

GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

NGUYỄN PHƯƠNG TÂM

NGHIÊN CỨU KỸ THUẬT
TÁI CẤU TRÚC CƠ SỞ DỮ LIỆU

Chuyên ngành: Khoa học máy tính

Mã số: 60.48.01

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

Đà Nẵng - Năm 2010

Công trình được hoàn thành tại
ĐẠI HỌC ĐÀ NẴNG

Người hướng dẫn khoa học: TS. Nguyễn Thanh Bình

Phản biện 1: PGS.TS. Đoàn Văn Ban

Phản biện 2: PGS.TSKH Trần Quốc Chiên

Luận văn được bảo vệ tại Hội đồng chấm Luận văn tốt nghiệp thạc sĩ kỹ thuật tại đại học Đà Nẵng vào ngày 15 tháng 10 năm 2010

** Có thể tìm hiểu luận văn tại:*

- Trung tâm Thông tin - Học liệu, Đại học Đà Nẵng.
- Trung tâm Học liệu, Đại học Đà Nẵng.

MỞ ĐẦU

1. Lý do chọn đề tài

Sự phát triển nhanh chóng của công nghệ thông tin và Internet vào nhiều lĩnh vực đời sống xã hội, quản lý kinh tế, khoa học kỹ thuật ... đã tạo ra nhiều cơ sở dữ liệu và nó không ngừng thay đổi theo thời gian. Mặt khác, các cơ sở dữ liệu của các dự án phần mềm đang triển khai cũng thay đổi theo yêu cầu của các nhà đầu tư dự án. Chính vì vậy mà phương pháp phát triển phần mềm truyền thống tạo ra một mô hình dữ liệu vật lý và logic trước khi bắt đầu dự án đã không còn phù hợp với thực tế.

Để giải quyết nhược điểm trên, phương pháp phát triển phần mềm linh hoạt bắt đầu xuất hiện vào đầu những năm 90. Mục tiêu của phương pháp này là phần mềm phải có khả năng biến đổi, phát triển và tiến hóa theo thời gian mà không cần phải làm lại từ đầu [14]. Phương pháp này được thực hiện dựa hai kỹ thuật chính đó là tái cấu trúc mã nguồn và kiểm thử. Tái cấu trúc mã nguồn là một kỹ thuật lập trình do Martin Fowler đề xuất năm 1999 nhằm thực hiện những thay đổi nhỏ trong mã nguồn để cải thiện thiết kế, và làm cho nó dễ hiểu và dễ hiệu chỉnh hơn. Và năm 2002, Scott W. Ambler, mô tả một kỹ thuật được gọi là tái cấu trúc cơ sở dữ liệu, đây là các kỹ thuật quan trọng trong phương pháp phát triển phần mềm linh hoạt, là kỹ thuật được thực hiện song song với tái cấu trúc mã nguồn giúp cho các đội phát triển dự án hoàn thành tốt công việc trong một môi trường luôn thay đổi.

Như vậy, vấn đề nghiên cứu và ứng dụng kỹ thuật tái cấu trúc mã nguồn và tái cấu trúc cơ sở dữ liệu là một xu hướng tất yếu và

cần thiết trong lĩnh vực phát triển phần mềm hiện nay. Và đó cũng chính là lý do mà tôi chọn nghiên cứu và thực hiện đề tài “Nghiên cứu kỹ thuật tái cấu trúc cơ sở dữ liệu” dưới sự hướng dẫn của TS. Nguyễn Thanh Bình.

2. Mục tiêu và nhiệm vụ nghiên cứu

Mục tiêu của đề tài là nghiên cứu về kỹ thuật tái cấu trúc cơ sở dữ liệu. Tái cấu trúc cơ sở dữ liệu là một thay đổi đơn giản trên lược đồ cơ sở dữ liệu để cải thiện thiết kế trong khi giữ lại cả hành vi và ngữ nghĩa thông tin của nó. Áp dụng các kỹ thuật tái cấu trúc cơ sở dữ liệu nhằm đảm bảo an toàn khi sửa chữa cơ sở dữ liệu thừa kế và hỗ trợ quy trình phát triển phần mềm theo hướng tiến hóa.

Với mục tiêu của tái cấu trúc cơ sở dữ liệu nêu trên, luận văn tập trung nghiên cứu quá trình tái cấu trúc cơ sở dữ liệu dựa trên những lược đồ cơ sở dữ liệu đã tồn tại để hỗ trợ cho một số quy trình phát triển phần mềm hiện đại như RUP, XP và AUP.

