

# BÀI TẬP THỰC HÀNH SỐ 3

## THỰC HÀNH THIẾT KẾ DỮ LIỆU

### I. MỤC TIÊU

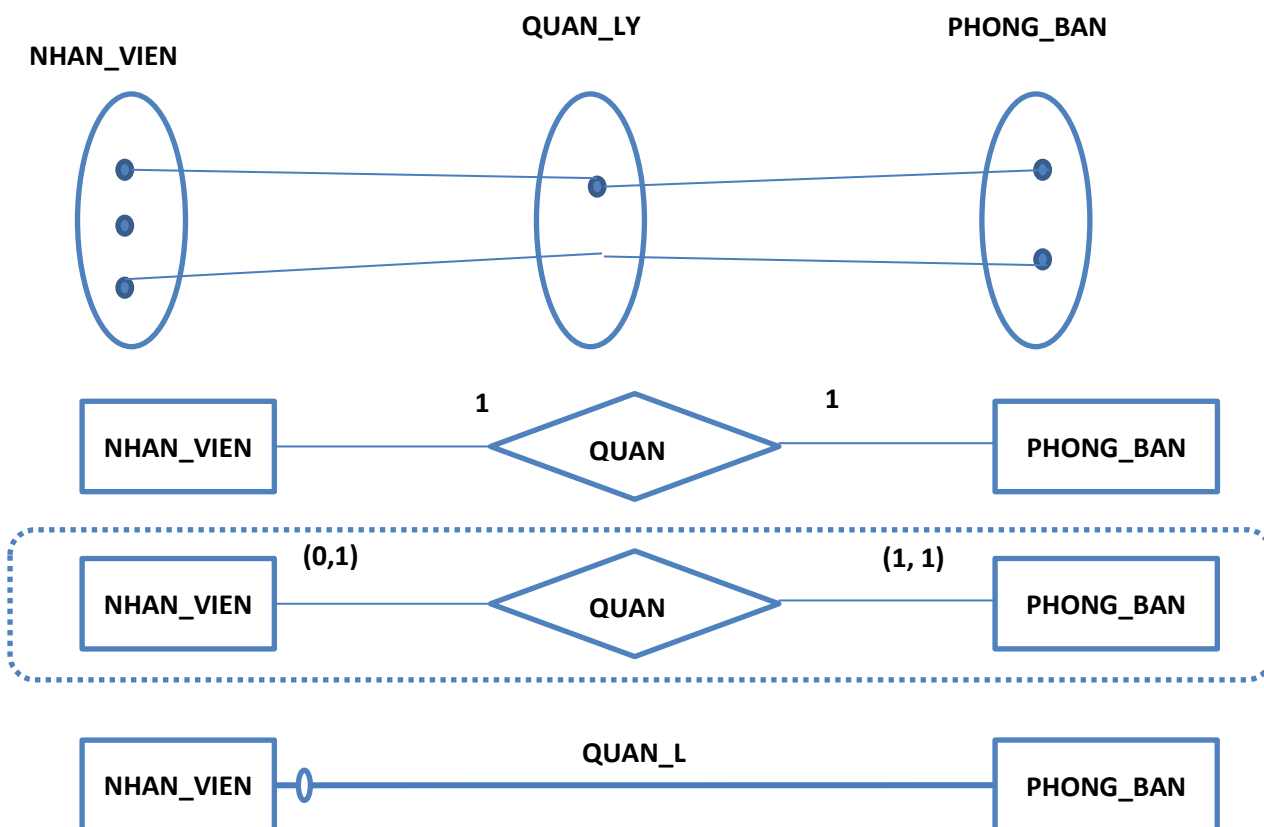
Thành thạo thiết kế dữ liệu:

- Thông tin tổng quát (Sơ đồ logic/ E-R):
  - Xác định danh sách các bảng dữ liệu: việc lưu trữ cần bao nhiêu bảng? Là các bảng nào?
  - Xác định danh sách các liên kết: các bảng dữ liệu có quan hệ, liên kết giữa chúng như thế nào?
  - Thiết kế dữ liệu với tính đúng đắn
  - Thiết kế dữ liệu với yêu cầu chất lượng
  - Thiết kế dữ liệu với yêu cầu hệ thống
- Thông tin chi tiết:
  - Lập Bảng thuộc tính mô tả chi tiết từng thành phần lưu trữ của mỗi bảng

### II. CƠ SỞ LÝ THUYẾT

#### 2.1. Sơ đồ thực thể liên kết

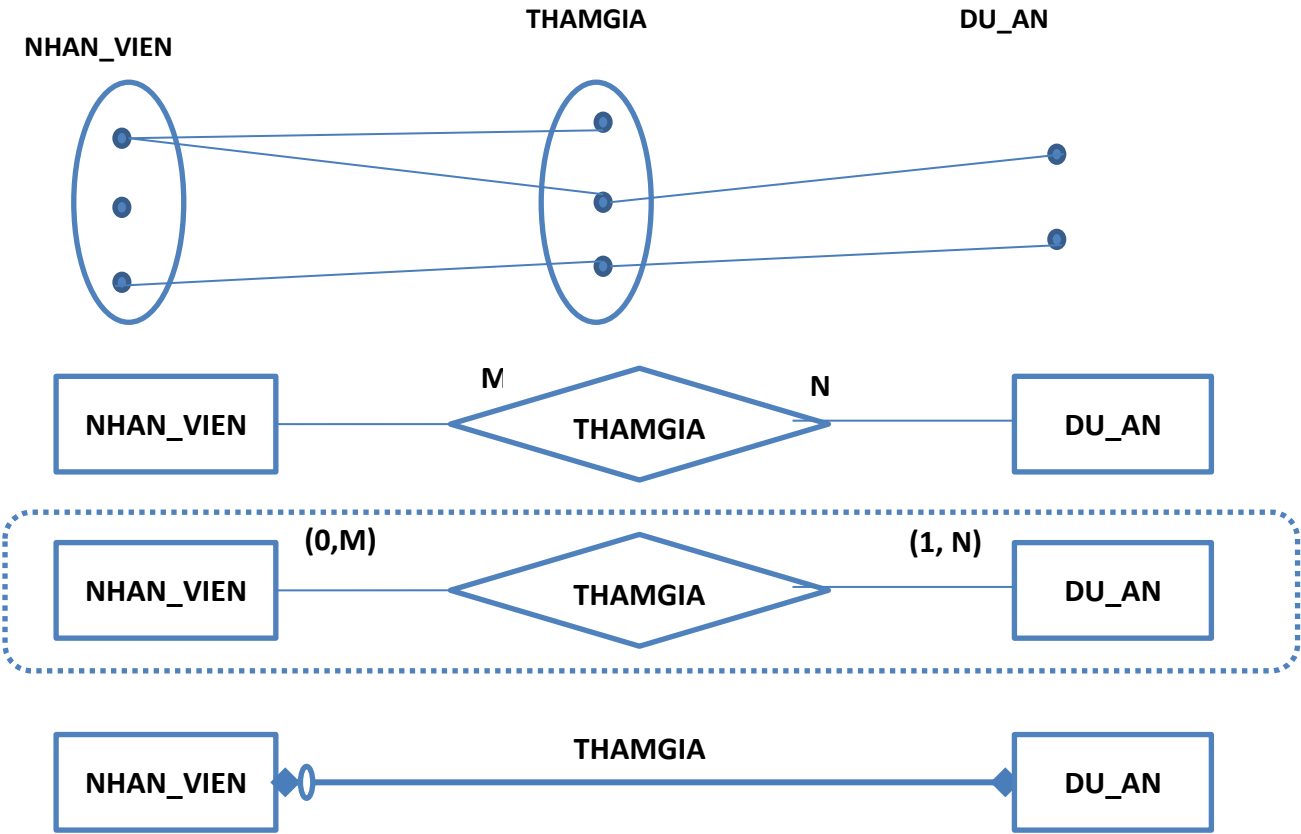
