

**ĐẠI HỌC FPT
FPT UNIVERSITY**



BÁO CÁO MÔN HỌC

**XÂY DỰNG HỆ THỐNG CSDL
QUẢN LÝ THƯ VIỆN**

**(BUILDING A DATABASE SYSTEM FOR
LIBRARY MANAGEMENT)**

MÔN HỌC: HỆ THỐNG CƠ SỞ DỮ LIỆU
MÃ MÔN: DBI202
LỚP: SE2043

GIẢNG VIÊN HƯỚNG DẪN:
TS. VŨ THANH PHONG

NHÓM THỰC HIỆN (NHÓM 3):

1. Nguyễn Ngọc Phúc
2. Thân Nhật Huy
3. Võ Hoàng Đình Trường
4. Nguyễn Thành An
5. Nguyễn Quang Thiên Phú
6. Phạm Ngọc Hưng

TP. HỒ CHÍ MINH, THÁNG 01 NĂM 2026

Mục lục

1	GIỚI THIỆU TỔNG QUAN	2
1.1	Phân tích bối cảnh và vấn đề bài toán	2
1.1.1	Bối cảnh bài toán	2
1.1.2	Vấn đề cần giải quyết	2
1.2	Yêu cầu chức năng từ người dùng	2
1.2.1	Quản lý sách (Book Management)	2
1.2.2	Quản lý tác giả (Author Management)	3
1.2.3	Quản lý nhà xuất bản (Publisher Management)	3
1.2.4	Quản lý bản sao sách (Book Copy Management)	3
1.2.5	Quản lý thành viên (Member Management)	4
1.2.6	Quản lý mượn/trả sách (Loan Management)	4
1.2.7	Báo cáo thống kê (Reporting Requirements)	4
1.3	Phạm vi dự án và công cụ sử dụng	5
1.3.1	Phạm vi dự án	5
1.3.2	Công cụ sử dụng	6
2	BÁO CÁO 1: PHÂN RÃ	6
2.1	Phân công nhiệm vụ	6
2.2	Cơ sở lý thuyết	6
2.2.1	Khái niệm về Phân rã (Decomposition)	6
2.2.2	Các thành phần chính trong mô hình E/R	7
2.2.3	Quy trình xác định E/R Diagram	8
2.3	Tập thực thể	8
2.3.1	BOOK (SÁCH)	8
2.3.2	AUTHOR (TÁC GIẢ)	8
2.3.3	PUBLISHER (NHÀ XUẤT BẢN)	9
2.3.4	BOOK_COPY (BẢN SAO SÁCH)	9
2.3.5	MEMBER (THÀNH VIÊN)	9
2.3.6	LOAN (MƯỢN/TRẢ)	10
2.3.7	Tổng hợp thực thể	10
2.4	Các mối quan hệ	10
2.5	Thuộc tính chi tiết	11
2.5.1	Bảng trung gian BOOK_AUTHOR	11
2.6	Sơ đồ Thực thể - Mối quan hệ (ERD)	12
2.6.1	Sơ đồ ERD hoàn chỉnh	12
2.6.2	Mô tả chi tiết các mối quan hệ	12
2.6.3	Các quyết định thiết kế chính	14
2.7	Kết luận báo cáo 1	14
2.7.1	Tổng kết kết quả	14
3	BÁO CÁO 2: MÔ HÌNH QUAN HỆ (RELATIONAL DATA MODEL)	16
3.1	Giới thiệu báo cáo 2	16
3.2	Cơ sở lý thuyết	16
3.2.1	Mô hình Quan hệ (Relational Model)	16
3.2.2	Khóa chính (Primary Key)	16
3.2.3	Khóa ngoại (Foreign Key)	17
3.2.4	Phụ thuộc hàm (Functional Dependency)	17

3.3	Quy tắc chuyển đổi từ ERD sang Mô hình Quan hệ	18
3.3.1	Quy tắc cơ bản	18
3.3.2	Quy tắc chi tiết cho từng loại quan hệ	18
3.4	Chuyển đổi ERD sang Lược đồ Quan hệ	19
3.4.1	Lược đồ quan hệ cho các thực thể cơ bản	19
3.4.2	Bảng trung gian cho quan hệ M:N	21
3.5	Tổng hợp Lược đồ Quan hệ Hệ Thống	22
3.5.1	Tất cả Lược đồ Quan hệ	22
3.5.2	Sơ đồ Mô hình Quan hệ (Relational Data Model Diagram)	23
3.6	Phụ thuộc Hàm (Functional Dependencies)	23
3.6.1	Định nghĩa và Ký hiệu	23
3.6.2	Các Phụ thuộc Hàm cho từng Bảng	24
3.6.3	Tổng hợp các Phụ thuộc Hàm	25
3.7	Đánh giá Mô hình Quan hệ	26
3.7.1	Tính toàn vẹn dữ liệu (Data Integrity)	26
3.7.2	Ưu điểm của Mô hình Quan hệ hiện tại	26
3.7.3	Hạn chế và Cải tiến tiềm năng	26
3.8	Nhận dạng các Bất thường (Anomaly Recognition)	27
3.8.1	Mục đích của việc nhận dạng bất thường	27
3.8.2	Phân tích sơ bộ các bất thường	27
3.8.3	Kết luận sơ bộ	27
3.9	Kết luận báo cáo 2	28
3.9.1	Tổng kết kết quả	28
4	BÁO CÁO 3: TRỪU TƯỢNG HÓA (NORMALIZATION)	30
4.1	Phân công nhiệm vụ	30
4.2	Giới thiệu báo cáo 3	30
4.3	Cơ sở lý thuyết	30
4.3.1	Khái niệm về Chuẩn hóa (Normalization)	30
4.3.2	Ba mức chuẩn hóa chính	31
4.3.3	Các loại Phụ thuộc Hàm (Functional Dependencies)	31
4.4	Phân tích Phụ thuộc Hàm chi tiết	32
4.4.1	Tổng hợp các Phụ thuộc Hàm từ Báo cáo 2	32
4.5	Kiểm tra First Normal Form (1NF)	32
4.5.1	Định nghĩa 1NF	32
4.5.2	Phân tích 1NF cho từng bảng	33
4.5.3	Tổng kết kiểm tra 1NF	35
4.6	Kiểm tra Second Normal Form (2NF)	35
4.6.1	Định nghĩa 2NF	35
4.6.2	Phân tích 2NF cho từng bảng	36
4.6.3	Tổng kết kiểm tra 2NF	36
4.7	Kiểm tra Third Normal Form (3NF)	37
4.7.1	Định nghĩa 3NF	37
4.7.2	Phân tích 3NF cho từng bảng	37
4.7.3	Tổng kết kiểm tra 3NF	40
4.8	Tổng hợp Kết quả Chuẩn hóa	40
4.8.1	Tổng quan toàn bộ quá trình chuẩn hóa	40
4.8.2	Kết luận chung	40

4.9	Xác định các Bất thường (Anomaly Analysis)	41
4.9.1	Các loại bất thường trong cơ sở dữ liệu	41
4.9.2	Kết luận về các bất thường	42
4.10	Các quyết định Thiết kế và Lý do	42
4.10.1	Quyết định 1: Tách BOOK và BOOK_COPY	42
4.10.2	Quyết định 2: Sử dụng bảng trung gian BOOK_AUTHOR	42
4.10.3	Quyết định 3: Thực thể LOAN riêng biệt	43
4.11	So sánh với thiết kế chưa chuẩn hóa	43
4.11.1	Ví dụ: Thiết kế SAI (vi phạm 3NF)	43
4.11.2	Thiết kế ĐÚNG (đạt 3NF)	44
4.12	Kết luận báo cáo 3	44
4.12.1	Tổng kết kết quả	44
4.12.2	Kết luận cuối cùng	45
5	BÁO CÁO 4: THIẾT KẾ THUẬT TOÁN (ALGORITHM DESIGN)	46
5.1	Phân công nhiệm vụ	46
5.2	Giới thiệu báo cáo 4	46
5.3	Cơ sở lý thuyết	47
5.3.1	Các loại Ràng buộc (Constraints) trong SQL Server	47
5.3.2	Stored Procedures và Functions	47
5.4	Triển khai cơ sở dữ liệu trên SQL Server	47
5.4.1	Tạo cơ sở dữ liệu	47
5.4.2	Xóa bảng nếu tồn tại (để tránh lỗi)	48
5.5	Tạo bảng với đầy đủ ràng buộc	48
5.5.1	Bảng PUBLISHER (Nhà xuất bản)	48
5.5.2	Bảng AUTHOR (Tác giả)	48
5.5.3	Bảng MEMBER (Thành viên)	49
5.5.4	Bảng BOOK (Sách)	50
5.5.5	Bảng BOOK_COPY (Bản sao sách)	50
5.5.6	Bảng LOAN (Mượn/trả)	51
5.5.7	Bảng trung gian BOOK_AUTHOR	52
5.6	Tóm tắt các bảng và ràng buộc	53
5.7	Stored Procedures	53
5.7.1	Procedure 1: Mượn sách	53
5.7.2	Procedure 2: Trả sách	54
5.8	Functions	56
5.8.1	Function 1: Lấy danh sách sách của tác giả	56
5.8.2	Function 2: Tính phí phạt quá hạn	56
5.9	Triggers	56
5.9.1	Trigger 1: Ngăn mượn sách nếu thành viên quá hạn	56
5.9.2	Trigger 2: Tự động tính phí phạt khi trả sách	57
5.10	Các câu truy vấn mẫu	58
5.10.1	Query 1: Liệt kê sách đang được mượn	58
5.10.2	Query 2: Thống kê sách được mượn nhiều nhất	58
5.10.3	Query 3: Thành viên có sách quá hạn	59
5.11	Kết luận báo cáo 4	59
5.11.1	Tổng kết kết quả	59
5.11.2	Đánh giá và kế tiếp	60

6	BÁO CÁO 5: BÁO CÁO ĐỒ ÁN CUỐI KỲ	61
6.1	Phân công nhiệm vụ	61
6.2	Giới thiệu báo cáo 5	61
6.3	Chèn dữ liệu mẫu (Sample Data)	61
6.3.1	Dữ liệu bảng PUBLISHER	61
6.3.2	Dữ liệu bảng AUTHOR	62
6.3.3	Dữ liệu bảng MEMBER	62
6.3.4	Dữ liệu bảng BOOK	63
6.3.5	Dữ liệu bảng BOOK_AUTHOR	63
6.3.6	Dữ liệu bảng BOOK_COPY	64
6.3.7	Dữ liệu bảng LOAN	65
6.3.8	Tổng kết dữ liệu đã chèn	65
6.4	Kiểm tra Stored Procedures	66
6.4.1	Test 1: Mượn sách thành công	66
6.4.2	Test 2: Trả sách đúng hạn	66
6.4.3	Test 3: Trả sách quá hạn	66
6.4.4	Test 4: Mượn sách khi sách không có sẵn	67
6.5	Kiểm tra Functions	67
6.5.1	Test Function 1: ufn_GetBooksByAuthor	67
6.5.2	Test Function 2: ufn_CalculateOverdueFine	67
6.6	Kiểm tra Triggers	67
6.6.1	Test Trigger 1: Ngăn mượn nếu thành viên có sách quá hạn	67
6.6.2	Test Trigger 2: Tự động tính phí phạt	68
6.7	Các truy vấn báo cáo thống kê	68
6.7.1	Báo cáo 1: Top 10 sách được mượn nhiều nhất	68
6.7.2	Báo cáo 2: Thành viên có sách quá hạn	69
6.7.3	Báo cáo 3: Thống kê theo nhà xuất bản	69
6.7.4	Báo cáo 4: Doanh thu phí phạt theo tháng	70
6.8	Final Thoughts - Tổng kết và Đánh giá	70
6.8.1	Tổng kết toàn bộ dự án	70
6.8.2	Những bài học kinh nghiệm	71
6.8.3	Hạn chế và cải tiến tiềm năng	71
6.8.4	Kết luận	72

1. GIỚI THIỆU TỔNG QUAN

1.1. Phân tích bối cảnh và vấn đề bài toán

1.1.1. Bối cảnh bài toán

Một thư viện cần xây dựng một hệ thống cơ sở dữ liệu để lưu trữ và quản lý toàn bộ thông tin liên quan đến các hoạt động quản lý sách và dịch vụ cho độc giả. Hệ thống này phải đảm bảo khả năng theo dõi thông tin chi tiết về các cuốn sách (BOOK), các tác giả (AUTHOR), nhà xuất bản (PUBLISHER), bản sao sách (BOOK_COPY), và thành viên thư viện (MEMBER). Dữ liệu bao gồm thông tin xuất bản, thông tin mượn/trả sách, tình trạng sách, và các khoản phí phạt khi trả sách muộn.

1.1.2. Vấn đề cần giải quyết

Bài toán đặt ra là phải thiết kế một mô hình cơ sở dữ liệu quan hệ có khả năng biểu diễn chính xác các mối quan hệ phức tạp giữa các thực thể nói trên. Cụ thể:

- Mỗi cuốn sách có thể có nhiều tác giả, và mỗi tác giả có thể viết nhiều sách (mối quan hệ nhiều-nhiều).
- Mỗi cuốn sách có thể có nhiều bản sao, và mỗi bản sao có thể được mượn bởi một thành viên tại một thời điểm.
- Mỗi cuốn sách được xuất bản bởi một nhà xuất bản duy nhất.
- Thành viên thư viện có thể mượn nhiều sách, và mỗi lần mượn cần lưu trữ thông tin ngày mượn, ngày hết hạn, ngày trả, và khoản phí phạt (nếu có).
- Hệ thống phải hỗ trợ các báo cáo: sách được mượn nhiều nhất, lịch sử mượn sách của thành viên, các sách quá hạn, và thống kê sách theo tác giả hoặc nhà xuất bản.

Do đó, yêu cầu cốt lõi của bài toán là phân tích, xác định các thực thể, thuộc tính, và mối quan hệ giữa chúng, sau đó chuẩn hóa dữ liệu để đảm bảo tính toàn vẹn, tránh dư thừa và hỗ trợ truy vấn hiệu quả cho hoạt động quản lý của thư viện.

1.2. Yêu cầu chức năng từ người dùng

Dựa trên nhu cầu quản lý thư viện, cơ sở dữ liệu phải hỗ trợ các chức năng sau:

1.2.1. Quản lý sách (Book Management)

Yêu cầu:

- Lưu trữ thông tin chi tiết về các cuốn sách, bao gồm:
 - Mã sách (*BookID*) - định danh duy nhất
 - Tên sách (*Title*)
 - Thể loại/Sự phân loại (*SubjectCategory*)

- Năm xuất bản (*PublicationYear*)
- Nhà xuất bản (*Publisher*)

Mục đích: Hệ thống có thể tra cứu, tìm kiếm và quản lý danh mục sách hiệu quả.

1.2.2. Quản lý tác giả (Author Management)

Yêu cầu:

- Lưu trữ thông tin chi tiết về các tác giả:
 - Mã tác giả (*AuthorID*) - định danh duy nhất
 - Tên tác giả (*Name*)
 - Quốc tịch (*Nationality*)
 - Năm sinh (*YearOfBirth*)

Mục đích: Theo dõi danh mục tác giả và các tác phẩm của họ.

1.2.3. Quản lý nhà xuất bản (Publisher Management)

Yêu cầu:

- Lưu trữ thông tin về các nhà xuất bản:
 - Mã nhà xuất bản (*PublisherID*) - định danh duy nhất
 - Tên nhà xuất bản (*Name*)
 - Địa chỉ (*Address*)
 - Số điện thoại liên hệ (*ContactNumber*)

Mục đích: Quản lý thông tin đối tác cung cấp sách.

1.2.4. Quản lý bản sao sách (Book Copy Management)

Yêu cầu:

- Lưu trữ thông tin về từng bản sao vật lý của sách:
 - Mã bản sao (*CopyID*) - định danh duy nhất
 - Tình trạng (*Condition*) - mới, cũ, hư hại
 - Trạng thái (*Status*) - có sẵn, đang mượn, bảo dưỡng

Mục đích: Theo dõi chính xác số lượng sách sẵn sàng cho mượn.

1.2.5. Quản lý thành viên (Member Management)

Yêu cầu:

- Lưu trữ thông tin thành viên thư viện:
 - Mã thành viên (*MemberID*) - định danh duy nhất
 - Tên thành viên (*Name*)
 - Địa chỉ (*Address*)
 - Số điện thoại (*PhoneNumber*)
 - Loại thành viên (*MembershipType*) - sinh viên, giảng viên, độc giả thường

Mục đích: Quản lý thông tin độc giả và phân loại quyền lợi.

1.2.6. Quản lý mượn/trả sách (Loan Management)

Yêu cầu:

- Lưu trữ chi tiết các giao dịch mượn/trả:
 - Mã mượn (*LoanID*) - định danh duy nhất
 - Ngày mượn (*BorrowDate*)
 - Ngày hết hạn (*DueDate*)
 - Ngày trả (*ReturnDate*)
 - Phí phạt quá hạn (*OverdueFine*) - tính theo ngày

Mục đích: Theo dõi luồng sách mượn/trả và tính toán phí phạt.

1.2.7. Báo cáo thống kê (Reporting Requirements)

Yêu cầu:

- **Sách được mượn nhiều nhất:** Tổng hợp số lần mượn của từng sách
- **Lịch sử mượn của thành viên:** Xem tất cả sách một thành viên đã mượn
- **Sách quá hạn:** Liệt kê sách chưa trả và quá hạn
- **Thống kê theo tác giả:** Số lượng sách và lượt mượn theo từng tác giả
- **Thống kê theo nhà xuất bản:** Số lượng sách theo từng nhà xuất bản

Mục đích: Hỗ trợ quản lý ra quyết định về mua sách, điều chỉnh quy định, và cải thiện dịch vụ.