3. Đối tượng và phạm vi nghiên cứu

- Quy trình phát triển phần mềm theo hướng tiến hóa.
- Các kỹ thuật tái cấu trúc và tái cấu trúc cơ sở dữ liệu.
- Thử nghiệm thực tế các kỹ thuật tái cấu trúc cơ sở dữ liệu với các một cơ sở dữ liệu có sẵn.

Đề tài thuộc loại hình nghiên cứu ứng dụng.

5. Phương pháp nghiên cứu

- Thu thập và phân tích các tài liệu và thông tin liên quan đến đề tài.

- Thảo luận, lựa chọn phương hướng giải quyết vấn đề.
- Triển khai xây dựng chương trình ứng dụng.
- Kiểm tra, thử nghiệm và đánh giá kết quả.

6. Dự kiến kết quả

Nghiên cứu kỹ thuật tái cấu trúc và tái cấu trúc cơ sở dữ liệu. Dựa vào kết quả nghiên cứu được, áp dụng kỹ thuật đã nghiên cứu để thực hiện tái cấu trúc một cơ sở dữ liệu đã tồn tại.

Viết các bài báo công bố kết quả nghiên cứu liên quan đến đề tài.

7. Ý nghĩa khoa học và thực tiễn

Kết quả nghiên cứu có thể làm tài liệu tham khảo cho các đơn vị phát triển phần mềm đang cần tiến hành thay đổi cơ sở dữ liệu thừa kế hay đang phát triển phần mềm theo hướng tiếp cận linh hoạt.

Phần nghiên cứu lý thuyết sẽ cung cấp một cách nhìn tổng quát về kỹ thuật tái cấu trúc mã nguồn và tái cấu trúc cơ sở dữ liệu.

Phần ứng dụng cung cấp cách thức thực hiện kỹ thuật tái cấu trúc cơ sở dữ liệu để cải tiến hiệu suất làm việc trên cơ sở dữ liệu của các phần mềm truy xuất đến nó.

8. Đặt tên đề tài

“Nghiên cứu kỹ thuật tái cấu trúc cơ sở dữ liệu”

9. Bố cục luận văn

Toàn bộ nội dung luận văn được chia thành 3 chương:

Chương 1: Kỹ thuật tái cấu trúc mã nguồn (refactoring)

Chương 2: Kỹ thuật tái cấu trúc cơ sở dữ liệu (database refactoring)

Chương 3: Ứng dụng kỹ thuật tái cấu trúc cơ sở dữ liệu trên chương trình thực tế

CHƯƠNG 1 - KỸ THUẬT TÁI CẤU TRÚC MÃ NGUỒN (REFACTORING)

Trong chương này trình bày các vấn đề liên quan đến kỹ thuật tái cấu trúc mã nguồn gồm các mục: Định nghĩa, lý do thực hiện, các trường hợp nên thực hiện và áp dụng kỹ thuật tái cấu trúc mã nguồn để cải tiến mã nguồn.

1.1. ĐỊNH NGHĨA KỸ THUẬT TÁI CẤU TRÚC MÃ NGUỒN

Theo Martin Folwer và Kent Beck [6], tái cấu trúc mã nguồn có hai định nghĩa như sau:

Định nghĩa 1: *Tái cấu trúc mã nguồn (Refactoring)* (danh từ): *Sự thay đổi cấu trúc bên trong của phần mềm để làm cho nó trở nên dễ hiểu hơn và ít tốn chi phí trong việc cập nhật hay điều chỉnh mà không làm thay đổi ứng xử bên ngoài.*

Định nghĩa 2: *Tái cấu trúc mã nguồn (Refactor)* (động từ): *Cấu trúc lại phần mềm bằng cách áp dụng các bước tái cấu trúc (refactorings) mà không làm thay đổi ứng xử bên ngoài.*

Theo định nghĩa, mục tiêu đầu tiên của tái cấu trúc mã nguồn là làm cho chương trình dễ đọc và khi cần thiết có thể cập nhật thì vẫn không làm thay đổi hoặc có nhưng không đáng kể đến các hành vi ứng xử bên ngoài của phần mềm.

Mục tiêu kế tiếp mà chúng ta cần lưu ý đó là refactoring không làm thay đổi ứng xử bên ngoài của phần mềm. Phần mềm sẽ thực thi và xử lý các chức năng như trước. Bất kỳ người dùng nào cũng không thể cảm nhận về những sự thay đổi này.

