



CÔNG NGHỆ PHẦN MỀM

Bài 4:

TÁC VỤ THIẾT KẾ VÀ TỔ CHỨC DỮ LIỆU

Thời gian: 3 tiết



Giảng viên: ThS. Dương Thành Phết

Email: phetcm@gmail.com

Website: <http://www.thayphet.net>

Tel: 0918158670

facebook.com/DuongThanhPhet

NỘI DUNG

1. Tổng quan
2. Kết quả thiết kế
3. Quá trình thiết kế
4. Phương pháp thiết kế dữ liệu

4.1. TỔNG QUAN

- ✓ Mục tiêu chính của TK dữ liệu là mô tả cách thức tổ chức lưu trữ các dữ liệu của phần mềm. Có hai dạng lưu trữ chính:
 - **Dạng tập tin:** Chỉ thích hợp với một số phần mềm đặc thù (từ điển, trò chơi ...) chỉ chú trọng nhiều vào xử lý, hình thức giao diện, không chú trọng nhiều đến việc lưu trữ lại các thông tin được tiếp nhận trong quá trình sử dụng phần mềm thông tin này được tiếp nhận và xử lý ngay.
 - **Dạng cơ sở dữ liệu:** Cách tiếp cận này rất thông dụng và đóng vai trò quan trọng trong việc TK những phần mềm vừa và lớn hoặc phân tán ...

4.2. KẾT QUẢ CỦA THIẾT KẾ

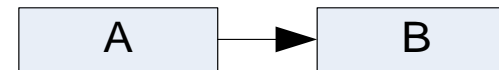
Cách thức tổ chức lưu trữ DL của PM được mô tả thông qua 2 loại thông tin:

- ✓ **Tổng quát:** Cung cấp góc nhìn tổng quan về các thành phần lưu trữ, gồm có:
 - DS *bảng dữ liệu*: liên quan đến việc lưu trữ các bảng dữ liệu cụ thể.
 - DS *liên kết*: liên quan đến việc hiện thực các mối liên kết dữ liệu giữa các bảng dữ liệu.
- ✓ **Thông tin chi tiết:**
 - DS các thuộc tính của từng thành phần: liên quan đến các thông tin cần lưu trữ của thành phần
 - DS MGT *toàn vẹn*: liên quan đến các quy định về tính hợp lệ của các thông tin được lưu trữ.

4.2. KẾT QUẢ CỦA THIẾT KẾ

- ✓ *Sơ đồ luận lý* là cách để biểu diễn thông tin tổng quát
- ✓ Bảng thuộc tính cùng MGT dùng để mô tả chi tiết các thành phần trong sơ đồ, cho phép thể hiện hệ thống các *bảng dữ liệu* cùng với *quan hệ liên kết* giữa chúng (gần giống ERD), với các ký hiệu được dùng như sau:

- Bảng: hình chữ nhật



- Liên kết: (xác định duy nhất một): Mũi tên

Mũi có nghĩa: 1 phần tử A sẽ xác định duy nhất 1 phần tử B, ngược lại 1 phần tử B có thể tương ứng với nhiều phần tử A.

4.2. KẾT QUẢ CỦA THIẾT KẾ

Ví dụ: Với phần mềm quản lý thư viện sơ đồ luận lý:



- ✓ Theo sơ đồ được tổ chức 3 bảng (DOCGIA, MUONSACH, SACH) với 2 liên kết.
- ✓ Sơ đồ trên chỉ là một trong nhiều cách thức tổ chức lưu trữ DL (*Xem tài liệu PT&TK thống thông tin*).
- ✓ Bảng thuộc tính cho phép mô tả chi tiết thành phần trong sơ đồ dạng: Thành phần, Ý nghĩa ...

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1					
2					
...					

4.2. KẾT QUẢ CỦA THIẾT KẾ

Bảng MGT cho phép mô tả các MGT giữa các thuộc tính cùng một thành phần hay nhiều thành phần khác nhau.

