TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

BÀI BÁO CÁO KẾT THÚC HỌC PHẦN HỆ
QUẢN TRỊ CƠ SỞ DỮ LIỆU

XÂY DỰNG HỆ THỐNG CSDL QUẢN LÝ TRUNG TÂM NGOẠI NGỮ BẰNG ORACLE

Giảng viên hướng dẫn: Th.S Nguyễn Thanh Trung

Sinh viên thực hiện:

1. Nguyễn Hữu Bình 21DH113496
2. Nguyễn Ngọc Kiều Nhi 21DH112759
3. Nguyễn Lê Thị Bảo Châu 21DH113505

Thành phố Hồ Chí Minh, ngày 30 tháng 03 năm 2024

MỤC LỤC

MỤC LỤC 2
DANH MỤC BẢNG BIỂU 4
CHƯƠNG 1: KHẢO SÁT HIỆN TRẠNG VÀ XÁC ĐỊNH YÊU CẦU 7
1.1. Khảo sát hiện trạng 7
1.2. Mô tả nội dung cần giải quyết của đề tài 8
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 12
2.1 Kiến trúc Oracle 12
2.2 Quản trị user và profile trong Oracle 13
2.3 Ngôn ngữ PL/SQL 14
2.4 Cơ chế sao lưu và phục hồi dữ liệu sau sự cố 15
2.5 Quản lý giao tác/ giao dịch 16
2.6 Xử lý đồng thời 17
CHƯƠNG 3 - PHÂN TÍCH HỆ THỐNG 19
3.1. PHÂN TÍCH CHỨC NĂNG 19
3.1.2. Mô hình phân cấp chức năng(Sơ đồ BFD) 19
3.1.2. Mô tả chi tiết các chức năng 19
3.2. PHÂN TÍCH DỮ LIỆU 3.2.1. Sơ đồ ERD 24
CHƯƠNG 4- THIẾT KẾ VÀ CÀI ĐẶT HỆ THỐNG 25
4.1. Mô hình dữ liệu quan hệ (Relational Data model) 25
4.2 Từ điển dữ liệu (Mô tả các bảng và các thuộc tính) 25
4.2.1. Bảng HOCVIEN 25
4.2.2 Bảng BAITHI 25
4.2.3 Bảng CHUCVU 26
4.2.4 Bảng DIEM 26
4.2.5 Bảng GIAOVIEN 26
4.2.6 Bảng HOADON 26
4.2.7 Bảng KHOAHOC 27
4.2.8 Bảng LICHDAY 27
4.2.9 Bảng NHANVIEN 27
4.2.10 Bảng SACH 28
4.2.11 Bảng CHITIETHOADON 28
4.2.12 Bảng DANHSACHDANGKY 28
4.2.13 Bảng HOADONSACH 28
4.3.1. Quản lý lưu trữ dữ liệu 28
4.3.1.1. Mô tả cách tổ chức lưu trữ dữ liệu (Datafile, Tablespace, Index, …) 28

4.3.1.2. Cách tổ chức lưu trữ dữ liệu (Datafile, Tablespace, Index, …) 29
4.3.1.3. Dữ liệu demo 34
4.3.2. Thiết kế và cài đặt các Stored Procedure/Function cần thiết cho hệ thống 41
4.3.2.1 Dùng để thực hiện các lệnh CRUD cơ bản cho các Table 41
4.3.2.2.Chức năng nghiệp vụ 43
4.3.3.3. Kết xuất dữ liệu báo cáo 47
4.3.3. Thiết kế các Stored Procedure hoặc Trigger 48
4.3.4. Thiết kế và cài đặt các giao tác (Transaction) 48
4.3.4.1. Mô tả các tình huống cập nhật dữ liệu có thể gây rủi ro về tính nhất quán dữ liệu 48
4.3.4.2. Mô tả ngăn ngừa bằng các transaction 51
4.3.5. Minh họa vấn đề xử lý đồng thời 52
4.3.6. Thiết kế và cài đặt mô hình phân quyền của hệ thống 55
CHƯƠNG 5: KẾT LUẬN 61
5.1. Kết quả đạt được và chưa đạt được trong đề tài 61
5.2. Hướng phát triển mở rộng ứng dụng trong tương lai 61
Tài liệu tham khảo 62
PHỤ LỤC 63
1. PROCEDURE 63
PROCEDURE 1: Cập nhật khoá học 63
PROCEDURE 2: Xoá khoá học 63
PROCEDURE 3: Lấy thông tin khoá học 64
PROCEDURE 4: Cập nhật thông tin học viên 65
PROCEDURE 5: Lấy thông tin học viên 66
PROCEDURE 6: Thêm nhân viên 66
PROCEDURE 7: Cập nhật thông tin nhân viên 67
PROCEDURE 8: Xoá nhân viên 68
PROCEDURE 9: Lấy thông tin nhân viên 68
PROCEDURE 10: Cập nhật lương nhân viên 68
PROCEDURE 11:Thay đổi chức vụ nhân viên 69
PROCEDURE 12: Xem lịch dạy 70
PROCEDURE 13: Cập nhật trạng thái 71
PROCEDURE 14: Xoá đăng ký 71
PROCEDURE 16: Thêm chức vụ 73
PROCEDURE 17:Cập nhật chức vụ 73
PROCEDURE 18: Lấy thông tin chức vụ 73
PROCEDURE 19: Thêm bài thi 74
PROCEDURE 20: Sửa bài thi 74
PROCEDURE 21: Xoá bài thi 74
PROCEDURE 22: Bán sách 74

PROCEDURE 23: Lấy thông tin bài thi 75
PROCEDURE 24: Thêm sách 75
PROCEDURE 25: Cập nhật sách 76
PROCEDURE 26: Xoá sách 76
PROCEDURE 27: Lấy thông tin sách 77
PROCEDURE 28: Xoá chức vụ 77
2. TRIGGER 78
TRIGGER 1: Kiểm tra tránh trùng tên khoá học 78
TRIGGER 2: Kiểm tra khoá học có người đăng ký rồi thì không cho xoá 78
TRIGGER 3: Tăng giảm lương không quá 30% 79
TRIGGER 4: Tránh trùng lịch dạy 79
TRIGGER 5: Kiểm tra ngày đăng ký không quá ngày hiện tại 80
TRIGGER 6: Kiểm tra trùng bài thi 81
TRIGGER 7 : Không thể xoá bài thi khi có điểm tồn tại 81
TRIGGER 8: Không được xoá chức vụ có người đang đảm nhiệm 81
TRIGGER 9: Tránh trùng chức vụ lúc thêm chức vụ mới 82

LỜI MỞ ĐẦU

Cuộc sống con người ngày càng phát triển thì nhu cầu sống của con người cũng được nângcao và những nhu cầu học tập nhưng cũng rất thiết thực. Đây là vấn đề đặt ra cho các nhà kinh doanh, làm sao để đáp ứng nhu cầu học tập, trau dồi kiến thức của mọi người để cải thiện đời sống ngày càng cao.

Nhóm chúng em đã xây dựng nên một cơ sở dữ liệu bán khoá học và quản lý con người để đáp ứng nhu cầu của mọi người, giúp họ tiết kiệm được thời gian, chi phí thay vì phải đi ra các trung tâm.

Đơn giản hơn, chỉ việc ngồi ở nhà với chiếc máy tính đã kết nối internet mọi người có thể thỏa sức tìm kiếm trung tâm ngoại ngữ uy tín và chất lượng mà mình yêu thích mọi lúc mọi nơi. Giúp cho người quản trị dễ dàng trong việc phân quyền quản lí, quản lí khách hàng và các khoá học của họ. Nắm bắt được xu thế hiện nay, đó là lý do nhóm chúng em chọn đề tài "*XÂY DỰNG HỆ THỐNG CSDL QUẢN LÝ TRUNG TÂM NGOẠI NGỮ BẰNG ORACLE*".

LỜI CẢM ƠN

Nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến tất cả thầy cô Trường Đại Học Ngoại Ngữ – Tin Học TP. Hồ Chí Minh nói chung và các thầy cô trong khoa Công nghệ thông tin nói riêng đã tận tình giảng dạy, truyền đạt những kiến thức và kinh nghiệm quý báu cho chúng em trong suốt quá trình học tập tại trường.

Trong suốt thời gian nhóm em làm bài báo cáo đề tài môn Các hệ quản trị cơ sở dữ liệu, nhóm em xin gửi lời cảm ơn chân thành và sâu sắc đến thầy Ths. Nguyễn Thanh Trung, người thầy hết lòng giúp đỡ và theo nhóm chúng em trong suốt quá trình thực hiện đề tài *"XÂY DỰNG HỆ THỐNG CSDL QUẢN LÝ TRUNG TÂM NGOẠI NGỮ BẰNG ORACLE"* này đã chỉ ra hướng đi đúng đắn để nhóm em có thể hoàn thành chỉn chu và tốt nhất bài báo cáo theo đúng thời hạnquy định.

Trong quá trình thực hiện đề tài, dù nhóm chúng em đã cố gắng hoàn thiện đề tài một cách tốt nhất nhưng do thời gian và kiến thức vốn có của nhóm còn hạn chế nên sẽ không tránh khỏi những thiếu sót nhất định, rất mong nhận được sự cảm thông, chia sẻ đến từ nhữngđóng góp và góp ý chân thành từ quý thầy cô khoa Công nghệ thông tin.

Sau cùng nhóm em xin gửi lời cảm ơn đến tất cả các bạn đã tham gia đóng góp ý kiến và giúp đỡ cho nhóm em trong suốt quá trình thực hiện đề tài này.

Nhóm em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 23 tháng 3 năm 2024

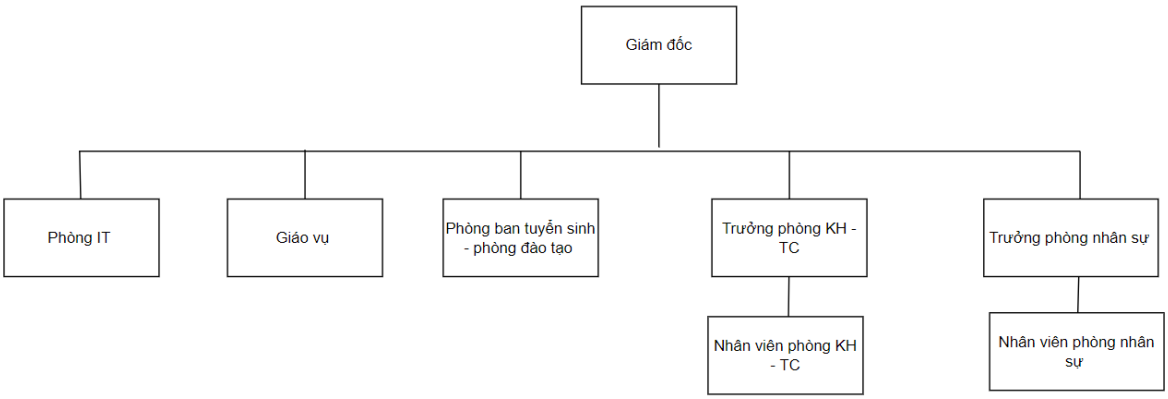
CHƯƠNG 1: KHẢO SÁT HIỆN TRẠNG VÀ XÁC ĐỊNH YÊU CẦU

1.1. Khảo sát hiện trạng

- Giới thiệu tổng quan về nghiệp vụ doanh nghiệp: Trung tâm đào tạo tiếng Anh là một cơ sở giáo dục tập trung vào việc cung cấp các khóa học ngoại ngữ, đặc biệt là tiếng Anh, cho các đối tượng học viên khác nhau như học sinh, sinh viên, người đi làm, người lớn tuổi, v.v. Quy trình hoạt động chính bao gồm:

    1. Tổ chức các khóa học tiếng Anh với các cấp độ khác nhau (từ cơ bản đến nâng cao) và các chuyên đề khác nhau (như IELTS, TOEIC, tiếng Anh giao tiếp, v.v.).
    2. Quản lý đăng ký khóa học và thu học phí từ học viên.
    3. Lập lịch dạy và sắp xếp giáo viên cho các khóa học.
    4. Tổ chức các kỳ thi đánh giá trình độ và cấp chứng chỉ cho học viên hoàn thành khóa học.

- Cơ cấu tổ chức của hệ thống: Trung tâm đào tạo tiếng Anh có cơ cấu tổ chức gồm các bộ phận chính sau:



    - Ban Giám đốc: Quản lý và điều hành hoạt động chung của trung tâm.
    - Phòng Kế hoạch- tài chính: Quản lý tài chính, thu chi học phí, lương nhân viên.
    - Phòng IT: Quản lý hệ thống công nghệ thông tin, phần mềm ứng dụng.

- Phòng nhân sự : Chịu trách nhiệm về nhiều mặt khác nhau liên quan đến quản lý nhân sự và phát triển tổ chức.
- Đội ngũ giáo viên: Giảng dạy các khóa học tiếng Anh.
- Phòng giáo vụ: Hỗ trợ các công việc hành chính, tư vấn học viên.

- Đánh giá hiện trạng hệ thống cũ và các vấn đề cần phải giải quyết của hệ thống mới: Hiện tại, trung tâm đang sử dụng các phần mềm quản lý riêng lẻ và ghi chép thủ công để theo dõi thông tin học viên, khóa học, lịch dạy, và thanh toán học phí. Điều này gây ra nhiều khó khăn và hạn chế như:

    1. Khó khăn trong việc truy xuất và tổng hợp dữ liệu một cách nhanh chóng và chính xác.
    2. Dữ liệu dễ bị mất mát hoặc sai sót do ghi chép thủ công.
    3. Không có cơ chế quản lý quyền truy cập và bảo mật dữ liệu.
    4. Khó khăn trong việc theo dõi và quản lý lịch sử giao dịch.
    5. Không có cơ chế sao lưu và phục hồi dữ liệu đầy đủ.

Hệ thống mới cần giải quyết các vấn đề trên bằng cách:

1. Xây dựng một cơ sở dữ liệu tập trung để lưu trữ và quản lý toàn bộ dữ liệu một cách có hệ thống.
2. Đảm bảo tính toàn vẹn, nhất quán và an toàn của dữ liệu.
3. Cung cấp các tính năng truy xuất, tổng hợp và báo cáo dữ liệu hiệu quả.
4. Có cơ chế quản lý người dùng, phân quyền truy cập và bảo mật dữ liệu.
5. Có khả năng sao lưu và phục hồi dữ liệu định kỳ hoặc khi có sự cố.

1.2. Mô tả nội dung cần giải quyết của đề tài

- Phạm vi, quy trình, nghiệp vụ mà đề tài sẽ giải quyết: Đề tài sẽ xây dựng một hệ thống quản lý trung tâm đào tạo tiếng Anh, bao gồm các quy trình và nghiệp vụ chính sau:

1. Quản lý thông tin học viên: Lưu trữ thông tin cá nhân, lịch sử đăng ký khóa học và thanh toán của học viên.
2. Quản lý thông tin giáo viên và nhân viên: Lưu trữ thông tin cá nhân, chức vụ, lịch dạy của giáo viên và nhân viên.
3. Quản lý khóa học: Lưu trữ thông tin chi tiết về các khóa học như tên khóa học, học phí, số buổi học, giáo trình sử dụng.
4. Quản lý đăng ký khóa học: Cho phép học viên đăng ký khóa học và theo dõi tình trạng đăng ký.
5. Quản lý thanh toán học phí: Lập hóa đơn, theo dõi tình trạng thanh toán của học viên.
6. Quản lý lịch dạy: Sắp xếp lịch dạy cho giáo viên theo khóa học và lớp học.
7. Quản lý tổ chức thi và kết quả học tập: Lưu trữ thông tin về các kỳ thi, điểm số và kết quả học tập của học viên.

- Các chứng từ, báo cáo, số liệu mà hệ thống cần quản lý: Hệ thống cần quản lý các loại chứng từ sau:

1. Hợp đồng đăng ký khóa học giữa trung tâm và học viên.
2. Hóa đơn thanh toán học phí của học viên.
3. Giấy chứng nhận hoàn thành khóa học cho học viên.

Ngoài ra, hệ thống cũng cần cung cấp các báo cáo và số liệu sau:

1. Báo cáo tình hình đăng ký khóa học theo từng khóa học, theo thời gian.
2. Báo cáo tình hình thanh toán học phí của học viên.
3. Báo cáo lịch dạy của giáo viên theo khóa học và thời gian biểu.
4. Báo cáo kết quả học tập (điểm số, xếp loại) của học viên theo khóa học.
5. Số liệu thống kê về số lượng học viên, giáo viên, khóa học, doanh thu, v.v.

- Các vấn đề liên quan đến hiệu năng xử lý dữ liệu trong hệ thống: Hệ thống cần đáp ứng các yêu cầu về hiệu năng xử lý dữ liệu sau:

  1. Khả năng xử lý và truy xuất dữ liệu nhanh chóng, đặc biệt trong thời gian cao điểm như khi mở đăng ký khóa học mới hoặc thời điểm tổ chức các kỳ thi.
  2. Khả năng xử lý đồng thời nhiều giao dịch (đăng ký, thanh toán, cập nhật dữ liệu mà không bị trì trệ hoặc xung đột dữ liệu.
  3. Khả năng mở rộng và nâng cấp khi có nhiều người dùng hơn hoặc lượng dữ liệu tăng lên trong tương lai.

- Các vấn đề về an toàn dữ liệu: Hệ thống cần đảm bảo an toàn dữ liệu bằng cách:

  1. Kiểm soát và phân quyền truy cập dữ liệu cho các đối tượng người dùng khác nhau (giáo viên, nhân viên, học viên) một cách phù hợp.
  2. Mã hóa dữ liệu nhạy cảm như thông tin cá nhân, tài khoản ngân hàng, v.v.
  3. Có cơ chế sao lưu và phục hồi dữ liệu định kỳ hoặc khi có sự cố để đảm bảo tính toàn vẹn và khả năng phục hồi của dữ liệu.

- Các vấn đề liên quan đến xử lý truy xuất đồng thời: Với nhiều người dùng (học viên, giáo viên, nhân viên) truy cập và thao tác trên hệ thống đồng thời, cần có các giải pháp sau:

  1. Cơ chế quản lý giao dịch (transaction) để đảm bảo tính nhất quán và toàn vẹn của dữ liệu khi có nhiều giao dịch đồng thời.
  2. Kiểm soát đồng bộ hóa dữ liệu (data concurrency control) để ngăn chặn xung đột dữ liệu khi nhiều người dùng đọc/ghi cùng một dữ liệu đồng thời.
  3. Cơ chế khóa dữ liệu (locking) để ngăn chặn việc cập nhật dữ liệu đồng thời trên cùng một đối tượng dữ liệu.