1.2. LÝ DO NÊN TÁI CẤU TRÚC MÃ NGUỒN

1.2.1 Tái cấu trúc mã nguồn cải thiện thiết kế phần mềm

Trong quá trình thiết kế phần mềm, nếu chúng ta áp dụng tái cấu trúc mã nguồn sẽ là một giải pháp hiệu quả vì tái cấu trúc mã nguồn sẽ làm chương trình trở nên gọn hơn, dễ hiểu và đảm bảo những hạn chế thấp nhất trong quá trình phát triển và cập nhật.

1.2.2 Tái cấu trúc mã nguồn làm mã nguồn phần mềm dễ hiểu

Trong chu kỳ sống của phần mềm sẽ có nhiều người cùng tham gia vào việc phát triển và bảo trì. Việc áp dụng tái cấu trúc (thông qua việc sửa đổi định danh, từ ngữ, cách đặt tên cho các thành phần trong mã nguồn) giúp làm cho đoạn mã nguồn tuân theo qui chuẩn để có khả năng đọc được và chương trình dễ hiểu hơn.

1.2.3 Tái cấu trúc mã nguồn giúp phát hiện và hạn chế lỗi

Tái cấu trúc giúp hiểu đoạn mã nguồn từ đó giúp chúng ta trong việc phát hiện lỗi. Ngoài ra việc sắp đặt lại các logic luồng làm việc của mã nguồn giúp cho luồng xử lý rõ ràng hơn và tránh các sai sót có khả năng xảy ra.

1.2.4 Tái cấu trúc mã nguồn giúp đẩy nhanh quá trình phát triển phần mềm

Tái cấu trúc mã nguồn giúp đẩy nhanh quá trình phát triển phần mềm thông qua các hiệu quả mà nó mang lại:

- Tăng tính dùng lại.
- Tăng tính tiến hóa.
- Tăng tính gắn gũi với người dùng.

Ngày nay tái cấu trúc chính là một chuẩn mực lập trình của mọi lập trình viên khi làm việc theo nhóm, khi bắt đầu làm việc ở công ty lớn, các lập trình viên sẽ được huấn luyện và đào tạo để tuân thủ các yêu cầu làm việc: như quy tắc đặt tên biến, khi viết mã nguồn áp dụng mẫu nào, xây dựng đơn vị kiểm thử ra sao ...

1.3. KHI NÀO THỰC HIỆN TÁI CẤU TRÚC MÃ NGUỒN

1.3.1 Tái cấu trúc mã nguồn khi thêm chức năng

1.3.2 Tái cấu trúc mã nguồn khi cần sửa lỗi

1.3.3 Tái cấu trúc mã nguồn khi thực hiện duyệt chương trình

1.4. ÁP DỤNG KỸ THUẬT TÁI CẤU TRÚC ĐỂ CẢI TIẾN MÃ NGUỒN

Tái cấu trúc là một trong những phương pháp nhằm nâng cao chất lượng phần mềm đã bắt đầu được nghiên cứu và ứng dụng những năm 90 trong quy trình phát triển phần mềm.

Qua quá trình nghiên cứu và phát triển, một tập các kỹ thuật tái cấu trúc đã được đặc tả chi tiết và phần lớn các kỹ thuật tái cấu trúc trên đã và đang dần được tích hợp vào trong các công cụ phát triển phần mềm nhằm hỗ trợ cho các nhà phát triển trong việc rút ngắn thời gian tạo nên các phần mềm có chất lượng cao và ổn định, đáp ứng tốt các yêu cầu hoạt động của hiện tại và những thay đổi cần thiết trong tương lai.

1.4.1 Các kỹ thuật tái cấu trúc mã nguồn

1.4.2 Cải tiến mã nguồn dựa trên kỹ thuật tái cấu trúc

1.5. NHẬN XÉT VÀ KẾT LUẬN

Kỹ thuật tái cấu trúc mã nguồn là một kỹ thuật làm thay đổi bên trong phần mềm, làm cho nó trở nên dễ đọc, dễ hiểu và dễ hiệu chỉnh hơn nhưng không làm thay đổi hành vi ứng xử bên ngoài. Kỹ thuật này hiện đang được nhiều quốc gia có nền công nghiệp phần mềm phát triển áp dụng rộng rãi. Trong một bài báo về tương lai của ngành công nghệ phần mềm, một chuyên gia trong lĩnh vực quản lý và tư vấn các chiến lược phần mềm Alex Iskold đã đưa ra một nhận định rằng nền công nghệ phần mềm trong tương lai gần sẽ phát triển theo phương pháp phần mềm phát triển linh hoạt thay cho phương pháp mô hình thác nước đã tồn tại [14]. Phương pháp phát triển phần mềm linh hoạt (Agile Development Method) ngoài việc đáp ứng khả năng tạo ra các phần mềm có sự ổn định cao còn có khả năng thích nghi và tiến hóa để thích hợp với môi trường hoạt động.