1.3. Phạm vi dự án và công cụ sử dụng

1.3.1. Phạm vi dự án

Trong phạm vi dự án (In Scope):

- Phân tích yêu cầu và thiết kế sơ đồ ERD
- Chuyển đổi ERD sang mô hình quan hệ (Relational Data Model)
- Chuẩn hóa cơ sở dữ liệu đến 3NF (Third Normal Form)
- Định nghĩa lược đồ cơ sở dữ liệu với các ràng buộc (Constraints)
- Triển khai cơ sở dữ liệu trên SQL Server
- Thêm dữ liệu mẫu (Sample Data)
- Tạo các stored procedures, functions, và triggers
- Viết các câu lệnh SQL truy vấn cho báo cáo

Ví dụ về các câu truy vấn SQL cần triển khai:

- Liệt kê tất cả các tác phẩm của một tác giả cụ thể
- Tìm các thành viên đang mượn sách quá hạn
- Thống kê sách được mượn nhiều nhất
- Tính tổng phí phạt của một thành viên
- Lấy lịch sử mượn sách của một thành viên

Ngoài phạm vi dự án (Out of Scope):

- Giao diện người dùng đồ họa (GUI/Web Application)
- Hệ thống thanh toán điện tử
- Quản lý kho phức tạp (đặt hàng, nhập hàng)
- Hệ thống thông báo tự động (email/SMS)
- Phân tích dữ liệu nâng cao (Data Analytics)
- Quản lý người dùng và phân quyền chi tiết

1.3.2. Công cụ sử dụng

Bảng 1: Công cụ và công nghệ sử dụng

Công cụ	Mục đích	Lý do chọn
SQL Server	Hệ quản trị CSDL	Đáng tin cậy, hỗ trợ T-SQL đầy đủ
SSMS	Công cụ phát triển	Giao diện trực quan, hỗ trợ debug T-SQL
draw.io	Tạo sơ đồ ERD	Công cụ trực quan, hỗ trợ AI
T-SQL	Ngôn ngữ truy vấn	Ngôn ngữ chuẩn của SQL Server
Git/GitHub	Quản lý phiên bản	Theo dõi thay đổi mã nguồn
Word/Markdown	Tài liệu hóa	Tạo báo cáo và tài liệu kỹ thuật

2. BÁO CÁO 1: PHÂN RÃ

2.1. Phân công nhiệm vụ

Bảng 2: Phân công nhiệm vụ nhóm

STT	Họ và tên	Phân công nhiệm vụ	Điểm
1	Nguyễn Ngọc Phúc	Phân tích yêu cầu hệ thống, tham gia thiết kế ERD, xây dựng các thực thể và mối quan hệ, tổng hợp nội dung báo cáo	10
2	Thân Nhật Huy	Phân tích yêu cầu hệ thống, tham gia thiết kế ERD, xây dựng các thực thể và mối quan hệ, hỗ trợ viết báo cáo	10
3	Võ Hoàng Đình Trường	Phân tích yêu cầu hệ thống, tham gia thiết kế ERD, xác định thuộc tính và ràng buộc dữ liệu, hỗ trợ kiểm tra tài liệu	10
4	Nguyễn Thành An	Phân tích yêu cầu hệ thống, tham gia thiết kế ERD, xác định mối quan hệ giữa các thực thể, hỗ trợ hoàn thiện báo cáo	10
5	Nguyễn Quang Thiên Phú	Phân tích yêu cầu hệ thống, tham gia thiết kế ERD, chuẩn hóa mô hình dữ liệu, rà soát nội dung báo cáo	10
6	Phạm Ngọc Hưng	Phân tích yêu cầu hệ thống, tham gia thiết kế ERD, kiểm tra tính nhất quán của mô hình, tổng hợp và chỉnh sửa báo cáo	10

2.2. Cơ sở lý thuyết

2.2.1. Khái niệm về Phân rã (Decomposition)

Phân rã trong thiết kế cơ sở dữ liệu là quá trình đầu tiên và quan trọng nhất trong việc chuyển đổi các yêu cầu nghiệp vụ phức tạp thành một mô hình dữ liệu có cấu trúc. Quá trình này bao gồm:

1. Xác định Thực thể (Entity Identification):

- Nhận diện các đối tượng chính trong hệ thống cần quản lý
- Mỗi thực thể đại diện cho một danh mục (category) của dữ liệu
- Ví dụ: BOOK (Sách), MEMBER (Thành viên), LOAN (Mượn/trả)

2. Xác định Thuộc tính (Attribute Identification):

- Mô tả các đặc điểm chi tiết của từng thực thể
- Phân loại thuộc tính: đơn (simple), phức hợp (composite), đa trị (multi-valued), dẫn xuất (derived)
- Xác định khóa chính (Primary Key) để định danh duy nhất

3. Xác định Mối quan hệ (Relationship Identification):

- Mô tả sự liên kết giữa các thực thể
- Xác định bội số (cardinality): 1-1, 1-N, N-M
- Xác định tính chất (optionality): bắt buộc hoặc tùy chọn

Kết quả: Sơ đồ Thực thể - Mối quan hệ (Entity-Relationship Diagram - ERD), giúp trực quan hóa cấu trúc dữ liệu trước khi chuyển đổi sang mô hình quan hệ.

2.2.2. Các thành phần chính trong mô hình E/R

1. Entity (Thực thể):

- **Định nghĩa:** Là tập hợp các đối tượng có cùng đặc điểm và thuộc tính
- **Biểu diễn:** Hình chữ nhật trong ERD
- **Đặc điểm:** Có tên duy nhất, có khóa chính (Primary Key) để định danh

2. Attribute (Thuộc tính):

- **Định nghĩa:** Mô tả các đặc trưng của thực thể hoặc mối quan hệ
- **Biểu diễn:** Hình ellip trong ERD
- **Phân loại:**

Bảng 3: Phân loại thuộc tính

Loại	Mô tả	Ví dụ
Simple (Đơn)	Không thể chia nhỏ	Title, Year
Composite (Phức hợp)	Có thể chia nhỏ	Address = Street + City
Multi-valued (Đa trị)	Có nhiều giá trị	Phone numbers
Derived (Dẫn xuất)	Giá trị tính từ thuộc tính khác	Age = Year - BirthYear
Key (Khóa)	Định danh duy nhất	BookID, ISBN

3. Relationship (Mối quan hệ):

- **Định nghĩa:** Mô tả sự liên kết giữa hai hoặc nhiều thực thể
- **Biểu diễn:** Hình thoi (diamond) trong ERD
- **Bội số (Cardinality):**

Bảng 4: Các loại bội số

Loại	Ý nghĩa
One-to-One (1:1)	Một bản ghi của A liên kết với một bản ghi của B
One-to-Many (1:N)	Một bản ghi của A liên kết với nhiều bản ghi của B
Many-to-Many (N:M)	Nhiều bản ghi của A liên kết với nhiều bản ghi của B

2.2.3. Quy trình xác định E/R Diagram

- Bước 1:** Phân tích yêu cầu bài toán - Đọc và hiểu rõ yêu cầu nghiệp vụ
Bước 2: Xác định các thực thể chính - Tìm các danh từ chính trong yêu cầu
Bước 3: Xác định mối quan hệ - Tìm các động từ mô tả sự liên kết
Bước 4: Liệt kê thuộc tính - Tìm các tính từ mô tả đặc điểm
Bước 5: Vẽ sơ đồ E/R - Sử dụng công cụ: draw.io, Visual Paradigm
Bước 6: Kiểm tra và hoàn thiện - Rà soát lại các yêu cầu

2.3. Tập thực thể

Sau khi phân tích yêu cầu bài toán, chúng tôi xác định được **6 thực thể chính** cần quản lý trong hệ thống thư viện:

2.3.1. BOOK (SÁCH)

Mô tả thực thể: Lưu trữ thông tin về các cuốn sách trong thư viện

Bảng 5: Thuộc tính thực thể BOOK

Thuộc tính	Kiểu dữ liệu	Mô tả	Ràng buộc
<i>BookID</i>	INT	Mã định danh duy nhất	PK, NOT NULL
<i>ISBN</i>	VARCHAR(17)	Mã số tiêu chuẩn quốc tế	UNIQUE
<i>Title</i>	VARCHAR(255)	Tên tiêu đề của sách	NOT NULL
<i>SubjectCategory</i>	VARCHAR(100)	Thể loại phân loại sách	NULL
<i>PublicationYear</i>	INT	Năm xuất bản sách	NULL
<i>PublisherID</i>	INT	Mã nhà xuất bản	FK

2.3.2. AUTHOR (TÁC GIẢ)

Mô tả thực thể: Lưu trữ thông tin cá nhân về các tác giả của sách

Bảng 6: Thuộc tính thực thể AUTHOR

Thuộc tính	Kiểu dữ liệu	Mô tả	Ràng buộc
<i>AuthorID</i>	INT	Mã định danh duy nhất	PK, NOT NULL
<i>Name</i>	VARCHAR(255)	Tên đầy đủ của tác giả	NOT NULL
<i>Nationality</i>	VARCHAR(100)	Quốc tịch của tác giả	NULL
<i>YearOfBirth</i>	INT	Năm sinh của tác giả	NULL

2.3.3. PUBLISHER (NHÀ XUẤT BẢN)

Mô tả thực thể: Lưu trữ thông tin về các nhà xuất bản

Bảng 7: Thuộc tính thực thể PUBLISHER

Thuộc tính	Kiểu dữ liệu	Mô tả	Ràng buộc
<i>PublisherID</i>	INT	Mã định danh duy nhất	PK, NOT NULL
<i>Name</i>	VARCHAR(255)	Tên nhà xuất bản	NOT NULL
<i>Address</i>	VARCHAR(500)	Địa chỉ của nhà xuất bản	NULL
<i>ContactNumber</i>	VARCHAR(50)	Số điện thoại liên hệ	NULL

2.3.4. BOOK_COPY (BẢN SAO SÁCH)

Mô tả thực thể: Lưu trữ thông tin về từng bản sao vật lý của sách

Bảng 8: Thuộc tính thực thể BOOK_COPY

Thuộc tính	Kiểu dữ liệu	Mô tả	Ràng buộc
<i>CopyID</i>	INT	Mã định danh duy nhất của bản sao	PK, NOT NULL
<i>BookID</i>	INT	Mã sách mà bản sao thuộc về	FK, NOT NULL
<i>Condition</i>	VARCHAR(50)	Tình trạng bản sao	NULL
<i>Status</i>	VARCHAR(20)	Trạng thái hiện tại	NOT NULL

Ghi chú:

- *Condition*: New (Mới), Good (Tốt), Fair (Trung bình), Poor (Kém)
- *Status*: Available (Sẵn sàng), Borrowed (Đang mượn), Maintenance (Bảo dưỡng), Lost (Mất)

2.3.5. MEMBER (THÀNH VIÊN)

Mô tả thực thể: Lưu trữ thông tin về thành viên thư viện

Bảng 9: Thuộc tính thực thể MEMBER

Thuộc tính	Kiểu dữ liệu	Mô tả	Ràng buộc
<i>MemberID</i>	INT	Mã định danh duy nhất	PK, NOT NULL
<i>Name</i>	VARCHAR(255)	Tên đầy đủ của thành viên	NOT NULL
<i>Address</i>	VARCHAR(500)	Địa chỉ liên lạc	NULL
<i>Phone</i>	VARCHAR(20)	Số điện thoại liên hệ	NULL
<i>MembershipType</i>	VARCHAR(50)	Loại thành viên	NOT NULL

Ghi thích: *MembershipType*: Student (Sinh viên), Faculty (Giảng viên), Standard (Thông thường), Premium (VIP)

2.3.6. LOAN (MƯỢN/TRẢ)

Mô tả thực thể: Lưu trữ thông tin về các giao dịch mượn/trả sách

Bảng 10: Thuộc tính thực thể LOAN

Thuộc tính	Kiểu dữ liệu	Mô tả	Ràng buộc
<i>LoanID</i>	INT	Mã định danh duy nhất	PK, NOT NULL
<i>MemberID</i>	INT	Mã thành viên mượn	FK, NOT NULL
<i>CopyID</i>	INT	Mã bản sao được mượn	FK, NOT NULL
<i>BorrowDate</i>	DATE	Ngày mượn sách	NOT NULL
<i>DueDate</i>	DATE	Ngày hết hạn phải trả	NOT NULL
<i>ReturnDate</i>	DATE	Ngày thực tế trả sách	NULL
<i>OverdueFine</i>	DECIMAL(10,2)	Phí phạt quá hạn	DEFAULT 0

2.3.7. Tổng hợp thực thể

Bảng 11: Bảng tổng hợp 6 thực thể chính

#	Tên thực thể	Mô tả	Khóa chính
1	BOOK	Danh mục sách trong thư viện	<i>BookID</i>
2	AUTHOR	Danh mục tác giả	<i>AuthorID</i>
3	PUBLISHER	Danh mục nhà xuất bản	<i>PublisherID</i>
4	BOOK_COPY	Các bản sao vật lý của sách	<i>CopyID</i>
5	MEMBER	Thành viên thư viện	<i>MemberID</i>
6	LOAN	Giao dịch mượn/trả sách	<i>LoanID</i>

2.4. Các mối quan hệ

Sau khi xác định các thực thể, chúng tôi phân tích và xác định được **5 mối quan hệ chính**:

Bảng 12: Bảng tổng hợp 5 mối quan hệ

#	Tên	Thực thể 1	Thực thể 2	Bội số	Mô tả
1	Publishes	PUBLISHER	BOOK	1 : N	Một nhà XB xuất bản nhiều sách
2	Writes	AUTHOR	BOOK	M : N	Tác giả viết sách (qua BOOK_AUTHOR)
3	Has	BOOK	BOOK_COPY	1 : N	Một sách có nhiều bản sao vật lý
4	Processes	MEMBER	LOAN	1 : N	Thành viên tạo nhiều giao dịch mượn
5	Records	LOAN	BOOK_COPY	1 : 1	Một giao dịch ghi nhận một bản sao

Ghi thích về cấu trúc mỗi quan hệ:

- **Quan hệ mượn sách:** Thành viên không trực tiếp kết nối đến bản sao sách. Thay vào đó, một thành viên tạo một giao dịch LOAN, và giao dịch đó ghi nhận bản sao sách cụ thể được mượn.
- Cấu trúc này cho phép lưu trữ đầy đủ thông tin về mỗi lần mượn: ngày mượn, ngày hết hạn, ngày trả, và phí phạt.
- Điều này cũng cho phép theo dõi lịch sử mượn của một thành viên qua thời gian.

2.5. Thuộc tính chi tiết

2.5.1. Bảng trung gian BOOK_AUTHOR

Vì mỗi quan hệ giữa AUTHOR và BOOK là nhiều-nhiều (M:N), chúng tôi cần một bảng trung gian:

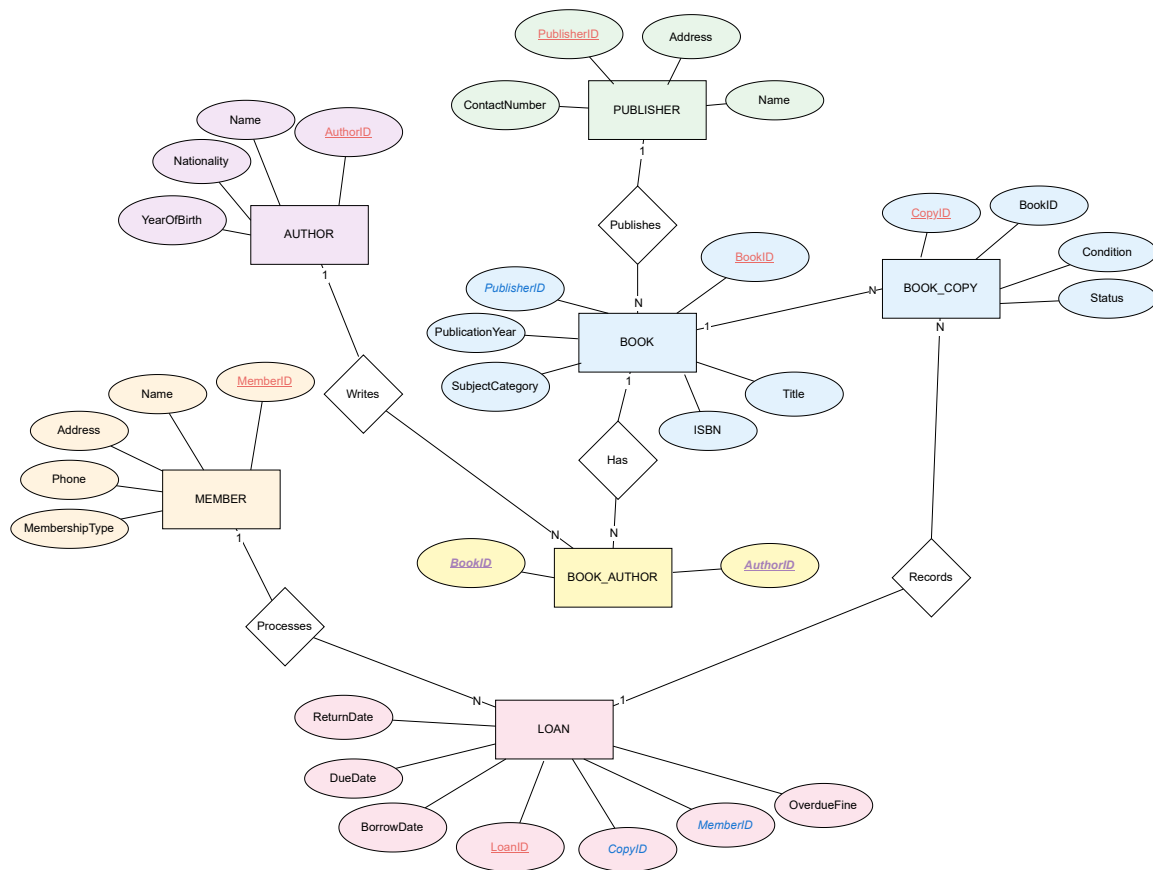
Bảng 13: Bảng trung gian BOOK_AUTHOR

Thuộc tính	Kiểu dữ liệu	Mô tả	Ràng buộc
<i>BookID</i>	INT	Khóa ngoại tham chiếu BOOK	PK, FK
<i>AuthorID</i>	INT	Khóa ngoại tham chiếu AUTHOR	PK, FK

Ràng buộc: PRIMARY KEY (*BookID*, *AuthorID*) - Khóa chính kép

2.6. Sơ đồ Thực thể - Mối quan hệ (ERD)

2.6.1. Sơ đồ ERD hoàn chỉnh



Hình 1: Sơ đồ Thực thể - Mối quan hệ (ERD) cho hệ thống Quản lý Thư viện

Chú thích ký hiệu trong sơ đồ:

- Khóa chính (Primary Key - PK): Được viết bằng chữ màu đỏ và gạch chân (Ví dụ: BookID, AuthorID, LoanID).
- Khóa ngoại (Foreign Key - FK): Được viết bằng chữ màu xanh dương và in nghiêng (Ví dụ: PublisherID trong bảng BOOK, MemberID trong bảng LOAN).
- Khóa chính vừa là Khóa ngoại (PK, FK): -Được viết bằng chữ màu tím, vừa gạch chân vừa in nghiêng.

