tiền của các hàng hóa trong một đơn hàng có chính xác và hợp lệ (tổng tiền không âm) \*/

CREATE or ALTER FUNCTION TinhTongThanhTienDonHang (@MaDH VARCHAR(10))

RETURNS FLOAT

AS

BEGIN

DECLARE @TongThanhTien FLOAT;

-- Tính tổng Thành Tiền = Đơn Giá Xuất \* Số Lượng cho tất cả các mã hàng trong đơn hàng

SELECT @TongThanhTien = SUM(HangHoaXuat.DonGiaXuat \* DatHangChiTiet.SoLuong)

FROM DatHangChiTiet JOIN HangHoaXuat ON DatHangChiTiet.MaHang = HangHoaXuat.MaHang

WHERE DatHangChiTiet.MaDH = @MaDH;

RETURN @TongThanhTien;

END;

GO;

SELECT dbo.TinhTongThanhTienDonHang('DH\_1000000') AS N'Tổng tiền'

SELECT \* FROM DatHangChiTiet

go;

-- kiểm tra tổng tiền có hợp lệ hay không(tổng tiền không âm).

create or alter proc TongTienCuaDatHang @MaDH varchar(10), @TongTien numeric(15,0) out

as

begin

select @TongTien=dbo.TinhTongThanhTienDonHang(@MaDH)

if @TongTien<0

begin

print (@TongTien)

print (N'Tổng tiền < 0: Không hợp lệ')

end

else

begin

print (@TongTien)

print (N'Tổng tiền > 0: Hợp lệ')

end

end

go;

declare @TongTien numeric (15,0)

exec TongTienCuaDatHang 'DH\_1000000', @TongTien out

/\* M2: kiểm tra xem một loại hàng hóa đã có trong kho hay chưa, nếu có thì thông báo “Đã có”,

ngược lại thì thêm hàng hóa mới vào kho. Yêu cầu đối với hàng hóa mới: đơn giá phải > 0 (MaHang, TenHang, DVT, DonGiaNhap) \*/p

go;

create or alter trigger kiemtraHangNhap

on HangHoaNhap

after insert

as

begin

declare @MaHang varchar(10), @DonGiaNhap float

select @MaHang = MaHang, @DonGiaNhap = DonGiaNhap

from inserted

if (select count(\*) from HangHoaNhap where MaHang = @MaHang) > 1

begin

print(N'Đã có loại hàng hóa này')

rollback

end

else

begin

if @DonGiaNhap >= 0

begin

print(N'Thêm loại hàng thành công')

end

else

begin

print(N'Đơn giá nhập <0: không hợp lệ')

rollback

end

end

end

go;

insert into HangHoaNhap (MaHang, TenHang, DVT, DonGiaNhap)

values ('H\_1001000', 'Hang Hoa1000', 'Cay', 1000000);

select \* from HangHoaNhap

go;

/\* M3: Tính tổng doanh thu của toàn bộ đơn hàng theo tháng/quý/năm. \*/

--Theo tháng

CREATE or alter PROCEDURE DoanhThuTheoThang

@thang INT,

@nam INT,

@TongTien numeric(15,0) out

AS

BEGIN

SELECT @TongTien = SUM(TongTien)

FROM dathang

WHERE MONTH(NgayLDH) = @thang AND YEAR(NgayLDH) = @nam;

-- Nếu không có dữ liệu, gán tổng tiền là 0

SET @TongTien = ISNULL(@TongTien, 0);

PRINT N'Doanh thu tháng ' + CAST(@thang AS VARCHAR(2)) + N' năm ' + CAST(@nam AS VARCHAR(4)) + ': ' + CAST(@TongTien AS VARCHAR(30));

END;

GO;

declare @tongtien numeric(15,0)

exec doanhthutheothang 10, 2024, @tongtien out

go;

--Theo quý

CREATE or alter PROCEDURE DoanhThuTheoQuyProc

@quy INT,

@nam INT,

@TongTien NUMERIC(15,0) out

AS

BEGIN

IF @quy = 1

BEGIN

SELECT @TongTien = SUM(TongTien)