##### 2.1.1. Mỗi quan hệ 1 - 1



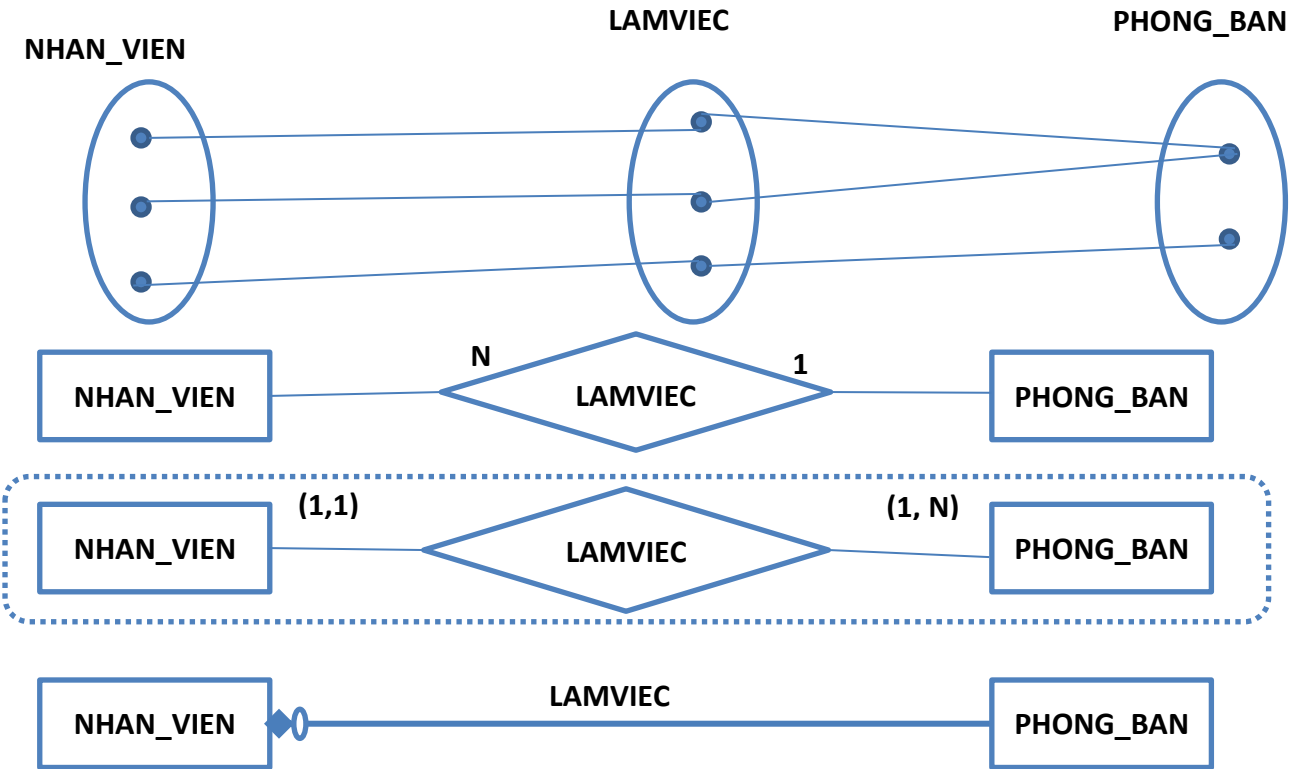


Thực hành công nghệ phần mềm

2.1.2. Mối quan hệ nhiều - nhiều (m-n)



2.1.3. Mối quan hệ một - nhiều (1 - n)



## 2.2. Lập bảng mô tả thuộc tính

III. Thành phần: DOC\_GIA

IV. Ý nghĩa: Thông tin về độc giả, có 2 loại: Basic và Premium.

- Loại Basic: được mượn tối đa 5 cuốn sách/lần với thời gian mượn  $\leq 7$  ngày.
- Loại Premium: được mượn tối đa 10 cuốn sách/lần với thời gian mượn  $\leq 30$  ngày