2.6.2. Mô tả chi tiết các mối quan hệ

1. Publishes (1:N) - PUBLISHER đến BOOK

- Một nhà xuất bản có thể xuất bản nhiều cuốn sách

- Mỗi cuốn sách được xuất bản bởi đúng một nhà xuất bản
- Ràng buộc: *PublisherID* trong BOOK là khóa ngoại tham chiếu đến PUBLISHER

2. Writes (M:N) - AUTHOR đến BOOK

- Một tác giả có thể viết nhiều cuốn sách
- Một cuốn sách có thể có nhiều tác giả
- Cần bảng trung gian BOOK_AUTHOR để quản lý mối quan hệ này
- Bảng BOOK_AUTHOR có khóa chính kép (*BookID*, *AuthorID*)

3. Has (1:N) - BOOK đến BOOK_COPY

- Một cuốn sách (title) có thể có nhiều bản sao vật lý
- Mỗi bản sao thuộc về một cuốn sách duy nhất
- Ràng buộc: *BookID* trong BOOK_COPY là khóa ngoại tham chiếu đến BOOK
- Khi xóa sách, tất cả bản sao cũng bị xóa (ON DELETE CASCADE)

4. Processes (1:N) - MEMBER đến LOAN

- Một thành viên có thể tạo nhiều giao dịch mượn
- Mỗi giao dịch mượn thuộc về một thành viên
- Ràng buộc: *MemberID* trong LOAN là khóa ngoại tham chiếu đến MEMBER
- Lưu trữ lịch sử đầy đủ của các giao dịch mượn/trả

5. Records (1:1) - LOAN đến BOOK_COPY

- Một giao dịch mượn ghi nhận một bản sao sách cụ thể
- Mỗi bản sao tại một thời điểm chỉ được ghi nhận trong một giao dịch đang hoạt động
- Ràng buộc: *CopyID* trong LOAN là khóa ngoại tham chiếu đến BOOK_COPY
- Không thể có hai giao dịch mượn đang hoạt động cho cùng một bản sao

Lưu ý về thiết kế:

- Thành viên không trực tiếp kết nối đến bản sao sách
- Việc mượn sách được thực hiện thông qua thực thể LOAN
- Thiết kế này cho phép lưu trữ đầy đủ thông tin về mỗi lần mượn: ngày mượn, ngày hết hạn, ngày trả, và phí phạt
- Hỗ trợ truy vấn lịch sử mượn của thành viên và thống kê báo cáo

2.6.3. Các quyết định thiết kế chính

1. Tách BOOK và BOOK_COPY:

- BOOK lưu thông tin chung về title/ISBN
- BOOK_COPY lưu thông tin về từng bản sao vật lý
- Giúp quản lý chính xác số lượng sách có sẵn

2. Sử dụng bảng trung gian BOOK_AUTHOR:

- Hỗ trợ mối quan hệ M:N giữa AUTHOR và BOOK
- Cho phép một sách có nhiều tác giả
- Cho phép một tác giả có nhiều sách

3. Tách LOAN thành thực thể riêng:

- Lưu trữ đầy đủ thông tin về mỗi giao dịch mượn
- Hỗ trợ tính toán phí phạt
- Lưu trữ lịch sử mượn/trả đầy đủ

2.7. Kết luận báo cáo 1

2.7.1. Tổng kết kết quả

Báo cáo 1 đã hoàn thành phân tích và thiết kế sơ đồ ERD cho hệ thống quản lý thư viện:

1. Xác định được 6 thực thể chính:

- BOOK - 6 thuộc tính
- AUTHOR - 4 thuộc tính
- PUBLISHER - 4 thuộc tính
- BOOK_COPY - 4 thuộc tính
- MEMBER - 5 thuộc tính
- LOAN - 7 thuộc tính

2. Xác định được 5 mối quan hệ:

- Publishes (1:N) - PUBLISHER → BOOK
- Writes (M:N) - AUTHOR ↔ BOOK (qua BOOK_AUTHOR)
- Has (1:N) - BOOK → BOOK_COPY
- Processes (1:N) - MEMBER → LOAN
- Records (1:1) - LOAN → BOOK_COPY

Lưu ý về thiết kế quan hệ:

- Việc mượn sách được mô hình hóa thông qua thực thể LOAN, không phải quan hệ trực tiếp giữa MEMBER và BOOK_COPY
- Thành viên tạo giao dịch LOAN, và giao dịch đó ghi nhận bản sao sách được mượn
- Thiết kế này cho phép lưu trữ đầy đủ thông tin về mỗi lần mượn và theo dõi lịch sử

3. Xác định được 1 bảng trung gian:

- BOOK_AUTHOR - quản lý mối quan hệ nhiều-nhiều

3. BÁO CÁO 2: MÔ HÌNH QUAN HỆ (RELATIONAL DATA MODEL)

3.1. Giới thiệu báo cáo 2

Báo cáo 2 là bước tiếp theo trong quá trình thiết kế cơ sở dữ liệu, chuyển đổi từ sơ đồ Thực thể - Mối quan hệ (ERD) đã hoàn thiện trong Báo cáo 1 sang **Mô hình Quan hệ (Relational Data Model)**. Đây là bước quan trọng để chuẩn bị cho quá trình chuẩn hóa dữ liệu trong Báo cáo 3.

Mục tiêu của Báo cáo 2:

- Chuyển đổi các thực thể (entities) thành các bảng (tables/relation schemas)
- Xác định khóa chính (Primary Keys) và khóa ngoại (Foreign Keys)
- Thiết lập các mối quan hệ giữa các bảng thông qua khóa ngoại
- Xác định các phụ thuộc hàm (Functional Dependencies)
- Nhận dạng các mẫu chưa chuẩn hóa để chuẩn bị cho chuẩn hóa 3NF

3.2. Cơ sở lý thuyết

3.2.1. Mô hình Quan hệ (Relational Model)

Định nghĩa: Mô hình quan hệ là một mô hình cơ sở dữ liệu thể hiện dữ liệu dưới dạng các quan hệ (relations), mà người dùng có thể nhận thức dưới dạng các bảng (tables) gồm các hàng (rows) và cột (columns).

Các thành phần cơ bản:

Bảng 14: Thành phần của Mô hình Quan hệ

Thành phần	Mô tả
Relation (Quan hệ)	Một bảng 2D chứa dữ liệu, tương ứng với một thực thể trong ERD
Tuple (Bộ)	Một hàng trong bảng, tương ứng với một bản ghi hoặc một thể hiện của thực thể
Attribute (Thuộc tính)	Một cột trong bảng, tương ứng với một thuộc tính của thực thể
Domain (Miền giá trị)	Tập hợp các giá trị hợp lệ cho một thuộc tính
Degree (Bậc)	Số lượng thuộc tính trong một quan hệ
Cardinality (Độ)	Số lượng bộ trong một quan hệ

3.2.2. Khóa chính (Primary Key)

Định nghĩa: Khóa chính là một thuộc tính hoặc tập hợp các thuộc tính định danh duy nhất mỗi bộ trong một quan hệ.

Đặc điểm:

- **Tính duy nhất:** Không có hai bộ nào có cùng giá trị khóa chính
- **Tính không rỗng:** Khóa chính không thể chứa giá trị NULL
- **Tính ổn định:** Giá trị khóa chính không nên thay đổi

Ký hiệu: Gạch dưới thuộc tính (ví dụ: BookID) hoặc ký hiệu PK

3.2.3. Khóa ngoại (Foreign Key)

Định nghĩa: Khóa ngoại là một thuộc tính trong một quan hệ tham chiếu đến khóa chính của một quan hệ khác, thiết lập mối quan hệ giữa hai bảng.

Đặc điểm:

- **Tham chiếu:** Giá trị khóa ngoại phải khớp với giá trị khóa chính được tham chiếu, hoặc là NULL
- **Tính toàn vẹn:** Đảm bảo tính toàn vẹn tham chiếu (Referential Integrity)
- **Hành vi cascade:** Xác định điều gì xảy ra khi dữ liệu được tham chiếu bị xóa hoặc cập nhật

Ký hiệu: Chỉ mũi tên từ khóa ngoại sang khóa chính được tham chiếu

3.2.4. Phụ thuộc hàm (Functional Dependency)

Định nghĩa: Phụ thuộc hàm là một ràng buộc giữa hai tập hợp thuộc tính trong một quan hệ. Được ký hiệu là $X \rightarrow Y$, đọc là "X xác định Y" hoặc "Y phụ thuộc hàm vào X".

Định nghĩa toán học: Trong một quan hệ R, $X \rightarrow Y$ nếu và chỉ nếu mỗi giá trị của X liên kết với chính xác một giá trị của Y.

Ví dụ:

- $\text{BookID} \rightarrow \text{Title}$: Mã sách xác định tên sách
- $\text{BookID} \rightarrow \text{ISBN}, \text{PublicationYear}, \text{PublisherID}$
- $\text{MemberID} \rightarrow \text{Name}, \text{Address}, \text{Phone}$

3.3. Quy tắc chuyển đổi từ ERD sang Mô hình Quan hệ

3.3.1. Quy tắc cơ bản

Bảng 15: Quy tắc chuyển đổi ERD sang Relational Model

Thành phần ERD	Quy tắc chuyển đổi
Thực thể mạnh	Chuyển thành một bảng (table) riêng biệt
Thuộc tính	Chuyển thành cột (column) của bảng
Khóa chính	Chuyển thành PRIMARY KEY constraint
Quan hệ 1:1	Thêm khóa ngoại vào một trong hai bảng, thường là bảng có tham gia bắt buộc
Quan hệ 1:N	Thêm khóa ngoại vào bảng "N" (bên nhiều) tham chiếu đến bảng "1" (bên một)
Quan hệ M:N	Tạo bảng trung gian (junction table) với hai khóa ngoại
Thực thể yếu	Thêm khóa ngoại tham chiếu đến thực thể sở hữu, có thể tạo khóa chính kép

3.3.2. Quy tắc chi tiết cho từng loại quan hệ

1. Quan hệ Một-Nhiều (1:N):

- Bên "1": Thực thể có khóa chính PK
- Bên "N": Thêm thuộc tính khóa ngoại FK tham chiếu đến PK của bên "1"
- **Ví dụ:** PUBLISHER (1) → BOOK (N)
- BOOK có *PublisherID* là khóa ngoại tham chiếu đến PUBLISHER

2. Quan hệ Nhiều-Nhiều (M:N):

- Tạo bảng trung gian mới
- Bảng trung gian có hai khóa ngoại FK tham chiếu đến PK của hai thực thể
- Khóa chính của bảng trung gian là kết hợp của hai khóa ngoại
- **Ví dụ:** AUTHOR (M) ↔ BOOK (N)
- Tạo BOOK_AUTHOR với PK (*BookID*, *AuthorID*)

3. Quan hệ Một-Một (1:1):

- Thêm khóa ngoại vào một trong hai bảng
- Khóa ngoại vừa là khóa ngoại vừa là khóa chính
- **Ví dụ:** LOAN (1) ↔ BOOK_COPY (1)
- LOAN có *CopyID* là FK và tham chiếu độc nhất

3.4. Chuyển đổi ERD sang Lược đồ Quan hệ

3.4.1. Lược đồ quan hệ cho các thực thể cơ bản

1. PUBLISHER (Nhà xuất bản)

PUBLISHER(PublisherID, Name, Address, ContactNumber)

Bảng 16: Lược đồ quan hệ PUBLISHER

Tên bảng	Khóa chính	Thuộc tính	Kiểu dữ liệu
PUBLISHER	<u>PublisherID</u>	PublisherID	INT
		Name	VARCHAR(255)
		Address	VARCHAR(500)
		ContactNumber	VARCHAR(50)

2. AUTHOR (Tác giả)

AUTHOR(AuthorID, Name, Nationality, YearOfBirth)

Bảng 17: Lược đồ quan hệ AUTHOR

Tên bảng	Khóa chính	Thuộc tính	Kiểu dữ liệu
AUTHOR	<u>AuthorID</u>	AuthorID	INT
		Name	VARCHAR(255)
		Nationality	VARCHAR(100)
		YearOfBirth	INT

3. MEMBER (Thành viên)

MEMBER(MemberID, Name, Address, Phone, MembershipType)

Bảng 18: Lược đồ quan hệ MEMBER

Tên bảng	Khóa chính	Thuộc tính	Kiểu dữ liệu
MEMBER	<u>MemberID</u>	MemberID	INT
		Name	VARCHAR(255)
		Address	VARCHAR(500)
		Phone	VARCHAR(20)
		MembershipType	VARCHAR(50)

4. BOOK (Sách)

BOOK(BookID, ISBN, Title, SubjectCategory, PublicationYear, PublisherID)

Bảng 19: Lược đồ quan hệ BOOK

Tên bảng	Khóa chính	Khóa ngoại	Thuộc tính	Kiểu dữ liệu
BOOK	<u>BookID</u>	PublisherID	BookID	INT
			ISBN	VARCHAR(17)
			Title	VARCHAR(255)
			SubjectCategory	VARCHAR(100)
			PublicationYear	INT
			PublisherID	INT

Ghi chú:

- PublisherID là khóa ngoại tham chiếu đến PUBLISHER (PublisherID)
- Ký hiệu: BOOK.PublisherID → PUBLISHER.PublisherID

5. BOOK_COPY (Bản sao sách)

BOOK_COPY(CopyID, BookID, Condition, Status)

Bảng 20: Lược đồ quan hệ BOOK_COPY

Tên bảng	Khóa chính	Khóa ngoại	Thuộc tính	Kiểu dữ liệu
BOOK_COPY	<u>CopyID</u>	BookID	CopyID	INT
			BookID	INT
			Condition	VARCHAR(50)
			Status	VARCHAR(20)

Ghi chú:

- BookID là khóa ngoại tham chiếu đến BOOK (BookID)
- Đây là thực thể yếu (weak entity) phụ thuộc vào BOOK
- Ký hiệu: BOOK_COPY.BookID → BOOK.BookID

6. LOAN (Mượn/trả)

LOAN(LoanID, MemberID, CopyID, BorrowDate, DueDate, ReturnDate, OverdueFine)

Bảng 21: Lược đồ quan hệ LOAN

Tên bảng	Khóa chính	Khóa ngoại	Thuộc tính	Kiểu dữ liệu
LOAN	<u>LoanID</u>	MemberID, CopyID	LoanID	INT
			MemberID	INT
			CopyID	INT
			BorrowDate	DATE
			DueDate	DATE
			ReturnDate	DATE
			OverdueFine	DECIMAL(10,2)

Ghi chú:

- MemberID là khóa ngoại tham chiếu đến MEMBER (MemberID)
- CopyID là khóa ngoại tham chiếu đến BOOK_COPY (CopyID)
- Ký hiệu:
 - LOAN.MemberID → MEMBER.MemberID
 - LOAN.CopyID → BOOK_COPY.CopyID

3.4.2. Bảng trung gian cho quan hệ M:N

BOOK_AUTHOR (Bảng trung gian Sách-Tác giả)

BOOK_AUTHOR(BookID, AuthorID)

Bảng 22: Lược đồ quan hệ BOOK_AUTHOR (Bảng trung gian)

Tên bảng	Khóa chính	Khóa ngoại	Mô tả
BOOK_AUTHOR	<u>BookID</u> , <u>AuthorID</u>	BookID → BOOK (BookID)	Khóa ngoại đến BOOK
		AuthorID → AUTHOR (AuthorID)	Khóa ngoại đến AUTHOR

Ghi chú:

- **Khóa chính kép:** (BookID, AuthorID)
- Mỗi cặp (BookID, AuthorID) là duy nhất - một tác giả chỉ được liệt kê một lần cho mỗi sách
- Quan hệ này thể hiện mối quan hệ nhiều-nhiều giữa AUTHOR và BOOK

3.5. Tổng hợp Lược đồ Quan hệ Hệ Thống

3.5.1. Tất cả Lược đồ Quan hệ

Tóm tắt tất cả 7 bảng trong hệ thống:

PUBLISHER	(<u>PublisherID</u> , Name, Address, ContactNumber)
AUTHOR	(<u>AuthorID</u> , Name, Nationality, YearOfBirth)
MEMBER	(<u>MemberID</u> , Name, Address, Phone, MembershipType)
BOOK	(<u>BookID</u> , ISBN, Title, SubjectCategory, PublicationYear, PublisherID*)
BOOK_COPY	(<u>CopyID</u> , BookID [†] , Condition, Status)
LOAN	(<u>LoanID</u> , MemberID [‡] , CopyID [‡] , BorrowDate, DueDate, ReturnDate, OverdueFine)
BOOK_AUTHOR	(<u>BookID</u> [†] , <u>AuthorID</u> [‡])

Chú thích:

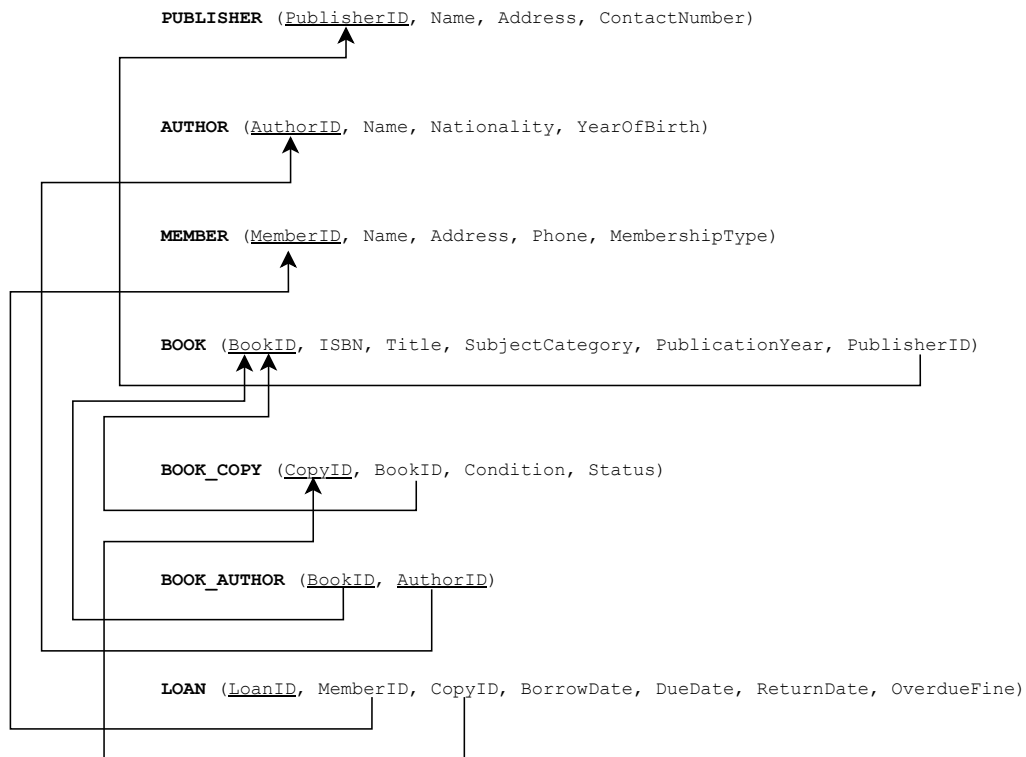
*, †, ‡ Các ký hiệu phía sau tên thuộc tính chỉ **Khóa ngoại** (Foreign Key) tham chiếu đến bảng tương ứng.

__ Các thuộc tính được gạch chân đóng vai trò là **Khóa chính** (Primary Key).

Bảng 23: Tổng hợp các bảng và khóa

STT	Tên bảng	Khóa chính (PK)	Khóa ngoại (FK)
1	PUBLISHER	PublisherID	—
2	AUTHOR	AuthorID	—
3	MEMBER	MemberID	—
4	BOOK	BookID	PublisherID → PUBLISHER
5	BOOK_COPY	CopyID	BookID → BOOK
6	LOAN	LoanID	MemberID → MEMBER
			CopyID → BOOK_COPY
7	BOOK_AUTHOR	BookID, AuthorID	BookID → BOOK
			AuthorID → AUTHOR

3.5.2. Sơ đồ Mô hình Quan hệ (Relational Data Model Diagram)



Hình 2: Sơ đồ Mô hình Quan hệ (Relational Data Model) cho hệ thống Quản lý Thư viện

Ghi chú ký hiệu trong sơ đồ:

- **PK (Primary Key):** Khóa chính - được hiển thị ở đầu danh sách cột, có gạch chân
- **FK (Foreign Key):** Khóa ngoại - tham chiếu đến bảng khác
- **Đường nối:** Thể hiện mối quan hệ giữa các bảng

3.6. Phụ thuộc Hàm (Functional Dependencies)

3.6.1. Định nghĩa và Ký hiệu

Phụ thuộc hàm (Functional Dependency - FD) mô tả mối quan hệ giữa các thuộc tính trong một bảng. Nếu thuộc tính X xác định thuộc tính Y, ta ký hiệu: $X \rightarrow Y$

Ví dụ: Nếu biết mã sách (BookID), ta biết được tên sách (Title), ISBN, năm xuất bản, và mã nhà xuất bản. Ký hiệu:

$BookID \rightarrow Title, ISBN, PublicationYear, PublisherID$

3.6.2. Các Phụ thuộc Hàm cho từng Bảng

1. Bảng PUBLISHER:

PublisherID → Name
PublisherID → Address
PublisherID → ContactNumber

Gộp: PublisherID → Name, Address, ContactNumber

Giải thích: Mã nhà xuất bản xác định duy nhất tên, địa chỉ và số điện thoại liên hệ của nhà xuất bản đó.