FROM dathang

WHERE YEAR(NgayLDH) = @nam

AND MONTH(NgayLDH) IN (1, 2, 3);

END

ELSE IF @quy = 2

BEGIN

SELECT @TongTien = SUM(TongTien)

FROM dathang

WHERE YEAR(NgayLDH) = @nam

AND MONTH(NgayLDH) IN (4, 5, 6);

END

ELSE IF @quy = 3

BEGIN

SELECT @TongTien = SUM(TongTien)

FROM dathang

WHERE YEAR(NgayLDH) = @nam

AND MONTH(NgayLDH) IN (7, 8, 9);

END

ELSE IF @quy = 4

BEGIN

SELECT @TongTien = SUM(TongTien)

FROM dathang

WHERE YEAR(NgayLDH) = @nam

AND MONTH(NgayLDH) IN (10, 11, 12);

END

SET @TongTien = ISNULL(@TongTien, 0);

PRINT N'Doanh thu quý ' + CAST(@quy AS VARCHAR(1)) + N' năm ' + CAST(@nam AS VARCHAR(4)) + ': ' + CAST(@TongTien AS VARCHAR(30));

END;

GO;

declare @tongtien numeric(15,0)

exec doanhthutheoquyproc 4, 2024, @tongtien out

go;

--Doanh thu theo năm

create or alter proc DoanhThuTheoNam (@nam int, @TongTien numeric (15, 0) out)

as

begin

select @TongTien = sum(TongTien)

from dathang

where year(NgayLDH) = @nam

PRINT N'Doanh thu năm ' + CAST(@nam AS VARCHAR(4)) + ': ' + CAST(@TongTien AS VARCHAR(30))

end

go;

declare @TongTien numeric(15,0)

exec doanhthutheonam 2024, @tongtien out;

go;

/\* M5: Kiểm tra số lượng hàng hóa của mỗi loại hàng hóa còn trong kho tại thời điểm kiểm tra.\*/

create or alter proc KiemTraTonKho @MaKho varchar(10), @MaHang varchar (10), @TonKho int out

as

begin

select @TonKho= (sum(NhapKhoChiTiet.ThucNhap)-sum(XuatKhoChiTiet.ThucXuat))

from NhapKhoChiTiet join NhapKho on NhapKhoChiTiet.MaPNK=NhapKho.MaPNK

join Kho on NhapKho.MaKho=Kho.MaKho

join XuatKho on Kho.MaKho=XuatKho.MaKho

join XuatKhoChiTiet on XuatKho.SoPXK=XuatKhoChiTiet.SoPXK

where NhapKhoChiTiet.MaHang=@MaHang

and XuatKhoChiTiet.MaHang=@MaHang

and NhapKho.MaKho=@MaKho

and XuatKho.MaKho=@MaKho

end

go;

declare @MaKho varchar(10), @MaHang varchar (10), @TonKho int

exec KiemTraTonKho 'K1001', 'H\_1000002', @TonKho out

print @TonKho

select \* from NhapKho

select \* from NhapKhoChiTiet

select \* from XuatKho

select \* from XuatKhoChiTiet

go;

--M6: Tính tổng chi phí hàng hóa nhập theo tháng/quý/năm.

--Tính chi phí theo tháng

create or alter procedure TongChiPhiNhapTheoThang

@Thang int,

@Nam int,

@TongChiPhi float out

as

begin

-- tính tổng chi phí nhập theo tháng và năm

select @TongChiPhi = sum(nk.TongTien)

from nhapkho nk

where month(nk.NgayNhap) = @Thang and year(nk.NgayNhap) = @Nam;

-- trả về kết quả và thông báo

print(N'Tổng chi phí hàng hóa nhập theo tháng ' + cast(@Thang as varchar(2)) + N' năm ' + cast(@Nam as varchar(4)) + N' là: ' + cast(isnull(@TongChiPhi, 0) as varchar(30)));

end;

go;

declare @TongChiPhi float

exec TongChiPhiNhapTheoThang 10, 2024, @TongChiPhi out

go;