Tuy nhiên, một vấn đề khó khăn trong tái cấu trúc là cơ sở dữ liệu. Hầu hết các ứng dụng thương mại đều liên kết chặt chẽ với lược đồ cơ sở dữ liệu. Mỗi sự thay đổi nhỏ trong lược đồ cơ sở dữ liệu đều ảnh hưởng đến chương trình ứng dụng do sự liên kết này. Trong phần tiếp theo của luận văn, chúng ta sẽ tìm hiểu kỹ thuật tái cấu trúc cơ sở dữ liệu và cách thức để thực hiện nó một cách dễ dàng nhất.

CHƯƠNG 2 - TÁI CẤU TRÚC CƠ SỞ DỮ LIỆU (DATABASE REFACTORING)

Tái cấu trúc cơ sở dữ liệu là một thay đổi đơn giản trên lược đồ cơ sở dữ liệu để cải thiện thiết kế nhưng vẫn giữ nguyên hành vi và ngữ nghĩa thông tin. Trong chương này trình bày các vấn đề liên quan đến kỹ thuật tái cấu trúc cơ sở dữ liệu gồm các mục: Phát triển cơ sở dữ liệu theo quy trình tiến hóa, định nghĩa kỹ thuật tái cấu trúc cơ sở dữ liệu, quy trình thực hiện và triển khai ứng dụng.

2.1. PHÁT TRIỂN CƠ SỞ DỮ LIỆU THEO QUY TRÌNH TIẾN HÓA

2.1.1 Lý do nên phát triển cơ sở dữ liệu theo quy trình tiến hóa

2.1.2 Kỹ thuật phát triển cơ sở dữ liệu theo quy trình tiến hóa

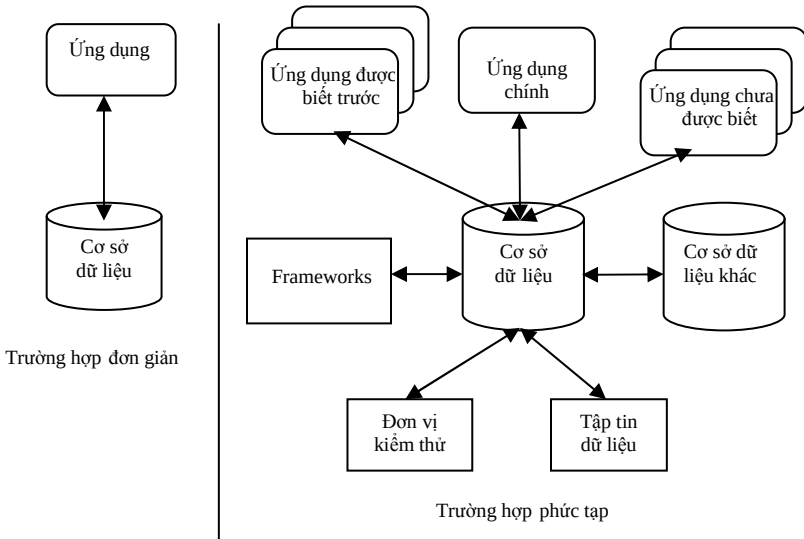
2.1.3 Một số trở ngại khi thực hiện theo quy trình tiến hóa

2.2. ĐỊNH NGHĨA KỸ THUẬT TÁI CẤU TRÚC CƠ SỞ DỮ LIỆU

2.2.1 Định nghĩa

Tái cấu trúc cơ sở dữ liệu là một thay đổi đơn giản trên lược đồ cơ sở dữ liệu để cải thiện thiết kế nhưng vẫn giữ nguyên hành vi và ngữ nghĩa thông tin, ta không thể thêm chức năng mới hoặc phá vỡ chức năng hiện có, cũng như không thể thêm dữ liệu mới hoặc thay đổi ý nghĩa của dữ liệu hiện có [4]. Theo quan điểm này, lược đồ cơ sở dữ liệu bao gồm cả cấu trúc, như định nghĩa bảng và khung nhìn, cũng như các thủ tục lưu trữ và trigger.

