**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**



**ĐỒ ÁN**

**MÔN HỌC: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**Đề tài: Quản lý thư viện**

Giảng viên hướng dẫn: **ThS. Đỗ Thị Minh Phụng**

Sinh viên thực hiện:

19521816 - Nguyễn Quốc Lưu

19521753 - Nguyễn Hải Linh

19522208 - Nguyễn Hữu Thắng

19520072 - Lê Kim Hằng

# MỤC LỤC

[MỤC LỤC 2](#_Toc76674307)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 8](#_Toc76674308)

[I. Đặt vấn đề 8](#_Toc76674309)

[II. Mục tiêu 9](#_Toc76674310)

[III. Công cụ sử dụng 11](#_Toc76674311)

[CHƯƠNG 2: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ DỮ LIỆU 12](#_Toc76674312)

[I. Đặc tả bài toán 12](#_Toc76674313)

[II. Phân tích yêu cầu 14](#_Toc76674314)

[1. Yêu cầu phi chức năng 14](#_Toc76674315)

[2. Thiết kế mô hình quan hệ 15](#_Toc76674316)

[a. Mô hình thực thể mối kết hợp 15](#_Toc76674317)

[b. Thiết kế dữ liệu 15](#_Toc76674318)

[i. Sơ đồ logic 15](#_Toc76674319)

[ii. Mô hình dữ liệu quan hệ 16](#_Toc76674320)

[c. Mô tả thành phần dữ liệu 16](#_Toc76674321)

[d. Mô tả ràng buộc toàn vẹn 21](#_Toc76674322)

[i. Ràng buộc khoá chính 21](#_Toc76674323)

[ii. Ràng buộc khoá ngoại 21](#_Toc76674324)

[CHƯƠNG 3: CÁC CÔNG NGHỆ SỬ DỤNG 23](#_Toc76674325)

[I. Oracle: 23](#_Toc76674326)

[II. JDBC (Java Database Connectivity): 23](#_Toc76674327)

[III. JavaSwing: 25](#_Toc76674328)

[IV. NetBeans: 25](#_Toc76674329)

[V. Git: 25](#_Toc76674330)

[CHƯƠNG 4: XÂY DỰNG VÀ QUẢN LÍ GIAO TÁC 27](#_Toc76674331)

[I. Trigger: 27](#_Toc76674332)

[1. Trigger trong Oracle 27](#_Toc76674333)

[2. Danh sách các trigger 29](#_Toc76674334)

[3. Mô tả một số Trigger 30](#_Toc76674335)

[a. Tự động tạo phiếu phạt 30](#_Toc76674336)

[b. Tính số ngày mượn 32](#_Toc76674337)

[II. Stored Procedure 35](#_Toc76674338)

[1. Stored Procedure trong Oracle 35](#_Toc76674339)

[2. Transaction trong Oracle 36](#_Toc76674340)

[2.1. Định nghĩa 36](#_Toc76674341)

[2.2. Cấu trúc Transaction 36](#_Toc76674342)

[2.3. Transaction control 37](#_Toc76674343)

[3. Danh sách các Stored Procedure: 37](#_Toc76674344)

[4. Mô tả một số Stored Procedure 39](#_Toc76674345)

[a. Thêm phiếu mượn sách 39](#_Toc76674346)

[b. Thêm phiếu trả sách 40](#_Toc76674347)

[CHƯƠNG 5: XỬ LÝ TRUY XUẤT ĐỒNG THỜI 43](#_Toc76674348)

[I. Các mức cô lập trong Oracle: 43](#_Toc76674349)

[1. Read Committed: 43](#_Toc76674350)

[2. Serializable: 43](#_Toc76674351)

[3. Read-only: 44](#_Toc76674352)

[II. Cơ chế khóa: 44](#_Toc76674353)

[III. Deadlock 46](#_Toc76674354)

[IV. Mô tả đồ án môn học 47](#_Toc76674355)

[1. Lost update 47](#_Toc76674356)

[a. Mô tả tình huống 47](#_Toc76674357)

[b. Bảng mô tả Lost Update 47](#_Toc76674358)

[c. Nguyên nhân và giải pháp 48](#_Toc76674359)

[2. Unrepeatable read 48](#_Toc76674360)

[a. Mô tả tình huống 48](#_Toc76674361)

[b. Bảng mô tả Unrepeatable read 48](#_Toc76674362)

[c. Nguyên nhân và giải pháp 50](#_Toc76674363)

[3. Phantom 50](#_Toc76674364)

[a. Mô tả tình huống 50](#_Toc76674365)

[b. Bảng mô tả Phantom 50](#_Toc76674366)

[4. Dirty read 51](#_Toc76674367)

[a. Mô tả tình huống 51](#_Toc76674368)

[b. Bảng mô tả Dirty read 52](#_Toc76674369)

[c. Nguyên nhân và giải pháp 53](#_Toc76674370)

[CHƯƠNG 6: THIẾT KẾ GIAO DIỆN 54](#_Toc76674371)

[I. Danh sách các màn hình 54](#_Toc76674372)

[1. Đăng nhập 54](#_Toc76674373)

[2. Đổi mật khấu 54](#_Toc76674374)

[3. Độc giả 54](#_Toc76674375)

[4. Thủ thư 54](#_Toc76674376)

[II. Mô tả các màn hình 55](#_Toc76674377)

[1. Đăng nhập 55](#_Toc76674378)

[2. Đổi mật khẩu 56](#_Toc76674379)

[3. Độc giả 58](#_Toc76674380)

[a. Trang chủ 58](#_Toc76674381)

[b. Thông tin sách 59](#_Toc76674382)

[4. Thủ thư 60](#_Toc76674383)

[a. Trang chủ 60](#_Toc76674384)

[b. Quản lý độc giả 61](#_Toc76674385)

[c. Form thêm,sửa độc giả 62](#_Toc76674386)

[d. Quản lý sách 65](#_Toc76674387)

[e. Form thêm, sửa sách 66](#_Toc76674388)

[f. Quản lý mượn sách 67](#_Toc76674389)

[g. Form thêm,sửa phiếu mượn sách 68](#_Toc76674390)

[h. Quản lý phiểu trả sách 70](#_Toc76674391)

[i. Form phiếu trả sách 72](#_Toc76674392)

[j. Quản lý phiếu phạt 73](#_Toc76674393)

[k. Hóa đơn phiếu phạt 74](#_Toc76674394)

[l. Báo cáo, thống kê 75](#_Toc76674395)

[m. Report Thống Kê Sách Đã Đang Còn Trong Thư Viện 76](#_Toc76674396)

[n. Report Thống Kê Sách Đã Mượn hết 77](#_Toc76674397)

[o. Report Thống Kê Top 5 Sách Được Mượn Nhiều Nhất Năm 77](#_Toc76674398)

[p. Report Thống Kê Số Lượt Mượn Sách Theo Mã Sách Vào Các Năm 77](#_Toc76674399)

[q. Report Thống Kê Số Lượt Mượn Sách Theo Tháng Trong Năm 78](#_Toc76674400)

[CHƯƠNG 7: KẾT LUẬN 79](#_Toc76674401)

[I. Kết quả đạt được 79](#_Toc76674402)

[II. Hạn chế 79](#_Toc76674403)

[III. Hướng phát triển 79](#_Toc76674404)

[PHỤ LỤC 1: BẢNG PHÂN CÔNG CÔNG VIỆC 81](#_Toc76674405)

[PHỤ LỤC 2: TÀI LIỆU THAM KHẢO 82](#_Toc76674406)

**LỜI CẢM ƠN**

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay,chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô, gia đình và bạn bè.

Với lòng biết ơn sâu sắc, em xin gửi đến cô Đỗ Thị Minh Phụng giảng viên Trường Đại học Công nghệ thông tin với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian vừa qua.

Và đặc biệt, trong học kỳ này, khoa đã tổ chức cho chúng em được tiếp cận với môn học mà theo em là rất hữu ích đối với sinh viên ngành khoa Hệ thống thông tin. Đó là môn học Hệ quản trị cơ sở dữ liệu. Em xin chân thành cảm ơn cô  đã tận tâm hướng dẫn chúng em qua từng buổi học với lớp cũng như những buổi nói chuyện, thảo luận. Nếu không có những lời hướng dẫn, dạy bảo của cô thì bài báo cáo này của chúng em rất khó có thể hoàn thiện được. Một lần nữa, chúng em xin chân thành cảm ơn cô.

Trong suốt quá trình làm việc, các thành viên đã nhóm tiếp thu những kiến thức mà cô truyền dạy và phát huy hiệu quả để xây dựng và hoàn thiện đề tài một cách tốt nhất.

Bước đầu còn hạn chế và nhiều bỡ ngỡ cũng như thời gian và kiến thức còn hạn hẹp, kinh nghiệm thực tế còn non yếu nên bài báo cáo này không thể tránh khỏi những thiếu sót trong cách hiểu, lỗi trình bày. Em rất mong nhận được những ý kiến đóng góp quý báu của cô để kiến thức của chúng em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiện bổ sung, nâng cao ý thức của mình.

Sau cùng, em xin kính chúc cô thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

**Nhóm em xin trân trọng cảm ơn!**

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1. **Đặt vấn đề**

Trong cuộc sống tất bật hiện nay, mọi việc đều được đơn giản và tối ưu hóa thời gian thông qua việc ứng dụng công nghệ thông tin vào xử lý các công việc hàng ngày**.** Công nghệ thông tin và truyền thông giờ đây đã có mặt ở nhiều phương diện trong cuộc sống hàng ngày của chúng ta, từ thương mại đến giải trí và thậm chí cả văn hóa, xã hội và giáo dục. Ngày nay, điện thoại di động, máy tính để bàn, thiết bị cầm tay, thư điện tử và việc sử dụng Internet đã trở thành tâm điểm trong văn hóa và cộng đồng của chúng ta, là cầu nối không thể tách rời trong nhịp sống toàn cầu hóa.

Ứng dụng và phát triển công nghệ thông tin nhằm thúc đẩy công cuộc đổi mới, phát triển nhanh và hiện đại hóa các các ngành nghề, hỗ trợ có hiệu quả quá trình chủ động hội nhập thế giới, nâng cao chất lượng cuộc sống nhân dân, và tạo khả năng đi tắt đón đầu để thực hiện thắng lợi sự nghiệp công nghiệp hóa, hiện đại hóa. Công nghệ thông tin giữ một vai trò rất quan trọng trong sự phát triển của xã hội trong thời đại ngày nay, là nhân tố quan trọng, là kênh kết nối trao đổi giữa các thành phần của xã hội, văn hóa, giáo dục, kinh tế và thời đại toàn cầu hóa.

Cùng với sự phát triển nhanh chóng của phần cứng máy tính thì phần mềm máy tính ngày nay được tạo ra với vô số tính năng phong phú, mức độ tin cậy cao, tốc độ xử lý cao để xử lý các thuật toán phức tạp hoặc lượng dữ liệu lớn;tính bảo mật đáp ứng mức độ rủi ro chấp nhận được đối với người dùng, phần mềm hay môi trường sử dụng.

