QUẢN LÝ MƯỢN SÁCH THƯ VIÊN

A. Khảo sát thực tế và yêu cầu

Hệ thống được sử dụng để quản lý việc mượn sách trong một thư viện. Các tài liệu cho độc giả mượn có các thuộc tính là mã tài liệu (khóa), tên tài liệu (tựa đề). Tài liệu gồm 2 loại: sách và báo tạp chí.

Mỗi tựa đề sách cần được biết do tác giả nào viết. Thông tin về tác giả gồm mã tác giả (khóa), tên tác giả, năm sinh. Một tác giả viết nhiều sách, một sách có thể đồng tác giả. Mỗi tựa đề sách có nhiều lần xuất bản (tái bản). Thông tin về một lần xuất bản gồm có: lần xuất bản, năm xuất bản, khổ giấy, số trang, nhà xuất bản, giá, có hoặc không kèm dĩa CD. Lần xuất bản được đánh 1, 2, 3... cho mỗi tựa đề sách, do đó có sự trùng nhau giữa các tựa đề sách khác nhau.

Mỗi lần xuất bản một tựa đề sách, thư viện nhập vào nhiều cuốn sách. Mỗi cuốn sách này được quản lý riêng dựa vào STT được đánh số từ 1, 2, 3, ... trong số các cuốn sách cùng tựa đề và cùng một lần xuất bản. Khi cho độc giả mượn, thông tin ghi trong thẻ độc giả phải xác định chính xác cuốn nào. Thông tin về mỗi cuốn sách này còn có thêm tình trạng để lưu tình trạng hiện tại của sách (tốt, rách, mất trang...).

Khác với việc cho mượn sách, việc cho mượn báo tạp chí không cần chỉ chính xác tờ nào trong số các tờ cùng tựa đề và cùng một lần xuất bản. Tuy nhiên trong số này (cùng tựa đề và cùng một lần xuất bản), độc giả mỗi lần chỉ có thể mượn 1 tờ. Mỗi tựa đề báo tạp chí cần các thông tin: năm bắt đầu phát hành, định kỳ (hàng ngày, hàng tuần, hay hàng tháng), nhà xuất bản; đối với mỗi kỳ xuất bản cần biết số lượng tờ thư viện nhập về, số lượng tờ còn lại trong thư viện hiện tại (thuộc tính này được tính từ số tờ thư viện nhập về trừ đi số tờ đang có độc giả mượn).

Thông tin về độc giả gồm số thẻ độc giả (khóa), ngày cấp thẻ, tên, nghề nghiệp, phái. Mỗi lần độc giả có thể mượn nhiều sách cũng như báo tạp chí, thông tin cần lưu là ngày mượn và ngày trả cho từng tài liệu mượn.

1. Xây dựng mô hình ERD
2. Xây dựng ứng dụng với GUI để:
   1. Quản lý người dùng / nhóm người dùng: mỗi nhóm người dùng khác nhau sẽ có quyền khác nhau  (việc quản lý thực hiện ở mức server và mức ứng dụng)
   2. Thêm/xoá/sửa các thực thể đã được mô tả
   3. Sao lưu, phục hồi dữ liệu.
   4. Thống kê dữ liệu
3. Lưu ý: trong demo nên sử dụng nonclustered index, view, trigger, stored procedure, user defined function.

B. Phân tích và thiết kế

1. Xây dựng mô hình ERD:

a. Xác định các thực thể

**TAILIEU**(MATAILIEU, TENTAILIEU)

**TUASACH**(MASACH)

**NHANVIEN**(MANHANVIEN, HO, TEN, CHUCVU, TRANGTHAIXOA)

**TACGIA**(MATACGIA, TENTACGIA, NAMSINH)

**LANXUATBAN**(MALANXB, LANXB, NAMXB, KHOGIAY, SOTRANG, NHAXB, GIA, DIACD)

**BAOTAPCHI**(MABAO, NAMPHATHANH, DINHKY, NHAXUATBAN)

**KYXUATBAN**(MAKYXB, SOLUONGNHAP, SOLUONGCON)

**CUONSACH**(STT , TINHTRANG)

**DOCGIA**(SOTHEDOCGIA, NGAYCAPTHE, TEN, NGHENGHIEP, PHAI)

**PHIEUMUON**(MAMUON, NGAYMUON, NGAYTRA, TRANGTHAITRA)

b. Mô hình ERD

A picture containing diagram, line, sketch, origami

Description automatically generated

c/ Mô hình dữ liệu quan hệ từ ERD (Đạt dạng chuẩn 3)

