Các hàm, thủ tục, trigger

| STT | Tên | Đặt tên | Mục đích |
| --- | --- | --- | --- |
| 1 | Hàm tính tiền lương cho nhân viên | TienLuong | Tính tiền lương cho nhân viên |
| 2 | Thủ tục tạo thông báo | TaoThongBao | Tạo thông báo khi chủ quán có thông tin mới muốn thông báo đến cho nhân viên trong quán |
| 3 | Thủ tục cập nhật thông tin nhân viên | CapNhatThongTin | Khi thông tin của nhân viên bị sai hoặc có sự thay đổi nào đó như địa chỉ, số điện thoại thì sử dụng hàm để sửa thông tin nhân viên để đảm bảo rằng tất cả thông tin của nhân viên luôn được cập nhật và chính xác, từ đó giúp chủ quán dễ dàng quản lý theo dõi và kiểm soát thông tin. |
| 4 | Thủ tục kiểm tra mã đăng nhập đã tồn tại hay chưa | CheckMaDangNhap | Kiểm tra mã đăng nhập đã tồn tại trong bảng nhân viên hay chưa |
| 5 | Thủ tục tạo mới thông tin nhân viên | TaoThongTinNV | Khi có nhân viên mới vào làm, thủ tục sẽ tạo thông tin cho nhân viên mới |
| 6 | Thủ tục cập nhật Ngày nghỉ trong bảng lịch làm | CapNhatNgayNghi | Cập nhật lại ngày nghỉ cho nhân viên nếu nhân viên xin nghỉ |
| 7 | Thủ tục để sửa đổi thời gian chấm công | CapNhatThoiGianChamCong | Trong trường hợp có sai sót hoặc lỗi hệ thống dẫn đến việc chấm công không chính xác, thủ tục này cho phép quản lý dễ dàng sửa đổi thời gian chấm công để đảm bảo tính chính xác. |
| 8 | Trigger cập nhật số giờ làm trong tháng | TinhSoGioLam | Tự động cập số giờ làm trong tháng sau khi chấm công (thêm 1 bản ghi mới trong bảng chấm công) |
| 9 | Trigger đảm bảo nhân viên không đăng ký hai ca làm việc trùng giờ trong cùng một ngày. | KiemTraTrungCaLamViec | Trigger này sẽ kiểm tra xem nhân viên đã đăng ký ca làm việc trùng giờ trong cùng một ngày chưa, nếu đã tồn tại thì sẽ phát sinh lỗi và hủy giao dịch. Nếu không có lỗi, trigger sẽ thực hiện chèn dữ liệu bình thường. |
| 10 | Trigger khi thêm mới nhân viên, đảm bảo rằng mã đăng nhập là mới và phải đủ 12 ký tự, số điện thoại phải hợp lệ | KiemTraThemNhanVien | Trigger này sẽ kiểm tra xem mã đăng nhập của nhân viên có lã mã mới hay không và có đủ 12 ký tự hay không, số điện thoại có hợp lệ (số điện thoại có 10 số) )hay không trước khi thêm bản ghi mới vào bảng nhân viên. |
| 11 | Thủ tục để xóa thông báo khi thông tin không còn phù hợp. | XoaThonTriggergBao | Thủ tục để cập xóa thông báo khi thông báo không còn phù hợp với tình hình thực tế để đảm bảo rằng nhân viên luôn nhận được thông tin phù hợp trong các tình hình khác nhau của quán. |
| 12 | Trigger để xóa thông tin bị lỗi thời | XoaThongBaoTuDong | Trigger này kiểm tra dung lượng của bản ghi và xóa những thông báo bị lỗi thời giải phóng dung lượng |
| 13 | kiểm tra tính hợp lệ của lương, đảm bảo mức lương phải nằm trong khoảng lương hợp lý tùy thuộc vào cấp bậc của nhân viên. | KiemTraMucLuong | Trigger này kiểm tra rằng mức lương phải nằm trong khoảng lương hợp lý dựa trên cấp bậc của nhân viên, điều này giúp ngăn chặn việc nhập các giá trị lương không hợp lý, có thể dẫn đến sự không công bằng trong quản lý lương. |

1. TienLuong

CREATE FUNCTION TienLuong (@SoGioLamTrongThang INT, @MaDangNhap CHAR(12))

RETURNS INT

AS

BEGIN

DECLARE @TienLuong INT, @CapBac CHAR(2), @LuongMoiGio INT

SELECT @CapBac = CapBac FROM CapBacNV WHERE MaDangNhap = @MaDangNhap

SET @LuongMoiGio = CASE WHEN @CapBac = 'S1' THEN 15000

WHEN @CapBac = 'S2' THEN 17000

WHEN @CapBac = 'SS' THEN 20000

ELSE 0

END

SET @TienLuong = @SoGioLamTrongThang \* @LuongMoiGio

RETURN @TienLuong

END

1. TaoThongBao:

CREATE PROCEDURE TaoThongBao

@TenThongBao NVARCHAR(100),

@NoiDungThongBao NVARCHAR(200)

AS

BEGIN

DECLARE @MaTB\_moi CHAR(4)

DECLARE @MaTB\_cu char(4)

DECLARE @A INT

SET @MaTB\_cu = (SELECT MAX(MaTB) FROM ThongBao)

IF @MaTB\_cu IS NULL

begin

SET @MaTB\_cu = 000

end

else

begin

SET @A = @MaTB\_cu + 1

SET @MaTB\_moi = concat(replicate('0', 4 - len(@A )), @A)

INSERT INTO ThongBao (MaTB, TenThongBao, ThoiGianThongBao, NoiDungThongBao)

VALUES (@MaTB\_moi, @TenThongBao, GETDATE(), @NoiDungThongBao)

SELECT \* FROM ThongBao WHERE MaTB = @MaTB\_moi

end

END

GO

EXEC TaoThongBao N'Thông báo chương trình khuyến mãi', N'Tặng khách hàng 1 con labubu khi mua trà sữa'

drop procedure TaoThongbao

select \* from ThongBao

1. CapNhatThongTin( dùng thủ tục)

CREATE or alter PROCEDURE CapNhatThongTinNV

@MaDangNhap CHAR(12),

@DiaChi NVARCHAR(60) = NULL,

@SDT CHAR(10) = NULL,

@TuoiNV INT = NULL,

@SoTaiKhoanNH VARCHAR(20) = NULL

AS

BEGIN

-- Kiểm tra nếu nhân viên có tồn tại trong bảng

IF EXISTS (SELECT 1 FROM NhanVien WHERE MaDangNhap = @MaDangNhap)

BEGIN

-- Cập nhật thông tin nhân viên chỉ khi tham số không phải NULL

IF @DiaChi IS NOT NULL

BEGIN

UPDATE NhanVien SET DiaChi = @DiaChi WHERE MaDangNhap = @MaDangNhap;

END

IF @SDT IS NOT NULL

BEGIN

UPDATE NhanVien SET SDT = @SDT WHERE MaDangNhap = @MaDangNhap;

END

IF @TuoiNV IS NOT NULL

BEGIN

UPDATE NhanVien SET TuoiNV = @TuoiNV WHERE MaDangNhap = @MaDangNhap;

END

IF @SoTaiKhoanNH IS NOT NULL

BEGIN

UPDATE NhanVien SET SoTaiKhoanNH = @SoTaiKhoanNH WHERE MaDangNhap = @MaDangNhap;

END

PRINT N'Cập nhật thông tin nhân viên thành công với mã đăng nhập: ' + @MaDangNhap;

END

ELSE

BEGIN

PRINT N'Nhân viên không tồn tại với mã đăng nhập: ' + @MaDangNhap;

END

END

EXEC CapNhatThongTinNV

@MaDangNhap = 'NV00000’1000',

@DiaChi = N' Văn Đồng',

@SDT = '0304820673';

select\*from NhanVien

1. CheckMaDangNhap
2. TaoThongTinNV

CREATE or alter PROCEDURE sp\_ThemNhanVienmoi

@MaDangNhap CHAR(12),

@CCCD CHAR(12),

@TenNV NVARCHAR(50),

@TuoiNV INT,

@DiaChi NVARCHAR(60),

@SDT CHAR(10),

@SoTaiKhoanNH VARCHAR(20),

@MatKhau VARCHAR(20)

AS

BEGIN

-- Kiểm tra xem mã đăng nhập đã tồn tại chưa

IF EXISTS (SELECT @Madangnhap FROM NhanVien WHERE @MaDangNhap=MaDangNhap)

BEGIN

PRINT 'Tài khoản đã tồn tại';

END

ELSE

BEGIN

-- Thêm mới nhân viên

INSERT INTO NhanVien (MaDangNhap, CCCD, TenNV, TuoiNV, DiaChi, SDT, SoTaiKhoanNH, MatKhau)

VALUES (@MaDangNhap, @CCCD, @TenNV, @TuoiNV, @DiaChi, @SDT, @SoTaiKhoanNH, @MatKhau);