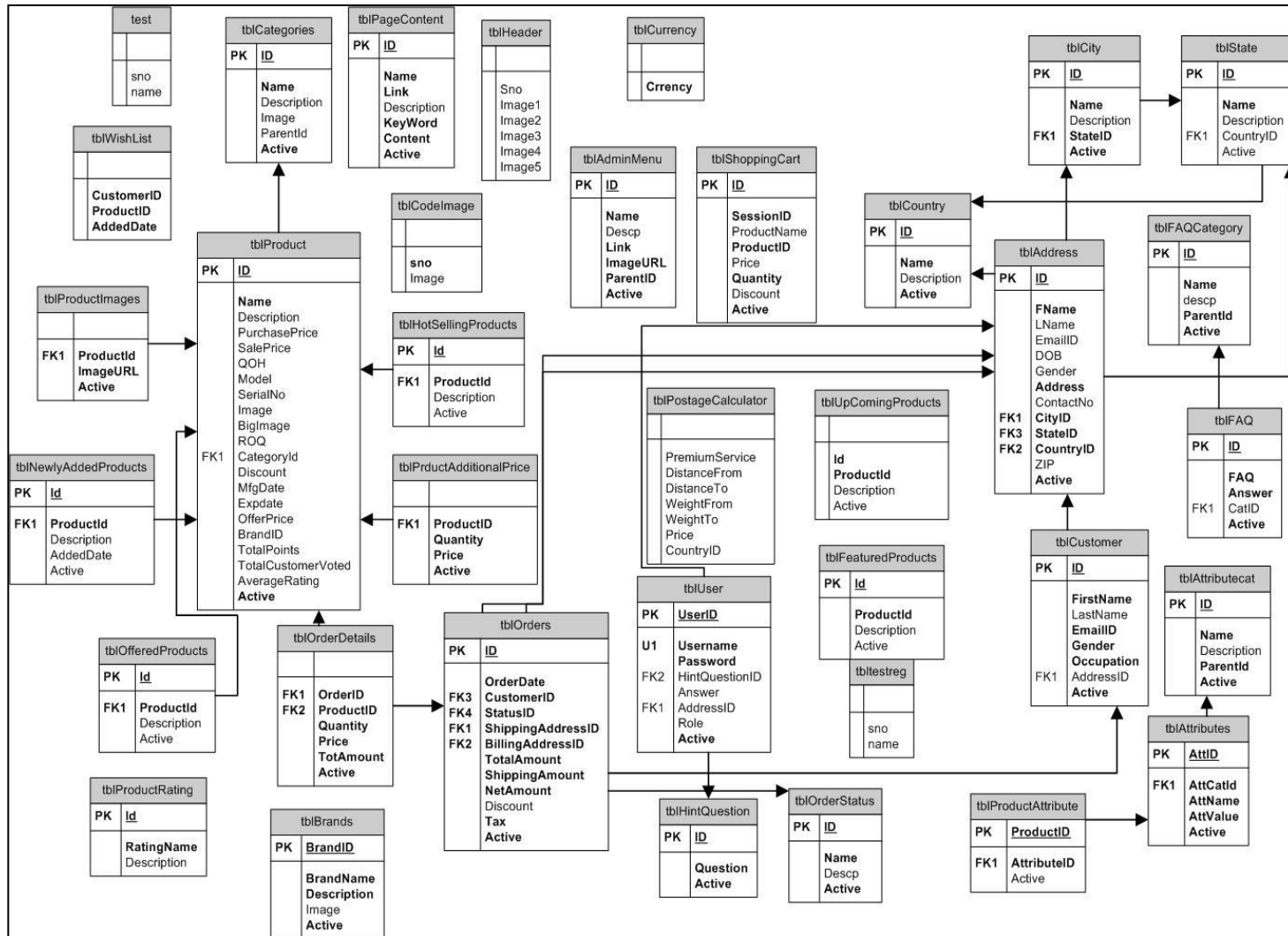
Mã Số	Mô tả miền giá trị	Thành phần liên quan	Ghi chú
RB1			
RB2			
...			

Ghi chú: Bảng thuộc tính cho phép mô tả chi tiết thành phần cần lưu trữ, sẽ dùng trong báo cáo về TK DL của PM. Tuy nhiên mô tả trên khá dài dòng, nên sẽ sử dụng một dạng trình bày cô đọng hơn dạng *lược đồ quan hệ* (gồm tên bảng và thuộc tính, các thuộc tính khóa được gạch chân).

- ✓ **DOC_GIA**(MDG, HoTen, LoaiDG, NgaySinh, NgayLapThe, DiaChi)
- ✓ **SACH**(MSACH, TenSach, TheLoai, NgayNhap, TacGia, NhaXuatBan, NamXuatBan)
- ✓ **MUON**(MDG, MSACH, NgayMuon, NgayTra)

