-- Nguyễn Văn Tú

-- 1) Dựa vào cột ngày vào làm, tính thâm niên cho nhân viên để tăng lương hoặc bổ nhiệm.

SELECT MaGiangVien, TenGiangVien, MONTH(NgayVaoLam) AS [Tháng vào làm], YEAR(NgayVaoLam) AS [Năm vào làm],

DATEDIFF(MONTH, NgayVaoLam, GETDATE()) / 12 AS [Thâm Niên Năm] --tham số thứ nhất là đơn vị ta muốn tính chênh lệch, than số thứ 2 là ngày bắt đầu, thứ 3 là ngày kết thúc

FROM GiangVien;



--Câu 2: Liệt kê tất cả các giảng viên ưu tú trong năm 2023.

--Biết rằng giảng viên ưu tú được đánh giá theo mỗi năm và

--là giảng viên mà có nhiều hơn 3 lần khen thưởng trong năm đó.

SELECT KT.MaGiangVien,

GV.TenGiangVien,

YEAR(KT.NgayKhenThuong) AS [Năm xét thưởng],

COUNT(KT.MaKhenThuong) AS [Số lần khen thưởng]

FROM KhenThuong KT

JOIN GiangVien GV ON KT.MaGiangVien = GV.MaGiangVien

WHERE YEAR(KT.NgayKhenThuong) = 2024

GROUP BY KT.MaGiangVien, GV.TenGiangVien, YEAR(KT.NgayKhenThuong)

HAVING COUNT(KT.MaKhenThuong) >= 3;



--3) Tìm ra nhân viên có lương cao nhất theo tháng (tháng 2 năm 2023)

WITH BangTam AS (

SELECT gv.MaGiangVien, gv.TenGiangVien, k.TenKhoa, cv.TenChucVu, bl.HeSoLuong, pc.TienPhuCap, bc.ThangCong,

bc.SoNgayLam AS TongNgayLam,

bc.SoNgayLam \* (bl.HeSoLuong \* 100000) + pc.TienPhuCap AS LuongThucTe

FROM GiangVien gv

JOIN Khoa k ON gv.MaKhoa = k.MaKhoa

JOIN ChucVu cv ON gv.MaChucVu = cv.MaChucVu

JOIN BangBacLuong bl ON gv.MaChucVu = bl.MaChucVu

JOIN PhuCap pc ON gv.MaChucVu = pc.MaChucVu

JOIN BangChamCong bc ON gv.MaGiangVien = bc.MaGiangVien

WHERE bc.ThangCong LIKE '2/2023')

SELECT MAX(LuongThucTe) [Lương cao nhất]

FROM BangTam;



------View------

--1) Tính tuổi của từng giảng viên(Để xem có ai đến tuổi về hưu)

CREATE VIEW [Tuổi Của Từng Giảng Viên] AS

SELECT MaGiangVien, TenGiangVien, YEAR(GETDATE())-YEAR(NgaySinh) AS [Tuổi]

FROM GiangVien

GROUP BY MaGiangVien, TenGiangVien, YEAR(GETDATE())-YEAR(NgaySinh);

SELECT \* FROM [dbo].[Tuổi Của Từng Giảng Viên];



--2) Liệt kê các giảng viên theo từng khoa

CREATE VIEW [Số Lượng Giảng Viên Theo Từng Khoa] AS

SELECT KH.MaKhoa,

KH.TenKhoa,

COUNT(KH.MaKhoa) AS [Số lượng giảng viên trong từng khoa]

FROM Khoa AS KH

JOIN GiangVien AS GV ON KH.MaKhoa = GV.MaKhoa

GROUP BY KH.MaKhoa,

KH.TenKhoa;

SELECT \* FROM [dbo].[Số Lượng Giảng Viên Theo Từng Khoa];