END

END

1. CapNhatNgayNghi
2. CapNhatThoiGianChamCong

CREATE PROCEDURE CapNhatThoiGianChamCong

@MaDangNhap CHAR(12),

@NgayChamCong DATE,

@GioVaoMoi TIME,

@GioRaMoi TIME

AS

BEGIN

-- Kiểm tra xem thông tin chấm công của nhân viên có tồn tại trong ngày cần cập nhật hay không

IF EXISTS (

SELECT 1

FROM ChamCongNV

WHERE MaDangNhap = @MaDangNhap AND NgayChamCong = @NgayChamCong

)

BEGIN

-- Cập nhật thời gian vào và ra

UPDATE ChamCong

SET TGVaoCa = @GioVaoMoi,

TGRaCa = @GioRaMoi

WHERE MaBangChamCong IN (

SELECT MaBangChamCong

FROM ChamCongNV

WHERE MaDangNhap = @MaDangNhap AND NgayChamCong = @NgayChamCong

);

-- Thông báo cập nhật thành công

PRINT 'Thời gian chấm công đã được cập nhật thành công.';

END

ELSE

BEGIN

-- Nếu không tìm thấy thông tin chấm công, thông báo lỗi

PRINT 'Không tìm thấy thông tin chấm công cho nhân viên vào ngày này.';

END

END

1. TinhSoGioLam

create trigger TinhSoGioLam

on ChamCong

after insert

as

begin

declare @MaBangChamCong char(4), @MaBangLuong char(4), @SoGioLamTrongThang int

select @MaBangChamCong = inserted.MaBangChamCong from inserted

select top 1 @MaBangLuong = Luong.MaBangLuong

from Luong join LuongNV on Luong.MaBangLuong = LuongNV.MaBangLuong

join ChamCongNV on LuongNV.MaDangNhap = ChamCongNV.MaDangNhap

where ChamCongNV.MaBangChamCong = @MaBangChamCong

select @SoGioLamTrongThang = sum(datediff(hour, TGVaoCa, TGRaCa))

from ChamCong

join ChamCongNV on ChamCong.MaBangChamCong = ChamCongNV.MaBangChamCong

where ChamCongNV.MaBangChamCong = @MaBangChamCong

and month (ChamCong.NgayChamCong) = MONTH(GETDATE())

and year(ChamCong.NgayChamCong) = YEAR(GETDATE())

update Luong

set SoGioLamTrongThang = @SoGioLamTrongThang

where MaBangLuong = @MaBangLuong

end

1. KiemTraTrungCaLamViec

CREATE TRIGGER KiemTraTrungCaLamViec

ON LichLamNV

FOR INSERT

AS

BEGIN

-- Kiểm tra trùng lặp ca làm việc cho nhân viên trong cùng một ngày

IF EXISTS (

SELECT 1

FROM LichLamNV LLNV

JOIN LichLam LL ON LL.MaBang = LLNV.MaBang

JOIN inserted I ON I.MaDangNhap = LLNV.MaDangNhap

WHERE LL.NgayLam = (SELECT NgayLam FROM LichLam WHERE MaBang = I.MaBang)

AND LLNV.MaDangNhap = I.MaDangNhap

)

BEGIN

-- Nếu có trùng lặp, phát sinh lỗi và hủy giao dịch

RAISERROR ('Nhân viên đã đăng ký ca làm việc trùng giờ trong cùng một ngày.', 16, 1);

ROLLBACK TRANSACTION;

END

ELSE

BEGIN

-- Nếu không có lỗi, chèn dữ liệu vào bảng

INSERT INTO LichLamNV (MaBang, MaDangNhap)

SELECT MaBang, MaDangNhap

FROM inserted;

END

END;

create or alter trigger KiemTraTrungCaLamViec

on LichLamNV

for insert

as

begin

-- Kiểm tra xem có nhân viên nào đăng ký trùng ca làm việc và ngày làm không

if exists (

select 1

from inserted I

join LichLam LL on LL.MaBang = I.MaBang

join LichLamNV LLNV on LLNV.MaDangNhap = I.MaDangNhap

join LichLam LL2 on LL2.MaBang = LLNV.MaBang

where LL.NgayLam = LL2.NgayLam

and LL.CaLam = LL2.CaLam

and LLNV.MaDangNhap = I.MaDangNhap

and LLNV.MaBang <> I.MaBang -- Điều kiện này đảm bảo kiểm tra với các bản ghi trước đó

)

begin

-- Nếu nhân viên đã đăng ký trùng ca làm trong ngày đó, rollback giao dịch

print N'Nhân viên đã đăng ký ca làm việc trùng giờ trong cùng một ngày'

rollback

end

end

1. KiemTraThemNhanVien

CREATE or alter TRIGGER KiemTraThemNhanVien

ON NhanVien

AFTER INSERT

AS

BEGIN

DECLARE @MaDangNhap nvarchar(12), @SoDienThoai nvarchar(10)

SELECT @MaDangNhap = inserted.MaDangNhap, @SoDienThoai = inserted.SDT

FROM inserted

IF LEN(@MaDangNhap) < > 12

BEGIN

print 'Mã đăng nhập phải có đúng 12 ký tự'

ROLLBACK

END

ELSE IF LEN(@SoDienThoai) < > 10

BEGIN

print 'Số điện thoại phải có đúng 10 chữ số'

ROLLBACK

END

END

1. XoaThongBao

CREATE PROCEDURE XoaThongBao

@MaTB CHAR(4)

AS

BEGIN

DELETE FROM ThongBao

WHERE MaTB = @MaTB;

IF @@ROWCOUNT = 0

BEGIN

PRINT 'Error!';

END

ELSE

BEGIN

PRINT 'Done!';

END

END

GO

EXEC XoaThongBao '1002'

1. XoaThongBaoTuDong

CREATE TRIGGER XoaThongBaoTuDong

ON ThongBao

AFTER INSERT

AS

BEGIN

DECLARE @Count INT;

SELECT @Count = COUNT(\*) FROM ThongBao;

IF @Count > 2000

BEGIN

DELETE FROM ThongBao

WHERE ThoiGianThongBao < DATEADD(YEAR, -1, GETDATE());

END

END

GO

create trigger XoaThongBaoTuDong

on ThongBao

after insert

as

begin

delete from ThongBao

where ThoiGianThongBao < dateadd(year, -1, getdate())

end

1. KiemTraMucLuong

CREATE TRIGGER KiemTraMucLuong

ON Luong

after INSERT, UPDATE

AS

BEGIN

DECLARE @MaBangLuong CHAR(5), @MaDangNhap CHAR(12), @CapBac CHAR(2),

@TienLuong INT, @LuongMoiGio INT

SELECT @MaBangLuong = MaBangLuong, @TienLuong = TienLuong FROM inserted

SELECT @MaDangNhap = MaDangNhap FROM LuongNV WHERE MaBangLuong = @MaBangLuong

SELECT @CapBac = CapBac FROM CapBacNV WHERE MaDangNhap = @MaDangNhap

SET @LuongMoiGio = CASE

WHEN @CapBac = 'S1' THEN 15000

WHEN @CapBac = 'S2' THEN 17000

WHEN @CapBac = 'SS' THEN 20000

ELSE 0

END

-- Kiểm tra xem lương có nằm trong khoảng hợp lý không

-- Ít nhất một tháng làm 12 ca và nhiều nhát làm 20 ca (1 ca lam co 5 tieng)

IF @TienLuong < (@LuongMoiGio \* 60) OR @TienLuong > (@LuongMoiGio \* 100 )

BEGIN

print N'Lương không hợp lệ! Vui lòng kiểm tra lại.'

ROLLBACK

END

END

INSERT INTO Luong (MaBangLuong, SoGioLamTrongThang, SoGioBatDauLam, TienLuong)

VALUES ('S001', 160, 9365, 100000) -- Giá trị không hợp lệ

Bảng NhanVien

CREATE PROCEDURE InsertNhanVien

AS

BEGIN

DECLARE @Counter INT = 1;

DECLARE @MaDangNhap CHAR(12);

DECLARE @CCCD CHAR(12);

DECLARE @TenNV NVARCHAR(50);

DECLARE @TuoiNV INT;

DECLARE @DiaChiValue NVARCHAR(60); -- Renamed variable

DECLARE @SDT CHAR(10);

DECLARE @SoTaiKhoanNH VARCHAR(20);

DECLARE @MatKhau VARCHAR(20);

DECLARE @FirstNames TABLE (ID INT, Name NVARCHAR(50));

DECLARE @LastNames TABLE (ID INT, Name NVARCHAR(50));

DECLARE @DiaChi TABLE (DiaChi NVARCHAR(60));

INSERT INTO @FirstNames (ID, Name) VALUES

(1, N'Nguyễn'),

(2, N'Trần'),

(3, N'Lê'),

(4, N'Phạm'),

(5, N'Ngô'),

(6, N'Đinh'),

(7, N'Hồ'),

(8, N'Tô'),

(9, N'Bùi'),

(10, N'Vũ')