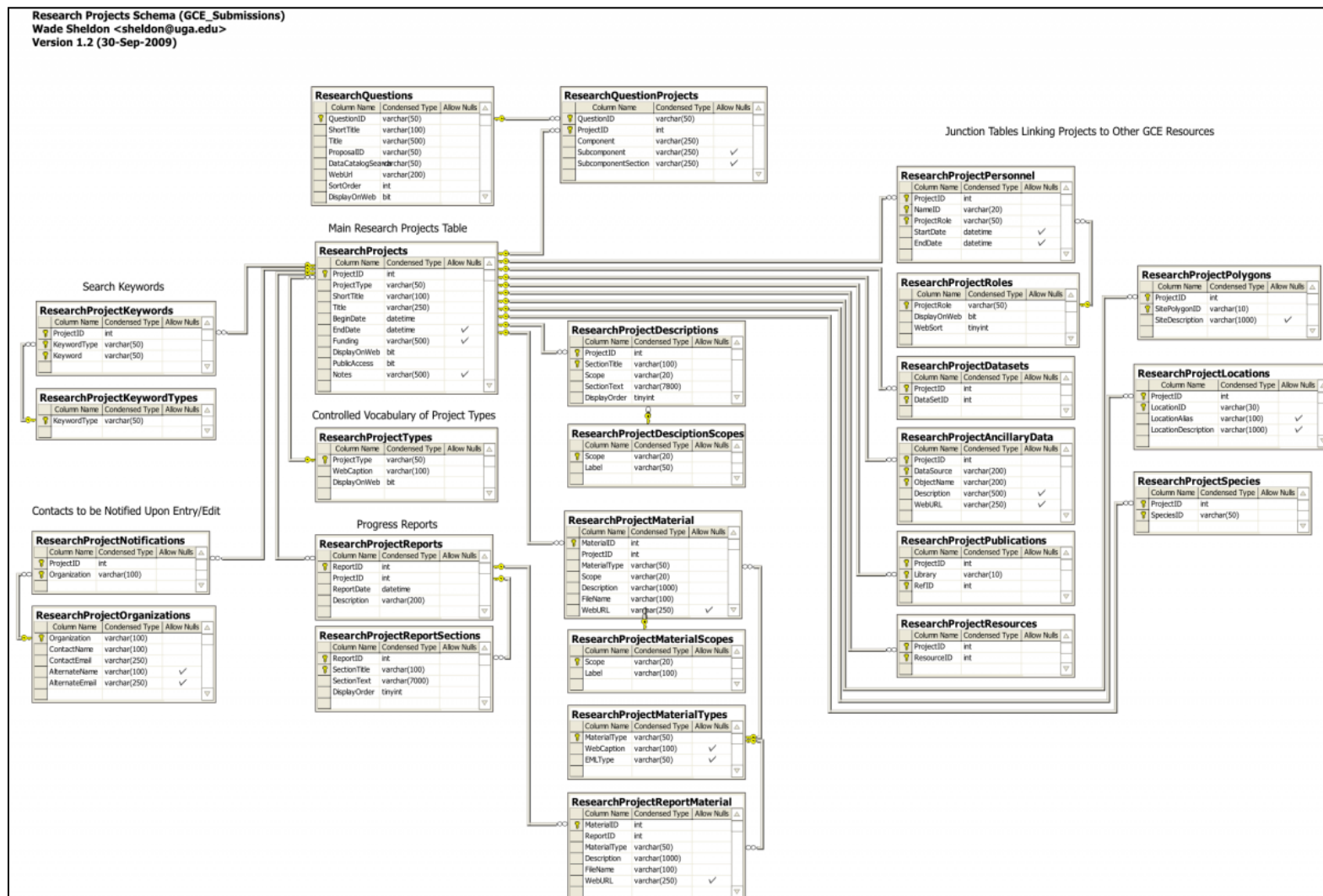
4.2. KẾT QUẢ CỦA THIẾT KẾ

Ví dụ:



4.2. KẾT QUẢ CỦA THIẾT KẾ

Ví dụ:



4.3. QUÁ TRÌNH THIẾT KẾ

Có 2 cách tiếp cận để TK DL:

- ✓ **Phương pháp trực tiếp:** Từ các yêu cầu tạo lập trực tiếp sơ đồ luận lý cùng với bảng thuộc tính và bảng miền giá trị (khó thực hiện đối với sơ đồ luận lý phức tạp).
- ✓ **Phương pháp gián tiếp:** Từ các yêu cầu tạo lập mô hình quan niệm dữ liệu, sau đó tạo lập sơ đồ luận lý, bảng thuộc tính, bảng miền giá trị và đưa vào mô hình này (dễ thực hiện hơn vì mô hình quan niệm dữ liệu thường đơn giản, chứa các thành phần dữ liệu bản chất nhất của phần mềm).

4.3. QUÁ TRÌNH THIẾT KẾ

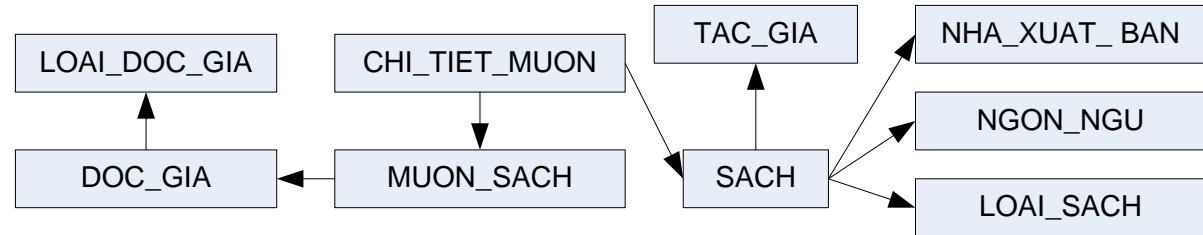
Ứng với 3 yêu cầu của PM, việc TK DL gồm 3 bước:

- ✓ **Thiết kế với tính đúng đắn:** Đảm bảo đầy đủ và chính xác về mặt ngữ nghĩa các thông tin liên quan đến các công việc trong yêu cầu.
- ✓ **Thiết kế với yêu cầu chất lượng:** Đảm bảo tính đúng đắn nhưng thỏa mãn thêm các yêu cầu chất lượng (tiến hóa, tốc độ nhanh, lưu trữ tối ưu) và bảo đảm tính đúng đắn khi cải tiến sơ đồ luận lý.
- ✓ **Thiết kế với yêu cầu hệ thống:** Đảm bảo tính đúng đắn và các yêu cầu chất lượng khác nhưng thỏa mãn thêm các yêu cầu hệ thống (phân quyền, cấu hình phần cứng, môi trường phần mềm ...)