- Lý do sử dụng hệ quản trị Oracle sẽ giải quyết các vấn đề nêu trên: Oracle là một hệ quản trị cơ sở dữ liệu (DBMS) mạnh mẽ, đáp ứng được các yêu cầu đã nêu trên nhờ các tính năng sau:

1. Khả năng lưu trữ và quản lý dữ liệu lớn, đáp ứng nhu cầu mở rộng trong tương lai.
2. Cung cấp các cơ chế đảm bảo tính toàn vẹn, nhất quán và an toàn dữ liệu như quản lý giao dịch, kiểm soát đồng bộ hóa, khóa dữ liệu, mã hóa dữ liệu.
3. Hỗ trợ xử lý đồng thời nhiều giao dịch và truy xuất dữ liệu hiệu quả.
4. Cung cấp các tính năng quản lý người dùng, phân quyền truy cập dữ liệu.
5. Có cơ chế sao lưu và phục hồi dữ liệu linh hoạt, đáng tin cậy.
6. Hỗ trợ ngôn ngữ lập trình PL/SQL để xây dựng các ứng dụng phức tạp.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Kiến trúc Oracle

Kiến trúc Oracle, còn được gọi là kiến trúc hệ thống Oracle, là một bộ sưu tập các thành phần và công nghệ được sử dụng trong hệ thống quản trị cơ sở dữ liệu Oracle. Kiến trúc này đóng vai trò quan trọng trong việc xây dựng và vận hành hệ thống cơ sở dữ liệu Oracle:

- **Oracle Database:**

  Oracle Database là thành phần cốt lõi và chính của kiến trúc Oracle. Đây là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ được phát triển và bảo trì bởi Oracle Corporation. Oracle Database cung cấp các tính năng quản lý dữ liệu, bảo mật, giao dịch và tối ưu hóa truy vấn để đảm bảo hiệu suất và tính nhất quán của hệ thống.

- **Instance:**

  Mỗi khi một cơ sở dữ liệu Oracle được khởi động, một instance sẽ được tạo ra. Instance là một quá trình chạy trên máy chủ và đại diện cho một phiên bản cụ thể của Oracle Database. Nó quản lý và điều phối các hoạt động của cơ sở dữ liệu, bao gồm bộ nhớ đệm, tiến trình truy vấn, quản lý phiên, quản lý khóa và ghi nhật ký.

- **Bộ nhớ đệm (Buffer Cache):**

  Bộ nhớ đệm là một khu vực trong bộ nhớ của hệ thống được sử dụng để lưu trữ dữ liệu và các khối dữ liệu gần đây được truy cập từ cơ sở dữ liệu. Bằng cách lưu trữ dữ liệu trong bộ nhớ đệm, Oracle giảm thiểu số lần truy cập đến đĩa cứng, làm tăng hiệu suất truy vấn và thời gian đáp ứng của hệ thống.

- **Quá trình truy vấn (Query Processing):**

  Quá trình truy vấn là quá trình xử lý và thực hiện các truy vấn SQL được gửi đến cơ sở dữ liệu. Oracle sử dụng một bộ tối ưu hóa mạnh mẽ để phân tích câu truy vấn, tạo ra kế hoạch thực thi tối ưu và chọn phương pháp truy xuất dữ liệu hiệu quả. Điều này đảm bảo rằng truy vấn được thực hiện một cách nhanh chóng và hiệu quả.

- **Quản lý giao dịch (Transaction Management):**

  Oracle hỗ trợ quản lý giao dịch để đảm bảo tính nhất quán của dữ liệu trong

các hoạt động ghi và đọc. Mỗi giao dịch được xem như một đơn vị không thể phân chia và có thể được xác nhận (commit) hoặc hủy bỏ (rollback). Quản lý giao dịch đảm bảo rằng dữ liệu luôn ở trạng thái nhất quán trong quá trình thực hiện các thay đổi.

- **Quản lý lỗi và phục hồi (Error Management and Recovery):**

  Oracle cung cấp các cơ chế để quản lý lỗi và phục hồi dữ liệu trong trường hợp xảy ra sự cố. Hệ thống có khả năng ghi nhật ký (log) các hoạt động và thay đổi dữ liệu, cho phép khôi phục dữ liệu từ trạng thái ghi nhật ký trong trường hợp cần thiết. Điều này đảm bảo tính an toàn và khả năng phục hồi của dữ liệu trong trường hợp lỗi phần cứng, lỗi phần mềm hoặc thảm họa tự nhiên.

- **Các công nghệ mở rộng và phân tán (Scalability and Distributed Technologies):**

  Oracle hỗ trợ các công nghệ mở rộng và phân tán để đáp ứng nhu cầu quy mô và tải công việc lớn. Các tính năng như phân chia dữ liệu (partitioning), mô hình dữ liệu phân tán (distributed database), và khả năng kết hợp cơ sở dữ liệu (database clustering) cho phép mở rộng hệ thống và phân tải công việc trên nhiều máy chủ.

- **Các công nghệ bảo mật (Security Technologies):**

  Oracle cung cấp các công nghệ bảo mật mạnh mẽ để bảo vệ dữ liệu và hệ thống. Các tính năng bao gồm kiểm soát truy cập người dùng, xác thực và ủy quyền, mã hóa dữ liệu, theo dõi và ghi lại các hoạt động người dùng, và quản lý các quyền truy cập dựa trên vai trò (role-based access control). Điều này đảm bảo rằng dữ liệu được bảo vệ và chỉ có người dùng có quyền mới có thể truy cập vào nó.

Tổng quan, kiến trúc Oracle là một hệ thống phức tạp và mạnh mẽ, bao gồm nhiều thành phần và công nghệ để quản lý dữ liệu và đáp ứng các yêu cầu kinh doanh. Từ việc quản lý cơ sở dữ liệu cốt lõi cho đến quá trình truy vấn, quản lý giao dịch, phục hồi dữ liệu và bảo mật, Oracle cung cấp các tính năng và công nghệ để xây dựng và vận hành hệ thống cơ sở dữ liệu đáng tin cậy và hiệu quả.

2.2 Quản trị user và profile trong Oracle

Trong hệ quản trị cơ sở dữ liệu Oracle, quản trị user và profile là hai khái niệm quan trọng để bảo mật và quản lý người dùng:

User trong Oracle:

- Người dùng (User): Là tài khoản được tạo để kết nối và làm việc với cơ sở dữ liệu Oracle. Mỗi người dùng có một tên đăng nhập và mật khẩu duy nhất. Người dùng này có thể được gán quyền và hạn chế truy cập theo nhu cầu công việc cụ thể.
- Quyền (Privilege): Là quyền hạn cụ thể mà người dùng có thể có trong cơ sở dữ liệu. Các quyền bao gồm quyền SELECT, INSERT, UPDATE, DELETE và nhiều quyền khác.
- Role: Là một tập hợp các quyền được gán cho người dùng. Người quản trị có thể tạo các role để dễ dàng quản lý quyền và gán chúng cho người dùng.

Profile trong Oracle:

- Profile: Là một tập hợp các giới hạn và cấu hình cho người dùng. Nó giúp quản trị viên kiểm soát và giới hạn sử dụng tài nguyên hệ thống của người dùng. Các thông số cấu hình có thể bao gồm giới hạn tài nguyên CPU, thời gian phiên làm việc, số lần đăng nhập không thành công trước khi tài khoản bị khóa, và nhiều hơn nữa.
- Resource Limit: Profile cũng có thể được sử dụng để giới hạn tài nguyên hệ thống sẵn có cho mỗi người dùng. Điều này bao gồm cả CPU, bộ nhớ, và tài nguyên I/O.
- Assign Profile: Mỗi người dùng có thể được gán một profile cụ thể. Khi người dùng kết nối vào cơ sở dữ liệu, các hạn chế và cấu hình từ profile đó sẽ áp dụng cho phiên làm việc của họ.

2.3 Ngôn ngữ PL/SQL

Ngôn ngữ PL/SQL (Procedural Language/Structured Query Language) là một ngôn ngữ lập trình kết hợp cả SQL và các cấu trúc lập trình procedural để tạo ra các chương trình lưu trữ trong cơ sở dữ liệu Oracle:

- **Procedural Language**: PL/SQL là một ngôn ngữ lập trình procedural, có nghĩa là nó hỗ trợ cấu trúc lập trình như điều kiện, vòng lặp, khai báo biến, và các chức năng procedural khác. Điều này giúp trong việc tổ chức và kiểm soát logic của chương trình.
- **Integration with SQL**: PL/SQL được thiết kế để tương tác chặt chẽ với SQL, cho phép nhúng các câu lệnh SQL trực tiếp vào mã PL/SQL và ngược lại. Điều này tạo ra khả năng linh hoạt khi thao tác dữ liệu trong cơ sở dữ liệu Oracle.

- **Block Structure**: Mỗi chương trình PL/SQL được tổ chức thành các khối (blocks) logic. Một khối có thể là một đơn giản là một câu lệnh SQL hoặc một đoạn mã procedural phức tạp hơn. Các khối PL/SQL có thể nhúng bên trong nhau để tạo thành cấu trúc phức tạp.
- **Exception Handling**: PL/SQL hỗ trợ cơ chế xử lý ngoại lệ (exception handling) để giúp quản lý và xử lý các tình huống đặc biệt như lỗi trong quá trình thực thi chương trình.
- **Procedures và Functions**: PL/SQL hỗ trợ việc định nghĩa các thủ tục (procedures) và hàm (functions). Thủ tục thường được sử dụng để thực hiện các tác vụ và không trả về giá trị, trong khi hàm trả về một giá trị.
- ***Packages***: Packages là cấu trúc tổ chức cao cấp hơn, cho phép đóng gói nhiều thủ tục, hàm và biến liên quan lại thành một đơn vị duy nhất. Điều này giúp trong việc quản lý và triển khai các đối tượng PL/SQL.
- **Triggers**: PL/SQL cung cấp triggers (còn được gọi là các bộ kích hoạt) để tự động thực hiện các hành động khi có sự kiện cụ thể xảy ra trong cơ sở dữ liệu, như INSERT, UPDATE, hoặc DELETE.
- **Dynamic** SQL: PL/SQL hỗ trợ sử dụng câu lệnh SQL động, giúp tạo ra các truy vấn có thể thay đổi tại thời điểm thực thi.

2.4 Cơ chế sao lưu và phục hồi dữ liệu sau sự cố

Cơ chế sao lưu và phục hồi dữ liệu là một phần quan trọng trong quản lý cơ sở dữ liệu để đảm bảo an toàn và khả dụng của dữ liệu. Trong Oracle, có các cơ chế và quy trình cụ thể để thực hiện sao lưu và phục hồi dữ liệu sau sự cố:

- **Sao lưu Dữ liệu**: Oracle hỗ trợ nhiều phương tiện sao lưu, bao gồm RMAN (Recovery Manager), Data Pump, và các công cụ sao lưu khác. RMAN thường được ưa chuộng vì nó cung cấp khả năng quản lý sao lưu và phục hồi mạnh mẽ, tự động và linh hoạt.
- **Recovery Catalog**: RMAN sử dụng Recovery Catalog để lưu trữ thông tin về các hoạt động sao lưu và phục hồi. Nó giúp theo dõi lịch sử sao lưu và phục hồi, cũng như quản lý các bản sao lưu.

- **Chế độ Archive Log**: Để thực hiện phục hồi đến một thời điểm cụ thể, Oracle yêu cầu cơ sở dữ liệu ở chế độ Archive Log. Trong chế độ này, redo log được lưu trữ vào các tập tin log archive, cho phép phục hồi đến bất kỳ điểm nào trong quá khứ.
- **Sao lưu Đầy đủ và Sao lưu Incremental**: Sao lưu đầy đủ là việc sao lưu toàn bộ cơ sở dữ liệu và logs. Sao lưu Incremental chỉ sao lưu các thay đổi kể từ sao lưu trước đó. Kết hợp cả hai loại sao lưu giúp giảm bớt thời gian và tài nguyên.
- **Point-in-Time Recovery (PITR):** Oracle cho phép thực hiện phục hồi đến một điểm cụ thể trong quá khứ, giúp khôi phục dữ liệu đến thời điểm trước khi sự cố xảy ra.
- **Flashback Technology**: Oracle cung cấp công nghệ Flashback, bao gồm Flashback Database và Flashback Table, cho phép quay trở lại trạng thái trước đó của toàn bộ cơ sở dữ liệu hoặc một bảng cụ thể mà không cần phục hồi từ sao lưu.
- **Automatic Diagnostic Repository (ADR):** Oracle sử dụng ADR để lưu trữ và quản lý thông tin liên quan đến sự cố. Nó giúp trong việc xác định và giải quyết vấn đề sau sự cố.
- **Quản lý Redo Log và Undo Tablespace**: Redo log ghi lại tất cả các thay đổi được thực hiện trên cơ sở dữ liệu, trong khi undo tablespace chứa các phiên bản trước đó của dữ liệu. Cả hai đều quan trọng trong quá trình phục hồi.

2.5 Quản lý giao tác/ giao dịch

Quản lý giao dịch (Transaction Management) là một khía cạnh quan trọng của hệ quản trị cơ sở dữ liệu Oracle để đảm bảo tính nhất quán và độ tin cậy của dữ liệu trong quá trình thực hiện các thao tác:

1. **Giao Dịch (Transaction):**

Giao dịch là một chuỗi các thao tác SQL được thực hiện trên cơ sở dữ liệu. Một giao dịch có thể bao gồm nhiều câu lệnh SQL như INSERT, UPDATE, DELETE, và SELECT.

2. **ACID Properties:**

- Oracle tuân theo các thuộc tính **ACID** (Atomicity, Consistency, Isolation, Durability) để đảm bảo tính nhất quán và an toàn của giao dịch.
- **Atomicity**: Giao dịch là một đơn vị không thể chia nhỏ, nghĩa là hoặc tất cả các thao tác thành công, hoặc không có thao tác nào được thực hiện.

- **Consistency**: Giao dịch đưa cơ sở dữ liệu từ một trạng thái hợp lệ đến một trạng thái hợp lệ khác mà không vi phạm ràng buộc dữ liệu.
- **Isolation**: Giao dịch đang thực hiện không ảnh hưởng đến giao dịch khác đang diễn ra đồng thời. Điều này đảm bảo tính cô lập giữa các giao dịch.
- **Durability**: Khi một giao dịch kết thúc thành công, các thay đổi được áp dụng vĩnh viễn và không bao giờ bị mất.

3. **Bắt đầu và kết thúc giao dịch:**

Giao dịch bắt đầu khi thực hiện một câu lệnh BEGIN TRANSACTION hoặc một câu lệnh SQL đầu tiên của giao dịch. Giao dịch kết thúc bằng câu lệnh COMMIT hoặc ROLLBACK.

4. **Commit và Rollback:**

- **COMMIT**: Giao dịch sẽ kết thúc thành công và tất cả các thay đổi sẽ được lưu trữ vào cơ sở dữ liệu.
- **ROLLBACK**: Giao dịch sẽ bị hủy và tất cả các thay đổi sẽ bị loại bỏ, trả lại cơ sở dữ liệu về trạng thái trước giao dịch.

5. **Savepoints:**

Savepoints là điểm trong giao dịch mà bạn có thể đặt ra để có thể quay lại nếu cần thiết. Savepoints cho phép bạn thực hiện ROLLBACK đến một điểm nhất định trong giao dịch mà không ảnh hưởng đến các phần khác của giao dịch.

6. **Deadlocks và Locks:**

Oracle quản lý việc đồng thời nhiều giao dịch thông qua quá trình quản lý khóa (Lock Management) để đảm bảo tính nhất quán và cô lập giữa các giao dịch. Deadlocks là tình trạng mà hai hoặc nhiều giao dịch bị kẹt vì chúng đang chờ đợi nhau.

7. **Transaction Control Statements:**

Oracle cung cấp các câu lệnh SQL như COMMIT, ROLLBACK, SAVEPOINT để kiểm soát giao dịch.

2.6 Xử lý đồng thời

Xử lý đồng thời (Concurrency Control) là một khía cạnh quan trọng của quản lý cơ sở dữ liệu, đặc biệt là trong hệ thống cơ sở dữ liệu Oracle. Các hệ thống cơ sở dữ liệu thường phải xử lý nhiều giao dịch đồng thời và đảm bảo tính nhất quán và độ cô lập của

dữ liệu:

- **Giao Dịch (Transaction):**

Giao dịch là một chuỗi các hành động cơ sở dữ liệu mà theo đó cơ sở dữ liệu chuyển từ một trạng thái hợp lý sang một trạng thái khác. Giao dịch cần đảm bảo tính ACID (Atomicity, Consistency, Isolation, Durability).

- Giao Thức Đọc Bảo Toàn (Read Consistency):

Oracle sử dụng giao thức đọc bảo toàn để đảm bảo rằng một giao dịch đang đọc dữ liệu không bị ảnh hưởng bởi các giao dịch khác đang thay đổi dữ liệu cùng một lúc. Điều này giúp đảm bảo tính nhất quán của dữ liệu trong quá trình đọc.

- **Phân Cấp Độ Cô Lập (Isolation Levels):**

Oracle hỗ trợ các mức độ độ cô lập khác nhau cho các giao dịch, như READ COMMITTED, SERIALIZABLE, và các mức độ khác. Mức độ độ cô lập xác định mức độ giao tác khác có thể tác động đến dữ liệu của giao tác hiện tại.

- **Locking (Khóa):**

Oracle sử dụng khóa để đảm bảo tính nhất quán và độ cô lập. Có các loại khóa như khóa đọc, khóa ghi, và khóa cấp cao hơn như khóa hàng và khóa bảng. Sự quản lý khóa được thực hiện để tránh xung đột và đảm bảo tính nhất quán.

- **Kiểm Soát Xung Đột (Conflict Detection and Resolution):**

Oracle sử dụng các kỹ thuật kiểm soát xung đột để phát hiện và giải quyết xung đột giữa các giao dịch đồng thời. Các xung đột có thể xảy ra khi nhiều giao dịch cố gắng truy cập hoặc thay đổi cùng một dữ liệu.

- **Undo và Redo Logs:**

Oracle sử dụng undo logs để lưu lại phiên bản trước đó của dữ liệu và redo logs để lưu lại các thay đổi. Điều này hỗ trợ quá trình rollback (hoàn tác) và phục hồi (recovery) khi cần thiết.

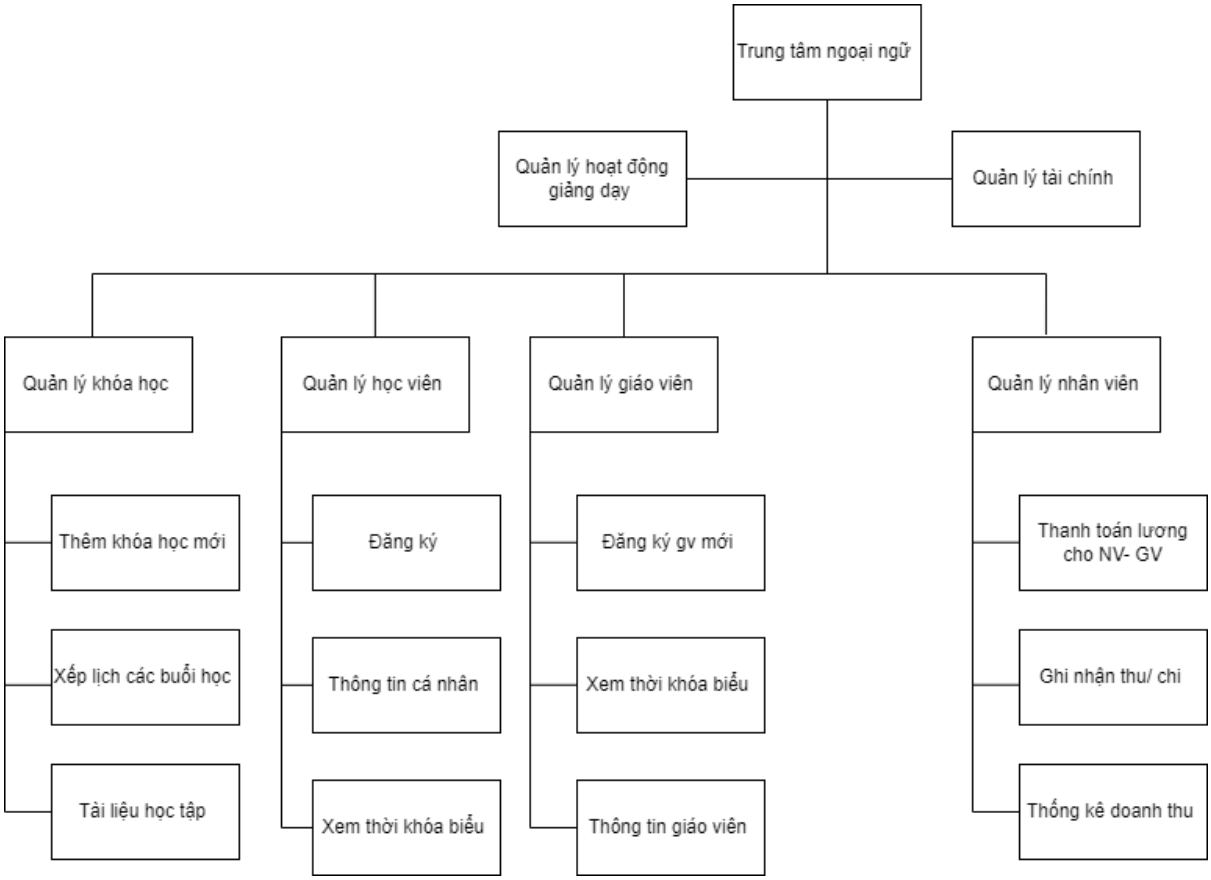
- **Dòng Thời Gian (Timestamps):**

Oracle sử dụng dòng thời gian để theo dõi thời điểm mỗi giao dịch được bắt đầu và kết thúc. Điều này giúp quản lý và xử lý đồng thời hiệu quả hơn.