Tái cấu trúc cơ sở dữ liệu khó hơn tái cấu trúc mã nguồn bởi vì tái cấu trúc mã nguồn chỉ giữ lại ngữ nghĩa hành vi trong khi tái cấu trúc cơ sở dữ liệu giữ lại cả ngữ nghĩa thông tin. Hơn nữa, tái cấu trúc cơ sở dữ liệu phức tạp hơn do có nhiều kết nối đến cơ sở dữ liệu. Kết nối thể hiện sự phụ thuộc giữa hai đối tượng. Hình 2.1 mô tả kiến trúc của hệ thống kết nối đến cơ sở dữ liệu.



Hình 2.1 Hai trường hợp kết nối cơ sở dữ liệu

2.2.2 Các kỹ thuật tái cấu trúc cơ sở dữ liệu (xem chi tiết trong cuốn toàn văn).

2.2.3 Một số lỗi trong cơ sở dữ liệu (Database Smells)

2.3. QUY TRÌNH THỰC HIỆN

Quy trình thực hiện tái cấu trúc cơ sở dữ liệu được mô tả ở hình 2.2. Quy trình bắt đầu khi người phát triển ứng dụng nhận thấy

lược đồ cơ sở dữ liệu cần phải được tái cấu trúc ví dụ thực hiện một yêu cầu mới để sửa lỗi.

2.3.1 Xác nhận việc thực hiện tái cấu trúc cơ sở dữ liệu là thích hợp.

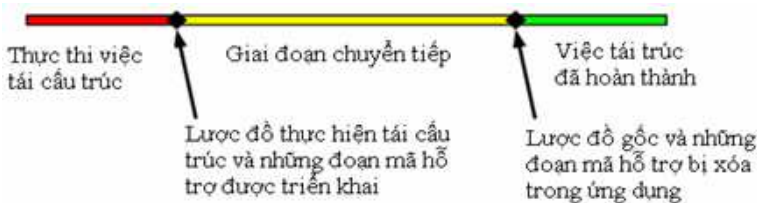
Đầu tiên, nhà quản trị cơ sở dữ liệu (DBA) xác định xem có cần thiết phải thực hiện việc tái cấu trúc không. Để xác định, DBA cần giải quyết ba vấn đề sau:

2.3.2 Chọn các kỹ thuật tái cấu trúc cơ sở dữ liệu phù hợp

Có rất nhiều kỹ thuật tái cấu trúc có thể áp dụng cho lược đồ cơ sở dữ liệu. Để xác định kỹ thuật nào phù hợp nhất cho lược đồ cơ sở dữ liệu, đầu tiên, ta phải phân tích và tìm hiểu vấn đề gặp phải.

2.3.3 Phản kháng (Deprecate) lược đồ cơ sở dữ liệu gốc.

Một kỹ thuật mà Pramod Sadalage và Peter Schuh (2002) đề xuất là giai đoạn phản kháng, hay còn gọi là giai đoạn chuyển tiếp, lược đồ gốc. Khi thực hiện tái cấu trúc, không đơn giản là làm việc với lược đồ hiện tại, mà cần phải làm việc với cả lược đồ gốc và mới song song cùng một lúc để cung cấp thời gian cho các đội ứng dụng khác thực hiện tái cấu trúc và triển khai lại hệ thống. Hình 2.3 mô tả vòng đời của việc tái cấu trúc cơ sở dữ liệu.



Hình 2.3 Vòng đời của việc tái cấu trúc cơ sở dữ liệu

2.3.4 Viết các đơn vị kiểm thử.

Để xác nhận cơ sở dữ liệu vẫn hoạt động tốt với các ứng dụng sau khi thực hiện tái cấu trúc, ta phải viết các đơn vị kiểm thử. Mục đích chính là đảm bảo các kiểm thử có tồn tại, và nên thực hiện các kiểm thử ít nhất một lần ở mức độ ứng dụng hoặc mức cơ sở dữ liệu.

2.3.5 Hiệu chỉnh lược đồ cơ sở dữ liệu

Một phần quan trọng của việc thực hiện một kỹ thuật tái cấu trúc là đảm bảo phần thay đổi lược đồ cơ sở dữ liệu được lưu trữ kèm theo với những tài liệu hướng dẫn phát triển cơ sở dữ liệu của dự án. Những tài liệu hướng dẫn này được cung cấp và hỗ trợ bởi nhóm DBA kèm theo tên của mỗi kỹ thuật.

