DROP DATABASE IF EXISTS QuanLyThuVien;

GO

CREATE DATABASE QuanLyThuVien;

GO

USE QuanLyThuVien;

GO

-- 1. Tạo bảng danh mục sách

CREATE TABLE DanhMucSach (

MaDanhMuc NVARCHAR(20) PRIMARY KEY,

TenDanhMuc NVARCHAR(255) NOT NULL,

MoTa NVARCHAR(500),

DanhMucCha NVARCHAR(20) NULL,

SoLuongSach INT DEFAULT 0 CHECK (SoLuongSach >= 0),

NgayTao DATE DEFAULT GETDATE(),

CapNhatLanCuoi DATE DEFAULT GETDATE(),

TrangThai NVARCHAR(20) DEFAULT N'Hoạt động'

CHECK (TrangThai IN (N'Hoạt động', N'Ngừng hoạt động'))

);

GO

CREATE TRIGGER trg\_SetNull\_DanhMucSach

ON DanhMucSach

INSTEAD OF DELETE

AS

BEGIN

SET NOCOUNT ON;

-- Cập nhật DanhMucCha thành NULL cho các danh mục con

UPDATE DanhMucSach

SET DanhMucCha = NULL

WHERE DanhMucCha IN (SELECT MaDanhMuc FROM deleted);

-- Sau đó xóa danh mục cha

DELETE FROM DanhMucSach

WHERE MaDanhMuc IN (SELECT MaDanhMuc FROM deleted);

END;

GO

-- 2. Tạo bảng sách

CREATE TABLE Sach (

MaSach VARCHAR(20) PRIMARY KEY,

ISBN VARCHAR(30) UNIQUE NOT NULL,

TenSach NVARCHAR(255) NOT NULL,

TacGia NVARCHAR(255) NOT NULL,

MaDanhMuc NVARCHAR(20) NOT NULL,

NamXuatBan INT NOT NULL CHECK (NamXuatBan > 0),

NXB NVARCHAR(255) NOT NULL,

SoBan INT NOT NULL CHECK (SoBan >= 0),

KhaDung INT NOT NULL CHECK (KhaDung >= 0),

ViTri NVARCHAR(255) NOT NULL,

CONSTRAINT fk\_Sach\_DanhMuc FOREIGN KEY (MaDanhMuc)

REFERENCES DanhMucSach(MaDanhMuc) ON DELETE CASCADE,

CONSTRAINT chk\_KhaDung\_SoBan CHECK (KhaDung <= SoBan)

);

GO

-- TRIGGER tự động cập nhật ngày khi danh mục được chỉnh sửa

CREATE TRIGGER trg\_CapNhatThoiGian\_DanhMucSach

ON DanhMucSach

AFTER UPDATE

AS

BEGIN

SET NOCOUNT ON;

UPDATE DanhMucSach

SET CapNhatLanCuoi = GETDATE()

WHERE MaDanhMuc IN (SELECT DISTINCT MaDanhMuc FROM inserted);

END;

GO

-- 3. Tạo bảng nhân viên

CREATE TABLE NhanVien (

ID VARCHAR(20) PRIMARY KEY,

HoTen NVARCHAR(100) NOT NULL,

GioiTinh NVARCHAR(10) CHECK (GioiTinh IN (N'Nam', N'Nữ')) NOT NULL,

ChucVu NVARCHAR(50) NOT NULL CHECK (ChucVu IN (N'Admin', N'Quản Lý', N'Nhân Viên')),

Email NVARCHAR(255) UNIQUE NOT NULL CHECK (Email LIKE '\_%@\_%.\_%'),

SoDienThoai VARCHAR(15) UNIQUE NOT NULL,

NgayVaoLam DATE NOT NULL CHECK (NgayVaoLam <= GETDATE()),

TrangThai NVARCHAR(20) NOT NULL CHECK (TrangThai IN (N'Đang làm', N'Tạm nghỉ')) DEFAULT N'Đang làm'

);

GO

-- 4. Tạo bảng Thành viên

CREATE TABLE ThanhVien (

MaThanhVien VARCHAR(10) PRIMARY KEY,

HoTen NVARCHAR(100) NOT NULL,

GioiTinh NVARCHAR(10) CHECK (GioiTinh IN (N'Nam', N'Nữ')) NOT NULL,

SoDienThoai VARCHAR(15) UNIQUE NOT NULL,

Email NVARCHAR(255) UNIQUE NOT NULL CHECK (Email LIKE '%\_@\_%.\_%'),

LoaiThanhVien NVARCHAR(50) NOT NULL CHECK (LoaiThanhVien IN (N'Sinh viên', N'Giảng viên', N'Thường')),

NgayDangKy DATE NOT NULL,

NgayHetHan DATE NOT NULL,

CHECK (NgayDangKy <= NgayHetHan),

TrangThai NVARCHAR(20) NOT NULL CHECK (TrangThai IN (N'Hoạt động', N'Hết hạn', N'Khóa'))

);

GO

-- 5. Tạo bảng phiếu mượn sách

CREATE TABLE PhieuMuon (

MaPhieu INT PRIMARY KEY IDENTITY(1,1),

MaThanhVien VARCHAR(10) NOT NULL,

NgayMuon DATE NOT NULL,

HanTra DATE NOT NULL,

NgayTraThucTe DATE NULL,

TrangThai NVARCHAR(50) CHECK (TrangThai IN (N'Đang mượn', N'Đã trả', N'Quá hạn')) DEFAULT N'Đang mượn',

MaSach VARCHAR(20) NOT NULL,

SoLuong INT NOT NULL CHECK (SoLuong > 0),

FOREIGN KEY (MaThanhVien) REFERENCES ThanhVien(MaThanhVien),

FOREIGN KEY (MaSach) REFERENCES Sach(MaSach)

);

GO

-- Thêm dữ liệu danh mục

INSERT INTO DanhMucSach (MaDanhMuc, TenDanhMuc, MoTa, DanhMucCha, SoLuongSach, NgayTao, CapNhatLanCuoi, TrangThai) VALUES

('CAT001', N'Văn học', N'Các tác phẩm văn học', NULL, 2100, '2023-01-01', '2024-03-05', N'Hoạt động'),

('CAT002', N'Giáo dục', N'Sách giáo dục và học tập', NULL, 1250, '2023-01-01', '2024-03-05', N'Hoạt động'),

('CAT003', N'Kinh tế & Kinh doanh', N'Sách về kinh tế và kinh doanh', NULL, 680, '2023-01-01', '2024-03-05', N'Hoạt động'),

('CAT004', N'Khoa học & Công nghệ', N'Sách khoa học và công nghệ', NULL, 780, '2023-01-01', '2024-03-05', N'Hoạt động'),

('CAT005', N'Phát triển bản thân', N'Sách phát triển kỹ năng cá nhân', NULL, 430, '2023-01-01', '2024-03-05', N'Hoạt động'),

('CAT101', N'Tiểu thuyết', N'Các tác phẩm văn học dài', 'CAT001', 1250, '2023-01-05', '2024-03-10', N'Hoạt động'),

('CAT102', N'Truyện ngắn', N'Các truyện ngắn và tuyển tập', 'CAT001', 430, '2023-01-05', '2024-03-10', N'Hoạt động'),

('CAT103', N'Thơ ca', N'Các tác phẩm thơ', 'CAT001', 210, '2023-01-05', '2024-03-10', N'Hoạt động'),