2. Bảng AUTHOR:

AuthorID → Name
AuthorID → Nationality
AuthorID → YearOfBirth

Gộp: AuthorID → Name, Nationality, YearOfBirth

Giải thích: Mã tác giả xác định duy nhất tên, quốc tịch và năm sinh của tác giả đó.

3. Bảng MEMBER:

MemberID → Name
MemberID → Address
MemberID → Phone
MemberID → MembershipType

Gộp: MemberID → Name, Address, Phone, MembershipType

Giải thích: Mã thành viên xác định duy nhất tên, địa chỉ, số điện thoại và loại thành viên.

4. Bảng BOOK:

BookID → ISBN
BookID → Title
BookID → SubjectCategory
BookID → PublicationYear
BookID → PublisherID

Gộp: BookID →

ISBN, Title, SubjectCategory, PublicationYear, PublisherID

Giải thích: Mã sách xác định duy nhất ISBN, tên, thể loại, năm xuất bản và mã nhà xuất bản của sách đó.

5. Bảng BOOK_COPY:

CopyID → BookID
CopyID → Condition
CopyID → Status

Gộp: CopyID → BookID, Condition, Status

Giải thích: Mã bản sao xác định duy nhất mã sách mà bản sao thuộc về, tình trạng và trạng thái hiện tại của bản sao.

6. Bảng LOAN:

LoanID → MemberID

LoanID → CopyID

LoanID → BorrowDate

LoanID → DueDate

LoanID → ReturnDate

LoanID → OverdueFine

Gộp: LoanID →

MemberID, CopyID, BorrowDate, DueDate, ReturnDate, OverdueFine

Giải thích: Mã giao dịch mượn xác định duy nhất thành viên mượn, bản sao được mượn, ngày mượn, ngày hết hạn, ngày trả và phí phạt.

7. Bảng BOOK_AUTHOR (Bảng trung gian):

Bảng này có khóa chính kép (BookID, AuthorID), nên không có phụ thuộc hàm từ một thuộc tính đơn lẻ. Các phụ thuộc hàm được xác định bởi cả hai khóa chính:

BookID, AuthorID → BookID

BookID, AuthorID → AuthorID

Giải thích: Cả hai thuộc tính đều là một phần của khóa chính, không có phụ thuộc hàm từ khóa này sang khóa kia.

3.6.3. Tổng hợp các Phụ thuộc Hàm

Bảng 24: Tổng hợp Phụ thuộc Hàm cho tất cả các bảng

Bảng	Phụ thuộc Hàm
PUBLISHER	PublisherID → Name, Address, ContactNumber
AUTHOR	AuthorID → Name, Nationality, YearOfBirth
MEMBER	MemberID → Name, Address, Phone, MembershipType
BOOK	BookID → ISBN, Title, SubjectCategory, PublicationYear, PublisherID
BOOK_COPY	CopyID → BookID, Condition, Status
LOAN	LoanID → MemberID, CopyID, BorrowDate, DueDate, ReturnDate, OverdueFine
BOOK_AUTHOR	(BookID, AuthorID) → BookID, AuthorID (khóa chính kép)

3.7. Đánh giá Mô hình Quan hệ

3.7.1. Tính toàn vẹn dữ liệu (Data Integrity)

1. Toàn vẹn thực thể (Entity Integrity):

- Mỗi bảng có khóa chính (Primary Key) xác định duy nhất mỗi dòng
- Khóa chính không chứa giá trị NULL
- Đảm bảo không có các bản ghi trùng lặp

2. Toàn vẹn tham chiếu (Referential Integrity):

- Mọi khóa ngoại đều tham chiếu đến khóa chính hợp lệ của bảng khác
- Không thể có giá trị khóa ngoại "mồ côi" (orphaned records)
- Được thực thi thông qua ràng buộc FOREIGN KEY

3. Toàn vẹn miền (Domain Integrity):

- Mỗi thuộc tính có kiểu dữ liệu phù hợp
- Giá trị thuộc tính nằm trong miền giá trị cho phép
- Các ràng buộc CHECK đảm bảo giá trị hợp lệ

3.7.2. Ưu điểm của Mô hình Quan hệ hiện tại

Bảng 25: Ưu điểm thiết kế

Thiết kế	Ưu điểm
Tách BOOK và BOOK_COPY	Quản lý chính xác số lượng bản sao vật lý của mỗi sách
Bảng trung gian BOOK_AUTHOR	Hỗ trợ mối quan hệ M:N giữa tác giả và sách
Thực thể LOAN riêng biệt	Lưu trữ đầy đủ thông tin về mỗi giao dịch mượn
Khóa ngoại với CASCADE	Tự động đồng bộ dữ liệu khi xóa/cập nhật

3.7.3. Hạn chế và Cải tiến tiềm năng

Các vấn đề cần chuẩn hóa trong Báo cáo 3:

1. Phân tích 1NF (First Normal Form):

- Kiểm tra xem tất cả thuộc tính có chứa giá trị nguyên thủy (atomic)
- Không có thuộc tính đa trị (multi-valued)
- Không có các nhóm lặp lại (repeating groups)

2. Phân tích 2NF (Second Normal Form):

- Kiểm tra các bảng có khóa chính kép (như BOOK_AUTHOR)

- Xác định phụ thuộc hàm một phần (partial dependencies)
- Loại bỏ các thuộc tính phụ thuộc vào một phần của khóa chính

3. Phân tích 3NF (Third Normal Form):

- Kiểm tra các phụ thuộc hàm truyền tải (transitive dependencies)
- Xác định các thuộc tính phi khóa phụ thuộc vào các thuộc tính phi khóa khác
- Tách các bảng để loại bỏ phụ thuộc truyền tải

3.8. Nhận dạng các Bất thường (Anomaly Recognition)

3.8.1. Mục đích của việc nhận dạng bất thường

Sau khi xác định các Phụ thuộc Hàm (Functional Dependencies) cho tất cả các bảng, bước tiếp theo là nhận dạng các bất thường tiềm ẩn trong cơ sở dữ liệu. Đây là preparation quan trọng cho quá trình chuẩn hóa (normalization) trong Báo cáo 3.

3.8.2. Phân tích sơ bộ các bất thường

1. Bảng với khóa chính đơn (Single Primary Key):

Các bảng PUBLISHER, AUTHOR, MEMBER, BOOK, BOOK_COPY, và LOAN đều có khóa chính đơn. Dựa trên các FD đã xác định:

- **Không có phụ thuộc hàm một phần (partial dependency):** Vì tất cả các bảng đều có khóa chính đơn, không có thuộc tính nào phụ thuộc vào một phần của khóa chính.
- **Không có phụ thuộc hàm truyền tải rõ ràng tại thời điểm này:** Tất cả thuộc tính phi khóa đều phụ thuộc trực tiếp vào khóa chính, không phụ thuộc vào các thuộc tính phi khóa khác.

2. Bảng trung gian BOOK_AUTHOR với khóa chính kép:

Bảng này có khóa chính kép (BookID, AuthorID):

- **Không có phụ thuộc hàm một phần:** Cả BookID và AuthorID đều cần thiết để định danh mỗi bản ghi.
- **Không có thuộc tính phi khóa:** Bảng chỉ chứa hai khóa ngoại, không có thuộc tính thêm nào có thể gây ra phụ thuộc truyền tải.

3.8.3. Kết luận sơ bộ

Dựa trên phân tích sơ bộ, cơ sở dữ liệu hiện tại:

- **Đã đáp ứng 1NF (First Normal Form):** Tất cả thuộc tính đều chứa giá trị nguyên thủy (atomic), không có thuộc tính đa trị hay nhóm lặp lại.
- **Đã đáp ứng 2NF (Second Normal Form):** Không có phụ thuộc hàm một phần do tất cả bảng đều có khóa chính đơn (ngoại trừ bảng trung gian đã được thiết kế đúng).

- **Cần phân tích kỹ hơn 3NF (Third Normal Form):** Trong Báo cáo 3, chúng tôi sẽ phân tích chi tiết xem có bất kỳ phụ thuộc hàm truyền tải nào tồn tại giữa các thuộc tính phi khóa.

Lưu ý: Phân tích này là bước sơ bộ dựa trên lược đồ quan hệ hiện tại. Báo cáo 3 sẽ thực hiện phân tích chuẩn hóa chi tiết với ví dụ dữ liệu cụ thể để xác minh các kết luận này.

3.9. Kết luận báo cáo 2

3.9.1. Tổng kết kết quả

Báo cáo 2 đã hoàn thành việc chuyển đổi từ Sơ đồ ERD sang Mô hình Quan hệ cho hệ thống quản lý thư viện:

1. Đã chuyển đổi thành công 6 thực thể ERD thành 6 bảng quan hệ:

- PUBLISHER - 4 thuộc tính
- AUTHOR - 4 thuộc tính
- MEMBER - 5 thuộc tính
- BOOK - 6 thuộc tính
- BOOK_COPY - 4 thuộc tính
- LOAN - 7 thuộc tính

2. Đã xác định tất cả Khóa chính (Primary Keys):

- PUBLISHER.PublisherID
- AUTHOR.AuthorID
- MEMBER.MemberID
- BOOK.BookID
- BOOK_COPY.CopyID
- LOAN.LoanID

3. Đã thiết lập tất cả Khóa ngoại (Foreign Keys):

- BOOK.PublisherID → PUBLISHER.PublisherID
- BOOK_COPY.BookID → BOOK.BookID
- LOAN.MemberID → MEMBER.MemberID
- LOAN.CopyID → BOOK_COPY.CopyID
- BOOK_AUTHOR.BookID → BOOK.BookID
- BOOK_AUTHOR.AuthorID → AUTHOR.AuthorID

4. Đã xác định các Phụ thuộc Hàm (Functional Dependencies):

- Tất cả 7 bảng đều đã được phân tích phụ thuộc hàm
- Mỗi bảng có khóa chính xác định duy nhất tất cả thuộc tính khác
- Không có phụ thuộc hàm một phần (vì không có bảng nào có khóa chính kép ngoài BOOK_AUTHOR)
- Không có phụ thuộc hàm truyền tải rõ ràng (sẽ phân tích kỹ hơn trong Báo cáo 3)

5. Đã tạo 1 bảng trung gian cho quan hệ M:N:

- BOOK_AUTHOR với khóa chính kép (BookID, AuthorID)

KẾT THÚC BÁO CÁO 2

Người thực hiện: Nhóm 3 - Lớp SE2043

Ngày hoàn thành: 30/01/2026

Giảng viên hướng dẫn: Vũ Thanh Phong

4. BÁO CÁO 3: TRỪ TƯỢNG HÓA (NORMALIZATION)

4.1. Phân công nhiệm vụ

Bảng 26: Phân công nhiệm vụ nhóm

STT	Họ và tên	Phân công nhiệm vụ	Điểm
1	Nguyễn Ngọc Phúc	Phân tích phụ thuộc hàm, kiểm tra 1NF/2NF/3NF cho các bảng PUBLISHER, AUTHOR, MEMBER	10
2	Thân Nhật Huy	Phân tích phụ thuộc hàm, kiểm tra 1NF/2NF/3NF cho các bảng BOOK, BOOK_COPY	10
3	Võ Hoàng Đình Trường	Phân tích phụ thuộc hàm, kiểm tra 1NF/2NF/3NF cho bảng LOAN, BOOK_AUTHOR	10
4	Nguyễn Thành An	Xác định các bất thường (anomalies) và đề xuất cải thiện	10
5	Nguyễn Quang Thiên Phú	Tổng hợp kết quả chuẩn hóa, viết nội dung báo cáo	10
6	Phạm Ngọc Hưng	Kiểm tra tính chính xác của phân tích, rà soát và chỉnh sửa báo cáo	10

4.2. Giới thiệu báo cáo 3

Báo cáo 3 là bước quan trọng trong quá trình thiết kế cơ sở dữ liệu, tập trung vào việc **Chuẩn hóa (Normalization)** lược đồ quan hệ đã thiết kế trong Báo cáo 2. Mục tiêu là tạo ra một cơ sở dữ liệu có cấu trúc tốt, tránh dư thừa dữ liệu và đảm bảo tính toàn vẹn.

Mục tiêu của Báo cáo 3:

- Phân tích chi tiết các Phụ thuộc Hàm (Functional Dependencies)
- Kiểm tra mức độ Chuẩn hóa First Normal Form (1NF)
- Kiểm tra mức độ Chuẩn hóa Second Normal Form (2NF)
- Kiểm tra mức độ Chuẩn hóa Third Normal Form (3NF)
- Xác định và giải quyết các bất thường (anomalies) trong cơ sở dữ liệu

4.3. Cơ sở lý thuyết

4.3.1. Khái niệm về Chuẩn hóa (Normalization)

Định nghĩa: Chuẩn hóa là quá trình tổ chức dữ liệu trong cơ sở dữ liệu nhằm giảm dư thừa và cải thiện tính toàn vẹn dữ liệu. Quá trình này chia các bảng lớn thành các bảng nhỏ hơn, có mối quan hệ với nhau.

Lợi ích của Chuẩn hóa:

- **Giảm dư thừa dữ liệu:** Loại bỏ việc lưu trữ cùng một thông tin ở nhiều nơi
- **Tránh các bất thường cập nhật:** Khi cập nhật dữ liệu, không cần cập nhật ở nhiều vị trí
- **Đảm bảo tính toàn vẹn:** Dữ liệu nhất quán và đáng tin cậy hơn
- **Cải thiện hiệu quả lưu trữ:** Giảm dung lượng ổ cứng cần thiết

4.3.2. Ba mức chuẩn hóa chính

Bảng 27: Tổng quan ba mức chuẩn hóa

Normal Form	Yêu cầu
1NF (First Normal Form)	<ul style="list-style-type: none">• Tất cả thuộc tính phải chứa giá trị nguyên thủy (atomic)• Không có thuộc tính đa trị (multi-valued)• Không có các nhóm lặp lại (repeating groups)
2NF (Second Normal Form)	<ul style="list-style-type: none">• Đã đáp ứng 1NF• Không có phụ thuộc hàm một phần (partial dependency)• Tất cả thuộc tính phi khóa phải phụ thuộc vào toàn bộ khóa chính
3NF (Third Normal Form)	<ul style="list-style-type: none">• Đã đáp ứng 2NF• Không có phụ thuộc hàm truyền tải (transitive dependency)• Thuộc tính phi khóa không phụ thuộc vào thuộc tính phi khóa khác

4.3.3. Các loại Phụ thuộc Hàm (Functional Dependencies)

1. Phụ thuộc hàm đầy đủ (Full Functional Dependency):

- Thuộc tính Y phụ thuộc vào toàn bộ tập hợp thuộc tính X
- Ký hiệu: $X \rightarrow Y$, trong đó Y không phụ thuộc vào bất kỳ tập con nào của X

2. Phụ thuộc hàm một phần (Partial Dependency):

- Thuộc tính Y phụ thuộc vào một phần của khóa chính X
- Chỉ xảy ra khi bảng có khóa chính kép (composite key)
- Vi phạm 2NF

3. Phụ thuộc hàm truyền tải (Transitive Dependency):

- Nếu $X \rightarrow Y$ và $Y \rightarrow Z$, thì $X \rightarrow Z$
- Thuộc tính Z phụ thuộc vào X thông qua Y
- Vi phạm 3NF

4.4. Phân tích Phụ thuộc Hàm chi tiết

4.4.1. Tổng hợp các Phụ thuộc Hàm từ Báo cáo 2

Trước khi thực hiện chuẩn hóa, chúng tôi tóm tắt lại tất cả các phụ thuộc hàm đã xác định trong Báo cáo 2:

Bảng 28: Tổng hợp Phụ thuộc Hàm cho tất cả các bảng

Bảng	Phụ thuộc Hàm
PUBLISHER	PublisherID \rightarrow Name, Address, ContactNumber
AUTHOR	AuthorID \rightarrow Name, Nationality, YearOfBirth
MEMBER	MemberID \rightarrow Name, Address, Phone, MembershipType
BOOK	BookID \rightarrow ISBN, Title, SubjectCategory, PublicationYear, PublisherID
BOOK_COPY	CopyID \rightarrow BookID, Condition, Status
LOAN	LoanID \rightarrow MemberID, CopyID, BorrowDate, DueDate, ReturnDate, OverdueFine
BOOK_AUTHOR	(BookID, AuthorID) \rightarrow BookID, AuthorID (khóa chính kép)

4.5. Kiểm tra First Normal Form (1NF)

4.5.1. Định nghĩa 1NF

Một bảng được gọi là ở First Normal Form (1NF) khi thỏa mãn các điều kiện sau:

1. Tất cả các thuộc tính đều chứa giá trị nguyên thủy (atomic values)
2. Không có thuộc tính đa trị (multi-valued attributes)
3. Không có các nhóm lặp lại (repeating groups)
4. Mỗi hàng trong bảng là duy nhất

4.5.2. Phân tích 1NF cho từng bảng

1. Bảng PUBLISHER:

Lược đồ: PUBLISHER(PublisherID, Name, Address, ContactNumber)

Bảng 29: Kiểm tra 1NF - PUBLISHER

Thuộc tính	Kiểu dữ liệu	Atomic?	Ghi chú
PublisherID	INT	[OK]	Giá trị đơn, duy nhất
Name	VARCHAR(255)	[OK]	Chuỗi ký tự đơn
Address	VARCHAR(500)	[OK]	Chuỗi ký tự đơn
ContactNumber	VARCHAR(50)	[OK]	Chuỗi ký tự đơn

Kết luận: Bảng PUBLISHER ĐÃ ĐÁP ỨNG 1NF. Tất cả thuộc tính đều chứa giá trị nguyên thủy.

2. Bảng AUTHOR:

Lược đồ: AUTHOR(AuthorID, Name, Nationality, YearOfBirth)

Bảng 30: Kiểm tra tính chuẩn hóa 1NF - Bảng AUTHOR

Thuộc tính	Kiểu dữ liệu	Atomic?	Ghi chú
AuthorID	INT	[OK]	Giá trị đơn, duy nhất
Name	VARCHAR(255)	[OK]	Chuỗi ký tự đơn
Nationality	VARCHAR(100)	[OK]	Chuỗi ký tự đơn
YearOfBirth	INT	[OK]	Giá trị số nguyên

Kết luận: Bảng AUTHOR ĐÃ ĐÁP ỨNG 1NF.

3. Bảng MEMBER:

Lược đồ: MEMBER(MemberID, Name, Address, Phone, MembershipType)

Bảng 31: Kiểm tra tính chuẩn hóa 1NF - Bảng MEMBER

Thuộc tính	Kiểu dữ liệu	Atomic?	Ghi chú
MemberID	INT	[OK]	Giá trị đơn, duy nhất
Name	VARCHAR(255)	[OK]	Chuỗi ký tự đơn
Address	VARCHAR(500)	[OK]	Chuỗi ký tự đơn
Phone	VARCHAR(20)	[OK]	Chuỗi ký tự đơn
MembershipType	VARCHAR(50)	[OK]	Chuỗi ký tự đơn

Kết luận: Bảng MEMBER ĐÃ ĐÁP ỨNG 1NF.