CHƯƠNG 3 - PHÂN TÍCH HỆ THỐNG

3.1. PHÂN TÍCH CHỨC NĂNG

3.1.2. Mô hình phân cấp chức năng(Sơ đồ BFD)



3.1.2. Mô tả chi tiết các chức năng

STT	Tên chức năng	Bộ phận liên quan	Chứng từ/ báo cáo liên quan	Dữ liệu cần lưu trữ	Dữ liệu cần truy xuất
02	Cập nhật chương trình học và tài liệu	Quản lý khóa học		Mã tài liệu, số lượng, tên tài liệu, giá	Mã tài liệu, số lượng, tên tài liệu, giá

03	Thêm khóa học mới	Quản lý khóa học		Thông tin và nội dung khóa học	Thông tin khóa học, thông tin chương trình học, thông tin lịch học, thông tin giảng viên
04	Sắp xếp các buổi học	Quản lý khóa học		Thời gian và địa điểm của buổi học, lịch trình buổi học	
05	Đăng ký	Quản lý giáo viên		Thông tin cá nhân và mật khẩu, thông tin đăng ký	
06	Nhập thông tin cá nhân	Quản lý học viên		Thông tin cá nhân (tên, địa chỉ,…)	Thông tin cá nhân(tên, ngày tháng năm sinh, giới tính,….)
07	Xem thời khóa biểu	Giáo viên, học viên			Mã khóa học, tên khóa học, thời gian học, giáo viên, phòng học
08	Đăng ký giáo viên mới	Giáo vụ		Tên giáo viên, tên, email, địa	

				chỉ, số điện thoại, giới tính	
10	Xem lịch dạy	Giáo viên			Tên giáo viên, môn học, lớp học, thời gian học
11	Xem thông tin giáo viên	Nhân viên		Thông tin giảng viên(tên, dạy môn gì, email, địa chỉ, ngày sinh, giới tính	Thông tin giảng viên(tên, dạy môn gì, email, địa chỉ, ngày sinh, giới tính
14	Ghi nhận thu/chi	Thu ngân	Phiếu thu/chi, hóa đơn, biên lai	Ngày thu/chi, mã giao dịch, loại thu/chi, số tiền,…	Ngày ghi nhận, loại giao dịch, số tiền, hình thức thanh toán, người thực hiện giao dịch, học phẩm
15	Thanh toán lương cho NV/GV	Quản lý nhân viên		Ngày thanh toán, mã giao dịch, loại thanh toán, số tiền, họ tên, mã, lương tháng mấy	Ngày thanh toán, , loại thanh toán, số tiền, họ tên, lương tháng mấy

3.1.3. Các ràng buộc toàn vẹn dữ liệu liên quan

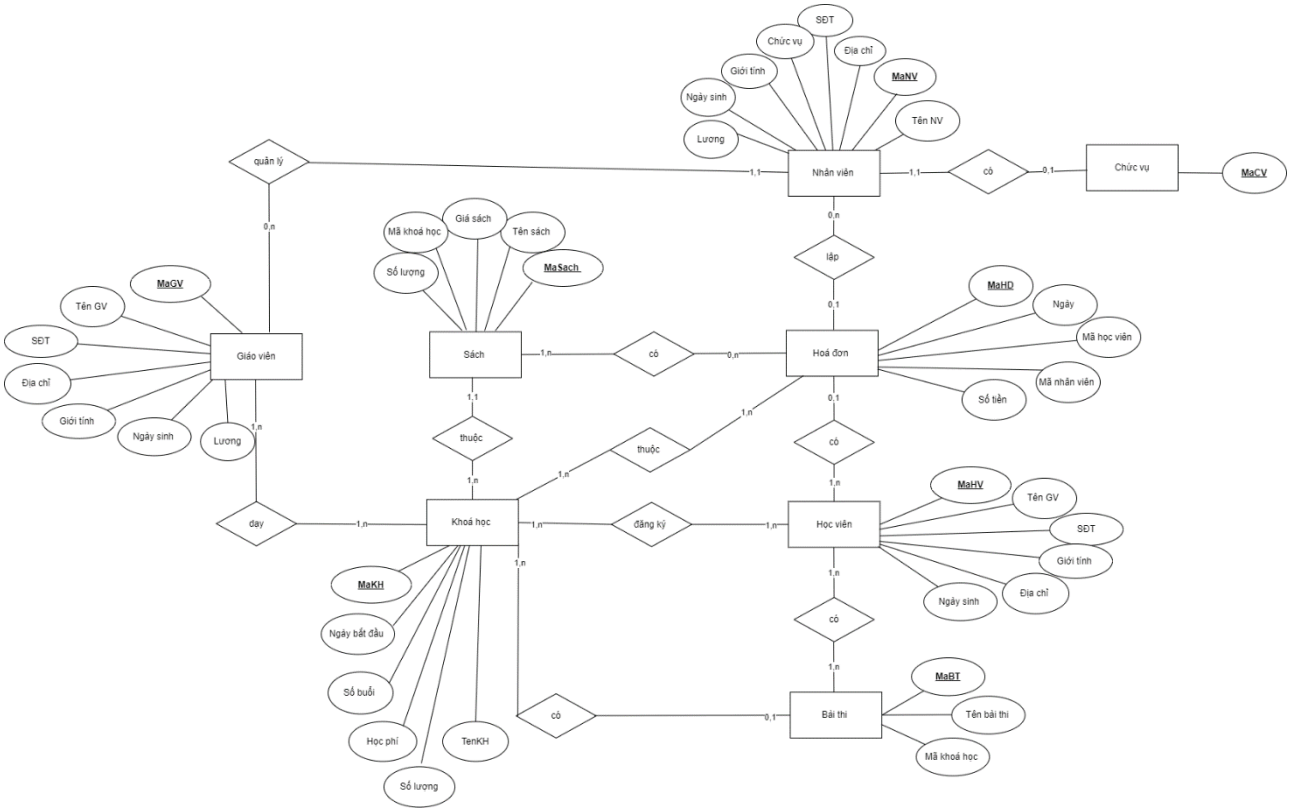
STT	Nội dung ràng buộc	Là Ràng buộc dữ liệu	Là Quy định

		(Data Rule)	**nghiệp vụ (Business Logic)**
1	Mỗi học viên có thể đăng ký nhiều khóa học nhưng không được trùng giờ học	x	
2	Một lớp không được quá sức chứa đã quy định	x	
3	Một học viên phải có ít nhất 1 bảng điểm	x	
4	Mỗi học viên phải có ít nhất 1 hóa đơn	x	
5	Khóa học sẽ được bắt đầu dạy khi số lượng học viên nhiều hơn ⅓ số lượng học viên tối đa của khóa học.	x	
6	Một khóa học phải có tối thiểu 1 giảng viên		x
7	Mỗi bài thi phải có 1 điểm cố định		x
8	Mỗi bài thi chỉ được phúc khảo 1 lần	x	
9	Mỗi giáo viên phải có ít nhất 1 bảng lương		x
10	Nhân viên phải có ít nhất 1 bảng lương		x

11	Học phẩm phải được quản lý bởi 1 hoặc nhiều nhân viên	x	

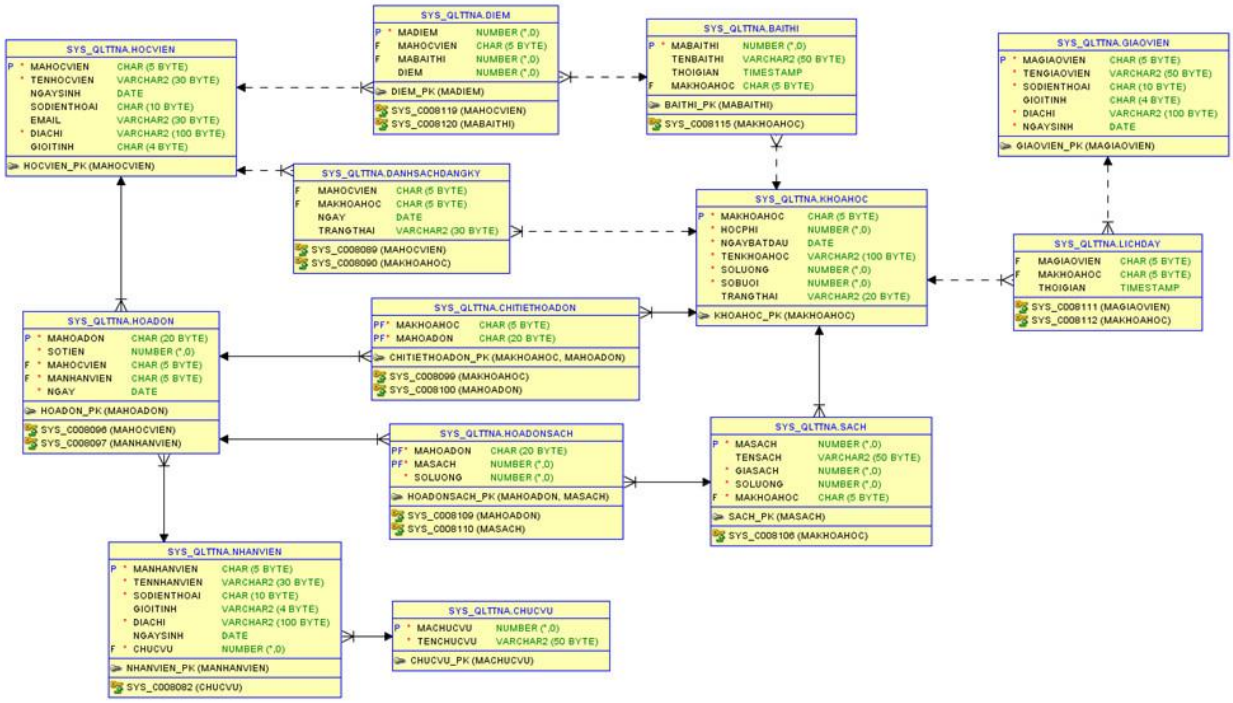
3.2. PHÂN TÍCH DỮ LIỆU

3.2.1. Sơ đồ ERD



CHƯƠNG 4- THIẾT KẾ VÀ CÀI ĐẶT HỆ THỐNG

4.1. Mô hình dữ liệu quan hệ (Relational Data model)



4.2 Từ điển dữ liệu (Mô tả các bảng và các thuộc tính)

4.2.1. Bảng HOCVIEN

STT	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc
1	MAHOCVIEN	CHAR	5	PK
2	TENHOCVIEN	VARCHAR2	30	
3	NGAYSINH	DATE		
4	SODIENTHOAI	CHAR	10	
5	EMAIL	VARCHAR2	30	
6	DIACHI	VARCHAR2	100	
7	GIOITINH	CHAR	4	

4.2.2 Bảng BAITHI

STT	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc
1	MABAITHI	NUMBER	(38,0)	PK
2	TENBAITHI	VARCHAR2	50	

3	THOIGIAN	TIMESTAMP	6	
4	MAKHOAHOC	CHAR	5	FK

4.2.3 Bảng CHUCVU

STT	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc
1	MACHUCVU	NUMBER	(38,0)	PK
2	TENCHUCVU	VARCHAR2	50	

4.2.4 Bảng DIEM

STT	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc
1	MADIEM	NUMBER	(38,0)	PK
2	MAHOCVIEN	CHAR	5	FK
3	MABAITHI	NUMBER	(38,0)	FK
4	DIEM	NUMBER	(38,0)	

4.2.5 Bảng GIAOVIEN

STT	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc
1	MAGIAOVIEN	CHAR	5	PK
2	TENGIAOVIEN	VARCHAR2	50	
3	SODIENTHOAI	CHAR	10	
4	GIOITINH	CHAR	4	
5	DIACHI	VARCHAR2	100	
6	NGAYSINH	DATE		

4.2.6 Bảng HOADON

STT	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc
1	MAHOADON	CHAR	20	PK
2	SOTIEN	NUMBER	(38,0)	
3	MAHOCVIEN	CHAR	5	FK

4	MANHANVIEN	CHAR	5	FK
5	NGAY	DATE		

4.2.7 Bảng KHOAHOC

STT	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc
1	MAKHOAHOC	CHAR	5	PK
2	HOCPHI	NUMBER	(38,0)	
3	NGAYBATDAU	DATE		
4	TENKHOAHOC	VARCHAR2	100	
5	SOLUONG	NUMBER	(38,0)	
6	SOBUOI	NUMBER	(38,0)	
7	TRANGTHAI	VARCHAR2	20	

4.2.8 Bảng LICHDAY

STT	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc
1	MAGIAOVIEN	CHAR	5	PK, FK
2	MAKHOAHOC	CHAR	5	PK, FK
3	THOIGIAN	TIMESTAMP	6	

4.2.9 Bảng NHANVIEN

STT	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc
1	MANHANVIEN	CHAR	5	PK
2	TENNHANVIEN	VARCHAR	30	
3	SODIENTHOAI	CHAR	10	
4	GIOITINH	VARCHAR	4	
5	DIACHI	VARCHAR	100	
6	NGAYSINH	DATE		
7	CHUCVU	NUMBER	(38,0)	FK

4.2.10 Bảng SACH

STT	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc
1	MASACH	NUMBER	(38,0)	PK
2	TENSACH	VARCHAR2	50	
3	GIASACH	NUMBER	(38,0)	
4	SOLUONG	NUMBER	(38,0)	
5	MAKHOAHOC	CHAR	5	FK

4.2.11 Bảng CHITIETHOADON

STT	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc
1	MAKHOAHOC	CHAR	5	PK, FK
2	MAHOADON	CHAR	5	PK, FK

4.2.12 Bảng DANHSACHDANGKY

STT	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc
1	MAHOCVIEN	CHAR	5	FK
2	MAKHOAHOC	CHAR	5	FK
3	NGAY	DATE		
4	TRANGTHAI	VARCHAR2	30	

4.2.13 Bảng HOADONSACH

STT	Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc
1	MAHOADON	CHAR	20	PK, FK
2	MASACH	NUMBER	(38,0)	PK, FK
3	SOLUONG	NUMBER	(38,0)	

4.3.1. Quản lý lưu trữ dữ liệu

4.3.1.1. Mô tả cách tổ chức lưu trữ dữ liệu (Datafile, Tablespace, Index, …)

- Datafile:

- Admin: lưu các lệnh gán quyền cho user SYS_QLTTNA

- Schema: lưu các câu lệnh tạo bảng
- Master_data: lưu dữ liệu của các bảng danh mục/phân loại
- Transaction_data**:** lưu dữ liệu của các bảng giao dịch/nghiệp vụ, procedure và trigger

- Tablespace: lấy theo giá trị mặc định của Oracle

4.3.1.2. Cách tổ chức lưu trữ dữ liệu (Datafile, Tablespace, Index, …)

Table 1: Tạo bảng CHUCVU

```
CREATE TABLE chucvu (
   machucvu  INT
      GENERATED ALWAYS AS IDENTITY
   NOT NULL PRIMARY KEY,
   tenchucvu VARCHAR2(50) NOT NULL
);
```

Table 2: Tạo bảng KHOAHOC

```
CREATE TABLE khoahoc (
   makhoahoc  CHAR(5) PRIMARY KEY CHECK ( makhoahoc LIKE 'KH%' ),
   hocphi     INT NOT NULL,
   ngaybatdau DATE NOT NULL,
   tenkhoahoc VARCHAR2(100) NOT NULL,
   soluong    INT NOT NULL,
   sobuoi     INT NOT NULL,
   trangthai  VARCHAR2(20) DEFAULT 'Chuẩn bị' CHECK ( trangthai IN ( 'Chuẩn bị',
'Đang dạy', 'Đã kết thúc' ) )
);
```

Table 3: Tạo bảng GIAOVIEN

```
CREATE TABLE giaovien (
   magiaovien  CHAR(5) PRIMARY KEY CHECK ( magiaovien LIKE 'GV%' ),
   tengiaovien VARCHAR2(50) NOT NULL,
   sodienthoai CHAR(10) NOT NULL,
   gioitinh    CHAR(4) CHECK ( gioitinh IN ( 'Nam', 'Nữ' ) ),
   diachi      VARCHAR2(100) NOT NULL,
```

```
    ngaysinh    DATE NOT NULL,
    luong int not null
);
```

Table 4: Tạo bảng NHANVIEN

```
CREATE TABLE nhanvien (
    manhanvien  CHAR(5) PRIMARY KEY CHECK ( manhanvien LIKE 'NV%' ),
    tennhanvien VARCHAR2(30) NOT NULL,
    sodienthoai CHAR(10) NOT NULL,
    gioitinh    VARCHAR2(4)
        CONSTRAINT gioitinh CHECK ( gioitinh IN ( 'Nam', 'Nữ' ) ),
    diachi      VARCHAR2(100) NOT NULL,
    ngaysinh    DATE,
    chucvu      INT NOT NULL,
    luong int not null,
    FOREIGN KEY ( chucvu )
        REFERENCES chucvu ( machucvu )
);
```

Table 5: Tạo bảng HOCVIEN

```
CREATE TABLE hocvien (
    mahocvien   CHAR(5) PRIMARY KEY CHECK ( mahocvien LIKE 'HV%' ),
    tenhocvien  VARCHAR2(30) NOT NULL,
    ngaysinh    DATE,
    sodienthoai CHAR(10),
    email       CHAR(30),
    diachi      VARCHAR2(100) NOT NULL,
    gioitinh    CHAR(4) CHECK ( gioitinh IN ( 'Nam', 'Nữ' ) )
);
```

Table 6: Tạo bảng DANHSACHDANGKY

```
CREATE TABLE danhsachdangky (
    mahocvien CHAR(5),
    makhoahoc CHAR(5),
```

```
   ngay      DATE,
   trangthai VARCHAR2(30) DEFAULT 'Chưa thanh toán' CHECK ( trangthai IN (
'Chưa thanh toán', 'Đã thanh toán' ) ),
   FOREIGN KEY ( mahocvien )
      REFERENCES hocvien ( mahocvien ),
   FOREIGN KEY ( makhoahoc )
      REFERENCES khoahoc ( makhoahoc )
);
```