Một ví dụ cụ thể, việc quản lý thư viện trong các trường Đại học.Thư viện là nơi có chức năng lưu trữ, phục vụ việc đọc và tra cứu thông tin đến nhiều lĩnh vực khoa học khác nhau. Thư viện phục vụ nhu cầu của độc giả dưới nhiều hình thức phục vụ nhu cầu của người đọc. Tuy nhiên việc sinh viên trong các trường học lại quá nhiều dẫn đến việc kiểm soát vất vả, mất nhiều thời gian. Nếu không có sự hỗ trợ của tin học hóa thì quy trình quản lí sẽ rất khó khăn, rườm rà, mất nhiều công sức mới có thể quản lí toàn bộ hồ sơ danh sách sinh viên mượn trả sách cũng như kho tàng sách mang nguồn tri thức cho thư viện. Đa số các quy trình trên đều mang tính thủ công rất ít tự động mà sự chính xác và hiệu quả không cao. Một số nghiệp vụ như tra cứu, thống kê và hiệu chỉnh thông tin rất là lãng phí thời gian và vô bổ. Bên cạnh đó lưu trữ rất khó khăn, vất vả và tốn nhiều chi phí. Đối với việc tra cứu sách của độc giả còn gặp nhiều khó khăn. Bên cạnh đó khả năng đáp ứng nhu cầu của độc giả trong thư viện là thấp vào những thời gian cao điểm như ôn thi bởi số lượng sách có hạn mà nhu cầu của độc gải rất cao.

Cùng với sự phát triển của công nghệ thông tin, việc đưa tin học hóa vào thư viện đã giảm bớt đươc công sức của cán bộ thư viện, làm tăng hiệu quả nghiên cứu của độc giả.Việc đưa tin học vào phần nào khắc phục được những nhược điểm của hệ thống cũ. Hệ thống thư viện mới sẽ đáp ứng một cách tối đa nhu cầu của độc giả khắc phục hoàn toàn những nhược điểm của hệ thống trước đó. Đó sẽ là một hệ thống với đầy đủ các chức năng quản lý, giúp dễ dàng phân loại các đầu danh sách trong thư viện, theo dõi số lượng mua vào và số lượng sách đã mượn, quản lý các cổng thông tin trực tuyến một cách dễ dàng nhất. Điều này sẽ hỗ trợ tối ưu cho việc tìm kiếm những cuốn sách nằm ở đâu, thuộc nhà xuất bản nào, giúp người đọc có thể tìm kiếm thông tin tài liệu một cách dễ dàng nhất.

1. **Mục tiêu**

Xây dựng một hệ thống cập nhật đồng thời và tương tác đồng thời giữa các sinh viên. Đó phải là một hệ thống quản lí tích hợp, thống nhất kết nối đồng thời và dễ dàng giữa hai đối tượng là học sinh và sinh viên. Điều này mang lại cho ban quản lí thư viện nhà trường dễ dàng tổng hợp và quản lí đồng bộ, mang tính chặt chẽ cao.

Ứng dụng giúp phân loại từng đối tượng độc giả khác nhau như: độc giả là sinh viên, cán bộ giáo viên, các đối tượng khác ngoài các đối tượng trên**.** Phần mềm cho phép tìm kiếm hồ sơ, thông tin bạn đọc theo nhiều trường dữ liệu khác nhau. Hỗ trợ tìm kiếm các loại sách theo nội dung cụ thể, nhanh chóng. Phần mềm còn quản lý các bạn đọc một cách đầy đủ nhất. Phần mềm này giúp quản lý số lượng sách độc giả được phép mượn tối đa và tối thiểu, số giờ cả số lượng đầu sách độc giả mượn. Giúp kiểm soát một cách tốt nhất việc mượn sách. Phần mềm còn quản lý chi tiết, tên sách, thể loại, ngôn ngữ, tác giả, người dịch, nhà xuất bản, số lượng, kiểm soát được sách hư hỏng, tổn thất hay mất sách. Lên các báo cáo cụ thể cho nhà trường nắm rõ. Phần mềm này còn đảm bảo tối ưu bảo mật quyền sử dụng và quản lý, giao diện đơn giản dễ sử dụng, đáng được các trường học sử dụng.

Trong đó hệ thống sẽ đáp ứng các yêu cầu sau đây:

* Giúp độc giả tra cứu sách theo loại sách, theo tên sách, theo tác giả, theo ngôn ngữ, … trên các máy tính trạm.
* Cung cấp cho thủ thư thông tin về các đầu sách một độc giả đang mượn và hạn phải trả, và các cuốn sách còn đang được mượn.
* Thống kê hàng tháng số sách cho mượn theo các chủ đề, tác giả,…
* Thống kê các đầu sách không có người mượn trên 1 năm, 2 năm, 3 năm.
* Hỗ trợ thủ thư cập nhật thông tin sách, xác nhận cho mượn sách và nhận lại sách khi độc giả trả sách.
* Hỗ trợ quản lý các thông tin về độc giả dựa trên thẻ độc giả, thông tin phiếu mượn.
* Hỗ trợ chức năng quản trị chung hệ thống, trong đó người quản trị chung có thể thay đổi thông tin hoặc thêm bớt các thủ thư.

1. **Công cụ sử dụng**

Dưới đây là các công cụ mà nhóm sử dụng trong quá trình thực hiện và xây dựng đề tài:

* Oracle Developer
* PowerDesigner
* Netbean

# CHƯƠNG 2: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ DỮ LIỆU

1. **Đặc tả bài toán**

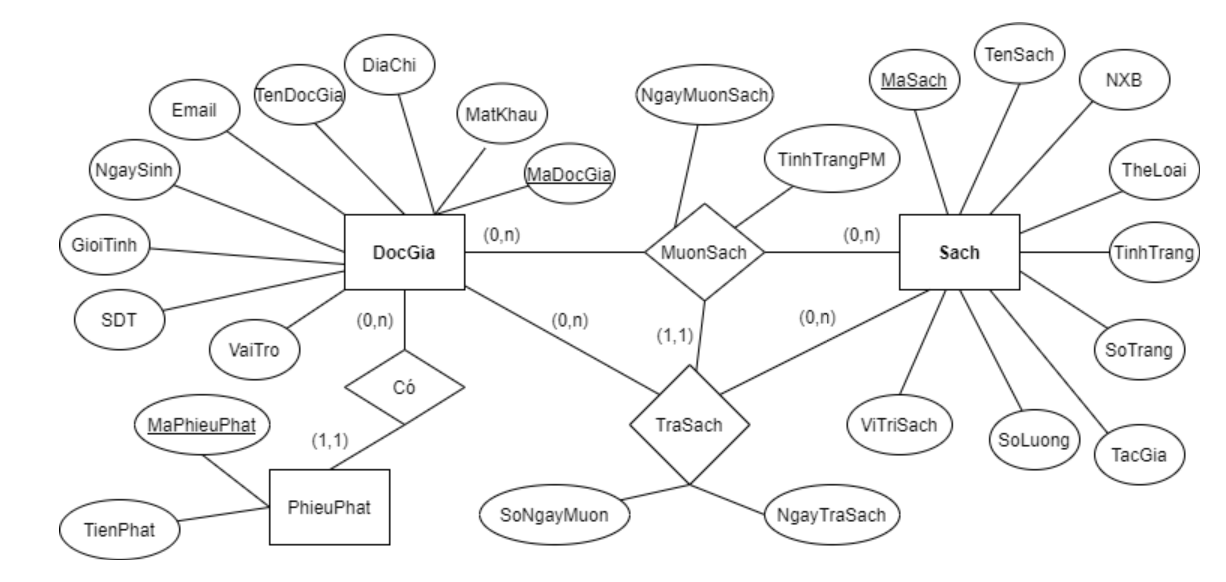
Thư viện sách là nơi dành cho ai có nhu cầu tìm hiểu thông tin tri thức từ căn bản tới chuyên sâu có thể tìm được cho mình các thể loại sách phủ hợp với nhu cầu và đúng chuyên môn của mình. Vì vậy cần một chương trình quản lí giúp nhân viên dễ dàng hơn trong công việc của mình.

Việc phân công quản lí thư viện được thực hiện như sau:

* Cơ cấu tổ chức:
* Quản trị hệ thống: admin sử dụng chức năng quản trị hệ thống để quản lí thông tin người dùng và phân quyền sử dụng cho người dùng: thủ thư, thủ kho,…
* Quản lí kho sách: quản lí toàn bộ thông tin của tất cả sách của thư viện, cho phép thêm, xóa, thay đổi thông tin của sách
* Quản lí độc giả: quản lí thông tin cá nhân cuả từng độc giả, thêm, xóa, sửa thông tin của độc giả
* Quản lí mượn trả: quản lí việc mượn trả sách của độc giã, hỗ trợ tạo phiếu mượn, gia hạn sách,…
* Thống kê: thống kê số lương sách đã được mượn trong 1 năm…giúp người quản lí thư viện nắm rõ tình hình mượn trả sách của thư viện.
* Hoạt động trả sách và cập nhật sách mới:
* Đối tượng hướng đến của phần mềm là độc giả(là người tới thư viện mượn sách.Tham gia trực tiếp vào việc mượn-trả sách) và thủ thư của thư viện.
* Thư viện tạo các thẻ sách gồm các thông tin: mã số sách, tên tác giả,  
  tên nhà xuất bản, năm xuất bản, số trang, giá tiền, tóm tắt nội dung.  
  Các đầu sách có thể được phân theo chuyên ngành hoặc tài liệu.
* Để sử dụng các dịch vụ ở thư viện thì phải bắt buộc phải có thẻ độc giả. Để đăng kí thẻ độc giả, phải làm thủ tục khai báo thông tin trực tiếp tại thư viện thông qua thủ thư. Sau khi hoàn thành thủ tục khai báo với thông tin phù hợp, sẽ được cấp phát 1 thẻ độc giả với đầy đủ các thông tin. Mỗi độc giả được cấp một thẻ độc giả, gồm các thông tin: tên, tuổi, địa chỉ, số chứng minh thư.
* Độc giả muốn mượn sách thì tra cứu các thẻ sách rồi ghi vào phiếu  
  mượn.
* Sau khi kiểm tra đầy đủ thông tin phiếu mượn, thủ thư kiểm tra điều  
  kiện mượn của độc giả và xác nhận cho phép mượn sách, thu tiền đặt  
  cọc của độc giả. Một số thông tin trong phiếu mượn được lưu lại để  
  quản lý, phiếu mượn sẽ được gài vào chỗ sách được lấy đi, sách được  
  giao cho độc giả.Khi trả sách, độc giả đem sách cần trả,xác nhận thẻ đọc sách, xác định phiếu mượn,việc trả sách được ghi nhận vào dòng ngày trả và tình trạng. Phiếu mượn được lưu lại để quản lí và theo dõi. Sách bị trả trễ lịch hẹn hoặc hỏng sẽ bị phạt.
* Khi thư viện cần bổ sung sách mới, thủ thư sẽ tìm hiểu thông tin sách cần mua và giá cả, sau đó lập phiếu nhập sách. Nếu phiếu nhập sách được xét duyệt thủ thư sẽ liên hệ với kế toán. Sau đó thủ thư sẽ tiến hành liên hệ với nhà cung cấp.Thông tin đơn đặt hàng sẽ được lưu vào hồ sơ phiếu nhập sách.Nếu sách hợp lệ, thanh toán hóa đơn cho NCC. Thủ thư cập nhật sách lại, tiến hành xử lí phân loại đánh dấu mã sách cho sách này. Lập báo cáo tình hình thư viện.

