

ĐƯỜNG ĐẠI HỌC KINH TẾ QUỐC DÂN
KHOA TIN HỌC KINH TẾ



Chủ biên: ThS. Trịnh Hoài Sơn



© 2004 Crystal Decisions Inc. All rights reserved. Crystal Decisions, Crystal Reports, Crystal Enterprise, Crystal Analytics, SiebelInfo, and SiebelInfo Web are trademarks or registered trademarks of Crystal Decisions Inc. All other trademarks referred to are the property of their respective owners.

crystal decisions



This version of Microsoft Visual Studio .NET is required.
This program is protected by Microsoft's ClickOnce technology, which allows it to check for updates at startup. Changes to the file may be made later in Microsoft Visual Studio .NET.

InstallShield X
Premier Edition

Meliasoft
EXCEL VBA Integration Suite

Giáo trình
**PHÁT TRIỂN ỨNG DỤNG
TRONG QUẢN LÝ**

NHÀ XUẤT BẢN ĐẠI HỌC KINH TẾ QUỐC DÂN

TRƯỜNG ĐẠI HỌC KINH TẾ QUỐC DÂN
KHOA TIN HỌC KINH TẾ
to m cs

Chủ biên: ThS. TRỊNH HOÀI SƠN

Giáo trình
PHÁT TRIỂN ỨNG DỤNG TRONG QUẢN LÝ

NHÀ XUẤT BẢN ĐẠI HỌC KINH TẾ QUỐC DÂN

Lời nói đầu

Đất nước Việt Nam đang trong quá trình hội nhập kinh tế quốc tế rất mạnh mẽ. Các doanh nghiệp của Việt Nam không ngừng tự đổi mới, hiện đại hóa công tác quản lý, nâng cao giá trị của bản thân để có thể đáp ứng được những đòi hỏi của quá trình hội nhập kinh tế quốc tế của đất nước.

Tin học hóa công tác quản lý hay là việc trang bị những phần mềm ứng dụng để hiện đại hóa công tác quản lý là một giải pháp cũng như là một xu hướng tất yếu của các doanh nghiệp Việt Nam trong quá trình phát triển và hội nhập.

Chương trình đào tạo chuyên ngành của sinh viên khoa Tin học Kinh tế Đại học Kinh tế Quốc dân bao gồm nhiều môn học có thể được phân chia thành các nhóm chính như sau: các môn học cung cấp kiến thức lý thuyết cơ sở, nền tảng về tin học và các lĩnh vực nghiên cứu của tin học; các môn học rèn luyện tư duy và kỹ năng lập trình; ...

Môn học Phát triển ứng dụng trong quản lý không thuộc vào các nhóm nêu trên mà nó là sự tổng hợp kiến thức của hầu hết các môn học chuyên ngành. Thực vậy, để phát triển một phần mềm ứng dụng trong quản lý được thương mại hóa thì đòi hỏi rất nhiều các mảng kiến thức khác nhau: từ những kiến thức mang tính lý thuyết về thông tin, hệ thống thông tin, cơ sở dữ liệu, công nghệ phần mềm... đến những kiến thức và kỹ năng cụ thể như lập trình cơ sở; lập trình nâng cao; cấu trúc dữ liệu và giải thuật...

Mục đích của môn học:

- Trang bị cho sinh viên kiến thức và kỹ năng để có thể thực hiện từng bước của quá trình phân tích, thiết kế đến xây dựng và đóng gói một phần mềm ứng dụng trong quản lý trên cơ sở ngôn ngữ lập trình cao cấp được sử dụng phổ biến trong thực tế. Sau khi kết thúc môn học, sinh viên thu được những kết quả sau:

- Nắm rõ các bước để xây dựng một phần mềm hoàn chỉnh mang tính thương mại

- Biết cách sử dụng và khai thác một số tiện ích quan trọng để tạo ra phần mềm quản lý mang tính chuyên nghiệp

- Có kinh nghiệm vận dụng ngay trong việc thực hiện làm Đề án chuyên ngành và Báo cáo thực tập tốt nghiệp cuối khóa.

Nội dung của giáo trình

Giáo trình được viết theo chương trình môn học thuộc chương trình khung ngành Hệ thống thông tin kinh tế tại trường Đại học Kinh tế Quốc dân và được Hiệu trưởng phê duyệt làm tài liệu chính thức dùng cho giảng dạy, học tập ở trường Đại học Kinh tế Quốc dân theo hợp đồng trách nhiệm số 10-08/HĐTN ngày 15 tháng 9 năm 2008.

Ngoài lời nói đầu và tài liệu tham khảo, giáo trình được chia thành 7 chương và phụ lục

Chương 1: Hệ thống thông tin quản lý với tổ chức doanh nghiệp

Chương 2: Ngôn ngữ lập trình Visual Basic 6.0

Chương 3: Xây dựng giao diện của ứng dụng trong quản lý

Chương 4: Lập trình liên kết và khai thác cơ sở dữ liệu

Chương 5: Tạo báo cáo đầu ra cho ứng dụng

Chương 6: Biên soạn tài liệu hướng dẫn sử dụng

Chương 7: Tạo bộ cài đặt cho ứng dụng

Phụ lục: Một số đề bài tập lớn dành cho việc thực hành phát triển ứng dụng trong quản lý và một tài liệu minh họa tất cả các bước của quy trình phân tích, thiết kế và xây dựng một phần mềm ứng dụng cụ thể.

Tác giả xin trân trọng cảm ơn tất cả các thầy, cô giáo trong khoa Tin học Kinh tế, Đại học Kinh tế Quốc dân, đặc biệt là các thầy, cô giáo có tên sau đây về những tư tưởng, đóng góp chuyên môn và cung cấp tài liệu cho giáo trình: PGS.TS Hàn Việt Thuận; TS. Cao Dinh Thi; TS. Đặng Quốc Vinh; ThS. Trần Công Uẩn.

Giáo trình mới được biên soạn lần đầu nên không tránh khỏi những thiếu sót. Rất mong nhận được những ý kiến đóng góp của thầy, cô giáo cũng như các bạn sinh viên để giáo trình có thể ngày một hoàn thiện hơn. Mọi ý kiến đóng góp xin gửi về: *Trịnh Hoài Sơn, giảng viên khoa Tin học Kinh tế, Đại học Kinh tế Quốc dân. Email: sonth@neu.edu.vn*

Hà Nội, tháng 3 năm 2010

Chương 1

HỆ THỐNG THÔNG TIN QUẢN LÝ VỚI TỔ CHỨC DOANH NGHIỆP

1.1. Lợi ích và thực trạng của việc tin học hóa công tác quản lý tại các doanh nghiệp Việt Nam

1.1.1. Lợi ích của việc tin học hóa công tác quản lý

Tin học hóa công tác quản lý là một giải pháp cũng như xu hướng tất yếu của một doanh nghiệp khi muốn phát triển, mở rộng trong tương lai. Thực vậy, khi mà quy mô và phạm vi hoạt động của doanh nghiệp phát triển đến một mức nào đó sẽ làm cho số lượng dữ liệu phát sinh cần phải xử lý cùng với độ phức tạp của bài toán quản lý ngày càng tăng lên. Việc duy trì một hệ thống thông tin cũ thủ công hay lạc hậu sẽ dần trở thành một gánh nặng, không đáp ứng được yêu cầu của công tác quản lý, là nguyên nhân cản trở sự phát triển, mở rộng của doanh nghiệp hiện tại và trong tương lai.

Thực hiện tin học hóa công tác quản lý đem lại cho doanh nghiệp những lợi ích sau đây:

- ❖ Có khả năng cung cấp thông tin một cách nhanh chóng cho các nhà quản lý để hỗ trợ việc ra quyết định, đồng thời hỗ trợ việc kiểm tra thi hành quyết định. Toàn bộ quá trình xử lý thông tin, tổng hợp báo cáo được thực hiện tự động hóa bởi hệ thống phần mềm ứng dụng khiến cho thời gian cần thiết để đáp ứng yêu cầu của nhà quản lý trong việc truy vấn và khai thác thông tin sẽ rút ngắn rất nhiều. Hơn nữa, một hệ thống thông tin được xây dựng toàn diện, tổng thể còn cho phép nhà quản lý có thể thực hiện việc kiểm tra, theo dõi quá trình, tiến độ và kết quả việc thực hiện các quyết định quản lý được ban hành. Đối với các công ty có quy mô lớn, tính chất theo kiểu tập đoàn, công ty mẹ con thì việc tin học hóa công tác quản lý sẽ giúp cho họ tổ chức thành công việc quản lý, thu thập tổng hợp số liệu từ các nguồn khác nhau.

- ❖ Doanh nghiệp sẽ giám chi phí về nhân công, chi phí cho việc lưu trữ, bảo quản và tiết kiệm được thời gian. Để có thể đảm bảo thực hiện nhiệm vụ của mình thì một hệ thống quản lý thủ công cần rất nhiều nhân lực, có khi lên đến hơn 10 người. Khi áp dụng tin học vào hoạt động quản lý thì hầu hết những công việc tổng hợp và xử lý số liệu phải thực hiện thủ công trước đây nay được làm tự động hóa bởi các phần mềm máy tính. Công việc của cán bộ trong hệ thống thông tin quản lý nay chỉ còn phải cập nhật số liệu phát sinh từ giao dịch hay hoạt động của tổ chức với sự trợ giúp của máy tính sau đó thực hiện in các báo cáo theo yêu cầu tại các mục chức năng tương ứng của phần mềm ứng dụng. Việc lưu trữ hệ thống chứng từ, hóa đơn thủ công trước đây cần rất nhiều không gian cũng như tốn rất nhiều thời gian nếu có nhu cầu tìm kiếm và tra cứu. Cơ sở dữ liệu của hệ thống quản lý tin học hóa được lưu trữ trong các phương tiện nhớ của máy tính điện tử, được quản trị một cách hợp nhất bởi một hệ quản trị cơ sở dữ liệu bảo đảm tối ưu về không gian lưu trữ cũng như thời gian truy suất và tìm kiếm.
- ❖ Tin học hóa công tác quản lý giúp doanh nghiệp có cơ hội tiếp cận những ứng dụng hiện đại của công nghệ thông tin như tham gia thương mại điện tử, với cơ hội mở rộng phạm vi giao dịch vượt ra ngoài biên giới quốc gia vươn tới các nước trong khu vực và trên thế giới, thời gian giao dịch có thể được tiến hành 24h/1 ngày và 7 ngày/1 tuần. Với hệ thống thông tin quản lý hiện đại, doanh nghiệp sẽ có uy tín hơn, được tin tưởng hơn trong mắt các đối tác bạn hàng trong khu vực và trên thế giới, vì vậy sẽ có cơ hội tiếp cận những hợp đồng lớn hơn, giá trị cao hơn.

Mặc dù đem lại rất nhiều lợi ích nhưng việc tin học hóa công tác quản lý cũng đem lại rất nhiều thách thức cho doanh nghiệp:

- ❖ Đòi hỏi chi phí khá lớn cho việc trang bị máy móc thiết bị tin học. Xây dựng hệ thống thông tin tin học hóa đòi hỏi doanh nghiệp cần đầu tư trang bị các thành phần của hệ thống thông tin mà quan trọng và tốn nhiều chi phí nhất đó là đầu tư trang bị phần cứng và phần mềm. Về phần cứng bao gồm máy tính, máy văn phòng, mạng máy tính và các thiết bị phụ trợ khác; phần mềm bao gồm hệ điều hành, hệ quản trị cơ sở dữ liệu, các phần mềm ứng dụng trong quản lý và các phần mềm tiện ích khác. Theo kinh nghiệm thực tế thì tổng số

tiền đầu tư cho toàn bộ thành phần trên thấp nhất là vài nghìn đô la và cao nhất thì có thể lên tới 100 nghìn đô la.

- ❖ Đòi hỏi cán bộ trong tổ chức phải có một nền tảng kiến thức về ngoại ngữ và tin học. Hệ thống thông tin được tin học hóa áp dụng những công nghệ tiên tiến, để vận hành và khai thác sử dụng hiệu quả thì đòi hỏi đội ngũ cán bộ bên cạnh kiến thức về chuyên môn còn đòi hỏi những kiến thức tổng quát về tin học và ngoại ngữ. Đối với những doanh nghiệp mới thành lập thì yêu cầu này hoàn toàn có thể đáp ứng tuy nhiên tại các doanh nghiệp lâu năm, nhất là các doanh nghiệp nhà nước, thì yêu cầu này không phải lúc nào cũng có thể đáp ứng.
- ❖ Nếu việc quản lý không chặt chẽ các máy móc tin học có thể bị sử dụng sai mục đích, làm giảm hiệu suất làm việc của nhân viên. Khi thay thế hệ thống quản lý thủ công bằng hệ thống quản lý tin học hóa, doanh nghiệp đã nâng cao năng suất và hiệu quả làm việc của cán bộ, nhân viên lên nhiều lần. Những công việc mà trước đây phải làm thủ công thì nay được làm tự động bởi phần mềm máy tính. Số lượng nhân lực cần thiết của bộ máy quản lý vì thế mà giảm đi chỉ còn một nửa thậm chí là một phần ba. Số lượng lao động đối dư này nếu không được quản lý, bố trí công việc hợp lý thì đây sẽ là một sự lãng phí nguồn lực đáng kể. Hơn nữa theo kết quả của một cuộc điều tra đăng trên mạng Internet thì có tới hơn một nửa thời gian ngày làm việc của các nhân viên văn phòng dành cho việc khai thác, sử dụng mạng internet vì mục đích giải trí cá nhân. Đây là một bài toán quản lý khó đối với các nhà quản lý doanh nghiệp khi quyết định trang bị hệ thống thông tin quản lý mới.
- ❖ Vấn đề bảo mật thông tin không được quan tâm đúng mức sẽ làm thất thoát thông tin ra ngoài tạo ra những bất lợi cho tổ chức. Thông thường chi phí cho vấn đề bảo mật và ngăn chặn tấn công từ bên ngoài thường chiếm 30% tổng chi phí xây dựng hệ thống thông tin mới. Việc coi nhẹ sự phòng thủ và bảo mật sẽ giúp doanh nghiệp tiết kiệm được khá nhiều chi phí đầu tư xây dựng hệ thống thông tin quản lý, quản trị kinh doanh của mình. Đây là một sai lầm bởi doanh nghiệp đã không tính trước được những tồn thất, thiệt hại khi hệ thống của họ bị tin tặc tấn công, không chế và phá hoại! Trong thời gian gần đây, thông tin về việc tin tặc tấn công và phá hoại hay cộ

lập một website của một doanh nghiệp nào đó diễn ra khá thường xuyên. Việc tấn công chủ yếu được thực hiện bằng cách khai thác lỗ hổng trong các ứng dụng web hay kỹ thuật mã hóa để vòng tránh được sự phòng chống của mạng với khả năng phòng thủ kém của doanh nghiệp. Và một khi những kẻ tấn công đã kiểm soát được hệ thống máy chủ của doanh nghiệp thì mọi điều tồi tệ nhất đều có thể xảy ra. Tuy nhiên không có nghĩa là một doanh nghiệp đầu tư tiền bạc vào lĩnh vực bảo mật và phòng chống thì có thể yên tâm trước sự tấn công của tin tặc. Thực tế đã chứng minh rằng những công ty có trang bị hệ thống bảo mật tốt nhất cũng đã từng bị tấn công và chịu nhiều thiệt hại.

1.1.2. Các giai đoạn phát triển của ứng dụng tin học trong một tổ chức trên thế giới nói chung và ở Việt Nam nói riêng

Lịch sử phát triển của quá trình tin học hóa công tác quản lý tại Việt Nam nói riêng và trên toàn thế giới nói chung đã trải qua 3 giao đoạn từ mức độ thấp đến mức độ cao, từ xử lý thủ công đến xử lý cơ giới bằng các loại máy tính cơ học và cuối cùng là xử lý tự động trên cơ sở các công cụ tin học hiện đại.

Theo Rechard Nolan thì có 6 giai đoạn phát triển của phát triển ứng dụng tin học trong các tổ chức

Giai đoạn 1: Giai đoạn khởi đầu

Trong giai đoạn này, máy tính được đưa vào tổ chức. Công việc xử lý dữ liệu được thực hiện bởi cán bộ trông coi máy tính+ cán bộ lập trình+ nhân viên nhập dữ liệu. Xử lý dữ liệu thường gắn liền bởi những nghiệp vụ được xác định rõ ràng, làm việc với một tập hợp các quy tắc nhất định, các lao động giản đơn, cực nhọc, đơn điệu, lặp lại. Đây thực là những vấn đề tuyệt vời cho tự động hóa. Những bài toán trong kế toán tài chính thường được áp dụng máy tính đầu tiên là vì vậy.

Trong giai đoạn này, cán bộ xử lý dữ liệu và người sử dụng học về công nghệ thông tin và học cách làm việc với nhau. Đó là một thời kỳ ngây ngô, vụng về. Người dùng không có khái niệm về cái mà anh ta chờ đợi. Họ cũng chẳng có lý do gì để thất vọng về những cái họ nhận được. Có thể nói đây là thời kỳ người mù dẫn đường người mù. Người sử dụng không hiểu rõ loại vấn đề yêu cầu và cách đánh giá hiệu quả của cách giải quyết. Giai đoạn này đã kết thúc.