('CAT104', N'Truyện thiếu nhi', N'Sách dành cho thiếu nhi', 'CAT001', 210, '2023-01-05', '2024-03-10', N'Hoạt động'),

('CAT201', N'Sách giáo khoa', N'Sách học tập các cấp', 'CAT002', 850, '2023-01-08', '2024-03-15', N'Hoạt động'),

('CAT202', N'Sách tham khảo', N'Sách bổ trợ kiến thức', 'CAT002', 320, '2023-01-08', '2024-03-15', N'Hoạt động'),

('CAT203', N'Từ điển & Bách khoa', N'Sách tra cứu', 'CAT002', 80, '2023-01-08', '2024-03-15', N'Hoạt động'),

('CAT301', N'Quản trị kinh doanh', N'Sách về quản lý và điều hành', 'CAT003', 280, '2023-01-10', '2024-03-18', N'Hoạt động'),

('CAT302', N'Marketing & Bán hàng', N'Sách về tiếp thị và bán hàng', 'CAT003', 170, '2023-01-10', '2024-03-18', N'Hoạt động'),

('CAT303', N'Tài chính & Đầu tư', N'Sách về tài chính cá nhân', 'CAT003', 230, '2023-01-10', '2024-03-18', N'Hoạt động'),

('CAT401', N'Công nghệ thông tin', N'Sách về CNTT và lập trình', 'CAT004', 310, '2023-01-15', '2024-03-20', N'Hoạt động');

GO

-- Thêm dữ liệu sách

INSERT INTO Sach (MaSach, ISBN, TenSach, TacGia, MaDanhMuc, NamXuatBan, NXB, SoBan, KhaDung, ViTri) VALUES

('B1001', '9780747532743', N'Harry Potter và Hòn đá Phù thủy', N'J.K. Rowling', 'CAT101', 1997, N'Nhà xuất bản Trẻ', 10, 8, 'A-12-3'),

('B1002', '9780747538486', N'Harry Potter và Phòng chứa Bí mật', N'J.K. Rowling', 'CAT101', 1998, N'Nhà xuất bản Trẻ', 8, 5, 'A-12-4'),

('B1003', '9780747542155', N'Harry Potter và Tên tù nhân ngục Azkaban', N'J.K. Rowling', 'CAT101', 1999, N'Nhà xuất bản Trẻ', 7, 4, 'A-12-5'),

('B1004', '9780439139595', N'Harry Potter và Chiếc cốc lửa', N'J.K. Rowling', 'CAT101', 2000, N'Nhà xuất bản Trẻ', 10, 7, 'A-12-6'),

('B1005', '9780439358064', N'Harry Potter và Hội Phượng Hoàng', N'J.K. Rowling', 'CAT101', 2003, N'Nhà xuất bản Trẻ', 12, 9, 'A-12-7'),

('B1006', '9780439785969', N'Harry Potter và Hoàng tử lai', N'J.K. Rowling', 'CAT101', 2005, N'Nhà xuất bản Trẻ', 15, 11, 'A-12-8'),

('B1007', '9780545139700', N'Harry Potter và Bảo bối Tử thần', N'J.K. Rowling', 'CAT101', 2007, N'Nhà xuất bản Trẻ', 20, 15, 'A-12-9'),

('B1008', '9780590353427', N'Chú bé phù thủy', N'Roald Dahl', 'CAT104', 1983, N'Nhà xuất bản Kim Đồng', 5, 3, 'B-03-2'),

('B1009', '9780747546290', N'Matilda', N'Roald Dahl', 'CAT104', 1988, N'Nhà xuất bản Kim Đồng', 7, 5, 'B-03-3'),

('B1010', '9780140328721', N'Bí kíp làm giàu', N'Napoleon Hill', 'CAT303', 1937, N'Nhà xuất bản Lao động', 3, 2, 'C-05-1'),

('B1011', '9780062457714', N'Sức mạnh của thói quen', N'Charles Duhigg', 'CAT203', 2012, N'Nhà xuất bản Lao động', 4, 2, 'C-07-2'),

('B1012', '9786048412234', N'Đắc nhân tâm', N'Dale Carnegie', 'CAT005', 1936, N'Nhà xuất bản Tổng hợp TPHCM', 10, 8, 'D-01-1'),

('B1013', '9786045512838', N'Nhà giả kim', N'Paulo Coelho', 'CAT101', 1988, N'Nhà xuất bản Văn học', 15, 12, 'A-05-2'),

('B1014', '9780671027032', N'7 Thói quen của người thành đạt', N'Stephen R. Covey', 'CAT005', 1989, N'Nhà xuất bản Trẻ', 6, 4, 'D-01-2'),

('B1015', '9780007442911', N'Đi tìm lẽ sống', N'Viktor E. Frankl', 'CAT203', 1946, N'Nhà xuất bản Trẻ', 5, 3, 'C-07-3');

GO

-- thêm dữ liệu vào bảng Nhân Viên

INSERT INTO NhanVien (ID, HoTen, GioiTinh, ChucVu, Email, SoDienThoai, NgayVaoLam, TrangThai) VALUES

('NV001', N'Nguyễn Văn Hòa', N'Nam', N'Admin', 'nguyenhoa@gmail.com', '0901123456', '2023-01-10', N'Đang làm'),

('NV002', N'Trần Thị Mai', N'Nữ', N'Quản Lý', 'tranmaiqly@gmail.com', '0912234567', '2022-03-15', N'Đang làm'),

('NV003', N'Lê Minh Tuấn', N'Nam', N'Nhân Viên', 'letuan\_nv@gmail.com', '0923345678', '2021-07-20', N'Đang làm'),

('NV004', N'Hoàng Đức Anh', N'Nam', N'Nhân Viên', 'hoangduca@gmail.com', '0934456789', '2020-10-05', N'Tạm nghỉ'),

('NV005', N'Vũ Thị Hồng', N'Nữ', N'Nhân Viên', 'vuthihong@gmail.com', '0945567890', '2023-05-12', N'Đang làm'),

('NV006', N'Phạm Quốc Bảo', N'Nam', N'Nhân Viên', 'phamquocbao@gmail.com', '0956678901', '2021-08-17', N'Tạm nghỉ'),

('NV007', N'Đặng Thúy Hằng', N'Nữ', N'Nhân Viên', 'dangthuyhang@gmail.com', '0967789012', '2022-02-10', N'Đang làm'),

('NV008', N'Bùi Văn Khánh', N'Nam', N'Nhân Viên', 'buivankhanh@gmail.com', '0978890123', '2020-11-25', N'Đang làm'),

('NV009', N'Ngô Thị Hạnh', N'Nữ', N'Nhân Viên', 'ngothihanh@gmail.com', '0989901234', '2019-06-30', N'Tạm nghỉ'),

('NV010', N'Đỗ Hoàng Sơn', N'Nam', N'Nhân Viên', 'dohoangson@gmail.com', '0991012345', '2023-09-05', N'Đang làm');

GO

-- Trigger để tự động cập nhật trạng thái thành viên khi hết hạn

CREATE TRIGGER trg\_CapNhatTrangThaiThanhVien

ON ThanhVien

AFTER INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

-- Cập nhật trạng thái thành "Hết hạn" cho các thành viên có ngày hết hạn đã qua

UPDATE ThanhVien

SET TrangThai = N'Hết hạn'

WHERE MaThanhVien IN (SELECT MaThanhVien FROM inserted)

AND NgayHetHan < GETDATE()

AND TrangThai = N'Hoạt động';

END;