-- 3) Tính thuế thu nhập cho từng giảng viên theo năm

CREATE VIEW TinhThueThuNhapGiangVien AS

WITH BangTam AS (

SELECT gv.MaGiangVien,

gv.TenGiangVien,

k.TenKhoa,

cv.TenChucVu,

bl.HeSoLuong,

pc.TienPhuCap,

YEAR(bc.ThangCong) AS [Năm Tính Thuế],

bc.SoNgayLam AS TongNgayLam,

bc.SoNgayLam \* (bl.HeSoLuong \* 100000) + pc.TienPhuCap AS LuongThucTe

FROM GiangVien gv

JOIN Khoa k ON gv.MaKhoa = k.MaKhoa

JOIN ChucVu cv ON gv.MaChucVu = cv.MaChucVu

JOIN BangBacLuong bl ON gv.MaChucVu = bl.MaChucVu

JOIN PhuCap pc ON gv.MaChucVu = pc.MaChucVu

JOIN BangChamCong bc ON gv.MaGiangVien = bc.MaGiangVien

WHERE YEAR(bc.ThangCong)=2023)

SELECT MaGiangVien,

TenGiangVien,

TenKhoa,

TenChucVu,

HeSoLuong,

TienPhuCap,

[Năm Tính Thuế],

TongNgayLam,

LuongThucTe,

CASE

WHEN LuongThucTe BETWEEN 0 AND 3600000 THEN LuongThucTe \* 0.24

WHEN LuongThucTe BETWEEN 3600000 AND 4500000 THEN LuongThucTe \* 0.28

WHEN LuongThucTe BETWEEN 4500000 AND 7500000 THEN LuongThucTe \* 0.30

WHEN LuongThucTe BETWEEN 7500000 AND 15000000 THEN LuongThucTe \* 0.32

ELSE LuongThucTe \* 0.40

END AS Thue

FROM BangTam;

SELECT \* FROM TinhThueThuNhapGiangVien;



------Procedure------

-- Câu 1) Tạo 1 Stored Procedure để xem dữ liệu trong bảng GiANGVIEN

CREATE PROCEDURE xemDuLieuGiangVien

AS

BEGIN

SELECT \* FROM GiangVien

END

-- Sử dụng Stored Procedures

EXECUTE xemDuLieuGiangVien



Hiển thị dữ liệu của bảng GiangVien thành công

-- Câu 2) Tạo stored procedures để thêm mới giảng viên vào bảng GiangVien

CREATE PROCEDURE themGiangVien

@teacherID NVARCHAR(10),

@teacherName NVARCHAR(255),

@dayOfBirth DATE,

@gender NVARCHAR(50),

@address NVARCHAR(255),

@phoneNumber NVARCHAR(20),

@falcutyID NVARCHAR(255),

@positionID NVARCHAR(255),

@ngayVaoLam DATE

AS

BEGIN

INSERT INTO GiangVien(MaGiangVien, TenGiangVien, NgaySinh, GioiTinh, DiaChi, SoDienThoai, MaKhoa, MaChucVu, NgayVaoLam)

VALUES (@teacherID, @teacherName, @dayOfBirth, @gender, @address, @phoneNumber, @falcutyID, @positionID, @ngayVaoLam)

END;

EXECUTE themGiangVien @teacherID='GV012', @teacherName='Leonardo Nguyen', @dayOfBirth='2003-03-05', @gender='Male', @address='Ha Noi City', @phoneNumber='0999999999', @falcutyID='K001', @positionID='CV001', @ngayVaoLam='2022-05-01';



Thêm mới giảng viên có mã GV012 thành công

-- Câu 3) Tạo Stored Procedure để sửa dữ liệu trên bảng GIANGVIEN

CREATE PROCEDURE updateGiangVien

@teacherID NVARCHAR(10),

@teacherName NVARCHAR(255),

@dayOfBirth DATE,

@gender NVARCHAR(50),

@address NVARCHAR(255),

@phoneNumber NVARCHAR(20),

@falcutyID NVARCHAR(255),

@positionID NVARCHAR(255),

@ngayVaoLam DATE

AS

BEGIN

UPDATE GiangVien

SET

TenGiangVien = @teacherName,

NgaySinh = @dayOfBirth,

DiaChi = @address,

SoDienThoai = @phoneNumber,

MaKhoa = @falcutyID,

MaChucVu = @positionID,

NgayVaoLam = @ngayVaoLam

WHERE MaGiangVien = @teacherID

END;