Table 7: Tạo bảng HOADON

```
CREATE TABLE hoadon (
   mahoadon   CHAR(20) PRIMARY KEY,
   sotien     INT NOT NULL,
   mahocvien  CHAR(5) NOT NULL,
   manhanvien CHAR(5) NOT NULL,
   ngay date not null,
   FOREIGN KEY ( mahocvien )
      REFERENCES hocvien ( mahocvien ),
   FOREIGN KEY ( manhanvien )
      REFERENCES nhanvien ( manhanvien )
);
```

Table 8: Tạo bảng CHITIETHOADON

```
CREATE TABLE chitiethoadon (
   makhoahoc CHAR(5),
   mahoadon  CHAR(20),
   FOREIGN KEY ( makhoahoc )
      REFERENCES khoahoc ( makhoahoc ),
   FOREIGN KEY ( mahoadon )
      REFERENCES hoadon ( mahoadon ),
   PRIMARY KEY ( makhoahoc,
                 mahoadon )
);
```

Table 9: Tạo bảng SACH

```
CREATE TABLE sach (
   masach    INT
      GENERATED BY DEFAULT ON NULL AS IDENTITY
   PRIMARY KEY,
   tensach   VARCHAR2(50),
   giasach   INT NOT NULL,
   soluong   INT NOT NULL,
   makhoahoc CHAR(5) NOT NULL,
   FOREIGN KEY ( makhoahoc )
      REFERENCES khoahoc ( makhoahoc )
);
```

Table 10: Tạo bảng HOADONSACH

```
CREATE TABLE hoadonsach (
   mahoadon CHAR(20),
   masach   INT,
   soluong  INT NOT NULL,
   FOREIGN KEY ( mahoadon )
      REFERENCES hoadon ( mahoadon ),
   FOREIGN KEY ( masach )
      REFERENCES sach ( masach ),
   PRIMARY KEY ( mahoadon,
                 masach )
);
```

Table 11: Tạo bảng LICHDAY

```
CREATE TABLE lichday (
   magiaovien CHAR(5),
   makhoahoc  CHAR(5),
   thoigian   TIMESTAMP,
   FOREIGN KEY ( magiaovien )
      REFERENCES giaovien ( magiaovien ),
```

```
    FOREIGN KEY ( makhoahoc )
        REFERENCES khoahoc ( makhoahoc )
);
```

Table 12: Tạo bảng BAITHI

```
CREATE TABLE baithi (
    mabaithi  INT
        GENERATED BY DEFAULT ON NULL AS IDENTITY
    PRIMARY KEY,
    tenbaithi VARCHAR2(50),
    thoigian  TIMESTAMP,
    makhoahoc CHAR(5),
    FOREIGN KEY ( makhoahoc ) REFERENCES khoahoc ( makhoahoc )
);
```

Table 13: Tạo bảng DIEM

```
CREATE TABLE diem (
    madiem    INT
        GENERATED BY DEFAULT ON NULL AS IDENTITY
    PRIMARY KEY,
    mahocvien CHAR(5),
    mabaithi  INT,
    diem      INT CHECK ( diem >= 0 AND diem <= 10 ),
    FOREIGN KEY ( mahocvien ) REFERENCES hocvien ( mahocvien ),
    FOREIGN KEY ( mabaithi ) REFERENCES baithi ( mabaithi )
);
CREATE OR REPLACE TYPE HoaDonSachType AS OBJECT (
    MaSach INT,
    SoLuong INT
);

CREATE OR REPLACE TYPE HoaDonSachList AS TABLE OF HoaDonSachType;
```

```
CREATE OR REPLACE TYPE ChiTietHoaDonType AS OBJECT (
    MaKhoaHoc CHAR(5)
);

CREATE OR REPLACE TYPE ChiTietHoaDonList AS TABLE OF
ChiTietHoaDonType;
```

4.3.1.3. Dữ liệu demo

--Thêm chức vụ

```
insert into ChucVu(TenChucVu) values ('Giám đốc');
insert into ChucVu(TenChucVu) values ('Trưởng phòng kế toán');
insert into ChucVu(TenChucVu) values ('Thu ngân');
insert into ChucVu(TenChucVu) values ('Kế toán');
insert into ChucVu(TenChucVu) values ('Trưởng phòng IT');
insert into ChucVu(TenChucVu) values ('Trưởng phòng marketing');
insert into ChucVu(TenChucVu) values ('Nhân viên IT');
insert into ChucVu(TenChucVu) values ('Nhân viên marketing');
```

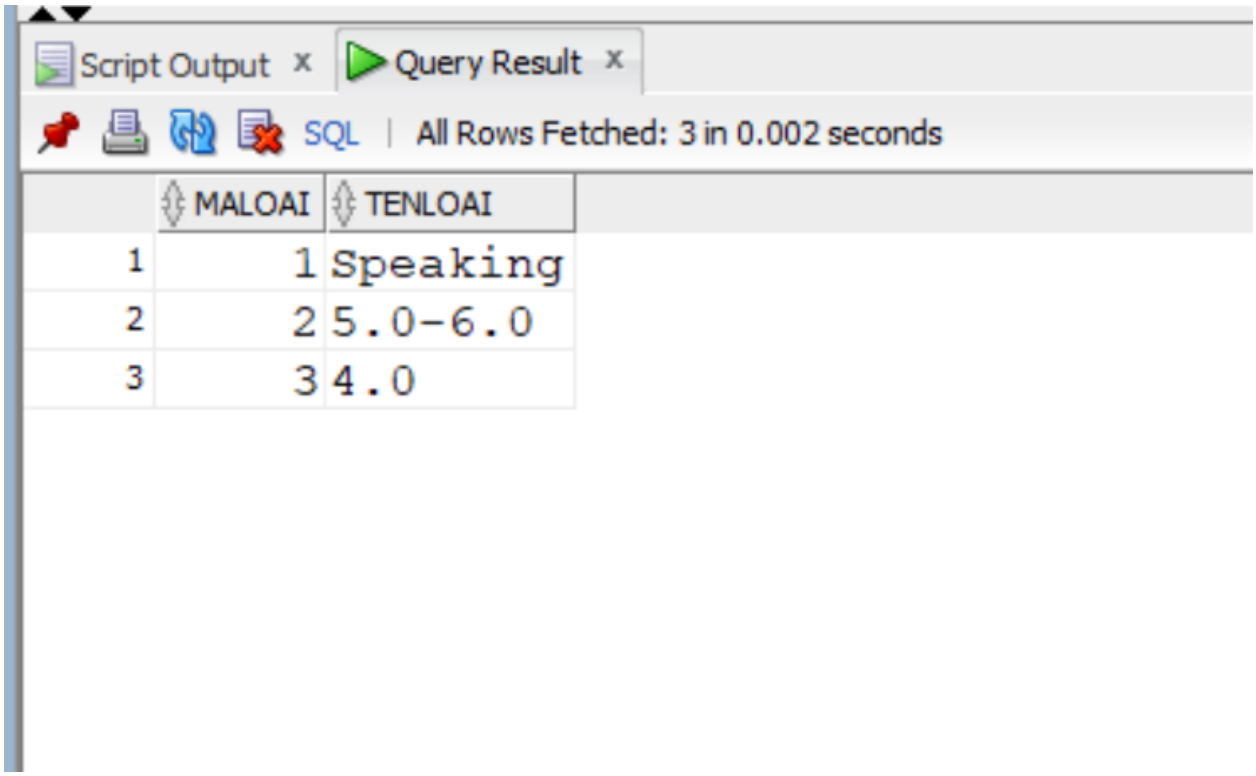
Script Output × Query Result ×

SQL | All Rows Fetched: 8 in 0.051 seconds

	MACHUCVU	TENCHUCVU
1	1	Giám đốc
2	2	Trưởng phòng kế toán
3	3	Thu ngân
4	4	Kế toán
5	5	Trưởng phòng IT
6	6	Trưởng phòng marketing
7	7	Nhân viên IT
8	8	Nhân viên marketing

--Thêm loại khoá học

```
insert into LoaiKhoaHoc(TenLoai) values('Speaking');
insert into LoaiKhoaHoc(TenLoai) values('5.0-6.0');
insert into LoaiKhoaHoc(TenLoai) values('4.0');
insert into LoaiKhoaHoc(TenLoai) values('6.0-6.5');
insert into LoaiKhoaHoc(TenLoai) values('6.0-7.0');
```



	MALOAI	TENLOAI
1	1	Speaking
2	2	5.0-6.0
3	3	4.0

--Thêm khoá học

```
insert into
KhoaHoc(MaKhoaHoc,HocPhi,NgayBatDau,TenKhoaHoc,SoLuong,SoBuoi,Loai)
values ('KH001',5000000,to_date('01/01/2024','dd/mm/yyyy'),'IELTS
Foundation',30,30,1);
insert into
KhoaHoc(MaKhoaHoc,HocPhi,NgayBatDau,TenKhoaHoc,SoLuong,SoBuoi,Loai)
values ('KH002',9000000,to_date('02/01/2024','dd/mm/yyyy'),'IELTS Intensive',30,50,2);

insert into
KhoaHoc(MaKhoaHoc,HocPhi,NgayBatDau,TenKhoaHoc,SoLuong,SoBuoi,Loai)
values ('KH003',20000000,to_date('03/01/2024','dd/mm/yyyy'),'IELTS Fighter',20,50,3);
insert into
KhoaHoc(MaKhoaHoc,HocPhi,NgayBatDau,TenKhoaHoc,SoLuong,SoBuoi,Loai)
values ('KH004',3000000,to_date('04/01/2024','dd/mm/yyyy'),'IELTS Basic',50,30,2);
insert into
KhoaHoc(MaKhoaHoc,HocPhi,NgayBatDau,TenKhoaHoc,SoLuong,SoBuoi,Loai)
values ('KH005',4000000,to_date('05/01/2024','dd/mm/yyyy'),'IELTS Pre
Foundation',40,30,1);
insert into
KhoaHoc(MaKhoaHoc,HocPhi,NgayBatDau,TenKhoaHoc,SoLuong,SoBuoi,Loai)
values ('KH006',5000000,to_date('06/01/2024','dd/mm/yyyy'),'Mất gốc',20,20,2);
```

Script Output | Query Result

All Rows Fetched: 6 in 0.015 seconds

	MAKHOAHOC	HOCPHI	NGAYBATDAU	TENKHOAHOC	SOLUONG	SOBUOI	TRANGTHAI	LOAI
1	KH001	5000000	01-JAN-24	IELTS Foundation	30	30	Chuẩn bị	1
2	KH002	9000000	02-JAN-24	IELTS Intensive	30	50	Chuẩn bị	2
3	KH003	20000000	03-JAN-24	IELTS Fighter	20	50	Chuẩn bị	3
4	KH004	3000000	04-JAN-24	IELTS Basic	50	30	Chuẩn bị	2
5	KH005	4000000	05-JAN-24	IELTS Pre Foundation	40	30	Chuẩn bị	1
6	KH006	5000000	06-JAN-24	Mất gốc	20	20	Chuẩn bị	2

-- **Thêm giáo viên**

insert into GiaoVien(MaGiaoVien,TenGiaoVien,SoDienThoai,GioiTinh,DiaChi,NgaySinh,luong) values ('GV001','Nguyễn Hữu Bình','0987562173','Nam','15/22/1 Nguyễn Hữu Thọ, phường Tân Kiểng, quận 7, TP Hồ Chí Minh',to_date('03/11/2000','dd/mm/yyyy'),15000000);

insert into GiaoVien(MaGiaoVien,TenGiaoVien,SoDienThoai,GioiTinh,DiaChi,NgaySinh,luong) values ('GV002','Nguyễn Ngọc Kiều Nhi','0984325921','Nữ','44/15 đường Nguyễn Thị Minh Khai, phường 3, quận 1, TP Hồ Chí Minh',to_date('16/01/2000','dd/mm/yyyy'),10000000);

insert into GiaoVien(MaGiaoVien,TenGiaoVien,SoDienThoai,GioiTinh,DiaChi,NgaySinh,luong) values ('GV003','Lê Thị Bảo Châu','0845263712','Nữ','232 Nguyễn Thị Thập, phường Tân Phú, quận 7, TP Hồ Chí Minh',to_date('13/09/2000','dd/mm/yyyy'),20000000);

insert into GiaoVien(MaGiaoVien,TenGiaoVien,SoDienThoai,GioiTinh,DiaChi,NgaySinh,luong) values ('GV004','Trần Gia Huy','0836427329','Nam','32 đường Nguyễn Văn Cừ, phường 8, quận 5, TP Hồ Chí Minh',to_date('08/12/2000','dd/mm/yyyy'),12000000);

Script Output | Query Result

All Rows Fetched: 4 in 0.005 seconds

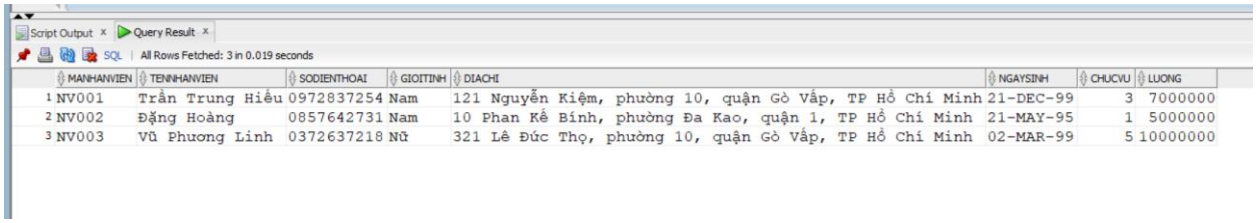
	MAGIAOVIEN	TENGIAOVIEN	SODIENTHOAI	GIOITINH	DIACHI	NGAYSINH	LUONG
1	GV001	Nguyễn Hữu Bình	0987562173	Nam	15/22/1 Nguyễn Hữu Thọ, phường Tân Kiểng, quận 7, TP Hồ Chí Minh	03-NOV-00	15000000
2	GV002	Nguyễn Ngọc Kiều Nhi	0984325921	Nữ	44/15 đường Nguyễn Thị Minh Khai, phường 3, quận 1, TP Hồ Chí Minh	16-JAN-00	10000000
3	GV003	Lê Thị Bảo Châu	0845263712	Nữ	232 Nguyễn Thị Thập, phường Tân Phú, quận 7, TP Hồ Chí Minh	13-SEP-00	20000000
4	GV004	Trần Gia Huy	0836427329	Nam	32 đường Nguyễn Văn Cừ, phường 8, quận 5, TP Hồ Chí Minh	08-DEC-00	12000000

--Thêm nhân viên

insert into NhanVien(MaNhanVien, TenNhanVien, SoDienThoai, GioiTinh, DiaChi, NgaySinh, ChucVu,luong)
values ('NV001','Trần Trung Hiếu','0972837254','Nam','121 Nguyễn Kiệm, phường 10, quận Gò Vấp, TP Hồ Chí Minh',to_date('1999-12-21','yyyy/mm/dd'),3,7000000);

insert into NhanVien(MaNhanVien, TenNhanVien, SoDienThoai, GioiTinh, DiaChi, NgaySinh, ChucVu,luong)

values ('NV002','Đặng Hoàng','0857642731','Nam','10 Phan Kế Bính, phường Đa Kao, quận 1, TP Hồ Chí Minh',to_date('1995-05-21','yyyy/mm/dd') ,1,5000000);
insert into NhanVien(MaNhanVien, TenNhanVien, SoDienThoai, GioiTinh, DiaChi, NgaySinh, ChucVu,luong)
values ('NV003','Vũ Phương Linh','0372637218','Nữ','321 Lê Đức Thọ, phường 10, quận Gò Vấp, TP Hồ Chí Minh',to_date('1999-03-02','yyyy/mm/dd'),5,10000000);


Script Output × Query Result ×
SQL | All Rows Fetched: 3 in 0.019 seconds

	MANHANVIEN	TENNHANVIEN	SODIENTHOAI	GIOITINH	DIACHI	NGAYSINH	CHUCVU	LUONG
1	NV001	Trần Trung Hiếu	0972837254	Nam	121 Nguyễn Kiệm, phường 10, quận Gò Vấp, TP Hồ Chí Minh	21-DEC-99	3	7000000
2	NV002	Đặng Hoàng	0857642731	Nam	10 Phan Kế Bính, phường Đa Kao, quận 1, TP Hồ Chí Minh	21-MAY-95	1	5000000
3	NV003	Vũ Phương Linh	0372637218	Nữ	321 Lê Đức Thọ, phường 10, quận Gò Vấp, TP Hồ Chí Minh	02-MAR-99	5	10000000

--Thêm học viên
insert into HocVien(MaHocVien, TenHocVien, NgaySinh, SoDienThoai, Email, DiaChi, GioiTinh)
values ('HV001','Nguyễn Tường Vy',to_date('2008/04/03','yyyy/mm/dd'),'0678932841','nguyentuongvy03@gmail.com','10/3 Thích Quảng Đức, phường 3, quận Phú Nhuận, TP Hồ Chí Minh','Nữ');
insert into HocVien(MaHocVien, TenHocVien, NgaySinh, SoDienThoai, Email, DiaChi, GioiTinh)
values ('HV002','Trần Mạnh Cường',to_date('2006/08/21','yyyy/mm/dd'),'0678522351','cuongtm1008@gmail.com','103/11/3 Cách mạng tháng 8, phường 11, quận Tân Bình, TP Hồ Chí Minh','Nam');

--Thêm danh sách đăng ký
insert into DanhSachDangKy (MaHocVien, MaKhoaHoc,Ngay)
values ('HV001','KH001',to_date('2023/11/13','yyyy/mm/dd'));
insert into DanhSachDangKy (MaHocVien, MaKhoaHoc,Ngay)
values ('HV002','KH005',to_date('2023/12/30','yyyy/mm/dd'));
insert into DanhSachDangKy (MaHocVien, MaKhoaHoc,Ngay)
values ('HV001','KH003',to_date('2024/01/13','yyyy/mm/dd'));
insert into DanhSachDangKy (MaHocVien, MaKhoaHoc, Ngay)
values ('HV002','KH002',to_date('2024/02/13','yyyy/mm/dd'));

Script Output | Query Result

All Rows Fetched: 4 in 0.003 seconds

	MAHOCVIEN	MAKHOAHOC	NGAY	TRANGTHAI
1	HV001	KH001	13-NOV-23	Chưa thanh toán
2	HV002	KH005	30-DEC-23	Chưa thanh toán
3	HV001	KH003	13-JAN-24	Chưa thanh toán
4	HV002	KH002	13-FEB-24	Chưa thanh toán

-- Thêm hoá đơn

```
insert into HoaDon(MaHoaDon, SoTien, MaHocVien, MaNhanVien,ngay)
values ('HDT3001',3000000,'HV001','NV001',to_date('21/03/2021','dd/mm/yyyy'));
insert into HoaDon(MaHoaDon, SoTien, MaHocVien, MaNhanVien,ngay)
values ('HDT3002',9000000,'HV002','NV002',to_date('21/03/2021','dd/mm/yyyy'));
insert into HoaDon(MaHoaDon, SoTien, MaHocVien, MaNhanVien,ngay)
values ('HDT3003',10000000,'HV001','NV003',to_date('21/03/2021','dd/mm/yyyy'));
```

Script Output | Query Result

All Rows Fetched: 3 in 0.013 seconds

	MAHOADON	SOTIEN	MAHOCVIEN	MANHANVIEN	NGAY
1	HDT3001	3000000	HV001	NV001	21-MAR-21
2	HDT3002	9000000	HV002	NV002	21-MAR-21
3	HDT3003	10000000	HV001	NV003	21-MAR-21