4.3. QUÁ TRÌNH THIẾT KẾ

Ví dụ 1: Xét phần mềm QL thư viện, với phương pháp trực tiếp :

Sơ đồ luận lý

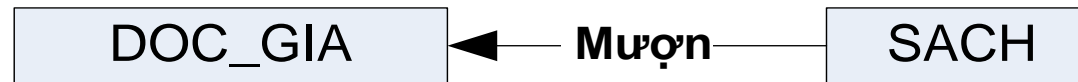


Các bảng thuộc tính

- ✓ **DOC_GIA**(MDG, MLDG, HoTen, NgaySinh, DiaChi, DienThoai)
- ✓ **SACH**(MSACH, MTG, MNXB, MLSACH, MNN, TenSach, NgayMua, SoTrang)
- ✓ **PHIEU_MUON**(MPHM, NgayMuon)
- ✓ **CHI_TIET_MUON**(MPHM, MSACH, NgayTra)
- ✓ **LOAI_SACH**(MLSACH, TenLS, GhiChu)
- ✓ **LOAI_DOC_GIA**(MLDG, TenLoaiDocGia, GhiChu)
- ✓ **NHA_XUAT_BAN**(MNXB, TenNhaXuatBan, GhiChu)
- ✓ **TAC_GIA**(MTG, Ten, GhiChu)
- ✓ **NGON_NGU**(MNN, Ten, GhiChu)

4.3. QUÁ TRÌNH THIẾT KẾ

Với phương pháp gián tiếp, ngoài kết quả cuối cùng tương tự phương pháp trực tiếp, ta còn kết quả trung gian là mô hình quan niệm dữ liệu như sau:



Sơ đồ lớp đối tượng với 2 đối tượng chính Sách, Độc giả và 1 quan hệ mượn giữa 2 lớp đối tượng trên.

Mô hình chi tiết các thành phần trong sơ đồ lớp: Xem chi tiết ở phụ lục B

4.3. QUÁ TRÌNH THIẾT KẾ

Ví dụ 2: Xét phần mềm với 4 yêu cầu: Lập thẻ độc giả, Nhận sách, Cho mượn sách, Trả sách

Thiết kế dữ liệu với tính đúng đắn

Sơ đồ luận lý



Chi tiết các bảng:

- ✓ **DOC_GIA**(MDG,MLDG,HoTen, NgaySinh, DiaChi,DienThoai)
- ✓ **SACH**(MSACH,MTG,MNXB,MLSACH,MNN, TenSach, NgayMua, SoTrang)
- ✓ **MUON_SACH**(MDG, MSACH, NgayMuon, NgayTra, TienPhat)

4.3. QUÁ TRÌNH THIẾT KẾ

Thiết kế dữ liệu với tính tiến hóa

Sơ đồ luận lý



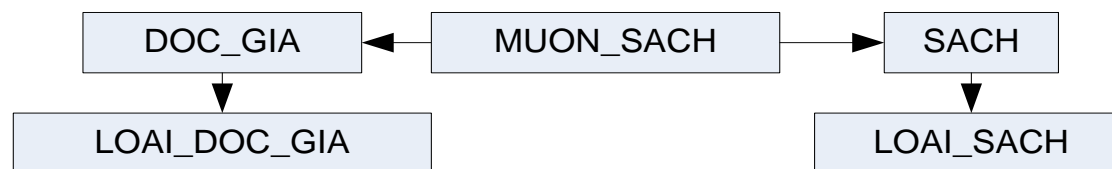
Chi tiết các bảng:

- ✓ **DOC_GIA**(MDG, MLDG, HoTen, NgaySinh, DiaChi, DienThoai, NgayHetHan)
- ✓ **SACH**(MSACH, Tensach, MTL, NgayNhap, TacGgia, NamXuatBan, NhaXuatBan)
- ✓ **MUON_SACH**(MDG, MSACH, NgayMuon, NgayTra, TienPhat)
- ✓ **THE_LOAI**(MTL, TenTheLoai, GhiChu)
- ✓ **LOAI_DG**(MLDG, TenLoaiDocGia, GhiChu)

4.3. QUÁ TRÌNH THIẾT KẾ

Thiết kế với tính hiệu quả (truy xuất nhanh)

Sơ đồ luận lý:



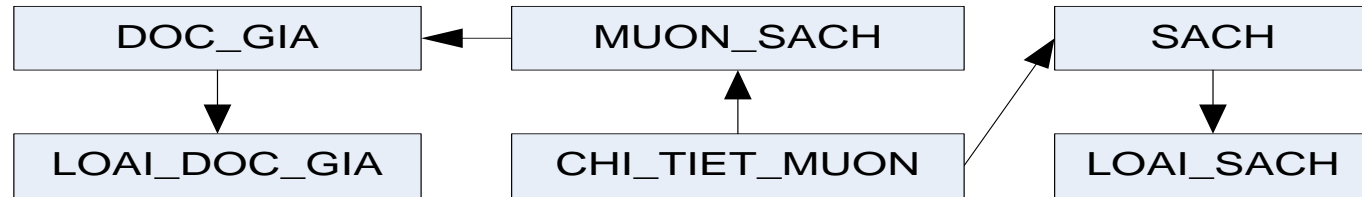
Bảng thuộc tính

- ✓ **DOC_GIA**(MDG, MLDG, HoTen, NgaySinh, DiaChi, DienThoai, NgayHetHan, SoSachMuon, TinhTrangTra)
- ✓ **SACH**(MSACH, TenSach, MTL, NgayNhap, TacGia, NamXuatBan, NhaXuatBan, TinhTrangMuon)
- ✓ **MUON_SACH**(MDG, MSACH, NgayMuon, NgayTra, TienPhat)
- ✓ **THE_LOAI**(MTL, TenTheLoai, GhiChu)
- ✓ **LOAI_DOC_GIA**(MLDG, TenLDG, GhiChu)