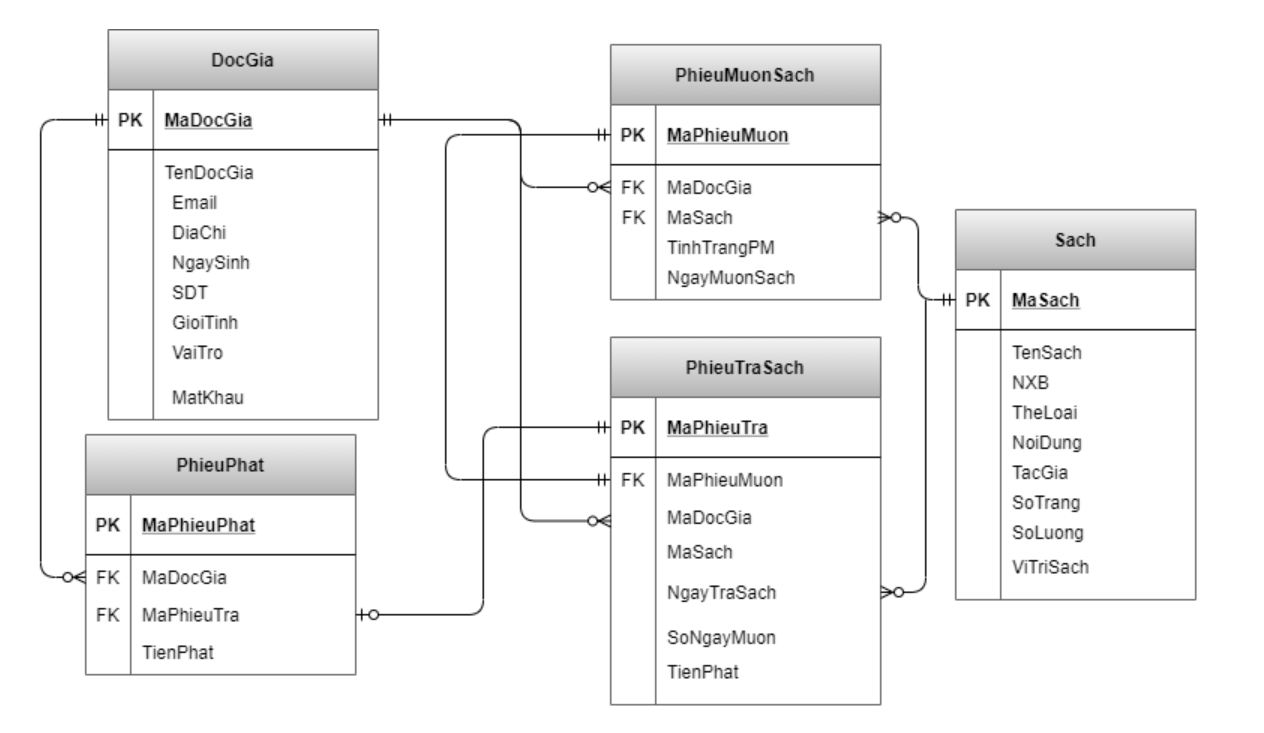
1. **Phân tích yêu cầu**
   * + 1. **Yêu cầu phi chức năng**

* **Dễ sửa lỗi**: xác định lỗi nhanh, khi sửa lỗi thì không ảnh hưởng tới các chức năng khác.
* **Dễ bảo trì**: thêm hoặc thay đổi các chức năng mới nhanh, không ảnh hưởng tới các chức năng khác.
* **Dễ thích ứng**: hệ thống vẫn hoạt động được khi đổi sang một hệ quản trị mới.
* **Dễ sử dụng**: phần mềm dễ sử dụng, giao diện than thiện với người dùng.
* **Tính chính xác**: dữ liệu trong hệ thống và dữ liệu xuất ra cần phải chính xác, đồng nhất.
* **Tính linh hoạt**: có thể thực hiện nhiều thao tác một lúc.
* **Tính thao tác**: thao tác giữa các thành phần trong data base phải được tương thích với nhau.
* **Kiểm thử**: dễ dàng kiểm tra các hoạt động của phần mềm bằng các dữ liệu mẫu để phát hiện lỗi.
* **Tính khả chuyên**: có thể chạy được trên nhiều hệ thống, trên nhiều hệ điều hành.
* **Tính an toàn**: hệ thống có thể phục hồi thông tin đã xóa, và có thể xóa vĩnh viễn.
* **Tính thực thi**: có thể xuất danh sách độc giả, danh sách tựa sách, cuốn sách, … và in ra máy khi cần.
  + - 1. **Thiết kế mô hình quan hệ**
         1. **Mô hình thực thể mối kết hợp**



* + - * 1. **Thiết kế dữ liệu**

1. **Sơ đồ logic**



1. **Mô hình dữ liệu quan hệ**

**DocGia** (MaDocGia, Password, HoTen, GioiTinh, NgSinh ,SDT, DiaChi , Email, VaiTro)

**Sach** (MaSach, TenSach, TheLoai, SoTrang, TacGia, NXB, ViTri, SoLuong, TinhTrang)

**PhieuMuonSach** (MaPhieuMuon, MaDocGia, MaSach, NgayMuonSach, TinhTrangPM)

**PhieuTraSach** (MaPhieuTra, MaPhieuMuon, MaDocGia, MaSach, NgayTraSach, SoNgayMuon, SoNgayTraTre)

**PhieuPhat** (MaPhieuPhat, MaDocGia, MaPhieuTra, TienPhat)

* + - * 1. **Mô tả thành phần dữ liệu**

Bảng **DOCGIA**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| MaDocGia | varchar(10) | Mã độc giả và tên tài khoản để đăng nhập | Khóa chính |
| MatKhau | varchar(20) | Mật khẩu đăng nhập |  |
| HoTen | varchar2(100) | Họ tên của độc giả | Ràng buộc not null |
| GioiTinh | varchar2(20) | Giới tính của độc giả |  |
| NgSinh | date | Ngày sinh của độc giả |  |
| SDT | varchar(20) | Số điện thoại của độc giả |  |
| DiaChi | varchar2(200) | Địa chỉ liên hệ của độc giả | Ràng buộc not null |
| Email | varchar(30) | Email của độc giả |  |
| VaiTro | Varchar2(50) | Vai trò là độc giả hay thủ thư | Ràng buộc not null |

Bảng **SACH**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| MaSach | varchar(10) | Mã của sách | Khóa chính |
| TenSach | Varchar2(200) | Tên của sách | Ràng buộc not null |
| TheLoai | varchar2(100) | Tên thể loại của sách |  |
| SoTrang | int | Số trang của sách |  |
| TacGia | varchar2(200) | Tác giả của cuốn sách | Ràng buộc not null |
| NXB | varchar2(100) | Nhà xuất bản | Ràng buộc not null |
| ViTri | varchar2(20) | Vị trí của sách trong thư viện | Ràng buộc not null |
| SoLuong | int | Số lượng sách ở trong thư viện |  |
| TinhTrang | varchar2(20) | Tình trạng sách |  |

Bảng **PHIEUMUONSACH**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| MaPhieuMuon | varchar(10) | Mã phiếu mượn sách | Khoá chính |
| MaDocGia | varchar(10) | Mã của độc giả | Khóa ngoại tham chiếu đến khóa chính MaDocGia của bảng DOCGIA |
| MaSach | varchar(10) | Mã của sách | Khóa ngoại tham chiếu đến khóa chính MaSach của bảng SACH |
| NgayMuonSach | date | Ngày mượn sách | Ràng buộc not null |
| TinhTrangPM | varchar2(30) | Tình trạng phiếu mượn | Ràng buộc not null |

Bảng **PHIEUTRASACH**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| MaPhieuTra | varchar(10) | Mã phiếu trả sách | Khoá chính |
| MaPhieuMuon | varchar(10) | Mã phiếu mượn sách | Khóa ngoại tham chiếu đến khóa chính MaPhieuMuon của bảng PHIEUMUONSACH |
| MaDocGia | varchar(10) | Mã của độc giả | Khóa ngoại tham chiếu đến khóa chính MaDocGia của bảng DOCGIA |
| MaSach | varchar(10) | Mã của sách | Khóa ngoại tham chiếu đến khóa chính MaSach của bảng SACH |
| NgayTraSach | date | Ngày trả sách | Ràng buộc not null |
| SoNgayMuon | int | Số ngày mượn sách |  |
| SoNgayTraTre | int | Số ngày trả trễ |  |

Bảng **PHIEUPHAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| MaPhieuPhat | varchar(10) | Mã phiếu phat | Khoá chính |
| MaDocGia | varchar(10) | Mã độc giả | Khóa ngoại tham chiếu đến khóa chính MaDocGia của bảng DOCGIA |
| MaPhieuTra | varchar(10) | Mã phiếu trả sách | Khóa ngoại tham chiếu đến khóa chính MaPhieuTra của bảng PHIEUTRASACH |
| TienPhat | int | Số tiền phạt |  |

* + - * 1. **Mô tả ràng buộc toàn vẹn**

1. **Ràng buộc khoá chính**

**RB1 :** thuộc tính MaDocGia trong bảng DOCGIA là duy nhất

**RB2:** thuộc tính MaSach trong bảng SACH là duy nhất

**RB3 :** thuộc tính MaPhieuMuon trong bảng PHIEUMUONSACH là duy nhất

**RB4 :** thuộc tính MaPhieuTra trong bảng PHIEUTRASACH là duy nhất

**RB5 :** thuộc tính MaPhieuPhat trong bảng PHIEUPHAT là duy nhất

1. **Ràng buộc khoá ngoại**

**Bảng PHIEUMUONSACH**

**RB1:** Mã của độc giả khi mượn sách cũng là mã độc giả được quản lý

**RB2:** Mã sách trong phiếu mượn cũng là mã sách được quản lý

**Bảng PHIEUTRASACH**

**RB1:** Mã của phiếu mượn sách khi trả sách cũng là mã phiếu mượn được quản lý

**RB2:** Mã của độc giả khi trả sách cũng là mã độc giả được quản lý

**RB3:** Mã sách trong phiếu trả sách cũng là mã sách được quản lý

**Bảng PHIEUPHAT**

**RB1:** Mã của độc giả khi bị phạt trùng với mã độc giả được quản lý

**RB2:**Mã của phiếu trả sách khi bị phạt trùng với mã phiếu trả được quản lý

1. **Ràng buộc miền dữ liệu**

**Bảng DOCGIA**

**RB1:** Thuộc tính GioiTinh chỉ có 2 giá trị : “Nam”, “Nữ”

**RB2:** Thuộc tính VaiTro có các giá trị : Thủ thư, Độc giả.

**Bảng SACH**

**RB1:** Thuộc tính TinhTrang chỉ có 2 giá trị : “Hết”, “Còn”

**Bảng PHIEUMUONSACH**

**RB1:** Thuộc tính TinhTrangPM chỉ có 2 giá trị : “Đang mượn”, “Đã trả”

# CHƯƠNG 3: CÁC CÔNG NGHỆ SỬ DỤNG

1. **Oracle:**

Cơ sở dữ liệu Oracle (Oracle DB) là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) từ Oracle Corporation. Được phát triển vào năm 1977 bởi Lawrence Ellison và các nhà phát triển khác, Oracle DB là một trong những công cụ cơ sở dữ liệu quan hệ được sử dụng rộng rãi và đáng tin nhất.