--Test update dữ liệu

EXECUTE updateGiangVien @teacherID='GV012', @teacherName='Leonardo Nguyen Van', @dayOfBirth='2003-03-01', @gender='Male', @address='Viet Tri City', @phoneNumber='0968146590', @falcutyID='K002', @positionID='CV001', @ngayVaoLam='2022-01-01';

EXECUTE xemDuLieuGiangVien;



Giảng viên có mã GV012 đã được sửa lại tên và ngày vào làm

-- Câu 4) Xóa dữ liệu trong bảng GiangVien (xóa theo mã giảng viên)

CREATE PROCEDURE deleteByTeacherID

@teacherID NVARCHAR(10)

AS

BEGIN

DELETE FROM GiangVien

WHERE MaGiangVien=@teacherID;

END;

-- Execute

EXECUTE deleteByTeacherID @teacherID='GV012';

EXECUTE xemDuLieuGiangVien;



Giảng viên có mã GV012 đã được xóa khỏi cơ sở dữ liệu

-- Câu 5) Tìm kiếm giảng viên theo tên (Search by name)

CREATE PROCEDURE searchByTeacherName

@teacherName NVARCHAR(255)

AS

BEGIN

SELECT \*

FROM GiangVien

WHERE TenGiangVien LIKE '%' + @teacherName + '%'

END;

--Execute

EXECUTE searchByTeacherName @teacherName='anh';

-- nếu muốn tìm kiếm bằng tiếng việt thì đặt trong ngoặc vuông.



Lọc ra các giảng viên trong tên có chứa chữ “Anh” thành công

-- Câu 6: So sánh lương của một người bất kỳ với lương trung bình

CREATE PROCEDURE soSanhLuongTrungBinh

@MaNhanVien NVARCHAR(10) -- Tham số đầu vào để nhận mã nhân viên cần so sánh

AS

BEGIN

DECLARE @avgLuong DECIMAL(10,2)

-- Tính lương trung bình của tất cả nhân viên

SELECT @avgLuong = AVG(CAST(bc.SoNgayLam \* (bl.HeSoLuong \* 100000) + pc.TienPhuCap AS DECIMAL(10,2)))

FROM GiangVien gv

JOIN BangChamCong bc ON gv.MaGiangVien = bc.MaGiangVien

JOIN BangBacLuong bl ON gv.MaChucVu = bl.MaChucVu

JOIN PhuCap pc ON gv.MaChucVu = pc.MaChucVu

PRINT N'Lương trung bình: ' + CAST(@avgLuong AS NVARCHAR(20))

-- Tính lương của nhân viên cụ thể

DECLARE @luongNhanVien DECIMAL(10,2)

SELECT @luongNhanVien = CAST(bc.SoNgayLam \* (bl.HeSoLuong \* 100000) + pc.TienPhuCap AS DECIMAL(10,2))

FROM GiangVien gv

JOIN BangChamCong bc ON gv.MaGiangVien = bc.MaGiangVien

JOIN BangBacLuong bl ON gv.MaChucVu = bl.MaChucVu

JOIN PhuCap pc ON gv.MaChucVu = pc.MaChucVu

WHERE gv.MaGiangVien = @MaNhanVien

PRINT N'Lương của nhân viên: ' + CAST(@luongNhanVien AS NVARCHAR(20))

-- So sánh lương của nhân viên với lương trung bình

IF @luongNhanVien > @avgLuong

PRINT N'Lương của nhân viên này cao hơn lương trung bình.'

ELSE IF @luongNhanVien = @avgLuong

PRINT N'Lương của nhân viên này bằng lương trung bình.'

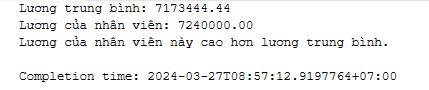
ELSE

PRINT N'Lương của nhân viên này thấp hơn lương trung bình.'