2.3.6 Di chuyển dữ liệu nguồn

Nhiều kỹ thuật tái cấu trúc cơ sở dữ liệu yêu cầu thao tác trên dữ liệu nguồn. Giống như việc hiệu chỉnh lược đồ cơ sở dữ liệu, ta sẽ tạo kịch bản thực hiện di chuyển dữ liệu được yêu cầu. Các kịch bản này cũng được đánh số thứ tự để dễ quản lý.

2.3.7 Hiệu chỉnh các chương trình truy cập bên ngoài.

2.3.8 Chạy kiểm thử hồi quy

2.3.9 Viết tài liệu về công việc thực hiện

2.3.10 Thông báo về việc tái cấu trúc.

2.4. TRIỂN KHAI ỨNG DỤNG

Việc tái cấu trúc không chỉ thực hiện ở lược đồ cơ sở dữ liệu cá nhân và sandbox tích hợp mà cần được triển khai vào sản xuất. Để

triển khai tái cấu trúc cơ sở dữ liệu vào sản xuất, ta cần áp dụng chiến lược sau:

2.4.1 Triển khai giữa các sandbox.

2.4.2 Áp dụng tái cấu trúc cơ sở dữ liệu từng phần

2.4.3 Lập cửa sổ triển khai

2.4.4 Triển khai hệ thống

2.4.5 Loại bỏ các lược đồ phản kháng

2.5. NHẬN XÉT VÀ KẾT LUẬN

Tái cấu trúc cơ sở dữ liệu là một kỹ thuật làm thay đổi cơ sở dữ liệu để cải tiến thiết kế nhưng không làm thay đổi hành vi ứng xử bên ngoài cũng như ngữ nghĩa thông tin của nó. Tương tự như kỹ thuật tái cấu trúc mã nguồn, tái cấu trúc cơ sở dữ liệu cũng được ứng dụng rộng rãi để phù hợp với quy trình phát triển phần mềm linh hoạt.

Hiện tại kỹ thuật mới này cùng với kỹ thuật tái cấu trúc mã nguồn đang được áp dụng và triển khai ở các quốc gia có nền công nghiệp phần mềm phát triển (Mỹ, Nhật, Ấn Độ,...) để phù hợp qui trình phát triển phần mềm linh hoạt đang được xem là một tiêu chuẩn cho phát triển phần mềm trong tương lai gần.

Như vậy, vấn đề nghiên cứu và ứng dụng kỹ thuật tái cấu trúc cơ sở dữ liệu là một xu hướng tất yếu và cần thiết trong lĩnh vực phát triển công nghệ phần mềm hiện nay.

CHƯƠNG 3 - ỨNG DỤNG KỸ THUẬT TÁI CẤU TRÚC CƠ SỞ DỮ LIỆU TRÊN CHƯƠNG TRÌNH THỰC TẾ

Trong chương này trình bày ứng dụng kỹ thuật tái cấu trúc cơ sở dữ liệu trên chương trình thực tế, từ đơn giản đến phức tạp. Các ứng dụng bao gồm chương trình quản lý hồ sơ tuyển sinh đại học và chương trình quản lý thư viện.

3.1. TÁI CẤU TRÚC CƠ SỞ DỮ LIỆU CHƯƠNG TRÌNH HỒ SƠ TUYỂN SINH ĐẠI HỌC.

3.1.1 Mô tả chương trình ban đầu

- ❖ **Yêu cầu bài toán:** Lập trình một chương trình quản lý hồ sơ tuyển sinh, chương trình chỉ xử lý cập nhật và tìm kiếm hồ sơ của thí sinh.
- ❖ **Các chức năng của chương trình**

Chương trình quản lý hồ sơ tuyển sinh chỉ có hai chức năng chính là cập nhật và tìm kiếm hồ sơ của thí sinh.

- ❖ **Cơ sở dữ liệu chỉ gồm một bảng Hồ sơ.**

HOSO	
	MAHS
	NganhDuThi
	KhoiThi
	HoTen
	NgaySinh
	QueQuan
	GioiTinh
	Diachi

Hình 3.1 Lược đồ cơ sở dữ liệu ban đầu

- ❖ **Mã nguồn và giao diện chương trình trước khi thực hiện tái cấu trúc cơ sở dữ liệu:** xem trong cuốn toàn văn.