Giai đoạn 2: Giai đoạn lan rộng

Các thao tác để xử lý dữ liệu đã trở thành dễ dàng hơn và dễ tiếp cận hơn, người sử dụng đã thấy hứng thú hơn với công nghệ mới. Do đó yêu cầu ứng dụng máy tính tăng nhanh. Giai đoạn này cần bộ xử lý dữ liệu tự động đánh giá quá cao khả năng của máy tính. Không phải là ít lần chúng ta nghe thấy rằng máy tính giỏi giang có thể thay thế người quản lý bằng các nút bấm; rằng máy tính sẽ ra quyết định. Chẳng có gì ngạc nhiên khi thấy kết luận lão xược đó không được các nhà quản lý chấp nhận và phản ứng tiêu cực của họ đối với sự phát triển của ứng dụng công nghệ thông tin trong quản lý. Tuy nhiên cũng có khá nhiều người sử dụng ngây thơ đã bị những ảo tưởng như vậy kích động dẫn tới thời kỳ tăng trưởng không có kiểm soát những ứng dụng trong xử lý dữ liệu tự động.

Giai đoạn 3: Giai đoạn kiểm soát ứng dụng

Việc có quá nhiều yêu cầu tin học hóa, sự thiếu hiểu biết thấu đáo về CNTT và thiếu kinh nghiệm đã làm cho nhiều ứng dụng bị hạn chế về thời gian, vượt chi phí cho phép và hệ thống xử lý làm việc không tốt. Chúng không đáp ứng được sự mong đợi của các nhà quản lý cấp cao khi họ xem xét về lượng tiền đã chi ra và lợi nhuận tính được. Do đó, các nhà quản lý dữ liệu tự động bắt đầu xem xét kỹ lưỡng các yêu cầu về tự động hóa xử lý dữ liệu. Các nhà quản lý thấy rằng có thể quản lý bộ phận xử lý dữ liệu cũng gần như quản lý các bộ phận kỹ thuật trong tổ chức. Cán bộ quản lý xử lý dữ liệu bắt đầu suy nghĩ theo nghĩa kinh doanh. Người đặt yêu cầu xử lý tự động phải biện minh cho các yêu cầu của mình. Trách nhiệm của người sử dụng các nguồn lực thông tin đã được đặt ra trong tổ chức.

Một loại cán bộ mới ra đời - cán bộ quản lý có khả năng về CNTT. Vì cán bộ xử lý dữ liệu và người yêu cầu phải tiến hành phân tích chi phí/lợi nhuận cho các ứng dụng do đó cán bộ xử lý dữ liệu tự động phải học về kinh doanh còn người sử dụng phải học thêm về CNTT. Điều này có ảnh hưởng rất mạnh tới các hoạt động kinh doanh và các dự án mà họ đề xuất.

Giai đoạn 4: Giai đoạn tích hợp

Trong những năm 70, công nghệ máy tính tăng trưởng nhanh và công nghệ thông tin lúc đó ở vào thời kỳ cạnh tranh đổi mới. Một số người cho rằng công nghệ thông tin mới đưa vào có thể đủ thay thế cho 10 năm sử dụng có hiệu quả những gì đã có. Công nghệ phần mềm mới và các ngôn ngữ thế hệ 4 đã cho phép tạo ra sự tích hợp chức năng quản lý kinh doanh

và xử lý dữ liệu tự động, kết quả trực tiếp tập trung quản lý thông tin trong một cấu trúc đơn giản. Trong giai đoạn này người sử dụng không còn phải xếp hàng dài trước cửa phòng lập trình để đề nghị ưu tiên cho vấn đề của họ. Họ tự làm những công việc của chính họ trên máy tính. Giá cả của máy tính và phần mềm giảm xuống thấp phù hợp với nguồn lực tài chính của người dùng. Bộ phận chuyên trách về xử lý dữ liệu tự động tập trung những hoạt động của mình vào những công việc dịch vụ, cung cấp các tiện ích và trợ giúp kỹ thuật cho những người sử dụng.

Giai đoạn 5: Giai đoạn quản trị dữ liệu

Có thể nói, đây là giai đoạn hiện nay của các hệ thống thông tin. Bộ phận hệ thống thông tin đã nhận ra rằng thông tin là nguồn lực và mọi người phải được sử dụng nguồn lực ấy một cách dễ dàng. Chính vì thế, thông tin phải được quản lý một cách thích hợp. Dữ liệu phải được lưu trữ và duy trì sao cho mọi người sử dụng có thể tiếp cận tới chúng như một tài nguyên dùng chung. Để có thể dùng chung, mô hình dữ liệu phải được xây dựng độc lập với ứng dụng. Tư tưởng này cho phép người sử dụng phát triển ứng dụng của mình để sử dụng dữ liệu chung đó. Giai đoạn này đặc trưng bằng uy lực của người sử dụng, người mà giờ đây có trách nhiệm chính đối với sự tích hợp và sử dụng riêng tài nguyên thông tin của doanh nghiệp. Môi trường Client/Server đang phát triển mạnh. Server lưu trữ dữ liệu còn Client tra cứu, xem xét và xin các báo cáo.

Giai đoạn 6: Giai đoạn chín muồi

Ở giai đoạn này là sự đan kết hoàn toàn nguồn lực thông tin vào toàn bộ các hoạt động của tổ chức từ cấp chiến lược trở xuống. Cán bộ thông tin cấp cao là thành viên của đội ngũ quản lý cao cấp, đóng góp phần chính cho các quyết định kinh doanh và khai thác CNTT cho việc giành lợi thế cạnh tranh.

Xét thực tế tại Việt Nam, quá trình ứng dụng tin học vào công tác quản lý đã trải qua 3 giai đoạn:

- ❖ Giai đoạn 1: Những năm 90 của thế kỷ trước: Khi đó giá trị một máy tính khoảng vài chục triệu đồng. Đó là một khoản chi phí không nhỏ đối với hầu hết các doanh nghiệp. Hơn nữa, việc ứng dụng tin học nói chung trên toàn thế giới chưa cao. Ở VN hầu như chưa có các phần mềm ứng dụng phục vụ quản lý, kiến thức, trình độ tin học nói chung của toàn xã hội ở mức rất thấp, nhiều người mới chỉ lặn lội

biết đến sự xuất hiện của máy vi tính. Chỉ có một số ít doanh nghiệp có điều kiện trang bị máy tính để trợ giúp hoạt động quản lý với công việc đơn giản như soạn thảo văn bản, quản lý giấy tờ hay giải quyết các bài toán kinh tế đơn giản.

- ❖ **Giai đoạn 2: Trước năm 2000:** Giá bán của máy tính và các thiết bị tin học khác đã rẻ hơn nhiều. Quan trọng hơn là ở Việt Nam đã bắt đầu xuất hiện ngày càng nhiều các chương trình máy tính ứng dụng phục vụ hoạt động quản lý. Ngôn ngữ lập trình chủ yếu được sử dụng để xây dựng các chương trình phần mềm là Foxpro. Máy tính đã được ứng dụng trong phạm vi rộng hơn chủ yếu được ứng dụng vào lĩnh vực tài chính kế toán. Việc tin học hóa trong giai đoạn này diễn ra một cách tự phát không theo quy hoạch tổng thể. Tại các bộ phận, phòng ban trong doanh nghiệp tùy vào khả năng và nhu cầu công việc mà người ta chủ động trang bị máy tính và các phần mềm ứng dụng.
- ❖ **Giai đoạn 3: Từ 2000 đến nay:** Trong điều kiện máy tính thực sự trở nên phổ biến, các ứng dụng của nó tác động đến mọi lĩnh vực đời sống xã hội nhất là trong lĩnh vực quản lý. Một số doanh nghiệp trẻ ở Việt Nam đã bắt đầu xây dựng HTTTQL tổng thể và dưới sự quản lý của giám đốc CNTT. Các doanh nghiệp bước đầu tham gia vào TMĐT và thị trường chứng khoán. Tuy nhiên có thực trạng tại các doanh nghiệp đi trước (đã có sự ứng dụng tin học quản lý thời gian trước) gặp khó khăn trong việc đồng bộ hóa các ứng dụng tại các bộ phận chúc năng vào một tổng thể duy nhất.

1.1.3. Các lý do xây dựng một HTTT mới

Câu hỏi đầu tiên của việc phát triển một hệ thống thông tin mới là cái gì bắt buộc một tổ chức phải tiến hành phát triển hệ thống thông tin. Chúng ta có thể kể một số lý do chủ yếu như sau:

- ❖ *Những vấn đề về quản lý:*

Mục tiêu cuối cùng của việc xây dựng một hệ thống thông tin đó là có được những khả năng cung cấp cho các thành viên của tổ chức những công cụ trợ giúp quản lý một cách tốt nhất. Và các lí do về các yêu cầu của quản lý là lí do quan trọng nhất cho việc xây dựng một phần mềm quản lý mới. Thực vậy khi mà quy mô hay phạm vi hoạt động của đơn vị đã đạt tới một mức độ nào đó dẫn tới một khối lượng dữ liệu phát sinh cần xử lý và độ

phức tạp của bài toán quản lý trở nên không thể đáp ứng nổi nếu làm một cách thủ công. Đồng thời nhu cầu của sự phát triển và mở rộng trong tương lai đòi hỏi đơn vị phải đầu tư xây dựng một hệ thống thông tin mới.

❖ *Những yêu cầu mới của nhà quản lý:*

Trong xã hội thông tin hiện nay, các doanh nghiệp hoạt động theo cơ chế thị trường, coi thương trường như chiến trường, thì để thực hiện thành công nhiệm vụ quản lý của mình các nhà quản lý ngày càng đòi hỏi nhiều hơn với hệ thống thông tin. Họ mong muốn nhận được những thông tin chính xác nhất và đầy đủ nhất trong một khoảng thời gian là ngắn nhất.

Yêu cầu này không ngừng được tăng lên và đây là một động lực để thúc đẩy đơn vị thay đổi hệ thống thông tin cũ bằng một hệ thống mới hiện đại hơn.

❖ *Sự thay đổi của công nghệ:*

Năng lực tính toán cũng như khả năng lưu trữ của máy tính có khả năng tăng lên rất nhanh chóng tuân theo định luật Moore ("cứ sau 18 tháng thì năng lực tính toán của máy tính lại tăng lên gấp đôi")

Đó là sự phát triển của phần cứng máy tính, còn đối với phần mềm của máy tính nó còn có tốc độ phát triển nhanh hơn nữa; những ngôn ngữ lập trình, hệ quản trị cơ sở dữ liệu mới hơn, tối ưu hơn không ngừng được cài tiến ra đời thay thế cho các thế hệ trước đó.

Hai lý do trên cũng tạo ra động lực rất lớn cho việc thay thế một hệ thống thông tin mới cho hệ thống thông tin cũ.

❖ *Sự thay đổi về sách lược chính trị:*

Các quốc gia luôn không ngừng hoàn thiện bộ máy quản lý của mình, các chính sách pháp luật, quy định thể chế thường xuyên sửa đổi và cập nhật để phù hợp với thực tế phát triển của xã hội.

Những quy định, những chuẩn mực trong việc xử lý dữ liệu, tổ chức hoạt động quản lý chắc chắn cũng phải thay đổi. Vì vậy, những hệ thống thông tin mới sẽ trở nên không phù hợp và nó cần phải thay thế bằng một hệ thống thông tin mới.

1.2. Khái niệm phần mềm và những vấn đề liên quan

Máy tính không thể hoạt động được nếu thiếu các chương trình phần mềm. Phần mềm làm cho phần cứng máy tính ứng dụng được vào các vấn

dè cần giải quyết và làm cho nó trở thành có ích. Đối với các nhà quản lý, điều quan trọng là phải hiểu rõ: Phần mềm thực sự là gì? Có những loại phần mềm nào? Sử dụng chúng để phục vụ công việc của mình ra sao?

1.2.1. Khái niệm phần mềm

- Phần mềm là tập hợp gồm 3 yếu tố: các chương trình máy, các cấu trúc dữ liệu phù hợp để chương trình có thể thực hiện những chức năng của nó và các tài liệu hướng dẫn sử dụng các chương trình ấy.

1.2.2. Phân loại phần mềm

Có nhiều cách phân loại phần mềm. Theo một cách được nhiều người thừa nhận: Phần mềm gồm 2 loại: phần mềm hệ thống và phần mềm ứng dụng.

- **Phần mềm hệ thống:** Bao gồm các chương trình hướng dẫn những hoạt động cơ bản của một máy tính như hiện thông tin lên màn hình, lưu trữ dữ liệu trên đĩa từ, in kết quả, liên lạc với các thiết bị ngoại vi, phân tích và thực hiện các lệnh của người dùng. Các chương trình thuộc phần mềm hệ thống giúp cho phần cứng của máy tính hoạt động một cách có hiệu quả. Phần mềm hệ thống phục vụ đồng đảo người dùng nên thường được các chuyên gia lập trình hệ thống biên soạn và bán trên thị trường. Các nhà lập trình hệ thống thường dùng ngôn ngữ cấp thấp để thu được những bản dịch với chất lượng cao, đảm bảo tiết kiệm thời gian và bộ nhớ cho máy tính. Phần mềm hệ thống lại được chia làm 4 loại.

+ Hệ điều hành: là tập hợp các chương trình có chức năng điều khiển, quản lý và giám sát sự hoạt động của các thiết bị phần cứng đồng thời tạo ra môi trường thích hợp cho các phần mềm ứng dụng hoạt động, đảm bảo khai thác tối ưu các tài nguyên của hệ thống. Microsoft là nhà sản xuất hệ điều hành window dành cho máy tính cá nhân với các thế hệ như Window 95, window 98, window XP và mới nhất hiện nay là window VISTA.

+ Các chương trình tiện ích: là các phần mềm mở rộng, bổ sung thêm các chức năng cho hệ điều hành để giúp tạo sự thuận tiện cho người dùng cũng như nâng cao sự tối ưu trong việc khai thác tài nguyên hệ thống. Các chương trình tiện ích thực hiện những nhiệm vụ như sửa soạn các đĩa từ để lưu trữ dữ liệu, cung cấp thông tin về các tệp trên đĩa, sao chép dữ liệu từ đĩa này sang đĩa khác. Norton Utilities, sản phẩm của công ty Symantec, là một bộ sưu tập các chương trình tiện ích được dùng rất phổ biến. Bộ chương trình này có thể sửa chữa những dữ liệu trên các đĩa từ bị hỏng, bảo mật dữ

liệu một cách chắc chắn hơn bằng cách giấu kín các tệp hay giúp người dùng giải quyết các vấn đề trực tiếp của ổ đĩa.

+ **Chương trình điều khiển thiết bị:** là các phần mềm giúp hệ điều hành có thể nhận biết và điều khiển hoạt động của các thiết bị phần cứng. Khi nâng cấp, mở rộng thêm một thiết bị ngoại vi mới cho máy tính như ổ đĩa DVD, card màn hình, morden... trước hết hệ điều hành sẽ tự động tìm kiếm chương trình điều khiển cho thiết bị tương ứng ở trong thư viện trình điều khiển thiết bị có sẵn của hệ điều hành. Nếu tìm thấy, hệ điều hành sẽ tự động cài đặt phần mềm điều khiển và thiết bị mới có thể được nhận biết và hoạt động được ngay. Nếu không tìm thấy (có thể là thiết bị mới quá hay là lạ quá) thì người dùng phải chủ động cài đặt chương trình điều khiển trên đĩa CD đi kèm để hệ thống có thể nhận biết và khai thác sử dụng thiết bị mới.

+ **Chương trình dịch:** là chương trình có chức năng dịch các chương trình viết bằng ngôn ngữ thuật toán sang ngôn ngữ máy để máy tính có thể hiểu, xử lý được và ngược lại, dịch lại kết quả xử lý của máy tính sang ngôn ngữ bậc cao và chuyển tới người dùng. Mỗi một ngôn ngữ lập trình thì đều đi kèm theo nó một chương trình dịch.

❖ **Phần mềm ứng dụng:**

Là những phần mềm được làm ra để đáp ứng một yêu cầu nào đó của người dùng. Phần mềm ứng dụng cũng được chia làm 4 loại:

+ **Phần mềm nâng suất:** giúp nâng cao năng suất và hiệu quả làm việc của người dùng. Một số phần mềm loại này như: hệ soạn thảo văn bản giúp người dùng biên soạn các chế bản điện tử nhanh hơn, đẹp hơn và chuyên nghiệp hơn; các chương trình bảng tính điện tử giúp người dùng lập và xử lý các bảng dữ liệu phức tạp trong một thời gian ngắn nhất; các phần mềm đồ họa giúp người dùng sáng tác hay biên tập các tác phẩm đồ họa có chất lượng thẩm mỹ cao trong một khoảng thời gian ngắn nhất...

+ **Phần mềm kinh doanh:** là phần mềm có chức năng quản lý các hoạt động, các giao dịch phát sinh trong quá trình hoạt động của các doanh nghiệp sản xuất hàng hoá dịch vụ; giúp các doanh nghiệp hoàn thành những nhiệm vụ xử lý thông tin có tính chất thủ tục, lặp đi lặp lại hàng ngày, hàng tháng, hàng quý hay hàng năm. Phần mềm kinh doanh điển hình như phần mềm quản lý tài chính, kê toán; phần mềm quản trị sản xuất...