Ký hiệu:

Chữ đen gạch dưới: Khóa chính

Chữ đỏ: Khóa ngoại

Chữ đỏ có gạch dưới: Khóa tương đương

**TAILIEU**(MATAILIEU, TENTAILIEU, LOAITAILIEU)

**TUASACH**(MASACH, MATAILIEU)

**TACGIA**(MATACGIA, TENTACGIA, NAMSINH)

**CHITETSACH**(ID, MASACH, MATACGIA)

**LANXUATBAN**(MALANXB, LANXB, NAMXB, KHOGIAY, SOTRANG, NHAXB, GIA, DIACD, MASACH )

**BAOTAPCHI**(MABAO, NAMPHATHANH, DINHKY, NHAXUATBAN, MATAILIEU)

**KYXUATBAN**(MAKYXB, SOLUONGNHAP, SOLUONGCON, MABAO)

**CUONSACH**(ID, STT, MALANXB, TINHTRANG)

**DOCGIA**(SOTHEDOCGIA, NGAYCAPTHE, TEN, NGHENGHIEP, PHAI)

**NHANVIEN**(MANHANVIEN, HO, TEN, CHUCVU, TRANGTHAIXOA)

**PHIEUMUON**(MAMUON, NGAYMUON, NGAYTRA, TRANGTHAITRA, SOTHEDOCGIA, MAKYXB, IDCUONSACH, MANHANVIENLAP)

d/ Từ điển dữ liệu:  
Tất cả Mã-ID có thể set Int, những thuộc tính không đề cập có thể mang giá trị NvarChar

**TAILIEU**(MATAILIEU, TENTAILIEU, LOAITAILIEU)

**TUASACH**(MASACH, MATAILIEU)

**TACGIA**(MATACGIA, TENTACGIA, NAMSINH) NAMSINH:DateTime

**CHITETSACH**(ID, MASACH, MATACGIA)

**LANXUATBAN**(MALANXB, LANXB, NAMXB, KHOGIAY, SOTRANG, NHAXB, GIA, DIACD, MASACH ) LANXB: Int, NAMXB: Int, SOTRANG: Int, GIA: float

**BAOTAPCHI**(MABAO, NAMPHATHANH, DINHKY, NHAXUATBAN, MATAILIEU): NAMPHATHANH: Int, DINHKY: DateTime

**KYXUATBAN**(MAKYXB, SOLUONGNHAP, SOLUONGCON, MABAO)  
SOLUONGNHAP and SOLUONGCON: Int

**CUONSACH**(ID, STT, MALANXB, TINHTRANG)

**DOCGIA**(SOTHEDOCGIA, NGAYCAPTHE, TEN, NGHENGHIEP, PHAI)

NGAYCAPTHE: DateTime

**NHANVIEN**(MANHANVIEN, HO, TEN, CHUCVU, TRANGTHAIXOA)  
TRANGTHAIXOA: BIT (0,1)

**PHIEUMUON**(MAMUON, NGAYMUON, NGAYTRA, TRANGTHAITRA, SOTHEDOCGIA, MAKYXB, IDCUONSACH, MANHANVIENLAP)  
NGAYMUON and NGAYTRA:DateTime, TRANGTHAITRA: BIT(0,1)  
“MAKYXB” AND “IDCUONSACH” IN PHIEUMUON CAN NULL

e/   
Tạo CSDL

Tạo bảng

Tạo ràng buộc: khoá chính, khoá ngoại ...

Tạo rule

ALTER TABLE DOCGIA

ADD CONSTRAINT CK\_PHAI CHECK (UPPER(PHAI) IN ('NAM', 'NỮ'))

ALTER TABLE [dbo].[NHANVIEN]

ADD CONSTRAINT [CK\_CHUCVU] CHECK (

([CHUCVU]=N'NHÂN VIÊN' COLLATE Latin1\_General\_CI\_AI) OR

([CHUCVU]=N'QUẢN LÝ' COLLATE Latin1\_General\_CI\_AI)

);

ALTER TABLE [dbo].[PHIEUMUON]

ADD CONSTRAINT [CK\_PHIEUMUON\_NGAYTRA] CHECK (NGAYTRA >= NGAYMUON);