--Thêm chi tiết hoá đơn

```
insert into ChiTietHoaDon(MaKhoaHoc, MaHoaDon)
values ('KH001','HDT3001');
insert into ChiTietHoaDon(MaKhoaHoc, MaHoaDon)
values ('KH002','HDT3002');
insert into ChiTietHoaDon(MaKhoaHoc, MaHoaDon)
values ('KH003','HDT3003');
```

Script Output x | Query Result x

SQL | All Rows Fetched: 3 in 0.003 seconds

	MAKHOAHOC	MAHOADON
1	KH001	HDT3001
2	KH002	HDT3002
3	KH003	HDT3003

--Thêm sách

insert into Sach(TenSach, GiaSach, SoLuong, MaKhoaHoc)
values ('Tài liệu IELTS Foundation',1200000,5,'KH001');
insert into Sach(TenSach, GiaSach, SoLuong, MaKhoaHoc)
values ('Tài liệu tiếng Anh mất gốc','800000',10,'KH006');

Script Output x | Query Result x

SQL | All Rows Fetched: 2 in 0.002 seconds

	MASACH	TENSACH	GIASACH	SOLUONG	MAKHOAHOC
1	1	Tài liệu IELTS Foundation	1200000	5	KH001
2	2	Tài liệu tiếng Anh mất gốc	800000	10	KH006

--Thêm hoá đơn sách

insert into HoaDonSach(MaHoaDon,MaSach,SoLuong)
values ('HDT3001',1,2);
insert into HoaDonSach(MaHoaDon,MaSach,SoLuong)
values ('HDT3002',2,3);
insert into HoaDonSach(MaHoaDon,MaSach,SoLuong)
values ('HDT3003',1,2);

Script Output | Query Result

SQL | All Rows Fetched: 3 in 0.017 seconds

	MAHOADON	MASACH	SOLUONG
1	HDT3001	1	2
2	HDT3002	2	3
3	HDT3003	1	2

--Thêm lịch dạy

insert into LichDay(MaGiaoVien, MaKhoaHoc, ThoiGian)
values ('GV001','KH001',to_timestamp('2024-03-02 15:00:00','yyyy/mm/dd HH24:MI:SS'));
insert into LichDay(MaGiaoVien, MaKhoaHoc, ThoiGian)
values ('GV002','KH002',to_timestamp('2024-03-02 17:00:00','yyyy/mm/dd HH24:MI:SS'));
insert into LichDay(MaGiaoVien, MaKhoaHoc, ThoiGian)
values ('GV003','KH003',to_timestamp('2024-03-03 19:00:00','yyyy/mm/dd HH24:MI:SS'));

Script Output | Query Result

SQL | All Rows Fetched: 3 in 0.009 seconds

	MAGIAOVIEN	MAKHOAHOC	THOIGIAN
1	GV001	KH001	02-MAR-24 03.00.00.000000000 PM
2	GV002	KH002	02-MAR-24 05.00.00.000000000 PM
3	GV003	KH003	03-MAR-24 07.00.00.000000000 PM

--Thêm bài thi

insert into BaiThi(TenBaiThi, ThoiGian, MaKhoaHoc)
values ('Kiểm tra chất lượng đầu vào',to_timestamp('2024-02-13 09:00:00','yyyy/mm/dd HH24:MI:SS'),'KH001');
insert into BaiThi(TenBaiThi, ThoiGian, MaKhoaHoc)
values ('Kiểm tra chất lượng đầu vào',to_timestamp('2024-02-13 09:00:00','yyyy/mm/dd HH24:MI:SS'),'KH002');
insert into BaiThi(TenBaiThi, ThoiGian, MaKhoaHoc)
values ('Kiểm tra xét bằng', to_timestamp('2024-03-10 16:00:00','yyyy/mm/dd HH24:MI:SS'),'KH002');

Script Output × Query Result ×

SQL | All Rows Fetched: 3 in 0.014 seconds

	MABAITHI	TENBAITHI	THOIGIAN	MAKHOAHOC
1	1	Kiểm tra chất lượng đầu vào	13-FEB-24 09.00.00.000000000 AM	KH001
2	2	Kiểm tra chất lượng đầu vào	13-FEB-24 09.00.00.000000000 AM	KH002
3	3	Kiểm tra xét bằng	10-MAR-24 04.00.00.000000000 PM	KH002

--Thêm điểm

insert into Diem(MaHocVien, MaBaiThi, Diem)
values ('HV001',1,5);
insert into Diem(MaHocVien, MaBaiThi, Diem)
values ('HV002',2,7);

Script Output × Query Result ×

SQL | All Rows Fetched: 2 in 0.02 seconds

	MADIEM	MAHOCVIEN	MABAITHI	DIEM
1	1	HV001	1	5
2	2	HV002	2	7

4.3.2. Thiết kế và cài đặt các Stored Procedure/Function cần thiết cho hệ thống

4.3.2.1 Dùng để thực hiện các lệnh CRUD cơ bản cho các Table

Thêm lịch dạy

```
CREATE OR REPLACE PROCEDURE themlichday (
   p_magiaovien giaovien.magiaovien%TYPE,
   p_makhoahoc  khoahoc.makhoahoc%TYPE,
   p_thoigian   lichday.thoigian%TYPE
) AS
   v_count INT := 0;
   v_sobuoi int;
BEGIN
   insert into lichday(magiaovien,makhoahoc,thoigian)
   values (p_magiaovien,p_makhoahoc,p_thoigian);
   dbms_output.put_line('Thêm lịch dạy thành công');
END;
/
```

Sửa lịch dạy

```
create or replace procedure sualichday(
   p_magiaovien giaovien.magiaovien%type,
   p_makhoahoc khoahoc.makhoahoc%type,
   p_thoigiancu lichday.thoigian%type,
   p_thoigianmoi lichday.thoigian%type)
as
   v_lichday lichday%rowtype;
begin
   select * into v_lichday
   from lichday
   where magiaovien=p_magiaovien
   and makhoahoc=p_makhoahoc
   and thoigian=p_thoigiancu;

   if p_thoigiancu>sysdate then
      raise_application_error(-20010,'Lịch dạy đã qua, không thể sửa');
   end if;

   update lichday
   set thoigian=p_thoigianmoi
   where magiaovien=p_magiaovien
   and makhoahoc=p_makhoahoc
   and thoigian=p_thoigiancu;

   dbms_output.put_line('Cập nhật hành công');
end;
/
```

Thêm học viên

```
CREATE OR REPLACE PROCEDURE ThemHocVien (
   p_MaHocVien IN HocVien.MaHocVien%TYPE,
   p_TenHocVien IN HocVien.TenHocVien%TYPE,
   p_NgaySinh IN HocVien.NgaySinh%TYPE,
   p_SoDienThoai IN HocVien.SoDienThoai%TYPE,
   p_Email IN HocVien.Email%TYPE,
   p_DiaChi IN HocVien.DiaChi%TYPE,
   p_GioiTinh IN HocVien.GioiTinh%TYPE
) AS
BEGIN
   IF p_NgaySinh >= SYSDATE THEN
      DBMS_OUTPUT.PUT_LINE('Lỗi: Ngày sinh không hợp lệ.');
      RETURN;
   END IF;
```

```
   INSERT INTO HocVien (MaHocVien, TenHocVien, NgaySinh, SoDienThoai, Email,
DiaChi, GioiTinh)
   VALUES (p_MaHocVien, p_TenHocVien, p_NgaySinh, p_SoDienThoai, p_Email,
p_DiaChi, p_GioiTinh);

   COMMIT;
   DBMS_OUTPUT.PUT_LINE('Học viên đã được thêm vào thành công.');
EXCEPTION
   WHEN DUP_VAL_ON_INDEX THEN
      DBMS_OUTPUT.PUT_LINE('Lỗi: Mã học viên đã tồn tại.');
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
END ThemHocVien;
```

4.3.2.2. Chức năng nghiệp vụ

Hóa đơn

```
-- Định nghĩa kiểu dữ liệu danh sách chi tiết sách
CREATE OR REPLACE TYPE HoaDonSachType AS OBJECT (
   MaSach VARCHAR2(100),
   SoLuong NUMBER
);
/
CREATE OR REPLACE TYPE HoaDonSachList AS TABLE OF HoaDonSachType;
/

-- Định nghĩa kiểu dữ liệu danh sách chi tiết khoá học
CREATE OR REPLACE TYPE ChiTietHoaDonType AS OBJECT (
   MaKhoaHoc VARCHAR2(100)
);
CREATE OR REPLACE TYPE ChiTietHoaDonList AS TABLE OF
ChiTietHoaDonType;

-- Tạo procedure để thêm hóa đơn
CREATE OR REPLACE PROCEDURE ThemHoaDon (
   p_MaHoaDon IN HoaDon.MaHoaDon%TYPE,
   p_SoTien IN HoaDon.SoTien%TYPE,
```

```
   p_MaHocVien IN HoaDon.MaHocVien%TYPE,
   p_ngay in Hoadon.ngay%type,
   p_MaNhanVien IN HoaDon.MaNhanVien%TYPE,
   p_ChiTietSach IN HoaDonSachList DEFAULT NULL,
   p_ChiTietKhoaHoc IN ChiTietHoaDonList DEFAULT NULL
) AS
   v_TongSoTienChiTietSach INT := 0;
   v_TongSoTienChiTietKhoaHoc INT := 0;
   v_TongSoTienHoaDon INT;
   v_temp int;
BEGIN
   set transaction isolation level serializable;

   FOR rec IN (SELECT * FROM TABLE(p_ChiTietSach)) LOOP
      -- Lấy mã sách từ list
      SELECT GiaSach into v_temp  FROM Sach WHERE MaSach = rec.MaSach;
      v_TongSoTienChiTietSach := v_TongSoTienChiTietSach + rec.SoLuong * v_temp;
   END LOOP;

   -- Tính tổng số tiền từ khoá học
   FOR rec IN (SELECT * FROM TABLE(p_ChiTietKhoaHoc)) LOOP
      -- Lấy mã khoá học từ list
      SELECT HocPhi into v_temp FROM KhoaHoc WHERE MaKhoaHoc =
rec.MaKhoaHoc;
      v_TongSoTienChiTietKhoaHoc := v_TongSoTienChiTietKhoaHoc + v_temp;
   END LOOP;
   dbms_output.put_line(v_TongSoTienChiTietSach);
   dbms_output.put_line(v_TongSoTienChiTietKhoaHoc);
   dbms_output.put_line(p_SoTien);
   -- Tính tổng số tiền của hóa đơn
   v_TongSoTienHoaDon := v_TongSoTienChiTietSach +
v_TongSoTienChiTietKhoaHoc;

   -- Kiểm tra xem tổng số tiền của hóa đơn có bằng số tiền nhập vào không
   IF p_SoTien != v_TongSoTienHoaDon THEN
      RAISE_APPLICATION_ERROR(-20001, 'Lỗi: Số tiền của hóa đơn không khớp
với tổng số tiền của các mục chi tiết.');
   END IF;

   -- Thêm thông tin vào bảng HoaDon
   INSERT INTO HoaDon (MaHoaDon, SoTien, MaHocVien, MaNhanVien,Ngay)
   VALUES (p_MaHoaDon, p_SoTien, p_MaHocVien, p_MaNhanVien,p_Ngay);

   -- Thêm chi tiết hóa đơn sách vào bảng HoaDonSach nếu có
    IF p_ChiTietSach IS NOT NULL THEN
      FOR i IN 1..p_ChiTietSach.COUNT LOOP
```

```
            INSERT INTO HoaDonSach (MaHoaDon, MaSach, SoLuong)
            VALUES (p_MaHoaDon, p_ChiTietSach(i).MaSach,
p_ChiTietSach(i).SoLuong);
            bansach(p_ChiTietSach(i).MaSach,p_chitietsach(i).soluong);
        END LOOP;
    END IF;

    -- Thêm chi tiết hóa đơn khoá học vào bảng ChiTietHoaDon nếu có
    IF p_ChiTietKhoaHoc IS NOT NULL THEN
        FOR i IN 1..p_ChiTietKhoaHoc.COUNT LOOP
            INSERT INTO ChiTietHoaDon (MaKhoaHoc, MaHoaDon)
            VALUES (p_ChiTietKhoaHoc(i).MaKhoaHoc, p_MaHoaDon);
        END LOOP;
    END IF;

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Hóa đơn đã được thêm thành công và số tiền đã được
kiểm tra.');
EXCEPTION
    WHEN OTHERS THEN
        -- Rollback transaction nếu có lỗi
        ROLLBACK;
        RAISE_APPLICATION_ERROR(-20002, 'Lỗi: ' || SQLERRM);
END ThemHoaDon;
/

-- lệnh test
DECLARE
    -- Khai báo các biến đầu vào
    v_MaHoaDon HoaDon.MaHoaDon%TYPE := 'HD001';
    v_SoTien HoaDon.SoTien%TYPE := 7400000;
    v_MaHocVien HoaDon.MaHocVien%TYPE := 'HV001';
    v_MaNhanVien HoaDon.MaNhanVien%TYPE := 'NV001';
    v_Ngay Hoadon.ngay%type := to_date('20/03/2024','dd/mm/yyyy');
    -- Khai báo danh sách chi tiết sách và chi tiết khoá học
    v_ChiTietSach HoaDonSachList := HoaDonSachList();
    v_ChiTietKhoaHoc ChiTietHoaDonList := ChiTietHoaDonList();

BEGIN
    -- Thêm một chi tiết sách vào danh sách chi tiết sách
    v_ChiTietSach.EXTEND;
    v_ChiTietSach(1) := HoaDonSachType(1, 2); -- Giả sử mã sách là 1 và số lượng là 2
    -- Thêm một chi tiết khoá học vào danh sách chi tiết khoá học
    v_ChiTietKhoaHoc.EXTEND;
    v_ChiTietKhoaHoc(1) := ChiTietHoaDonType('KH001'); -- Giả sử mã khoá học là
'KH001'
```

```
   -- Gọi procedure ThemHoaDon với các tham số đã được khai báo
   ThemHoaDon(v_MaHoaDon, v_SoTien, v_MaHocVien, v_Ngay, v_MaNhanVien,
v_ChiTietSach, v_ChiTietKhoaHoc);

   -- In ra thông báo sau khi thực thi procedure thành công
   DBMS_OUTPUT.PUT_LINE('Procedure đã được thực thi thành công.');
EXCEPTION
   WHEN OTHERS THEN
      -- Xử lý nếu có lỗi xảy ra
      DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
END;
```

--- **Khoá học**

```
CREATE OR REPLACE PROCEDURE themkhoahoc (
   p_makhoahoc  IN khoahoc.makhoahoc%TYPE,
   p_hocphi     IN khoahoc.hocphi%TYPE,
   p_ngaybatdau IN khoahoc.ngaybatdau%TYPE,
   p_tenkhoahoc IN khoahoc.tenkhoahoc%TYPE,
   p_soluong    IN khoahoc.soluong%TYPE,
   p_sobuoi     IN khoahoc.sobuoi%TYPE
) AS
   v_count INT := 0;
BEGIN
   INSERT INTO khoahoc (
      makhoahoc,
      hocphi,
      ngaybatdau,
      tenkhoahoc,
      soluong,
      sobuoi
   ) VALUES (
      p_makhoahoc,
      p_hocphi,
      p_ngaybatdau,
```

```
        p_tenkhoahoc,
        p_soluong,
        p_sobuoi
    );

    COMMIT;
    dbms_output.put_line('Khóa học được thêm thành công.');
END;
```

4.3.3.3. Kết xuất dữ liệu báo cáo

```
CREATE OR REPLACE PROCEDURE ThongKeLuongHocVienMoi(p_ngaybatdau
varchar,p_ngayketthuc varchar)
as
    v_count int;
begin
    select count(*) into v_count from danhsachdangky
    where to_date(ngay,'dd/mm/yyyy') between to_date(p_ngaybatdau,'dd/mm/yyyy') and
to_date(p_ngayketthuc,'dd/mm/yyyy');

    dbms_output.put_line('Từ ngày '||p_ngaybatdau|| 'đến ngày '|| p_ngayketthuc||' có
'||v_count||' học viên đăng ký mới');

end;
/
exec ThongKeLuongHocVienMoi('01/01/2024','23/03/2024');

CREATE OR REPLACE PROCEDURE ThongKeDoanhThuTheoThang(p_ngaybatdau
varchar,p_ngayketthuc varchar)
as
    v_total int;
begin
    select sum(sotien) into v_total
    from hoadon
    where to_date(ngay,'dd/mm/yyyy') between to_date(p_ngaybatdau,'dd/mm/yyyy') and
to_date(p_ngayketthuc,'dd/mm/yyyy');

    dbms_output.put_line('Doanh thu từ ngày '|| p_ngaybatdau|| ' tới ngày
'||p_ngayketthuc||' là : '||v_total);
end;
/

exec ThongKeDoanhThuTheoThang('01/01/2024','24/03/2024');
```

4.3.3. Thiết kế các Stored Procedure hoặc Trigger

Trigger Trg_Insert_Khoahoc

```
CREATE OR REPLACE TRIGGER trg_insert_khoahoc BEFORE
   INSERT OR UPDATE ON khoahoc
   FOR EACH ROW
DECLARE
   v_count INT;
BEGIN
   SELECT
      COUNT(*)
   INTO v_count
   FROM
      khoahoc
   WHERE
      lower(tenkhoahoc) = lower(:new.tenkhoahoc)
      OR makhoahoc = :new.makhoahoc;

   IF v_count > 0 THEN
      raise_application_error(-20001, 'Khoá học đã tồn tại');
   END IF;
END;
/
```

(Xem thêm ở phụ lục)

4.3.4. Thiết kế và cài đặt các giao tác (Transaction)

4.3.4.1. Mô tả các tình huống cập nhật dữ liệu có thể gây rủi ro về tính nhất quán dữ liệu

Tình huống có thể gây rủi ro về tính nhất quán dữ liệu:

- Lúc thêm hoá đơn, có phần cập nhật số lượng sách nếu hoá đơn có sách. Cài đặt transaction read only để tránh việc số lượng sách sau khi cập nhật không nhất quán.

```
CREATE OR REPLACE PROCEDURE ThemHoaDon (
 p_MaHoaDon IN HoaDon.MaHoaDon%TYPE,
 p_SoTien IN HoaDon.SoTien%TYPE,
 p_MaHocVien IN HoaDon.MaHocVien%TYPE,
 p_ngay in Hoadon.ngay%type,
 p_MaNhanVien IN HoaDon.MaNhanVien%TYPE,
 p_ChiTietSach IN HoaDonSachList DEFAULT NULL,
 p_ChiTietKhoaHoc IN ChiTietHoaDonList DEFAULT NULL
```

```
) AS
 v_TongSoTienChiTietSach INT := 0;
 v_TongSoTienChiTietKhoaHoc INT := 0;
 v_TongSoTienHoaDon INT;
 v_temp int;
BEGIN
 set transaction isolation level serializable;
 FOR rec IN (SELECT * FROM TABLE(p_ChiTietSach)) LOOP
 -- Lấy mã sách từ list
 SELECT GiaSach into v_temp FROM Sach WHERE MaSach = rec.MaSach;
 v_TongSoTienChiTietSach := v_TongSoTienChiTietSach + rec.SoLuong *
v_temp;
 END LOOP;
 -- Tính tổng số tiền từ khoá học
 FOR rec IN (SELECT * FROM TABLE(p_ChiTietKhoaHoc)) LOOP
 -- Lấy mã khoá học từ list
 SELECT HocPhi into v_temp FROM KhoaHoc WHERE MaKhoaHoc =
rec.MaKhoaHoc;
 v_TongSoTienChiTietKhoaHoc := v_TongSoTienChiTietKhoaHoc + v_temp;
 END LOOP;
 -- Tính tổng số tiền của hóa đơn
 v_TongSoTienHoaDon := v_TongSoTienChiTietSach +
v_TongSoTienChiTietKhoaHoc;
 -- Kiểm tra xem tổng số tiền của hóa đơn có bằng số tiền nhập vào không
 IF p_SoTien != v_TongSoTienHoaDon THEN
 RAISE_APPLICATION_ERROR(-20001, 'Lỗi: Số tiền của hóa đơn không khớp
với tổng số tiền của các mục chi tiết.');
 END IF;
 -- Thêm thông tin vào bảng HoaDon
 INSERT INTO HoaDon (MaHoaDon, SoTien, MaHocVien, MaNhanVien,Ngay)
 VALUES (p_MaHoaDon, p_SoTien, p_MaHocVien, p_MaNhanVien,p_Ngay);
```

```
 -- Thêm chi tiết hóa đơn sách vào bảng HoaDonSach nếu có
 IF p_ChiTietSach IS NOT NULL THEN
 FOR i IN 1..p_ChiTietSach.COUNT LOOP
 INSERT INTO HoaDonSach (MaHoaDon, MaSach, SoLuong)
 VALUES (p_MaHoaDon, p_ChiTietSach(i).MaSach,
p_ChiTietSach(i).SoLuong);
 bansach(p_ChiTietSach(i).MaSach,p_chitietsach(i).soluong);
 END LOOP;
 END IF;
 -- Thêm chi tiết hóa đơn khoá học vào bảng ChiTietHoaDon nếu có
 IF p_ChiTietKhoaHoc IS NOT NULL THEN
 FOR i IN 1..p_ChiTietKhoaHoc.COUNT LOOP
 INSERT INTO ChiTietHoaDon (MaKhoaHoc, MaHoaDon)
 VALUES (p_ChiTietKhoaHoc(i).MaKhoaHoc, p_MaHoaDon);
 END LOOP;
 END IF;
 COMMIT;
 DBMS_OUTPUT.PUT_LINE('Hóa đơn đã được thêm thành công và số tiền đã được
kiểm tra.');
EXCEPTION
 WHEN OTHERS THEN
 -- Rollback transaction nếu có lỗi
 ROLLBACK;
 RAISE_APPLICATION_ERROR(-20002, 'Lỗi: ' || SQLERRM);
END ThemHoaDon;
/
-- lệnh test
DECLARE
 -- Khai báo các biến đầu vào
 v_MaHoaDon HoaDon.MaHoaDon%TYPE := 'HD001';
 v_SoTien HoaDon.SoTien%TYPE := 150000;
```

```
 v_MaHocVien HoaDon.MaHocVien%TYPE := 'HV001';
 v_MaNhanVien HoaDon.MaNhanVien%TYPE := 'NV001';
 v_Ngay Hoadon.ngay%type := sysdate;
 -- Khai báo danh sách chi tiết sách và chi tiết khoá học
 v_ChiTietSach HoaDonSachList := HoaDonSachList();
 v_ChiTietKhoaHoc ChiTietHoaDonList := ChiTietHoaDonList();
BEGIN
 -- Thêm một chi tiết sách vào danh sách chi tiết sách
 v_ChiTietSach.EXTEND;
 v_ChiTietSach(1) := HoaDonSachType(101, 2); -- Giả sử mã sách là 101 và số
lượng là 2
 -- Thêm một chi tiết khoá học vào danh sách chi tiết khoá học
 v_ChiTietKhoaHoc.EXTEND;
 v_ChiTietKhoaHoc(1) := ChiTietHoaDonType('KH001'); -- Giả sử mã khoá học là
'KH001'
 -- Gọi procedure ThemHoaDon với các tham số đã được khai báo
 ThemHoaDon(v_MaHoaDon, v_SoTien, v_MaHocVien, v_Ngay, v_MaNhanVien,
v_ChiTietSach, v_ChiTietKhoaHoc);

 -- In ra thông báo sau khi thực thi procedure thành công
 DBMS_OUTPUT.PUT_LINE('Procedure đã được thực thi thành công.');
EXCEPTION
 WHEN OTHERS THEN
 -- Xử lý nếu có lỗi xảy ra
 DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
END;
/
```

4.3.4.2. Mô tả ngăn ngừa bằng các transaction

- Lúc cập nhật hoá đơn, có phần xoá chi tiết hoá đơn cũ và thêm chi tiết hoá đơn mới. Cài đặt transaction read only để tránh việc số lượng sách sau khi cập nhật được nhất quán

```
CREATE OR REPLACE PROCEDURE CapNhatHoaDon (
   p_MaHoaDon IN HoaDon.MaHoaDon%TYPE,
   p_SoTien IN HoaDon.SoTien%TYPE,
   p_MaHocVien IN HoaDon.MaHocVien%TYPE,
   p_MaNhanVien IN HoaDon.MaNhanVien%TYPE,
   p_ChiTietSach IN HoaDonSachList DEFAULT NULL,
   p_ChiTietKhoaHoc IN ChiTietHoaDonList DEFAULT NULL
) AS
   v_TongSoTienChiTietSach INT := 0;
   v_TongSoTienChiTietKhoaHoc INT := 0;
   v_TongSoTienHoaDon INT;
   v_temp int;
BEGIN
   set transaction isolation level serializable;
   -- Tính tổng số tiền từ sách
   FOR rec IN (SELECT * FROM TABLE(p_ChiTietSach)) LOOP
      -- Lấy mã sách từ list
      SELECT GiaSach into v_temp FROM Sach WHERE MaSach = rec.MaSach;
      v_TongSoTienChiTietSach := v_TongSoTienChiTietSach + rec.SoLuong * v_temp;
   END LOOP;

   -- Tính tổng số tiền từ khoá học
   FOR rec IN (SELECT * FROM TABLE(p_ChiTietKhoaHoc)) LOOP
      -- Lấy mã khoá học từ list
      SELECT HocPhi into v_temp FROM KhoaHoc WHERE MaKhoaHoc =
rec.MaKhoaHoc;
      v_TongSoTienChiTietKhoaHoc := v_TongSoTienChiTietKhoaHoc + v_temp;
   END LOOP;

   -- Tính tổng số tiền của hóa đơn
   v_TongSoTienHoaDon := v_TongSoTienChiTietSach +
v_TongSoTienChiTietKhoaHoc;

   -- Kiểm tra xem tổng số tiền của hóa đơn có bằng số tiền nhập vào không
   IF p_SoTien != v_TongSoTienHoaDon THEN
      RAISE_APPLICATION_ERROR(-20001, 'Lỗi: Số tiền của hóa đơn không khớp
với tổng số tiền của các mục chi tiết.');
   END IF;

   -- Cập nhật thông tin vào bảng HoaDon
   UPDATE HoaDon
   SET SoTien = p_SoTien,
      MaHocVien = p_MaHocVien,
      MaNhanVien = p_MaNhanVien
   WHERE MaHoaDon = p_MaHoaDon;
```

```
    for sach_rec in (select * from hoadonsach where mahoadon=p_MaHoaDon)
    loop
        HoanSach(sach_rec.masach,sach_rec.soluong);
    end loop;


    -- Xóa các chi tiết cũ của hóa đơn
    DELETE FROM HoaDonSach WHERE MaHoaDon = p_MaHoaDon;
    DELETE FROM ChiTietHoaDon WHERE MaHoaDon = p_MaHoaDon;

    -- Thêm chi tiết hóa đơn sách vào bảng HoaDonSach nếu có
    IF p_ChiTietSach IS NOT NULL THEN
        FOR i IN 1..p_ChiTietSach.count LOOP
            INSERT INTO HoaDonSach (MaHoaDon, MaSach, SoLuong)
            VALUES (p_MaHoaDon, p_ChiTietSach(i).MaSach,
p_ChiTietSach(i).SoLuong);
            bansach(p_ChiTietSach(i).MaSach, p_ChiTietSach(i).SoLuong);
        END LOOP;
    END IF;

    -- Thêm chi tiết hóa đơn khoá học vào bảng ChiTietHoaDon nếu có
    IF p_ChiTietKhoaHoc IS NOT NULL THEN
        FOR i IN 1..p_ChiTietKhoaHoc.count LOOP
            INSERT INTO ChiTietHoaDon (MaKhoaHoc, MaHoaDon)
            VALUES (p_ChiTietKhoaHoc(i).MaKhoaHoc, p_MaHoaDon);
        END LOOP;
    END IF;

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Hóa đơn đã được cập nhật thành công và số tiền đã
được kiểm tra.');
EXCEPTION
    WHEN OTHERS THEN
        -- Rollback transaction nếu có lỗi
        ROLLBACK;
        RAISE_APPLICATION_ERROR(-20002, 'Lỗi: ' || SQLERRM);
END CapNhatHoaDon;
/
--test
DECLARE
    -- Khai báo các biến đầu vào
    v_MaHoaDon HoaDon.MaHoaDon%TYPE := 'HD001';
    v_SoTien HoaDon.SoTien%TYPE := 10600000;
    v_MaHocVien HoaDon.MaHocVien%TYPE := 'HV002';
    v_MaNhanVien HoaDon.MaNhanVien%TYPE := 'NV002';
```

```
   -- Khai báo danh sách chi tiết sách và chi tiết khoá học
   v_ChiTietSach HoaDonSachList := HoaDonSachList();
   v_ChiTietKhoaHoc ChiTietHoaDonList := ChiTietHoaDonList();

BEGIN
   -- Thêm một chi tiết sách vào danh sách chi tiết sách
   v_ChiTietSach.EXTEND;
   v_ChiTietSach(1) := HoaDonSachType(2, 2); -- Giả sử mã sách là 102 và số lượng là
3
   -- Thêm một chi tiết khoá học vào danh sách chi tiết khoá học
   v_ChiTietKhoaHoc.EXTEND;
   v_ChiTietKhoaHoc(1) := ChiTietHoaDonType('KH002'); -- Giả sử mã khoá học là
'KH002'

   -- Gọi procedure CapNhatHoaDon với các tham số đã được khai báo
   CapNhatHoaDon(v_MaHoaDon, v_SoTien, v_MaHocVien, v_MaNhanVien,
v_ChiTietSach, v_ChiTietKhoaHoc);

   -- In ra thông báo sau khi thực thi procedure thành công
   DBMS_OUTPUT.PUT_LINE('Procedure đã được thực thi thành công.');
EXCEPTION
   WHEN OTHERS THEN
      -- Xử lý nếu có lỗi xảy ra
      DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
END;
```

4.3.5. Minh họa vấn đề xử lý đồng thời

- Mô tả các tình huống tranh chấp dữ liệu có thể xảy ra giữa các thao tác

SL = 30	T1	T2
T1	Read (SL)	
T2		Read (SL)
T3	SL := SL+10	
T4	Write (SL)	
T5		SL := SL - 10
T6		Write (SL)
	SL = 40	SL = 30

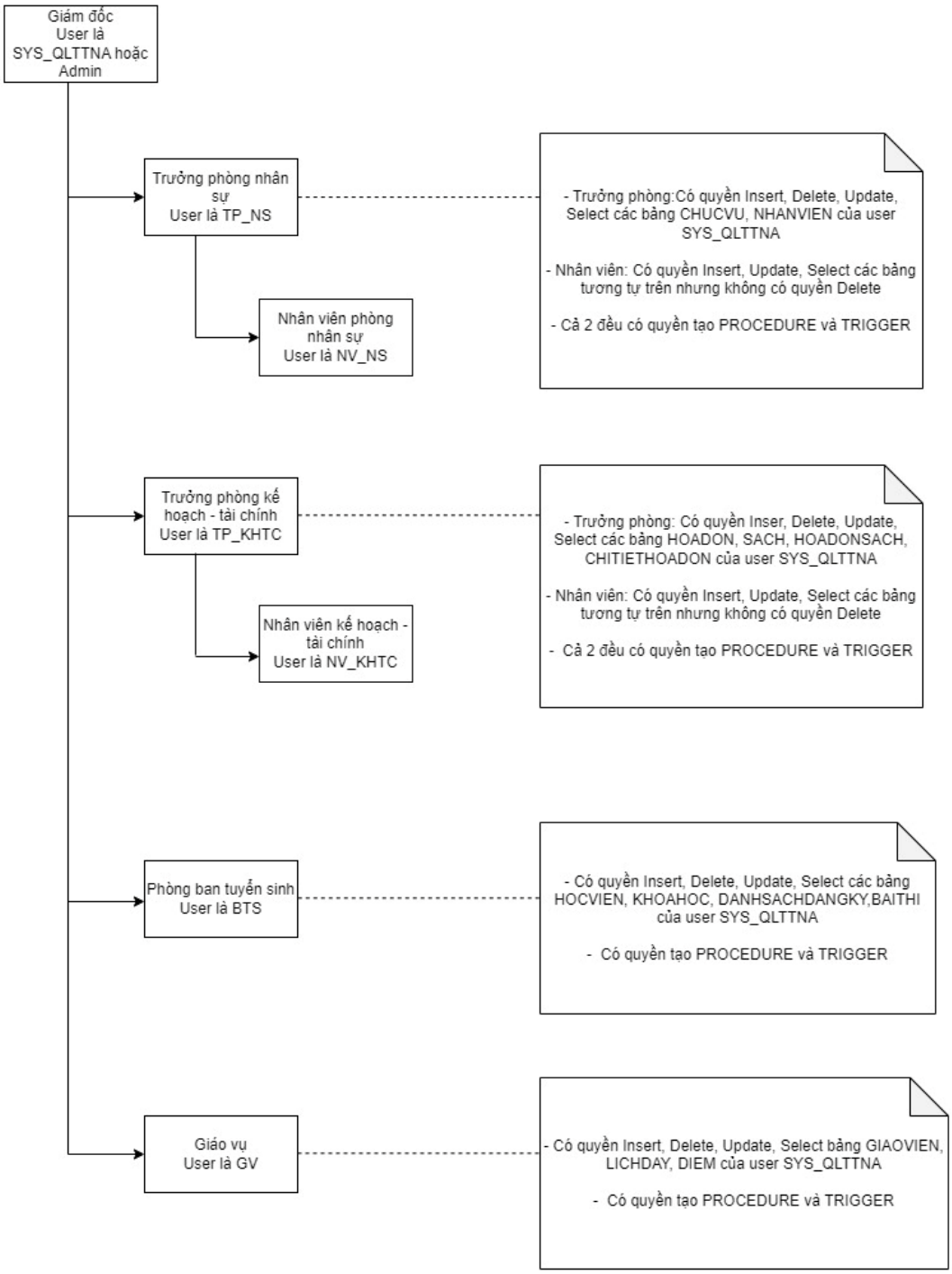
- Cài đặt mô phỏng các tình huống này (tại Server, hoặc từ phía Client)

Session 1	Session 2
Select * From sach where masach = 1;	

	Select * from sach where masach=1;
Update sach set soluong=soluong+10 where masach=1;	
	Update sach set soluong=soluong-10;
	Commit;
Commit;	

4.3.6. Thiết kế và cài đặt mô hình phân quyền của hệ thống

- Thiết kế phân quyền cụ thể cho user, áp dụng dụng các profile , …




```
Alter session set "_oracle_script" = true;
create user SYS_QLTTNA identified by 1;
grant create session to SYS_QLTTNA;
grant create table to SYS_QLTTNA;
Alter user SYS_QLTTNA quota 10M on users ;
```

```
-- Cấp quyền viết trigger
GRANT CREATE TRIGGER TO SYS_QLTTNA;
GRANT CREATE PROCEDURE TO SYS_QLTTNA;
--
GRANT UNLIMITED TABLESPACE TO SYS_QLTTNA;
GRANT EXECUTE ON DBMS_LOCK TO SYS_QLTTNA;
-- Cấp này mới tạo được bảng nhật ký có id tự động
GRANT CREATE SEQUENCE TO SYS_QLTTNA;
-- DROP USER SYS_QLNS CASCADE;

---Cấp quyền cho user TP_NS
CREATE USER TP_NS  IDENTIFIED BY 1;
grant create session to TP_NS ;
Alter user TP_NS  quota 10M on users ;
-- Cấp quyền thêm, sửa, xóa, xem trên các bảng
GRANT INSERT, UPDATE, DELETE, SELECT ON SYS_QLTTNA.CHUCVU TO
TP_NS ;
GRANT INSERT, UPDATE, DELETE, SELECT ON SYS_QLTTNA.NHANVIEN TO
TP_NS ;
--
GRANT CREATE TRIGGER TO TP_NS;
GRANT CREATE PROCEDURE TO TP_NS;

-- Cấp quyền cho user NV_NS (Phòng nhân sự):
-- Tạo user NV_NS và cài đặt mật khẩu
CREATE USER NV_NS IDENTIFIED BY 1;
grant create session to NV_PKT;
Alter user NV_NS quota 10M on users ;
-- Cấp quyền thêm, sửa, xóa, xem trên các bảng
GRANT INSERT, UPDATE,  SELECT ON  SYS_QLTTNA.CHUCVU TO NV_NS;
GRANT INSERT, UPDATE,  SELECT ON SYS_QLTTNA.NHANVIEN TO NV_NS;
```

```
GRANT CREATE TRIGGER TO NV_NS;
GRANT CREATE PROCEDURE TO NV_NS;

--Cấp quyền cho user TP_KHTC
CREATE USER TP_KHTC IDENTIFIED BY 1;
grant create session to TP_KHTC;
Alter user TP_KHTC quota 10M on users ;
-- Cấp quyền thêm, sửa, xóa, xem trên các bảng
GRANT INSERT, UPDATE, DELETE, SELECT ON SYS_QLTTNA.HOADON TO
TP_KHTC;
GRANT INSERT, UPDATE, DELETE,SELECT ON SYS_QLTTNA.SACH TO
TP_KHTC;
GRANT INSERT, UPDATE, DELETE,SELECT ON SYS_QLTTNA.HOADONSACH
TO TP_KHTC;
GRANT INSERT, UPDATE, DELETE,SELECT ON
SYS_QLTTNA.CHITIETHOADON TO TP_KHTC;

--
GRANT CREATE TRIGGER TO TP_KHTC;
GRANT CREATE PROCEDURE TO TP_KHTC;

-- Cấp quyền cho user NV_KHTC (Phòng kế hoạch - tài chính):
-- Tạo user NV_KHTC và cài đặt mật khẩu
CREATE USER NV_KHTC IDENTIFIED BY 1;
grant create session to NV_KHTC;
Alter user NV_KHTC quota 10M on users ;
-- Cấp quyền thêm, sửa, xóa, xem trên các bảng
GRANT INSERT, UPDATE, SELECT ON SYS_QLTTNA.HOADON TO NV_KHTC;
GRANT INSERT, UPDATE, SELECT ON SYS_QLTTNA.SACH TO NV_KHTC;
GRANT INSERT, UPDATE, SELECT ON SYS_QLTTNA.HOADONSACH TO
```

```
NV_KHTC;
GRANT INSERT, UPDATE, SELECT ON SYS_QLTTNA.CHITIETHOADON TO NV_KHTC;
--
GRANT CREATE TRIGGER TO NV_KHTC;
GRANT CREATE PROCEDURE TO NV_KHTC;
-- Cấp quyền cho user BTS (Ban tuyển sinh)
CREATE USER BTS IDENTIFIED BY 1;
grant create session to BTS;
Alter user BTS quota 10M on users ;
-- Cấp quyền thêm, sửa, xóa, xem trên các bảng
GRANT INSERT, UPDATE, DELETE, SELECT ON SYS_QLTTNA.HOCVIEN TO BTS;
GRANT INSERT, UPDATE, DELETE,SELECT ON SYS_QLTTNA.KHOAHOC TO BTS;
GRANT INSERT, UPDATE, DELETE,SELECT ON SYS_QLTTNA.DANHSACHDANGKY TO BTS;
GRANT INSERT, UPDATE, DELETE,SELECT ON SYS_QLTTNA.BAITHI TO BTS;
GRANT CREATE TRIGGER TO BTS;
GRANT CREATE PROCEDURE TO BTS;
-- 4Cấp quyền cho user GV (Giáo vụ ):
-- Tạo user GV và cài đặt mật khẩu
CREATE USER GV IDENTIFIED BY 1;
grant create session to GV;
Alter user GV quota 10M on users ;
-- Cấp quyền thêm, sửa, xóa, xem trên các bảng
GRANT INSERT, UPDATE, SELECT ON SYS_QLTTNA.GIAOVIEN TO GV;
GRANT INSERT, UPDATE, SELECT ON SYS_QLTTNA.LICH TO GV;
GRANT INSERT, UPDATE, SELECT ON SYS_QLTTNA.DIEM TO GV;
--
GRANT CREATE TRIGGER TO GV;
```