4. Bảng BOOK:

Lược đồ: BOOK(BookID, ISBN, Title, SubjectCategory, PublicationYear, PublisherID)

Bảng 32: Kiểm tra tính chuẩn hóa 1NF - Bảng BOOK

Thuộc tính	Kiểu dữ liệu	Atomic?	Ghi chú
BookID	INT	[OK]	Giá trị đơn, duy nhất
ISBN	VARCHAR(17)	[OK]	Chuỗi ký tự đơn
Title	VARCHAR(255)	[OK]	Chuỗi ký tự đơn
SubjectCategory	VARCHAR(100)	[OK]	Chuỗi ký tự đơn
PublicationYear	INT	[OK]	Giá trị số nguyên
PublisherID	INT	[OK]	Khóa ngoại, giá trị đơn

Kết luận: Bảng BOOK ĐÃ ĐÁP ỨNG 1NF.

5. Bảng BOOK_COPY:

Lược đồ: BOOK_COPY (CopyID, BookID, Condition, Status)

Bảng 33: Kiểm tra tính chuẩn hóa 1NF - Bảng BOOK_COPY

Thuộc tính	Kiểu dữ liệu	Atomic?	Ghi chú
CopyID	INT	[OK]	Giá trị đơn, duy nhất
BookID	INT	[OK]	Khóa ngoại, giá trị đơn
Condition	VARCHAR(50)	[OK]	Chuỗi ký tự đơn
Status	VARCHAR(20)	[OK]	Chuỗi ký tự đơn

Kết luận: Bảng BOOK_COPY ĐÃ ĐÁP ỨNG 1NF.

6. Bảng LOAN:

Lược đồ: LOAN (LoanID, MemberID, CopyID, BorrowDate, DueDate, ReturnDate, OverdueFine)

Bảng 34: Kiểm tra tính chuẩn hóa 1NF - Bảng LOAN

Thuộc tính	Kiểu dữ liệu	Atomic?	Ghi chú
LoanID	INT	[OK]	Giá trị đơn, duy nhất
MemberID	INT	[OK]	Khóa ngoại, giá trị đơn
CopyID	INT	[OK]	Khóa ngoại, giá trị đơn
BorrowDate	DATE	[OK]	Giá trị ngày đơn
DueDate	DATE	[OK]	Giá trị ngày đơn
ReturnDate	DATE	[OK]	Giá trị ngày đơn (có thể NULL)
OverdueFine	DECIMAL(10,2)	[OK]	Giá trị số đơn

Kết luận: Bảng LOAN ĐÃ ĐÁP ỨNG 1NF.

7. Bảng BOOK_AUTHOR (Bảng trung gian):

Lược đồ: BOOK_AUTHOR (BookID, AuthorID)

Bảng 35: Kiểm tra tính chuẩn hóa 1NF - Bảng BOOK_AUTHOR

Thuộc tính	Kiểu dữ liệu	Atomic?	Ghi chú
BookID	INT	[OK]	Khóa chính, Khóa ngoại, Giá trị đơn
AuthorID	INT	[OK]	Khóa chính, Khóa ngoại, Giá trị đơn

Kết luận: Bảng BOOK_AUTHOR **ĐÃ ĐÁP ỨNG 1NF**.

4.5.3. Tổng kết kiểm tra 1NF

Bảng 36: Tổng kết kiểm tra 1NF cho tất cả các bảng

STT	Tên bảng	Đã đáp ứng 1NF?
1	PUBLISHER	CÓ
2	AUTHOR	CÓ
3	MEMBER	CÓ
4	BOOK	CÓ
5	BOOK_COPY	CÓ
6	LOAN	CÓ
7	BOOK_AUTHOR	CÓ

Kết luận chung: Tất cả 7 bảng trong cơ sở dữ liệu đều **ĐÃ ĐÁP ỨNG 1NF**. Không có thuộc tính nào chứa giá trị không nguyên thủy, không có thuộc tính đa trị và không có nhóm lặp lại.

4.6. Kiểm tra Second Normal Form (2NF)

4.6.1. Định nghĩa 2NF

Một bảng được gọi là ở Second Normal Form (2NF) khi thỏa mãn các điều kiện sau:

1. Đã đáp ứng 1NF
2. Không có phụ thuộc hàm một phần (partial dependency)
3. Tất cả thuộc tính phi khóa phải phụ thuộc vào **toàn bộ** khóa chính, không phải chỉ một phần của khóa chính

Lưu ý quan trọng: Phụ thuộc hàm một phần chỉ có thể xảy ra khi bảng có **khóa chính kép (composite primary key)**. Nếu bảng chỉ có khóa chính đơn, thì bảng tự động đáp ứng 2NF (nếu đã ở 1NF).

4.6.2. Phân tích 2NF cho từng bảng

Nhóm 1: Các bảng có khóa chính ĐƠN (tự động đáp ứng 2NF nếu đã ở 1NF)

Các bảng này chỉ có khóa chính đơn, nên không thể có phụ thuộc hàm một phần:

Bảng 37: Kiểm tra 2NF - Bảng có khóa chính đơn

STT	Tên bảng	Khóa chính	Đã đáp ứng 2NF?
1	PUBLISHER	PublisherID	CÓ
2	AUTHOR	AuthorID	CÓ
3	MEMBER	MemberID	CÓ
4	BOOK	BookID	CÓ
5	BOOK_COPY	CopyID	CÓ
6	LOAN	LoanID	CÓ

Giải thích: Vì tất cả các bảng này đều có khóa chính đơn (single attribute primary key), không thể tồn tại phụ thuộc hàm một phần. Mọi thuộc tính phi khóa đều phụ thuộc vào toàn bộ khóa chính.

Nhóm 2: Bảng có khóa chính KÉP (cần kiểm tra chi tiết)

7. Bảng BOOK_AUTHOR:

Lược đồ: BOOK_AUTHOR (BookID, AuthorID) với khóa chính kép (BookID, AuthorID)

Phân tích:

Bảng này chỉ chứa hai thuộc tính, và cả hai đều là một phần của khóa chính:

- BookID: Khóa chính, khóa ngoại tham chiếu đến BOOK
- AuthorID: Khóa chính, khóa ngoại tham chiếu đến AUTHOR

Kiểm tra phụ thuộc hàm:

- Không có thuộc tính phi khóa nào trong bảng này
- Do đó, không thể có phụ thuộc hàm một phần
- Bảng được thiết kế đúng như một bảng trung gian (junction table) cho quan hệ M:N

Kết luận: Bảng BOOK_AUTHOR **ĐÃ ĐÁP ỨNG 2NF**.

4.6.3. Tổng kết kiểm tra 2NF

Bảng 38: Tổng kết kiểm tra 2NF cho tất cả các bảng

STT	Tên bảng	Loại PK	Partial Dependency?	Đã đáp ứng 2NF?
1	PUBLISHER	Đơn	Không có	CÓ
2	AUTHOR	Đơn	Không có	CÓ
3	MEMBER	Đơn	Không có	CÓ
4	BOOK	Đơn	Không có	CÓ
5	BOOK_COPY	Đơn	Không có	CÓ
6	LOAN	Đơn	Không có	CÓ
7	BOOK_AUTHOR	Kép	Không có	CÓ

Kết luận chung: Tất cả 7 bảng trong cơ sở dữ liệu đều **ĐÃ ĐÁP ỨNG 2NF**. Không có phụ thuộc hàm một phần nào tồn tại trong cơ sở dữ liệu.

4.7. Kiểm tra Third Normal Form (3NF)

4.7.1. Định nghĩa 3NF

Một bảng được gọi là ở Third Normal Form (3NF) khi thỏa mãn các điều kiện sau:

1. Đã đáp ứng 2NF
2. Không có phụ thuộc hàm truyền tải (transitive dependency)
3. Mọi thuộc tính phi khóa phải phụ thuộc trực tiếp vào khóa chính, không phụ thuộc vào thuộc tính phi khóa khác

Phụ thuộc hàm truyền tải (Transitive Dependency):

- Nếu $A \rightarrow B$ và $B \rightarrow C$, thì $A \rightarrow C$
- Đây là phụ thuộc truyền tải vì C phụ thuộc vào A thông qua B
- Vi phạm 3NF nếu C là thuộc tính phi khóa

4.7.2. Phân tích 3NF cho từng bảng

1. Bảng PUBLISHER:

Lược đồ: PUBLISHER(PublisherID, Name, Address, ContactNumber)

Các phụ thuộc hàm:

$$\text{PublisherID} \rightarrow \text{Name}, \text{Address}, \text{ContactNumber}$$

Phân tích:

- PublisherID là khóa chính
- Tất cả thuộc tính phi khóa (Name, Address, ContactNumber) phụ thuộc trực tiếp vào PublisherID
- Không có thuộc tính phi khóa nào phụ thuộc vào thuộc tính phi khóa khác
- Ví dụ: Address không phụ thuộc vào Name

Kết luận: Bảng PUBLISHER **ĐÃ ĐÁP ỨNG 3NF**.

2. Bảng AUTHOR:

Lược đồ: AUTHOR(AuthorID, Name, Nationality, YearOfBirth)

Các phụ thuộc hàm:

$$\text{AuthorID} \rightarrow \text{Name}, \text{Nationality}, \text{YearOfBirth}$$

Phân tích:

- AuthorID là khóa chính

- Tất cả thuộc tính phi khóa (Name, Nationality, YearOfBirth) phụ thuộc trực tiếp vào AuthorID
- YearOfBirth không phụ thuộc vào Nationality (một quốc tịch có nhiều năm sinh khác nhau)
- Không có phụ thuộc truyền tải

Kết luận: Bảng AUTHOR ĐÃ ĐÁP ỨNG 3NF.

3. Bảng MEMBER:

Lược đồ: MEMBER(MemberID, Name, Address, Phone, MembershipType)

Các phụ thuộc hàm:

MemberID \rightarrow Name, Address, Phone, MembershipType

Phân tích:

- MemberID là khóa chính
- Tất cả thuộc tính phi khóa (Name, Address, Phone, MembershipType) phụ thuộc trực tiếp vào MemberID
- MembershipType không phụ thuộc vào Address hoặc Phone
- Không có phụ thuộc truyền tải

Kết luận: Bảng MEMBER ĐÃ ĐÁP ỨNG 3NF.

4. Bảng BOOK:

Lược đồ: BOOK(BookID, ISBN, Title, SubjectCategory, PublicationYear, PublisherID)

Các phụ thuộc hàm:

BookID \rightarrow ISBN, Title, SubjectCategory, PublicationYear, PublisherID

Phân tích:

- BookID là khóa chính
- Tất cả thuộc tính phi khóa phụ thuộc trực tiếp vào BookID
- PublisherID là khóa ngoại, không phải là thuộc tính phi khóa
- SubjectCategory không phụ thuộc vào PublicationYear
- Không có phụ thuộc truyền tải

Kết luận: Bảng BOOK ĐÃ ĐÁP ỨNG 3NF.

5. Bảng BOOK_COPY:

Lược đồ: BOOK_COPY(CopyID, BookID, Condition, Status)

Các phụ thuộc hàm:

CopyID \rightarrow BookID, Condition, Status

Phân tích:

- CopyID là khóa chính
- BookID là khóa ngoại, tham chiếu đến BOOK
- Condition và Status phụ thuộc trực tiếp vào CopyID
- Condition không phụ thuộc vào BookID (cùng một sách có thể có bản sao ở nhiều tình trạng khác nhau)
- Không có phụ thuộc truyền tải

Kết luận: Bảng BOOK_COPY ĐÃ ĐÁP ỨNG 3NF.

6. Bảng LOAN:

Lược đồ: LOAN(LoanID, MemberID, CopyID, BorrowDate, DueDate, ReturnDate, OverdueFine)

Các phụ thuộc hàm:

LoanID → MemberID, CopyID, BorrowDate, DueDate, ReturnDate, OverdueFine

Phân tích chi tiết:

- LoanID là khóa chính
- MemberID và CopyID là khóa ngoại
- BorrowDate, DueDate, ReturnDate, OverdueFine phụ thuộc trực tiếp vào LoanID
- **Cần kiểm tra:** OverdueFine có phụ thuộc vào DueDate hoặc ReturnDate không?

Phân tích phụ thuộc tiềm ẩn:

Lưu ý: Trong thực tế, OverdueFine có thể được tính toán từ DueDate và ReturnDate. Tuy nhiên, trong thiết kế này:

- OverdueFine được lưu trữ như một thuộc tính (có thể do nghiệp vụ phức tạp)
- Phí phạt có thể được tính thủ công và nhập vào hệ thống
- Điều này không vi phạm 3NF vì OverdueFine không phụ thuộc hàm vào DueDate hoặc ReturnDate theo nghĩa toán học (cùng một DueDate có thể có OverdueFine khác nhau tùy vào quyết định của thủ thư)

Kết luận: Bảng LOAN ĐÃ ĐÁP ỨNG 3NF.

7. Bảng BOOK_AUTHOR:

Lược đồ: BOOK_AUTHOR(BookID, AuthorID)

Phân tích:

- Bảng này chỉ chứa hai khóa chính, không có thuộc tính phi khóa
- Không thể có phụ thuộc truyền tải khi không có thuộc tính phi khóa

Kết luận: Bảng BOOK_AUTHOR ĐÃ ĐÁP ỨNG 3NF.

4.7.3. Tổng kết kiểm tra 3NF

Bảng 39: Tổng kết kiểm tra 3NF cho tất cả các bảng

STT	Tên bảng	Transitive Dependency?	Đã đáp ứng 3NF?
1	PUBLISHER	Không có	CÓ
2	AUTHOR	Không có	CÓ
3	MEMBER	Không có	CÓ
4	BOOK	Không có	CÓ
5	BOOK_COPY	Không có	CÓ
6	LOAN	Không có	CÓ
7	BOOK_AUTHOR	Không có	CÓ

4.8. Tổng hợp Kết quả Chuẩn hóa

4.8.1. Tổng quan toàn bộ quá trình chuẩn hóa

Bảng 40: Tổng hợp kết quả chuẩn hóa 1NF - 2NF - 3NF

STT	Tên bảng	1NF	2NF	3NF
1	PUBLISHER	[OK]	[OK]	[OK]
2	AUTHOR	[OK]	[OK]	[OK]
3	MEMBER	[OK]	[OK]	[OK]
4	BOOK	[OK]	[OK]	[OK]
5	BOOK_COPY	[OK]	[OK]	[OK]
6	LOAN	[OK]	[OK]	[OK]
7	BOOK_AUTHOR	[OK]	[OK]	[OK]

4.8.2. Kết luận chung

Sau khi phân tích chi tiết qua ba mức chuẩn hóa (1NF, 2NF, 3NF), chúng tôi kết luận:

- **Tất cả 7 bảng trong cơ sở dữ liệu đều đã đáp ứng 3NF.**
- **Không cần phải thực hiện bất kỳ thay đổi nào để chuẩn hóa thêm.**
- **Lược đồ cơ sở dữ liệu hiện tại đã được thiết kế tốt từ đầu.**

Điều này có nghĩa là:

1. Không có dư thừa dữ liệu không cần thiết
2. Không có các bất thường khi cập nhật (update anomalies)
3. Không có các bất thường khi chèn (insert anomalies)
4. Không có các bất thường khi xóa (delete anomalies)
5. Cơ sở dữ liệu đảm bảo tính toàn vẹn dữ liệu

4.9. Xác định các Bất thường (Anomaly Analysis)

4.9.1. Các loại bất thường trong cơ sở dữ liệu

Mặc dù cơ sở dữ liệu đã đạt 3NF, chúng tôi vẫn phân tích các bất thường tiềm ẩn có thể xảy ra trong hệ thống quản lý thư viện:

1. Bất thường khi chèn (Insertion Anomaly):

Khả năng xảy ra: **THẤP**

Bảng 41: Phân tích Insertion Anomaly

Bảng	Phân tích
PUBLISHER	Không thể xảy ra - có thể thêm nhà XB mới mà không cần sách
AUTHOR	Không thể xảy ra - có thể thêm tác giả mới mà không cần sách
BOOK	Không thể xảy ra - có thể thêm sách mới
BOOK_COPY	Không thể xảy ra - có thể thêm bản sao mới
MEMBER	Không thể xảy ra - có thể thêm thành viên mới
LOAN	Không thể xảy ra - chỉ tạo giao dịch khi có mượn thực tế

2. Bất thường khi cập nhật (Update Anomaly):

Khả năng xảy ra: **THẤP**

Bảng 42: Phân tích Update Anomaly

Bảng	Phân tích
PUBLISHER	Không có thông tin NXB bị lặp lại ở nhiều bảng (chỉ FK trong BOOK)
AUTHOR	Không có thông tin tác giả bị lặp lại (qua BOOK_AUTHOR)
MEMBER	Không có thông tin thành viên bị lặp lại
BOOK	Mỗi sách chỉ xuất hiện một lần, cập nhật đơn giản
LOAN	Mỗi giao dịch độc lập, không ảnh hưởng lẫn nhau

3. Bất thường khi xóa (Deletion Anomaly):

Khả năng xảy ra: **THẤP**

Bảng 43: Phân tích Deletion Anomaly

Bảng	Phân tích
PUBLISHER	Xóa sách không làm mất thông tin NXB (bảng PUBLISHER độc lập)
AUTHOR	Xóa sách không làm mất thông tin tác giả (qua BOOK_AUTHOR)
BOOK_COPY	Xóa bản sao không làm mất thông tin sách (bảng BOOK độc lập)
LOAN	Xóa giao dịch không làm mất thông tin thành viên hay bản sao

4.9.2. Kết luận về các bất thường

Tất cả các bất thường đã được ngăn chặn nhờ thiết kế đạt 3NF:

- Không có mất dữ liệu khi xóa (do các thực thể độc lập)
- Không cần cập nhật nhiều nơi khi thay đổi thông tin
- Không bị giới hạn khi thêm dữ liệu mới

4.10. Các quyết định Thiết kế và Lý do

4.10.1. Quyết định 1: Tách BOOK và BOOK_COPY

Quyết định: Tách riêng thông tin sách (BOOK) và thông tin bản sao vật lý (BOOK_COPY)

Lý do:

- Một cuốn sách (title) có thể có nhiều bản sao vật lý
- Thông tin về sách (title, ISBN, tác giả) là chung cho tất cả bản sao
- Thông tin về bản sao (tình trạng, trạng thái) là riêng cho từng bản sao
- Giảm dư thừa dữ liệu (không phải lặp lại thông tin sách cho mỗi bản sao)

Lợi ích:

- Dễ quản lý số lượng sách có sẵn
- Dễ theo dõi tình trạng từng bản sao
- Dễ mở rộng (thêm/sửa bản sao không ảnh hưởng thông tin sách)

4.10.2. Quyết định 2: Sử dụng bảng trung gian BOOK_AUTHOR

Quyết định: Tạo bảng trung gian BOOK_AUTHOR thay vì thêm AuthorID vào BOOK

Lý do:

- Mọi quan hệ giữa BOOK và AUTHOR là nhiều-nhiều (M:N)
- Một cuốn sách có thể có nhiều tác giả
- Một tác giả có thể viết nhiều sách
- Không thể lưu trữ nhiều tác giả trong một cột của bảng BOOK (vi phạm 1NF)

Lợi ích:

- Đảm bảo 1NF (không có thuộc tính đa trị)
- Dễ quản lý danh sách tác giả cho mỗi sách
- Dễ truy vấn tất cả sách của một tác giả
- Hỗ trợ mở rộng trong tương lai (ví dụ: thêm vai trò tác giả)

4.10.3. Quyết định 3: Thực thể LOAN riêng biệt

Quyết định: Không tạo mối quan hệ trực tiếp giữa MEMBER và BOOK_COPY, thay vào đó dùng thực thể LOAN

Lý do:

- Mỗi giao dịch mượn cần lưu nhiều thông tin (ngày mượn, ngày hết hạn, ngày trả, phí phạt)
- Thành viên có thể mượn cùng một sách nhiều lần (khác thời điểm)
- Cần lưu lịch sử mượn đầy đủ

Lợi ích:

- Lưu trữ đầy đủ thông tin mỗi giao dịch
- Dễ truy vấn lịch sử mượn của thành viên
- Dễ tính toán phí phạt
- Hỗ trợ báo cáo thống kê

4.11. So sánh với thiết kế chưa chuẩn hóa

4.11.1. Ví dụ: Thiết kế SAI (vi phạm 3NF)

Giả sử chúng ta thiết kế sai bảng BOOK như sau:

`BOOK(BookID, ISBN, Title, PublisherID, PublisherName, PublisherAddress)`

Các phụ thuộc hàm trong thiết kế SAI:

`BookID → ISBN, Title, PublisherID`
`PublisherID → PublisherName, PublisherAddress`
`BookID → PublisherName, PublisherAddress (transitive dependency)`

Vấn đề (vi phạm 3NF):

- `PublisherName` và `PublisherAddress` phụ thuộc vào `PublisherID`, không phụ thuộc trực tiếp vào `BookID`
- Đây là phụ thuộc hàm truyền tải (transitive dependency)

Các bất thường xảy ra:

- **Insertion:** Không thể thêm nhà XB mới mà không có sách
- **Update:** Khi đổi địa chỉ NXB, phải cập nhật tất cả sách của NXB đó
- **Deletion:** Khi xóa cuốn sách cuối cùng của một NXB, mất thông tin NXB
- **Redundancy:** Thông tin NXB bị lặp lại cho mọi sách

4.11.2. Thiết kế ĐÚNG (đạt 3NF)

Tách thành hai bảng:

```
PUBLISHER(PublisherID, Name, Address)
BOOK(BookID, ISBN, Title, PublisherID)
```

Lợi ích:

- Thêm/sửa/xóa nhà XB độc lập với sách
- Không dư thừa thông tin NXB
- Tất cả các bất thường được giải quyết

4.12. Kết luận báo cáo 3

4.12.1. Tổng kết kết quả

Báo cáo 3 đã hoàn thành phân tích chi tiết quá trình chuẩn hóa cơ sở dữ liệu cho hệ thống quản lý thư viện:

1. Đã phân tích chi tiết các Phụ thuộc Hàm (Functional Dependencies):

- Xác định các phụ thuộc hàm cho tất cả 7 bảng
- Không có phụ thuộc hàm một phần (partial dependency)
- Không có phụ thuộc hàm truyền tải (transitive dependency)

2. Đã kiểm tra qua ba mức chuẩn hóa:

- **1NF (First Normal Form):** Tất cả 7 bảng đều ĐÃ ĐÁP ỨNG
- **2NF (Second Normal Form):** Tất cả 7 bảng đều ĐÃ ĐÁP ỨNG
- **3NF (Third Normal Form):** Tất cả 7 bảng đều ĐÃ ĐÁP ỨNG

3. Đã xác định các quyết định thiết kế chính:

- Tách BOOK và BOOK_COPY để quản lý bản sao vật lý
- Sử dụng bảng trung gian BOOK_AUTHOR cho quan hệ M:N
- Tạo thực thể LOAN riêng để lưu giao dịch mượn/trả

4. Đã phân tích các bất thường tiềm ẩn:

- Không có bất thường khi chèn (insertion anomaly)
- Không có bất thường khi cập nhật (update anomaly)
- Không có bất thường khi xóa (deletion anomaly)

4.12.2. Kết luận cuối cùng

Cơ sở dữ liệu hiện tại của hệ thống quản lý thư viện **ĐÃ ĐẠT 3NF** và không cần thay đổi nào thêm. Lược đồ đã được thiết kế tốt từ đầu, đảm bảo:

- Tính toàn vẹn dữ liệu
- Hiệu quả lưu trữ
- Tránh dư thừa
- Dễ mở rộng trong tương lai

KẾT THÚC BÁO CÁO 3

Người thực hiện: Nhóm 3 - Lớp SE2043

Ngày hoàn thành: 02/02/2026

Giảng viên hướng dẫn: Vũ Thanh Phong

5. BÁO CÁO 4: THIẾT KẾ THUẬT TOÁN (ALGORITHM DESIGN)

5.1. Phân công nhiệm vụ

Bảng 44: Phân công nhiệm vụ nhóm

STT	Họ và tên	Phân công nhiệm vụ	Điểm
1	Nguyễn Ngọc Phúc	Viết CREATE TABLE cho PUBLISHER, AUTHOR, MEMBER với đầy đủ ràng buộc	10
2	Thân Nhật Huy	Viết CREATE TABLE cho BOOK, BOOK_COPY với đầy đủ ràng buộc	10
3	Võ Hoàng Đình Trường	Viết CREATE TABLE cho LOAN, BOOK_AUTHOR với đầy đủ ràng buộc	10
4	Nguyễn Thành An	Viết Stored Procedures và Functions cho nghiệp vụ mượn/trả sách	10
5	Nguyễn Quang Thiên Phú	Viết Triggers để đảm bảo tính toàn vẹn dữ liệu phức tạp	10
6	Phạm Ngọc Hưng	Kiểm tra và tổng hợp toàn bộ mã T-SQL, tạo file .sql hoàn chỉnh	10

5.2. Giới thiệu báo cáo 4

Báo cáo 4 là bước quan trọng chuyển từ thiết kế lý thuyết sang thực tế triển khai trên SQL Server. Sau khi đã hoàn thiện lược đồ cơ sở dữ liệu đạt 3NF trong Báo cáo 3, chúng tôi thực hiện:

Mục tiêu của Báo cáo 4:

- Xây dựng các bảng đã chuẩn hóa với đầy đủ ràng buộc
- Triển khai Primary Keys (PK), Foreign Keys (FK)
- Thêm các ràng buộc bổ sung: CHECK, DEFAULT, UNIQUE
- Tạo Stored Procedures, Functions, và Triggers
- Đảm bảo tính toàn vẹn dữ liệu ở mức cơ sở dữ liệu

5.3. Cơ sở lý thuyết

5.3.1. Các loại Ràng buộc (Constraints) trong SQL Server

Bảng 45: Các loại ràng buộc trong SQL Server

Loại ràng buộc	Mô tả
PRIMARY KEY	Định danh duy nhất mỗi bản ghi, không cho phép NULL
FOREIGN KEY	Đảm bảo tính toàn vẹn tham chiếu giữa các bảng
UNIQUE	Đảm bảo giá trị trong cột là duy nhất (cho phép 1 NULL)
CHECK	Xác thực dữ liệu theo điều kiện tùy chỉnh
DEFAULT	Tự động gán giá trị mặc định khi không có giá trị được chỉ định
NOT NULL	Không cho phép giá trị NULL trong cột

5.3.2. Stored Procedures và Functions

Stored Procedure:

- Là tập hợp các câu lệnh T-SQL được lưu sẵn trong cơ sở dữ liệu
- Có thể nhận tham số đầu vào và trả về kết quả
- Lợi ích: Bảo mật (ẩn logic), Hiệu suất (được biên dịch sẵn), Tái sử dụng

Function:

- **Scalar Function:** Trả về một giá trị đơn
- **Table-Valued Function:** Trả về một bảng dữ liệu
- Có thể sử dụng trong câu lệnh SELECT, WHERE

Trigger:

- Là stored procedure tự động thực thi khi có sự kiện INSERT, UPDATE, DELETE
- Dùng để thực hiện các ràng buộc nghiệp vụ phức tạp
- Ba loại: AFTER, INSTEAD OF

5.4. Triển khai cơ sở dữ liệu trên SQL Server

5.4.1. Tạo cơ sở dữ liệu

```
-- ẠTạo cơ sở dữ liệu
CREATE DATABASE LibraryDB;
GO

-- Sử dụng cơ sở dữ liệu
USE LibraryDB;
GO
```

Listing 1: Tạo cơ sở dữ liệu LibraryDB

5.4.2. Xóa bảng nếu tồn tại (để tránh lỗi)

```
-- Xóa bảng theo thứ tự (con trước, cha sau)
DROP TABLE IF EXISTS LOAN;
DROP TABLE IF EXISTS BOOK_AUTHOR;
DROP TABLE IF EXISTS BOOK_COPY;
DROP TABLE IF EXISTS BOOK;
DROP TABLE IF EXISTS MEMBER;
DROP TABLE IF EXISTS AUTHOR;
DROP TABLE IF EXISTS PUBLISHER;
GO
```

Listing 2: Xóa bảng theo thứ tự đúng

5.5. Tạo bảng với đầy đủ ràng buộc

5.5.1. Bảng PUBLISHER (Nhà xuất bản)

```
CREATE TABLE PUBLISHER (
    PublisherID      INT          NOT NULL      IDENTITY(1,1),
    Name             VARCHAR(255) NOT NULL,
    Address          VARCHAR(500) NULL,
    ContactNumber    VARCHAR(50)  NULL,

    -- Primary Key
    CONSTRAINT PK_PUBLISHER PRIMARY KEY (PublisherID),

    -- Check Constraints
    CONSTRAINT CK_PUBLISHER_Name
        CHECK (LEN(Name) > 0),
    CONSTRAINT CK_PUBLISHER_ContactNumber
        CHECK (ContactNumber IS NULL OR ContactNumber LIKE '[0-9]%')
);
GO
```

Listing 3: Tạo bảng PUBLISHER

Giải thích ràng buộc:

- IDENTITY(1,1): Tự động tăng từ 1
- CK_PUBLISHER_Name: Tên nhà XB không được rỗng
- CK_PUBLISHER_ContactNumber: Số điện thoại phải bắt đầu bằng chữ số

5.5.2. Bảng AUTHOR (Tác giả)

```
CREATE TABLE AUTHOR (
    AuthorID        INT          NOT NULL      IDENTITY(1,1),
    Name            VARCHAR(255) NOT NULL,
    Nationality     VARCHAR(100) NULL,
```

```

YearOfBirth      INT                NULL,

-- Primary Key
CONSTRAINT PK_AUTHOR PRIMARY KEY (AuthorID),

-- Check Constraints
CONSTRAINT CK_AUTHOR_Name
    CHECK (LEN(Name) > 0),
CONSTRAINT CK_AUTHOR_YearOfBirth
    CHECK (YearOfBirth IS NULL
        OR (YearOfBirth BETWEEN 1000 AND YEAR(GETDATE()) AND
            YearOfBirth <= YEAR(GETDATE()) - 10))
);
GO

```

Listing 4: Tạo bảng AUTHOR

Giải thích ràng buộc:

- CK_AUTHOR_YearOfBirth: Năm sinh phải hợp lý (từ năm 1000 đến năm hiện tại - 10)

5.5.3. Bảng MEMBER (Thành viên)

```

CREATE TABLE MEMBER (
    MemberID      INT                NOT NULL    IDENTITY(1,1),
    Name          VARCHAR(255)       NOT NULL,
    Address       VARCHAR(500)       NULL,
    Phone         VARCHAR(20)        NULL,
    MembershipType VARCHAR(50)       NOT NULL,
    JoinDate      DATE               NOT NULL    DEFAULT (GETDATE()),

-- Primary Key
CONSTRAINT PK_MEMBER PRIMARY KEY (MemberID),

-- Check Constraints
CONSTRAINT CK_MEMBER_Name
    CHECK (LEN(Name) > 0),
CONSTRAINT CK_MEMBER_MembershipType
    CHECK (MembershipType IN ('Student', 'Faculty', 'Standard', 'Premium')),
CONSTRAINT CK_MEMBER_Phone
    CHECK (Phone IS NULL OR Phone LIKE '[0-9]%')
);
GO

```

Listing 5: Tạo bảng MEMBER

Giải thích ràng buộc:

- CK_MEMBER_MembershipType: Loại thành viên phải là một trong các giá trị cho phép
- JoinDate DEFAULT (GETDATE()): Tự động gán ngày hiện tại

5.5.4. Bảng BOOK (Sách)

```
CREATE TABLE BOOK (
    BookID          INT          NOT NULL    IDENTITY(1,1),
    ISBN            VARCHAR(17)  NULL,
    Title           VARCHAR(255) NOT NULL,
    SubjectCategory VARCHAR(100) NULL,
    PublicationYear  INT          NULL,
    PublisherID     INT          NOT NULL,

    -- Primary Key
    CONSTRAINT PK_BOOK PRIMARY KEY (BookID),

    -- Foreign Key
    CONSTRAINT FK_BOOK_PUBLISHER FOREIGN KEY (PublisherID)
        REFERENCES PUBLISHER(PublisherID)
        ON UPDATE CASCADE
        ON DELETE NO ACTION,

    -- Unique Constraint
    CONSTRAINT UQ_BOOK_ISBN UNIQUE (ISBN),

    -- Check Constraints
    CONSTRAINT CK_BOOK_Title
        CHECK (LEN(Title) > 0),
    CONSTRAINT CK_BOOK_PublicationYear
        CHECK (PublicationYear IS NULL
            OR PublicationYear BETWEEN 1000 AND YEAR(GETDATE()) +
                5),
    CONSTRAINT CK_BOOK_ISBN
        CHECK (ISBN IS NULL
            OR ISBN LIKE '[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
            OR ISBN LIKE '[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
);
GO
```

Listing 6: Tạo bảng BOOK

Giải thích ràng buộc:

- FK_BOOK_PUBLISHER: Khóa ngoại tham chiếu PUBLISHER, cho phép cập nhật cascade, không cho xóa nếu có sách
- UQ_BOOK_ISBN: Mã ISBN phải duy nhất
- CK_BOOK_ISBN: Định dạng ISBN (10 hoặc 13 số)

5.5.5. Bảng BOOK_COPY (Bản sao sách)

```

CREATE TABLE BOOK_COPY (
    CopyID          INT          NOT NULL    IDENTITY(1,1),
    BookID          INT          NOT NULL,
    Condition       VARCHAR(50)  NOT NULL,
    Status          VARCHAR(20)  NOT NULL    DEFAULT ('Available'),

    -- Primary Key
    CONSTRAINT PK_BOOK_COPY PRIMARY KEY (CopyID),

    -- Foreign Key
    CONSTRAINT FK_BOOK_COPY_BOOK FOREIGN KEY (BookID)
        REFERENCES BOOK(BookID)
        ON UPDATE CASCADE
        ON DELETE CASCADE,

    -- Check Constraints
    CONSTRAINT CK_BOOK_COPY_Condition
        CHECK (Condition IN ('New', 'Good', 'Fair', 'Poor')),
    CONSTRAINT CK_BOOK_COPY_Status
        CHECK (Status IN ('Available', 'Borrowed', 'Maintenance', '
                        Lost'))
);
GO

```

Listing 7: Tạo bảng BOOK_COPY

Giải thích ràng buộc:

- FK_BOOK_COPY_BOOK: ON DELETE CASCADE - xóa sách thì xóa luôn bản sao
- CK_BOOK_COPY_Condition: Tình trạng chỉ là các giá trị cho phép
- CK_BOOK_COPY_Status: Trạng thái mặc định là 'Available'

5.5.6. Bảng LOAN (Mượn/trả)

```

CREATE TABLE LOAN (
    LoanID          INT          NOT NULL    IDENTITY(1,1),
    MemberID        INT          NOT NULL,
    CopyID          INT          NOT NULL,
    BorrowDate      DATE         NOT NULL    DEFAULT (GETDATE()),
    DueDate         DATE         NOT NULL,
    ReturnDate      DATE         NULL,
    OverdueFine     DECIMAL(10,2) NOT NULL    DEFAULT (0),

    -- Primary Key
    CONSTRAINT PK_LOAN PRIMARY KEY (LoanID),

    -- Foreign Keys
    CONSTRAINT FK_LOAN_MEMBER FOREIGN KEY (MemberID)
        REFERENCES MEMBER(MemberID)
        ON UPDATE CASCADE

```

```

        ON DELETE NO ACTION,
CONSTRAINT FK_LOAN_BOOK_COPY FOREIGN KEY (CopyID)
    REFERENCES BOOK_COPY(CopyID)
    ON UPDATE CASCADE
    ON DELETE NO ACTION,

-- Unique Constraint
CONSTRAINT UQ_LOAN_CopyID_Active
    UNIQUE (CopyID)
    WHERE (ReturnDate IS NULL),

-- Check Constraints
CONSTRAINT CK_LOAN_DueDate
    CHECK (DueDate >= BorrowDate),
CONSTRAINT CK_LOAN_ReturnDate
    CHECK (ReturnDate IS NULL OR ReturnDate >= BorrowDate),
CONSTRAINT CK_LOAN_OverdueFine
    CHECK (OverdueFine >= 0)
);
GO

```

Listing 8: Tạo bảng LOAN

Giải thích ràng buộc:

- UQ_LOAN_CopyID_Active WHERE (ReturnDate IS NULL): Mỗi bản sao chỉ được mượn bởi một người tại một thời điểm
- CK_LOAN_DueDate: Ngày hết hạn phải sau ngày mượn
- CK_LOAN_ReturnDate: Ngày trả phải sau hoặc bằng ngày mượn

5.5.7. Bảng trung gian BOOK_AUTHOR

```

CREATE TABLE BOOK_AUTHOR (
    BookID          INT     NOT NULL,
    AuthorID        INT     NOT NULL,

-- Primary Key (Composite)
CONSTRAINT PK_BOOK_AUTHOR PRIMARY KEY (BookID, AuthorID),

-- Foreign Keys
CONSTRAINT FK_BOOK_AUTHOR_BOOK FOREIGN KEY (BookID)
    REFERENCES BOOK(BookID)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
CONSTRAINT FK_BOOK_AUTHOR_AUTHOR FOREIGN KEY (AuthorID)
    REFERENCES AUTHOR(AuthorID)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);
GO

```

Listing 9: Tạo bảng BOOK_AUTHOR

Giải thích:

- Khóa chính kép (BookID, AuthorID)
- ON DELETE CASCADE trên cả hai khóa ngoại

5.6. Tóm tắt các bảng và ràng buộc

Bảng 46: Tóm tắt các bảng và ràng buộc đã tạo

STT	Tên bảng	PK	FK	CHECK	UNIQUE
1	PUBLISHER	[OK]	–	2	–
2	AUTHOR	[OK]	–	2	–
3	MEMBER	[OK]	–	3	–
4	BOOK	[OK]	1	3	1
5	BOOK_COPY	[OK]	1	2	–
6	LOAN	[OK]	2	3	1*
7	BOOK_AUTHOR	[OK] [†]	2	–	–