+ **Phần mềm giáo dục, tham khảo:** là phần mềm giúp cung cấp những

kiến thức, thông tin cho người dùng về một lĩnh vực nào đó; giúp người dùng học thêm về một chủ đề nào đó hay là giúp tra cứu về một đối tượng, một sự kiện hay một chủ đề bất kỳ. Điện hình như các phần mềm dạy học; các phần mềm từ điển; các phần mềm bách khoa toàn thư điện tử...

+ Phần mềm giải trí: là phần mềm người dùng thư giãn, giải trí. Bao gồm các trò chơi, các phần mềm xem phim, nghe nhạc...

1.2.3. Đặc điểm riêng của phần mềm so với các sản phẩm khác

- Là sản phẩm của ngành công nghiệp phần mềm, phần mềm hàm chứa một lượng tri thức rất lớn được tạo ra chủ yếu dựa vào tri thức và trí tuệ của người sản xuất phần mềm. Phần mềm không bao gồm các nguyên liệu thô như sắt, thép, nhựa, gỗ... như các sản phẩm thông thường khác.

- Không bị khấu hao hình theo thời gian. Mà ngược lại phần mềm có thể làm tăng giá trị cho đối tượng sử dụng nó lên nhiều lần.

- Dễ dàng nhân bản bởi bản chất phần mềm được lưu trữ dưới dạng các tệp dữ liệu trên các phương tiện nhớ của máy tính. Vì thế, nó dễ dàng được nhân bản nếu cần.

- Dễ dàng chuyển giao vì phần mềm không có kích thước và khối lượng giống như các sản phẩm khác. Nó có thể được sao chép và truyền trên mạng máy tính một cách nhanh gọn.

1.2.4. Các tiêu thức để đánh giá một phần mềm tốt

Đứng trên quan điểm người dùng cuối, một phần mềm tốt được đánh giá dựa trên những tiêu thức sau:

- Đáp ứng đầy đủ yêu cầu của người dùng. Đây là tiêu chuẩn đầu tiên khi đánh giá, nghiệm thu một phần mềm. Để được đánh giá tốt, phần mềm phải đáp ứng đầy đủ hệ thống các yêu cầu của người dùng đặt ra lúc ban đầu.
- Chứa ít lỗi tiềm tàng (tính ổn định). Phần mềm phải lường trước được các tình huống có thể gây ra lỗi và có biện pháp ngăn chặn hay là cảnh báo kịp thời.
- Giá thành không vượt quá dự kiến ban đầu. Phù hợp với dự trù kinh phí tương ứng với các chức năng đã được đặt hàng.
- Dễ vận hành, điều khiển và khai thác sử dụng (thân thiện với người sử dụng).

- Có đầy đủ tài liệu về thiết kế và hướng dẫn sử dụng. Cho phép người dùng có được những thông tin trợ giúp thích hợp, tương ứng với chức năng của phần mềm mà họ đang khai thác. Hơn nữa, khi có đầy đủ tài liệu về thiết kế cho phép khi có nhu cầu người ta có thể tiếp cận nhanh chóng tới cấu trúc bên trong cũng như tu tướng, phong cách thiết kế của phần mềm để có thể thực hiện những sự sửa đổi hay nâng cấp phù hợp.
- Có tính bảo mật cao, có khả năng ngăn chặn những sự truy nhập hay khai thác không được phép. Khả năng phân quyền chi tiết cho từng người sử dụng phù hợp với yêu cầu và nhiệm vụ của họ.

1.3. Các bước tổng quát của quy trình phân tích, thiết kế và xây dựng phần mềm

Phương pháp được trình bày ở đây có 7 giai đoạn. Mỗi giai đoạn bao gồm một dãy các công đoạn được liệt kê kèm theo dưới đây. Cần phải lưu ý rằng từ đây trở đi cuối mỗi giai đoạn là phải kèm theo việc ra quyết định về việc tiếp tục hay chấm dứt sự phát triển hệ thống. Quyết định này được trợ giúp dựa vào nội dung báo cáo mà phân tích viên hoặc nhóm phân tích viên trình bày cho các nhà sử dụng. Phát triển hệ thống là một quá trình lặp. Tuỳ theo kết quả của một giai đoạn có thể, và đôi khi là cần thiết, phải quay về giai đoạn trước để tìm cách khắc phục những sai sót. Một số nhiệm vụ được thực hiện trong suốt quá trình; đó là việc lập kế hoạch cho giai đoạn tới, kiểm soát những nhiệm vụ đã hoàn thành, đánh giá dự án và lập tài liệu về hệ thống và về dự án. Và sau đây là mô tả sơ lược các giai đoạn của việc phát triển hệ thống thông tin.

Bước 1: Đánh giá yêu cầu

Đây là bước đầu tiên và quan trọng nhất mà kết quả của nó có ý nghĩa quyết định tới sự thành công của toàn bộ dự án. Nó có mục đích tìm hiểu về hệ thống đang tồn tại, đánh giá nhu cầu của sự phát triển đối với sự phát triển của hệ thống mới với các khía cạnh:

- Khả thi về mặt kỹ thuật để bảo đảm rằng mọi yêu cầu về chức năng của hệ thống mới đều có thể tìm được giải pháp hay kỹ thuật thực hiện
- Khả thi về mặt tài chính để đảm bảo mọi yêu cầu về chi phí cho việc sản xuất phần mềm đều được đáp ứng và phù hợp với khả năng, dự trù kinh phí của người dùng

- Khả thi về thời gian để đảm bảo rằng thời gian dự kiến cho việc thiết kế phần mềm là hợp lý và có thể được người dùng chấp nhận
- Khả thi về nhân lực đảm bảo rằng từ phía nhà sản xuất có đủ đội ngũ nhân lực trong việc xây dựng phần mềm đồng thời về phía người dùng cũng chuẩn bị được đội ngũ nhân lực cho việc khai thác sử dụng và vận hành hệ thống

Thông thường, bước đánh giá yêu cầu càng được thực hiện chi tiết và chính xác bao nhiêu thì các bước sau sẽ đơn giản và dễ thành công bấy nhiêu

Bước 2: Phân tích chi tiết

Phân tích chi tiết được tiến hành sau khi có sự đánh giá thuận lợi về yêu cầu. Những mục đích chính của phân tích chi tiết là hiểu rõ các vấn đề của hệ thống đang nghiên cứu, xác định những nguyên nhân đích thực của những vấn đề đó, xác định những đòi hỏi và những ràng buộc áp đặt đối với hệ thống và xác định mục tiêu mà hệ thống thông tin mới phải đạt được. Trên cơ sở nội dung báo cáo phân tích chi tiết sẽ quyết định tiếp tục tiến hành hay thôi phát triển một hệ thống mới. Để làm những việc đó giai đoạn phân tích chi tiết bao gồm các công đoạn sau đây.

- Lập kế hoạch phân tích chi tiết.
- Nghiên cứu môi trường của hệ thống đang tồn tại.
- Nghiên cứu hệ thống thực tại.
- Dưa ra chẩn đoán và xác định các yếu tố giải pháp.
- Đánh giá lại tính khả thi.
- Thay đổi đề xuất của dự án.
- Chuẩn bị và trình bày báo cáo phân tích chi tiết.

Bước 3: Thiết kế lô gíc

Bước này nhằm xác định tất cả các thành phần lô gíc của một hệ thống thông tin, cho phép loại bỏ được các vấn đề của hệ thống thực tế và đạt được những mục tiêu đã được thiết lập ở giai đoạn trước. Mô hình lô gíc của hệ thống mới sẽ bao hàm thông tin mà hệ thống mới sẽ sản sinh ra (nội dung của Outputs), nội dung của cơ sở dữ liệu (các tệp, các quan hệ giữa các tệp), các xử lý và hợp thức hóa sẽ phải thực hiện (các xử lý) và các dữ liệu sẽ được nhập vào (các Inputs). Mô hình lô gíc sẽ phải được những người sử dụng xem xét và chuẩn y. Thiết kế lô gíc bao gồm những công đoạn sau:

- Thiết kế cơ sở dữ liệu.
- Thiết kế xử lý.
- Thiết kế các luồng dữ liệu vào.
- Chỉnh sửa tài liệu cho mức lô gic.
- Hợp thức hóa mô hình lô gic.

Bước 4: Đề xuất các phương án của giải pháp

Mô hình lô gic của hệ thống mới mô tả cái mà hệ thống này sẽ làm. Khi mô hình này được xác định và chuẩn y bởi người sử dụng, thì phân tích viên hoặc nhóm phân tích viên phải nghiên về các phương tiện để thực hiện hệ thống này. Đó là việc xây dựng các phương án khác nhau để cụ thể hóa mô hình lô gic. Mỗi một phương án là một phác họa của mô hình vật lý ngoài của hệ thống nhưng chưa phải là một mô tả chi tiết. Tất nhiên là người sử dụng sẽ thấy dễ dàng hơn khi lựa chọn dựa trên những mô hình vật lý ngoài được xây dựng chi tiết nhưng chi phí cho việc tạo ra chúng là rất lớn.

Để giúp những người sử dụng lựa chọn giải pháp vật lý thoả mãn tốt hơn các mục tiêu đã định ra trước đây, nhóm phân tích viên phải đánh giá các chi phí và lợi ích (hữu hình và vô hình) của mỗi phương án và phải có những khuyến nghị cụ thể. Một báo cáo sẽ được trình lên những người sử dụng và một buổi trình bày sẽ được thực hiện. Những người sử dụng sẽ chọn lấy một phương án tỏ ra đáp ứng tốt nhất các yêu cầu của họ mà vẫn tôn trọng các ràng buộc của tổ chức. Sau đây là các công đoạn của giai đoạn đề xuất các phương án giải pháp:

- Xác định các ràng buộc tin học và ràng buộc tổ chức.
- Xây dựng các phương án của giải pháp.
- Đánh giá các phương án của giải pháp.
- Chuẩn bị và trình bày báo cáo của giai đoạn đề xuất các phương án giải pháp.

Bước 5: Thiết kế vật lý ngoài

Bước này được tiến hành sau khi một phương án giải pháp được lựa chọn. Thiết kế vật lý bao gồm hai tài liệu kết quả cần có: trước hết là một tài liệu bao chứa tất cả các đặc trưng của hệ thống mới sẽ cần cho việc thực hiện kỹ thuật; và tiếp đó là tài liệu dành cho người sử dụng và nó mô tả cả

phản thù công và cả những giao diện với những phần tin học hoá. Những công đoạn chính của thiết kế vật lý ngoài là:

- Lập kế hoạch thiết kế vật lý ngoài
- Thiết kế chi tiết các giao diện (vào/ra)
- Thiết kế cách thức tương tác với phần tin học hoá
- Thiết kế các thủ tục thủ công
- Chuẩn bị và trình bày báo cáo về thiết kế vật lý ngoài

Bước 6: Triển khai kỹ thuật hệ thống

Kết quả quan trọng nhất của bước thực hiện kỹ thuật là phần tin học hoá của hệ thống thông tin, có nghĩa là phần mềm. Những người chịu trách nhiệm về giai đoạn này phải cung cấp các tài liệu như các bản hướng dẫn sử dụng và thao tác cũng như các tài liệu mô tả về hệ thống. Các hoạt động chính của việc triển khai thực hiện kỹ thuật hệ thống là như sau:

- Lập kế hoạch thực hiện kỹ thuật
- Thiết kế vật lý trong
- Lập trình
- Thủ nghiệm hệ thống
- Chuẩn bị tài liệu

Bước 7: Cài đặt và khai thác

Cài đặt hệ thống là pha trong đó việc chuyển từ hệ thống cũ sang hệ thống mới được thực hiện. Đề việc chuyển đổi này được thực hiện với những va chạm ít nhất, cần phải lập kế hoạch một cách cẩn thận. Giai đoạn này bao gồm các công đoạn:

- Lập kế hoạch cài đặt
- Chuyển đổi
- Khai thác và bảo trì
- Đánh giá

1.4. Các thành viên chính của dự án phát triển hệ thống thông tin

Số lượng các thành viên tham gia vào dự án phát triển hệ thống thay đổi tùy theo quy mô và sự phức tạp của dự án. Sau đây là một dạng cấu hình

tương đối phổ biến do Y. C. Gagnon đưa ra. Cần lưu ý rằng cùng một con người có thể tuỳ theo hoàn cảnh mà thuộc vào một hay nhiều nhóm:

- **Những người ra quyết định.** Họ kiểm soát các nguồn lực được dùng trong hệ thống. Họ có quyền lực tác động vào việc phát triển hệ thống. Họ có tiếng nói quyết định trong việc lựa chọn mục tiêu cũng như tiến hành thực hiện hệ thống mới. Đó là bộ phận lãnh đạo cao nhất của tổ chức.

- **Những nhà quản lý** trông coi quá trình phát triển và hoặc vận hành hệ thống. Họ là đại diện, ở thứ bậc thấp hơn, của những người ra quyết định. Họ lao động trong sự hợp tác với các phân tích viên.

- **Phân tích viên và thiết kế viên** phân tích thiết kế và cài đặt hệ thống trong sự cộng tác với các nhà ra quyết định và các nhà quản lý.

- **Người sử dụng cuối** tương tác với hệ thống theo sự cần thiết hoặc tùy chọn. Họ sử dụng đầu ra của hệ thống. Họ tiếp xúc trực tiếp với hệ thống trong những khoảng thời gian ngắn. Đó là những người sử dụng tại các giao diện vào/ra của hệ thống, những nhà quản lý mà hệ thống đang được xây dựng cho học.

- **Người sử dụng - thao tác viên** là những người mà vai trò nhiệm vụ của họ là gắn liền với hệ thống khi nó trở thành tác nghiệp được. Họ tạo ra những đầu vào (*Inputs*) hoặc nhận các đầu ra từ hệ thống (*Outputs*) để rồi đưa cho những người sử dụng cuối.

- **Những người sử dụng gián tiếp.** Đó là những người chịu ảnh hưởng gián tiếp của hệ thống như sử dụng nguồn lực hiếm, ảnh hưởng xã hội ...

- **Các lập trình viên** chịu trách nhiệm xây dựng các chi tiết của cấu trúc hệ thống.

- **Các hướng dẫn viên** hướng dẫn cho những người sử dụng thao tác viên hoặc các nhóm người khác cách thức sử dụng hệ thống.

1.5. Vai trò phân tích viên hệ thống

Phân tích viên hệ thống nói chung không phải là người duy nhất có trách nhiệm trong một dự án phát triển hệ thống. Trong trường hợp một hệ thống rất lớn người ta có thể có một đội ngũ gồm một chủ dự án, một số phân tích viên, một số người sử dụng, một số lập trình viên và trợ lý quản trị dự án. Đối với hệ thống nhỏ có thể chỉ có một người đóng đồng thời vai trò chủ dự án, phân tích viên, lập trình viên và thư ký. Để thực hiện những chức năng của mình một cách dễ dàng, phân tích viên phải có kiến thức

trong nhiều lĩnh vực như quản lý, kinh doanh, hệ thống thông tin, kỹ thuật CNTT ...

Cán bộ phân tích phải hiểu rõ công việc mà người sử dụng phải làm, những vấn đề và những khó khăn họ gặp. Để làm điều đó việc hiểu biết các phương pháp thu thập thông tin và các phương pháp trình bày hệ thống là cần thiết ... Phân tích viên cũng cần phải biết đề xuất các giải pháp cho vấn đề gặp phải và phải biết thiết kế lô gíc cho hệ thống tương ứng. Phân tích viên cũng cần phải biết chuyển đổi các yếu tố lô gíc sang thành các phương án cụ thể và đánh giá chi phí và lợi ích của các phương án về mặt tài chính cũng như xã hội. Anh ta cũng phải biết chuyển những đề xuất thành các đặc tả cụ thể chính xác để lập trình viên có thể thực hiện chúng hoặc tự thực hiện lấy. Những hiểu biết về lập trình, thử nghiệm hệ thống và những phương pháp cài đặt cũng là cần thiết hữu ích.

Ngoài ra, phân tích viên phải có một số phẩm chất nhân bản cao liên quan chặt chẽ với hoàn cảnh của dự án đang tiến hành. Trong thực tế, việc đưa ra một dự án thường nảy sinh sự không an tâm ở những người sử dụng có liên quan. Một số người nhìn thấy nó như một phương tiện để cắp trên đánh giá năng lực, một số khác ngại thay đổi thói quen, một số khác sợ mất quyền lực và cũng có người thấy việc làm của mình bị đe dọa. Những nỗi sợ hãi và không an tâm đó đôi khi dẫn tới việc người sử dụng kháng cự lại ngay lập tức sự thay đổi và giảm khả năng hợp tác có hiệu quả trong việc nghiên cứu. Chẳng có gì khó chịu hơn đối với người sử dụng khi người phân tích tạo ra một cảm tưởng là anh ta hiểu biết tốt hơn người sử dụng cách thức thực hiện nhiệm vụ của họ.

Trong những hoàn cảnh như vậy, việc phân tích viên thể hiện những phẩm chất nhân văn sẽ có những hiệu quả rất lớn.