INSERT INTO @LastNames (ID, Name) VALUES

(1, N'An'),

(2, N'Bình'),

(3, N'Chi'),

(4, N'Duy'),

(5, N'Hoàng'),

(6, N'Hương'),

(7, N'Khánh'),

(8, N'Linh'),

(9, N'Mai'),

(10, N'Nam')

INSERT INTO @DiaChi (DiaChi) VALUES

(N'Nguyễn Thị Minh Khai'),

(N'Đường Lê Lợi'),

(N'Đường Trần Hưng Đạo'),

(N'Đường Phạm Văn Đồng'),

(N'Đường Hoàng Văn Thụ')

WHILE @Counter <= 1000

BEGIN

-- Tạo mã đăng nhập ngẫu nhiên

SET @MaDangNhap = ‘NV’ + RIGHT('0000000000’' + cast(@counter as varchar),10)

-- Tạo số CCCD ngẫu nhiên

SET @CCCD = RIGHT('000000000000' + CAST(ABS(CHECKSUM(NEWID())) AS VARCHAR(12)), 12);

-- Tạo tên ngẫu nhiên

SET @TenNV = (SELECT TOP 1 Name FROM @FirstNames ORDER BY NEWID()) + ' ' +

(SELECT TOP 1 Name FROM @LastNames ORDER BY NEWID());

-- Tạo tuổi ngẫu nhiên từ 18 đến 60

SET @TuoiNV = ABS(CHECKSUM(NEWID()) % 8) + 18;

-- Tạo địa chỉ ngẫu nhiên

SET @DiaChiValue = (SELECT TOP 1 DiaChi FROM @DiaChi ORDER BY

NEWID());

-- Tạo số điện thoại ngẫu nhiên

SET @SDT = '0' + RIGHT('0000000000' + CAST(ABS(CHECKSUM(NEWID())) AS VARCHAR(10)), 9);

-- Tạo số tài khoản ngân hàng ngẫu nhiên

SET @SoTaiKhoanNH = RIGHT('00000000000000000000' + CAST(ABS(CHECKSUM(NEWID())) AS VARCHAR(20)), 20);

-- Tạo mật khẩu ngẫu nhiên

SET @MatKhau = LEFT(NEWID(), 8); -- Mật khẩu dài 8 ký tự

-- Chèn dữ liệu vào bảng NhanVien

INSERT INTO NhanVien (MaDangNhap, CCCD, TenNV, TuoiNV, DiaChi, SDT, SoTaiKhoanNH, MatKhau)

VALUES (@MaDangNhap, @CCCD, @TenNV, @TuoiNV, @DiaChiValue, @SDT, @SoTaiKhoanNH, @MatKhau);

SET @Counter = @Counter + 1;

END

END

Bảng Chủ Quán

CREATE PROCEDURE InsertchuQuan

AS

BEGIN

DECLARE @Counter INT = 1;

DECLARE @SoCCCD CHAR(12);

DECLARE @TenChuQuan NVARCHAR(50);

DECLARE @TuoiChuQuan INT;

DECLARE @NoiOHienTai NVARCHAR(50);

DECLARE @SoDienThoai CHAR(10);

DECLARE @Username CHAR(12);

DECLARE @Pass VARCHAR(20);

DECLARE @FirstNames TABLE (ID INT, Name NVARCHAR(50));

DECLARE @LastNames TABLE (ID INT, Name NVARCHAR(50));

DECLARE @DiaChi TABLE (DiaChi NVARCHAR(100));

INSERT INTO @FirstNames (ID, Name) VALUES

(1, N'Nguyễn'),

(2, N'Trần'),

(3, N'Lê'),

(4, N'Phạm'),

(5, N'Ngô'),

(6, N'Đinh'),

(7, N'Hồ'),

(8, N'Tô'),

(9, N'Bùi'),

(10, N'Vũ')

INSERT INTO @LastNames (ID, Name) VALUES

(1, N'An'),

(2, N'Bình'),

(3, N'Chi'),

(4, N'Duy'),

(5, N'Hoàng'),

(6, N'Hương'),

(7, N'Khánh'),

(8, N'Linh'),

(9, N'Mai'),

(10, N'Nam')

INSERT INTO @DiaChi (DiaChi) VALUES

(N'Nguyễn Thị Minh Khai'),

(N'Đường Lê Lợi'),

(N'Đường Trần Hưng Đạo'),

(N'Đường Phạm Văn Đồng'),

(N'Đường Hoàng Văn Thụ')

-- Tạo mã đăng nhập ngẫu nhiên

WHILE @Counter <= 1000

BEGIN

-- Tạo mã đăng nhập ngẫu nhiên

SET @Username = 'CQ' + RIGHT('0000000000' + cast(@counter as varchar),10)

-- Tạo số CCCD ngẫu nhiên

SET @SoCCCD = RIGHT('000000000000' + CAST(ABS(CHECKSUM(NEWID())) AS VARCHAR(12)), 12);

-- Tạo tên ngẫu nhiên

SET @TenChuQuan = (SELECT TOP 1 Name FROM @FirstNames ORDER BY NEWID()) + ' ' +

(SELECT TOP 1 Name FROM @LastNames ORDER BY NEWID());

-- Tạo tuổi ngẫu nhiên từ 18 đến 80

SET @TuoiChuQuan = ABS(CHECKSUM(NEWID()) % 63) + 18;

-- Tạo địa chỉ hiện tại ngẫu nhiên

SET @NoiOHienTai = (SELECT TOP 1 DiaChi FROM @DiaChi ORDER BY NEWID());

-- Tạo số điện thoại ngẫu nhiên

SET @SoDienThoai = '0' + RIGHT('0000000000' + CAST(ABS(CHECKSUM(NEWID())) AS VARCHAR(10)), 9);

-- Tạo username ngẫu nhiên

SET @Username = RIGHT('000000000000' + CAST(ABS(CHECKSUM(NEWID())) AS VARCHAR(12)), 12);

-- Tạo mật khẩu ngẫu nhiên

SET @Pass = LEFT(NEWID(), 8); -- Mật khẩu dài 8 ký tự

-- Chèn dữ liệu vào bảng ChuQuan

INSERT INTO ChuQuan (SoCCCD, TenChuQuan, TuoiChuQuan, NoiOHienTai, SoDienThoai, Username, Password)

VALUES (@SoCCCD, @TenChuQuan, @TuoiChuQuan, @NoiOHienTai, @SoDienThoai, @Username, @Pass);

SET @Counter = @Counter + 1;

END

END;

-- Gọi thủ tục để thực thi

EXEC InsertChuQuan;

Bảng CapBacNV

CREATE or alter PROCEDURE CapBac

AS

BEGIN

DECLARE @Counter INT = 0; -- Đếm số bản ghi đã chèn thành công

DECLARE @MaDangNhap CHAR(12);

DECLARE @CapBac CHAR(2);

-- Giả định các cấp bậc

DECLARE @CapBacOptions TABLE (CapBac CHAR(2));

INSERT INTO @CapBacOptions (CapBac) VALUES

(N'S1'), -- Cấp thấp nhất với mức lương 15k

(N'S2'), -- Cấp thứ 2 với mức lương 17k

(N'SS'); -- Cấp cao nhất trong nhân viên với mức lương 20k

WHILE @Counter < 1000

BEGIN

-- Chọn ngẫu nhiên MaDangNhap từ bảng NhanVien

SET @MaDangNhap = (SELECT TOP 1 MaDangNhap FROM NhanVien ORDER BY NEWID());

-- Chọn ngẫu nhiên cấp bậc từ danh sách

SET @CapBac = (SELECT TOP 1 CapBac FROM @CapBacOptions ORDER BY NEWID());

-- Chèn dữ liệu vào bảng CapBacNV

BEGIN TRY

INSERT INTO CapBacNV (MaDangNhap, CapBac)

VALUES (@MaDangNhap, @CapBac);

SET @Counter = @Counter + 1; -- Tăng biến đếm chỉ khi chèn thành công

END TRY

BEGIN CATCH

-- Nếu chèn thất bại do trùng lặp, bỏ qua và tiếp tục

-- Không cần làm gì ở đây, chỉ cần tiếp tục vòng lặp

END CATCH

END

END;

GO

exec capbac

select \* from CapBacNV

Bảng LichLam

CREATE OR ALTER PROCEDURE TaoDuLieuLichLam

AS

BEGIN

DECLARE @i INT = 0; -- Bắt đầu từ 0 để tạo ngày làm việc từ năm 2024

DECLARE @Ca INT;

DECLARE @MaBang CHAR(4);

DECLARE @NgayLam DATE;

DECLARE @CaLam NVARCHAR(10);

DECLARE @NgayNghi DATE;

-- Vòng lặp để tạo dữ liệu cho từng ngày làm việc từ 01/01/2024 đến 31/12/2024

WHILE @i < 366 -- Tổng cộng có 366 ngày trong năm 2024 (năm nhuận)

BEGIN

SET @NgayLam = DATEADD(DAY, @i, '2024-01-01'); -- Bắt đầu từ ngày 01/01/2024

-- Lặp qua từng ca (1: Sáng, 2: Chiều, 3: Tối) cho mỗi ngày

SET @Ca = 1;