TT	Tên thuộc tính (Field name)	Kiểu dữ liệu	Độ rộng	Not NULL	Ràng buộc / Miền giá trị	Diễn giải
1	MADG	Chuỗi	10	Yes	'B000012345' B: Basic, P: Premium	
2	TENDG					

## 2.3. Thiết kế dữ liệu với tính đúng đắn

- Bảo đảm rằng các thông tin cần thiết để thực hiện 1 yêu cầu nghiệp vụ, biểu mẫu, quy định được lưu trữ đầy đủ và chính xác về mặt ngữ nghĩa
- Căn cứ theo danh sách yêu cầu nghiệp vụ, các biểu mẫu và quy định đi kèm để thiết kế dữ liệu
  - o Với mỗi yêu cầu cần xác định rõ **cần lưu trữ các thông tin gì?** (dựa vào luồng dữ liệu đọc/ghi trong sơ đồ luồng dữ liệu tương ứng) và tìm cách bổ sung các thuộc tính để lưu trữ các thông tin này → chỉ xem xét tính đúng đắn
  - o Dựa trên Biểu mẫu, xác định các đối tượng, thông tin cần lưu trữ và mối quan hệ giữa các đối tượng
  - o Cần chọn các yêu cầu theo **thứ tự từ đơn giản đến phức tạp** (thông thường yêu cầu tra cứu là đơn giản nhất). Với yêu cầu phức tạp có thể phải bổ sung vào sơ đồ logic nhiều thành phần mới.
  - o Khóa của các thành phần phải **dựa trên ngữ nghĩa** tương ứng trong thế giới thực
  - o Thêm 1 cột Bảng truy xuất vào bảng yêu cầu nghiệp vụ (ghi rõ tên các Bảng tương ứng trong CSDL cần truy xuất vào để đọc/ghi dữ liệu)

Ví dụ: Xét phần mềm quản lý thư viện với 4 yêu cầu sau:

- o Lập thẻ độc giả
- o Nhập sách mới
- o Mượn sách
- o Trả sách

Xác định các đối tượng cần lưu trữ: Độc giả, Sách, Thông tin mượn trả sách

1 độc giả có thể mượn nhiều sách

1 quyển sách có thể được mượn bởi nhiều độc giả

Quan hệ giữa SÁCH và ĐỘC\_GIA là quan hệ nhiều – nhiều → Có đối tượng trung gian là MUON\_TRA\_SÁCH



Chi tiết các bảng:

DOC\_GIA (MaDG, HoTen, LoaiDG, NgaySinh, NgayLapThe, NgayHetHan)

SACH(MaSach, TenSach, TheLoai, NgayNhap, TacGia, NhaXB, NamXB)

MUON\_TRA\_SACH(MaDG, MaSach, NgayMuon, NgayTra, TienPhat)

## 2.4. Thiết kế dữ liệu với yêu cầu chất lượng

- Vẫn bảo đảm tính đúng đắn nhưng thỏa mãn thêm các yêu cầu chất lượng khác (tiến hóa, tốc độ nhanh, lưu trữ tối ưu, ...)
- Cần chú ý bảo đảm tính đúng đắn khi cải tiến sơ đồ logic
- Căn cứ vào danh sách các yêu cầu chất lượng để cải tiến sơ đồ logic đã thiết kế ở bước 2.3
- Có 03 loại yêu cầu:
  - o Tính tối ưu lưu trữ - tiến hóa (khả năng thay đổi, mở rộng trong tương lai)
  - o Tính hiệu quả - truy xuất nhanh (tối ưu về mặt tốc độ xử lý)
  - o Tính hiệu quả - lưu trữ tối ưu (tối ưu về mặt lưu trữ)