GRANT CREATE PROCEDURE TO GV;

CHƯƠNG 5: KẾT LUẬN

5.1. Kết quả đạt được và chưa đạt được trong đề tài

Đã thực hiện được các bước phân tích, thiết kế và cài đặt Oracle, cụ thể như sau:

- Bước phân tích đã tạo ra được sơ đồ ERD để biểu diễn các dữ liệu cần lưu trữ, đáp ứng cho các chức năng của hệ thống

- Bước thiết kế mức quan niệm đã tạo ra được Lược đồ CSDL quan hệ đạt dạng chuẩn 3, đảm bảo lưu trữ dữ liệu hiệu quả

- Bước thiết kế mức Logic đã rà soát các phép kết thường xuyên được sử dụng (và bổ sung một số Index) để tăng tốc độ truy xuất

- Bước thiết kế mức Vật lý: đề tài đã chọn cài đặt CSDL trên Hệ quản trị CSDL Oracle

- Các bước cài đặt trên SQL

+ Đã tạo được 13 table

+ Đã vẽ được 1 sơ đồ ERD

+ Đã vẽ được 1 lược đồ quan hệ

+ Đã cài được 9 trigger cho RBTV

+ Đã cài được 38 procedure để đáp ứng dữ liệu cho các chức năng của hệ thống

- Những vấn đề chưa thực hiện

+ Mở khóa học 1 kèm 1

+ Có thể chuyển đổi sang các hệ quản trị CSDL khác

5.2. Hướng phát triển mở rộng ứng dụng trong tương lai

+ Mở thêm nhiều khóa học cho học viên

+ Mở thêm nhiều chi nhánh

Tài liệu tham khảo

PHỤ LỤC

1. PROCEDURE

PROCEDURE 1: Cập nhật khoá học

```
CREATE OR REPLACE PROCEDURE capnhatkhoahoc (
   p_makhoahoc  IN khoahoc.makhoahoc%TYPE,
   p_hocphi     IN khoahoc.hocphi%TYPE DEFAULT NULL,
   p_ngaybatdau IN khoahoc.ngaybatdau%TYPE DEFAULT NULL,
   p_tenkhoahoc IN khoahoc.tenkhoahoc%TYPE DEFAULT NULL,
   p_soluong    IN khoahoc.soluong%TYPE DEFAULT NULL,
   p_sobuoi     IN khoahoc.sobuoi%TYPE DEFAULT NULL,
   p_trangthai  IN khoahoc.trangthai%TYPE DEFAULT NULL,
   p_tenloai    IN loaikhoahoc.tenloai%TYPE
) AS
   v_loaikhoahoc loaikhoahoc.tenloai%TYPE;
BEGIN
   SELECT
      tenloai
   INTO v_loaikhoahoc
   FROM
      loaikhoahoc
   WHERE
      lower(tenloai) = lower(p_tenloai);
   UPDATE khoahoc
   SET
      hocphi = nvl(p_hocphi, hocphi),
      ngaybatdau = nvl(p_ngaybatdau, ngaybatdau),
      tenkhoahoc = nvl(p_tenkhoahoc, tenkhoahoc),
      soluong = nvl(p_soluong, soluong),
      sobuoi = nvl(p_sobuoi, sobuoi),
      trangthai = nvl(p_trangthai, trangthai),
      loai = nvl(v_loaikhoahoc, loai)
   WHERE
      makhoahoc = p_makhoahoc;
   COMMIT;
   dbms_output.put_line('Thông tin khóa học được cập nhật thành công.');
END;
```

PROCEDURE 2: Xoá khoá học

```
CREATE OR REPLACE PROCEDURE xoakhoahoc (
   p_makhoahoc IN khoahoc.makhoahoc%TYPE
) AS
BEGIN
   DELETE FROM khoahoc
   WHERE
```

```
        makhoahoc = p_makhoahoc;
    COMMIT;
    dbms_output.put_line('Khóa học được xóa thành công.');
EXCEPTION
    WHEN no_data_found THEN
        raise_application_error(-20002, 'Không tìm thấy khóa học với Mã Khoa Học đã
nhập.');
END;
```

PROCEDURE 3: Lấy thông tin khoá học

```
CREATE OR REPLACE PROCEDURE laythongtinkhoahoc (
    p_makhoahoc IN khoahoc.makhoahoc%TYPE DEFAULT NULL
) AS
    CURSOR khoahoccursor IS
    SELECT
        makhoahoc,
        hocphi,
        ngaybatdau,
        tenkhoahoc,
        soluong,
        sobuoi,
        trangthai,
        tenloai
    FROM
            khoahoc
        JOIN loaikhoahoc ON khoahoc.loai = loaikhoahoc.maloai
    WHERE
        makhoahoc = nvl(p_makhoahoc, makhoahoc)
        OR p_makhoahoc IS NULL;
BEGIN
    FOR rec IN khoahoccursor LOOP
        dbms_output.put_line('Thông tin khóa học:');
        dbms_output.put_line('Mã Khoa Học: '
                        || nvl(rec.makhoahoc, 'NULL'));
        dbms_output.put_line('Học Phí: '
                        || nvl(to_char(rec.hocphi), 'NULL'));

        dbms_output.put_line('Ngày Bắt Đầu: '
                        || nvl(to_char(rec.ngaybatdau, 'DD-MON-YYYY'), 'NULL'));

        dbms_output.put_line('Tên Khóa Học: '
                        || nvl(rec.tenkhoahoc, 'NULL'));
        dbms_output.put_line('Số Lượng: '
                        || nvl(to_char(rec.soluong), 'NULL'));

        dbms_output.put_line('Số Buổi: '
```

```
                    || nvl(to_char(rec.sobuoi), 'NULL'));

      dbms_output.put_line('Trạng Thái: '
                    || nvl(rec.trangthai, 'NULL'));
      dbms_output.put_line('Loại khoá học: '
                    || nvl(rec.tenloai, 'NULL'));
      dbms_output.put_line('------------------------');
   END LOOP;
END;
/
exec LayThongTinKhoaHoc('KH001');
exec LayThongTinKhoaHoc;
```

PROCEDURE 4: Cập nhật thông tin học viên

```
CREATE OR REPLACE PROCEDURE CapNhatThongTinHocVien (
   p_MaHocVien IN HocVien.MaHocVien%TYPE,
   p_TenHocVien IN HocVien.TenHocVien%TYPE DEFAULT NULL,
   p_NgaySinh IN HocVien.NgaySinh%TYPE DEFAULT NULL,
   p_SoDienThoai IN HocVien.SoDienThoai%TYPE DEFAULT NULL,
   p_Email IN HocVien.Email%TYPE DEFAULT NULL,
   p_DiaChi IN HocVien.DiaChi%TYPE DEFAULT NULL,
   p_GioiTinh IN HocVien.GioiTinh%TYPE DEFAULT NULL
) AS
BEGIN
   IF p_NgaySinh >= SYSDATE THEN
      DBMS_OUTPUT.PUT_LINE('Lỗi: Ngày sinh không hợp lệ.');
      RETURN;
   END IF;
   UPDATE HocVien
   SET
      TenHocVien = NVL(p_TenHocVien, TenHocVien),
      NgaySinh = NVL(p_NgaySinh, NgaySinh),
      SoDienThoai = NVL(p_SoDienThoai, SoDienThoai),
      Email = NVL(p_Email, Email),
      DiaChi = NVL(p_DiaChi, DiaChi),
      GioiTinh = NVL(p_GioiTinh, GioiTinh)
   WHERE MaHocVien = p_MaHocVien;
   IF SQL%ROWCOUNT = 0 THEN
      DBMS_OUTPUT.PUT_LINE('Không tìm thấy học viên với Mã Học Viên đã
nhập.');
   ELSE
      COMMIT;
      DBMS_OUTPUT.PUT_LINE('Thông tin học viên đã được cập nhật thành công.');
   END IF;
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
```

```
END CapNhatThongTinHocVien;
```

PROCEDURE 5: Lấy thông tin học viên

```
CREATE OR REPLACE PROCEDURE LayThongTinHocVien (
   p_MaHocVien IN HocVien.MaHocVien%TYPE DEFAULT NULL
) AS
BEGIN
   FOR rec IN (SELECT *
            FROM HocVien
            WHERE MaHocVien = NVL(p_MaHocVien, MaHocVien))
   LOOP
      DBMS_OUTPUT.PUT_LINE('Thông tin học viên:');
      DBMS_OUTPUT.PUT_LINE('Mã Học Viên: ' || NVL(rec.MaHocVien, 'NULL'));
      DBMS_OUTPUT.PUT_LINE('Tên Học Viên: ' || NVL(rec.TenHocVien, 'NULL'));
      DBMS_OUTPUT.PUT_LINE('Ngày Sinh: ' || NVL(TO_CHAR(rec.NgaySinh, 'DD-
MON-YYYY'), 'NULL'));
      DBMS_OUTPUT.PUT_LINE('Số Điện Thoại: ' || NVL(rec.SoDienThoai, 'NULL'));
      DBMS_OUTPUT.PUT_LINE('Email: ' || NVL(rec.Email, 'NULL'));
      DBMS_OUTPUT.PUT_LINE('Địa Chỉ: ' || NVL(rec.DiaChi, 'NULL'));
      DBMS_OUTPUT.PUT_LINE('Giới Tính: ' || NVL(rec.GioiTinh, 'NULL'));
      DBMS_OUTPUT.PUT_LINE('------------------------');
   END LOOP;
   IF SQL%NOTFOUND THEN
      DBMS_OUTPUT.PUT_LINE('Không tìm thấy học viên với Mã Học Viên đã
nhập.');
   END IF;
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
END LayThongTinHocVien;
/
exec Laythongtinhocvien('HV001');
exec Laythongtinhocvien;
```

PROCEDURE 6: Thêm nhân viên

```
CREATE OR REPLACE PROCEDURE themnhanvien (
   p_manhanvien  IN nhanvien.manhanvien%TYPE,
   p_tennhanvien IN nhanvien.tennhanvien%TYPE,
   p_sodienthoai IN nhanvien.sodienthoai%TYPE,
   p_gioitinh    IN nhanvien.gioitinh%TYPE,
   p_diachi      IN nhanvien.diachi%TYPE,
   p_ngaysinh    IN nhanvien.ngaysinh%TYPE,
   p_chucvu      IN nhanvien.chucvu%TYPE,
   p_luong nhanvien.luong%type
) AS
BEGIN
```

```
    INSERT INTO nhanvien (
        manhanvien,
        tennhanvien,
        sodienthoai,
        gioitinh,
        diachi,
        ngaysinh,
        chucvu,
        luong
    ) VALUES (
        p_manhanvien,
        p_tennhanvien,
        p_sodienthoai,
        p_gioitinh,
        p_diachi,
        p_ngaysinh,
        p_chucvu,
        p_luong
    );
    dbms_output.put_line('Nhân viên đã được thêm vào thành công.');
END;
```

PROCEDURE 7: Cập nhật thông tin nhân viên

```
CREATE OR REPLACE PROCEDURE capnhatthongtinnhanvien (
    p_manhanvien  IN nhanvien.manhanvien%TYPE,
    p_tennhanvien IN nhanvien.tennhanvien%TYPE DEFAULT NULL,
    p_sodienthoai IN nhanvien.sodienthoai%TYPE DEFAULT NULL,
    p_gioitinh    IN nhanvien.gioitinh%TYPE DEFAULT NULL,
    p_diachi      IN nhanvien.diachi%TYPE DEFAULT NULL,
    p_ngaysinh    IN nhanvien.ngaysinh%TYPE DEFAULT NULL,
    p_chucvu      IN nhanvien.chucvu%TYPE DEFAULT NULL,
    p_luong nhanvien.luong%type default null
) AS
BEGIN
    UPDATE nhanvien
    SET
        tennhanvien = nvl(p_tennhanvien, tennhanvien),
        sodienthoai = nvl(p_sodienthoai, sodienthoai),
        gioitinh = nvl(p_gioitinh, gioitinh),
        diachi = nvl(p_diachi, diachi),
        ngaysinh = nvl(p_ngaysinh, ngaysinh),
        chucvu = nvl(p_chucvu, chucvu),
        luong = nvl(p_luong,luong)
    WHERE
        manhanvien = p_manhanvien;
END;
```

PROCEDURE 8: Xoá nhân viên

```
CREATE OR REPLACE PROCEDURE xoanhanvien (
   p_manhanvien IN nhanvien.manhanvien%TYPE
) AS
BEGIN
   DELETE FROM nhanvien
   WHERE
      manhanvien = p_manhanvien;
   IF SQL%rowcount = 0 THEN
      dbms_output.put_line('Không tìm thấy nhân viên với Mã Nhân Viên đã nhập.');
   ELSE
      COMMIT;
      dbms_output.put_line('Nhân viên đã được xóa thành công.');
   END IF;
EXCEPTION
   WHEN OTHERS THEN
      ROLLBACK;
      dbms_output.put_line('Lỗi: ' || sqlerrm);
END;
```

PROCEDURE 9: Lấy thông tin nhân viên

```
CREATE OR REPLACE PROCEDURE laythongtinnhanvien (
   p_manhanvien IN nhanvien.manhanvien%TYPE DEFAULT NULL
) AS
   CURSOR nhanviencursor IS
   SELECT
      nhanvien.*,chucvu.tenchucvu
   FROM
         nhanvien
      JOIN chucvu ON nhanvien.chucvu = chucvu.machucvu
   WHERE
      manhanvien = nvl(p_manhanvien, manhanvien);
BEGIN
   FOR rec IN nhanviencursor LOOP
      dbms_output.put_line('Thông tin nhân viên:');
      dbms_output.put_line('Mã Nhân Viên: '
                     || nvl(rec.manhanvien, 'NULL'));
      dbms_output.put_line('Tên Nhân Viên: '
                     || nvl(rec.tennhanvien, 'NULL'));
      dbms_output.put_line('Số Điện Thoại: '
                     || nvl(rec.sodienthoai, 'NULL'));
      dbms_output.put_line('Giới Tính: '
                     || nvl(rec.gioitinh, 'NULL'));
      dbms_output.put_line('Địa Chỉ: '
```

```
                    || nvl(rec.diachi, 'NULL'));
      dbms_output.put_line('Ngày Sinh: '
                    || nvl(to_char(rec.ngaysinh, 'YYYY-MM-DD'), 'NULL'));

      dbms_output.put_line('Chức Vụ: '
                    || nvl(rec.tenchucvu, 'NULL'));
      dbms_output.put_line('Lương: '
                    || rec.luong);

      dbms_output.put_line('------------------------');
   END LOOP;

   IF SQL%notfound THEN
      dbms_output.put_line('Không tìm thấy nhân viên với Mã Nhân Viên đã nhập.');
   END IF;
END;
```

/**PROCEDURE 10: Cập nhật lương nhân viên**

```
CREATE OR REPLACE PROCEDURE capnhatluongnhanvien(
   p_manhanvien nhanvien.manhanvien%type,
   p_luong nhanvien.luong%type)
as
begin
   update nhanvien
   set luong=p_luong
   where manhanvien=p_manhanvien;
   if sql%rowcount >0 then
      dbms_output.put_line('Cập nhật lương nhân viên thành công');
      commit;
   end if;
end;
```

PROCEDURE 11:Thay đổi chức vụ nhân viên

```
CREATE OR REPLACE PROCEDURE thaydoichucvunhanvien (
   p_manhanvien nhanvien.manhanvien%TYPE,
   p_chucvu     chucvu.tenchucvu%TYPE
) AS
   v_machucvu chucvu.machucvu%TYPE;
   v_count    INT;
BEGIN
   SELECT
      COUNT(*)
   INTO v_count
   FROM
      chucvu
   WHERE
```

```
        lower(tenchucvu) = lower(p_chucvu);
    IF v_count = 0 THEN
        raise_application_error(-20010, 'Chức vụ không tồn tại');
    END IF;
    SELECT
        machucvu
    INTO v_machucvu
    FROM
        chucvu
    WHERE
        lower(tenchucvu) = lower(p_chucvu);
    UPDATE nhanvien
    SET
        chucvu = v_machucvu
    WHERE
        manhanvien = p_manhanvien;
    IF SQL%rowcount > 0 THEN
        dbms_output.put_line('Cập nhật thành công');
    ELSE
        dbms_output.put_line('Không có nhân viên này');
    END IF;
END;
/
--test đúng
exec ThayDoiChucVuNhanVien('NV001','Thu ngân');
--test sai
exec ThayDoiChucVuNhanVien('NV001','giamdoc');
```

PROCEDURE 12: Xem lịch dạy

```
CREATE OR REPLACE PROCEDURE xemlichday (
    p_magiaovien giaovien.magiaovien%TYPE default null
) AS
    CURSOR lichday_cursor IS
    SELECT
        lichday.*,
        khoahoc.tenkhoahoc,
        giaovien.tengiaovien
    FROM
            lichday
        JOIN khoahoc ON lichday.makhoahoc = khoahoc.makhoahoc
        JOIN giaovien ON giaovien.magiaovien = lichday.magiaovien
    WHERE
        lichday.magiaovien = nvl(p_magiaovien, lichday.magiaovien);
BEGIN
    FOR rec IN lichday_cursor LOOP
```

```
        dbms_output.put_line(rec.tengiaovien
                        || ' '
                        || rec.tenkhoahoc
                        || ' '
                        || to_char(rec.thoigian, 'dd-mm-yyyy HH24:MI:SS'));
    END LOOP;
END;
exec XemLichDay;
exec XemLichDay('GV001');
exec XemLichDay('dasdas');
```

PROCEDURE 13: Cập nhật trạng thái

```
CREATE OR REPLACE PROCEDURE CapNhatTrangThai (
    p_MaHocVien IN DanhSachDangKy.MaHocVien%TYPE,
    p_MaKhoaHoc IN DanhSachDangKy.MaKhoaHoc%TYPE,
    p_TrangThai IN DanhSachDangKy.TrangThai%TYPE
) AS
BEGIN
    UPDATE DanhSachDangKy
    SET TrangThai = p_TrangThai
    WHERE MaHocVien = p_MaHocVien AND MaKhoaHoc = p_MaKhoaHoc;
    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Không tìm thấy đăng ký của học viên với Mã Học
Viên và Mã Khóa Học đã nhập.');
    ELSE
        COMMIT;
        DBMS_OUTPUT.PUT_LINE('Trạng thái đăng ký đã được cập nhật thành công.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
END CapNhatTrangThai;
```