WHILE @Ca <= 3

BEGIN

-- Cập nhật MaBang với định dạng 4 ký tự, bắt đầu từ L001 cho ngày 01/01/2024

SET @MaBang = 'L' + RIGHT('000' + CAST((@i \* 3 + @Ca + 1) AS VARCHAR), 3);

SET @CaLam = CASE @Ca

WHEN 1 THEN N'Sáng'

WHEN 2 THEN N'Chiều'

WHEN 3 THEN N'Tối' END;

-- Để NgàyNghi là NULL cho tất cả các ca

SET @NgayNghi = NULL;

-- Chèn dữ liệu vào bảng LichLam

INSERT INTO LichLam (MaBang, NgayLam, CaLam, NgayNghi)

VALUES (@MaBang, @NgayLam, @CaLam, @NgayNghi);

SET @Ca = @Ca + 1; -- Tăng ca lên 1

END

SET @i = @i + 1; -- Tăng ngày lên 1

END

END;

exec TaoDuLieuLichLam

Bảng LichLamNV

CREATE OR ALTER PROCEDURE TaoDuLieuLichLamNV

AS

BEGIN

DECLARE @SoLuongLich INT = (SELECT COUNT(\*) FROM LichLam);

DECLARE @index INT = 1;

DECLARE @MaBang CHAR(4);

DECLARE @MaDangNhap CHAR(12);

WHILE @index <= @SoLuongLich

BEGIN

SET @MaBang = 'L' + RIGHT('000' + CAST(@index AS VARCHAR), 3);

-- Lấy MaDangNhap ngẫu nhiên từ bảng NhanVien

SET @MaDangNhap = (SELECT TOP 1 MaDangNhap

FROM NhanVien

ORDER BY NEWID());

INSERT INTO LichLamNV (MaBang, MaDangNhap)

VALUES (@MaBang, @MaDangNhap);

SET @index = @index + 1; -- Tăng chỉ số lên 1

END

END;

exec TaoDuLieuLichLamNV

Bảng ChamCong

CREATE OR ALTER PROCEDURE TaoDuLieuChamCong

AS

BEGIN

DECLARE @MaBangChamCong CHAR(4);

DECLARE @NgayChamCong DATE;

DECLARE @CaLam NVARCHAR(10);

DECLARE @TGVaoCa DATETIME;

DECLARE @TGRaCa DATETIME;

-- Chèn dữ liệu vào bảng ChamCong từ bảng LichLam

INSERT INTO ChamCong (MaBangChamCong, NgayChamCong, TGVaoCa, TGRaCa, CaLam)

SELECT

MaBang,

NgayLam,

CASE CaLam

WHEN N'Sáng' THEN CAST(CONVERT(VARCHAR, NgayLam, 120) + ' 07:00:00' AS DATETIME)

WHEN N'Chiều' THEN CAST(CONVERT(VARCHAR, NgayLam, 120) + ' 12:00:00' AS DATETIME)

ELSE CAST(CONVERT(VARCHAR, NgayLam, 120) + ' 17:00:00' AS DATETIME)

END AS TGVaoCa,

DATEADD(HOUR, 5,

CASE CaLam

WHEN N'Sáng' THEN CAST(CONVERT(VARCHAR, NgayLam, 120) + ' 07:00:00' AS DATETIME)

WHEN N'Chiều' THEN CAST(CONVERT(VARCHAR, NgayLam, 120) + ' 12:00:00' AS DATETIME)

ELSE CAST(CONVERT(VARCHAR, NgayLam, 120) + ' 17:00:00' AS DATETIME)

END) AS TGRaCa,

CaLam

FROM LichLam;

END;

EXEC TaoDuLieuChamCong;

SELECT \* FROM ChamCong

Bảng ChamCongNV

CREATE OR ALTER PROCEDURE TaoDuLieuChamCongNV

AS

BEGIN

DECLARE @MaBangChamCong CHAR(4);

DECLARE @MaDangNhap CHAR(12);

-- Chèn dữ liệu vào bảng ChamCongNV dựa trên bảng ChamCong và LichLam

INSERT INTO ChamCongNV (MaBangChamCong, MaDangNhap)

SELECT

c.MaBangChamCong,

n.MaDangNhap

FROM ChamCong c

JOIN LichLamNV l ON c.MaBangChamCong = l.MaBang

JOIN NhanVien n ON l.MaDangNhap = n.MaDangNhap; -- Giả sử l chứa MaDangNhap

END;

EXEC TaoDuLieuChamCongNV;

SELECT \* FROM ChamCongNV

Bảng Lương

--Hàm tính tiền lương cho nhân viên

CREATE FUNCTION TienLuong (@SoGioLamTrongThang INT, @MaDangNhap CHAR(12))

RETURNS INT

AS

BEGIN

DECLARE @TienLuong INT, @CapBac CHAR(2), @LuongMoiGio INT

SELECT @CapBac = CapBac FROM CapBacNV WHERE MaDangNhap = @MaDangNhap

SET @LuongMoiGio = CASE WHEN @CapBac = 'S1' THEN 15000

WHEN @CapBac = 'S2' THEN 17000

WHEN @CapBac = 'SS' THEN 20000

ELSE 0

END

SET @TienLuong = @SoGioLamTrongThang \* @LuongMoiGio

RETURN @TienLuong

END

– trigger Tinh So Gio Lam Trong Thang

create trigger TinhSoGioLam

on ChamCong

after insert

as

begin

declare @MaBangChamCong char(4), @MaBangLuong char(4), @SoGioLamTrongThang int

select @MaBangChamCong = inserted.MaBangChamCong from inserted

select top 1 @MaBangLuong = Luong.MaBangLuong

from Luong join LuongNV on Luong.MaBangLuong = LuongNV.MaBangLuong

join ChamCongNV on LuongNV.MaDangNhap = ChamCongNV.MaDangNhap

where ChamCongNV.MaBangChamCong = @MaBangChamCong

select @SoGioLamTrongThang = sum(datediff(hour, TGVaoCa, TGRaCa))

from ChamCong

join ChamCongNV on ChamCong.MaBangChamCong = ChamCongNV.MaBangChamCong

where ChamCongNV.MaBangChamCong = @MaBangChamCong

and month (ChamCong.NgayChamCong) = MONTH(GETDATE())

and year(ChamCong.NgayChamCong) = YEAR(GETDATE())

update Luong

set SoGioLamTrongThang = @SoGioLamTrongThang

where MaBangLuong = @MaBangLuong

end

– tao insert du lieu

create proc InsertLuong

as

begin

declare @MaBangLuong char(4), @SoGioBatDauLam int, @TienLuong int,

@MaDangNhap char(12), @dem int = 1

while @dem <= 1000

begin

-- Tạo mã bảng lương

set @MaBangLuong = 'S' + RIGHT('000' + cast(@dem as varchar), 3)

-- Kiểm tra mã bảng lương đã tồn tại hay chưa

while exists (select 1 from Luong where MaBangLuong = @MaBangLuong)

begin

-- Tạo lại mã bảng lương nếu đã tồn tại

set @MaBangLuong = 'S' + RIGHT('000' + cast(@dem as varchar), 3)

end

-- Lấy mã đăng nhập ngẫu nhiên

select top 1 @MaDangNhap = MaDangNhap from NhanVien order by NEWID()

-- Tạo giờ bắt đầu làm ngẫu nhiên từ 900 đến 10000

set @SoGioBatDauLam = floor(900 + (rand() \* 9100))

-- Tạm thời thiết lập tiền lương là 0, trigger sẽ cập nhật sau

set @TienLuong = 0

insert into Luong (MaBangLuong, SoGioLamTrongThang, SoGioBatDauLam, TienLuong)

values (@MaBangLuong, 0, @SoGioBatDauLam, @TienLuong)

set @dem = @dem + 1

end

end

CREATE OR ALTER PROC InsertLuong

AS

BEGIN

DECLARE @MaBangLuong CHAR(4),

@SoGioBatDauLam INT,

@TienLuong INT,

@MaDangNhap CHAR(12),

@dem INT = 1,

@SoGioLamTrongThang INT;

WHILE @dem <= 1000

BEGIN

-- Tạo mã bảng lương

SET @MaBangLuong = 'S' + RIGHT('000' + CAST(@dem AS VARCHAR), 3);

-- Kiểm tra mã bảng lương đã tồn tại hay chưa

WHILE EXISTS (SELECT 1 FROM Luong WHERE MaBangLuong = @MaBangLuong)

BEGIN

-- Tạo lại mã bảng lương nếu đã tồn tại

SET @MaBangLuong = 'S' + RIGHT('000' + CAST(@dem AS VARCHAR), 3);

END

-- Lấy mã đăng nhập ngẫu nhiên

SELECT TOP 1 @MaDangNhap = MaDangNhap FROM NhanVien ORDER BY NEWID();

-- Tạo giờ bắt đầu làm ngẫu nhiên từ 900 đến 10000

SET @SoGioBatDauLam = FLOOR(900 + (RAND() \* 9100));