1.6. Một số nguy cơ đối mặt trong quá trình sản xuất phần mềm

Trong quá trình sản xuất và xây dựng phần mềm thì một công ty phần mềm thường phải đối mặt với một số nguy cơ, điều đó làm chất lượng phần mềm giảm đi

- ❖ Không có phương pháp mô tả rõ ràng, đủ nghĩa, yêu cầu của người dùng dẫn tới khi bàn giao sản phẩm có thể phát sinh các trục trặc
- ❖ Với những phần mềm quy mô lớn, với những tài liệu đặc tả cố định trong thời gian dài vì vậy khó đáp ứng nhu cầu thay đổi người dùng một cách kịp thời trong thời gian đó

- ❖ Nếu không có phương pháp luận thiết kế nhất quán mà thực hiện thiết kế theo cách riêng của công ty hay của nhóm lập trình sẽ dẫn đến việc suy giảm chất lượng phần mềm do phải phụ thuộc quá nhiều vào con người
- ❖ Nếu không có chuẩn trong việc thiết kế tài liệu về quy trình sản xuất phần mềm thì những đặc điểm không rõ ràng và làm giảm chất lượng phần mềm
- ❖ Nếu không thực hiện việc kiểm thử tính đúng đắn của phần mềm trong từng giai đoạn mà chỉ thực hiện ở giai đoạn cuối và phát hiện lỗi thì việc bàn giao sản phẩm thường không đúng hạn
- ❖ Nếu coi trọng lập trình hơn thiết kế thường làm giảm chất lượng phần mềm
- ❖ Phần lớn trong quá trình phát triển phần mềm có nhiều thao tác do con người thực hiện → năng suất lao động giảm
- ❖ Không chứng minh được tính đúng đắn của phần mềm do vây độ tin cậy phần mềm giảm
- ❖ Tiêu chuẩn đánh giá một phần mềm tốt không thể định lượng hóa, do vây khó đánh giá phần mềm đúng đắn hay không
- ❖ Khi đầu tư nhân lực lớn vào việc bảo trì phần mềm làm giảm hiệu suất lao động của nhân viên
- ❖ Công việc bảo trì kéo dài làm giảm chất lượng tư liệu và ảnh hưởng xấu đến những việc khác
- ❖ Việc quản lý dự án lỏng lẻo kéo đến việc quản lý lịch trình không rõ ràng
- ❖ Không có tiêu chuẩn ước lượng nhân lực của dự án → kéo dài thời hạn và vượt kinh phí dự án

CÂU HỎI ÔN TẬP CHƯƠNG 1

1. Hãy trình bày hiểu biết của bạn về tình hình tin học hóa công tác quản lý tại các doanh nghiệp nước ta hiện nay.
2. Tại sao nói “Tin học hóa công tác quản lý là giải pháp phát triển cũng như là xu thế của các doanh nghiệp Việt Nam trong giai đoạn hội nhập kinh tế quốc tế hiện nay”, bạn hãy bình luận về phát biểu trên.
3. Những mối lợi thu được khi tin học hóa công tác quản lý tại doanh nghiệp?
4. Những người được hưởng lợi từ việc tin học hóa công tác quản lý?
5. Những vấn đề ngăn cản doanh nghiệp khi tiến hành tin học hóa công tác quản lý?
6. Những nguy cơ phải đối mặt của doanh nghiệp khi thực hiện tin học hóa công tác quản lý?
7. Những nguyên nhân dẫn đến việc phát triển hệ thống thông tin mới?
8. Các thành viên tham gia dự án phát triển một HTTT?
9. Khái niệm phần mềm, công nghệ phần mềm?
10. Phân loại phần mềm?
11. Các tiêu chuẩn đánh giá một phần mềm thương mại tốt?
12. Các bước của quá trình phân tích thiết kế xây dựng và cài đặt một HTTT

Chương 2

NGÔN NGỮ LẬP TRÌNH VISUAL BASIC 6.0

2.1. Giới thiệu chung

2.1.1. Giới thiệu Visual Basic

Là một sản phẩm trong bộ Visual Studio của hãng Microsoft, ra đời năm 1991 với phiên bản đầu tiên visual Basic 1.0. Cho đến năm 1998 phiên bản Visual Basic 6.0 ra đời và sau đó chuyển sang một thế hệ ngôn ngữ lập trình mới Visual Basic .NET. Còn với phiên bản Visual Basic 6.0 (VB6) Microsoft phát hành các bản Service Pack với mục đích để sửa lỗi và hoàn thiện hơn. Tại thời điểm biên soạn giáo trình này đang có phiên bản Visual Basic 6 Service Pack 6.

Các ấn bản khác nhau của VB 6

+ **Learning Edition:** là phiên bản cơ bản nhất, nó cho phép viết nhiều kiểu ứng dụng khác nhau, tuy nhiên nó không có 1 số công cụ điều khiển có trong các phiên bản khác

+ **Professional Edition:** thiết kế cho người dùng chuyên nghiệp, nó chứa tất cả các tính năng và công cụ có trong phiên bản Learning Edition và có bổ sung thêm thư viện và các công cụ điều khiển.

+ **EnterPrise Edition:** đây là phiên bản đầy đủ nhất dành cho các phân tích ứng dụng chuyên nghiệp. Nó chứa các công cụ để hỗ trợ lập trình theo nhóm

Visual Basic là một ngôn ngữ lập trình đa năng, sử dụng để xây dựng các phần mềm hoạt động trong môi trường window hay trên mạng Internet. Nó là sự kế thừa của ngôn ngữ lập trình Basic với những ưu điểm chính như sau:

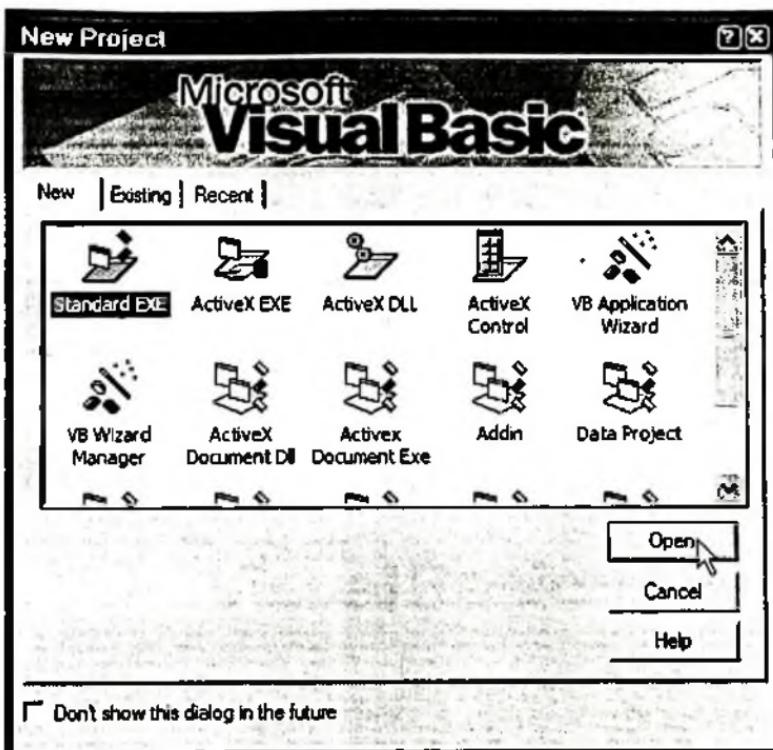
+ Bao gồm mọi đặc điểm của Basic nên rất quen thuộc để sử dụng

+ Cung cấp nhiều công cụ điều khiển có sẵn để hỗ trợ lập trình viên, nhất là trong lập trình ứng dụng CSDL

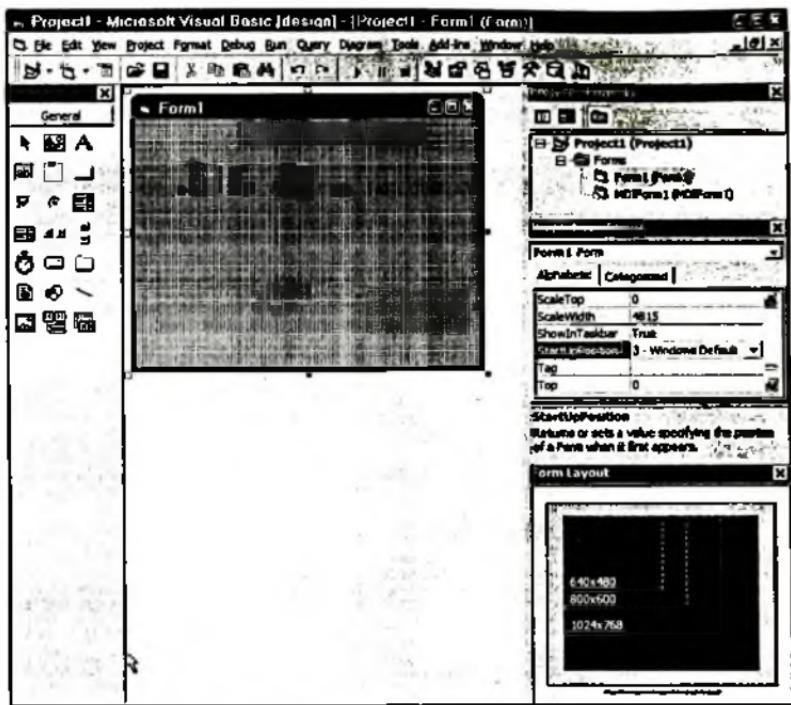
+ Là ngôn ngữ có tính trực quan rất cao, có cấu trúc logic chặt chẽ ở mức độ vừa phải, rất dễ dàng học được thành thạo

2.1.2. Giao diện làm việc của VB

- Cửa sổ đầu tiên, lựa chọn các kiểu dự án để phát triển



Chúng ta sẽ chọn Standard EXE để làm hiện ra giao diện của môi trường phát triển tích hợp Integrated Development Environment (IDE) của Visual Basic.



- Thanh tiêu đề (Titlebar)

Project1 - Microsoft Visual Basic [design] - {Form1} (Form)

Là thanh trên cùng của màn hình giao diện chính của VB. Thanh này chứa tên của dự án đang được thiết kế.

- Thanh thực đơn (MenuBar)

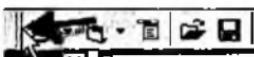
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help

Chứa đầy đủ các mục chức năng mà bạn sử dụng để làm việc với VB6, kể cả các menu để truy cập các chức năng đặc biệt dành cho việc lập trình chẳng hạn như Project, Format, hoặc Debug. Trong Menu Add-Ins có Add-Ins Manager cho phép bạn gắn thêm những menu con để chạy các chương trình tiện ích cho việc lập trình.

Thanh công cụ(Toolbar)



Các toolbars có hình các icons cho phép bạn click để thực hiện công việc tương đương với dùng một menu command, nhưng nhanh và tiện hơn. Bạn dùng menu command **View | Toolbars** (click lên menu command View cho popupmenu hiện ra rồi click command con Toolbars) để làm cho các toolbars hiện ra hay biến mất đi. Bạn có thể thay đổi vị trí một toolbar bằng cách nắm vào hai gạch vertical nằm bên trái toolbar rồi dời toolbar đi chỗ khác (nắm ở đây nghĩa là dể trỏ chuột lên chỗ chấm đó trong hình phía dưới rồi bấm xuống và giữ nút bên trái của mouse, trong khi kéo pointer đi nơi khác).



Ngoài ra bạn cũng có thể sửa đổi các toolbars theo ý thích bằng cách dùng Menu command **View | Toolbars | Customize...**

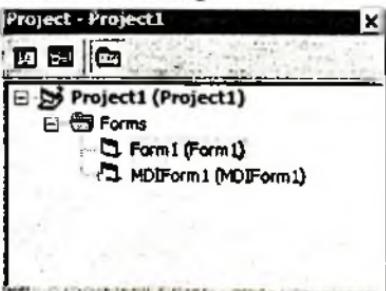
- Hộp công cụ (Toolbox)

Đây là hộp đồ nghề với các công cụ, gọi là controls, mà bạn có thể đặt lên các form trong lúc thiết kế (design). Nếu Toolbox biến mất, bạn có thể display nó trở lại bằng cách dùng menu command **View | Toolbox**. Bạn có thể khiến toolbox display nhiều controls hơn bằng cách chọn **Components...** từ context menu (chọn Toolbox rồi bấm nút phải của mouse để display context menu) hay dùng menu command **Project | Components**. Ngoài việc trình bày Toolbox mặc định, bạn có thể tạo cách trình bày khác bằng cách chọn **Add Tab...** từ context menu và bổ sung các control cho tab từ kết quả.



- Project Explorer

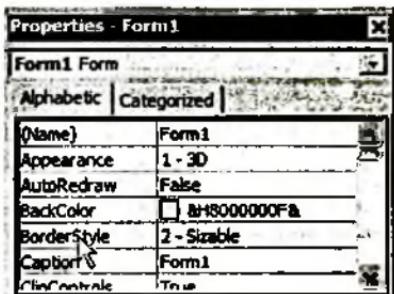
Sẽ liệt kê các forms và các modules trong project hiện hành của bạn. Một **project** là sự tập hợp các files mà bạn sử dụng để tạo một trình ứng dụng. Tức là, trong VB6, khi nói viết một program có nghĩa



là triển khai một project. Để hiển thị cửa sổ Project Explorer bạn vào Menu View | Project Explorer hoặc bấm tổ hợp phím Ctrl+R

- Properties window

Liệt kê các thuộc tính của các forms hoặc controls được chọn. Một property là một đặc tính của một object chẳng hạn như size, caption, hoặc color. Khi bạn sửa đổi một property bạn sẽ thấy hiệu quả ngay lập tức, thí dụ thay đổi property Font của một Label sẽ thấy Label ấy được display bằng Font chữ mới. Khi bạn chọn một Property của control hay form trong Properties window, phía bên phải ở chỗ value của property có thể display ba chấm (...) hay một tam giác chia xuống. Bấm vào đó để display một dialog cho bạn chọn value. Bấm phím F4 để hiển thị cửa sổ thuộc tính của đối tượng đang được lựa chọn.

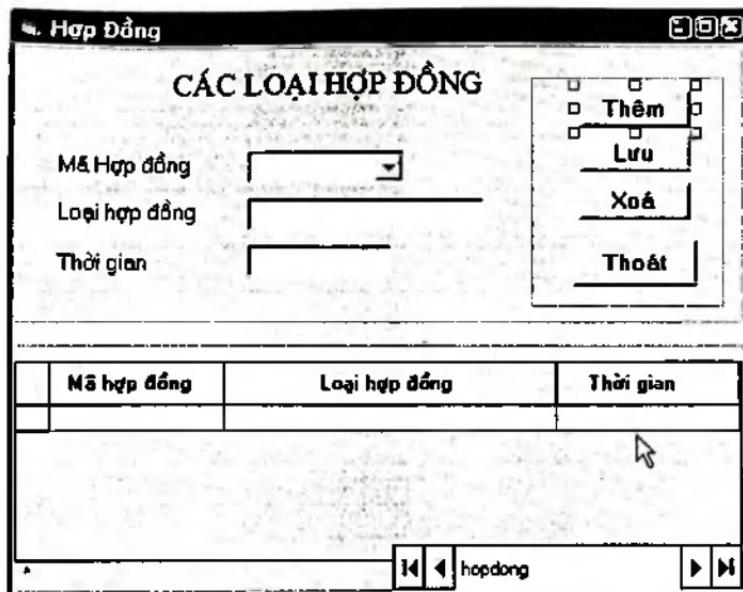


- Form Layout

Bạn dùng form Layout để chỉnh vị trí của các forms khi form hiện ra lần đầu lúc chương trình chạy. Dùng context command **Resolution Guides** để thấy nếu dùng một màn ảnh với độ mịn (resolution) tệ hơn, thí dụ như 640 X 480, thì nó sẽ nhỏ như thế nào.

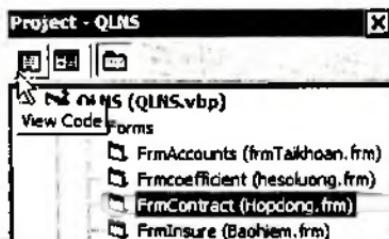


- Form Designer



Dùng để thiết kế giao diện lập trình. Bạn bổ sung các controls, các đồ họa (graphics), các hình ảnh từ hộp Toolbox vào form để tạo ra chức năng mà bạn muốn. Mỗi form trong trình ứng dụng của bạn có designer form riêng của nó. Khi bạn maximise một form designer, nó chiếm cả khu làm việc. Muốn làm cho nó trở lại cỡ bình thường và đồng thời để thấy các form designers khác, click nút Restore Window ở góc bên phải, phía trên.

- View Code button

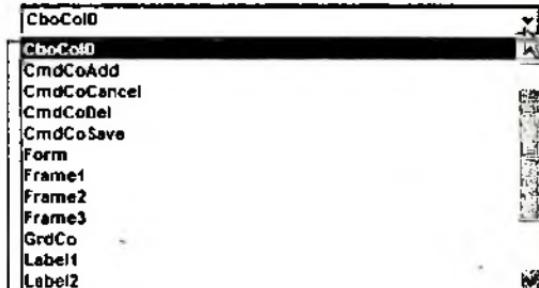


Cửa sổ Code là nơi bạn thực hiện lập trình để đáp ứng các sự kiện có thể xảy ra khi ứng dụng hoạt động.