Hệ thống được xây dựng xung quanh bởi một framework cơ sở dữ liệu quan hệ, trong đó các đối tượng dữ liệu có thể được truy cập trực tiếp bởi người dùng (hoặc giao diện người dùng ứng dụng) thông qua ngôn ngữ truy vấn có cấu trúc (SQL). Oracle là một kiến trúc cơ sở dữ liệu quan hệ có thể mở rộng hoàn toàn và thường được sử dụng bởi các doanh nghiệp toàn cầu, họ quản lý và xử lý dữ liệu trên các mạng diện rộng và cục bộ. Cơ sở dữ liệu Oracle có thành phần mạng riêng để cho phép liên lạc qua các mạng.

Oracle DB còn được gọi là Oracle RDBMS hay gọi tắt, chỉ là Oracle.

1. **JDBC (Java Database Connectivity):**

JDBC, là viết tắt của Java Database Connectivity, là một Java API chuẩn để kết nối giữa ngôn ngữ lập trình Java và các cơ sở dữ liệu đa dạng. Khi sử dụng JDBC chúng ta có thể thực hiện nhiều tác vụ đa dạng khi làm việc với cơ sở dữ liệu như tạo, xóa cơ sở dữ liệu; tạo và thực thi các lệnh SQL hoặc MySQL; tạo, xóa các bản ghi; …

Trước khi có JDBC, ODBC API được sử dụng là API để kết nối và thực thi truy vấn với cơ sở dữ liệu. Nhưng API này có một hạn chế rõ ràng là nó sử dụng ODBC driver được viết bằng ngôn ngữ C (là phụ thuộc nền tảng và không an toàn).

Do đó, Java đã định nghĩa riêng API (là JDBC API) để thực hiện các thao tác khi làm việc với cơ sở dữ liệu, và tất nhiên các JDBC driver này được viết bằng ngôn ngữ Java để cung cấp các lợi thế rõ rệt của ngôn ngữ này.

JDBC API cung cấp các Class và Interface sau:

* + - * DriverManager: Lớp này quản lý các Database Driver. Ánh xạ các yêu cầu kết nối từ ứng dụng Java với Data driver thích hợp bởi sử dụng giao thức kết nối phụ.
      * Driver: Interface này xử lý các kết nối với Database Server. Hiếm khi, chúng ta tương tác trực tiếp với các đối tượng Driver này. Thay vào đó, chúng ta sử dụng các đối tượng DriverManager để quản lý các đối tượng kiểu này.
      * Connection: Đối tượng Connection biểu diễn ngữ cảnh giao tiếp. Interface này chứa nhiều phương thức đa dạng để tạo kết nối với một Database.
      * Statement: Chúng ta sử dụng các đối tượng được tạo từ Interface này để đệ trình các lệnh SQL tới Database. Ngoài ra, một số Interface kết thừa từ nó cung chấp nhận thêm các tham số để thực thi các thủ tục đã được lưu trữ.
      * ResultSet: Các đối tượng này giữ dữ liệu được thu nhận từ một Database sau khi chúng ta thực thi một truy vấn SQL. Nó nóng vai trò như một Iterator để cho phép chúng ta vọc qua dữ liệu của nó.
      * SQLException: Lớp này xử lý bất cứ lỗi nào xuất hiện trong khi làm việc với Database.

1. **JavaSwing:**

JavaSwing là một phần của JFC (Java Foundation Class), gồm một tập các API dùng để thiết kế giao diện cho các chương trình của Java. Được phát triển từ AWT, Swing về sau có những nhiều chức năng và linh hoạt hơn.

Ưu điểm của Swing so với AWT:

* + - * Cung cấp thêm các đối tượng mới để xây dựng giao diện đồ họa.
      * Look-and-feel: tùy biến để các thành phần giao diện của Swing nhìn giống như các thành phần giao diện của hệ điều hành.
      * Hỗ trợ các thao tác sử dụng bàn phím thay chuột.
      * Sử dụng tài nguyên hiệu quả hơn.

1. **NetBeans:**

NetBean là một IDE dành cho lập trình bằng ngôn ngữ Java, nó có thể chạy trên một số nền tảng như Windows, Linux, Solaris, OS X và các nền tảng khác hỗ trợ JVM hoặc tương thích với máy ảo Java.

Có một số mô-đun trong NetBeans IDE. Trình lược tả Netbeans để giám sát các ứng dụng Java chủ yếu được sử dụng để tìm rò rỉ bộ nhớ, tối ưu hóa tốc độ và làm được nhiều hơn thế. Công cụ GUI hoặc Graphical User Interface là một trong những công cụ thiết kế và phát triển tốt nhất để phát triển các prototype và thiết kế Swing GUI bằng cách kéo, thả các GUI Components.NetBean cũng đã tích hợp NetBeans JavaScript Editor, mở rộng hỗ trợ cho Ajax, CSS và JavaScript cùng với những thứ khác.

1. **Git:**

Git là một hệ thống quản lý phiên bản (Version Control System) dùng để quản lý và kiểm tra các phiên bản mã nguồn khác nhau trong quá trình phát triển mã nguồn. Git sử dụng mô hình phân tán, ngược lại so với SVN hoặc CSV. Mỗi nơi lưu source sẽ được gọi là Repositories, không cần lưu trữ tập trung một nơi, mà mỗi thành viên trong team sẽ có một repository ở máy của riêng mình. Có 2 dạng là Local repository và Remote repositories. Lập trình viên có thể thực hiện mọi công việc trên bản sao (local) repository trên máy tính. Sau khi hoàn thiện công việc rồi có thể đẩy lên Remote repository đểchia sẻ cho người khác.

Git có nhiều ưu điểm, đặc biệt là giải quyết xung đột (conflict) khi chia tách và nhập các nhánh phần mềm. Trong khi với SVN, việc này luôn là ác mộng vì các xung đột trong quá trình tách/nhập, còn với Git việc này quá dễ dàng.

# CHƯƠNG 4: XÂY DỰNG VÀ QUẢN LÍ GIAO TÁC

1. **Trigger:**

Trigger trong Oracle

Trigger là một chương trình lưu trữ nó sẽ tự động thực thi khi một vài sự kiện xảy ra. Thực tế Trigger được viết để phản hồi lại những sự kiện xảy ra khi thực hiện các lệnh dưới đây:

* Các câu lệnh thuộc nhóm lệnh kiểm soát dữ liệu (DELETE, INSERT, or UPDATE).
* Các câu lệnh thuộc nhóm lệnh định nghĩa dữ liệu DML (CREATE, ALTER, or DROP).
* Các câu lệnh hoạt động dữ liệu (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).
* Lợi ích khi sử dụng Trigger:
* Ngăn ngừa những giao tác không hợp lệ.
* Tự động tạo ra những cột dữ liệu phụ.
* Nhân rộng bảng đồng bộ.
* Cú pháp:

CREATE [OR REPLACE ] TRIGGER trigger\_name

{BEFORE | AFTER | INSTEAD OF }

{INSERT [OR] | UPDATE [OR] | DELETE}

[OF col\_name]

ON table\_name

[FOR EACH ROW]

WHEN (condition)

DECLARE

Declaration-statements

BEGIN

Executable-statements

EXCEPTION

Exception-handling-statements

END;

* Giải thích:
* CREATE [OR REPLACE ] TRIGGER trigger\_name: tạo hoặc thay thế một trigger đã tồn tại thành trigger\_name.
* {BEFORE | AFTER | INSTEAD OF }: cái này chỉ định khi nào trigger được thực thi. INSTEAD OF được sử dụng để tạo trigger trên view.
* {INSERT [OR] | UPDATE [OR] | DELETE}: chỉ định cụ thể lệnh nào thuộc loại lệnh DML được thực thi.
* [OF col\_name]: chỉ định cụ thể cột nào sẽ được cập nhật
* ON table\_name: chỉ định trigger sẽ được thực thi trên bảng nào.
* [REFERENCING OLD AS o NEW AS n]: cho phép dùng giá trị new hay old cho các lệnh DML, như là Insert, Update, Delete.
* [FOR EACH ROW]: cho biết cụ thể trigger sẽ thực thi trên từng dòng.
* WHEN (condition): cho biết dòng đó có điều kiện gì thì trigger mới thực thi được.

Danh sách các trigger

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên | Thao tác | Bảng | Nội dung | Ghi chú |
| 1 | AUTO\_INCREMENT\_TRIGGER | INSERT | PhieuPhat | Tăng Mã phiếu phạt tự động |  |
| 2 | update\_TinhTrangSach | INSERT OR UPDATE | Sach | Cập nhật tình trạng sách theo số lượng sách |  |
| 3 | update\_SONGAYTRATRE | INSERT OR UPDATE | PhieuTraSach | Tính số ngày trả trễ khi mượn sách quá 30 ngày |  |
| 4 | update\_SOLUONGSACH | INSERT OR UPDATE | Sach | Cập nhật lại số lượng sách |  |
| 5 | update\_SONGAYMUON | UPDATE | PhieuMuonSach | Tính số ngày mượn sách |  |
| 6 | update\_SONGAYMUON1 | INSERT OR UPDATE | PhieuTraSach | Tính số ngày mượn sách |  |
| 7 | TaoPhieuPhat | INSERT OR UPDATE | PhieuTraSach | Tự động tạo phiếu phạt khi trả trễ sách |  |
| 8 | UPDATE\_SACH | INSERT OR UPDATE OR DELETE | PhieuMuonSach | Cập nhật lại số lượng sách khi mượn sách |  |
| 9 | UPDATE\_SACH1 | INSERT OR DELETE | PhieuTraSach | Cập nhật lại số lượng sách khi trả sách |  |
|  | | | | | |

* + - 1. Mô tả một số Trigger
         1. Tự động tạo phiếu phạt

Tên: TaoPhieuPhat

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| **Bảng** | Thêm | Xoá | Sửa |
| PHIEUTRASACH | + | + | +(NgayTraSach) |

Mã PL/SQL:

create or replace TRIGGER TaoPhieuPhat AFTER INSERT OR UPDATE

ON PHIEUTRASACH FOR EACH ROW

DECLARE

SoNgTraTre INT;

SoTienPhat INT;

TEMP INT;

BEGIN

select :NEW.NGAYTRASACH - PM.NGAYMUONSACH - 30

INTO SoNgTraTre from PHIEUMUONSACH PM where :NEW.MAPHIEUMUON = PM.MAPHIEUMUON;

SoTienPhat :=1000\*SoNgTraTre;