-- Tính số giờ làm trong tháng từ bảng ChamCongNV

SELECT @SoGioLamTrongThang = SUM(DATEDIFF(HOUR, TGVaoCa, TGRaCa))

FROM ChamCongNV join ChamCong on ChamCong.MaBangChamCong = ChamCongNV.MaBangChamCong

WHERE MaDangNhap = @MaDangNhap

AND MONTH(NgayChamCong) = MONTH(GETDATE())

AND YEAR(NgayChamCong) = YEAR(GETDATE());

IF @SoGioLamTrongThang IS NULL

SET @SoGioLamTrongThang = 0;

-- Tính tiền lương dựa trên số giờ làm

SET @TienLuong = dbo.TienLuong(@SoGioLamTrongThang, @MaDangNhap);

-- Chèn thông tin lương vào bảng Luong

INSERT INTO Luong (MaBangLuong, SoGioLamTrongThang, SoGioBatDauLam, TienLuong)

VALUES (@MaBangLuong, @SoGioLamTrongThang, @SoGioBatDauLam, @TienLuong);

SET @dem = @dem + 1;

END

END

exec InsertLuong

Bảng LuongNV

create proc InsertLuongNV

as

begin

declare @MaBangLuong char(4), @MaDangNhap CHAR(12), @dem int = 1

while @dem <= 1000

begin

set @MaDangNhap = (select top 1 MaDangNhap from NhanVien order by NEWID())

set @MaBangLuong = (select top 1 MaBangLuong from Luong order by NEWID())

insert into LuongNV ( MaDangNhap, MaBangLuong)

values (@MaDangNhap, @MaBangLuong)

set @dem = @dem+ 1

end

end

exec InsertLuongNV

Bảng ThongBao

1. **Hàm tạo thông báo cho chủ quán:**

*( Yêu cầu: mã thông báo = max mã thông báo trong table thông báo + 1, thời gian thông báo = getdate(),tên thông báo, nội dung thông báo )*

go

CREATE PROCEDURE TaoDuLieDuLieuThongBao

AS

BEGIN

DECLARE @Counter INT = 1;

DECLARE @MaTB\_moi CHAR(4);

DECLARE @MaTB\_cu CHAR(4);

DECLARE @TenThongBao NVARCHAR(100);

DECLARE @NoiDungThongBao NVARCHAR(200);

DECLARE @ThongBao TABLE (TenThongBao NVARCHAR(100), NoiDungThongBao NVARCHAR(200));

INSERT INTO @ThongBao (TenThongBao, NoiDungThongBao) VALUES

(N'Thông báo khuyến mãi trà sữa', N'Giảm giá 20% cho các loại trà sữa'),

(N'Thông báo khai trương cửa hàng mới', N'Khai trương cửa hàng mới tại 225 Lê Thanh Nghị'),

(N'Thông báo sự kiện âm nhạc', N'Sự kiện âm nhạc vào ngày mai với ca sĩ khách mời '),

(N'Thông báo chương trình tri ân khách hàng', N'Tri ân khách hàng với nhiều phần quà hấp dẫn'),

(N'Thông báo ngày hội hoa sen', N'Ngày hội hoa sen với các mã trà sữa từ hoa sen giảm giá 30%');

set @MaTB\_cu = 0001

IF @MaTB\_cu IS NOT NULL

BEGIN

SET @MaTB\_moi = RIGHT('000' + CAST(CAST(@MaTB\_cu AS INT) + 1 AS VARCHAR(4)), 4)

END

WHILE @Counter <= 1000

BEGIN

-- Chọn một thông báo ngẫu nhiên từ bảng tạm

SELECT TOP 1

@TenThongBao = TenThongBao,

@NoiDungThongBao = NoiDungThongBao

FROM @ThongBao

ORDER BY NEWID();

-- Chèn thông báo vào bảng ThongBao

INSERT INTO ThongBao (MaTB, TenThongBao, ThoiGianThongBao, NoiDungThongBao)

VALUES (@MaTB\_moi, @TenThongBao, GETDATE(), @NoiDungThongBao);

-- Tăng mã thông báo cho lần lặp tiếp theo

SET @MaTB\_moi = RIGHT('000' + CAST(CAST(@MaTB\_moi AS INT) + 1 AS VARCHAR(4)), 4);

SET @Counter = @Counter + 1;

END

END

GO

-- Thực hiện thủ tục

EXEC TaoDuLieuThongBao;

drop procedure TaoThongBao

select \* from ThongBao

Bang ThongBaoNV

CREATE PROCEDURE TaoDuLieuThongBaoNV

AS

BEGIN

DECLARE @SoLuongThongBao INT = (SELECT COUNT(\*) FROM ThongBao);

DECLARE @index INT = 1;

DECLARE @MaTB CHAR(4);

DECLARE @MaDangNhap CHAR(12);

WHILE @index <= @SoLuongThongBao

BEGIN

SET @MaTB = (SELECT TOP 1 MaTB FROM ThongBao ORDER BY NEWID());

SET @MaDangNhap = (SELECT TOP 1 MaDangNhap

FROM NhanVien

ORDER BY NEWID());

INSERT INTO ThongBaoNV (MaTB, MaDangNhap)

VALUES (@MaTB, @MaDangNhap);

SET @index = @index + 1; -- Tăng chỉ số lên 1

END

END;

GO

-- Thực hiện thủ tục

EXEC TaoDuLieuThongBaoNV;

drop procedure TaoDuLieuThongBaoNV

select \* from ThongBaoNV

select \* from ThongBao

Phân tích

1. Hàm tính tiền lương cho nhân viên

| Tính tiền lương cho nhân viên  Loại: Hàm | |
| --- | --- |
| Mô tả: Tính tiền lương cho nhân viên theo tháng | |
| Input | Số giờ làm trong tháng, Mã đăng nhập |
| Output | Tiền lương |
| Process | 1. Lấy cấp bậc của nhân viên, điều kiện MaDangNhap = mã đăng nhập  2. Tính lương mỗi giờ:  - Nếu cấp bậc = S1 -- > 15000  - Nếu cấp bậc = S2 -- > 17000  - Nếu cấp bậc = SS -- > 20000  - Còn lại là 0  3. Tiền lương = số giờ làm trong tháng \* lương mỗi giờ |
| Code | create or alter function TienLuong (@SoGioLamTrongThang int, @MaDangNhap char(12))  returns int  as  begin  declare @TienLuong int, @CapBac char(2), @LuongMoiGio int    select @CapBac = CapBac from CapBacNV where MaDangNhap = @MaDangNhap    set @LuongMoiGio = case when @CapBac = 'S1' then 15000  when @CapBac = 'S2' then 17000  when @CapBac = 'SS' then 20000  else 0  end  set @TienLuong = @SoGioLamTrongThang \* @LuongMoiGio  return @TienLuong  end |

1. Thủ tục tạo thông báo

| Tạo thông báo  Loại: Thủ tục | |
| --- | --- |
| Mô tả: Tạo thông báo khi chủ quán có thông tin mới muốn thông báo đến cho nhân viên trong quán | |
| Input | Tên thông báo, Nội dung thông báo |
| Output | N/A |
| Process | 1. Lấy mã thông báo cũ = mã thông báo lớn nhất ở trong bảng ThongBao  2. Nếu mã thông báo cũ trống thì set mã thông báo cũ = 000  Ngược lại, tạo mã thông báo mới = mã thông báo cũ + 1 và đảm bảo mã thông báo mới có 4 ký tự  3. Nhập dữ liệu vào bảng ThongBao |
| Code | create or alter procedure TaoThongBao  @TenThongBao nvarchar(100),  @NoiDungThongBao nvarchar(200)  as  begin  declare @MaTB\_moi char(4)  declare @MaTB\_cu char(4)  declare @A int  set @MaTB\_cu = (select max(MaTB) from ThongBao)  if @MaTB\_cu is null  begin  set @MaTB\_cu = 000  end  else  begin  set @A = @MaTB\_cu + 1  set @MaTB\_moi = concat(replicate('0', 4 - len(@A)), @A)  insert into ThongBao (MaTB, TenThongBao, ThoiGianThongBao, NoiDungThongBao)  values (@MaTB\_moi, @TenThongBao, getdate(), @NoiDungThongBao)  select \* from ThongBao where MaTB = @MaTB\_moi  end  end  exec TaoThongBao N'Thông báo chương trình khuyến mãi', N'Tặng khách hàng 1 con labubu khi mua trà sữa'  select \* from ThongBao |