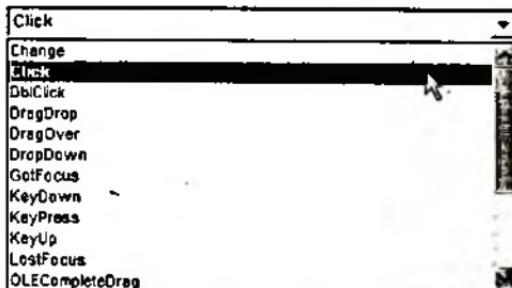
```
Dim RsCo As New ADODB.Recordset
Dim rsNeptudong As New ADODB.Recordset
Private Sub chonUpdate()
    GrdCo.AllowUpdate = True
End Sub
Private Sub khongchonUpdate()
    GrdCo.AllowUpdate = False
End Sub
Private Sub CboCoID_Click()
    Dim str As String
    str = "SELECT TenHD, Thoigian FROM LoaiHD WHERE MaHD = " & CboCoID.Text & ""
    rsNeptudong.Open str, conn
    TxCoName.Text = rsNeptudong!TenHD
    TxCoTime.Text = rsNeptudong!Thoigian
    rsNeptudong.Close
    If CboCoID.Text <> "" Then
        CmdCoDel.Enabled = True
    End If
End Sub
```

Phía trên cửa sổ Code bạn thấy có 2 hộp menu đồ xuống.

Tại hộp menu bên trái bạn lựa chọn đối tượng muốn lập trình.



Tại hộp menu bên phải bạn lựa chọn sự kiện tương ứng với đối tượng đã được lựa chọn trong hộp menu bên phải sẽ được lập trình



Ví dụ: Nếu muốn lập trình cho sự kiện nút “Thêm mới” được lựa chọn (được bấm). Tại cửa sổ menu bên trái ta chọn tên của nút “Thêm mới” là CmdCoAdd và tại cửa sổ menu đồ xuống bên phải tương ứng ta chọn sự kiện “Click”

Trong Code window bạn có thể chọn display tất cả Sub của code cùng một lúc như trong hình hay display mỗi lần chỉ một Sub bằng cách click button có hình ba dòng nằm ở góc bên trái phía dưới.

The screenshot shows the Microsoft Visual Basic IDE in design mode. The title bar reads "QLNS - Microsoft Visual Basic [design] - [FrmContr...]" with standard window controls. The menu bar includes File, Edit, View, Project, Format, Debug, Run, Query, Diagram, Tools, Add-Ins, Window, and Help. The toolbar below has icons for opening files, saving, printing, and other common operations. The main code editor window has two sub-editors visible. The top one is titled "CmdCoAdd" and contains the following VBA code:

```
Private Sub CmdCoAdd_Click()
    Dim traloi As Integer
    traloi = MsgBox("Bạn muốn thêm loại hợp đồng?", vbInformation)
    If traloi = vbNo Then
        Exit Sub
    Else
        CboCoID.Text = ""
        TxtCoName.Text = ""
        TxtCoTime.Text = ""
        CboCoID.SetFocus
    End If
    'CmdCoSave.Enabled = True
End Sub
```

The bottom sub-editor is titled "Click" and contains:

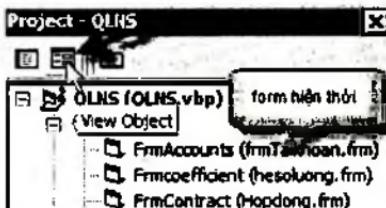
```
Private Sub CmdCoCancel_Click()
    Unload Me
    MDIMain.Show
End Sub
```

Immediate Window

Dùng để gõ rồi (debug) trình ứng dụng của bạn. Bạn có thể display dữ kiện trong khi chạy chương trình ứng dụng. Khi chương trình đang tạm ngừng ở một break point, bạn có thể thay đổi giá trị các biến hay chạy một dòng chương trình để hiển thị cửa sổ chọn menu View | Immediate Window

```
Immediate
?txtgID.Text
?KTOID
?TXXTNgPass.Text
?thID
?ID='tinhoc'
?100+49
149
```

- View form button



Click lên nút này để hiện ra cửa sổ thiết kế giao diện của Form hiện thời.

2.1.3. Các thành phần của dự án phần mềm (Project)

Dự án là một sản phẩm phần mềm có thể hoạt động như một ứng dụng độc lập chạy trong môi trường Window hay là mạng Internet. Mỗi dự án được tạo nên bởi các thành phần như sau:

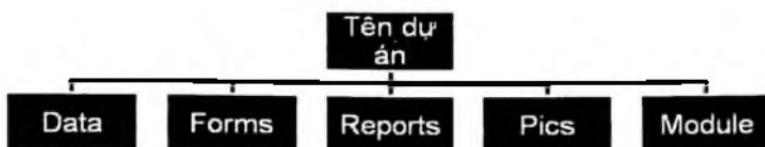
- Form (giao diện) là các màn hình giao tiếp giữa chương trình và người sử dụng. Mỗi form trong dự án sẽ tương ứng với một tệp có phần mở rộng là .frm

- Báo cáo (Report) là sản phẩm đầu ra của một dự án phần mềm, là kết quả của quá trình xử lý dữ liệu, được tổ chức theo mẫu theo quy định. Báo cáo có thể được thiết kế với công cụ tích hợp sẵn trong VB là Data Report hay thông qua một phần mềm chuyên nghiệp trong việc tạo báo cáo ví dụ như Crystal Report.

- CSDL (Database) là nơi chứa các dữ liệu đầu vào cũng như đầu ra phục vụ cho hoạt động của dự án phần mềm. CSDL được tạo lập và quản trị bởi 1 hệ CSDL độc lập và được liên kết khai thác sử dụng thông qua các công cụ của Visual Basic.

- Module: đây là một thư viện các hàm hay thủ tục. Nó được sử dụng để đáp ứng nhu cầu cho việc cập nhật và xử lý dữ liệu của phần mềm. Nó có thể được chia sẻ hay thừa kế giữa các dự án phần mềm. Tệp chứa Module có phần mở rộng là *.bas hoặc *.cls

Ghi chú: các thành phần khác nhau trong một dự án nên được lưu trữ trong các thư mục con tương ứng nằm trong thư mục thống nhất của dự án.



Tệp điều khiển của dự án để trực tiếp trong thư mục của dự án và có phần mở rộng *.vbp

2.1.4. Một số thao tác cơ bản trong Visual Basic

2.1.4.1. Tạo dự án

Mỗi lần khởi động Visual Basic, ta sẽ thấy hộp thoại New Project. Từ đây, ta có thể chọn loại dự án mà ta muốn tạo và ấn Open.

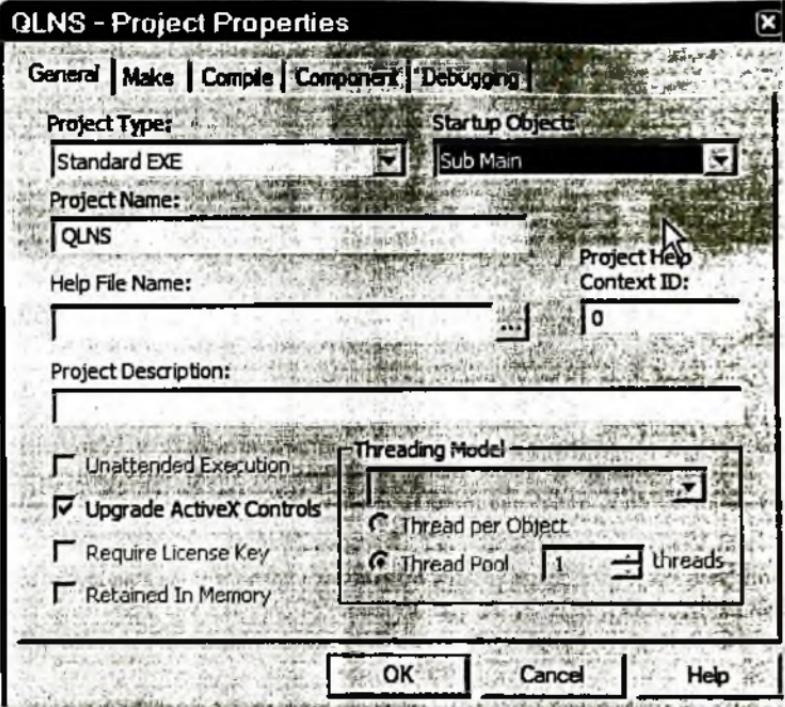
Khi Visual Basic đã có sẵn, ta có thể tạo dự án bằng cách: từ menu File chọn New Project. Hộp thoại New Project xuất hiện, ta chọn loại dự án cần thiết và nhấn OK.

2.1.4.2. Đổi thuộc tính dự án

Một số thông tin liên quan đến dự án như tên dự án, số phiên bản, chuỗi ký tự dùng hiển thị trên thanh tiêu đề khi ứng dụng hoàn thành. Ta có thể xem các thông tin khác trong hộp thoại Project Properties.

Hộp thoại Project Properties

1. trong cửa sổ Project Explorer, nhấn nút phải chuột lên tệp tin dự án
2. trong menu ngữ cảnh, Chọn Project Name Properties. Hộp thoại Project Properties xuất hiện
3. Hoặc là từ menu Properties, chọn Project Name Properties

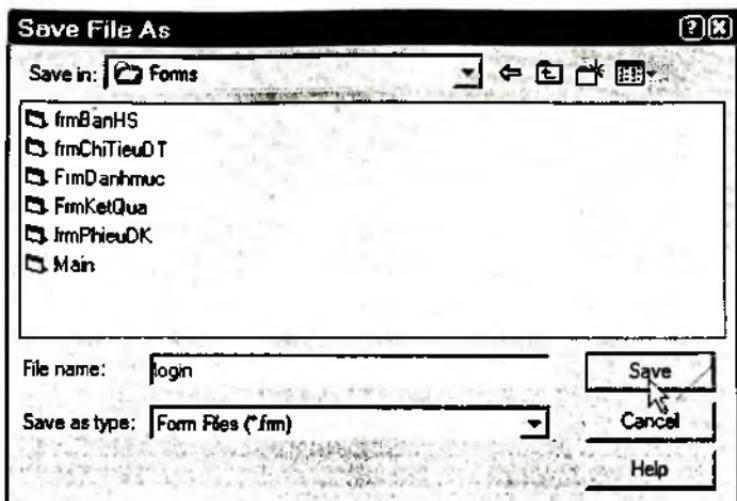


2.1.4.3. Lưu và đặt tên dự án

Lưu dự án

Khi lưu dự án, từng tập tin trong dự án sẽ được lưu trước kẽ tiếp là tập tin dự án. Trong lần lưu dự án đầu tiên, Visual Basic đề nghị tên cho từng tập tin, thường nó lấy tên biểu mẫu và có phần mở rộng tùy thuộc vào loại tập tin.

1. từ menu file chọn Save Project
2. nếu đây là lần đầu lưu dự án hoặc ta vừa thêm một biểu mẫu hoặc module, hộp thoại save File as xuất hiện lần lượt cho từng tập tin
3. Chú ý: tùy thuộc vào loại tập tin của dự án mà bạn lưu nó vào thư mục riêng biệt. Như vậy sẽ dễ quản lý hơn.



Đổi tên

Ta không nhất thiết dùng tên mà Visual Basic đề nghị, mà có thể đặt tùy ý. Tuy nhiên nên đặt tên sao cho gợi nhớ. Quy tắc đặt tên sẽ được trình bày trong phần sau:

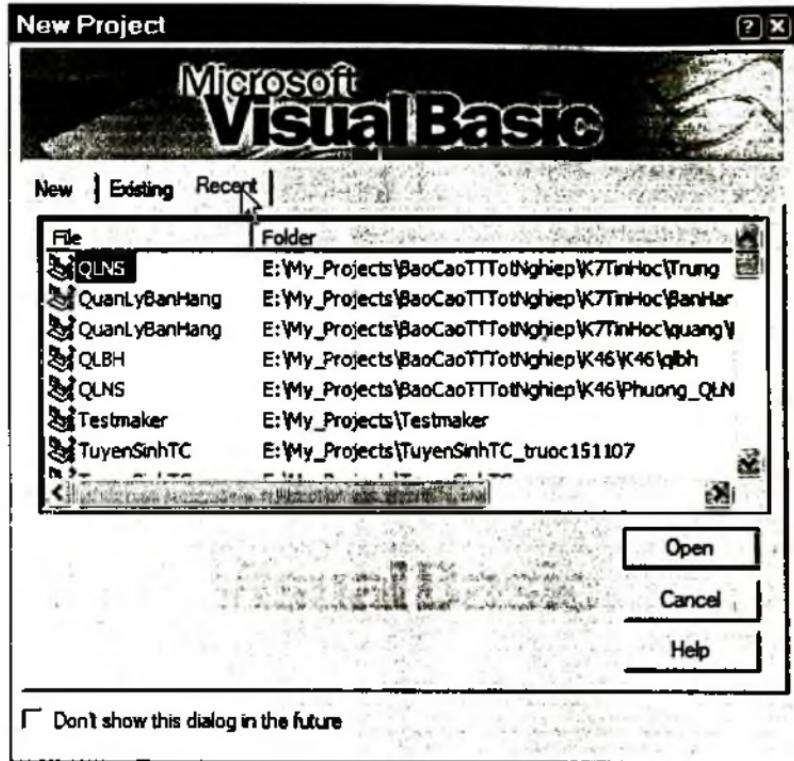
1. Dưa vào một tên và ấn nút save
2. Tập tin cuối cùng được lưu là tập tin dự án. Nếu ta đã đặt tên cho dự án thông qua hộp thoại project properties, Visual Basic sẽ tự động đề nghị Project_Name.vbp. Lúc này, ta có thể đổi lại tên khác tuỳ thích, ví dụ như SaveTest.vbp.

2.1.4.4. Mở dự án có sẵn

Ta có một số dự án đang làm việc. Khi khởi động Visual Basic, chọn menu File. Phần dưới menu liệt kê danh sách các dự án mới nhất mà ta đã làm việc, chọn dự án cần mở. Nếu dự án không xuất hiện trong danh sách, ta phải chỉ ra đường dẫn.

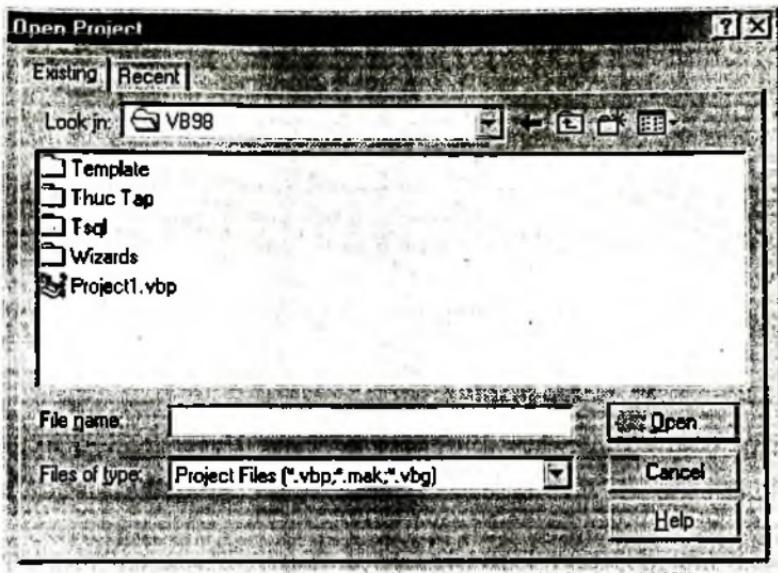
Mở tập tin vào lúc khởi động Visual Basic

Lần đầu khởi động Visual Basic, hộp thoại New project xuất hiện. Ta có thể chọn mở dự án mới nhất hoặc có sẵn trên đĩa ngay từ hộp thoại này. Nếu không muốn hộp thoại này xuất hiện mỗi lần khởi động Visual Basic, xoá đánh dấu trên hộp đánh dấu (checkbox) nằm ở bên dưới hộp thoại.



Mở dự án có sẵn

- a. Từ menu File, chọn Open Project.
- b. trên trang Existing, chuyển đến thư mục chứa dự án. Nếu dự án cần mở là dự án lưu gần nhất, chuyển sang trang Recent.
- c. chọn tên tệp tin dự án và nhấn OK.



2.1.4.5. Thêm xoá và lưu tập tin trong dự án

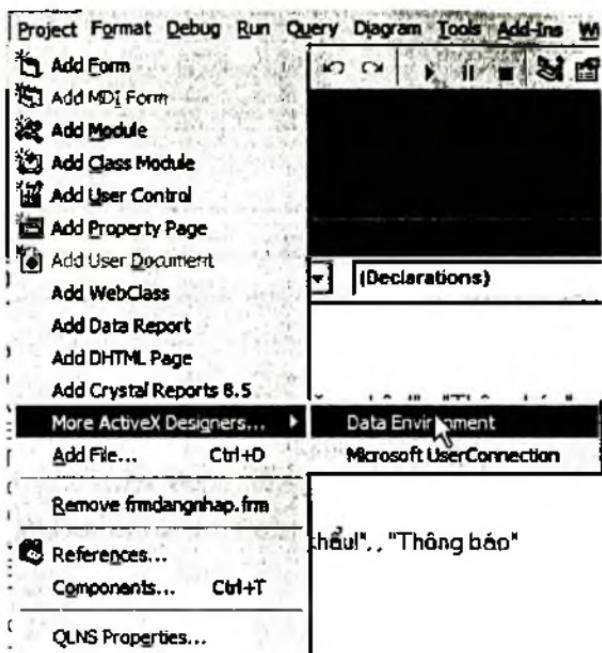
Thêm mới tập tin

Thông thường, một biểu mẫu cần nhiều biểu mẫu hoặc module. Ví dụ, muốn thêm hộp thoại About, ta cần thêm một biểu mẫu.