SELECT COUNT(\*) INTO TEMP FROM PHIEUPHAT PP WHERE PP.MAPHIEUTRA = :NEW.MAPHIEUTRA;

IF(SoNgTraTre>0)THEN

IF(TEMP=0)THEN

INSERT INTO PHIEUPHAT(MADOCGIA,MAPHIEUTRA,TIENPHAT) VALUES (:NEW.MADOCGIA,:NEW.MAPHIEUTRA,SoTienPhat);

ELSE

UPDATE PHIEUPHAT SET MADOCGIA = :NEW.MADOCGIA,

MAPHIEUTRA = :NEW.MAPHIEUTRA,

TIENPHAT = SoTienPhat

WHERE MAPHIEUTRA = :NEW.MAPHIEUTRA;

END IF;

ELSE

IF(TEMP>0)THEN

DELETE FROM PHIEUPHAT WHERE MAPHIEUTRA = :NEW.MAPHIEUTRA;

END IF;

END IF;

END;

Các bước thực hiện:

* Bước 1: Lấy số ngày trả trễ của Phiếu trả sách đó
* Bước 2: Tính Số tiền phạt = 1000\*Số ngày trả trễ
* Bước 3: Kiểm tra phiếu phạt ứng với phiếu trả đó đã tồn tại hay chưa
* Bước 4: + Nếu Số ngày trả trễ của phiếu trả đó > 0 thì tiếp tục kiểm tra:
* Nếu phiếu phạt chưa tồn tại thì tạo mới phiếu phạt
* Ngược lại, cập nhật lại phiếu phạt ứng với phiếu trả đó

+ Ngược lại, tiếp tục kiểm tra:

* Nếu phiếu phạt đã tồn tại thì xoá phiếu phạt đó
  1. Tính số ngày mượn

Tên: update\_SONGAYMUON, update\_SONGAYMUON1

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| **Bảng** | Thêm | Xoá | Sửa |
| PHIEUMUONSACH | - | - | +(NgayMuonSach) |
| PHIEUTRASACH | + | - | +(MaPhieuMuon,NgayTraSach) |

Mã PL/SQL:

create or replace TRIGGER update\_SONGAYMUON AFTER UPDATE

ON PHIEUMUONSACH FOR EACH ROW

DECLARE

SoNgM INT;

TEMP INT;

BEGIN

SELECT COUNT(\*) INTO TEMP

FROM PHIEUTRASACH PT WHERE :NEW.MAPHIEUMUON = PT.MAPHIEUMUON;

IF(TEMP>0) THEN

select PT.NGAYTRASACH - :NEW.NGAYMUONSACH

INTO SoNgM FROM PHIEUTRASACH PT WHERE :NEW.MAPHIEUMUON = PT.MAPHIEUMUON;

UPDATE PHIEUTRASACH SET SONGAYMUON = SoNgM WHERE MAPHIEUMUON = :NEW.MAPHIEUMUON;

END IF;

END;

create or replace TRIGGER update\_SONGAYMUON1 BEFORE INSERT OR UPDATE

ON PHIEUTRASACH FOR EACH ROW

DECLARE

SoNgM INT;

BEGIN

select :NEW.NGAYTRASACH - PM.NGAYMUONSACH

INTO SoNgM from PHIEUMUONSACH PM where :NEW.MAPHIEUMUON = PM.MAPHIEUMUON;

:NEW.SONGAYMUON := SoNgM;

END;

Các bước thực hiện:

Đối với Trigger update\_SONGAYMUON:

* Bước 1: Kiểm tra phiếu trả ứng với phiếu mượn đó đã tồn tại hay chưa
* Bước 2: Nếu đã tồn tại thì tính số ngày mượn. Sau đó, cập nhật lại số ngày mượn cho phiếu trả sách

Đối với Trigger update\_SONGAYMUON1:

* Bước 1: Tính số ngày mượn
* Bước 2: Cập nhật lại số ngày mượn cho phiếu trả sách

1. **Stored Procedure**
   * + 1. Stored Procedure trong Oracle

Stored procedure là một hoặc một tập các lệnh SQL được biên dịch để thực hiện cùng lúc mỗi khi được gọi đến. Về bản chất giống với hàm, thủ tục trong các ngôn ngữ lập trình khác.

Stored procedure giúp giảm thời gian giao tiếp giữa các ứng dụng với hệ quản trị, bởi vì thay vì gửi nhiều câu lệnh dài thì ta chỉ cần gọi tới một thủ tục và trong thủ tục này sẽ thực hiện nhiều câu lệnh SQL.

* Cú pháp:

CREATE [OR REPLACE] PROCEDURE procedure\_name

[(parameter\_name [IN | OUT | IN OUT] type [, ...])]

{IS | AS}

BEGIN

< procedure\_body >

END procedure\_name;

* Giải thích:
* procedure\_name : tên của procedure
* [OR REPLACE]: thay thế những procedure đang tồn tại.
* IN: cho phép nhập tham số. OUT: giá trị trả về cụ thể của procedure.
* < procedure\_body >: chứa những câu lệnh của procedure.
* AS được sử dụng thay vì IS để tạo ra những procedure độc lập.
  + - 1. Transaction trong Oracle
  1. Định nghĩa

Transaction là một đơn vị làm việc có tính chất logic. Chứa một hoặc nhiều câu lệnh SQL. Ảnh hưởng của tất cả các câu lệnh SQL trong một transaction có thể là commit tất cả (áp dụng cho cơ sở dữ liệu) hoặc rollback tất cả(hoàn tác từ cơ sở dữ liệu).

* 1. Cấu trúc Transaction

Gồm 1 hay nhiều câu truy vấn, có thể bao gồm DDL,DML.

Có bắt đầu và kết thúc.

*Bắt đầu transaction:*

* + - * Transaction bắt đầu khi câu lệnh SQL đầu tiên được thực thi, bao gồm DDL,DML hoặc lệnh SET TRANSACTION.
      * TRANSACTION NAME: Đặt tên cho transaction, bắt đầu transasction.
      * SET TRANSACTION NAME <Tên transaction> (trước lệnh này nên có 1 lệnh COMMIT).

SET TRANSACTION NAME 'ten\_transaction';

* + - * Khi 1 transaction mới bắt đầu , hệ quản trị oracle sẽ gán nó vào **undo data segment** ( ghi nhận lại các thao tác của transaction trước khi commit, để có thể rollback khi có lỗi).

*Kết thúc transaction:*

* + - * Transaction có thể kết thúc trong nhiều trường hợp:
        + Gặp lệnh COMMIT hoặc ROLLBACK mà không có savepoint.
        + Gặp các câu lệnh DDL như create, drop, rename, alter.
        + User ngắt kết nối đến hệ quản trị đột ngột, transaction sẽ tự động commit.
        + Các ứng dụng đang kết nối đến hệ quản trị bị dừng đột ngột, transaction sẽ tự động rollback.
* Một transaction đang thực thi là 1 transaction đã bắt đầu nhưng chưa được COMMIT hoặc ROLLBACK.
  1. Transaction control

Gồm các lệnh để quản lý sự thay đổi của DML lên database, gồm 1 số lệnh chính:

* SAVEPOINT: Xác định 1 điểm trong transaction để rollback về khi có sự cố.
* COMMIT: Kết thúc transaction, **lưu thay đổi vĩnh viễn**, xóa tất cả SAVEPOINT, mở transaction locks.
* ROLLBACK: phục hồi lại dữ liệu trước khi thay đổi.

ROLLBACK:

* Hoàn tác mọi thay đổi.
* Mở tất cả khóa.
* Xóa toàn bộ savepoints.
* Kết thúc transaction.
  + - 1. Danh sách các Stored Procedure:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên Procedure | Tham số đầu vào | Tham số đầu ra | Nội dung | Ghi chú |
| 1 | INSERT\_PHIEUMUON | ID\_PhieuMuon, ID\_DocGia, ID\_Sach, NgMuonSach, TinhTrang |  | Thêm phiếu mượn sách: Số lượng sách = 0 thì không được mượn |  |
| 2 | INSERT\_PHIEUTRA | ID\_PhieuTra, ID\_PhieuMuon, ID\_DocGia, ID\_Sach, NgTraSach |  | Thêm phiếu trả sách: phiếu trả chỉ trả cho 1 phiếu mượn |  |
| 3 | DOI\_MAT\_KHAU | ID\_DocGia, MK\_DocGia |  | Đổi mật khẩu cho tài khoản đăng nhập |  |
| 4 | INSERT\_DOCGIA | ID\_DocGia, HOTEN\_DG, GIOITINH\_DG, NGSINH\_DG, SDT\_DG, DIACHI\_DG, EMAIL\_DG, VAITRO\_DG |  | Thêm độc giả mới |  |
| 5 | DELETE\_DOCGIA | ID\_DocGia |  | Xoá độc giả |  |
| 6 | UPDATE\_DOCGIA | ID\_DocGia, HOTEN\_DG, GIOITINH\_DG, NGSINH\_DG, SDT\_DG, DIACHI\_DG, EMAIL\_DG, VAITRO\_DG |  | Cập nhật lại thông tin độc giả |  |

* + - 1. Mô tả một số Stored Procedure
         1. Thêm phiếu mượn sách
* Tên: INSERT\_PHIEUMUON
* Nội dung: Thêm phiếu mượn sách khi số lượng sách = 0 thì không được mượn
* Tham số đầu vào: ID\_PhieuMuon, ID\_DocGia, ID\_Sach, NgMuonSach, TinhTrang
* Tham số đầu ra: không có

Mã PL/SQL:

CREATE OR REPLACE PROCEDURE INSERT\_PHIEUMUON

(

ID\_PhieuMuon IN PHIEUMUONSACH.MaPhieuMuon%TYPE,

ID\_DocGia IN PHIEUMUONSACH.MaDocGia%TYPE,

ID\_Sach IN PHIEUMUONSACH.MaSach%TYPE,

NgMuonSach IN PHIEUMUONSACH.NgayMuonSach%TYPE,

TinhTrang IN PHIEUMUONSACH.TinhTrangPM%TYPE

)

AS

TEMP INT;

BEGIN

SELECT SOLUONG INTO TEMP

FROM SACH WHERE MASACH = ID\_Sach;

IF(TEMP>0)THEN

INSERT INTO PHIEUMUONSACH (MaPhieuMuon, MaDocGia, MaSach, NgayMuonSach, TinhTrangPM)

VALUES (ID\_PhieuMuon,ID\_DocGia,ID\_Sach,NgMuonSach,TinhTrang);

END IF;

END;

Các bước thực hiện:

* Bước 1: Lấy thông tin số lượng sách ở trong thư việnmà độc giả cần mượn
* Bước 2: Kiểm tra số lương sách:

+ Nếu số lượng >0 thì cho phép thêm phiếu mượn

+ Và ngược lại thì không được thêm phiếu mượn

1. Thêm phiếu trả sách

* Tên: INSERT\_PHIEUTRA
* Nội dung: Thêm phiếu trả sách: phiếu trả chỉ trả cho 1 phiếu mượn
* Tham số đầu vào: ID\_PhieuTra, ID\_PhieuMuon, ID\_DocGia, ID\_Sach, NgTraSach
* Tham số đầu ra: không có