END

EXECUTE soSanhLuongTrungBinh @MaNhanVien='GV003';

DROP PROCEDURE soSanhLuongTrungBinh;



So sánh lương của giảng viên bất kỳ với mức lương trung bình thành công

------Trigger------

/\*Câu 1: Yêu cầu chấm công phải lớn hơn hoặc bằng 0 và nhỏ hơn hoặc bằng 31\*/

IF EXISTS (SELECT NAME FROM sys.triggers WHERE NAME = 'tg\_KiemTraDuLieuTruocKhiThem')

DROP TRIGGER tg\_KiemTraDuLieuTruocKhiThem;

GO

CREATE TRIGGER tg\_KiemTraDuLieuTruocKhiThem

ON BangChamCong

AFTER INSERT, UPDATE

AS

BEGIN

DECLARE @SoNgayLam INT;

-- Kiểm tra từng dòng dữ liệu trong inserted

SELECT @SoNgayLam = SoNgayLam FROM inserted;

-- Nếu giá trị của SoNgayLam không thỏa mãn điều kiện, phát sinh lỗi và rollback giao dịch

IF @SoNgayLam < 0 OR @SoNgayLam > 31

BEGIN

PRINT N'Bạn phải nhập số ngày công lớn hơn hoặc bằng 0 và nhỏ hơn 31';

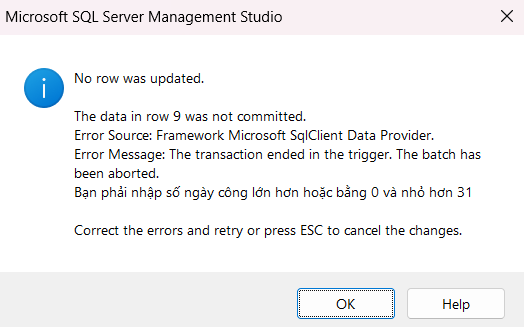
ROLLBACK TRANSACTION; -- Lưu ý: Sử dụng ROLLBACK chỉ khi bạn muốn quay lại trạng thái trước khi thực hiện cập nhật.

RETURN;

END

END

GO



Thông báo lỗi khi ta cố gắng thêm hoặc sửa giá trị ngày công nhỏ hơn 0 hoặc nhỏ hơn 31

/\*Câu 2: Kiểm tra nếu dữ liệu đã có ở bảng con(bảng chứa fk) rồi thì không cho xóa ở bảng cha(bảng có pk)\*/

IF EXISTS (SELECT NAME FROM sys.triggers WHERE NAME = 'KiemTraDuLieuTruocKhiXoa')

DROP TRIGGER KiemTraDuLieuTruocKhiXoa;

GO

CREATE TRIGGER KiemTraDuLieuTruocKhiXoa

ON Khoa INSTEAD OF DELETE

AS

DECLARE @maKhoa NVARCHAR(255), @ErrMsg CHAR(200)

SET @maKhoa = (SELECT COUNT (Khoa.MaKhoa)

FROM deleted, Khoa

WHERE Khoa.MaKhoa = deleted.MaKhoa)

IF @maKhoa > 0

BEGIN

SET @ErrMsg = N'Bạn không thể xóa dữ liệu ở bảng KHOA vì nó đã được nhập ở trong bảng MONHOC(Quan hệ khóa chính - khóa phụ). Nếu bạn muốn xóa dữ liệu ở bảng KHOA thì cần xóa ở MONHOC trước.'

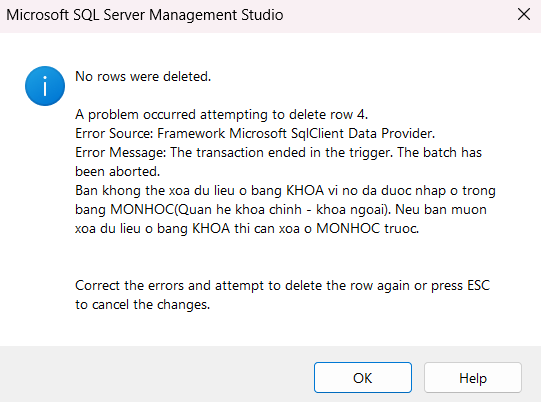
PRINT @ErrMsg

ROLLBACK TRANSACTION

END

ELSE

DELETE Khoa WHERE MaKhoa IN (SELECT MaKhoa FROM deleted);



Chương trình thông báo lỗi khi ta cố gắng xóa một bộ dữ liệu ở bảng chứa khóa chính trong khi dữ liệu này đã tồn tại ở bảng chứa khóa ngoại rồi.