Dưới đây là các dạng tập tin có thể thêm vào dự án: bằng cách chọn trong menu Project

- Biểu mẫu (form): tập tin.frm chứa mô tả của một biểu mẫu và các điều khiển, kè cả các thuộc tính của chúng. Nó cũng chứa khai báo các hằng, biến và thủ tục.
- Lớp (Class) là một trong những tính năng quan trọng nhất của Visual Basic, được dùng trong lập trình hướng đối tượng để định nghĩa các khuôn mẫu cho các đối tượng.
- Module chuẩn chứa các khai báo kiểu, hằng, biến, thủ thuộc phạm vi public hoặc ở mức module.
- Tập tin tài nguyên: chứa hình ảnh, chuỗi ký tự và các dữ liệu khác ta có thể soạn thảo mà không cần sửa lại mã nguồn.

- Tài liệu ActiveX (.dob) tương tự biểu mẫu nhưng được hiển thị trong trình duyệt xem WEB, như là Internet Explorer.

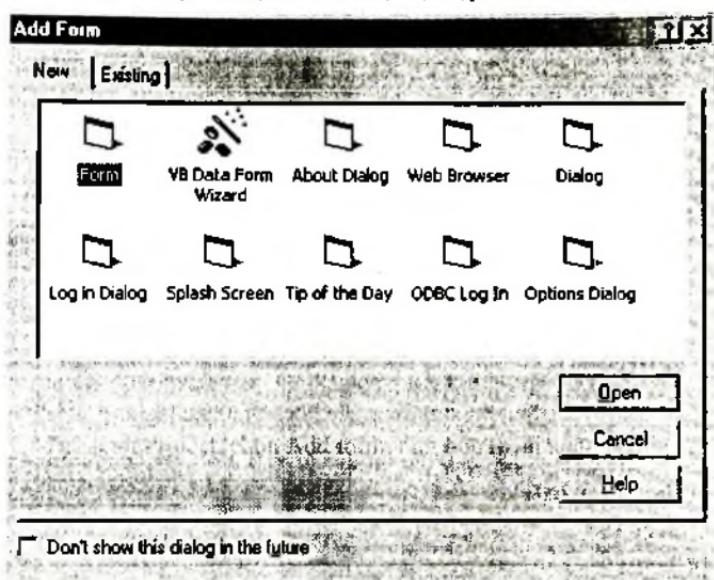


- Module điều khiển (.ctl) và module Property page (.pag) tương tự biểu mẫu, nhưng được dùng để tạo điều khiển ActiveX và danh sách các thuộc tính của chúng để hiển thị khi thiết kế
- Điều khiển ActiveX (.ocx) có thể được thêm vào hộp công cụ để dùng trong biểu mẫu. Khi Visual Basic được cài đặt, một số tệp tin dạng này kèm theo Visual Basic sẽ được chép vào trong máy
- Đối tượng, như là Worksheet của Excel
- Tham chiếu (Reference)
- Trình thiết kế ActiveX: là công cụ dùng để thiết kế các lớp cho đối tượng. Giao diện thiết kế biểu mẫu là một trình thiết kế mặc định

Các Form thông dụng là nút lệnh điều khiển khung được Visual Basic cung cấp sẵn có thể thêm vào dự án bằng hai cách:

Cách 1:

- Từ menu Project, chọn Add form, một hộp thoại xuất hiện



- Nếu muốn tạo mới, chọn tab New. Visual Basic cung cấp các danh sách có sẵn
- Nếu muốn dùng một tệp tin có sẵn, chọn Tab Existing, chọn tên tệp tin, nhấn Open

Cách 2:

- Nhấn nút chuột trong cửa sổ Project Explorer
- Trong menu ngữ cảnh chọn Add
- Xuất hiện hộp thoại như trên

Xoá tệp tin

- Chọn tệp tin trong cửa sổ Project Explorer
- Từ menu Project chọn Remove
- Tham chiếu tệp tin bị xoá trong dự án (thực chất nó vẫn được lưu trên đĩa)

Khi một tệp tin trong dự án bị xoá Visual Basic sẽ cập nhật những thay

đổi này trong tập tin.vbp khi ta lưu dự án. Do đó nếu ta xoá tập tin bên ngoài Visual Basic, tập tin dự án sẽ không được cập nhật. Khi ta mở lại dự án Visual Basic sẽ báo lỗi là thiếu tập tin

Lưu tập tin

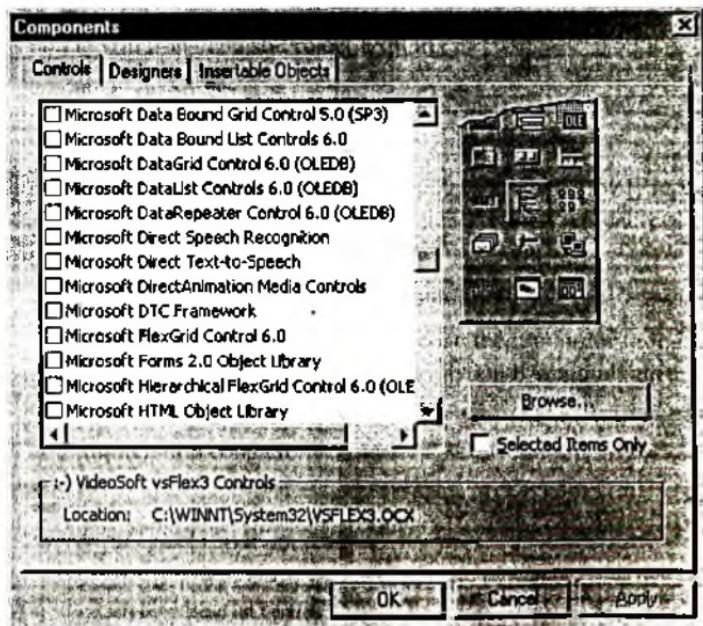
- Chọn tập tin trong cửa sổ Project Explorer
- Từ menu chọn Save

2.1.4.6. Thêm điều khiển vào dự án

Thêm điều khiển ActiveX

Ta có thể thêm vào dự án một điều khiển ActiveX và các đối tượng nhúng được bằng cách thêm nó vào hộp công cụ

1. Từ menu Project chọn components
2. Để thêm một điều khiển (.ocx) hoặc thêm một đối tượng nhúng vào hộp công cụ, chọn vào hộp đánh dấu bên trái tên điều khiển
3. Chọn OK để đóng hộp thoại. Các điều khiển đánh dấu sẽ hiển thị trên hộp công cụ.



Để thêm điều khiển ActiveX vào hộp thoại Components, nhấn nút Browse để tìm đường dẫn cho tệp tin *.OCX. Mỗi điều khiển ActiveX có kèm theo một tệp tin mở rộng là .OCA. Tệp tin này chứa các thông tin của thư viện lưu trữ và các dữ liệu liên quan đến điều khiển. Các tệp tin.OCA chứa trong cùng thư mục với điều khiển ActiveX và được tạo lại khi cần.

Xoá điều khiển khỏi dự án

1. Từ menu project, chọn Components
2. Hộp thoại Components xuất hiện, chọn điều khiển mà ta muốn xoá, xoá hộp đánh dấu kế bên nó

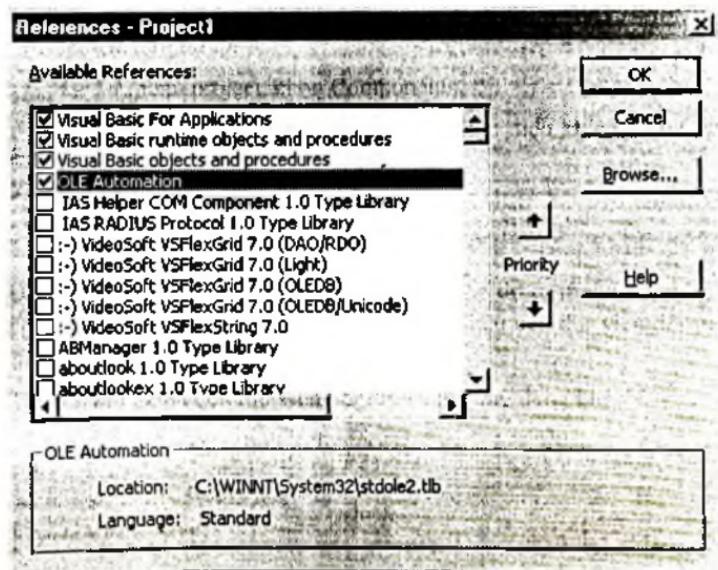
Điều khiển sẽ bị xoá khỏi hộp công cụ

Sử dụng đối tượng của ứng dụng khác

Để lấy đối tượng từ một ứng dụng nào đó, ví dụ ta muốn sử dụng thư viện đối tượng của Microsoft Excel, đặt tham chiếu đến thư viện đối tượng của ứng dụng đó.

Thêm tham chiếu đến thư viện đối tượng của ứng dụng khác

1. Từ menu Project, chọn References



2. Hộp thoại References xuất hiện, chọn vào hộp đánh dấu. Nếu tên tham chiếu chưa có sẵn trong danh sách, nhấn Browse vào ứng dụng và ấn OK.

Nếu không muốn tiếp tục sử dụng đối tượng của thư viện tham chiếu, ta nên xoá đánh dấu tham chiếu để giảm số tham chiếu mà Visual Basic đang quản lý, giảm được thời gian biên dịch dự án.

Khi có tham chiếu đến thư viện đối tượng, ta có thể lấy được đối tượng cùng với các thuộc tính và phương thức của nó bằng cách vào menu view, chọn Object Browser

Thêm tệp tin tài nguyên vào dự án

Tệp tin tài nguyên chứa toàn bộ các hình ảnh, biểu tượng, chuỗi văn bản hiển thị trên màn hình và các thành phần khác liên quan đến việc địa phương hóa ứng dụng.

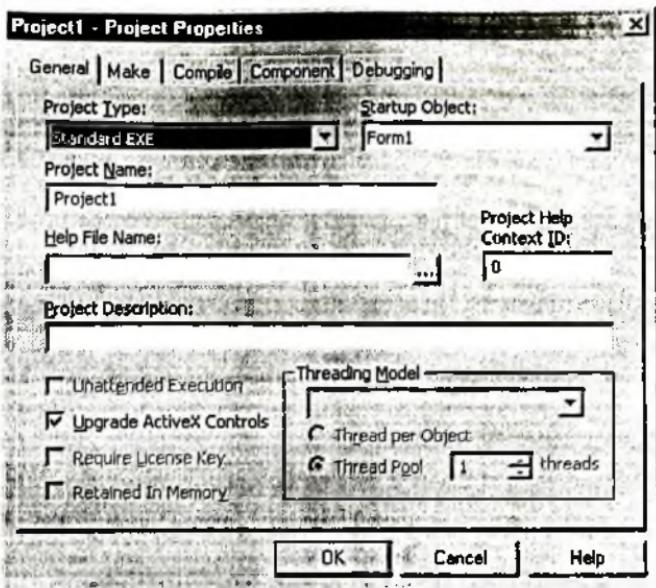
- Từ menu project, chọn Add File
- Chọn tệp tin tài nguyên có sẵn (.RES) và chọn Open

Một dự án đơn giản chỉ có một tệp tin tài nguyên, nếu thêm một tệp tin.RES thứ hai, Visual Basic sẽ báo lỗi.

2.1.4.7. Sửa đổi thuộc tính dự án

Từ menu Project, chọn <tên dự án> Properties. Hộp thoại Project Properties xuất hiện. Những sửa đổi trên hộp thoại này sẽ được lưu trên tệp tin.vbp

Chọn lựa	Giải thích
Startup Object	Tên biểu mẫu được hiển thị hoặc tên thủ tục được thi hành đầu tiên khi chương trình được khởi động
Project Name	Tên dự án, nó không được chứa dấu chấm, khoảng trống và phải bắt đầu bằng chữ cái. Tên dự án không được quá 37 ký tự
Help File	Tên tệp tin hỗ trợ kèm theo dự án
Project Help Context ID	số ID của chủ đề Help được hiển thị khi người sử dụng click vào nút “?”
Project Description	Tên gợi nhớ của dự án. Nó được hiển thị trong References và Object Browser



2.1.4.8. Tao tệp tin EXE

Các ví dụ trên đây được thi hành thông qua nút Start của Visual Basic hoặc nhấn F5. Tuy nhiên khi chương trình hoàn tất, ta cần có một tệp tin thi hành, hay tập tin EXE. Để tạo tệp tin thi hành trước hết đòi hỏi dự án phải không chứa bất kỳ một lỗi về cú pháp nào. Bạn có thể kiểm tra bằng cách vào menu RUN chọn Start With Full Compile. Khi dự án đã được sửa hết lỗi về cú pháp, bạn vào menu File chọn Make Tên dự án. EXE.

2.2. Đối tượng và thuộc tính

2.2.1. Khái niệm

- **Đối tượng (diều khiển - control):** là những thứ mà chúng ta sử dụng để thiết kế và thực hiện một dự án phần mềm. Đối tượng có thể là một biểu mẫu, nút lệnh, thanh công cụ, hộp văn bản...

- **Thuộc tính của đối tượng:** là những đặc điểm của đối tượng sử dụng để mô tả, định nghĩa đối tượng hay để phân biệt đối tượng này với đối tượng khác. Mỗi đối tượng trong Vb thì đều có 1 tập hợp các thuộc tính được quy định trước và thông qua cửa sổ thuộc tính thì chúng ta có thể thiết

lập và thay đổi mọi thuộc tính của đối tượng. Trên màn hình thiết kế Form, ấn F4 để hiện thuộc tính của đối tượng đang được lựa chọn.

2.2.2. Một số thuộc tính thường gặp của đối tượng

a. Name - Tên của đối tượng

- Là thuộc tính để phân biệt đối tượng này với đối tượng khác, nó được sử dụng để truy xuất hay gán giá trị cho các thuộc tính khác của đối tượng khi lập trình. Mỗi một đối tượng khi tạo mới thì đều được gán một tên mặc định và chúng ta thường thay đổi tên mặc định này để đảm bảo hiệu quả quản lý và dễ dàng hơn cho lập trình..

- Quy tắc đặt tên: Tên của đối tượng được đặt theo quy tắc là một chuỗi có tối đa 40 ký tự bắt đầu bằng chữ cái không chứa dấu khoảng cách, không chứa dấu tiếng Việt và thường có 3 ký tự đầu tiên được gọi là tiền tố với chức năng để phân loại đối tượng.

- Chúng ta thường sử dụng các tiền tố như sau:

Loại đối tượng	Tiền tố	Ví dụ đặt tên
ComboBox	cbo	cboDMHang
CheckBox	chk	chkLuaChon
Command button	Cmd	cmdOK
Form	Frm	frmLogin
Grid	Grd	grdDanhMuc
Image	img	imgAnh
Label	Lbl	lblTBao
List	Lst	LstDSach
Menu	Mnu	mnuHeThong
Nút tùy chọn Option button	Opt	optGioiTinh
Hộp chứa hình vẽ PictureBox	Pic	picHinhVe
Đối tượng thời gian Timer	Tmr	tmrThoiGian
Hộp văn bản TextBox	Txt	txtMaHang

b. Thuộc tính tiêu đề - Caption

Có chức năng để xác định tiêu đề cho đối tượng. Đó là một dãy kí tự bắt kí do lập trình viên nhập vào để mô tả chức năng, ý nghĩa của đối tượng. Tiêu đề thường cùng xuất hiện với đối tượng ở trên biểu mẫu. Trên tiêu đề nếu có kí tự & thì khi xuất hiện trên biểu mẫu, chữ cái liền sau kí tự & sẽ được gạch chân để thay vì bấm chuột trực tiếp vào đối tượng ta bấm Alt + kí tự được gạch chân. Ví dụ nếu viết caption của nút xác nhận là "&Thực hiện" ta sẽ nhận được "Thực hiện" và có thể bấm tổ hợp phím Alt+T thay vì bấm chuột trực tiếp vào nút.

c. Thuộc tính khả ứng (Enable)

- Có giá trị mặc định là True. Chức năng để cho phép hoặc không khả năng can thiệp của người sử dụng vào đối tượng khi ứng dụng hoạt động.

d. Thuộc tính Font: Xác định Font chữ, kích thước chữ, kiểu chữ chủ yếu là các đối tượng liên quan đến văn bản như textbox, label....

e. Fore Color: màu chữ

f. Back Color: màu nền (Background)

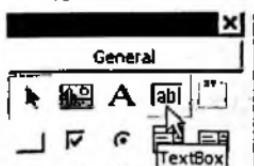
g. Tab Index: quy định thứ tự của việc nhận điều khiển khi ta bấm phím Tab lúc ứng dụng hoạt động.