4.3. QUÁ TRÌNH THIẾT KẾ

Thiết kế dữ liệu với tính hiệu quả (lưu trữ tối ưu)

Sơ đồ luận lý



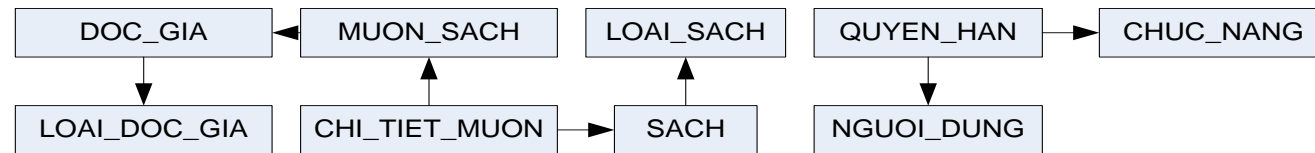
Chi tiết các bảng thuộc tính

- ✓ **DOC_GIA**(MDG, MLDG, HoTen, NgaySinh, DiaChi, DienThoai, NgayHetHan, SoSachMuon, TinhTrangTra)
- ✓ **SACH**(MSACH, Tensach, MTL, NgayNhap, TacGia, NamXuatBan, NhaXuanBan, TinhTrangMuon)
- ✓ **MUON_SACH**(MDG, MSACH, NgayMuon, NgayTra, TienPhat)
- ✓ **CHI_TIET_MUON**(MMUON, MSACH, NgayTra, TienPhat)
- ✓ **THE_LOAI**(MTL, TenTheLoai, GhiChu)
- ✓ **LOAI_DOC_GIA**(MLDG, TenLoaiDocGia, GhiChu)

4.3. QUÁ TRÌNH THIẾT KẾ

Thiết kế dữ liệu với yêu cầu phân quyền hệ thống

Sơ đồ luận lý



Chi tiết các bảng

- ✓ **DOC_GIA**(MDG, MLDG, HoTen, NgaySinh, DiaChi, DienThoai, NgayHetHan, SoSachMuon, TinhTrangTra)
- ✓ **SACH**(MSACH, Tensach, MTL, NgayNhap, TacGia, NamXuatBan, NhaXuanBan, TinhTrangMuon)
- ✓ **MUON_SACH**(MDG, MSACH, NgayMuon, NgayTra, TienPhat)
- ✓ **CHI_TIET_MUON**(MMUON, MSACH, NgayTra, TienPhat)
- ✓ **THE_LOAI**(MTL, TenTheLoai, GhiChu)
- ✓ **LOAI_DOC_GIA**(MLDG, TenLoaiDocGia, GhiChu)
- ✓ **NGUOI_DUNG**(MND, HoTen, Ghichu)
- ✓ **CHUC_NANG**(MCN, Ten_ChucNang, GhiChu)
- ✓ **QUYEN_HAN**(MND, MCN)

4.4. PHƯƠNG PHÁP THIẾT KẾ DỮ LIỆU

1. Phương pháp trực tiếp
2. Phương pháp gián tiếp

4.4.1. PHƯƠNG PHÁP TRỰC TIẾP

Bước 1

- ✓ Lập sơ đồ với 1 thành phần duy nhất,
- ✓ Đánh giá tính đúng đắn so với yêu cầu, chuyển sang bước 2 nếu cần.

Bước 2

- ✓ Tách 1 số thuộc tính để tạo ra các thành phần mới,
- ✓ Xác định liên kết giữa các thành phần,
- ✓ Đánh giá tính đúng đắn so với yêu cầu, lặp lại bước 2 nếu cần.

4.4.1. PHƯƠNG PHÁP TRỰC TIẾP

Cách 1: Dùng thành phần SACH

MSach, Ten, TheLoai, NgayMua, TacGia, NhaXuatBan,
NamXuatBan HoTenDocGia, LoaiDocGia, NgayLamThe,
NgayMuon, NgayTra

Cách 2: Dùng thành phần SACH, DOC_GIA

Cách 2.1: Chỉ lưu trữ lần mượn sách cuối cùng

- ✓ **SACH**(MSACH, MADG, Ten, TheLoai, NgayMua, TacGia, NhaXuatBan, NamXuatBan, NgayMuon, NgayTra)
- ✓ **DOC_GIA**(MDG, HoTen, LoaiDocGia, NgayLamThe)

4.4.1. PHƯƠNG PHÁP TRỰC TIẾP

Cách 2.2: Chỉ cho phép độc giả mượn tối đa 1 quyển sách

- ✓ **SACH**(MSACH, Ten, TheLoai, NgayMua, TacGia, NhaXB, NamXB, NgayMuon, NgayTra)
- ✓ **DOC_GIA**(MDG, MSACH, HoTen, LoaiDocGia, NgayLamThe, NgayMuon)

Cách 3: Dùng thành phần SACH, DOC_GIA, MUON_SACH

- ✓ **SACH**(MSACH, Ten, TheLoai, NgayMua, TacGia, NhaXuatBan, NamXuatBan, NgayMuon, NgayTra)
- ✓ **DOC_GIA**(MDG, HoTen, LoaiDocGia, NgayLamThe)
- ✓ **MUON_SACH**(MMUON, MDG, MSACH, NgayMuon, NgayTra)

4.4.2. PHƯƠNG PHÁP GIÁN TIẾP

Bước 1

- ✓ Lập sơ đồ lớp
- ✓ Xác định các lớp đối tượng
- ✓ Xác định quan hệ giữa các lớp đối tượng
- ✓ Lập sơ đồ.