GO

-- Thêm dữ liệu thành viên

INSERT INTO ThanhVien (MaThanhVien, HoTen, GioiTinh, SoDienThoai, Email, LoaiThanhVien, NgayDangKy, NgayHetHan, TrangThai) VALUES

('M0001', N'Nguyễn Văn An', N'Nam', '0987654321', 'nguyenvana@gmail.com', N'Sinh viên', '2023-01-01', '2025-01-01', N'Hoạt động'),

('M0002', N'Trần Thị Bích Ngọc', N'Nữ', '0912345678', 'bichngoc@gmail.com', N'Sinh viên', '2023-01-15', '2025-01-15', N'Hoạt động'),

('M0003', N'Lê Hoàng Nam', N'Nam', '0905678123', 'lehoangnam@gmail.com', N'Giảng viên', '2023-02-10', '2024-02-10', N'Hoạt động'),

('M0004', N'Phạm Thu Hương', N'Nữ', '0977234567', 'huongpham@gmail.com', N'Sinh viên', '2023-03-05', '2024-03-05', N'Hoạt động'),

('M0005', N'Hoàng Thị Lan', N'Nữ', '0921456789', 'hoanglan@gmail.com', N'Thường', '2023-03-20', '2025-03-20', N'Hoạt động'),

('M0006', N'Vũ Đức Thành', N'Nam', '0968234890', 'ducthanh123@gmail.com', N'Sinh viên', '2023-04-10', '2025-04-10', N'Hoạt động'),

('M0007', N'Bùi Thanh Mai', N'Nữ', '0945678234', 'thanhmai\_bui@gmail.com', N'Giảng viên', '2023-05-15', '2025-05-15', N'Hoạt động'),

('M0008', N'Đỗ Quang Huy', N'Nam', '0982345678', 'huydo@gmail.com', N'Sinh viên', '2023-06-05', '2024-06-05', N'Hoạt động'),

('M0009', N'Nguyễn Thị Kim Anh', N'Nữ', '0909123456', 'kimanh99@gmail.com', N'Thường', '2023-06-20', '2025-06-20', N'Hết hạn'),

('M0010', N'Lý Văn Duy', N'Nam', '0915678901', 'lyvduy@gmail.com', N'Sinh viên', '2023-07-10', '2024-07-10', N'Hoạt động'),

('M0011', N'Nguyễn Minh Tú', N'Nam', '0911123456', 'minhtu@gmail.com', N'Sinh viên', '2023-08-10', '2024-08-10', N'Hoạt động'),

('M0012', N'Phạm Hoàng Yến', N'Nữ', '0988112233', 'hoangyen\_pham@gmail.com', N'Sinh viên', '2023-09-05', '2025-09-05', N'Hoạt động'),

('M0013', N'Bùi Hữu Nghĩa', N'Nam', '0977556677', 'huunghia.bui@gmail.com', N'Giảng viên', '2023-10-15', '2025-10-15', N'Hoạt động'),

('M0014', N'Lê Hải Đăng', N'Nam', '0933445566', 'haidang\_le@gmail.com', N'Thường', '2023-11-20', '2025-11-20', N'Hoạt động'),

('M0015', N'Trần Thu Trang', N'Nữ', '0966778899', 'trangtran@gmail.com', N'Sinh viên', '2023-12-01', '2024-12-01', N'Hoạt động');

GO

-- Thêm dữ liệu phiếu mượn sách

-- Thêm trigger để tự động cập nhật trạng thái "Quá hạn" cho phiếu mượn

CREATE TRIGGER trg\_CapNhatTrangThaiQuaHan

ON PhieuMuon

AFTER INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

-- Cập nhật trạng thái thành "Quá hạn" nếu đã quá hạn trả

UPDATE PhieuMuon

SET TrangThai = N'Quá hạn'

WHERE MaPhieu IN (SELECT MaPhieu FROM inserted)

AND HanTra < GETDATE()

AND TrangThai = N'Đang mượn';

END;

GO

INSERT INTO PhieuMuon (MaThanhVien, NgayMuon, HanTra, NgayTraThucTe, TrangThai, MaSach, SoLuong) VALUES

('M0001', '2025-03-01', '2025-03-15', NULL, N'Đang mượn', 'B1001', 1),

('M0002', '2025-02-20', '2025-03-10', '2025-03-09', N'Đã trả', 'B1003', 1),

('M0003', '2025-03-05', '2025-03-19', NULL, N'Đang mượn', 'B1005', 2),

('M0004', '2025-03-08', '2025-03-22', NULL, N'Đang mượn', 'B1012', 1),

('M0005', '2025-02-28', '2025-03-14', NULL, N'Quá hạn', 'B1010', 1),

('M0006', '2025-02-25', '2025-03-11', '2025-03-10', N'Đã trả', 'B1007', 1),

('M0007', '2025-03-10', '2025-03-24', NULL, N'Đang mượn', 'B1009', 2),

('M0008', '2025-02-15', '2025-03-01', NULL, N'Quá hạn', 'B1011', 1),

('M0009', '2025-03-01', '2025-03-15', NULL, N'Đang mượn', 'B1004', 1),

('M0010', '2025-02-27', '2025-03-13', '2025-03-15', N'Đã trả', 'B1006', 1), -- Trả trễ 2 ngày

('M0011', '2025-03-02', '2025-03-16', NULL, N'Đang mượn', 'B1013', 1),

('M0012', '2025-03-06', '2025-03-20', NULL, N'Đang mượn', 'B1002', 1),

('M0013', '2025-03-09', '2025-03-23', NULL, N'Đang mượn', 'B1008', 1),

('M0014', '2025-02-22', '2025-03-08', '2025-03-12', N'Đã trả', 'B1014', 1), -- Trả trễ 4 ngày

('M0015', '2025-03-04', '2025-03-18', NULL, N'Đang mượn', 'B1015', 1);

GO

-- Tạo các stored procedure

CREATE PROCEDURE [dbo].[sp\_ThemSach]

@MaSach VARCHAR(20),

@ISBN VARCHAR(30),

@TenSach NVARCHAR(255),

@TacGia NVARCHAR(255),

@MaDanhMuc NVARCHAR(20),

@NamXuatBan INT,

@NXB NVARCHAR(255),

@SoBan INT,

@ViTri NVARCHAR(255)

AS

BEGIN

SET NOCOUNT ON;

-- Kiểm tra xem mã sách hoặc ISBN đã tồn tại chưa

IF EXISTS (SELECT 1 FROM Sach WHERE MaSach = @MaSach OR ISBN = @ISBN)

BEGIN

RAISERROR(N'Mã sách hoặc ISBN đã tồn tại!', 16, 1);

RETURN;

END

-- Kiểm tra danh mục sách có tồn tại không

IF NOT EXISTS (SELECT 1 FROM DanhMucSach WHERE MaDanhMuc = @MaDanhMuc)

BEGIN

RAISERROR(N'Mã danh mục không tồn tại!', 16, 1);

RETURN;

END

-- Kiểm tra năm xuất bản hợp lệ

IF @NamXuatBan <= 0 OR @NamXuatBan > YEAR(GETDATE())

BEGIN

RAISERROR(N'Năm xuất bản không hợp lệ!', 16, 1);

RETURN;

END

-- Kiểm tra số bản sách phải lớn hơn hoặc bằng 0

IF @SoBan < 0

BEGIN

RAISERROR(N'Số bản sách không được âm!', 16, 1);

RETURN;

END

BEGIN TRANSACTION;