1. Thủ tục cập nhất thông tin nhân viên

| Cập nhật thông tin nhân viên  Loại: Thủ tục | |
| --- | --- |
| Mô tả: Khi thông tin của nhân viên bị sai hoặc có sự thay đổi nào đó như địa chỉ, số điện thoại thì sử dụng hàm để sửa thông tin nhân viên để đảm bảo rằng tất cả thông tin của nhân viên luôn được cập nhật và chính xác, từ đó giúp chủ quán dễ dàng quản lý theo dõi và kiểm soát thông tin. | |
| Input | MaDangNhap, CCCD, TenNV, TuoiNV , DiaChi, SDT, SoTaiKhoanNH, MatKhau |
| Output | In ra giá trị chuỗi ‘Cập nhật không thành công’ nếu số dòng update <= 0  In ra giá trị chuỗi ‘Cập nhật thành công’ nếu số dòng update > 0 |
| Process | 1. Update bảng nhân viên thay đổi thông tin với điều kiện MaDangNhap = @MaDangNhap 2. Nếu số dòng update = 0 thì in ra ‘Cập nhật không thành công’   Ngược lại, in ra 'Cập nhật thành công' |
| Code | create or alter PROC CapNhatThongTin(@MaDangNhap char(12),  @CCCD char(12),  @TenNV nvarchar(50) ,  @TuoiNV int ,  @DiaChi nvarchar(60),  @SDT char(10) ,  @SoTaiKhoanNH varchar(20) ,  @MatKhau varchar(20) )  as  begin  update NhanVien  set CCCD = @CCCD,  TenNV = @TenNV,  TuoiNV = @TuoiNV,  DiaChi = @DiaChi,  SDT = @SDT,  SoTaiKhoanNH = @SoTaiKhoanNH,  MatKhau = @MatKhau  where MaDangNhap = @MaDangNhap  if @@ROWCOUNT = 0  begin  print N'Cập nhật không thành công'  end  else  begin  print N'Cập nhật thành công'  end  end  -- gọi thủ tục  exec CapNhatThongTin @MaDangNhap = 'NV0000000020',  @CCCD = '123456789012',  @TenNV = N'Nguyễn Văn A',  @TuoiNV = 30,  @DiaChi = N'123 Đường ABC, Quận 1',  @SDT = '0123456789',  @SoTaiKhoanNH = '1234567890',  @MatKhau = 'MatKhauMoi' |

1. Thủ tục kiểm tra mã đăng nhập đã tồn tại hay chưa

| Kiểm tra mã đăng nhập đã tồn tại hay chưa?  Loại: Thủ tục | |
| --- | --- |
| Mô tả: Kiểm tra mã đăng nhập đã tồn tại trong bảng nhân viên hay chưa | |
| Input | mã đăng nhập |
| Output | trả về chuỗi: - Mã đăng nhập đã tồn tại / Mã đăng nhập chưa tồn tại |
| Process | 1. kiểm tra xem có tồn tại nhân viên có MaDangNhap = mã đăng nhập không  - Nếu có --> Mã đăng nhập đã tồn tại  - Còn lại --> Mã đăng nhập chưa tồn tại |
| Code | create or alter PROC CheckMaDangNhap(@MaDangNhap char(12), @KiemTra nvarchar(50) output)  as  begin  if exists (select \* from NhanVien where MaDangNhap = @MaDangNhap)  begin  set @KiemTra = N'Mã đăng nhập đã tồn tại'  end  else  begin  set @KiemTra = N'Mã đăng nhập chưa tồn tại'  end  end  declare @mdn varchar(12), @kt nvarchar(50)  exec CheckMaDangNhap 'NV0000000001', @kt output  print @kt |

1. Thủ tục tạo mới thông tin nhân viên

| Tạo mới thông tin nhân viên  Loại: Thủ tục | |
| --- | --- |
| Mô tả: Khi có nhân viên mới vào làm, thủ tục sẽ tạo thông tin cho nhân viên mới | |
| Input | MaDangNhap, CCCD, TenNV, TuoiNV , DiaChi, SDT, SoTaiKhoanNH, MatKhau |
| Output | trả về chuỗi: Mã đăng nhập đã tồn tại  N/A |
| Process | 1. kiểm tra xem có tồn tại nhân viên có MaDangNhap = mã đăng nhập không  - Nếu có --> Mã đăng nhập đã tồn tại  - Còn lại --> Nhập dữ liệu vào bảng NV |
| Code | create or alter PROC TaoThongTinNV ( @MaDangNhap char(12),  @CCCD char(12),  @TenNV nvarchar(50) ,  @TuoiNV int ,  @DiaChi nvarchar(60),  @SDT char(10) ,  @SoTaiKhoanNH varchar(20) ,  @MatKhau varchar(20),  @kiemtra nvarchar(50) output)  as  begin  if exists (select \* from NhanVien where MaDangNhap = @MaDangNhap)  begin  set @KiemTra = N'Mã đăng nhập đã tồn tại'  end  else  begin  insert into NhanVien  values( @MaDangNhap, @CCCD, @TenNV, @TuoiNV, @DiaChi, @SDT, @SoTaiKhoanNH, @MatKhau)  end  end  declare @mdn char(12), @cc char(12), @ten nvarchar(50) ,  @tuoi int , @dc nvarchar(60), @sodt char(10) ,  @stk varchar(20) , @mk varchar(20), @kt nvarchar(50)  exec TaoThongTinNV 'NV0000000001', '123456789012', 'ABC', 20, 'Quang Ngai', '0965525315', '0987654321', 'ABCDE', @kt output  print @kt  declare @mdn char(12),  @cc char(12),  @ten nvarchar(50) ,  @tuoi int ,  @dc nvarchar(60),  @sodt char(10) ,  @stk varchar(20) ,  @mk varchar(20),  @kt nvarchar(50)  exec TaoThongTinNV 'NV5487294765', '124456789015', 'ABC', 20, 'Quang Ngai', '0961525315', '0980654321', 'ABCDE', @kt output  print @kt  select \* from NhanVien |

1. Thủ tục cập nhật Ngày nghỉ trong bảng lịch làm

| Cập nhật ngày nghỉ  Loại: Thủ tục | |
| --- | --- |
| Mô tả: Cập nhật lại ngày nghỉ cho nhân viên nếu nhân viên xin nghỉ | |
| Input | ngày |
| Output | Trả về chuỗi: 'Cập nhật không thành công' hoặc 'Cập nhật thành công' |
| Process | 1. cập nhật cột NgayNghi = ngày, điều kiện NgayLam = ngày  2. Nếu số dòng cập nhật = 0 thì trả về chuỗi 'Cập nhật không thành công'  Ngược lại, trả về chuỗi 'Cập nhật thành công' |
| Code | create or alter PROC CapNhatNgayNghi (@Ngay date, @kiemtra nvarchar(50) output)  as  begin  update LichLam  set NgayNghi = @Ngay  where NgayLam = @Ngay  if @@ROWCOUNT = 0  begin  set @kiemtra = N'Cập nhật không thành công'  end  else  begin  set @kiemtra = N'Cập nhật thành công'  end  end  declare @nn date, @kt nvarchar(50)  exec CapNhatNgayNghi '2024-10-17', @kt output  print @kt |

1. Thủ tục để sửa đổi thời gian chấm công

| Sửa đổi thời gian chấm công  Loại: Thủ tục | |
| --- | --- |
| Mô tả: Trong trường hợp có sai sót hoặc lỗi hệ thống dẫn đến việc chấm công không chính xác, thủ tục này cho phép quản lý dễ dàng sửa đổi thời gian chấm công để đảm bảo tính chính xác. | |
| Input | ngày chấm công, ca làm, thời gian vào ca, thời gian ra ca |
| Output | Trả về chuỗi: 'Cập nhật không thành công' hoặc 'Cập nhật thành công' |
| Process | 1. cập nhật TGVaoCa = thời gian vào ca  TGRaCa = thời gian ra  với điều kiện NgayChamCong = ngày chấm công và CaLam = ca làm  2. Nếu số dòng cập nhật = 0 thì trả về chuỗi 'Cập nhật không thành công'  Ngược lại, trả về chuỗi 'Cập nhật thành công' |
| Code | create or alter PROC CapNhatThoiGianChamCong ( @NgayChamCong date,  @CaLam nvarchar(20),  @TGVaoCa datetime,  @TGRaCa datetime,  @kiemtra nvarchar(50) output)  as  begin  update ChamCong  set TGVaoCa = @TGVaoCa,  TGRaCa = @TGRaCa  where NgayChamCong = @NgayChamCong  and CaLam = @CaLam  if @@ROWCOUNT = 0  begin  set @kiemtra = N'Cập nhật không thành công'  end  else  begin  set @kiemtra = N'Cập nhật thành công'  end  end  declare @ngay date, @ca nvarchar(20), @vao datetime, @ra datetime, @kt nvarchar(50)  exec CapNhatThoiGianChamCong '2024-01-01', N'Sang', '2024-01-01 07:20:00.000', '2024-01-01 12:20:00.000', @kt output  print @kt |