--Tính chi phí theo quý/ năm

CREATE or alter PROCEDURE TongChiPhiNhapTheoQuy

@quy INT,

@nam INT,

@TongChiPhi FLOAT out

AS

BEGIN

IF @quy = 1

BEGIN

SELECT @TongChiPhi = SUM(TongTien)

FROM NhapKho

WHERE YEAR(NgayNhap) = @nam

AND MONTH(NgayNhap) IN (1, 2, 3);

END

ELSE IF @quy = 2

BEGIN

SELECT @TongChiPhi = SUM(TongTien)

FROM NhapKho

WHERE YEAR(NgayNhap) = @nam

AND MONTH(NgayNhap) IN (4, 5, 6);

END

ELSE IF @quy = 3

BEGIN

SELECT @TongChiPhi = SUM(TongTien)

FROM NhapKho

WHERE YEAR(NgayNhap) = @nam

AND MONTH(NgayNhap) in (7, 8, 9);

END

ELSE IF @quy = 4

BEGIN

SELECT @TongChiPhi = SUM(TongTien)

FROM NhapKho

WHERE YEAR(NgayNhap) = @nam

AND MONTH(NgayNhap) in (10, 11, 12);

END

SET @TongChiPhi = ISNULL(@TongChiPhi, 0);

PRINT N'Chi phí nhập quý: ' + CAST(@quy AS VARCHAR(1)) + N' năm ' + CAST(@nam AS VARCHAR(4)) + ': ' + CAST(@TongChiPhi AS VARCHAR(30));

END;

GO;

declare @TongChiPhi float

exec TongChiPhiNhapTheoQuy 4, 2024, @TongChiPhi out

go;

--Tính chi phí theo năm

create or alter procedure TongChiPhiNhapTheoNam

@Nam int,

@TongChiPhi float out

as

begin

-- tính tổng chi phí nhập theo năm

select @TongChiPhi = sum(nk.TongTien)

from nhapkho nk

where year(nk.NgayNhap) = @Nam;

-- trả về kết quả và thông báo

print(N'Tổng chi phí hàng hóa nhập năm ' + cast(@Nam as varchar(4)) + N' là: ' + cast(isnull(@TongChiPhi, 0) as varchar(30)));

end;

go;

declare @TongChiPhi float

exec TongChiPhiNhapTheoNam 2024, @TongChiPhi out

go;

/\* M7: Module thêm khách hàng, nếu trong csdl Mã khách hàng đã tồn tại thì thông báo “Đã có”,

ngược lại thì thêm thông tin của khách hàng mới (Mã khách hàng, tên khách hàng, số điện thoại, địa chỉ) \*/

create or alter trigger kiemtraThemMoiKH

on KhachHang

after insert

as

begin

declare @MaKH varchar(10), @SDTKH varchar(11)

select @MaKH= MaKH

from inserted

select @SDTKH= SDTKH

from inserted

if (select count(\*) from KhachHang where MaKH = @MaKH and SDTKH = @SDTKH) > 1

begin

print(N'Khách hàng đã tồn tại')

rollback

end

else

begin

print(N'Thêm Khách hàng thành công')

end

end

go;

insert into KhachHang(MaKH, TenKH, SDTKH, DiaChiKH)

values ('1001002', 'Nguyen Van A', '0905755990', 'Da Nang')

select \* from KhachHang

go;

/\* M8: module thêm thông tin của một nhân viên mới, nếu trong csdl đã có mã nhân viên thì thông báo “Đã có”,

ngược lại thì thêm thông tin của nhân viên mới (Mã nhân viên, tên nhân viên, vị trí) \*/

create or alter trigger capnhatTTNV

on NhanVien

after insert

as

begin

declare @MaNV varchar(10)

select @MaNV=MaNV from inserted

if (select count(\*) from NhanVien where MaNV=@MaNV)>1

begin

print(N'Đã có')

rollback

end

else

begin

print(N'Thêm nhân viên thành công')

end

end

go;

insert into NhanVien (MaNV, TenNV, ViTri) values ('NV\_1001002', 'Nguyen Thi B', 'NV Sales')

select \* from NhanVien

go;