3.1.2 Tiến hành tái cấu trúc cơ sở dữ liệu

- ❖ **Xác định việc tái cấu trúc cơ sở dữ liệu là cần thiết**
- ❖ **Hiệu chỉnh lược đồ cơ sở dữ liệu và quá trình chuyển tiếp.**
- ❖ **Di chuyển dữ liệu.**
- ❖ **Cập nhật chương trình bên ngoài.**

3.1.3 Nhận xét

Tái cấu trúc cơ sở dữ liệu cho chương trình Quản lý hồ sơ tuyển sinh ở trên là trường hợp đơn giản và ít xảy ra trong thực tế. Cơ sở dữ liệu của chương trình chỉ có một bảng nên khi áp dụng kỹ thuật Split Column cho bảng này thì quá trình cập nhật cơ sở dữ liệu trở nên dễ dàng bởi nó không có các kết nối đến các bảng khác. Mặt khác, chương trình bên ngoài truy cập đến nó cũng khá đơn giản, chỉ có hai công việc là cập nhật và tìm kiếm cũng làm cho quá trình cập nhật chương trình bên ngoài không phức tạp.

Tái cấu trúc cơ sở dữ liệu cho chương trình Quản lý hồ sơ tuyển sinh ở trên là trường hợp đơn giản và ít xảy ra trong thực tế. Tiếp theo, ta xét chương trình thứ hai, chương trình Quản lý thư viện.

3.2. TÁI CẤU TRÚC CƠ SỞ DỮ LIỆU CHƯƠNG TRÌNH QUẢN LÝ THƯ VIỆN

3.2.1 Mô tả chương trình ban đầu

❖ **Yêu cầu bài toán:** Lập trình một chương trình quản lý thư viện cho trường Cao đẳng CNTT Hữu Nghị Việt Hàn.

❖ **Các chức năng của chương trình**

Chương trình Quản lý thư viện trường CD CNTT Hữu Nghị Việt Hàn gồm các chức năng sau:

- Quản lý kho sách: cập nhật thông tin về sách và sách hỏng của thư viện.

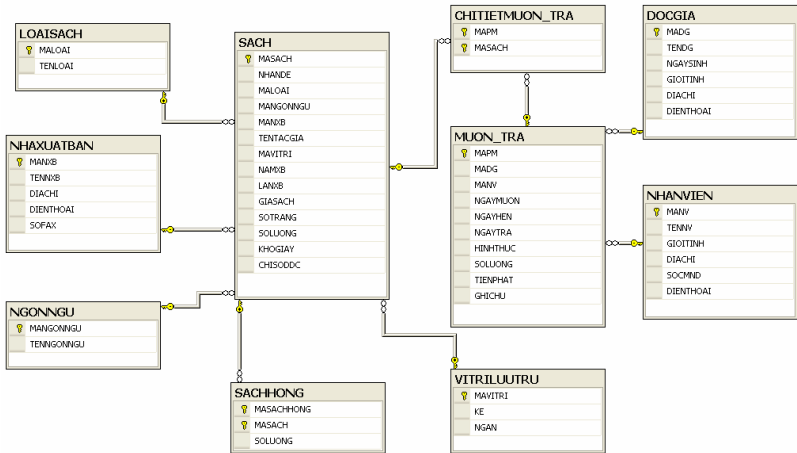
- Quản lý độc giả: cập nhật thông tin độc giả và in thẻ thư viện cho độc giả.

- Quản lý mượn trả: Quản lý tình hình mượn sách và gia hạn mượn cho độc giả.

- Thống kê tìm kiếm: Tìm kiếm thông tin sách, độc giả, mượn trả sách và thống kê số lượng sách hiện có trong thư viện, số lượng sách mượn, số lượng sách hỏng và sách trễ hạn.

Ngoài ra, chương trình cho phép cập nhật các thông tin về nhân viên, nhà xuất bản, loại sách, ngôn ngữ và vị trí lưu trữ.

Lược đồ cơ sở dữ liệu



Hình 3.3 Lược đồ cơ sở dữ liệu trước khi tái cấu trúc

3.2.2 Tiến trình tái cấu trúc cơ sở dữ liệu

❖ Xác định việc tái cấu trúc cơ sở dữ liệu là cần thiết.