Bước 2

- ✓ Ánh xạ từ sơ đồ lớp vào sơ đồ luận lý,
- ✓ Ánh xạ các lớp đối tượng,
- ✓ Ánh xạ các quan hệ giữa các lớp đối tượng.

Bước 3

- ✓ Hoàn chỉnh sơ đồ luận lý,
- ✓ Bổ sung các thành phần theo yêu cầu,
- ✓ Mô tả chi tiết các thuộc tính của các thành phần.

4.4.2.1. LẬP SƠ ĐỒ LỚP

Ví dụ: Với phần mềm QL thư viện, 2 đối tượng chính là Độc giả, Sách và quan hệ mượn sách.



4.4.2.2. ẢNH XẠ SƠ ĐỒ LỚP

Ảnh xạ lớp đối tượng sao cho mỗi đối tượng trong sơ đồ lớp tương ứng với 1 thành phần trong sơ đồ luận lý

Sơ đồ lớp:

DOC_GIA

SACH

Sơ đồ luận lý:

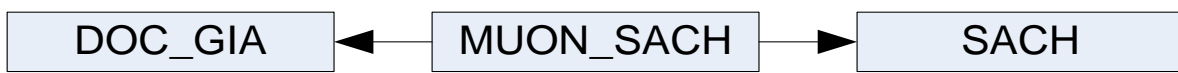
DOC_GIA

SACH

4.4.2.3. ẢNH XẠ QUAN HỆ

- ✓ Quan hệ 1- n : Biểu diễn liên kết giữa 2 lớp đối tượng A và B (1 A với nhiều B) xác định duy nhất từ A sang B trong sơ đồ luận lý.
- ✓ Quan hệ m - n : Biểu diễn liên kết giữa 2 lớp đối tượng A và B tương ứng với 1 thành phần C trong sơ đồ luận lý và xác định duy nhất với A, B.

Sơ đồ lớp: 

Sơ đồ luận lý: 

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Bổ sung các thành phần

- ✓ *Đối tượng phụ*: tương ứng với 1 thành phần trong sơ đồ luận lý
- ✓ Các thành phần khác: Xét tính đúng đắn và bổ sung thêm nếu cần thiết.

Mô tả chi tiết thuộc tính các thành phần

- ✓ Thuộc tính *khóa chính*:
- ✓ Mỗi thành phần ứng với đối tượng (chính hoặc phụ) cần 1 thuộc tính khóa riêng,
- ✓ Các thành phần còn lại, tùy theo ý nghĩa sử dụng, sẽ có thuộc tính khóa riêng hay dùng tổ hợp thuộc tính khóa của các thành phần khác.

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

- ✓ Ví dụ: Các thành phần DOC_GIA, SACH, NHA_XUAT_BAN, TAC_GIA sẽ có khóa chính MDG, MSACH, MNXB, MTG. Thành phần MUON có khóa chính là MMUON.
- ✓ Thuộc tính *khóa ngoại*: Thể hiện đúng liên kết giữa các thành phần trong sơ đồ luận lý: nếu A xác định duy nhất B thì A có thuộc tính là khoá chính của B (là khóa ngoại của A)
- ✓ Ví dụ: Thành phần MUON có 2 khóa ngoại: MDG, MSACH; Thành phần SACH có 2 khoá ngoại: MNXB, MTG, MDG
- ✓ *Các thuộc tính khác*: Cần dựa vào yêu cầu lưu trữ (Tên, Sự phân loại, ngày tháng, vị trí, độ đo, tính chất ...)

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Ví dụ:

- ✓ DOC_GIA có thuộc tính khác như: HoTen, LoaiDG, Ngaysinh, Ngayhethan, Diachi;
- ✓ SACH có thuộc tính khác như: TenSach, LoaiSach, NgayMua, GiaTien

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Thiết kế dữ liệu với tính đúng đắn

Các bước thực hiện:

- ✓ Bước 1: Chọn một yêu cầu và xác định sơ đồ luận lý cho yêu cầu.
- ✓ Bước 2: Bổ sung thêm một yêu cầu và xem xét lại sơ đồ luận lý, nếu sơ đồ luận lý vẫn đáp ứng được thì tiếp tục bước 3, ngược lại thì bổ sung vào sơ đồ thuộc tính mới cùng với các thuộc tính và liên kết tương ứng.
- ✓ Bước 3: Quay lại bước 2 cho đến khi đã xem xét đầy đủ các yêu cầu dựa trên

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Ghi chú:

- ✓ Với mỗi yêu cầu cần xác định rõ cần lưu trữ các thông tin 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,
- ✓ 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 luận lý nhiều thành phần mới.
- ✓ Thuộc tính *khóa* của các thành phần phải dựa trên ngữ nghĩa tương ứng trong thế giới thực.