### 2.4.1. Thiết kế dữ liệu với tính tối ưu lưu trữ - tiến hóa

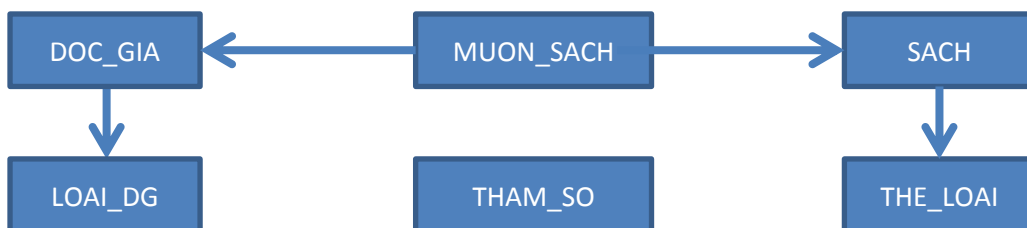
- Căn cứ vào các Quy định, Biểu mẫu của từng Yêu cầu nghiệp vụ để định nghĩa các bảng Tham số:
  - o **Bảng tham số Quy định:** Nếu trong quy định, biểu mẫu có chứa tham số thì bổ sung tham số này vào bảng THAM\_SO (ngoại trừ một số trường hợp đặc biệt). Mỗi tham số trong Quy định  $\Leftrightarrow$  1 Tham số cần lưu trữ
  - o **Bảng tham số Thể Loại (category):** Nếu trong quy định, biểu mẫu có các thông tin mà miền giá trị là rời rạc (chỉ có một số giới hạn các giá trị) thì tạo bảng mới tương ứng (bảng thể loại).
  - o **Cấu trúc chung của bảng Tham số:** {Mã tham số, Giá trị, Ghi chú}, Ghi chú: mục đích sử dụng của Tham số
- Các cách xây dựng bảng Tham số:
  - o **01 bảng Tham số duy nhất dùng chung cho cả hệ thống:** áp dụng với các hệ thống có ít Tham số cần lưu trữ. Giá trị sẽ định nghĩa kiểu String, và khi cần sẽ ép kiểu sang kiểu Dữ liệu tương ứng
  - o **N bảng Tham số tương ứng với từng kiểu dữ liệu** (VD: ThamSoKieuInt, ThamSoKieuDate, ...): áp dụng cho các hệ thống lớn có nhiều Tham số cần lưu trữ.
  - o **Bảng tham số Thể loại:** mỗi thể loại  $\Leftrightarrow$  1 bảng tham số

Ví dụ:

Loại độc giả có miền giá trị rời rạc: B – Basic, P - Premium  $\rightarrow$  tách thành bảng Loai\_DG

Tương tự, Thể loại sách có miền giá trị rời rạc: Sách Giáo Khoa, Sách Kinh tế, Sách chính trị, Sách văn học nước ngoài, ...  $\rightarrow$  tách riêng thành bảng THE\_LOAI.

Một số quy định liên quan với sách và độc giả (tuổi tối thiểu, tối đa, số lượng sách cho mượn tối đa, ...) có thể thay đổi theo thời gian  $\rightarrow$  tách riêng thành bảng THAM\_SO để lưu trữ các định nghĩa này.



Chi tiết các bảng:

DOC\_GIA (MaDG, HoTen, **MaLoaiDG**, NgaySinh, NgayLapThe, NgayHetHan)

SACH (MaSach, TenSach, **MaTheLoai**, NgayNhap, TacGia, NhaXB, NamXB)

MUON\_SACH (MaDG, MaSach, NgayMuon, NgayTra, TienPhat)

**LOAI\_DG (MaLoaiDG, TenLoaiDG, GhiChu)**

**THE\_LOAI (MaTheLoai, TenTheLoai, GhiChu)**

**THAM\_SO (MaThamSo, GiaTri, GhiChu)**

MaThamSo	GiaTri	GhiChu
TS1	18	Tuổi tối thiểu của độc giả
TS2	55	Tuổi tối đa của độc giả
TS3	10	Số năm tối đa xuất bản trước đó
TS4	5	Số sách mượn tối đa
TS5	7	Số ngày mượn tối đa
TS6	10000	Đơn giá phạt của 1 ngày trả sách trễ hạn