* *UNIQUE Filtered: Chỉ áp dụng khi ReturnDate IS NULL*

[†] *Khóa chính kép (BookID, AuthorID)*

5.7. Stored Procedures

5.7.1. Procedure 1: Mượn sách

```
CREATE PROCEDURE dbo.usp_BorrowBook
    @MemberID      INT,
    @CopyID         INT,
    @DueDays        INT = 14
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @DueDate DATE;

    BEGIN TRY
        BEGIN TRANSACTION;

        -- 1. Kiểm tra thành viên tồn tại
        IF NOT EXISTS (SELECT 1 FROM MEMBER WHERE MemberID =
            @MemberID)
        BEGIN
            RAISERROR(N'Ồ! Thành viên không tồn tại!', 16, 1);
            RETURN;
        END
    END TRY
```

```

-- 2. Kiểm tra bản sao sách tồn tại và AVAILABLE
IF NOT EXISTS (
    SELECT 1 FROM BOOK_COPY
    WHERE CopyID = @CopyID AND Status = 'Available'
)
BEGIN
    RAISERROR(N'Ồi: Sách không có sẵn để mượn!', 16, 1);
    RETURN;
END

-- 3. Tính ngày hết hạn
SET @DueDate = DATEADD(DAY, @DueDays, GETDATE());

-- 4. Vào giao dịch mượn
INSERT INTO LOAN (MemberID, CopyID, BorrowDate, DueDate)
VALUES (@MemberID, @CopyID, GETDATE(), @DueDate);

-- 5. Cập nhật trạng thái bản sao
UPDATE BOOK_COPY
SET Status = 'Borrowed'
WHERE CopyID = @CopyID;

COMMIT TRANSACTION;

SELECT N'Mượn sách thành công!' AS Message,
       @DueDate AS DueDate;
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    THROW;
END CATCH
END;
GO

```

Listing 10: Stored Procedure mượn sách

5.7.2. Procedure 2: Trả sách

```

CREATE PROCEDURE dbo.usp_ReturnBook
    @CopyID INT,
    @ReturnDate DATE = NULL
AS
BEGIN
    SET NOCOUNT ON;

    IF @ReturnDate IS NULL
        SET @ReturnDate = CAST(GETDATE() AS DATE);

    BEGIN TRY
        BEGIN TRANSACTION;
    
```

```

DECLARE @LoanID INT, @DueDate DATE, @DaysOverdue INT, @Fine
        DECIMAL(10,2);

-- 1. lấy thông tin giao dịch mượn
SELECT @LoanID = LoanID, @DueDate = DueDate
FROM LOAN
WHERE CopyID = @CopyID AND ReturnDate IS NULL;

IF @LoanID IS NULL
BEGIN
    RAISERROR(N'Ồi: Không tìm thấy giao dịch mượn đang hoạt
        động!', 16, 1);
    RETURN;
END

-- 2. Tính phí phạt (nếu quá hạn)
SET @DaysOverdue = DATEDIFF(DAY, @DueDate, @ReturnDate);

IF @DaysOverdue > 0
    SET @Fine = @DaysOverdue * 5000; -- 5000 ĐVN/ngày
ELSE
    SET @Fine = 0;

-- 3. cập nhật giao dịch mượn
UPDATE LOAN
SET ReturnDate = @ReturnDate,
    OverdueFine = @Fine
WHERE LoanID = @LoanID;

-- 4. cập nhật trạng thái bản sao
UPDATE BOOK_COPY
SET Status = 'Available'
WHERE CopyID = @CopyID;

COMMIT TRANSACTION;

SELECT N'Ấp sách thành công!' AS Message,
        @DaysOverdue AS DaysOverdue,
        @Fine AS OverdueFine;
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    THROW;
END CATCH
END;
GO

```

Listing 11: Stored Procedure trả sách

5.8. Functions

5.8.1. Function 1: Lấy danh sách sách của tác giả

```
CREATE FUNCTION dbo.ufn_GetBooksByAuthor(@AuthorName VARCHAR(255))
RETURNS TABLE
AS
RETURN
(
    SELECT b.BookID, b.Title, b.ISBN, b.PublicationYear, p.Name AS
        Publisher
    FROM BOOK b
    INNER JOIN BOOK_AUTHOR ba ON b.BookID = ba.BookID
    INNER JOIN AUTHOR a ON ba.AuthorID = a.AuthorID
    INNER JOIN PUBLISHER p ON b.PublisherID = p.PublisherID
    WHERE a.Name = @AuthorName
);
GO
```

Listing 12: Table-Valued Function

5.8.2. Function 2: Tính phí phạt quá hạn

```
CREATE FUNCTION dbo.ufn_CalculateOverdueFine(
    @DueDate DATE,
    @ReturnDate DATE,
    @DailyRate DECIMAL(10,2) = 5000
)
RETURNS DECIMAL(10,2)
AS
BEGIN
    IF @ReturnDate IS NULL OR @DueDate IS NULL
        RETURN 0;

    DECLARE @DaysOverdue INT;
    SET @DaysOverdue = DATEDIFF(DAY, @DueDate, @ReturnDate);

    IF @DaysOverdue > 0
        RETURN @DaysOverdue * @DailyRate;
    ELSE
        RETURN 0;
END;
GO
```

Listing 13: Scalar Function

5.9. Triggers

5.9.1. Trigger 1: Ngăn mượn sách nếu thành viên quá hạn

```

CREATE TRIGGER trg_PreventBorrow_OverdueMember
ON LOAN
INSTEAD OF INSERT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @MemberID INT;
    SELECT @MemberID = MemberID FROM inserted;

    -- Kiểm tra thành viên có sách quá hạn chưa trả
    IF EXISTS (
        SELECT 1 FROM LOAN
        WHERE MemberID = @MemberID
            AND ReturnDate IS NULL
            AND DueDate < CAST(GETDATE() AS DATE)
    )
    BEGIN
        RAISERROR(
            N'Ồi: Thành viên đang có sách quá hạn chưa trả. Không
            thể mượn thêm!',
            16, 1
        );
        RETURN;
    END

    -- Nếu không vi phạm, thực hiện INSERT
    INSERT INTO LOAN (MemberID, CopyID, BorrowDate, DueDate,
        ReturnDate, OverdueFine)
    SELECT MemberID, CopyID, BorrowDate, DueDate, ReturnDate,
        OverdueFine
    FROM inserted;
END;
GO

```

Listing 14: Trigger kiểm tra thành viên bị quá hạn

5.9.2. Trigger 2: Tự động tính phí phạt khi trả sách

```

CREATE TRIGGER trg_CalculateOverdueFine
ON LOAN
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    -- Chỉ xử lý khi ReturnDate được cập nhật từ NULL sang có giá trị
    IF UPDATE(ReturnDate)
    BEGIN
        UPDATE LOAN

```

```

        SET OverdueFine = dbo.ufn_CalculateOverdueFine(
            i.DueDate,
            i.ReturnDate,
            5000
        )
    FROM inserted i
    INNER JOIN LOAN l ON l.LoanID = i.LoanID
    WHERE i.ReturnDate IS NOT NULL;
END
END;
GO

```

Listing 15: Trigger tự động tính phí phạt

5.10. Các câu truy vấn mẫu

5.10.1. Query 1: Liệt kê sách đang được mượn

```

SELECT
    l.LoanID,
    m.Name AS MemberName,
    m.Phone AS MemberPhone,
    b.Title AS BookTitle,
    bc.CopyID,
    l.BorrowDate,
    l.DueDate,
    CASE
        WHEN l.ReturnDate IS NULL AND GETDATE() > l.DueDate
            THEN N'Quá ặhn'
        WHEN l.ReturnDate IS NULL
            THEN N'Đang ượmn'
        ELSE N'Đã ảtr'
    END AS Status
FROM LOAN l
INNER JOIN MEMBER m ON l.MemberID = m.MemberID
INNER JOIN BOOK_COPY bc ON l.CopyID = bc.CopyID
INNER JOIN BOOK b ON bc.BookID = b.BookID
WHERE l.ReturnDate IS NULL
ORDER BY l.DueDate;

```

5.10.2. Query 2: Thống kê sách được mượn nhiều nhất

```

SELECT TOP 10
    b.BookID,
    b.Title,
    b.ISBN,
    COUNT(l.LoanID) AS TimesBorrowed
FROM BOOK b
INNER JOIN BOOK_COPY bc ON b.BookID = bc.BookID
LEFT JOIN LOAN l ON bc.CopyID = l.CopyID

```

```
GROUP BY b.BookID, b.Title, b.ISBN
ORDER BY TimesBorrowed DESC;
```

5.10.3. Query 3: Thành viên có sách quá hạn

```
SELECT
    m.MemberID,
    m.Name,
    m.Phone,
    b.Title,
    l.DueDate,
    DATEDIFF(DAY, l.DueDate, GETDATE()) AS DaysOverdue,
    DATEDIFF(DAY, l.DueDate, GETDATE()) * 5000 AS EstimatedFine
FROM LOAN l
INNER JOIN MEMBER m ON l.MemberID = m.MemberID
INNER JOIN BOOK_COPY bc ON l.CopyID = bc.CopyID
INNER JOIN BOOK b ON bc.BookID = b.BookID
WHERE l.ReturnDate IS NULL
    AND l.DueDate < CAST(GETDATE() AS DATE)
ORDER BY DaysOverdue DESC;
```

5.11. Kết luận báo cáo 4

5.11.1. Tổng kết kết quả

Báo cáo 4 đã hoàn thành việc triển khai cơ sở dữ liệu trên SQL Server:

1. Đã tạo 7 bảng với đầy đủ ràng buộc:

- 7 Primary Keys
- 7 Foreign Keys với hành vi CASCADE phù hợp
- 15 Check Constraints
- 2 Unique Constraints (bao gồm filtered unique)
- 2 Default Constraints

2. Đã tạo các Stored Procedures:

- usp_BorrowBook: Xử lý nghiệp vụ mượn sách
- usp_ReturnBook: Xử lý nghiệp vụ trả sách với tính phí phạt

3. Đã tạo Functions:

- ufn_GetBooksByAuthor: Lấy danh sách sách theo tác giả
- ufn_CalculateOverdueFine: Tính phí phạt quá hạn

4. Đã tạo Triggers:

- trg_PreventBorrow_OverdueMember: Ngăn mượn nếu thành viên có sách quá hạn

- `trg_CalculateOverdueFine`: Tự động tính phí phạt

5. Đã viết các câu truy vấn mẫu:

- Liệt kê sách đang được mượn
- Thống kê sách được mượn nhiều nhất
- Thành viên có sách quá hạn

5.11.2. Đánh giá và kế tiếp

Điểm mạnh của thiết kế:

- Ràng buộc toàn vẹn dữ liệu được thực thi ở mức cơ sở dữ liệu
- Stored procedures ẩn logic nghiệp vụ, tăng bảo mật
- Triggers tự động hóa các nghiệp vụ phức tạp
- Tên ràng buộc theo quy chuẩn (PK_, FK_, CK_, UQ_)

Bước tiếp theo (Báo cáo 5):

- Chèn dữ liệu mẫu (Sample Data)
- Kiểm tra tất cả stored procedures, functions, triggers
- Tạo các query phức tạp cho báo cáo thống kê

KẾT THÚC BÁO CÁO 4

Người thực hiện: Nhóm 3 - Lớp SE2043

Ngày hoàn thành: 03/02/2026

Giảng viên hướng dẫn: Vũ Thanh Phong

6. BÁO CÁO 5: BÁO CÁO ĐỒ ÁN CUỐI KỲ

6.1. Phân công nhiệm vụ

Bảng 47: Phân công nhiệm vụ nhóm

STT	Họ và tên	Phân công nhiệm vụ	Điểm
1	Nguyễn Ngọc Phúc	Chuẩn bị dữ liệu mẫu cho PUBLISHER, AUTHOR, BOOK, viết script INSERT	10
2	Thân Nhật Huy	Chuẩn bị dữ liệu mẫu cho BOOK_COPY, MEMBER, LOAN, viết script INSERT	10
3	Võ Hoàng Đình Trường	Thực hiện testing Stored Procedures, ghi lại kết quả	10
4	Nguyễn Thành An	Thực hiện testing Functions và Triggers, ghi lại kết quả	10
5	Nguyễn Quang Thiên Phú	Viết các truy vấn phức tạp cho báo cáo thống kê, tổng hợp kết quả	10
6	Phạm Ngọc Hưng	Kiểm tra toàn bộ, tổng hợp báo cáo cuối cùng, viết Final Thoughts	10

6.2. Giới thiệu báo cáo 5

Báo cáo 5 là báo cáo cuối cùng, tổng kết toàn bộ quá trình thiết kế và triển khai cơ sở dữ liệu Quản lý Thư viện. Báo cáo này bao gồm:

Nội dung báo cáo 5:

- Chèn dữ liệu mẫu (Sample Data) vào tất cả các bảng
- Thực hiện và kiểm tra các Stored Procedures
- Thực hiện và kiểm tra các Functions
- Thực hiện và kiểm tra các Triggers
- Các truy vấn báo cáo thống kê
- Tổng kết và Final Thoughts

6.3. Chèn dữ liệu mẫu (Sample Data)

6.3.1. Dữ liệu bảng PUBLISHER

```
INSERT INTO PUBLISHER (Name, Address, ContactNumber) VALUES
(N'Nhà xuất bản Giáo dục Việt Nam', N'81 Trần Hưng Đạo, Hà Nội', '024-38225162'),
(N'Nhà xuất bản Trẻ', N'161B Nguyễn Du, Quận 1, TP.HCM', '028-38223642'),
(N'Nhà xuất bản Trẻ hợp tác TP.HCM', N'61 Nguyễn Thị Minh Khai, Quận 1, TP.HCM', '028-38229685'),
```

```
(N'Nhà xuất bản Đà Nẵng', N'36 Điện Biên Phủ, TP. Đà Nẵng', '
0236-3822565'),
(N'NXB Công an Nhân dân', N'43 Trần Phú, Hà Nội', '024-38263267'),
(N'NXB Lao động', N'53 Nguyễn Du, Hai Bà Trưng, Hà Nội', '
024-38225261'),
(N'NXB Hồng Đức', N'36A Trần Hưng Đạo, Hoàn Kiếm, Hà Nội', '024-39382008
'),
(N'NXB Alpha Books', N'46 Đường số 7, Phường 4, Quận 5, TP.HCM', '
028-38388067');
GO
```

Listing 16: INSERT dữ liệu vào PUBLISHER

6.3.2. Dữ liệu bảng AUTHOR

```
INSERT INTO AUTHOR (Name, Nationality, YearOfBirth) VALUES
(N'Nguyen Nhat Anh', N'Việt Nam', 1980),
(N'To Hoai', N'Việt Nam', 1965),
(N'Nam Cao', N'Việt Nam', 1975),
(N'Tran Dang Khoa', N'Việt Nam', 1988),
(N'J.K. Rowling', N'Anh', 1965),
(N'Haruki Murakami', N'Nhật Bản', 1949),
(N'George Orwell', N'Anh', 1903),
(N'F. Scott Fitzgerald', N'Mỹ', 1896),
(N'Ernest Hemingway', N'Mỹ', 1899),
(N'Jane Austen', N'Anh', 1775),
(N'Leo Tolstoy', N'Nga', 1828),
(N'Victor Hugo', N'Pháp', 1802),
(N'Albert Camus', N'Pháp', 1913),
(N'Gabriel Garcia Marquez', N'Colombia', 1927),
(N'Isabel Allende', N'Chile', 1942);
GO
```

Listing 17: INSERT dữ liệu vào AUTHOR

6.3.3. Dữ liệu bảng MEMBER

```
INSERT INTO MEMBER (Name, Address, Phone, MembershipType) VALUES
(N'Nguyen Van An', N'123 Đường ABC, Quận 1, TP.HCM', '0901234567', '
Student'),
(N'Tran Thi Bich', N'456 Đường XYZ, Quận 3, TP.HCM', '0912345678', '
Faculty'),
(N'Le Van Cuong', N'789 Đường DEF, Quận 5, TP.HCM', '0923456789', '
Standard'),
(N'Pham Thi Dung', N'321 Đường GHI, Đà Nẵng', '0934567890', 'Premium
'),
(N'Vu Van Em', N'654 Đường JKL, Hà Nội', '0945678901', 'Student'),
(N'Do Thi Phong', N'987 Đường MNO, Hải Phòng', '0956789012', 'Faculty
'),
(N'Hoang Van Quy', N'147 Đường PQR, Cần Thơ', '0967890123', 'Standard
'),
```

```
(N'Nguyen Thi Sang', N'258 Đường STU, ệNgh An', '0978901234', '
Student'),
(N'Tran Van Tan', N'369 Đường VWX, ềHu', '0989012345', 'Premium'),
(N'Le Thi Uyen', N'741 Đường YZ, Buôn Ma ộThut', '0990123456', '
Faculty');
GO
```

Listing 18: INSERT dữ liệu vào MEMBER

6.3.4. Dữ liệu bảng BOOK

```
INSERT INTO BOOK (ISBN, Title, SubjectCategory, PublicationYear,
PublisherID) VALUES
('978-604-0-00123-4', N'Tôi ấthy hoa vàng trên ỏc xanh', N'ăVn ợhc',
2010, 1),
('978-604-0-00234-5', N'ấMt ềbic', N'ăVn ợhc', 2018, 1),
('978-604-0-00345-6', N'Cho tôi xin ộmt vé đi ồtui ơth', N'ăVn ợhc',
2015, 2),
('978-604-0-00456-7', N'Bên kia rào ấcm', N'ăVn ợhc', 2012, 3),
('978-0-7475-3269-9', N'Harry Potter và the Philosopher's Stone', N'
Fantasy', 1997, 5),
('978-0-7475-3849-3', N'Harry Potter và Chamber of Secrets', N'
Fantasy', 1998, 5),
('978-4-77-01536-8', N'Norwegian Wood', N'Romance', 1987, 4),
('978-0-452-28423-4', N'1Q84', N'Fiction', 2009, 4),
('978-0-452-28423-5', N'Animal Farm', N'Political Fiction', 1945, 6),
('978-0-7432-7356-5', N'The Great Gatsby', N'Classic', 1925, 7),
('978-0-684-80323-7', N'For Whom the Bell Tolls', N'Classic', 1940,
6),
('978-0-14-143951-8', N'Pride and Prejudice', N'Romance', 1813, 8),
('978-0-14-303999-8', N'War and Peace', N'Classic', 1869, 8),
('978-0-679-72126-3', N'Les Misérables', N'Classic', 1862, 9),
('978-0-375-40653-5', N'The Plague', N'Philosophy', 1947, 10),
('978-0-06-088328-7', N'One Hundred Years of Solitude', N'Magic
Realism', 1967, 11),
('978-1-5011-7033-3', N'The House of the Spirits', N'Magic Realism',
1982, 7),
('978-604-0-00567-8', N'Làng', N'ăVn ợhc', 2011, 1),
('978-604-0-00678-9', N'Chí Phèo', N'ăVn ợhc', 2009, 1),
('978-604-0-00789-0', N'ỚS đồ', N'ăVn ợhc', 2013, 2);
GO
```