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Thiết kế dữ liệu và yêu cầu chất lượng

Mục tiêu:

- ✓ Xem xét đánh giá sơ đồ luận lý theo các yêu cầu về chất lượng rồi cập nhật lại sơ đồ để bảo đảm các tiêu chuẩn về chất lượng.
- ✓ Tính đúng đắn cần ưu tiên hàng đầu, và cần chú ý sự hơn kém nhau giữa các phần mềm, mức độ thỏa mãn các tiêu chuẩn chất lượng (đặc biệt là tính tiến hóa).

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Xem xét tính tiến hóa

- ✓ Để bảo đảm tính tiến hóa, sơ đồ luận lý sẽ được bổ sung cập nhật lại nhiều thành phần qua các bước thiết kế chi tiết.
- ✓ Trong bước thiết kế dữ liệu, ta sẽ giới hạn xem xét đến các *thuộc tính có giá trị rời rạc*, tương ứng khi *miền giá trị* chỉ bao gồm *một số giá trị nhất định*.
- ✓ Các giá trị này thông thường thuộc về tập hợp có độ biến động rất ít trong quá trình sử dụng phần mềm.

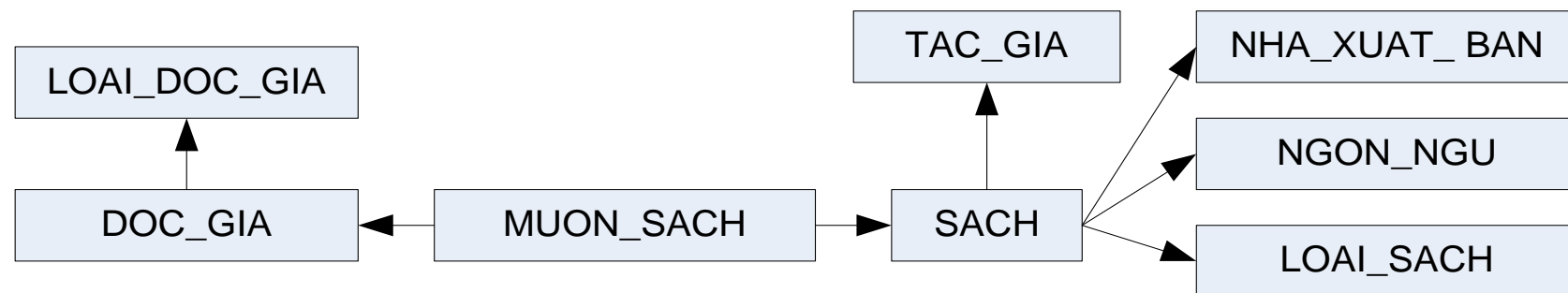
4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Ví dụ:

- ✓ LOAI_DOC_GIA: Có 3 loại độc giả là 'A', 'B', 'C' và khả năng có thêm loại độc giả mới rất thấp.
- ✓ Ngôn ngữ : Chỉ có 3 loại ngôn ngữ 'Việt', 'Anh', 'Pháp' và khả năng thêm sách thuộc ngôn ngữ mới rất thấp.
- ✓ Cần lưu ý khả năng biến động trên tập hợp giá trị của thuộc tính rời rạc là thấp nhưng không phải là không có. Nếu không chuẩn bị trước trong thiết kế có thể một số chức năng sẽ không thực hiện được (ví dụ sẽ không thể thêm sách mới với ngôn ngữ tiếng Hoa).
- ✓ Để chuẩn bị tốt, sẽ tách các thuộc tính này thành một thành phần trong sơ đồ luận lý. Khi đó hoàn toàn có thể cập nhật lại tập hợp các giá trị này tương ứng với các biến động.

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Sơ đồ luận lý khi tách các thuộc tính rời rạc như sau:



4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Xem xét tính hiệu quả (tốc độ)

- ✓ Phạm vi xem xét: Giới hạn xem xét việc tăng tốc độ thực hiện của phần mềm bằng cách bổ sung thêm các thuộc tính vào các bảng dùng lưu trữ các thông tin đã tính toán trước (Ví dụ: sổ sách đang mượn của độc giả)
- ✓ Các thông tin này phải được tự động cập nhật khi có bất kỳ thay đổi thông tin gốc liên quan (Độc giả mượn thêm hoặc trả sách)

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Các bước tiến hành:

Bước 1: Chọn một yêu cầu và xem xét cần bổ sung thông tin gì trên bộ nhớ phụ để tăng tốc độ thực hiện (các thông tin xử lý phải đọc mà không cần tính toán)

Bước 2: Làm lại bước 1 đến khi đã xét đầy đủ các yêu cầu tiếp theo

Ghi chú: Sau mỗi bước, phải lập DS các thuộc tính tính toán cùng với thông tin liên quan thông tin gốc, xử lý tự động cập nhật thông tin gốc. Nếu thông tin gốc thường xuyên bị thay đổi, việc bổ sung thuộc tính tính toán để tăng tốc độ thực hiện sẽ mất ý nghĩa và hiệu ứng xấu.