Mã PL/SQL:

CREATE OR REPLACE PROCEDURE INSERT\_PHIEUTRA

(

ID\_PhieuTra IN PHIEUTRASACH.MaPhieuTra%TYPE,

ID\_PhieuMuon IN PHIEUTRASACH.MaPhieuMuon%TYPE,

ID\_DocGia IN PHIEUTRASACH.MaDocGia%TYPE,

ID\_Sach IN PHIEUTRASACH.MaSach%TYPE,

NgTraSach IN PHIEUTRASACH.NgayTraSach%TYPE

)

AS

TEMP INT;

BEGIN

SELECT COUNT(\*) INTO TEMP

FROM PHIEUTRASACH WHERE MAPHIEUMUON = ID\_PhieuMuon;

IF(TEMP=0)THEN

INSERT INTO PHIEUTRASACH (MaPhieuTra, MaPhieuMuon, MaDocGia, MaSach, NgayTraSach)

VALUES (ID\_PhieuTra,ID\_PhieuMuon,ID\_DocGia,ID\_Sach,NgTraSach);

END IF;

END;

Các bước thực hiện:

* Bước 1: Lấy số lượng phiếu trả tương ứng với phiếu mượn mà phiếu trả cần thêm
* Bước 2: Kiểm tra phiếu trả đã tồn tại hay chưa:

+ Nếu chưa tồn tại thì Nếu số lượng >0 thì cho phép thêm phiếu trả

+ Và ngược lại thì không được thêm phiếu trả

# CHƯƠNG 5: XỬ LÝ TRUY XUẤT ĐỒNG THỜI

1. **Các mức cô lập trong Oracle:**
   * + 1. **Read Committed:**
          1. Trong Read Committed, mọi truy vấn được thực hiện bởi một transaction chỉ nhìn thấy dữ liệu đã được commit. Mức cô lập này là mặc định. Nó phù hợp với môi trường cơ sở dữ liệu trong đó ít transaction có khả năng xung đột.
          2. Xung đột ghi trong Read Committed:

Trong Read Committed, một xung đột ghi xảy ra khi transaction cố gắng thay đổi một dòng đã được cập nhật bởi một transaction khác mà chưa được commit hay rollback.

Transaction ngăn chặn sửa đổi dòng được gọi là blocking transaction. Transaction Read Committed chờ blocking transaction kết thúc và giải phóng khóa rồi transaction Read Committed mới lấy được khóa và thực hiện thay đổi.

* + - 1. **Serializable:**
         1. Ở mức cô lập tuần tự(serializable isolationlevel), transaction chỉ thấy những thay đổi đã được commit trước thời điểm transaction có mức cô lập tuần tự này bắt đầu và các thay đổi được thực hiện bởi chính transaction này.
         2. Một transaction tuần tự hoạt động trong một môi trường chỉ một mình nó như thể không có người dùng nào khác đang sửa đổi dữ liệu trong cơ sở dữ liệu. mức cô lập tuần tự phù hợp với các môi trường:

Với cơ sở dữ liệu lớn và các transaction ngắn chỉ cập nhật một vài dòng.

Trường hợp có 2 transaction đồng thời sửa đổi cùng 1 dòng là tương đối thấp.

Trường hợp các transaction tương đối dài chủyếu chỉ đọc.

* + - * 1. mức cô lập tuần tự, Bất kỳ dòng nào được đọc bởi transaction tuần tự được đảm bảo giống nhau khi đọc lại. Bất kỳ truy vấn nào cũng được đảm bảo trả về cùng kết quả trong suất thời gian transaction đó hoạt động, do đó, những thay đổi được thực hiện bởi transaction khác sẽ không hiển thị đối với truy vấn của transaction tuần tự bất kể những thay đổi của transaction khác đã chạy được bao lâu. Transaction tuần tự(serializable) ngăn ngừa được Dirty reads, Nonrepeatable reads, Phantom reads.
        2. Trong cơ sở dữ liệu Oracle, nếu transaction Tuần tự thực hiện thay đổi 1 dòng nào đó mà dòng này đã được thay đổi bởi một transaction khác đã commit sau khi transaction tuần bắt đầu thực thi thì sẽ xuất hiện 1 lỗi đó là:

ORA-08177: Cannot serialize access for this transaction

* + - 1. **Read-only:**

Mức cô lập Read-only tương tự như mức cô lập Serializable, nhưng các transaction có mức cô lập Read-only chỉ được đọc dữ liệu, không được sửa đổi dữ liệu trong transaction (trừtrường hợp là người dùng là SYS).

1. **Cơ chế khóa:**

“Khóa” là một cơ chếgiúp ngăn chặn các tương tác “có hại” đến hệ thống.Vậy, thế nào gọi là tương tác “có hại” ?

Tương tác gọi là “có hại” khi nó cập nhật dữ liệu sai hoặc thay đổi sai cấu trúc dữ liệu, khiến dữ liệu không nhất quán giữa, thường xảy ra khi nhiều transaction cùng thao tác lên 1 đơn vị dữ liệu. Vì vậy, cơ chế khóa đóng vai trò cực kì quan trọng để đảm bảo cơ sở dữ liệu đồng thời và nhất quán.

Điều gì xảy ra bên trong cơ chế khóa ?

Hệ quản trị có thể thực thi một vài cơ chế khóa, tùy vào các thao tác khác nhau lên dữ liệu.Nhìn chung, hệ quản trị sử dụng 2 loại chính: exclusive locks(khóa độc quyền) và share locks(khóa chia sẻ). Tại một thời điểm, chỉ có 1 excusive lock được thực thi trên 1 đơn vị dữ liệu, nhưng nhiều share locks có thể cùng thực thi trên 1 đơn vị dữ liệu.

Cơ chế khóa tác động đến sự tương tác giữa việc đọc và ghi dữ liệu:

Một dòng dữ liệu chỉ bị khóa khi nó đang được thay đổi bởi việc hành động ghi. Khi có 1 câu truy vấn cập nhật dữ liệu của 1 dòng, hệ quản trị sẽ khóa dòng đó lại.

Một hành động ghi lên 1 dòng dữ liệu sẽ ngăn chặn các hành động khác cùng đồng thời ghi lên dòng dữ liệu đó.

Hành động đọc dữ liệu không ngăn cản hành động ghi: Vì hành động đọc không khóa đơn vị dữ liệu nó đang đọc, nên hành động ghi có thể thao tác lên đơn vị dữ liệu này. Chỉ có một trường hợp ngoại lệ là câu lệnh **Select... for update** sẽ lock đơn vị dữ liệu nó đang đọc.

Hành động ghi dữ liệu không ngăn cản hành động đọc: Khi một đơn vị dữ liệu đang được thao tác bởi hành động ghi, hệ quản trị sẽ sử dụng phiên bản trước đó của đơn vị dữ liệu để trả về cho hành động đọc.

Cơ chế khóa làm được gì?

Cơ chế khóa phải thỏa các yêu cầu quan trọng của cơ sở dữ liệu: tính nhất quán và tính toàn vẹn

Hệ quản trị Oracle đảm bảo dữ liệu đồng thời, nhất quán và toàn vẹn giữa các transactions thông qua cơ chế khóa. Cơ chế khóa diễn ra tự động và không cần sự can thiệp của người dùng.

Các chế độ khóa:

* Hệ quản trị Oracle tự động sử dụng chế độ khóa ở cấp cần thiết thấp nhất để đảm bảo khả năng truy cập đồng thời của nhiều tiến trình lên cùng đơn vị dữ liệu.
* Sử dụng chế độ khóa cấp càng thấp, dữ liệu có thể được truy cập bởi càng nhiều user. Ngược lại, nếu sử dụng mức khóa càng cao, dữ liệu càng hạn chế người truy cập.
* Hệ quản trị Oracle sử dụng 2 chế độ khóa chính cho cơ sở dữ liệu nhiều người dùng :
* Exclusive lock mode: Transaction nhận được exclusive lock khi nó có nhu cầu thay đổi dữ liệu. Các transaction khác sẽ không thể thao tác lên đơn vị dữ liệu này cho đến khi transaction đang giữ exclusive lock nhả khóa.
* Share lock mode: Chế độ này cho phép chia sẻ cùng 1 đơn vị dữ liệu. Nhiều giao tác đọc có thể chia sẻ dữ liệu cho nhau, mỗi giao tác giữ một share lock để ngăn hành động ghi vào cùng thời điểm của các giao tác ghi. Nhiều giao tác có thể được cấp share lock trên cùng đơn vị dữ liệu.

1. **Deadlock**
   * + 1. Deadlock là tình huống khi một hay nhiều người dùng đang chờmột đơn vị dữ liệu mà đang bị các người dùng khác cũng đang chờ đơn vị dữ liệu khác block.
       2. Hệ quản trị Oracle tự động phát hiện và xử lý deadlock bẳng cách rollback một transaction gây deadlock, giải phóng 1 dòng đang bị khóa. Cơ sở dữ liệu sẽ trả về thông báo cho transaction bị rollback

## **IV. Mô tả đồ án môn học**

1. **Lost update**
2. Mô tả tình huống

Thủ thư thứ nhất cập nhật số lượng cuốn sách “Dữ liệu lớn” vào trong thư viện. Ngay sau đó 1 thủ thư khác thấy có một vài cuốn sách bị mất nên đã cập nhật lại số lượng sách cho cuốn sách đó. Như vậy thao tác cập nhật của thủ thư thứ nhất khống có tác dụng.

1. Bảng mô tả Lost Update

|  |  |  |
| --- | --- | --- |
| Session 1 | Session 2 | Giải thích |
| UPDATE SACH  SET SoLuong = 20;  WHERE TenSach = ‘Dữ liệu lớn’;  ---------------------------------------  MaSach TenSach SoLuong  --------- ---------- -----------  S1011 Dữ liệu lớn 20 | No action. | Session 1Cập nhật số lượng cuốn sách “Dữ liệu lớn” vào hệ thống |
| No action. | UPDATE SACH  SET SoLuong = 15;  WHERE TenSach = ‘Dữ liệu lớn’; | Session 2 thực hiện lệnh cập nhật số lượng sách cho sách “Dữ liệu lớn” |
| No action. | Commit; | Session 2 thực hiện xong và kết thúc |
| SELECT MaSach, TenSach, SoLuong  FROM SACH  WHERE TenSach = ‘Dữ liệu lớn’;  ---------------------------------------  MaSach TenSach SoLuong  --------- ---------- -----------  S1011 Dữ liệu lớn 15 | No action. | Session 1 kiểm tra số lượng sách “Dữ liệu lớn” có trong thư viện thì thì thấy dữ liệu cập nhật bị thay đổi |

1. Nguyên nhân và giải pháp

* Nguyên nhân: Lệnh cập nhật của session 1 bị mất vì đã bị session 2 ghi chồng lên ở thời điểm sau đó
* Giải pháp: Sử dụng câu lệnh “LOCK TABLE SACH IN SHARE MODE” trước khi cập nhật dữ liệu. Câu lệnh này chỉ cho các giao tác khác chỉ được đọc mà không được ghi lên dữ liệu đó