Listing 19: INSERT dữ liệu vào BOOK

6.3.5. Dữ liệu bảng BOOK_AUTHOR

```
INSERT INTO BOOK_AUTHOR (BookID, AuthorID) VALUES
(1, 1), -- Tôi ấthy hoa vàng trên ỏc xanh - Nguyen Nhat Anh
(2, 2), -- ấMt ềbic - To Hoai
(3, 3), -- Cho tôi xin ộmt vé đi ồtui ơth - Nguyen Nhat Anh
(4, 4), -- Bên kia rào ấcm - Tran Dang Khoa
```

```

(5, 5), -- Harry Potter 1 - J.K. Rowling
(6, 5), -- Harry Potter 2 - J.K. Rowling
(7, 6), -- Norwegian Wood - Haruki Murakami
(8, 6), -- 1Q84 - Haruki Murakami
(9, 7), -- Animal Farm - George Orwell
(10, 8), -- The Great Gatsby - F. Scott Fitzgerald
(11, 9), -- For Whom the Bell Tolls - Ernest Hemingway
(12, 10), -- Pride and Prejudice - Jane Austen
(13, 11), -- War and Peace - Leo Tolstoy
(14, 12), -- Les Misérables - Victor Hugo
(15, 13), -- The Plague - Albert Camus
(16, 14), -- One Hundred Years of Solitude - Gabriel Garcia Marquez
(17, 15), -- The House of the Spirits - Isabel Allende
(18, 2), -- Làng - Tô Hoài
(19, 4), -- Chí Phèo - Nam Cao
(20, 3); -- Ô S đỏ - Nam Cao
GO

```

Listing 20: INSERT dữ liệu vào BOOK_AUTHOR

6.3.6. Dữ liệu bảng BOOK_COPY

```

INSERT INTO BOOK_COPY (BookID, Condition, Status) VALUES
-- Sách ềtng ệVit
(1, 'New', 'Available'), (1, 'New', 'Available'), (1, 'Good', 'Available'),
(2, 'Good', 'Available'), (2, 'Fair', 'Available'),
(3, 'New', 'Available'), (3, 'Good', 'Available'), (3, 'Good', 'Available'),
(4, 'Fair', 'Available'), (4, 'Poor', 'Maintenance'),
-- Harry Potter
(5, 'New', 'Available'), (5, 'New', 'Available'), (5, 'Good', 'Borrowed'),
(6, 'Good', 'Available'), (6, 'Fair', 'Available'),
-- Murakami
(7, 'Good', 'Available'), (7, 'Good', 'Available'),
(8, 'New', 'Available'), (8, 'New', 'Available'), (8, 'Fair', 'Available'),
-- Classics
(9, 'Good', 'Available'), (9, 'Fair', 'Available'),
(10, 'New', 'Available'), (10, 'Good', 'Available'), (10, 'Good', 'Available'),
(11, 'Fair', 'Available'), (11, 'Poor', 'Available'),
(12, 'Good', 'Available'), (12, 'Good', 'Available'),
(13, 'New', 'Available'), (13, 'Fair', 'Available'),
(14, 'Good', 'Available'), (14, 'Good', 'Available'),
(15, 'Fair', 'Available'), (15, 'Fair', 'Available'),
(16, 'New', 'Available'), (16, 'Good', 'Available'),
(17, 'Good', 'Available'), (17, 'Fair', 'Available'),
-- More Vietnamese
(18, 'Good', 'Available'), (18, 'Good', 'Available'),

```

```
(19, 'Fair', 'Available'),
(20, 'New', 'Available'), (20, 'Good', 'Available');
GO
```

Listing 21: INSERT dữ liệu vào BOOK_COPY

6.3.7. Dữ liệu bảng LOAN

```
INSERT INTO LOAN (MemberID, CopyID, BorrowDate, DueDate, ReturnDate,
    OverdueFine) VALUES
-- mượn đang hoạt động (chưa trả)
(1, 1, '2026-01-15', '2026-01-29', NULL, 0),
(2, 5, '2026-01-20', '2026-02-03', NULL, 0),
(3, 7, '2026-01-25', '2026-02-08', NULL, 0),
(4, 10, '2026-01-10', '2026-01-24', NULL, 0), -- Quá hạn!
(5, 12, '2026-01-28', '2026-02-11', NULL, 0),
-- mượn đã trả đúng hạn
(1, 2, '2025-12-01', '2025-12-15', '2025-12-14', 0),
(2, 4, '2025-12-05', '2025-12-19', '2025-12-18', 0),
(3, 8, '2025-12-10', '2025-12-24', '2025-12-22', 0),
(6, 9, '2025-12-15', '2025-12-29', '2025-12-28', 0),
(7, 11, '2025-12-20', '2026-01-03', '2026-01-02', 0),
-- mượn đã trả (quá hạn - có phạt)
(4, 13, '2025-11-01', '2025-11-15', '2025-11-20', 25000), -- 5 ngày
    quá hạn
(5, 14, '2025-11-10', '2025-11-24', '2025-11-30', 60000), -- 12 ngày
    quá hạn
(8, 15, '2025-11-15', '2025-11-29', '2025-11-25', 0), -- Trả sớm
(9, 16, '2025-11-20', '2025-12-04', '2025-12-10', 60000), -- 12 ngày
    quá hạn
(10, 17, '2025-11-25', '2025-12-09', '2025-12-08', 0); -- Trả sớm
GO
```

Listing 22: INSERT dữ liệu vào LOAN

6.3.8. Tổng kết dữ liệu đã chèn

Bảng 48: Tổng kết dữ liệu mẫu

STT	Tên bảng	Số bản ghi
1	PUBLISHER	8
2	AUTHOR	15
3	MEMBER	10
4	BOOK	20
5	BOOK_AUTHOR	20
6	BOOK_COPY	35
7	LOAN	15
TỔNG CỘNG		123

6.4. Kiểm tra Stored Procedures

6.4.1. Test 1: Mượn sách thành công

Kịch bản test: Thành viên ID=1 mượn bản sao ID=2

```
EXEC dbo.usp_BorrowBook
    @MemberID = 1,
    @CopyID = 3,
    @DueDays = 14;
```

Kết quả mong đợi:

- Thông báo "Mượn sách thành công!"
- Ngày hết hạn = 14 ngày sau ngày mượn
- Trạng thái bản sao sách chuyển thành 'Borrowed'
- Bản ghi mới được thêm vào LOAN

Verifying query:

```
SELECT l.*, bc.Status
FROM LOAN l
INNER JOIN BOOK_COPY bc ON l.CopyID = bc.CopyID
WHERE l.CopyID = 3 AND l.ReturnDate IS NULL;
```

6.4.2. Test 2: Trả sách đúng hạn

Kịch bản test: Trả bản sao ID=3 (vừa mượn ở Test 1)

```
EXEC dbo.usp_ReturnBook
    @CopyID = 3,
    @ReturnDate = '2026-02-05';
```

Kết quả mong đợi:

- Thông báo "Trả sách thành công!"
- DaysOverdue = 0 (trả đúng hạn)
- OverdueFine = 0
- Trạng thái bản sao chuyển thành 'Available'

6.4.3. Test 3: Trả sách quá hạn

Kịch bản test: Trả bản sao ID=5 (đã mượn từ 2026-01-20, hạn 2026-02-03, trả 2026-02-10)

```
EXEC dbo.usp_ReturnBook
    @CopyID = 5,
    @ReturnDate = '2026-02-10';
```

Kết quả mong đợi:

- DaysOverdue = 7 (quá hạn 7 ngày)
- OverdueFine = $7 \times 5000 = 35000$ VNĐ

6.4.4. Test 4: Mượn sách khi sách không có sẵn

Kịch bản test: Thử mượn bản sao đã được mượn

```
EXEC dbo.usp_BorrowBook
    @MemberID = 2,
    @CopyID = 5,    -- Copy này đang Borrowed
    @DueDays = 14;
```

Kết quả mong đợi:

- Lỗi: "Sách không có sẵn để mượn!"
- Không có bản ghi mới được tạo

6.5. Kiểm tra Functions

6.5.1. Test Function 1: ufn_GetBooksByAuthor

Kịch bản test: Lấy danh sách sách của J.K. Rowling

```
SELECT * FROM dbo.ufn_GetBooksByAuthor('J.K. Rowling');
```

Kết quả mong đợi:

- Harry Potter và the Philosopher's Stone
- Harry Potter và Chamber of Secrets

6.5.2. Test Function 2: ufn_CalculateOverdueFine

Kịch bản test: Tính phí phạt cho các trường hợp khác nhau

```
-- âTr đúng ăhn
SELECT dbo.ufn_CalculateOverdueFine('2026-01-01', '2026-01-10', 5000)
    AS Fine;
-- ếtKt ăqu: 0

-- âTr ộmun 5 ngày
SELECT dbo.ufn_CalculateOverdueFine('2026-01-01', '2026-01-16', 5000)
    AS Fine;
-- ếtKt ăqu: 25000

-- ưCha ătr (NULL)
SELECT dbo.ufn_CalculateOverdueFine('2026-01-01', NULL, 5000) AS Fine
;
-- ếtKt ăqu: 0
```

6.6. Kiểm tra Triggers

6.6.1. Test Trigger 1: Ngăn mượn nếu thành viên có sách quá hạn

Kịch bản test: Thành viên ID=4 đang có sách quá hạn (CopyID=10)

```
-- ửTh ượn sách khác
EXEC dbo.usp_BorrowBook
    @MemberID = 4,
    @CopyID = 6,
    @DueDays = 14;
```

Kết quả mong đợi:

- Lỗi: "Thành viên đang có sách quá hạn chưa trả. Không thể mượn thêm!"
- Không thể mượn sách mới

Sau khi trả sách quá hạn:

```
-- ảTr sách quá ạn
EXEC dbo.usp_ReturnBook @CopyID = 10, @ReturnDate = '2026-02-05';

-- ửTh ượn ại
EXEC dbo.usp_BorrowBook @MemberID = 4, @CopyID = 6, @DueDays = 14;
-- ết ả: ượn thành công!
```

6.6.2. Test Trigger 2: Tự động tính phí phạt

Kịch bản test: Update ReturnDate cho giao dịch quá hạn

```
-- ảTo giao ịch ượn
INSERT INTO LOAN (MemberID, CopyID, BorrowDate, DueDate, ReturnDate,
    OverdueFine)
VALUES (1, 4, '2026-01-01', '2026-01-15', NULL, 0);

-- ậCp ậht ReturnDate (sau ạn)
UPDATE LOAN
SET ReturnDate = '2026-01-25'
WHERE CopyID = 4 AND ReturnDate IS NULL;

-- ểKim tra OverdueFine
SELECT LoanID, DueDate, ReturnDate, OverdueFine
FROM LOAN
WHERE CopyID = 4;
```

Kết quả mong đợi:

- OverdueFine tự động được tính = 10 ngày × 5000 = 50000 VNĐ

6.7. Các truy vấn báo cáo thống kê

6.7.1. Báo cáo 1: Top 10 sách được mượn nhiều nhất

```
SELECT TOP 10
    b.BookID,
    b.Title,
    COUNT(l.LoanID) AS TotalLoans,
```



```

COUNT(CASE WHEN l.ReturnDate IS NULL THEN 1 END) AS
    CurrentlyBorrowed
FROM BOOK b
INNER JOIN BOOK_COPY bc ON b.BookID = bc.BookID
LEFT JOIN LOAN l ON bc.CopyID = l.CopyID
GROUP BY b.BookID, b.Title
ORDER BY TotalLoans DESC;

```

Listing 23: Thống kê sách phổ biến

Kết quả mẫu:

BookID	Title	TotalLoans	CurrentlyBorrowed
1	Tôi thấy hoa vàng...	3	1
5	Harry Potter 1	2	1
10	The Great Gatsby	3	0
...			

6.7.2. Báo cáo 2: Thành viên có sách quá hạn

```

SELECT
    m.MemberID,
    m.Name,
    m.Phone,
    m.MembershipType,
    b.Title,
    l.BorrowDate,
    l.DueDate,
    DATEDIFF(DAY, l.DueDate, GETDATE()) AS DaysOverdue,
    DATEDIFF(DAY, l.DueDate, GETDATE()) * 5000 AS EstimatedFine
FROM LOAN l
INNER JOIN MEMBER m ON l.MemberID = m.MemberID
INNER JOIN BOOK_COPY bc ON l.CopyID = bc.CopyID
INNER JOIN BOOK b ON bc.BookID = b.BookID
WHERE l.ReturnDate IS NULL
    AND l.DueDate < CAST(GETDATE() AS DATE)
ORDER BY DaysOverdue DESC;

```

Listing 24: Danh sách thành viên quá hạn

Kết quả mẫu:

Thành viên Phạt ước tính	Sách	Ngày mượn	Quá hạn (ngày)
Phạm Thị Dung 60,000 VNĐ	The Great Gatsby	2026-01-10	12
...			

6.7.3. Báo cáo 3: Thống kê theo nhà xuất bản

```

SELECT
    p.PublisherID,
    p.Name AS PublisherName,
    COUNT(DISTINCT b.BookID) AS TotalBooks,
    COUNT(DISTINCT bc.CopyID) AS TotalCopies,
    COUNT(l.LoanID) AS TotalLoans
FROM PUBLISHER p
LEFT JOIN BOOK b ON p.PublisherID = b.PublisherID
LEFT JOIN BOOK_COPY bc ON b.BookID = bc.BookID
LEFT JOIN LOAN l ON bc.CopyID = l.CopyID
GROUP BY p.PublisherID, p.Name
ORDER BY TotalBooks DESC;

```

Listing 25: Thống kê theo nhà xuất bản

6.7.4. Báo cáo 4: Doanh thu phí phạt theo tháng

```

SELECT
    YEAR(l.ReturnDate) AS Year,
    MONTH(l.ReturnDate) AS Month,
    COUNT(*) AS TotalReturned,
    SUM(l.OverdueFine) AS TotalFineAmount,
    AVG(l.OverdueFine) AS AverageFine
FROM LOAN l
WHERE l.ReturnDate IS NOT NULL
    AND l.OverdueFine > 0
GROUP BY YEAR(l.ReturnDate), MONTH(l.ReturnDate)
ORDER BY Year DESC, Month DESC;

```

Listing 26: Thống kê doanh thu phí phạt

6.8. Final Thoughts - Tổng kết và Đánh giá

6.8.1. Tổng kết toàn bộ dự án

Quá trình thực hiện đồ án cơ sở dữ liệu Quản lý Thư viện đã đi qua 5 báo cáo, từ khâu phân tích yêu cầu đến triển khai thực tế:

Bảng 49: Tổng tiến độ thực hiện dự án

Báo cáo	Tên	Hoàn thành	Kết quả chính
1	Decomposition	[OK]	Xác định 6 thực thể, 5 mối quan hệ, ER Diagram
2	Pattern Recognition	[OK]	Chuyển sang Relational Model, xác định FDs
3	Abstraction	[OK]	Chuẩn hóa đến 3NF, tất cả bảng đạt chuẩn
4	Algorithm Design	[OK]	Tạo bảng, SP, Functions, Triggers
5	Final Report	[OK]	Chèn dữ liệu, testing, báo cáo

6.8.2. Những bài học kinh nghiệm

1. Về thiết kế cơ sở dữ liệu:

- Phân tích yêu cầu kỹ lưỡng ngay từ đầu rất quan trọng
- Chuẩn hóa (normalization) giúp tránh nhiều vấn đề trong tương lai
- Việc tách BOOK và BOOK_COPY là quyết định đúng đắn

2. Về triển khai SQL Server:

- Sử dụng constraint ở mức cơ sở dữ liệu giúp đảm bảo tính toàn vẹn
- Stored procedures giúp ẩn logic nghiệp vụ và tăng bảo mật
- Triggers hữu ích cho các ràng buộc phức tạp nhưng cần sử dụng cẩn thận

3. Về làm việc nhóm:

- Phân chia nhiệm vụ rõ ràng giúp hoàn thành dự án hiệu quả
- Review code lẫn nhau giúp tránh lỗi sai
- Giao tiếp và phối hợp là chìa khóa thành công

6.8.3. Hạn chế và cải tiến tiềm năng

Các hạn chế hiện tại:

1. **Giao diện người dùng:** Chưa có UI/UX, chỉ có backend
2. **Xác thực người dùng:** Chưa có hệ thống login/phân quyền
3. **Đặt sách trước:** Chưa có chức năng reservation (đặt trước)
4. **Thông báo tự động:** Chưa có email/SMS nhắc hạn trả sách
5. **Báo cáo nâng cao:** Có thể thêm nhiều báo cáo thống kê hơn

Đề xuất cải tiến trong tương lai:

1. **Web Application:** Xây dựng giao diện web với ASP.NET hoặc React
2. **Authentication:** Thêm bảng USER với roles (Admin, Librarian, Member)
3. **Reservation System:** Thêm bảng RESERVATION cho đặt sách trước
4. **Notification:** Tính năng gửi email tự động khi sách quá hạn
5. **Analytics:** Dashboard với biểu đồ thống kê trực quan
6. **Barcode/QR Code:** Quản lý sách bằng mã vạch hoặc QR code

6.8.4. Kết luận

Đồ án Cơ sở dữ liệu Quản lý Thư viện đã được hoàn thành thành công với đầy đủ các yêu cầu:

- **Đã hoàn thành:** 5 báo cáo theo yêu cầu bài tập
- **Đã thiết kế:** ER Diagram đầy đủ với 6 thực thể chính
- **Đã chuẩn hóa:** Tất cả 7 bảng đều đạt 3NF
- **Đã triển khai:** Full T-SQL với tables, SP, functions, triggers
- **Đã kiểm thử:** Dữ liệu mẫu, test cases, và kết quả

Cơ sở dữ liệu này có thể được sử dụng như một nền tảng vững chắc để xây dựng ứng dụng quản lý thư viện hoàn chỉnh trong thực tế. Với thiết kế tốt, hệ thống có thể mở rộng và phát triển theo nhu cầu thực tế của thư viện.

Cam kết của nhóm: Chúng tôi đã nỗ lực hết sức để hoàn thành đồ án này một cách tốt nhất. Mặc dù còn những hạn chế, nhưng đây là một kinh nghiệm quý báu giúp chúng tôi hiểu rõ quy trình thiết kế và triển khai cơ sở dữ liệu trong thực tế.

"Học hành không bao giờ kết thúc, và mỗi dự án là một cơ hội để học hỏi thêm."

KẾT THÚC BÁO CÁO 5

Người thực hiện: Nhóm 3 - Lớp SE2043

Ngày hoàn thành: 05/02/2026

Giảng viên hướng dẫn: Vũ Thanh Phong

KẾT THÚC TOÀN BỘ ĐỒ ÁN

CƠ SỞ DỮ LIỆU QUẢN LÝ THƯ VIỆN

(Library Management Database)

Người thực hiện: Nhóm 3 - Lớp SE2043

Đại học FPT

Tháng 02 năm 2026

TÀI LIỆU THAM KHẢO

1. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database System Concepts* (7th ed.). McGraw-Hill Education.
2. Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems* (7th ed.). Pearson.
3. Microsoft Documentation. (2025). *SQL Server Documentation*. Retrieved from <https://docs.microsoft.com/sql>
4. Course Materials - DBI202 - FPT University
5. GeeksforGeeks. (2025). *ER Diagram of Library Management System*. Retrieved from <https://www.geeksforgeeks.o>

KẾT THÚC TOÀN BỘ BÁO CÁO

Người thực hiện: Nhóm 3 - Lớp SE2043

Ngày hoàn thành: 26/01/2026

Giảng viên hướng dẫn: Vũ Thanh Phong