Tạo giá trị mặc định

ALTER TABLE [dbo].[DOCGIA] ADD CONSTRAINT [DF\_DOCGIA\_PHAI] DEFAULT (N'NAM') FOR [PHAI]

ALTER TABLE [dbo].[NHANVIEN] ADD CONSTRAINT [DF\_NHANVIEN\_CHUCVU] DEFAULT (N'NHÂN VIÊN') FOR [CHUCVU]

Tạo chỉ mục

-- BAOTAPCHI

IF EXISTS (SELECT 1 FROM sys.indexes WHERE name = 'IX\_BAOTAPCHI\_MABAO')

BEGIN

DROP INDEX IX\_BAOTAPCHI\_MABAO ON dbo.BAOTAPCHI;

END

GO

CREATE INDEX IX\_BAOTAPCHI\_MABAO ON dbo.BAOTAPCHI (MABAO);

GO

-- CHITETSACH

IF EXISTS (SELECT 1 FROM sys.indexes WHERE name = 'IX\_CHITETSACH\_ID')

BEGIN

DROP INDEX IX\_CHITETSACH\_ID ON dbo.CHITETSACH;

END

GO

CREATE INDEX IX\_CHITETSACH\_ID ON dbo.CHITETSACH (ID);

GO

-- CUONSACH

IF EXISTS (SELECT 1 FROM sys.indexes WHERE name = 'IX\_CUONSACH\_ID')

BEGIN

DROP INDEX IX\_CUONSACH\_ID ON dbo.CUONSACH;

END

GO

CREATE INDEX IX\_CUONSACH\_ID ON dbo.CUONSACH (ID);

GO

-- DOCGIA

IF EXISTS (SELECT 1 FROM sys.indexes WHERE name = 'IX\_DOCGIA\_SOTHEDOCGIA')

BEGIN

DROP INDEX IX\_DOCGIA\_SOTHEDOCGIA ON dbo.DOCGIA;

END

GO

CREATE INDEX IX\_DOCGIA\_SOTHEDOCGIA ON dbo.DOCGIA (SOTHEDOCGIA);

GO

-- KYXUATBAN

IF EXISTS (SELECT 1 FROM sys.indexes WHERE name = 'IX\_KYXUATBAN\_MAKYXB')

BEGIN

DROP INDEX IX\_KYXUATBAN\_MAKYXB ON dbo.KYXUATBAN;

END

GO

CREATE INDEX IX\_KYXUATBAN\_MAKYXB ON dbo.KYXUATBAN (MAKYXB);

GO

-- LANXUATBAN

IF EXISTS (SELECT 1 FROM sys.indexes WHERE name = 'IX\_LANXUATBAN\_MALANXB')

BEGIN

DROP INDEX IX\_LANXUATBAN\_MALANXB ON dbo.LANXUATBAN;

END

GO

CREATE INDEX IX\_LANXUATBAN\_MALANXB ON dbo.LANXUATBAN (MALANXB);

GO

-- PHIEUMUON

IF EXISTS (SELECT 1 FROM sys.indexes WHERE name = 'IX\_PHIEUMUON\_MAMUON')

BEGIN

DROP INDEX IX\_PHIEUMUON\_MAMUON ON dbo.PHIEUMUON;

END

GO

CREATE INDEX IX\_PHIEUMUON\_MAMUON ON dbo.PHIEUMUON (MAMUON);

GO

-- TACGIA

IF EXISTS (SELECT 1 FROM sys.indexes WHERE name = 'IX\_TACGIA\_MATACGIA')

BEGIN

DROP INDEX IX\_TACGIA\_MATACGIA ON dbo.TACGIA;

END

GO

CREATE INDEX IX\_TACGIA\_MATACGIA ON dbo.TACGIA (MATACGIA);

GO

-- TAILIEU

IF EXISTS (SELECT 1 FROM sys.indexes WHERE name = 'IX\_TAILIEU\_MATAILIEU')

BEGIN

DROP INDEX IX\_TAILIEU\_MATAILIEU ON.dbo.TAILIEU;

END

GO

CREATE INDEX IX\_TAILIEU\_MATAILIEU ON.dbo.TAILIEU (MATAILIEU);

GO

Tạo các view

1. nonclustered index, view, trigger, stored procedure, user defined function.