#### 2.4.2. Thiết kế dữ liệu với tính hiệu quả - truy xuất nhanh (tối ưu về tốc độ xử lý)

- Căn cứ vào các Yêu cầu xử lý tính toán và Yêu cầu báo cáo, xác định các trường dữ liệu cần tính toán (ví dụ: Số sách đang mượn của độc giả, Tổng số nợ hiện tại của khách hàng), và tạo bổ sung thêm các thuộc tính vào các bảng dùng lưu trữ các thông tin đã tính toán trước (theo quy tắc nào đó từ thông tin gốc đã được lưu trữ).
- Các thông tin này phải được tự động cập nhật khi có bất kỳ thay đổi nào trên thông tin gốc liên quan (Ví dụ: Độc giả mượn thêm hay trả sách, Khách hàng trả tiền hay mua thêm), hoặc có thể được tính toán sau bằng cách viết các hàm Trigger, Procedure để cập nhật dữ liệu (vd: tính cước khi hết tháng)
- Sau mỗi bước phải lập bảng danh sách các thuộc tính tính toán cùng với thông tin liên quan: Thông tin gốc, Xử lý tự động cập nhật thông tin gốc (chi tiết về các xử lý này mô tả trong phần thiết kế xử lý)

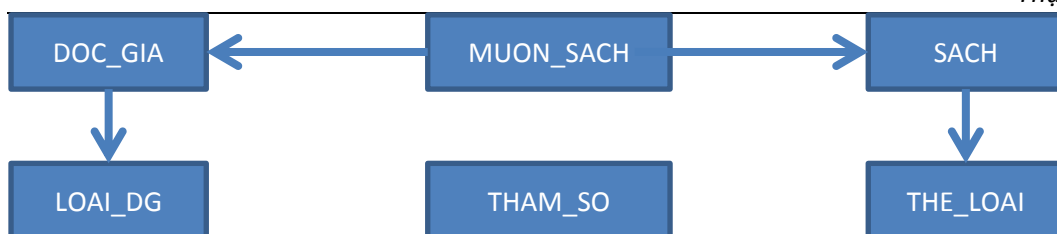
TT	Thuộc tính	Bảng của thuộc tính	Bảng của thông tin gốc	Xử lý tự động cập nhật
1				

Ví dụ:

2. thống kê nhanh số lượng sách đang mượn và tình trạng trả trễ của độc giả
3. tra cứu nhanh tình trạng mượn của sách để biết quyền sách nào có thể cho mượn

Để giảm thời gian truy xuất,

- thêm 2 trường **SoSachDangMuon**, **TinhTrangTraTre** vào bảng DOC\_GIA, thay vì truy xuất bảng MUON\_SACH để tính toán số lượng sách đang mượn, và tình trạng trả sách của độc giả mỗi lần cần làm thống kê
- thêm trường **TinhTrangMuon** vào bảng SACH



#### Chi tiết các bảng:

DOC\_GIA (MaDG, HoTen, MaLoaiDG, NgaySinh, NgayLapThe, NgayHetHan, **SoSachDangMuon**, **TinhTrangTraTre**)

SACH (MaSach, TenSach, MaTheLoai, NgayNhap, TacGia, NhaXB, NamXB, **TinhTrangMuon**)

MUON\_SACH (MaDG, MaSach, NgayMuon, NgayTra, TienPhat)

LOAI\_DG (MaLoaiDG, TenLoaiDG, GhiChu)

THE\_LOAI (MaTheLoai, TenTheLoai, GhiChu)