-- Thêm sách mới vào hệ thống

INSERT INTO Sach (MaSach, ISBN, TenSach, TacGia, MaDanhMuc, NamXuatBan, NXB, SoBan, KhaDung, ViTri)

VALUES (@MaSach, @ISBN, @TenSach, @TacGia, @MaDanhMuc, @NamXuatBan, @NXB, @SoBan, @SoBan, @ViTri);

-- Cập nhật số lượng sách trong danh mục

UPDATE DanhMucSach

SET SoLuongSach = SoLuongSach + @SoBan

WHERE MaDanhMuc = @MaDanhMuc;

COMMIT;

PRINT N'Thêm sách thành công!';

END;

GO

CREATE PROCEDURE [dbo].[sp\_SuaSach]

@MaSach VARCHAR(20),

@ISBN VARCHAR(30),

@TenSach NVARCHAR(255),

@TacGia NVARCHAR(255),

@MaDanhMuc NVARCHAR(20),

@NamXuatBan INT,

@NXB NVARCHAR(255),

@SoBan INT,

@KhaDung INT,

@ViTri NVARCHAR(255)

AS

BEGIN

SET NOCOUNT ON;

-- Kiểm tra xem sách có tồn tại không

IF NOT EXISTS (SELECT 1 FROM Sach WHERE MaSach = @MaSach)

BEGIN

PRINT N'Mã sách không tồn tại!';

RETURN;

END

-- Kiểm tra số bản khả dụng không vượt quá tổng số bản

IF @KhaDung > @SoBan

BEGIN

PRINT N'Số bản khả dụng không thể lớn hơn tổng số bản!';

RETURN;

END

-- Cập nhật thông tin sách

UPDATE Sach

SET

ISBN = @ISBN,

TenSach = @TenSach,

TacGia = @TacGia,

MaDanhMuc = @MaDanhMuc,

NamXuatBan = @NamXuatBan,

NXB = @NXB,

SoBan = @SoBan,

KhaDung = @KhaDung,

ViTri = @ViTri

WHERE MaSach = @MaSach;

PRINT N'Cập nhật sách thành công!';

END;

GO

CREATE PROCEDURE [dbo].[sp\_XoaSach]

@MaSach VARCHAR(20)

AS

BEGIN

SET NOCOUNT ON;

-- Kiểm tra xem sách có tồn tại không

IF NOT EXISTS (SELECT 1 FROM Sach WHERE MaSach = @MaSach)

BEGIN

PRINT N'Mã sách không tồn tại!';

RETURN;

END

-- Xóa sách

DELETE FROM Sach WHERE MaSach = @MaSach;

PRINT N'Xóa sách thành công!';

END;

GO

CREATE PROCEDURE sp\_ThemDanhMucSach

@MaDanhMuc NVARCHAR(20),

@TenDanhMuc NVARCHAR(255),

@MoTa NVARCHAR(500) = NULL,

@DanhMucCha NVARCHAR(20) = NULL,

@SoLuongSach INT = 0,

@TrangThai NVARCHAR(20) = N'Hoạt động'

AS

BEGIN

SET NOCOUNT ON;

-- Kiểm tra xem MaDanhMuc đã tồn tại chưa

IF EXISTS (SELECT 1 FROM DanhMucSach WHERE MaDanhMuc = @MaDanhMuc)

BEGIN

RAISERROR(N'Mã danh mục đã tồn tại!', 16, 1);

RETURN;

END

-- Kiểm tra DanhMucCha có tồn tại hay không

IF @DanhMucCha IS NOT NULL AND NOT EXISTS (SELECT 1 FROM DanhMucSach WHERE MaDanhMuc = @DanhMucCha)

BEGIN

RAISERROR(N'Danh mục cha không tồn tại!', 16, 1);

RETURN;

END

-- Chèn dữ liệu vào bảng DanhMucSach

INSERT INTO DanhMucSach (MaDanhMuc, TenDanhMuc, MoTa, DanhMucCha, SoLuongSach, NgayTao, CapNhatLanCuoi, TrangThai)

VALUES (@MaDanhMuc, @TenDanhMuc, @MoTa, @DanhMucCha, @SoLuongSach, GETDATE(), GETDATE(), @TrangThai);

PRINT N'Thêm danh mục sách thành công!';

END;

GO

CREATE PROCEDURE sp\_SuaDanhMucSach

@MaDanhMuc NVARCHAR(20),

@TenDanhMuc NVARCHAR(255) = NULL,

@MoTa NVARCHAR(500) = NULL,

@DanhMucCha NVARCHAR(20) = NULL,

@SoLuongSach INT = NULL,

@TrangThai NVARCHAR(20) = NULL

AS

BEGIN

SET NOCOUNT ON;

-- Kiểm tra xem danh mục có tồn tại không

IF NOT EXISTS (SELECT 1 FROM DanhMucSach WHERE MaDanhMuc = @MaDanhMuc)

BEGIN

RAISERROR(N'Mã danh mục không tồn tại!', 16, 1);

RETURN;

END

-- Kiểm tra DanhMucCha có tồn tại hay không (nếu có thay đổi)

IF @DanhMucCha IS NOT NULL AND NOT EXISTS (SELECT 1 FROM DanhMucSach WHERE MaDanhMuc = @DanhMucCha)

BEGIN

RAISERROR(N'Danh mục cha không tồn tại!', 16, 1);

RETURN;

END

-- Cập nhật danh mục

UPDATE DanhMucSach

SET

TenDanhMuc = ISNULL(@TenDanhMuc, TenDanhMuc),

MoTa = ISNULL(@MoTa, MoTa),

DanhMucCha = ISNULL(@DanhMucCha, DanhMucCha),

SoLuongSach = ISNULL(@SoLuongSach, SoLuongSach),

CapNhatLanCuoi = GETDATE(),

TrangThai = ISNULL(@TrangThai, TrangThai)

WHERE MaDanhMuc = @MaDanhMuc;

PRINT N'Cập nhật danh mục sách thành công!';

END;

GO

CREATE PROCEDURE sp\_XoaDanhMucSach

@MaDanhMuc NVARCHAR(20)

AS

BEGIN

SET NOCOUNT ON;

-- Kiểm tra xem danh mục có tồn tại không

IF NOT EXISTS (SELECT 1 FROM DanhMucSach WHERE MaDanhMuc = @MaDanhMuc)

BEGIN

RAISERROR(N'Mã danh mục không tồn tại!', 16, 1);

RETURN;

END

-- Kiểm tra xem danh mục có danh mục con không

IF EXISTS (SELECT 1 FROM DanhMucSach WHERE DanhMucCha = @MaDanhMuc)

BEGIN

RAISERROR(N'Không thể xóa danh mục vì có danh mục con!', 16, 1);

RETURN;

END

-- Xóa danh mục

DELETE FROM DanhMucSach WHERE MaDanhMuc = @MaDanhMuc;

PRINT N'Xóa danh mục sách thành công!';

END;

GO

CREATE PROCEDURE sp\_ThemThanhVien

@MaThanhVien VARCHAR(10),

@HoTen NVARCHAR(100),

@GioiTinh NVARCHAR(10),

@SoDienThoai VARCHAR(15),

@Email NVARCHAR(255),

@LoaiThanhVien NVARCHAR(50),

@NgayDangKy DATE = NULL,

@NgayHetHan DATE = NULL,

@TrangThai NVARCHAR(20) = N'Hoạt động'

AS

BEGIN

SET NOCOUNT ON;