Việc làm tăng tốc độ truy xuất sẽ dẫn đến việc lưu trữ không tối ưu. Thứ tự xem xét các yêu cầu theo thứ tự từ đầu đến cuối (không cần chọn như các bước trong thiết kế dữ liệu)

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Xem xét tính hiệu quả (lưu trữ):

- ✓ Tính hiệu quả trong thiết kế dữ liệu sẽ được xem xét dưới góc độ lưu trữ tối ưu.
- ✓ Vấn đề đặt ra là xây dựng sơ đồ luận lý sao cho vẫn bảo đảm lưu trữ đầy đủ thông tin theo yêu cầu nhưng với dung lượng lưu trữ nhỏ nhất có thể có.
- ✓ Đặc biệt quan trọng với các PM với hệ thống lưu trữ lớn và nhiều phát sinh cần lưu trữ theo thời gian.
- ✓ Khi đó ta cần đặc biệt quan tâm đến các thành phần mà dữ liệu tương ứng được phát sinh nhiều theo thời gian.
- ✓ Ta sẽ tìm cách bố trí lại sơ đồ luận lý sao cho vẫn đảm bảo thông tin mà dung lượng lưu trữ lại ít hơn.

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Các bước tiến hành:

Bước 1:

- ✓ Lập danh sách các bảng cần được xem xét để tối ưu hóa việc lưu trữ, cụ thể là:
 - Xem xét và xác định các công việc có tần suất thực hiện thường xuyên, và bổ sung chúng 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.

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Bước 2:

- ✓ Tối ưu việc lưu trữ các bảng có khối lượng dữ liệu lưu trữ lớn thông qua việc tối ưu lưu trữ từng thuộc tính: Xác định các thuộc tính mà việc lưu trữ chưa tối ưu và ưu tiên xem xét các thuộc tính có kiểu chuỗi,
- ✓ Tối ưu việc lưu trữ tùy theo từng trường hợp cụ thể, 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 (ví dụ thuộc tính TacGia, NhaXuatBan trong bảng SACH)
- ✓ Việc tối ưu hoá có thể thực hiện thông qua việc bổ sung các bảng mới (bảng TAC_GIA, NHA_XUAT_BAN) và cấu trúc lại bảng SACH (thay thuộc tính TAC_GIA bằng MTG, NHA_XB bằng MNXB)

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Bước 3:

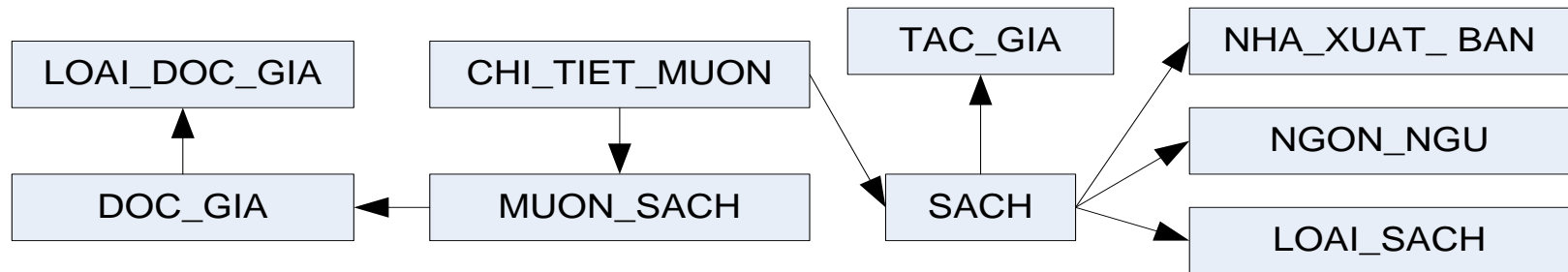
- ✓ Tối ưu hóa các bảng mà khóa gồm nhiều thuộc tính.
- ✓ Phân rã bảng thành hai bảng, một bảng chứa các thuộc tính mà giá trị được lặp lại nhiều lần trong cùng một lần thực hiện công việc.
- ✓ Bảng này cần có khóa riêng (được bảng còn lại sử dụng để tham chiếu).

Lưu ý:

- ✓ Việc phân rã giúp cho lưu trữ tối ưu, tuy nhiên: Tốc độ truy xuất có thể sẽ chậm hơn, Việc xử lý khó khăn nên Cần cân nhắc trước khi thực hiện phân rã
- ✓ Việc đánh giá khóa riêng cho bảng đã phân ra có thể cần kiểm tra thêm số phụ thuộc hàm

4.4.2.4. HOÀN CHỈNH SƠ ĐỒ LUẬN LÝ

Ví dụ: Xét phần mềm quản lý thư viện.



Mô tả chi tiết các thuộc tính: **Xem chi tiết phụ lục B**

TÓM TẮT

Quan tâm đến những vấn đề liên quan đến kỹ thuật thiết kế dữ liệu như sau:

- ✓ Quá trình thiết kế
- ✓ Phương pháp thiết kế dữ liệu trực tiếp / gián tiếp
 - Lập sơ đồ lớp
 - Ánh xạ sơ đồ lớp
 - Ánh xạ quan hệ
 - Hoàn chỉnh sơ đồ luận lý.

BÀI TẬP

1. Phụ lục A trang 170
2. Phụ lục B trang 179