THAM\_SO (MaThamSo, GiaTri, GhiChu) → Nội dung bảng tham số không đổi

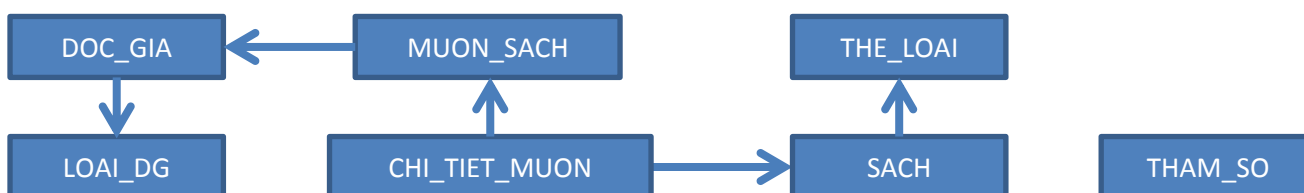
TT	Thuộc tính	Bảng của thuộc tính	Bảng của thông tin gốc	Xử lý tự động cập nhật
1	SoSachDangMuon	DOC_GIA	MUON, CT_MUON	Cho mượn sách Nhận trả sách
2	TinhTrangTraTre	DOC_GIA	MUON, CT_MUON	Khởi động phần mềm
3	TinhTrangSach	SACH	CT_MUON	Cho mượn sách Nhận trả sách

#### 2.4.3. Thiết kế dữ liệu với tính hiệu quả - lưu trữ tối ưu (tối ưu về không gian lưu trữ)

- Tối ưu hóa lưu trữ của phần mềm thông qua việc tổ chức lại CSDL (các kỹ thuật lưu trữ tối ưu khác như nén, mã hóa, ... sẽ không trình bày)
- Tổ chức lại CSDL sẽ chú trọng đến các loại bảng sau:
  - Các bảng có khối lượng lưu trữ lớn và đặc biệt dữ liệu lưu trữ được phát sinh thường xuyên theo chu kỳ thời gian ngắn (tuần, ngày, giờ, ...). Các bảng này nếu không lưu trữ tối ưu sẽ dẫn tới hao phí rất nhiều dung lượng bộ nhớ phụ.
  - Các bảng mà khóa của bảng bao gồm nhiều thuộc tính. Việc sử dụng nhiều thuộc tính trong khóa rất có khả năng dẫn tới lưu trữ không tối ưu (Ví dụ: Bảng MUON\_SACH trong quản lý thư viện, Bảng HOA\_DON trong quản lý nhà sách)
- Các bước thực hiện:
  - Bước 1: Lập danh sách các bảng cần được xem xét để tối ưu hóa việc lưu trữ
    - Xem xét và xác định các công việc có tần suất thực hiện thường xuyên và bổ sung vào danh sách chọn các bảng được sử dụng tương ứng của công việc này
    - Xem xét các bảng mà khóa của bảng bao gồm nhiều thuộc tính và bổ sung bảng này vào danh sách được chọn

- **Bước 2: Tối ưu hóa việc lưu trữ các bảng có khối lượng lưu trữ lớn thông qua việc tối ưu hóa lưu trữ từng thuộc tính trong bảng**
  - Xác định các thuộc tính mà việc lưu trữ chưa tối ưu. Ưu tiên xem xét các thuộc tính có kiểu chuỗi.
  - Tối ưu hóa việc lưu trữ này theo từng trường hợp cụ thể. Một trong các trường hợp thông dụng nhất là chuỗi có kích thước lớn và giá trị được sử dụng nhiều lần trong các mẫu tin khác nhau. VD: thuộc tính *TacGia*, *NhaXB* trong bảng *SACH*
  - Với trường hợp trên, việc tối ưu hóa có thể thực hiện thông qua việc bổ sung các bảng mới (bảng *TAC\_GIA*, *NHA\_XB*) và tổ chức lại cấu trúc bảng *SACH* (thay *TacGia* = *MaTG*, *NhaXB* = *MaNXB*)
- **Bước 3: Tối ưu hóa việc lưu trữ các bảng mà khóa của bảng gồm nhiều thuộc tính**
  - Phân rã bảng đang xét *A* thành 2 bảng *B* và *C*
  - *B*: chứa các thuộc tính mà giá trị được lặp lại nhiều lần trong cùng 1 lần thực hiện công việc tương ứng trong thế giới thực. *B* cần có khóa riêng (sẽ được *C* sử dụng để tham chiếu đến)
  - *C*: chứa các thuộc tính còn lại và khóa của *B*