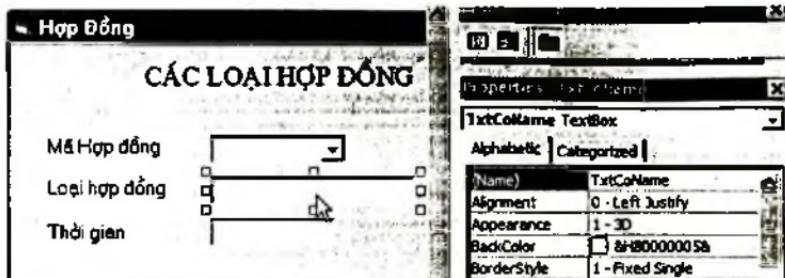
h. Index chứa chỉ số của đối tượng khi chúng ta quản lý các đối tượng theo mảng.

i. Tool Tip Text: là một chuỗi kí tự mô tả ngắn gọn về đối tượng. Nó tự động được hiện ra mỗi khi người dùng di chuyển chuột tới vị trí của đối tượng ở trên form khi ứng dụng hoạt động.

2.2.3. Một số đối tượng thường dùng để thiết kế Form

a. Hộp văn bản - Text Box





Là một điều khiển rất thông dụng dùng để nhận dữ liệu từ người sử dụng cũng như hiển thị dữ liệu trên màn hình. Visual basic và Windows tự động xử lý những hoạt động như hiển thị ký tự khi người sử dụng gõ vào, chèn và xoá ký tự, cuộn dữ liệu, đánh dấu văn bản, cắt dán,...

- Thuộc tính đặc thù:

- + Alignment: xác định hình thức căn lề cho dữ liệu trong hộp.
 - = 0: căn trái
 - = 1: căn phải
 - = 2: căn giữa

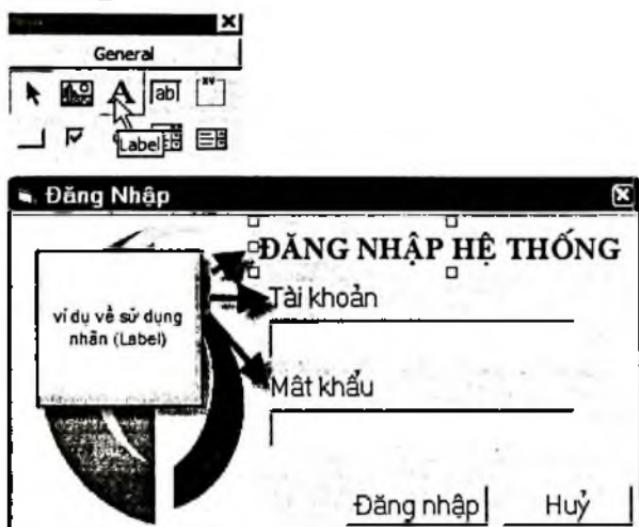
+ Multi Line = True/False: cho phép hoặc không việc hiển thị hoặc soạn thảo nhiều dòng văn bản trong hộp Text box.

+ Password char = Password Character: thường được dùng với những hộp văn bản chứa mật khẩu hay những thông tin bí mật. Thay vì việc hiển thị chính xác kí tự do người dùng nhập vào thì hệ thống chỉ hiển thị kí tự do người dùng định nghĩa trước.



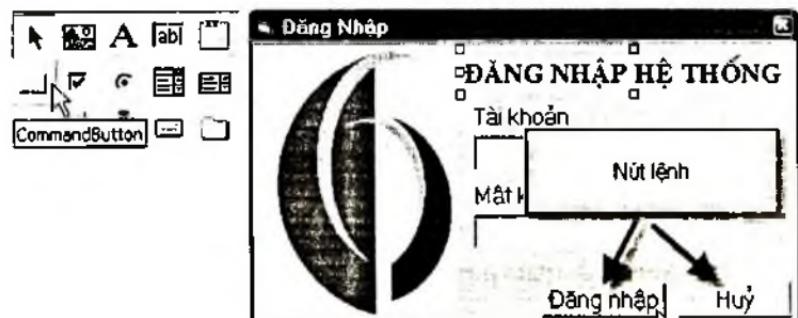
+ Text: để có thể cập nhật hay truy xuất nội dung dữ liệu trong hộp văn bản.

b. Nhãn - Label:



Là một chuỗi ký tự hiển thị trên biểu mẫu có chức năng thường là để giải nghĩa hay mô tả các đối tượng khác. Chúng ta thiết lập hay truy xuất nội dung của nhãn thông qua thuộc tính Caption.

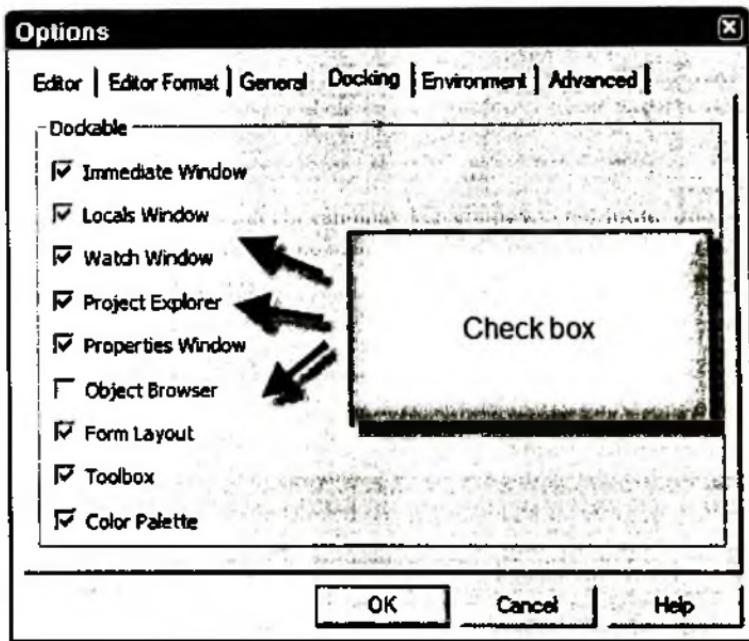
c. Nút lệnh - Command button:



Được hiển thị dưới dạng nút bấm có chức năng để khởi động một chức năng nào đó thông qua sự kiện bấm nút. Từ cửa sổ thiết kế, muôn lập trình

cho sự kiện nút được bấm ta chỉ cần nháy kép chuột vào nút tương ứng làm hiện ra cửa sổ lập trình với thủ tục hiện thời là Sub cmdButton_Click().

b. Check box_Hộp kiểm:



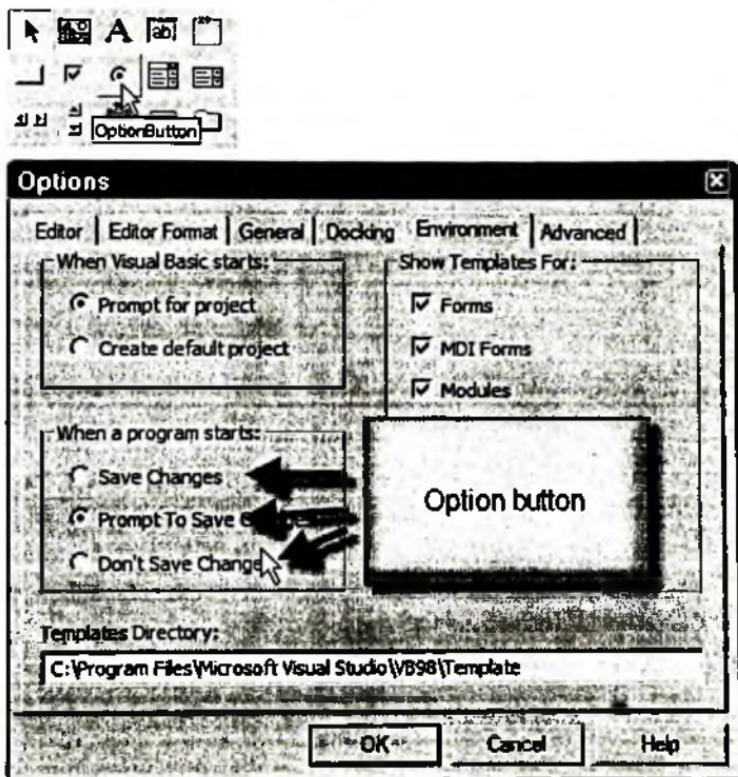
Là đối tượng được sử dụng theo nhóm trên biểu mẫu với chức năng để người dùng lựa chọn một hoặc nhiều phương án trả lời cho câu hỏi được đặt ra. Truy xuất (thiết lập) trạng thái của Check box thông qua thuộc tính Value.

Value = 0 > không được chọn

= 1 > được chọn

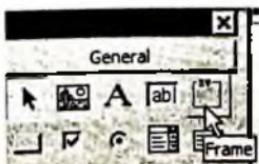
= 2 > không thể lựa chọn

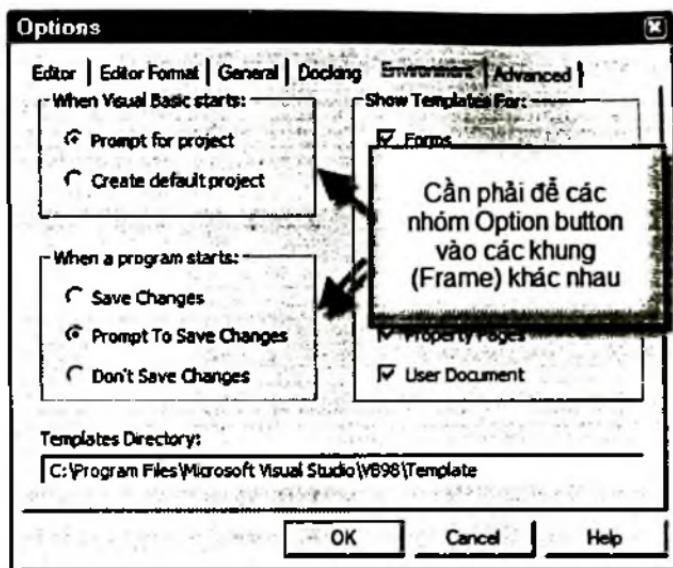
c. Option Button _ Nút lựa chọn:



Số đối tượng ít nhất là 2, nhằm mục đích cho phép người dùng lựa chọn duy nhất một phương án trong số các phương án được đề xuất. Chúng ta muốn tổ chức nhiều sự lựa chọn ta phải để mỗi nhóm trong những khung hay frame khác nhau.

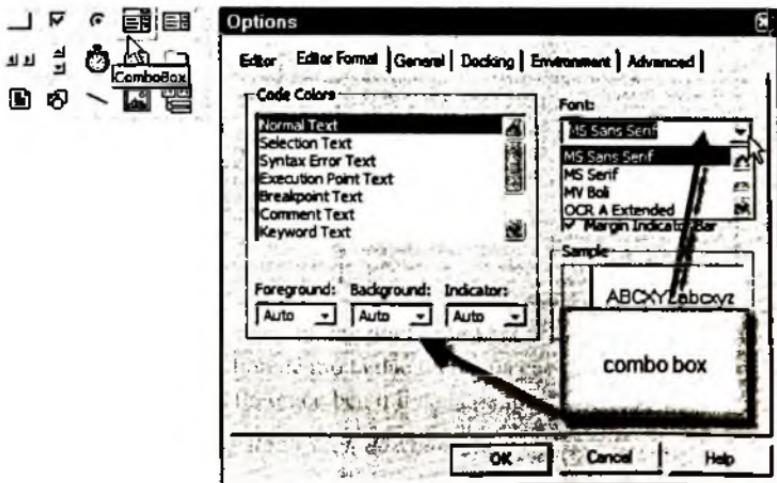
f. Frame _ Khung:





Có chức năng chia nhóm những đối tượng khác trên biểu mẫu. Chúng ta thiết lập thuộc tính Caption để mô tả đặc điểm của các đối tượng chứa trong nhóm. Nó sẽ được hiển thị ở góc bên trái của khung.

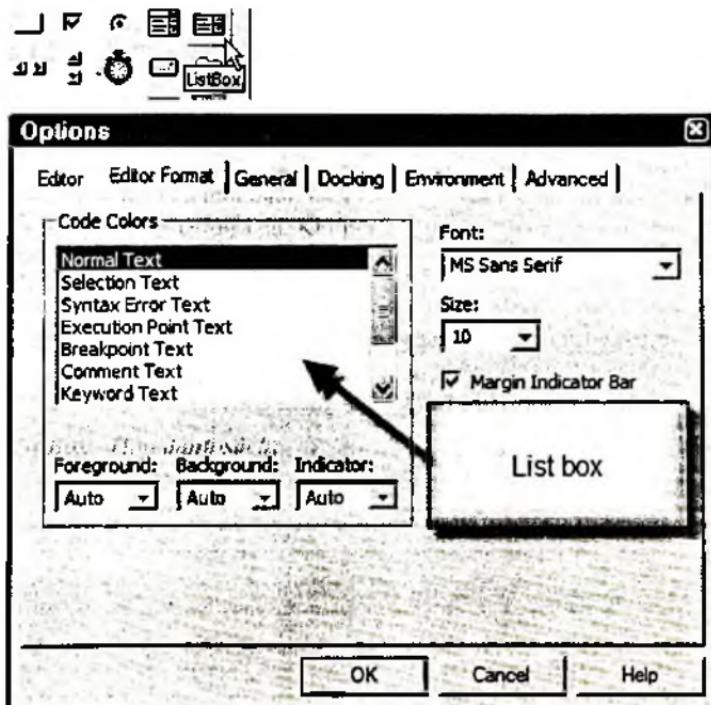
g. ComboBox _ hộp danh sách đồ xuống:



Là đối tượng để chứa một danh sách các giá trị có thể lựa chọn, chủ yếu được sử dụng để cập nhật dữ liệu. Khi hộp ComboBox đang có điều khiển, ta bấm chuột có nút có hình ▼ để hiển thị danh sách (hoặc ấn F4).

Nạp danh sách các lựa chọn vào hộp ComboBox khi thiết kế thông qua các thuộc tính List tại cửa sổ thuộc tính. Khi lập trình, ta có thể nạp danh sách các lựa chọn vào hộp combo Box thông qua thuộc tính AddItem. Truy xuất đến giá trị đã được lựa chọn thông qua thuộc tính Text.

h. List Box_ Hộp danh sách:



Là đối tượng sử dụng để hiển thị và cho phép lựa chọn một đối tượng trong danh sách. Nạp dữ liệu vào ListBox cũng thông qua List và thủ tục Add Item tương tự như với hộp ComboBox.

2.3. Các cấu trúc điều khiển trong VB

2.3.1. Biến, hàng và các kiểu dữ liệu trong VisualBasic

a. Khai báo biến

Biến là một đối tượng trong chương trình có thể nhận các giá trị khác nhau phục vụ cho mục đích của chương trình.

Để khai báo biến ta dùng lệnh Dim:

Dim <Tên biến> [As<kiểu dữ liệu>]

Biến khai báo trong thủ tục chỉ tồn tại khi thủ tục thi hành. Nó sẽ biến mất khi thủ tục chấm dứt. Giá trị của biến trong thủ tục là cục bộ đối với thủ tục đó, nghĩa là ta không thể truy nhập biến từ bên ngoài thủ tục. Nhờ đó, ta có thể dùng trùng tên biến cục bộ trong những thủ tục khác nhau.

Kiểu dữ liệu trong khai báo Dim có thể là những kiểu cơ bản như Integer, String hoặc Currency. Ta cũng có thể dùng đối tượng của VB (như Object, Form1, TextBox) hoặc của các ứng dụng khác.

Khai báo biến trong phần Declarations của một môđun nghĩa là biến đó tồn tại và có tầm hoạt động trong môđun đó.

Khai báo biến với từ khoá Public nghĩa là biến đó tồn tại và có tầm hoạt động của toàn ứng dụng.

Khai báo biến cục bộ với từ khoá Static nghĩa là mặc dù biến đó biến mất khi thủ tục chấm dứt, nhưng giá trị của nó vẫn được giữ lại để tiếp tục hoạt động khi thủ tục được gọi trong lần sau.

Khai báo ngầm

Nghĩa là ta không cần khai báo tường minh trước khi sử dụng biến.

```
Function SafeSqr(num)
    TempVal = Abs(num)
    SafeSqr = Sqr(TempVal)
End Function
```

Mặc dù cách này có vẻ thuận tiện nhưng có thể gây lỗi nếu ta gõ nhầm tên biến.

```
Function SafeSqr(num)
    TempVal = Abs(num)
```

```
SafeSqr = Sqr(TemVal)
```

```
End Function
```

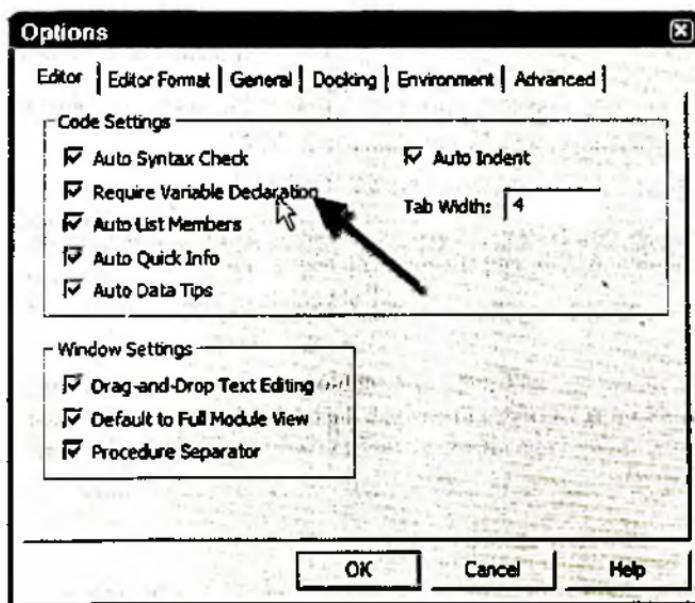
Hàm này trả về zero. Khi VB gấp tên mới, nó tạo ra một biến khác với tên đó.