/\* M9: Module thêm nhà cung cấp, nếu trong csdl Mã nhà cung cấp đã tồn tại thì thông báo “Đã có”,

ngược lại thì thêm thông tin của nhà cung cấp (Mã nhà cung cấp, tên nhà cung cấp, tên người giao hàng, mã thuế, địa chỉ, stk) \*/

create or alter trigger kiemtraThemMoiNCC

on NhaCungCap

after insert

as

begin

declare @MaNCC varchar(10)

select @MaNCC= MaNCC

from inserted

if (select count(\*) from NhaCungCap where MaNCC=@MaNCC)>1

begin

print(N'Nhà cung cấp đã tồn tại')

rollback

end

else

begin

print(N'Thêm nhà cung cấp thành công')

end

end

go;

insert into NhaCungCap (MaNCC, TenNCC, TenNGH, MaThue, DiaChi, STK)

values ('MC\_3009', 'Cong Ty Quy Thien', 'Tran Van C', 'MT\_4000', 'Da Nang', '72057225')

select \* from NhaCungCap

go;

-- M10: Module tìm tháng có doanh thu cao nhất và thấp nhất trong năm yêu cầu.

CREATE or alter proc M10(

@nyc INT,

@tcn VARCHAR(10) OUTPUT, -- Tên tháng có doanh thu cao nhất

@dtcn FLOAT OUTPUT, -- Doanh thu cao nhất

@ttn VARCHAR(10) OUTPUT, -- Tên tháng có doanh thu thấp nhất

@dttn FLOAT OUTPUT ) -- Doanh thu thấp nhất

AS

BEGIN

-- Tìm tháng có doanh thu cao nhất

SELECT TOP 1

@tcn = MONTH(NgayLDH),

@dtcn = SUM(TongTien)

FROM DatHang

WHERE YEAR(NgayLDH) = @nyc -- So sánh năm từ biến đầu vào

GROUP BY MONTH(NgayLDH)

ORDER BY sum(TongTien) DESC;

-- Tìm tháng có doanh thu thấp nhất

SELECT TOP 1

@ttn =MONTH(NgayLDH),

@dttn = SUM(TongTien)

FROM DatHang

WHERE YEAR(NgayLDH) = @nyc -- So sánh năm từ biến đầu vào

GROUP BY MONTH(NgayLDH)

ORDER BY sum(TongTien) ASC;

END;

GO;

declare @nyc INT, @dtcn FLOAT, @dttn FLOAT, @tcn VARCHAR(10), @ttn VARCHAR(10);

exec dbo.M10 2024, @tcn OUTPUT, @dtcn OUTPUT, @ttn OUTPUT, @dttn OUTPUT;

print N'Tháng cao nhất: ' + @tcn + ' | Doanh thu cao nhất: ' + cast(@dtcn AS VARCHAR(50));

print N'Tháng thấp nhất: ' + @ttn + ' | Doanh thu thấp nhất: ' + cast(@dttn AS VARCHAR(50));

GO;

--M11. Module tìm ra top 5 khách hàng đặt nhiều đơn hàng nhất trong theo năm yêu cầu.

Create or alter function M11(@nyc int)

returns table

AS

return SELECT TOP 5 KhachHang.TenKH,COUNT(\*) AS SoDonHang

FROM DatHang JOIN KhachHang ON DatHang.MaKH = KhachHang.MaKH

WHERE YEAR(NgayLDH) = @nyc

GROUP BY KhachHang.TenKH

ORDER BY SoDonHang DESC;

GO;

select \*from dbo.M11(2024)

GO;

-- M12: Module tìm ra top 10 sản phẩm được bán nhiều nhất trong tháng và năm yêu cầu với điều kiện giá của sản phẩm trên 50.000 đồng.