1. Trigger cập nhật số giờ làm trong tháng

| Cập nhật số giờ làm trong tháng  Loại: Trigger | |
| --- | --- |
| Mô tả: Trigger này kiểm tra rằng mức lương phải nằm trong khoảng lương hợp lý dựa trên cấp bậc của nhân viên, điều này giúp ngăn chặn việc nhập các giá trị lương không hợp lý, có thể dẫn đến sự không công bằng trong quản lý lương. | |
| Loại | after |
| Ký sinh | ChamCong |
| Sự kiện | insert |
| Xử lý | 1. lấy MaBangChamCong ChamCong từ bảng insertedaBangLuongChamCongChamCong  2. lấy MaBangLuongMaBangLuong từ bảng Luong nối bảng ChamCongNV với điều kiện MaBangChamCong = @MaBangChamCongMaBangLuong, điều kiện ChamCongNV.MaBangChamCong = @MaBangChamCong  3. Tính SoGioLamTrongThang = tổng giờ của (tgraca - tgvaoca) với điều kiện MaBangChamCong = MaBangChamCong và tháng chấm công = tháng hiện tại và năm chấm công = năm hiện tạiTính số giờ làm trong tháng = tổng (giờ, thời gian ra ca - thời gian vào ca), điều kiện ChamCongNV.MaBangChamCong = @MaBangChamCong và month (ChamCong.NgayChamCong) = MONTH(GETDATE())  và year(ChamCong.NgayChamCong) = YEAR(GETDATE())  4. Cập nhật bảng Luong: SoGioLamTrongThang = @SoGioLamTTrongThang với điều kiện MaBangLuong = @MaBangLuongCập nhật  SoGioLamTrongThang = số giờ làm trong tháng  điều kiện MaBangLuong = @MaBangLuong |
| Code | create or alter trigger TinhSoGioLam  on ChamCong  after insert  as  begin  declare @MaBangChamCong char(4),  @MaBangLuong char(4),  @SoGioLamTrongThang int  select @MaBangChamCong = inserted.MaBangChamCong  from inserted  select top 1 @MaBangLuong = Luong.MaBangLuong  from Luong join LuongNV on Luong.MaBangLuong = LuongNV.MaBangLuong  join ChamCongNV on LuongNV.MaDangNhap = ChamCongNV.MaDangNhap  where ChamCongNV.MaBangChamCong = @MaBangChamCong  select @SoGioLamTrongThang = sum(datediff(hour, TGVaoCa, TGRaCa))  from ChamCong join ChamCongNV on ChamCong.MaBangChamCong = ChamCongNV.MaBangChamCong  where ChamCongNV.MaBangChamCong = @MaBangChamCong  and month(ChamCong.NgayChamCong) = month(getdate())  and year(ChamCong.NgayChamCong) = year(getdate())  update Luong  set SoGioLamTrongThang = @SoGioLamTrongThang  where MaBangLuong = @MaBangLuong  endcreate trigger TinhSoGioLam  on ChamCong  after insert  as  begin  declare @MaBangChamCong char(4), @MaBangLuong char(4), @SoGioLamTrongThang int  select @MaBangChamCong = inserted.MaBangChamCong from inserted  select top 1 @MaBangLuong = Luong.MaBangLuong  from Luong join LuongNV on Luong.MaBangLuong = LuongNV.MaBangLuong  join ChamCongNV on LuongNV.MaDangNhap = ChamCongNV.MaDangNhap  where ChamCongNV.MaBangChamCong = @MaBangChamCong  select @SoGioLamTrongThang = sum(datediff(hour, TGVaoCa, TGRaCa))  from ChamCong  join ChamCongNV on ChamCong.MaBangChamCong = ChamCongNV.MaBangChamCong  where ChamCongNV.MaBangChamCong = @MaBangChamCong  and month (ChamCong.NgayChamCong) = MONTH(GETDATE())  and year(ChamCong.NgayChamCong) = YEAR(GETDATE())  update Luong  set SoGioLamTrongThang = @SoGioLamTrongThang  where MaBangLuong = @MaBangLuong  end |

1. Trigger đảm bảo nhân viên không đăng ký hai ca làm việc trùng giờ trong cùng một ngày.

| Kiểm tra ca làm việc trùng hay  Loại: Trigger | |
| --- | --- |
| Mô tả: Trigger này sẽ kiểm tra xem nhân viên đã đăng ký ca làm việc trùng giờ trong cùng một ngày chưa, nếu đã tồn tại thì sẽ phát sinh lỗi và hủy giao dịch. Nếu không có lỗi, trigger sẽ thực hiện chèn dữ liệu bình thường. | |
| Loại | for |
| Kí sinh | LichLamNV |
| Sự kiện | insert |
| Xử lý | 1. Kiểm tra có tồn tại 1 bản ghi nào trong bảng inserted trùng trong bảng LichLam với điều kiện NgayLam = NgayLam và CaLam = CaLam và MaDangNhap = MaDangNhap và MaBang khác MaBang  Nếu có, in ra 'Nhân viên đã đăng ký ca làm việc trùng giờ trong cùng một ngày' + rollback |
| Code | create or alter trigger KiemTraTrungCaLamViec  on LichLamNV  for insert  as  begin  -- Kiểm tra xem có nhân viên nào đăng ký trùng ca làm việc và ngày làm không  if exists (  select 1  from inserted I  join LichLam LL on LL.MaBang = I.MaBang  join LichLamNV LLNV on LLNV.MaDangNhap = I.MaDangNhap  join LichLam LL2 on LL2.MaBang = LLNV.MaBang  where LL.NgayLam = LL2.NgayLam  and LL.CaLam = LL2.CaLam  and LLNV.MaDangNhap = I.MaDangNhap  and LLNV.MaBang <> I.MaBang -- Điều kiện này đảm bảo kiểm tra với các bản ghi trước đó  )  begin  -- Nếu nhân viên đã đăng ký trùng ca làm trong ngày đó, rollback giao dịch  print N'Nhân viên đã đăng ký ca làm việc trùng giờ trong cùng một ngày'  rollback  end  end  -- test  insert into LichLam values ('LLLM', '2024-12-10', 'Sang', Null)  insert into LichLamNV values ('LLLM', 'NV0000000001')  insert into LichLamNV values ('LLLM', 'NV0000000001') |

1. Trigger Kiểm tra thêm nhân viên

| Kiểm tra thêm nhân viên  Loại: Trigger | |
| --- | --- |
| Mô tả: Trigger này sẽ kiểm tra xem mã đăng nhập của nhân viên có lã mã mới hay không và có đủ 12 ký tự hay không, số điện thoại có hợp lệ (số điện thoại có 10 số) )hay không trước khi thêm bản ghi mới vào bảng nhân viên. | |
| Loại | for |
| Kí sinh | NhanVien |
| Sự kiện | insert |
| Xử lý | 1. tìm MaDangNhap, Sdt trong bảng inserted --> @MaDangNhap, @SDT  2. kiểm tra nếu @MaDangNhap không bằng 12 ký tự --> 'Mã đăng nhập phải có đúng 12 ký tự' + rollback  3. kiểm tra nếu @SDT không bằng 10 ký tự --> 'Số điện thoại phải có đúng 10 chữ số' + rollback |
| Code | create or alter trigger KiemTraThemNhanVien  on NhanVien  for insert  as  begin  declare @MaDangNhap nvarchar(12), @SoDienThoai nvarchar(10)    select @MaDangNhap = inserted.MaDangNhap, @SoDienThoai = inserted.SDT  from inserted  if len(@MaDangNhap) < > 12  begin  print N'Mã đăng nhập phải có đúng 12 ký tự'  rollback  end  else if len(@SoDienThoai) < > 10  begin  print N'Số điện thoại phải có đúng 10 chữ số'  rollback  end  end  -- test  insert into NhanVien values  ('NV9876543213', '123456789011', 'ABC', 20, 'Quang Ngai', '0965525353', '0987654321', 'ABCDE')  insert into NhanVien values  ('NV987654321', '123456789013', 'ABC', 20, 'Quang Ngai', '0965525374', '0987654321', 'ABCDE')  insert into NhanVien values  ('NV9876543217', '123456789018', 'ABC', 20, 'Quang Ngai', '09655253', '0987654321', 'ABCDE') |

1. Kiểm tra trùng ca làm việc

| Trigger đảm bảo nhân viên không đăng ký hai ca làm việc trùng giờ trong cùng một ngày.  Loại: Trigger | |
| --- | --- |
| Mô tả: Trigger này sẽ kiểm tra xem nhân viên đã đăng ký ca làm việc trùng giờ trong cùng một ngày chưa, nếu đã tồn tại thì sẽ phát sinh lỗi và hủy giao dịch. Nếu không có lỗi, trigger sẽ thực hiện chèn dữ liệu bình thường. | |
| Loại | after |
| Ký sinh | LichLamNV |
| Sự kiện | insert |
| Xử lý | 1. Nếu có tồn tại ít nhất 1 bản ghi nào từ bảng inserted với điều kiện LichLam.NgayLam = LichLam2.NgayLam  and LichLam.CaLam = LichLam2.CaLam  and LichLamNV.MaDangNhap = Inserted.MaDangNhap  and LichLamNV.MaBang <> Inserted.MaBang  → print N'Nhân viên đã đăng ký ca làm việc trùng giờ trong cùng một ngày' + rollback |
| Code | create or alter trigger KiemTraTrungCaLamViec  on LichLamNV  for insert  as  begin  -- Kiểm tra xem có nhân viên nào đăng ký trùng ca làm việc và ngày làm không  if exists (  select 1  from inserted I  join LichLam LL on LL.MaBang = I.MaBang  join LichLamNV LLNV on LLNV.MaDangNhap = I.MaDangNhap  join LichLam LL2 on LL2.MaBang = LLNV.MaBang  where LL.NgayLam = LL2.NgayLam  and LL.CaLam = LL2.CaLam  and LLNV.MaDangNhap = I.MaDangNhap  and LLNV.MaBang <> I.MaBang -- Điều kiện này đảm bảo kiểm tra với các bản ghi trước đó  )  begin  -- Nếu nhân viên đã đăng ký trùng ca làm trong ngày đó, rollback giao dịch  print N'Nhân viên đã đăng ký ca làm việc trùng giờ trong cùng một ngày'  rollback  end  end  -- test  insert into LichLam values ('LLLM', '2024-12-10', 'Sang', Null)  insert into LichLamNV values ('LLLM', 'NV0000000001')  insert into LichLamNV values ('LLLM', 'NV0000000001') |