-- Nếu không cung cấp ngày đăng ký, lấy ngày hiện tại

SET @NgayDangKy = ISNULL(@NgayDangKy, GETDATE());

-- Nếu không cung cấp ngày hết hạn, mặc định là 2 năm sau ngày đăng ký

SET @NgayHetHan = ISNULL(@NgayHetHan, DATEADD(YEAR, 2, @NgayDangKy));

-- Kiểm tra mã thành viên đã tồn tại chưa

IF EXISTS (SELECT 1 FROM ThanhVien WHERE MaThanhVien = @MaThanhVien)

BEGIN

RAISERROR(N'Mã thành viên đã tồn tại!', 16, 1);

RETURN;

END

-- Kiểm tra số điện thoại đã tồn tại chưa

IF EXISTS (SELECT 1 FROM ThanhVien WHERE SoDienThoai = @SoDienThoai)

BEGIN

RAISERROR(N'Số điện thoại đã tồn tại trong hệ thống!', 16, 1);

RETURN;

END

-- Kiểm tra email đã tồn tại chưa

IF EXISTS (SELECT 1 FROM ThanhVien WHERE Email = @Email)

BEGIN

RAISERROR(N'Email đã tồn tại trong hệ thống!', 16, 1);

RETURN;

END

-- Kiểm tra giới tính hợp lệ

IF @GioiTinh NOT IN (N'Nam', N'Nữ')

BEGIN

RAISERROR(N'Giới tính phải là Nam hoặc Nữ!', 16, 1);

RETURN;

END

-- Kiểm tra loại thành viên hợp lệ

IF @LoaiThanhVien NOT IN (N'Sinh viên', N'Giảng viên', N'Thường')

BEGIN

RAISERROR(N'Loại thành viên phải là Sinh viên, Giảng viên hoặc Thường!', 16, 1);

RETURN;

END

-- Kiểm tra trạng thái hợp lệ

IF @TrangThai NOT IN (N'Hoạt động', N'Hết hạn', N'Khóa')

BEGIN

RAISERROR(N'Trạng thái phải là Hoạt động, Hết hạn hoặc Khóa!', 16, 1);

RETURN;

END

-- Thêm thành viên mới

INSERT INTO ThanhVien (MaThanhVien, HoTen, GioiTinh, SoDienThoai, Email, LoaiThanhVien, NgayDangKy, NgayHetHan, TrangThai)

VALUES (@MaThanhVien, @HoTen, @GioiTinh, @SoDienThoai, @Email, @LoaiThanhVien, @NgayDangKy, @NgayHetHan, @TrangThai);

PRINT N'Thêm thành viên thành công!';

END;

GO

-- Stored procedure để sửa thông tin thành viên

CREATE PROCEDURE sp\_SuaThanhVien

@MaThanhVien VARCHAR(10),

@HoTen NVARCHAR(100) = NULL,

@GioiTinh NVARCHAR(10) = NULL,

@SoDienThoai VARCHAR(15) = NULL,

@Email NVARCHAR(255) = NULL,

@LoaiThanhVien NVARCHAR(50) = NULL,

@NgayDangKy DATE = NULL,

@NgayHetHan DATE = NULL,

@TrangThai NVARCHAR(20) = NULL

AS

BEGIN

SET NOCOUNT ON;

-- Kiểm tra mã thành viên có tồn tại không

IF NOT EXISTS (SELECT 1 FROM ThanhVien WHERE MaThanhVien = @MaThanhVien)

BEGIN

RAISERROR(N'Mã thành viên không tồn tại!', 16, 1);

RETURN;

END

-- Lấy thông tin hiện tại của thành viên

DECLARE @CurrentHoTen NVARCHAR(100)

DECLARE @CurrentGioiTinh NVARCHAR(10)

DECLARE @CurrentSoDienThoai VARCHAR(15)

DECLARE @CurrentEmail NVARCHAR(255)

DECLARE @CurrentLoaiThanhVien NVARCHAR(50)

DECLARE @CurrentNgayDangKy DATE

DECLARE @CurrentNgayHetHan DATE

DECLARE @CurrentTrangThai NVARCHAR(20)

SELECT @CurrentHoTen = HoTen,

@CurrentGioiTinh = GioiTinh,

@CurrentSoDienThoai = SoDienThoai,

@CurrentEmail = Email,

@CurrentLoaiThanhVien = LoaiThanhVien,

@CurrentNgayDangKy = NgayDangKy,

@CurrentNgayHetHan = NgayHetHan,

@CurrentTrangThai = TrangThai

FROM ThanhVien

WHERE MaThanhVien = @MaThanhVien;

-- Kiểm tra số điện thoại mới đã tồn tại chưa (nếu có thay đổi)

IF @SoDienThoai IS NOT NULL AND @SoDienThoai <> @CurrentSoDienThoai AND

EXISTS (SELECT 1 FROM ThanhVien WHERE SoDienThoai = @SoDienThoai)

BEGIN

RAISERROR(N'Số điện thoại đã tồn tại trong hệ thống!', 16, 1);

RETURN;

END

-- Kiểm tra email mới đã tồn tại chưa (nếu có thay đổi)

IF @Email IS NOT NULL AND @Email <> @CurrentEmail AND

EXISTS (SELECT 1 FROM ThanhVien WHERE Email = @Email)

BEGIN

RAISERROR(N'Email đã tồn tại trong hệ thống!', 16, 1);

RETURN;

END

-- Kiểm tra giới tính hợp lệ (nếu có thay đổi)

IF @GioiTinh IS NOT NULL AND @GioiTinh NOT IN (N'Nam', N'Nữ')

BEGIN

RAISERROR(N'Giới tính phải là Nam hoặc Nữ!', 16, 1);

RETURN;

END

-- Kiểm tra loại thành viên hợp lệ (nếu có thay đổi)

IF @LoaiThanhVien IS NOT NULL AND @LoaiThanhVien NOT IN (N'Sinh viên', N'Giảng viên', N'Thường')

BEGIN

RAISERROR(N'Loại thành viên phải là Sinh viên, Giảng viên hoặc Thường!', 16, 1);

RETURN;

END

-- Kiểm tra trạng thái hợp lệ (nếu có thay đổi)

IF @TrangThai IS NOT NULL AND @TrangThai NOT IN (N'Hoạt động', N'Hết hạn', N'Khóa')

BEGIN

RAISERROR(N'Trạng thái phải là Hoạt động, Hết hạn hoặc Khóa!', 16, 1);

RETURN;

END

-- Cập nhật thông tin thành viên

UPDATE ThanhVien

SET HoTen = ISNULL(@HoTen, @CurrentHoTen),

GioiTinh = ISNULL(@GioiTinh, @CurrentGioiTinh),

SoDienThoai = ISNULL(@SoDienThoai, @CurrentSoDienThoai),

Email = ISNULL(@Email, @CurrentEmail),

LoaiThanhVien = ISNULL(@LoaiThanhVien, @CurrentLoaiThanhVien),

NgayDangKy = ISNULL(@NgayDangKy, @CurrentNgayDangKy),

NgayHetHan = ISNULL(@NgayHetHan, @CurrentNgayHetHan),

TrangThai = ISNULL(@TrangThai, @CurrentTrangThai)

WHERE MaThanhVien = @MaThanhVien;

PRINT N'Cập nhật thành viên thành công!';

END;

GO

-- Stored procedure để xóa thành viên

CREATE PROCEDURE sp\_XoaThanhVien

@MaThanhVien VARCHAR(10)

AS

BEGIN

SET NOCOUNT ON;