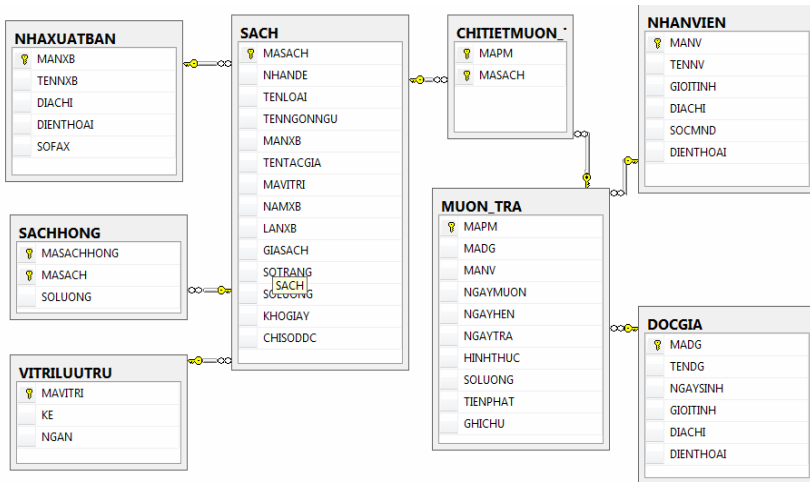
Với lược đồ cơ sở dữ liệu trên, ta nhận thấy đã đạt chuẩn 3NF. Tuy nhiên, mỗi lần thực hiện truy xuất thông tin về sách ta phải thực hiện liên kết nhiều bảng dữ liệu, ảnh hưởng đến hiệu năng của cơ sở dữ liệu. Trong trường hợp này ta áp dụng các kỹ thuật Move Column, Drop Column và Drop Table để thực hiện.

❖ Thực hiện kỹ thuật Move Column

❖ Thực hiện kỹ thuật Drop Column

❖ Thực hiện kỹ thuật Drop Table

Lược đồ cơ sở dữ liệu sau khi thực hiện tái cấu trúc



Hình 3.10 Lược đồ cơ sở dữ liệu sau khi tái cấu trúc

3.2.3 Nhận xét

Quá trình thực hiện tái cấu trúc cơ sở dữ liệu chương trình quản lý thư viện trở nên phức tạp hơn do cơ sở dữ liệu có nhiều kết nối và chương trình truy cập bên ngoài cũng phức tạp hơn.

KẾT LUẬN

1. Đánh giá kết quả đề tài

Qua quá trình nghiên cứu và triển khai ứng dụng kỹ thuật tái cấu trúc cơ sở dữ liệu, đề tài đã đạt được một số kết quả:

- Nắm rõ cơ sở lý thuyết của kỹ thuật tái cấu trúc mã nguồn và tái cấu trúc cơ sở dữ liệu.

- Dấu hiệu nhận biết mã xấu trong chương trình và các giải pháp cải tiến dựa trên kỹ thuật tái cấu trúc mã nguồn

- Dấu hiệu nhận biết các lỗi trong cơ sở dữ liệu và áp dụng kỹ thuật tái cấu trúc cơ sở dữ liệu để cải tiến hoạt động của cơ sở dữ liệu.

- Nắm rõ được quy trình thực hiện tái cấu trúc cơ sở dữ liệu và áp dụng vào thực tế.

- Bài báo công bố các kết quả nghiên cứu và ứng dụng của đề tài.

2. Phạm vi ứng dụng

Với những kết quả đạt được, đề tài có ý nghĩa thực tiễn trong lĩnh vực nghiên cứu và ứng dụng công nghệ mới của ngành công nghệ phần mềm tiên tiến, đặc biệt trong quy trình phát triển phần mềm theo mô hình tiến hóa.

Đề tài có khả năng ứng dụng trong phạm vi và lĩnh vực:

- Triển khai ứng dụng tái cấu trúc cơ sở dữ liệu trong quy trình phát triển phần mềm để cải tiến hiệu năng làm việc của cơ sở dữ liệu.

- Tài liệu tham khảo cho sinh viên chuyên ngành khoa học máy tính và các nhà quản trị cơ sở dữ liệu.

- Cơ sở lý thuyết cho việc xây dựng lại cơ sở dữ liệu thừa kế.

3. Hạn chế và hướng phát triển

Luận văn chỉ mới đưa ra phương pháp áp dụng kỹ thuật tái cấu trúc cơ sở dữ liệu một cách thủ công, chưa giới thiệu hoặc xây dựng được công cụ hỗ trợ. Luận văn cũng chưa áp dụng được các công cụ tái cấu trúc mã nguồn cho chương trình bên ngoài sau khi thực hiện tái cấu trúc cơ sở dữ liệu. Nhưng dựa trên nghiên cứu lý thuyết về tái cấu trúc mã nguồn và tái cấu trúc cơ sở dữ liệu, đề tài có thể mở rộng và tiếp tục xây dựng được các công cụ ứng dụng kỹ thuật tái cấu trúc cơ sở dữ liệu và áp dụng các công cụ tái cấu trúc mã nguồn để tự động cập nhật chương trình truy cập cơ sở dữ liệu.