1. Thủ tục xóa thông báo Kiểm tra thêm mới nhân viên

| Trigger khi thêm mới nhân viên, đảm bảo rằng mã đăng nhập là mới và phải đủ 12 ký tự, số điện thoại phải hợp lệ  Loại: Trigger | |
| --- | --- |
| Mô tả: Trigger này sẽ kiểm tra xem mã đăng nhập của nhân viên có lã mã mới hay không và có đủ 12 ký tự hay không, số điện thoại có hợp lệ (số điện thoại có 10 số) )hay không trước khi thêm bản ghi mới vào bảng nhân viên. | |
| Loại | after |
| Ký sinh | NhanVien |
| Sự kiện | insert |
| Xử lý | 1. tìm MaDangNhap, Sdt trong bảng inserted --> @MaDangNhap, @SDT  2. kiểm tra nếu @MaDangNhap không bằng 12 ký tự --> 'Mã đăng nhập phải có đúng 12 ký tự' + rollback  3. kiểm tra nếu @SDT không bằng 10 ký tự --> 'Số điện thoại phải có đúng 10 chữ số' + rollback |
| Code | create or alter trigger KiemTraThemNhanVien  on NhanVien  for insert  as  begin  declare @MaDangNhap nvarchar(12), @SoDienThoai nvarchar(10)    select @MaDangNhap = inserted.MaDangNhap, @SoDienThoai = inserted.SDT  from inserted  if len(@MaDangNhap) < > 12  begin  print N'Mã đăng nhập phải có đúng 12 ký tự'  rollback  end  else if len(@SoDienThoai) < > 10  begin  print N'Số điện thoại phải có đúng 10 chữ số'  rollback  end  end  -- test  insert into NhanVien values  ('NV9876543213', '123456789011', 'ABC', 20, 'Quang Ngai', '0965525353', '0987654321', 'ABCDE')  insert into NhanVien values  ('NV987654321', '123456789013', 'ABC', 20, 'Quang Ngai', '0965525374', '0987654321', 'ABCDE')  insert into NhanVien values  ('NV9876543217', '123456789018', 'ABC', 20, 'Quang Ngai', '09655253', '0987654321', 'ABCDE') |

1. Xóa thông báo

| Thủ tục để xóa thông báo  Loại: Thủ tục | |
| --- | --- |
| Mô tả: Thủ tục để cập xóa thông báo khi thông báo không còn phù hợp với tình hình thực tế để đảm bảo rằng nhân viên luôn nhận được thông tin phù hợp trong các tình hình khác nhau của quán. | |
| Input | Mã thông báo |
| Output | Chuỗi giá trị ‘Error’ nếu số dòng đã xóa <= 0 / Còn lại là ‘Done’ |
| Xử lý | 1. xóa thông báo, điều kiện MaTB = mã thông báo  2. kiểm tra xóa thành công hay chưa:  - nếu @@rowcount = 0 --> 'Error'  - Ngược lại: 'Done' |
| Code | create procedure XoaThongBao (@MaTB char(4))  as  begin  delete from ThongBao  where MaTB = @MaTB;  if @@rowcount = 0  begin  print 'Error!';  end  else  begin  print 'Done!';  end  end  exec XoaThongBao '1002' |

1. Xóa thông báo tự động

| Trigger để xóa thông báo lỗi thời  Loại: Trigger | |
| --- | --- |
| Mô tả: Trigger này kiểm tra dung lượng của bản ghi và xóa những thông báo bị lỗi thời giải phóng dung lượng | |
| Loại | after |
| Ký sinh | ThongBao |
| Sự kiện | insert |
| Xử lý | 1. xóa thông báo với điều kiện thời gian đã qua 1 năm: ThoiGianThongBao < dateadd(year, -1, getdate()) |
| Code | create trigger XoaThongBaoTuDong  on ThongBao  after insert  as  begin  delete from ThongBao  where ThoiGianThongBao < dateadd(year, -1, getdate())  end |

| Xóa thông báo  Loại: Thủ tục | |
| --- | --- |
| Mô tả: Thủ tục để cập xóa thông báo khi thông báo không còn phù hợp với tình hình thực tế để đảm bảo rằng nhân viên luôn nhận được thông tin phù hợp trong các tình hình khác nhau của quán. | |
| Input | mã thông báo |
| Output | chuỗi giá trị: Error / Done |
| Process | 1. xóa thông báo, điều kiện MaTB = mã thông báo  2. kiểm tra xóa thành công hay chưa:  - nếu số dòng cập nhật = 0 --> 'Error'  - Ngược lại: 'Done' |
| Code | create procedure XoaThongBao (@MaTB char(4))  as  begin  delete from ThongBao  where MaTB = @MaTB;  if @@rowcount = 0  begin  print 'Error!';  end  else  begin  print 'Done!';  end  end  exec XoaThongBao '1002' |

1. Trigger để xóa thông tin bị lỗi thời

| Xóa thông báo quá thời gian  Loại: Trigger | |
| --- | --- |
| Mô tả: Trigger này kiểm tra dung lượng của bản ghi và xóa những thông báo bị lỗi thời giải phóng dung lượng | |
| Loại | after |
| Kí sinh | ThongBao |
| Sự kiện | insert |
| Xử lý | 1. Xóa thông báo với điều kiện thời gian đã qua 1 năm: ThoiGianThongBao < dateadd(year, -1, getdate()) |
| Code | create trigger XoaThongBaoTuDong  on ThongBao  after insert  as  begin  delete from ThongBao  where ThoiGianThongBao < dateadd(year, -1, getdate())  end |

13. Trigger kiểm tra tính hợp lệ của lương

| Kiểm tra mức lương Nhân viên  Loại: Trigger | |
| --- | --- |
| Mô tả: Trigger này kiểm tra rằng mức lương phải nằm trong khoảng lương hợp lý dựa trên cấp bậc của nhân viên, điều này giúp ngăn chặn việc nhập các giá trị lương không hợp lý, có thể dẫn đến sự không công bằng trong quản lý lương. | |
| Loại | after |
| Ký sinh | Luong |
| Sự kiện | insert, update |
| Xử lý | 1. lấy MaBangLuong, TienLuong từ bảng inserted → @MaBangLuong, @TienLuong  2. lấy MaDangNhap từ bảng LuongNV --> @MaDangNhap  3. lấy Capbac từ bảng CapBacNV --> @CapBac  4. tính lương mỗi giờ:  - Nếu cấp bậc = S1 -- > 15000  - Nếu cấp bậc = S2 -- > 17000  - Nếu cấp bậc = SS -- > 20000  - Còn lại là 0  5. kiểm tra lương có hợp lý không: (ít nhất một tháng làm 12 ca và nhiều nhất làm 20 ca (1 ca làm có 5 tiếng))  - Nếu không: print 'Lương không hợp lệ! Vui lòng kiểm tra lại' + rollback |
| Code | create trigger KiemTraMucLuong  on Luong  after insert, update  as  begin  declare @MaBangLuong char(5), @MaDangNhap char(12), @CapBac char(2),  @TienLuong int, @LuongMoiGio int  select @MaBangLuong = MaBangLuong, @TienLuong = TienLuong from inserted  select @MaDangNhap = MaDangNhap from LuongNV where MaBangLuong = @MaBangLuong  select @CapBac = CapBac from CapBacNV where MaDangNhap = @MaDangNhap  set @LuongMoiGio = case  when @CapBac = 'S1' then 15000  when @CapBac = 'S2' then 17000  when @CapBac = 'SS' then 20000  else 0  end  -- kiểm tra xem lương có nằm trong khoảng hợp lý không  if @TienLuong < (@LuongMoiGio \* 60) or @TienLuong > (@LuongMoiGio \* 100 )  begin  print N'Lương không hợp lệ! Vui lòng kiểm tra lại.'  rollback  end  end  insert into Luong (MaBangLuong, SoGioLamTrongThang, SoGioBatDauLam, TienLuong)  values ('S001', 160, 9365, 100000) |