-- Kiểm tra mã thành viên có tồn tại không

IF NOT EXISTS (SELECT 1 FROM ThanhVien WHERE MaThanhVien = @MaThanhVien)

BEGIN

RAISERROR(N'Mã thành viên không tồn tại!', 16, 1);

RETURN;

END

-- Kiểm tra thành viên có đang mượn sách không

IF EXISTS (SELECT 1 FROM PhieuMuon WHERE MaThanhVien = @MaThanhVien AND TrangThai = N'Đang mượn')

BEGIN

RAISERROR(N'Không thể xóa thành viên này vì họ đang mượn sách!', 16, 1);

RETURN;

END

-- Xóa thành viên

DELETE FROM ThanhVien WHERE MaThanhVien = @MaThanhVien;

PRINT N'Xóa thành viên thành công!';

END;

GO

-- Stored procedure để thêm nhân viên

CREATE PROCEDURE sp\_ThemNhanVien

@ID VARCHAR(20),

@HoTen NVARCHAR(100),

@GioiTinh NVARCHAR(10),

@ChucVu NVARCHAR(50),

@Email NVARCHAR(255),

@SoDienThoai VARCHAR(15),

@NgayVaoLam DATE = NULL,

@TrangThai NVARCHAR(20) = N'Đang làm'

AS

BEGIN

SET NOCOUNT ON;

-- Nếu không cung cấp ngày vào làm, lấy ngày hiện tại

SET @NgayVaoLam = ISNULL(@NgayVaoLam, GETDATE());

-- Kiểm tra ID đã tồn tại chưa

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

BEGIN

RAISERROR(N'ID nhân viên đã tồn tại!', 16, 1);

RETURN;

END

-- Kiểm tra số điện thoại đã tồn tại chưa

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

BEGIN

RAISERROR(N'Số điện thoại đã tồn tại trong hệ thống!', 16, 1);

RETURN;

END

-- Kiểm tra email đã tồn tại chưa

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

BEGIN

RAISERROR(N'Email đã tồn tại trong hệ thống!', 16, 1);

RETURN;

END

-- Kiểm tra giới tính hợp lệ

IF @GioiTinh NOT IN (N'Nam', N'Nữ')

BEGIN

RAISERROR(N'Giới tính phải là Nam hoặc Nữ!', 16, 1);

RETURN;

END

-- Kiểm tra chức vụ hợp lệ

IF @ChucVu NOT IN (N'Admin', N'Quản Lý', N'Nhân Viên')

BEGIN

RAISERROR(N'Chức vụ phải là Admin, Quản Lý hoặc Nhân Viên!', 16, 1);

RETURN;

END

-- Kiểm tra trạng thái hợp lệ

IF @TrangThai NOT IN (N'Đang làm', N'Tạm nghỉ')

BEGIN

RAISERROR(N'Trạng thái phải là Đang làm hoặc Tạm nghỉ!', 16, 1);

RETURN;

END

-- Thêm nhân viên mới

INSERT INTO NhanVien (ID, HoTen, GioiTinh, ChucVu, Email, SoDienThoai, NgayVaoLam, TrangThai)

VALUES (@ID, @HoTen, @GioiTinh, @ChucVu, @Email, @SoDienThoai, @NgayVaoLam, @TrangThai);

PRINT N'Thêm nhân viên thành công!';

END;

GO

-- Stored procedure để sửa thông tin nhân viên

CREATE PROCEDURE sp\_SuaNhanVien

@ID VARCHAR(20),

@HoTen NVARCHAR(100) = NULL,

@GioiTinh NVARCHAR(10) = NULL,

@ChucVu NVARCHAR(50) = NULL,

@Email NVARCHAR(255) = NULL,

@SoDienThoai VARCHAR(15) = NULL,

@NgayVaoLam DATE = NULL,

@TrangThai NVARCHAR(20) = NULL

AS

BEGIN

SET NOCOUNT ON;

-- Kiểm tra ID có tồn tại không

IF NOT EXISTS (SELECT 1 FROM NhanVien WHERE ID = @ID)

BEGIN

RAISERROR(N'ID nhân viên không tồn tại!', 16, 1);

RETURN;

END

-- Lấy thông tin hiện tại của nhân viên

DECLARE @CurrentHoTen NVARCHAR(100)

DECLARE @CurrentGioiTinh NVARCHAR(10)

DECLARE @CurrentChucVu NVARCHAR(50)

DECLARE @CurrentEmail NVARCHAR(255)

DECLARE @CurrentSoDienThoai VARCHAR(15)

DECLARE @CurrentNgayVaoLam DATE

DECLARE @CurrentTrangThai NVARCHAR(20)

SELECT @CurrentHoTen = HoTen,

@CurrentGioiTinh = GioiTinh,

@CurrentChucVu = ChucVu,

@CurrentEmail = Email,

@CurrentSoDienThoai = SoDienThoai,

@CurrentNgayVaoLam = NgayVaoLam,

@CurrentTrangThai = TrangThai

FROM NhanVien

WHERE ID = @ID;

-- Kiểm tra số điện thoại mới đã tồn tại chưa (nếu có thay đổi)

IF @SoDienThoai IS NOT NULL AND @SoDienThoai <> @CurrentSoDienThoai AND

EXISTS (SELECT 1 FROM NhanVien WHERE SoDienThoai = @SoDienThoai)

BEGIN

RAISERROR(N'Số điện thoại đã tồn tại trong hệ thống!', 16, 1);

RETURN;

END

-- Kiểm tra email mới đã tồn tại chưa (nếu có thay đổi)

IF @Email IS NOT NULL AND @Email <> @CurrentEmail AND

EXISTS (SELECT 1 FROM NhanVien WHERE Email = @Email)

BEGIN

RAISERROR(N'Email đã tồn tại trong hệ thống!', 16, 1);

RETURN;

END

-- Kiểm tra giới tính hợp lệ (nếu có thay đổi)

IF @GioiTinh IS NOT NULL AND @GioiTinh NOT IN (N'Nam', N'Nữ')

BEGIN

RAISERROR(N'Giới tính phải là Nam hoặc Nữ!', 16, 1);

RETURN;

END

-- Kiểm tra chức vụ hợp lệ (nếu có thay đổi)

IF @ChucVu IS NOT NULL AND @ChucVu NOT IN (N'Admin', N'Quản Lý', N'Nhân Viên')

BEGIN

RAISERROR(N'Chức vụ phải là Admin, Quản Lý hoặc Nhân Viên!', 16, 1);

RETURN;

END

-- Kiểm tra trạng thái hợp lệ (nếu có thay đổi)

IF @TrangThai IS NOT NULL AND @TrangThai NOT IN (N'Đang làm', N'Tạm nghỉ')

BEGIN

RAISERROR(N'Trạng thái phải là Đang làm hoặc Tạm nghỉ!', 16, 1);

RETURN;

END

-- Cập nhật thông tin nhân viên

UPDATE NhanVien

SET HoTen = ISNULL(@HoTen, @CurrentHoTen),

GioiTinh = ISNULL(@GioiTinh, @CurrentGioiTinh),

ChucVu = ISNULL(@ChucVu, @CurrentChucVu),

Email = ISNULL(@Email, @CurrentEmail),

SoDienThoai = ISNULL(@SoDienThoai, @CurrentSoDienThoai),

NgayVaoLam = ISNULL(@NgayVaoLam, @CurrentNgayVaoLam),

TrangThai = ISNULL(@TrangThai, @CurrentTrangThai)