Create or alter function M12(@namyc int, @tyc int)

returns @trave table (TenHang varchar(50),

ThucXuat int)

as

Begin

Insert into @trave

Select Top 10 TenHang, Sum(ThucXuat)

from XuatKhoChiTiet JOIN XuatKho ON XuatKho.SoPXK = XuatKhoChiTiet.SoPXK

JOIN DatHangChiTiet ON DatHangChiTiet.MaHang = XuatKhoChiTiet.MaHang

JOIN HangHoaXuat ON HangHoaXuat.MaHang = XuatKhoChiTiet.MaHang

WHERE MONTH(NgayXuat) = @tyc -- Tìm trong tháng hiện tại

AND YEAR(NgayXuat) = @namyc -- Tìm trong năm hiện tại

AND DonGiaXuat > 50000 -- Lớn hơn 50.000 đồng.

GROUP BY HangHoaXuat.TenHang -- Nhóm theo tên sản phẩm

ORDER BY Sum(ThucXuat) DESC; -- Sắp xếp theo số lượng xuất giảm dần

return

end

go;

select \* from dbo.M12(2024, 10)

go;

--M13.Module tìm ra top 3 nhân viên bán được nhiều đơn hàng nhất trong năm yêu cầu.

Create function M13(@nyc int)

returns @trave table (TenNV varchar(50),

Sl int)

As

Begin

Insert into @trave

Select Top 3 TenNV, Count(MaDH) as SLDH

from NhanVien join DatHang on DatHang.MaNV = NhanVien.MaNV

where year(NgayLDH) = @nyc

Group by TenNV

Order by Count(DatHang.MaDH) DESC

return

end

go;

select \* from dbo.M13(2024)

go;

**R8: Thực hiện tạo bảng User và thiết lập kiểm tra đăng nhập với các phương pháp xác thực đầu vào dữ liệu, tham số hóa truy vấn sql và tạo thủ tục.**

**--Bước 1 : Tạo bảng lưu Users**

Create table Users(

id int IDENTITY(1,1) primary key,

username varchar(200) COLLATE SQL\_Latin1\_General\_CP1\_CS\_AS,

passwords varchar(200) COLLATE SQL\_Latin1\_General\_CP1\_CS\_AS

)

--Tạo thủ tục insert dữ liệu

Create proc InsertUsers(    @i\_username VARCHAR(200),

@i\_password VARCHAR(200))

As

Begin

declare @ktra int

set @ktra = (Select Count(\*) from Users where username = @i\_username)

if @ktra > 0

begin

print 'Chen that bai'

return

end

Insert into Users(username,passwords)

values (@i\_username,@i\_password)

print 'Da chen'

end

--

Exec InsertUsers 'Banhang','banhang'

select \* from Users

**--Bước 2: Tạo thủ tục, xác thực dữ liệu đầu vào và tham số hóa truy vấn dữ liệu.**

Create proc CheckLogins

    @i\_username varchar(200),

    @i\_password varchar(200),

@tbao varchar(200) output

As

Begin

    -- 1. Kiểm tra tên đăng nhập và mật khẩu không để trống.

    If @i\_username IS NULL OR @i\_username = ''

    Begin

        Set @tbao = N'Tên đăng nhập hoặc mật khẩu sai';

        return;

    end

    If @i\_password IS NULL OR @i\_password = ''

    Begin

        set @tbao = N'Tên đăng nhập hoặc mật khẩu sai';

        return;

    end

    -- 2. Kiểm tra tên đăng nhập chỉ chứa chữ cái và số

    If PATINDEX('%[^a-zA-Z0-9\_]%', @i\_username) > 0

    Begin

        set @tbao = N'Tên đăng nhập hoặc mật khẩu sai.';

        return;

    end

    -- 3. Kiểm tra tên đăng nhập và mật khẩu

    Declare @ktra int;

    Select @ktra = COUNT(\*)

    from Users

    where username = @i\_username AND passwords = @i\_password; -- Kiểm tra mật khẩu

    If @ktra < 1

    begin

        SET @tbao = N'Tên đăng nhập hoặc mật khẩu sai.';

        return;

    end

    -- Đăng nhập thành công

    set @tbao = N'Đăng nhập thành công.';

end

-- Gọi thủ tục lưu trữ

Declare @tbao varchar(200)

Exec CheckLogins 'banhang', 'banhang',@tbao =@tbao output

Print @tbao