PROCEDURE 14: Xoá đăng ký

```
CREATE OR REPLACE PROCEDURE XoaDangKy (
    p_MaHocVien IN DanhSachDangKy.MaHocVien%TYPE,
    p_MaKhoaHoc IN DanhSachDangKy.MaKhoaHoc%TYPE,
    p_NgayDangKy IN danhsachdangky.ngay%type
) AS
BEGIN
    DELETE FROM DanhSachDangKy
    WHERE MaHocVien = p_MaHocVien
    AND MaKhoaHoc = p_MaKhoaHoc
    and ngay=p_ngaydangky
    and trangthai='Chưa thanh toán';
```

```
   IF SQL%ROWCOUNT = 0 THEN
      DBMS_OUTPUT.PUT_LINE('Không tìm thấy đăng ký của học viên với Mã Học
Viên và Mã Khóa Học đã nhập hoặc học viên đã đóng học phí và không thể xoá đăng
ký.');
   ELSE
      COMMIT;
      DBMS_OUTPUT.PUT_LINE('Đăng ký của học viên đã được xóa thành công.');
   END IF;
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
END XoaDangKy;
```

PROCEDURE 15: Lấy thông tin đăng ký

```
CREATE OR REPLACE PROCEDURE LayThongTinDangKy (
   p_MaHocVien IN DanhSachDangKy.MaHocVien%TYPE DEFAULT NULL,
   p_MaKhoaHoc IN DanhSachDangKy.MaKhoaHoc%TYPE DEFAULT NULL
) AS
BEGIN
   FOR rec IN (SELECT danhsachdangky.MaHocVien,TenHocVien,
danhsachdangky.MaKhoaHoc,tenkhoahoc, Ngay,danhsachdangky.TrangThai
            FROM DanhSachDangKy
            join hocvien on hocvien.mahocvien=danhsachdangky.mahocvien
            join khoahoc on khoahoc.makhoahoc=danhsachdangky.makhoahoc
            WHERE (danhsachdangky.MaHocVien = NVL(p_MaHocVien,
danhsachdangky.MaHocVien) OR p_MaHocVien IS NULL)
            AND (danhsachdangky.MaKhoaHoc = NVL(p_MaKhoaHoc,
danhsachdangky.MaKhoaHoc) OR p_MaKhoaHoc IS NULL))
   LOOP
      DBMS_OUTPUT.PUT_LINE('Thông tin đăng ký:');
      DBMS_OUTPUT.PUT_LINE('Mã Học Viên: ' || rec.MaHocVien);
      DBMS_OUTPUT.PUT_LINE('Tên Học Viên: ' || rec.TenHocVien);
      DBMS_OUTPUT.PUT_LINE('Mã Khóa Học: ' || rec.MaKhoaHoc);
      DBMS_OUTPUT.PUT_LINE('Tên Khoá học: ' || rec.tenkhoahoc);
      DBMS_OUTPUT.PUT_LINE('Ngày: ' || TO_CHAR(rec.Ngay, 'DD-MON-
YYYY'));
      DBMS_OUTPUT.PUT_LINE('Trạng Thái: ' || rec.TrangThai);
      DBMS_OUTPUT.PUT_LINE('------------------------');
   END LOOP;
   IF SQL%NOTFOUND THEN
      DBMS_OUTPUT.PUT_LINE('Không tìm thấy thông tin đăng ký.');
   END IF;
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
```

```
END LayThongTinDangKy;
exec laythongtindangky(null,'KH001')
exec laythongtindangky('hv002')
```

PROCEDURE 16: Thêm chức vụ

```
CREATE OR REPLACE PROCEDURE ThemChucVu (
   p_TenChucVu IN ChucVu.TenChucVu%TYPE
) AS
   v_count int :=0;
BEGIN
   INSERT INTO ChucVu (TenChucVu)
   VALUES (p_TenChucVu);
   DBMS_OUTPUT.PUT_LINE('ChucVu được thêm thành công.');
END;
```

PROCEDURE 17:Cập nhật chức vụ

```
CREATE OR REPLACE PROCEDURE CapNhatChucVu (
   p_MaChucVu IN ChucVu.MaChucVu%TYPE,
   p_TenChucVu IN ChucVu.TenChucVu%TYPE
) AS
   v_count int :=0;
BEGIN
   UPDATE ChucVu
   SET TenChucVu = p_TenChucVu
   WHERE MaChucVu =  p_MaChucVu;
END;
```

PROCEDURE 18: Lấy thông tin chức vụ

```
CREATE OR REPLACE PROCEDURE LayThongTinChucVu (
   p_MaChucVu IN ChucVu.MaChucVu%TYPE default null
) AS
   cursor cursor_chucvu is
   select * from chucvu
   where machucvu=nvl(p_MaChucVu,MaChucVu);
BEGIN
   -- Kiểm tra điều kiện và báo lỗi nếu cần
   FOR rec IN cursor_chucvu LOOP
      DBMS_OUTPUT.PUT_LINE('Mã ChucVu: ' || rec.MaChucVu || ', Tên ChucVu: ' ||
rec.TenChucVu);
   END LOOP;
END;
exec LayThongTinChucVu(1);
exec LayThongTinChucVu;
```

PROCEDURE 19: Thêm bài thi

```
CREATE OR REPLACE PROCEDURE ThemBaiThi(
   p_TenBaiThi baithi.tenbaithi%type,
   p_ThoiGian  baithi.thoigian%type,
   p_MaKhoaHoc baithi.makhoahoc%type)
as
begin
   insert into BaiThi(tenbaithi,thoigian,makhoahoc)
   values(p_TenBaiThi,p_ThoiGian,p_MaKhoaHoc);
   dbms_output.put_line('Thêm bài thi thành công');
end;
```

PROCEDURE 20: Sửa bài thi

```
CREATE OR REPLACE PROCEDURE SuaBaiThi(
   p_MaBaiThi baithi.mabaithi%type,
   p_TenBaiThi baithi.tenbaithi%type default null,
   p_ThoiGian  baithi.thoigian%type default null,
   p_MaKhoaHoc baithi.makhoahoc%type default null)
as
begin
   if p_tenbaithi is null and p_thoigian is null and p_makhoahoc is null then
      raise_application_error(-20010,'Cập nhật thất bại');
   end if;
   update baithi
   set tenbaithi=nvl(p_tenbaithi,tenbaithi),
   thoigian=nvl(p_thoigian,thoigian),
   makhoahoc=nvl(p_makhoahoc,makhoahoc)
   where mabaithi=p_mabaithi;
   dbms_output.put_line('Cập nhật bài thi thành công');
end;
```

PROCEDURE 21: Xoá bài thi

```
CREATE OR REPLACE PROCEDURE XoaBaiThi(
   p_MaBaiThi baithi.mabaithi%type)
as
begin
   delete baithi where mabaithi=p_mabaithi;
   dbms_output.put_line('Xoá bài thi thành công');
end;
```

PROCEDURE 22: Bán sách

```
CREATE OR REPLACE PROCEDURE BanSach( p_MaSach in sach.masach%type,
p_SoLuong in sach.soluong%type)
```

```
as
begin
   if p_SoLuong <=0 then
      raise_application_error(-20010,'Số lượng sách bán bị sai');
   end if;
   update sach set soluong=soluong-p_SoLuong
   where masach=p_MaSach;
   if sql%rowcount >0 then
      dbms_output.put_line('Bán sách thành công');
   end if;
end;
```

PROCEDURE 23: Lấy thông tin bài thi

```
CREATE OR REPLACE PROCEDURE LayThongTinBaiThi(
   p_mabaithi baithi.mabaithi%type default null
)
as
   cursor cursor_baithi is
   select baithi.*,khoahoc.tenkhoahoc
   from baithi
   join khoahoc on baithi.makhoahoc=khoahoc.makhoahoc
   where mabaithi=nvl(p_mabaithi,mabaithi);
begin
   for rec in cursor_baithi
   loop
      dbms_output.put_line('Thông tin bài thi: ');
      dbms_output.put_line('Mã bài thi: '||rec.mabaithi);
      dbms_output.put_line('Tên bài thi: '||rec.tenbaithi);
      dbms_output.put_line('Thời gian thi: '||to_date(rec.thoigian,'dd/mm/yyyy'));
      dbms_output.put_line('Mã khoá học: '|| rec.makhoahoc);
      dbms_output.put_line('Tên khoá học: '|| rec.tenkhoahoc);
      dbms_output.put_line('--------------------------------');
   end loop;
end;
```

PROCEDURE 24: Thêm sách

```
CREATE OR REPLACE PROCEDURE ThemSach (
   p_TenSach IN Sach.TenSach%TYPE,
   p_GiaSach IN Sach.GiaSach%TYPE,
   p_SoLuong IN Sach.SoLuong%TYPE,
   p_MaKhoaHoc IN Sach.MaKhoaHoc%TYPE
) AS
BEGIN
   INSERT INTO Sach (TenSach, GiaSach, SoLuong, MaKhoaHoc)
   VALUES (p_TenSach, p_GiaSach, p_SoLuong, p_MaKhoaHoc);
```

```
    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Sách đã được thêm thành công.');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
END ThemSach;
```

PROCEDURE 25: Cập nhật sách

```
CREATE OR REPLACE PROCEDURE CapNhatSach (
    p_MaSach IN Sach.MaSach%TYPE,
    p_TenSach IN Sach.TenSach%TYPE default null,
    p_GiaSach IN Sach.GiaSach%TYPE default null,
    p_SoLuong IN Sach.SoLuong%TYPE default null,
    p_MaKhoaHoc IN Sach.MaKhoaHoc%TYPE default null
) AS
BEGIN
    UPDATE Sach
    SET
        TenSach =nvl( p_TenSach,tensach),
        GiaSach =nvl( p_GiaSach,giasach),
        SoLuong = nvl(p_SoLuong,soluong),
        MaKhoaHoc = nvl(p_MaKhoaHoc,makhoahoc)
    WHERE MaSach = p_MaSach;
    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Không tìm thấy sách với Mã Sách đã nhập.');
    ELSE
        COMMIT;
        DBMS_OUTPUT.PUT_LINE('Thông tin sách đã được cập nhật thành công.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
END CapNhatSach;
```

PROCEDURE 26: Xoá sách

```
CREATE OR REPLACE PROCEDURE XoaSach (
    p_MaSach IN Sach.MaSach%TYPE
) AS
BEGIN
    DELETE FROM Sach WHERE MaSach = p_MaSach;
    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Không tìm thấy sách với Mã Sách đã nhập.');
    ELSE
        COMMIT;
        DBMS_OUTPUT.PUT_LINE('Sách đã được xóa thành công.');
```

```
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
END XoaSach;
```

PROCEDURE 27: Lấy thông tin sách

```
CREATE OR REPLACE PROCEDURE LayThongTinSach (
    p_MaSach IN Sach.MaSach%TYPE DEFAULT NULL
) AS
    CURSOR c_Sach IS
        SELECT MaSach, TenSach, GiaSach, sach.SoLuong, TenKhoaHoc
        FROM Sach
        join khoahoc on khoahoc.makhoahoc=sach.makhoahoc
        WHERE MaSach = NVL(p_MaSach, MaSach) OR p_MaSach IS NULL;
BEGIN
    FOR rec IN c_Sach LOOP
        DBMS_OUTPUT.PUT_LINE('Thông tin sách:');
        DBMS_OUTPUT.PUT_LINE('Mã Sách: ' || rec.MaSach);
        DBMS_OUTPUT.PUT_LINE('Tên Sách: ' || rec.TenSach);
        DBMS_OUTPUT.PUT_LINE('Giá Sách: ' || rec.GiaSach);
        DBMS_OUTPUT.PUT_LINE('Số Lượng: ' || rec.SoLuong);
        DBMS_OUTPUT.PUT_LINE('Mã Khóa Học: ' || rec.TenKhoaHoc);
        DBMS_OUTPUT.PUT_LINE('------------------------');
    END LOOP;
    IF c_Sach%NOTFOUND THEN
        DBMS_OUTPUT.PUT_LINE('Không tìm thấy sách với Mã Sách đã nhập.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
END LayThongTinSach;
```

PROCEDURE 28: Xoá chức vụ

```
CREATE OR REPLACE PROCEDURE XoaChucVu (
    p_MaChucVu IN ChucVu.MaChucVu%TYPE
) AS
BEGIN
    DELETE FROM ChucVu WHERE MaChucVu = p_MaChucVu;
END;
```

2. TRIGGER

TRIGGER 1: Kiểm tra tránh trùng tên khoá học

```
CREATE OR REPLACE TRIGGER trg_insert_khoahoc BEFORE
   INSERT OR UPDATE ON khoahoc
   FOR EACH ROW
DECLARE
   v_count INT;
BEGIN
   SELECT
      COUNT(*)
   INTO v_count
   FROM
      khoahoc
   WHERE
      lower(tenkhoahoc) = lower(:new.tenkhoahoc)
      OR makhoahoc = :new.makhoahoc;
   IF v_count > 0 THEN
      raise_application_error(-20001, 'Khoá học đã tồn tại');
   END IF;
END;
```

TRIGGER 2: Kiểm tra khoá học có người đăng ký rồi thì không cho xoá

```
CREATE OR REPLACE TRIGGER trg_delete_khoahoc BEFORE
   DELETE ON khoahoc
   FOR EACH ROW
DECLARE
   v_khoahoc_record khoahoc%rowtype;
   v_count          INT;
BEGIN
   SELECT
      COUNT(*)
   INTO v_count
   FROM
      danhsachdangky
   WHERE
         makhoahoc = :old.makhoahoc
      AND trangthai = 'Đã thanh toán';
   IF v_count > 0 THEN
      raise_application_error(-20010, 'Đã có người đăng ký và thanh toán khoá học này');
   ELSIF :old.trangthai NOT IN ( 'Chuẩn bị', 'Đã kết thúc' ) THEN
      raise_application_error(-20003, 'Khoá học đang được giảng dạy.');
   END IF;
END;
```

TRIGGER 3: Tăng giảm lương không quá 30%

```
create or replace trigger trg_update_salary_nhanvien
before update of luong
on nhanvien for each row
begin
   if (:new.luong/:old.luong) > 1.3 or (:new.luong/:old.luong)<0.7 then
      raise_application_error(-20010,'Lương cập nhật không tăng giảm quá 30%');
   end if;
   dbms_output.put_line('Cập nhật lương nhân viên thành công');
end;
```

TRIGGER 4: Tránh trùng lịch dạy

```
create or replace trigger trg_insert_lichday
before insert
on lichday for each row
declare
   v_count int;
   v_sobuoi int;
begin
    SELECT
      COUNT(*)
   INTO v_count
   FROM
      lichday
   WHERE
         magiaovien = :new.magiaovien
      AND thoigian = :new.thoigian;
   if v_count>0 then
      raise_application_error(-20010,'Giáo viên đã có lịch dạy ở thời gian này');
   end if;
   select count(*) into v_count
   from lichday
   where makhoahoc=:new.makhoahoc;
   select sobuoi into v_sobuoi from khoahoc where makhoahoc=:new.makhoahoc;
   if v_count>= v_sobuoi then
      raise_application_error(-20010,'Lịch học của khoá học này đã đủ, không thể xếp
thêm');
   end if;
   if updating then
      dbms_output.put_line('Cập nhật lịch dạy thành công');
   elsif inserting then
      dbms_output.put_line('Thêm lịch dạy thành công');
   end if;
end;
```

TRIGGER 5: Kiểm tra ngày đăng ký không quá ngày hiện tại

```
CREATE OR REPLACE TRIGGER trg_insert_danhsachdangky
before insert or update of mahocvien,makhoahoc, ngay
on danhsachdangky for each row
declare
   v_count int;
   v_total int;
begin
   if :new.ngay > sysdate then
      raise_application_error(-20010,'Ngày đăng ký không được lớn hơn ngày hiện tại');
   end if;

   select count(*) into v_count
   from danhsachdangky
   where makhoahoc=:new.makhoahoc;

   select soluong into v_total
   from khoahoc
   where makhoahoc=:new.makhoahoc;

   if v_total<=v_count then
      raise_application_error(-20010,'Khoá học đã đủ học viên, không thể đăng ký
thêm.');
   end if;

   select count(*) into v_count
   from danhsachdangky
   where mahocvien=:new.mahocvien
   and makhoahoc=:new.makhoahoc
   and trangthai='Chưa thanh toán';

   if v_count>0 then
      raise_application_error(-20010,'Học viên đã đăng ký và chưa thanh toán khoá học
này');
   end if;

   select count(*) into v_count
   from danhsachdangky
   where mahocvien=:new.mahocvien
   and trangthai='Chưa thanh toán';

   if v_count>5 then
      raise_application_error(-20010,'Học viên đã đăng ký nhiều hơn 5 khoá học mà chưa
thanh toán, hãy xoá đăng ký và thử lại');
   end if;
   dbms_output.put_line('Đăng ký thành công');
```

```
end;
/
```

TRIGGER 6: Kiểm tra trùng bài thi

```
CREATE OR REPLACE TRIGGER trg_insert_baithi
before insert or update
on baithi for each row
declare
   v_count int;
begin
   select count(*)
   into v_count
   from baithi
   where tenbaithi=:new.tenbaithi
   and thoigian=:new.thoigian
   and makhoahoc=:new.makhoahoc;
   if v_count>0 then
      raise_application_error(-20010,'Bài thi trùng');
   else
      dbms_output.put_line('Thêm bài thi thành công');
   end if;
end;
```

TRIGGER 7 : Không thể xoá bài thi khi có điểm tồn tại

```
CREATE OR REPLACE TRIGGER trg_delete_baithi
before delete
on baithi for each row
declare
   v_countDiem int;
begin
   select count(*) into v_countDiem
   from Diem
   where mabaithi=:old.mabaithi;

   if v_countDiem>0 then
      raise_application_error(-20010,'Xoá bài thi thất bại');
   else
      dbms_output.put_line('Xoá bài thi thành công');
   end if;
end;
```

TRIGGER 8: Không được xoá chức vụ có người đang đảm nhiệm

```
CREATE OR REPLACE TRIGGER trg_delete_chucvu
before delete
on chucvu for each row
```

```
declare
   v_count int;
begin
   select count(*) into v_count
   from nhanvien
   where nhanvien.chucvu=:old.machucvu;

   if v_count>0 then
      raise_application_error(-20010,'Hiện đang có nhân viên giữ chức vụ này nên không
thể xoá chức vụ');
   else
      DBMS_OUTPUT.PUT_LINE('ChucVu được xóa thành công.');
   end if;
end;
```

TRIGGER 9: Tránh trùng chức vụ lúc thêm chức vụ mới

```
CREATE OR REPLACE TRIGGER trg_insert_chucvu
before insert or update
on chucvu for each row
declare
   v_count int;
begin
   select count(*) into v_count
   from chucvu
   where lower(tenchucvu)=lower(:new.tenchucvu);
   if v_count>0 then
      raise_application_error(-20010,'Chức vụ đã tồn tại');
   elsif inserting then
      DBMS_OUTPUT.PUT_LINE('ChucVu được thêm thành công.');
   elsif updating then
      DBMS_OUTPUT.PUT_LINE('ChucVu được cập nhật thành công.');
   end if;
end;
```