WHERE ID = @ID;

PRINT N'Cập nhật thông tin nhân viên thành công!';

END;

GO

-- Stored procedure để xóa nhân viên

CREATE PROCEDURE sp\_XoaNhanVien

@ID VARCHAR(20)

AS

BEGIN

SET NOCOUNT ON;

-- Kiểm tra ID có tồn tại không

IF NOT EXISTS (SELECT 1 FROM NhanVien WHERE ID = @ID)

BEGIN

RAISERROR(N'ID nhân viên không tồn tại!', 16, 1);

RETURN;

END

-- Kiểm tra Admin không thể xóa

IF EXISTS (SELECT 1 FROM NhanVien WHERE ID = @ID AND ChucVu = N'Admin')

BEGIN

RAISERROR(N'Không thể xóa tài khoản Admin!', 16, 1);

RETURN;

END

-- Xóa nhân viên

DELETE FROM NhanVien WHERE ID = @ID;

PRINT N'Xóa nhân viên thành công!';

END;

GO

-- Stored procedure để tạo phiếu mượn sách

CREATE PROCEDURE sp\_MuonSach

@MaThanhVien VARCHAR(10),

@MaSach VARCHAR(20),

@SoLuong INT,

@NgayMuon DATE = NULL,

@HanTra DATE = NULL

AS

BEGIN

SET NOCOUNT ON;

-- Nếu không cung cấp ngày mượn, lấy ngày hiện tại

SET @NgayMuon = ISNULL(@NgayMuon, GETDATE());

-- Nếu không cung cấp hạn trả, mặc định là 14 ngày sau ngày mượn

SET @HanTra = ISNULL(@HanTra, DATEADD(DAY, 14, @NgayMuon));

-- Kiểm tra thành viên có tồn tại không

IF NOT EXISTS (SELECT 1 FROM ThanhVien WHERE MaThanhVien = @MaThanhVien)

BEGIN

RAISERROR(N'Mã thành viên không tồn tại!', 16, 1);

RETURN;

END

-- Kiểm tra sách có tồn tại không

IF NOT EXISTS (SELECT 1 FROM Sach WHERE MaSach = @MaSach)

BEGIN

RAISERROR(N'Mã sách không tồn tại!', 16, 1);

RETURN;

END

-- Kiểm tra thành viên có bị khóa hoặc hết hạn không

IF EXISTS (SELECT 1 FROM ThanhVien WHERE MaThanhVien = @MaThanhVien AND TrangThai IN (N'Khóa', N'Hết hạn'))

BEGIN

RAISERROR(N'Thành viên đã bị khóa hoặc tài khoản đã hết hạn!', 16, 1);

RETURN;

END

-- Kiểm tra số lượng sách còn đủ không

DECLARE @SoLuongKhaDung INT;

SELECT @SoLuongKhaDung = KhaDung FROM Sach WHERE MaSach = @MaSach;

IF @SoLuongKhaDung < @SoLuong

BEGIN

RAISERROR(N'Số lượng sách không đủ để mượn! Hiện có %d cuốn khả dụng.', 16, 1, @SoLuongKhaDung);

RETURN;

END

-- Kiểm tra số lượng sách thành viên đã mượn (giới hạn mỗi thành viên mượn tối đa 5 sách)

DECLARE @SoSachDangMuon INT;

SELECT @SoSachDangMuon = ISNULL(SUM(SoLuong), 0)

FROM PhieuMuon

WHERE MaThanhVien = @MaThanhVien AND TrangThai = N'Đang mượn';

IF (@SoSachDangMuon + @SoLuong) > 5

BEGIN

RAISERROR(N'Thành viên đã mượn %d cuốn, không thể mượn thêm %d cuốn nữa! Tối đa là 5 cuốn.', 16, 1, @SoSachDangMuon, @SoLuong);

RETURN;

END

-- Kiểm tra thành viên có phiếu mượn quá hạn không

IF EXISTS (SELECT 1 FROM PhieuMuon WHERE MaThanhVien = @MaThanhVien AND TrangThai = N'Quá hạn')

BEGIN

RAISERROR(N'Thành viên có sách mượn quá hạn chưa trả, không thể mượn thêm!', 16, 1);

RETURN;

END

BEGIN TRANSACTION;

-- Tạo phiếu mượn

INSERT INTO PhieuMuon (MaThanhVien, NgayMuon, HanTra, TrangThai, MaSach, SoLuong)

VALUES (@MaThanhVien, @NgayMuon, @HanTra, N'Đang mượn', @MaSach, @SoLuong);

-- Cập nhật số lượng sách khả dụng

UPDATE Sach

SET KhaDung = KhaDung - @SoLuong

WHERE MaSach = @MaSach;

COMMIT;

PRINT N'Tạo phiếu mượn thành công!';

END;

GO

-- Tạo stored procedure trả sách

CREATE PROCEDURE [dbo].[sp\_TraSach]

@MaPhieu INT,

@NgayTra DATE

AS

BEGIN

BEGIN TRY

BEGIN TRANSACTION

-- Kiểm tra phiếu mượn có tồn tại và đang ở trạng thái 'Đang mượn' hoặc 'Quá hạn'

IF NOT EXISTS (SELECT 1 FROM PhieuMuon

WHERE MaPhieu = @MaPhieu

AND (TrangThai = N'Đang mượn' OR TrangThai = N'Quá hạn'))

BEGIN

RAISERROR(N'Phiếu mượn không tồn tại hoặc không ở trạng thái cho phép trả sách!', 16, 1)

RETURN

END

-- Lấy thông tin từ phiếu mượn

DECLARE @MaSach VARCHAR(20)

DECLARE @SoLuong INT

DECLARE @HanTra DATE

DECLARE @NgayMuon DATE

SELECT @MaSach = MaSach, @SoLuong = SoLuong, @HanTra = HanTra, @NgayMuon = NgayMuon

FROM PhieuMuon

WHERE MaPhieu = @MaPhieu

-- Kiểm tra ngày trả phải lớn hơn ngày mượn

IF (@NgayTra <= @NgayMuon)

BEGIN

RAISERROR(N'Ngày trả thực tế phải lớn hơn ngày mượn!', 16, 1)

RETURN

END

-- Cập nhật trạng thái phiếu mượn thành 'Đã trả' và ghi nhận ngày trả thực tế

UPDATE PhieuMuon

SET TrangThai = N'Đã trả',

NgayTraThucTe = @NgayTra

WHERE MaPhieu = @MaPhieu

-- Cập nhật số lượng sách khả dụng

UPDATE Sach

SET KhaDung = KhaDung + @SoLuong

WHERE MaSach = @MaSach

COMMIT TRANSACTION

RETURN 0 -- Thành công

END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION

DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE()

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY()

DECLARE @ErrorState INT = ERROR\_STATE()

RAISERROR(@ErrorMessage, @ErrorSeverity, @ErrorState)

RETURN -1 -- Thất bại

END CATCH

END

GO

-- Tạo stored procedure sửa phiếu mượn

CREATE PROCEDURE [dbo].[sp\_SuaPhieuMuon]

@MaPhieu INT,

@NgayMuon DATE,

@HanTra DATE,

@NgayTraThucTe DATE = NULL,

@TrangThai NVARCHAR(50),

@SoLuong INT

AS

BEGIN

BEGIN TRY

BEGIN TRANSACTION

-- Kiểm tra phiếu mượn có tồn tại không

IF NOT EXISTS (SELECT 1 FROM PhieuMuon WHERE MaPhieu = @MaPhieu)