1. **Unrepeatable read**
2. Mô tả tình huống

Trong khi thủ thư thứ nhất đang thống kê các cuốn sách đang còn trong thư viện, thì thủ thư thứ 2 lại thêm 1 cuốn sách mới vào hệ thống , lúc này thủ thư thứ nhất lập báo cáo, thống kê các cuốn sách đang còn trong thư viện thì phát hiện dữ liệu bị sai sót

1. Bảng mô tả Unrepeatable read

|  |  |  |
| --- | --- | --- |
| Session 1 | Session 2 | Giải thích |
| SELECT MaSach, TenSach, SoLuong  FROM SACH  WHERE SoLuong > 0;  ---------------------------------------  MaSach TenSach SoLuong  --------- ---------- -----------  S1009 Trí tuệ nhân tạo 13  S1011 Dữ liệu lớn 20  S1023 Hai số phận 9 | No action. | Session 1truy vấn những cuốn sách đang còn trong thư viện |
| No action. | INSERT INTO SACH (MaSach, TenSach, TheLoai, SoTrang, TacGia, NXB, ViTri, SoLuong)  VALUES ('S1034','Nhà giả kim','cuộc đời',123,'Paul kim','NXB Văn học','KF2635',27); | Session 2 thực hiện lệnh thêm mới cuốn sách |
| No action. | Commit; | Session 2 thực hiện xong và kết thúc |
| SELECT MaSach, TenSach, SoLuong  FROM SACH  WHERE SoLuong > 0;  ---------------------------------------  MaSach TenSach SoLuong  --------- ---------- -----------  S1009 Trí tuệ nhân tạo 13  S1011 Dữ liệu lớn 20  S1023 Hai số phận 9  S1034 Nhà giả kim 27 | No action. | Session 1Truy vấn những cuốn sách đang còn trong thư viện thì thấy dữ liệu bị thay đổi |

1. Nguyên nhân và giải pháp

* Nguyên nhân: Session 1 đọc dòng dữ liệu mới do session 2 thay đổi dữ liệu trước đó.
* Giải pháp: Sử dụng câu lệnh “set transaction isolation level serializable” trước khi đọc dữ liệu

1. **Phantom**
2. Mô tả tình huống

Trong khi độc giả đang tìm cuốn sách “Nhà giả kim”

thấy vẫn đang còn trong thư viện, thì thủ thư lại xoá đi cuốn sách đó. Lúc này độc giả đó tìm lại cuốn sách “Nhà giả kim” thì thấy không còn nữa.

1. Bảng mô tả Phantom

|  |  |  |
| --- | --- | --- |
| Session 1 | Session 2 | Giải thích |
| SELECT MaSach, TenSach, SoLuong  FROM SACH  WHERE TenSach = ‘Nhà giả kim’;  ---------------------------------------  MaSach TenSach SoLuong  --------- ---------- -----------  S1009 Nhà giả kim 27 | No action. | Session 1tìm cuốn sách có tên “Nhà giả kim” trong thư viện |
| No action. | DELETE FROM SACH  WHERE TenSach = ‘Nhà giả kim’; | Session 2 thực hiện lệnh xoá cuốn sách “Nhà giả kim” |
| No action. | Commit; | Session 2 thực hiện xong và kết thúc |
| SELECT MaSach, TenSach, SoLuong  FROM SACH  WHERE TenSach = ‘Nhà giả kim’;  ---------------------------------------  MaSach TenSach SoLuong  --------- ---------- ----------- | No action. | Session 1tìm cuốn sách có tên “Nhà giả kim” trong thư viện thì thấy không còn trong thư viện nữa |

1. Nguyên nhân và giải pháp

* Nguyên nhân: Do dữ liệu đã bị session 2 xoá trước đó nên session 1 không tim thấy nữa.
* Giải pháp: Sử dụng câu lệnh “conn.setTransactionIsolation(Connection.TRANSACTION\_SER IALIZABLE” cho kết nối dữ liệu.

1. **Dirty read**
2. Mô tả tình huống

Thủ thư thêm phiếu mượn cho cuốn sách “Kể chuyện Bác Hồ” và cuốn này chỉ còn 1 cuốn. Ngay lúc đó có độc giả vào tìm cuốn sách “Kể chuyện Bác Hồ” thì thấy đã hết. Và sau đó, độc giả ban đầu định mượn lại không mượn nữa nên thủ thư huỷ phiêu mượn đó. Vậy sau đó sách đó vẫn đang còn trong thư viện.

1. Bảng mô tả Dirty read

|  |  |  |
| --- | --- | --- |
| Session 1 | Session 2 | Giải thích |
| SELECT MaSach, TenSach, SoLuong  FROM SACH  WHERE TenSach = ‘Kể chuyện Bác Hồ’;  --------------------------------------  MaSach TenSach SoLuong  --------- ----------- ----------  S1905 Kể chuyện Bác Hồ 1 | No action. | Session 1 tra cứu số lượng của cuốn sách “Kể chuyện Bác Hồ” trong thư viện |
| INSERT INTO PHIEUMUONSACH (MaPhieuMuon, MaDocGia, MaSach, NgayMuonSach)  VALUES ('PM1077','DG1009','S1905',TO\_DATE('02/07/2021','DD/MM/YYYY')); | No action. | Session 1 thêm phiếu mượn cho cuốn sách “Kể chuyện Bác Hồ” |
| No action. | SELECT MaSach, TenSach, SoLuong  FROM SACH  WHERE TenSach = ‘Kể chuyện Bác Hồ’;  ------------------------------------  MaSach TenSach SoLuong  --------- ---------- ----------  S1905 Kể chuyện Bác Hồ 0 | Session 2 tìm cuốn sách “Kể chuyện Bác Hồ” thì thấy đã hết |
|  | Commit; | Session 2 thực hiện xong và kết thúc |
| **Abort** |  | Session 1 thực hiện huỷ giao tác |

1. Nguyên nhân và giải pháp

* Nguyên nhân: Session đã đọc dữ liệu được ghi bởi sesion 1 trước đó nhưng sau đó session 1 huỷ việc ghi
* Giải pháp: thiết lập mức cô lập “Read Committed”

# CHƯƠNG 6: THIẾT KẾ GIAO DIỆN

1. **Danh sách các màn hình**
   1. Đăng nhập

|  |  |  |
| --- | --- | --- |
| STT | Màn hình | Chức năng |
| 1. | Đăng nhập | Đăng nhập |

* 1. Đổi mật khấu

|  |  |  |
| --- | --- | --- |
| STT | Màn hình | Chức năng |
| 1. | Đổi mật khấu | Đổi mật khấu |

* 1. Độc giả

|  |  |  |
| --- | --- | --- |
| STT | Màn hình | Chức năng |
| 1. | Trang chủ | Độc giả có thế đổi mật khẩu hoặc đăng xuất |
| 2. | Thông tin sách | Hiển thị danh sách các sách có trong thư viện và độc giả có thế tìm kiếm nhanh sách |

* 1. Thủ thư

|  |  |  |
| --- | --- | --- |
| STT | Màn hình | Chức năng |
| 1. | Trang chủ | Thủ thư có thế đổi mật khẩu hoặc đăng xuất |
| 2. | Quản lý độc giả | Hiển thị những thông tin đã có trong hệ thống |
| 3. | Quản lý sách | Hiển thị những thông tin đã có trong hệ thống |
| 4. | Quản lý mượn sách | Hiển thị những thông tin đã có trong hệ thống |
| 5. | Quản lý trả sách | Hiển thị những thông tin đã có trong hệ thống |
| 6. | Quản lý phiếu phạt | Hiển thị những thông tin đã có trong hệ thống |
| 7. | Báo cáo, thống kê | Hiển thị những thông tin đã có trong hệ thống |

1. **Mô tả các màn hình**
   1. Đăng nhập

Graphical user interface, application

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1. | Khung hình | Label | Hiển thị |
| 2. | Tên tài khoản | Text | Nhập tên tài khoản |
| 3. | Mật khẩu | Text | Nhập mật khẩu |
| 4. | Đăng nhập | Button | Thực hiện đăng nhập vào hệ thống |
| 5. | Đặt lại | Button | Thực hiện xóa tên tai khoản và mật khẩu khi người dùng nhập sai |

* 1. Đổi mật khẩu

Graphical user interface

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1. | Khung hình | Label | Hiển thị |
| 2. | Tên tài khoản | Text | Nhập tên tài khoản |
| 3. | Mật khẩu cũ | Text | Nhập mật khẩu cũ |
| 4. | Mật khẩu mới | Text | Nhập mật khẩu mới |
| 5. | Xác nhận mật khẩu mới | Text | Nhập lại mật khẩu mới |
| 6. | Xác nhận | Button | Thực hiện xác nhận đổi mật khẩu |
| 6. | Hủy | Button | Thực hiện hủy hành động đổi mật khẩu |

* 1. Độc giả

1. Trang chủ

Chart

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1. | Trang chủ | Button | Vào trang chủ |
| 2. | Thông tin sách | Button | Vào thong tin sách |
| 3. | Khung hình | Label | Hiển thị |
| 4. | Đổi mật khẩu | Button | Thực hiện truy cập vào trang đổi mật khẩu |
| 5. | Đăng xuất | Button | Thực hiện đăng xuất |

1. Thông tin sách

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1. | Khung hình | Label | Hiển thị |
| 2. | Từ khóa tìm kiếm | Text | Nhập tên sách muốn tìm |
| 3. | Tìm kiếm | Button | Thực hiện tìm kiếm |
| 4. | Đặt lại | Button | Xóa từ khóa mà người dùng đã nhập |
| 5. | Khung hình | Label | Hiển thị danh sách sách |

Graphical user interface, application

Description automatically generated

* 1. Thủ thư

1. Trang chủ

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1. | Khung hình | Label | Hiển thị |
| 2. | Đổi mật khẩu | Button | Truy cập vào trang đổi mật khẩu |
| 3. | Đăng xuất | Button | Đăng xuất khỏi hệ thống |
| 4. | Khung hình | Label | Hiển thị |

Chart

Description automatically generated

1. Quản lý độc giả

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1. | Khung hình | Label | Hiển thị |
| 2. | Từ khóa tìm kiếm | Text | Nhập thong tin độc giả |
| 3. | Tìm kiếm | Button | Thực hiện tìm kiếm |
| 4. | Đặt lại | Button | Xóa từ khóa đã nhập |
| 5. | Them | Button | Truy cập vào form thêm độc giả |
| 6. | Xóa | Button | xóa đôc giả |
| 7. | Sửa | Button | Truy cập vào form sửa độc giả |
| 8. | Làm mới | Button | Làm mới danh sách độc giả sau khi thêm xóa sửa |
| 9. | Khung hình | Label | Hiển thị |

Graphical user interface, table

Description automatically generated

1. Form thêm,sửa độc giả

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1. | Khung hình | Label | Hiển thị |
| 2. | Chỉnh sửa | Button | Truy cập vào form sửa |
| 3. | Mã độc giả | Text | Nhập mã độc giả |
| 4. | Họ tên | Text | Nhập họ tên độc giả |
| 5. | Giới tính | Combobox | Chọn giới tính |
| 6. | Ngày sinh | Text | Nhập ngày sinh |
| 7. | Sdt | Text | Nhập SDT |
| 8. | Địa chỉ | Text | Nhập địa chỉ |
| 9. | Email | Text | Nhập email |
| 10. | Vai trò | Combobox | Chọn vai trò |
| 11. | Xác nhận | Button | Xác nhận thêm độc giả |
| 12 | Hủy | Button | Hủy hành động thêm độc giả |