**Ví dụ:** Với 1 lần mượn, độc giả có thể mượn nhiều cuốn sách



**Chi tiết các bảng:**

DOC\_GIA (MaDG, HoTen, MaLoaiDG, NgaySinh, NgayLapThe, NgayHetHan, SoSachDangMuon, TinhTrangTraTre)

SACH (MaSach, TenSach, MaTheLoai, NgayNhap, TacGia, NhaXB, NamXB, TinhTrangMuon)

MUON\_SACH (**MaMuon**, MaDG, NgayMuon)

**CHI\_TIET\_MUON (MaMuon, MaSach, NgayTra, TienPhat)**

LOAI\_DG (MaLoaiDG, TenLoaiDG, GhiChu)

THE\_LOAI (MaTheLoai, TenTheLoai, GhiChu)

THAM\_SO (MaThamSo, GiaTri, GhiChu) → Nội dung bảng tham số không đổi

## 2.5. Thiết kế dữ liệu với yêu cầu hệ thống

- Vẫn bảo đảm tính đúng đắn và các yêu cầu chất lượng khác nhưng thỏa mãn thêm các yêu cầu hệ thống (bảo mật – security, hiệu suất – performance, phân quyền, cấu hình phần cứng, môi trường phần mềm, sao lưu backup dữ liệu, ...)
- Căn cứ vào danh sách các yêu cầu hệ thống để cải tiến sơ đồ logic đã thiết kế ở bước 2.3 và 2.4

### 2.5.1. Thiết kế dữ liệu theo yêu cầu về hiệu suất (Performance)

- Xem xét các thao tác xử lý tính toán và thao tác báo cáo → thiết kế dữ liệu với tính tối ưu về tốc độ xử lý như bước 2.4.2

**2.5.2. Thiết kế dữ liệu theo yêu cầu bảo mật (Security)**

Cấp độ	Thực hiện
Dữ liệu (data)	Mã hóa (encrypt) các trường dữ liệu nhạy cảm (mật khẩu, số thẻ visa/master, ...)
Bảng (table)	Ví dụ hệ thống B2C có 2 nhóm khách hàng Business (doanh nghiệp) và Customer (cá nhân): tách riêng 2 nhóm khách hàng này thành 02 bảng riêng
Hệ thống	Phân quyền sử dụng dựa trên vai trò của người sử dụng  Triển khai trên các server riêng theo từng nhóm người dùng hệ thống (ví dụ: 1 server cho nhóm khách hàng Business, 1 server cho nhóm khách hàng Customer, đóng gói riêng source code, UI, ...)

**2.5.3. Thiết kế dữ liệu theo yêu cầu về cấu hình phần cứng, phần mềm, thiết bị**

Tùy theo cấu hình của phần cứng, phần mềm, thiết bị, xác định các hạn chế:

- Về mặt bộ nhớ → thực hiện các bước thiết kế **tối ưu không gian lưu trữ** như bước 2.4.3
- Về mặt vi xử lý → thực hiện các bước thiết kế **tối ưu tốc độ xử lý** như bước 2.4.2

**2.5.4. Thiết kế dữ liệu theo yêu cầu backup, sao lưu dữ liệu**

Dựa trên các yêu cầu backup, sao lưu dữ liệu: xác định:

- Các bảng nào, trường dữ liệu nào cần backup, sao lưu
- Tần suất backup, sao lưu
- Thiết bị backup, sao lưu (ra bảng backup, ra bộ nhớ ngoài, ...)
- Tạo view, procedure, trigger để thực hiện việc backup, sao lưu