BEGIN

RAISERROR(N'Phiếu mượn không tồn tại!', 16, 1)

RETURN

END

-- Lấy thông tin hiện tại từ phiếu mượn

DECLARE @MaSach VARCHAR(20)

DECLARE @SoLuongCu INT

DECLARE @TrangThaiCu NVARCHAR(50)

SELECT @MaSach = MaSach, @SoLuongCu = SoLuong, @TrangThaiCu = TrangThai

FROM PhieuMuon

WHERE MaPhieu = @MaPhieu

-- Cập nhật thông tin phiếu mượn

UPDATE PhieuMuon

SET NgayMuon = @NgayMuon,

HanTra = @HanTra,

NgayTraThucTe = @NgayTraThucTe,

TrangThai = @TrangThai,

SoLuong = @SoLuong

WHERE MaPhieu = @MaPhieu

-- Kiểm tra ngày trả phải lớn hơn ngày mượn nếu có ngày trả thực tế

IF (@NgayTraThucTe IS NOT NULL AND @NgayTraThucTe <= @NgayMuon)

BEGIN

RAISERROR(N'Ngày trả thực tế phải lớn hơn ngày mượn!', 16, 1)

RETURN

END

-- Cập nhật số lượng sách khả dụng nếu có thay đổi trạng thái hoặc số lượng

IF (@TrangThaiCu <> @TrangThai OR @SoLuongCu <> @SoLuong)

BEGIN

-- Nếu trạng thái từ "Đang mượn" hoặc "Quá hạn" sang "Đã trả"

IF (@TrangThaiCu IN (N'Đang mượn', N'Quá hạn') AND @TrangThai = N'Đã trả')

BEGIN

-- Tăng số lượng khả dụng khi trả sách

UPDATE Sach

SET KhaDung = KhaDung + @SoLuongCu

WHERE MaSach = @MaSach

END

-- Nếu trạng thái từ "Đã trả" sang "Đang mượn" hoặc "Quá hạn"

ELSE IF (@TrangThaiCu = N'Đã trả' AND @TrangThai IN (N'Đang mượn', N'Quá hạn'))

BEGIN

-- Giảm số lượng khả dụng khi mượn lại

UPDATE Sach

SET KhaDung = KhaDung - @SoLuong

WHERE MaSach = @MaSach

END

-- Nếu chỉ thay đổi số lượng trong cùng trạng thái "Đang mượn" hoặc "Quá hạn"

ELSE IF (@TrangThaiCu IN (N'Đang mượn', N'Quá hạn') AND @TrangThai IN (N'Đang mượn', N'Quá hạn') AND @SoLuongCu <> @SoLuong)

BEGIN

-- Điều chỉnh số lượng khả dụng theo sự thay đổi

UPDATE Sach

SET KhaDung = KhaDung - (@SoLuong - @SoLuongCu)

WHERE MaSach = @MaSach

END

END

COMMIT TRANSACTION

RETURN 0 -- Thành công

END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION

DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE()

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY()

DECLARE @ErrorState INT = ERROR\_STATE()

RAISERROR(@ErrorMessage, @ErrorSeverity, @ErrorState)

RETURN -1 -- Thất bại

END CATCH

END

GO

-- Stored procedure để xóa phiếu mượn

CREATE PROCEDURE sp\_XoaPhieuMuon

@MaPhieu INT

AS

BEGIN

SET NOCOUNT ON;

-- Kiểm tra phiếu mượn có tồn tại không

IF NOT EXISTS (SELECT 1 FROM PhieuMuon WHERE MaPhieu = @MaPhieu)

BEGIN

RAISERROR(N'Mã phiếu mượn không tồn tại!', 16, 1);

RETURN;

END

-- Lấy thông tin của phiếu mượn

DECLARE @TrangThai NVARCHAR(50)

DECLARE @MaSach VARCHAR(20)

DECLARE @SoLuong INT

SELECT @TrangThai = TrangThai, @MaSach = MaSach, @SoLuong = SoLuong

FROM PhieuMuon

WHERE MaPhieu = @MaPhieu;

BEGIN TRANSACTION;

-- Nếu phiếu đang ở trạng thái "Đang mượn" hoặc "Quá hạn", cập nhật lại số lượng sách khả dụng

IF @TrangThai IN (N'Đang mượn', N'Quá hạn')

BEGIN

UPDATE Sach

SET KhaDung = KhaDung + @SoLuong

WHERE MaSach = @MaSach;

END

-- Xóa phiếu mượn

DELETE FROM PhieuMuon WHERE MaPhieu = @MaPhieu;

COMMIT;

PRINT N'Xóa phiếu mượn thành công!';

END;

GO

-- Tạo stored procedure sp\_ThongKeTongQuan

CREATE PROCEDURE [dbo].[sp\_ThongKeTongQuan]

AS

BEGIN

-- Thống kê sách

SELECT

COUNT(\*) AS TongSoSach,

SUM(KhaDung) AS TongSachKhaDung

FROM Sach;

-- Thống kê thành viên

SELECT

COUNT(\*) AS TongThanhVien

FROM ThanhVien

WHERE TrangThai = N'Hoạt động';

-- Thống kê nhân viên

SELECT

COUNT(\*) AS TongNhanVien

FROM NhanVien

WHERE TrangThai = N'Đang làm';

-- Thống kê phiếu mượn

SELECT

-- Sách mượn hôm nay

(SELECT COUNT(\*) FROM PhieuMuon WHERE CONVERT(DATE, NgayMuon) = CONVERT(DATE, GETDATE())) AS SachMuonHomNay,

-- Sách trả hôm nay

(SELECT COUNT(\*) FROM PhieuMuon WHERE CONVERT(DATE, NgayTraThucTe) = CONVERT(DATE, GETDATE())) AS SachTraHomNay,

-- Sách đang quá hạn

(SELECT COUNT(\*) FROM PhieuMuon WHERE TrangThai = N'Quá hạn') AS SachQuaHan,

-- Sách trả trễ (đã trả nhưng sau hạn trả)

(SELECT COUNT(\*) FROM PhieuMuon

WHERE TrangThai = N'Đã trả'

AND NgayTraThucTe > HanTra) AS SachTraTre

END;

GO

-- Thêm các index để tối ưu hiệu suất truy vấn thống kê

-- Index cho bảng Sach - tăng tốc truy vấn tổng số sách khả dụng

CREATE INDEX IX\_Sach\_KhaDung ON Sach(KhaDung);

GO

-- Index cho bảng PhieuMuon theo NgayMuon - tăng tốc truy vấn sách mượn theo ngày

CREATE INDEX IX\_PhieuMuon\_NgayMuon ON PhieuMuon(NgayMuon);

GO

-- Index cho bảng PhieuMuon theo NgayTraThucTe - tăng tốc truy vấn sách trả theo ngày

CREATE INDEX IX\_PhieuMuon\_NgayTraThucTe ON PhieuMuon(NgayTraThucTe);

GO

-- Index cho bảng PhieuMuon theo TrangThai - tăng tốc truy vấn sách quá hạn

CREATE INDEX IX\_PhieuMuon\_TrangThai ON PhieuMuon(TrangThai);

GO

-- Index tổng hợp cho truy vấn sách trả hôm nay

CREATE INDEX IX\_PhieuMuon\_TrangThai\_NgayTra ON PhieuMuon(TrangThai, NgayTraThucTe)

WHERE TrangThai = N'Đã trả';

GO