Graphical user interface

Description automatically generated

1. Quản lý sách

Graphical user interface

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1. | Từ khóa tìm kiếm | Text | Nhập từ khóa cần tìm |
| 2. | Tìm kiếm | Button | Thực hiện tìm kiếm |
| 3. | Đăt lại | Button | Xóa từ khóa đã nhập |
| 4. | Thêm | Button | Thêm truy cập vào form thêm sách |
| 5. | Xóa | Button | Xóa sach |
| 6. | Sửa | Button | Thêm truy cập vào form thêm sách |
| 7. | Làm mới | Button | Làm mới danh sách sách sau khi thêm xóa sửa |
| 8. | Khung hình | Label | Hiện thị |

1. Form thêm, sửa sách

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1. | Chỉnh sửa | Button | Sửa thong tin sách |
| 2. | Mã sách | Text | Nhập mã sách |
| 3. | Tên sách | Text | Nhập tên sách |
| 4. | Thể loại | Text | Nhập thể loại sách |
| 5. | Số trang | Text | Nhập số trang sách |
| 6. | Tác giả | Text | Nhập tác giả sách |
| 7. | NXB | Text | Nhập NXB |
| 8. | Vị trí | Text | Nhập vị trí sách |
| 9. | Số lưọng | Text | Nhập số lượng sách |
| 10 | Xác nhận | Button | Xác nhận thêm sách |
| 11 | Hủy | Button | Hủy hành động thêm sách |

Graphical user interface, chart, bar chart

Description automatically generated

1. Quản lý mượn sách

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1. | Từ khóa tìm kiếm | Text | Nhập từ khóa cần tìm |
| 2. | Tìm kiếm | Button | Thực hiện tìm kiếm |
| 3. | Đăt lại | Button | Xóa từ khóa đã nhập |
| 4. | Thêm | Button | Thêm truy cập vào form thêm phiếu mượn sách |
| 5. | Xóa | Button | Xóa phiếu mượn sách |
| 6. | Sửa | Button | Thêm truy cập vào form thêm phiếu mượn sách |
| 7. | Làm mới | Button | Làm mới danh sách phiếu sau khi thêm xóa sửa |
| 8. | Khung hình | Label | Hiện thị |

Table

Description automatically generated

1. Form thêm,sửa phiếu mượn sách

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Chỉnh sửa | Button | Sửa thong tin phiếu mượn |
| 2 | Mã phiếu mượn | Text | Nhập mã phiếu mượn |
| 3 | Mã độc giả | Text | Nhập mã độc giả |
| 4 | Mã sách | Text | Nhập mã sách |
| 5 | Ngày mượn sách | Text | Nhập ngày mượn sách |
| 6 | Tình trạng phiếu mượn | Combobox | Chọn tình trạng của phiếu |
| 7. | Xác nhận | Button | Xác nhận thêm phiếu mượn |
| 8 | Hủy | Button | Hủy hành độc thêm phiếu mượn |

Graphical user interface

Description automatically generated

1. Quản lý phiểu trả sách

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1. | Từ khóa tìm kiếm | Text | Nhập từ khóa cần tìm |
| 2. | Tìm kiếm | Button | Thực hiện tìm kiếm |
| 3. | Đăt lại | Button | Xóa từ khóa đã nhập |
| 4. | Thêm | Button | Thêm truy cập vào form thêm phiếu trả sách |
| 5. | Xóa | Button | Xóa phiếu trả sách |
| 6. | Sửa | Button | Thêm truy cập vào form thêm phiếu trả sách |
| 7. | Làm mới | Button | Làm mới danh sách phiếu sau khi thêm xóa sửa |
| 8. | Khung hình | Label | Hiện thị |

Graphical user interface, table

Description automatically generated

1. Form phiếu trả sách

Chart, bar chart

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Chỉnh sửa | Button | Sửa phiếu trả sách |
| 2 | Mã phiếu trả | Text | Nhập mã phiếu trả |
| 3 | Mã phiếu mượn | Text | Nhập mã phiếu mượn |
| 4 | Mã độc giả | Text | Nhập mã độc giả |
| 5 | Mã sách | Text | Nhập mã sách |
| 6. | Ngày trả sách | Text | Nhập ngày trả sách |
| 6. | Xác nhận | Button | Xác nhận thêm phiếu trả sách |
| 7. | Hủy | Button | Hùy hành động thêm phiếu trả sách |

1. Quản lý phiếu phạt

Table

Description automatically generated with medium confidence

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Mã phiếu phạt | Text | Nhập mã phiếu phạt |
| 2 | Tìm kiếm | Button | Thực hiện tìm kiếm |
| 3 | Đặt lại | Button | Xóa từ khóa đã |
| 4 | Tạo hóa đơn | Button | Tạo hóa đơn |
| 5 | Xóa | Button | Xóa phiếu phạt |
| 6 | Làm mới | Button | Làm mới danh sách phiếu sau khi thêm xóa sửa |
| 7 | Khung hình | Label | Hiển thị |

1. Hóa đơn phiếu phạt

Graphical user interface, application

Description automatically generated

1. Báo cáo, thống kê

Graphical user interface, application

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Thống kê sách đã mượn hết | Button | Truy cập vào form thống kê sách đã mượn hết |
| 2 | Thống kệ sách đang có | Button | Truy cập vào form thống kệ sách đang có |
| 3 | Thống kê số lượt mượn sách theo các năm | Button | Truy cập vào form thống kê số lượt mượn sách theo các năm |
| 4 | Mã sách | Text | Nhập mã sách |
| 5 | Năm | Text | Nhập năm |
| 6 | Thống kệ số lượt mượn theo từng tháng trong năm | Button | Truy cập vào form thống kệ số lượt mượn theo từng tháng trong năm với năm tương ứng được nhập |
| 7 | Thống kê top 5 sách được mượn nhiếu nhất theo năm | Button | Truy cập vào form thống kê top 5 sách được mượn nhiếu nhất theo năm |

1. Report Thống Kê Sách Đã Đang Còn Trong Thư Viện

Graphical user interface, table

Description automatically generated

1. Report Thống Kê Sách Đã Mượn hết

Table

Description automatically generated

1. Report Thống Kê Top 5 Sách Được Mượn Nhiều Nhất Năm

Table

Description automatically generated

1. Report Thống Kê Số Lượt Mượn Sách Theo Mã Sách Vào Các Năm

Graphical user interface, text, application

Description automatically generated

1. Report Thống Kê Số Lượt Mượn Sách Theo Tháng Trong Năm

Table

Description automatically generated

# CHƯƠNG 7: KẾT LUẬN

1. **Kết quả đạt được**

Trong suốt quá trình làm đồ án vừa qua, nhóm đã đạt được những điều sau:

* Biết cách thiết kế hoàn thiện một dự án nhỏ
* Nắm được kĩ năng phân chia công việc, sắp xếp thời gian, làm việc nhóm
* Hiểu và vận dụng được kiến thức của môn hệ quản trị cơ sở dữ liệu, cụ thể là làm việc trên Oracle, như:
* Hiểu được cách hoạt động của Oracle
* Các cú pháp và cách suy nghĩ của ngôn ngữ PL/SQL
* Cách thực hiện một Trigger, Function, Cursor, Stored Procedure
* Trang bị thêm kiến thức về khóa và các mức cô lập.
* Giải quyết được các trường hợp truy xuất đồng thời.

1. **Hạn chế**

Do kỹ năng của chúng em còn yếu nên đang còn gặp một số hạn chế như sau:

* Giao diện không được chăm chút.
* Vẫn còn một số vấn đề trong dữ liệu.
* Stored Procedure vẫn còn khá đơn giản.
* Các tình huống truy xuất đồng thời có thể không hợp lí.
* Một số cái chưa bám sát với thực tế.
* Không đáp ứng đúng như kỳ vọng ban đầu.

1. **Hướng phát triển**

Để khắc phục các hạn chế còn tồn tại cũng như phát triển phần mềm nhóm có một số đề xuất như sau:

* Phát triển được giao diện bắt mắt hơn.
* Phát triển một ứng dụng được nhiều người sử dụng.
* Chỉnh sửa úng dụng chạy nhanh hơn.
* Nên tìm hiểu sâu vào chuyên môn của đề tài hơn

# PHỤ LỤC 1: BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Công việc | Quốc Lưu | Hữu Thắng | Kim Hằng | Hải Linh |
| Phân tích bài toán, xác định chức năng | X | X | X | X |
| Thiết kế CSDL | X | X | X | X |
| Tạo bảng, tạo các ràng buộc | X |  |  | X |
| Xây dựng các Trigger, Procedure | X | X | X | X |
| Xử lí các vấn đề liên quan đến truy xuất đồng thời | X | X | X | X |
| Thiết kế giao diện |  | X | X |  |
| Lập trình | X |  |  | X |
| Test chức năng | X | X | X | X |
| Soạn và chỉnh sửa báo cáo | X | X | X | X |
| Phân công công việc | X |  |  |  |
|  | |  | | |
|

# PHỤ LỤC 2: TÀI LIỆU THAM KHẢO

1. Các slide môn Hệ quản trị cơ sở dữ liệu của khoa Hệ thống thông tin.
2. Seminar của các nhóm trong lớp.
3. tài liệu Oracle của hãng Oracle:

<https://docs.oracle.com/en/database/oracle/oracle-database/index.html>

1. Tài liệu về Java, Java Swing: <https://docs.oracle.com/javase/tutorial/>
2. Những đồ án tham khảo của nhưng anh chị khoá trước.

* **Link chứa thư mục chính của đồ án:** <https://drive.google.com/drive/folders/1wTfaW9cpcHhEXG5-S5c3NwqtnpwpBflf?usp=sharing>
* Link chứa **file báo cáo**: <https://drive.google.com/drive/folders/1-qKki0dh-IKO34rGrhZPhBsjRJVo2TVM?usp=sharing>
* Link chứa **các file script**: <https://drive.google.com/drive/folders/1v2wuA9lKGXDzK1RD0UeqR6gQ-LH2OOys?usp=sharing>
* Link chứa **Source code chương trình**: <https://drive.google.com/drive/folders/1HVNgxO24i4ruiBpv-TFPkbH277nLqKTs?usp=sharing>
* Link chứa **file hướng dẫn cài đặt chương trình**: <https://drive.google.com/drive/folders/1TJYaAXZsuZB0GPEPRrJpz86Fgq-NgZTb?usp=sharing>
* Link chứa **tài liệu tham khảo**: <https://drive.google.com/drive/folders/1eKcN4ijATSSIH9CONs8-jQNlXyjL0jrU?usp=sharing>
* Link chứa **video trình bày đồ án môn học**: <https://drive.google.com/drive/folders/1wMp3QU2Owa2GfJz3xfX6jmSIVvgxLegJ?usp=sharing>