Khai báo tường minh

Để tránh những rắc rối trên, ta nên quy định VB phải báo lỗi khi gấp một tên biến không khai báo. Ta đặt dòng lệnh:

Option Explicit

Trong phần Declarations của mô-đun. Một cách khác, từ menu Tools, chọn Options, chọn tab Editor và đánh dấu vào tùy chọn **Require Variable Declaration**.



VB tự động chèn dòng lệnh Option Explicit vào một mô-đun mới, nhưng không phải là những mô-đun đã được tạo. Do đó, đối với các mô-đun này, ta phải thêm dòng lệnh bằng tay.

Option Explicit chỉ hoạt động trên từng mô-đun. Vì vậy, ta phải thêm dòng này vào mỗi mô-đun của biểu mẫu, mô-đun chuẩn, hay mô-đun lớp.

Tâm hoạt động của biến

Tâm hoạt động	Private	Public
Thủ tục	Biến chỉ tồn tại và hoạt động trong thủ tục	Không có
Mô-đun	Biến chỉ tồn tại và hoạt động trong mô-đun	Biến tồn tại và hoạt động trên mọi mô-đun

Khai báo biến Static

Để khai báo tất cả các biến cục bộ trong một thủ tục là Static, ta đặt từ khoá Static vào tên thủ tục:

Static Function RunningTotal(num)

VB sẽ hiểu rằng tất cả các biến khai báo trong thủ tục này đều là Static, dù cho chúng được khai báo là Private, là Dim hoặc thậm chí khai báo ngầm.

Từ khoá Static có thể đặt ở đầu thủ tục Sub hoặc Function, kể cả thủ tục xử lý sự kiện hoặc những hàm Private.

b. Hàng

Dùng để chứa những dữ liệu tạm thời nhưng không thay đổi trong suốt thời gian chương trình hoạt động. Sử dụng hằng số làm chương trình sáng sủa và dễ đọc nhờ những tên gọi nhớ thay vì các con số. VB cung cấp một số hằng định nghĩa sẵn, nhưng ta có thể tự tạo hàng.

Ta có thể dùng cửa sổ Object Browser để xem danh sách các ứng dụng hàng có sẵn của VB và VBA(Visual basic for Application). Các ứng dụng khác cung cấp những thư viện đối tượng, như Microsoft Exel, Microsoft Project, hoặc các thư viện của điều khiển ActiveX cũng có hàng định nghĩa sẵn.

Trong trường hợp trùng tên hàng trong những thư viện khác nhau, ta có thể dùng cách chỉ rõ tham chiếu hàng:

[<Libname>][<tên mô-đun>]<tên hàng>

Libname là tên lớp, tên điều khiển hoặc tên thư viện.

Khai báo hàng

|Public|private|Const<tên hàng>[As<kiểu dữ liệu>]= <biểu thức>

Tâm hoạt động

Hàng cũng có tầm hoạt động tương tự biến:

- Hàng khai báo trong thủ tục chỉ hoạt động trong thủ tục
- Hàng khai báo trong mô-đun chỉ hoạt động trong mô-đun
- Hàng khai báo Public trong phần Declarations của mô-đun chuẩn có tầm hoạt động trên toàn ứng dụng. Khai báo Public không thể dùng trong mô-đun của biến mẫu hoặc mô-đun lớp.

c. Các kiểu dữ liệu sử dụng trong VB

- **Kiểu dữ liệu:** là sự quy định về cấu trúc, miền giá trị mà một biến thuộc kiểu đó có thể nhận được và tập hợp các phép toán xác định trên miền giá trị đó.

Kiểu số

- Decimal _ số thực 16 byte
- Double _ số thực 8 byte
- Integer _ số nguyên từ -32768 đến 32767
- Long _ từ -2 tỷ - đến 2 tỷ
- Single _ số nguyên 1 byte

Integer, Long, Double và Currency. Kiểu số tồn ít vùng chứa hơn kiểu Variant. Tất cả biến kiểu số có thể được gán cho nhau và cho biến Variant. VB làm tròn thay vì chặt bỏ phần thập phân trước khi gán nó cho số Integer.

Kiểu Integer tồn ít vùng nhớ hơn các kiểu khác, nó thường dùng làm biến đếm trong các vòng lặp For....Next.

Kiểu Single, Double, Currency dùng cho các số có phần thập phân. Currency hỗ trợ đến 4 chữ số phần thập phân và 15 chữ số cho phần nguyên, dùng cho các tính toán tiền tệ.

Các giá trị dấu chấm động được thể hiện là: $A \times 10^B$. Ví dụ:

$$1.2341E12 = 1.2341 \times 10^{12}$$

3.402823E+38 cho số Single hoặc 1.7976931486232D+308 cho số Double

Ta dùng các phép cộng (+), trừ(-) nhân(*), chia(/ hoặc \). Dấu / là số chia thập phân. Ví dụ: 5/3 cho kết quả là 1.666666666667. Trong khi 5\3 cho kết quả là 1, phần thập phân bị chặt bỏ. Phép tính này đặc biệt nhanh khi sử dụng trong vòng lặp.

Kiểu Byte

Nhận các giá trị trong khoảng 0 đến 255. Thường dùng để chứa dữ liệu nhị phân. Tất cả các thao tác trên kiểu Integer có thể thực hiện trên kiểu Byte, ngoại trừ dấu. Vì Byte là kiểu không dấu (trong khoảng từ 0-255), nó không thể nhận ra số âm.

Kiểu String

Mặc định, biến hay tham số kiểu chuỗi có chiều dài thay đổi, nó có thể tăng hoặc giảm tùy theo ta gán dữ liệu. Ta có thể khai báo chuỗi có chiều dài cố định:

```
Dim EmpName As String *50
```

Nếu ta gán một chuỗi ngắn hơn 50 ký tự, EmpName sẽ được thêm vào phần đuôi các ký tự khoảng trắng cho đầy 50 ký tự, nếu chuỗi gán vào dài hơn 50 ký tự, VB tự động chặt bỏ.

Khi làm việc với chuỗi, ta cần dùng các hàm Trim và RTrim để cắt bỏ các ký tự trắng không cần thiết. Ngoài ra, một số hàm thông dụng để thao tác trên chuỗi như:

- Len: Lấy chiều dài chuỗi
- Mid\$: Trích chuỗi con từ chuỗi gốc
- Left\$: Trích chuỗi con từ phần đầu chuỗi gốc.
- Right\$: Trích chuỗi con từ phần đuôi của chuỗi gốc.
- InStr: Tìm chuỗi con trong chuỗi gốc. Nếu hàm InStr trả về 0, nghĩa là không tìm thấy.

Tìm kiếm không phân biệt cỡ chữ. Nhưng nếu tham số thứ 3 là vbBinaryCompare thì đây là tìm kiếm chuỗi có phân biệt chữ in hoa và chữ in thường.

- Replace: Tìm và thay thế chuỗi

VB cho phép chuyển đổi một chuỗi thành một số nếu chuỗi đang thể hiện một con số. Ngược lại, ta cũng có thể chuyển một số thành chuỗi. Tuy nhiên nên cẩn thận, vì chuyển đổi một chuỗi có giá trị không phải số sẽ gây lỗi chương trình thi hành.

Một số lập trình viên Visual Basic thích dùng dấu + để nối chuỗi thay vì dùng dấu &. Mặc dù không khác nhau lắm, nhưng thực ra dùng dấu + có điểm bất tiện. Vì là phép toán, nó có kiểm tra kiểu. Nếu ta có một số và một chuỗi nối với nhau, nó sẽ chuyển đổi từ số sang chuỗi trước khi thực sự kết

nội. Hơn nữa, việc chuyển đổi này được làm tự động, không hề báo lỗi khi biến đổi.

Kiểu Boolean

Nếu ta có một biến có hai giá trị True/False, Yes/No, On/Off, ta nên dùng kiểu Boolean. Giá trị mặc định của Boolean là False.

Kiểu Date

Khi các kiểu dữ liệu khác được chuyển sang Date, giá trị đúng trước dấu chấm là ngày, giá trị đúng sau dấu chấm là giờ. Nửa đêm là 0, giữa ngày là 0,5. Dấu âm thể hiện ngày trước 30/12/1999. Kiểu Date đã giải quyết vấn đề Y2K.

Nhấn Ctrl-G để hiển thị cửa sổ Immediate.

Gõ vào:

“01/02/98” và nhấn Enter.

Nếu hiểu theo người Mỹ, “01/02/98” có nghĩa là ngày 2 tháng Giêng năm 1998, nếu hiểu theo người Anh thì đây là ngày 1 tháng 2 năm 1998. Nếu dùng ngày như trong hình trên thì VB hiểu rằng lấy 1 chia cho 2 rồi lấy kết quả chia cho 98!

Trở lại cửa sổ Immediate gõ vào: #01/02/98#

Dấu # cho biết là dữ liệu kiểu Date, không phải một biểu thức toán học. Tuy nhiên, định dạng ngày tháng hiển thị phụ thuộc vào quy định của Windows.

Hộp thoại này hiển thị khi người sử dụng nhập cúp chuột vào biểu tượng Regional Setting trong cửa sổ Control Panel của Windows. Nó cho phép quy định kiểu ngày tháng tùy thuộc quốc gia. Bên trong chương trình VB xử lý ngày tháng theo kiểu Mỹ #01/02/98# là ngày 2 tháng Giêng năm 1998, nhưng nếu máy đang dùng theo hệ Anh thì nó sẽ hiển thị trên cửa sổ Immediate là 2/1/98.

Kiểu Object

Biến kiểu Object chứa một địa chỉ 4 byte (32bit) trả đến đối tượng trong ứng dụng hiện hành hoặc các ứng dụng khác. Dùng lệnh Set để chỉ ra đối tượng thực sự:

Dim objDb As Object

Set objDb=OpenDatabase("c:\vb5\Biblio.mdb")

Khi khai báo biến đối tượng, nên chỉ ra tên lớp tường minh, như

TextBox thay vì Control, Database thay vì Object. Ứng dụng sẽ chạy nhanh hơn, ta có thể xem danh sách các lớp trong cửa sổ Object Browser.

Kiểu Variant

Có thể chứa mọi loại dữ liệu, số, thậm chí mảng. Ta không cần chuyên đổi kiểu dữ liệu, VB làm việc đó một cách tự động.

Dim Somevalue 'Variant by default

Somevalue = "17" 'SomeValue contains "17"(a two character string).

Somevalue = Somevalue - 15 'somevalue now contains the numeric value 2.

Variant cũng thuận tiện khi ta không biết trước kiểu dữ liệu

```
Private Sub cmdExplore_Click()
```

```
Dim VarVariant As Variant
```

```
VarVariant = 12
```

```
Form1.Print VarType(VarVariant)
```

```
VarVariant = "Peter"
```

```
Form1.Print VarType(VarVariant)
```

```
VarVariant = True
```

```
Form1.Print VarType(VarVariant)
```

```
VarVariant = #1/1/2001#
```

```
Form1.Print VarType(VarVariant)
```

```
End Sub
```

Hàm VarType kiểm tra kiểu dữ liệu

Giá trị VarType	Giải thích
0 - vbEmpty	Không chứa gì cả
1 - vbNull	Không có dữ liệu hợp lệ
2 - vbInteger	Dữ liệu Integer dạng chuẩn
3 - vbLong	Dữ liệu kiểu Long Integer
4 - vbSingle	Dữ liệu kiểu chấm động single
5 - vbDouble	Dữ liệu kiểu chấm động Double

6 - vbCurrency	Kiểu Currency
7 - vbDate	Kiểu ngày giờ
8 - vbString	Kiểu chuỗi đơn giản
9 - vbObject	Kiểu đối tượng
10 - vbError	Có một đối tượng Error
11 - vbBoolean	Kiểu giá trị Boolean chuẩn
12 - vbVariant	Kiểu Variant
13 - vbDataObject	Kiểu DAO chuẩn
14 - vbDecimal	Giá trị thuộc hệ thập phân Decimal
17 - vbByte	Kiểu Byte
36 - UserDefinedType	Kiểu do người dùng định nghĩa
8192 - vbArray	Kiểu mảng

Tuy nhiên cần chú ý khi dùng biến Variant:

- Nếu muốn thi hành các hàm số học, Variant phải chứa giá trị số.
- Nếu muốn nối chuỗi, dùng toán tử & thay vì toán tử +.

Giá trị Empty

Đôi khi ta cần kiểm tra một giá trị có được gán cho biến hay chưa. Biến Variant có giá trị Empty trước khi nó được gán giá trị. Giá trị Empty là một giá trị đặc biệt không phải zero, không phải chuỗi rỗng (""), không phải giá trị Null. Ta dùng hàm IsEmpty để kiểm tra giá trị Empty:

If IsEmpty(z) then z =0

Khi một biến Variant chứa giá trị Empty, ta có thể dùng nó trong biểu thức. Nó có thể được xem là 0 hoặc chuỗi rỗng tùy theo biểu thức.

Giá trị Empty biến mất khi có một giá trị bất kỳ được gán cho Variant. Muốn trở về giá trị Empty, ta gán từ khoá Empty cho Variant.

Giá trị Null

Biến Variant chứa giá trị Null dùng trong những ứng dụng cơ sở dữ liệu thì hiện không có dữ liệu hoặc dữ liệu không xác định.

Dùng hàm IsNull để kiểm tra biến Variant có chứa Null hay không. Biến

không bao giờ mang giá trị Null nếu ta không gán trực tiếp cho nó. Vì vậy, không cần phải dùng hàm IsNull.

Nếu gán Null cho một biến khác kiểu Variant, VB sẽ báo lỗi.

Giá trị Error

Trong một biến Variant, Error là một giá trị đặc biệt thể hiện một điều kiện lỗi vừa xảy ra trong thủ tục. Tuy nhiên, không như các lỗi khác, các xử lý lỗi thông thường của ứng dụng không xảy ra. Do đó, ta có thể xử lý dựa trên các giá trị lỗi. Giá trị Error được sinh ra bằng cách chuyển đổi giá trị lỗi dùng cho hàm CVErr.

d. Dữ liệu kiểu mảng(Array)

Mảng là một xâu các biến có cùng tên và cùng kiểu dữ liệu. Dùng Array làm chương trình đơn giản và rút gọn, vì ta có thể dùng vòng lặp. Mảng có biến trên và biến dưới, và các thành phần trong mảng là liên tục giữa 2 biến.

Khái niệm mảng ở đây khác với mảng các điều khiển (Control Array). Control Array không cho phép nạp hay thoát khỏi một thành phần ở giữa Array.

Có 2 loại biến mảng: mảng có chiều dài cố định và mảng động, có chiều dài thay đổi lúc thi hành. Mảng có chiều dài cố định có thể được khai báo Public trong ứng dụng. Private trong mô-đun hoặc Private trong một thủ tục.

Mảng có chiều dài cố định

Biên trên và biên dưới

Biên trên được xác định ngay lúc khai báo .

Dim counters(14) As Integer

Public sums(20) As Double

Mặc định, biên dưới là 0. Ta có thể khai báo tường minh biên dưới:

Dim counter(1 To 15) As Integer

Dim sums(100 To 120) As String

Mảng nhiều chiều

Ta khai báo một mảng 2 chiều có 10 phần tử

Static MatrixA(9, 9) As Double

Static MatrixA(1 To 10, 1 To 10) As Double

Dim MultiD(3, 1 To 10, 1 To 15)

Khai báo này tạo ra một mảng 3 chiều có kích cỡ $4 \times 10 \times 15$, là số phần tử của ma trận, 600

Nên thận trọng trong khi sử dụng các mảng nhiều chiều, nhất là các mảng các Variant vì nó lớn hơn các kiểu dữ liệu khác.

Mảng động (dynamic Array)

Mảng này có thể thay đổi kích cỡ. Là một trong những ưu điểm của Visual Basic, mảng động giúp quản lý bộ nhớ một cách hiệu quả. Ta có thể dùng một mảng lớn trong thời gian ngắn, sau đó xoá bỏ để trả vùng nhớ cho hệ thống.

Khai báo

- Khai báo Public hoặc Dim trong mô-đun, hoặc khai báo Static hay Dim trong thủ tục. Khai báo một mảng động bằng cách cho nó một danh sách không theo chiều nào cả.

Dim DynArray()

- Cấp phát số phần tử thực sự bằng dòng lệnh ReDim.

ReDim DynArray(x+1)

Sử dụng ReDim

Dòng lệnh ReDim chỉ có thể xuất hiện trong thủ tục. Khác với Dim hay Static, ReDim là một dòng lệnh thi hành, nó làm ứng dụng phải thực hiện một hành động lúc chạy chương trình. Sử dụng ReDim tương tự trong mảng có chiều dài cố định, dùng thay đổi số phần tử cũng như biên trên hoặc biên dưới. Tuy nhiên, số chiều không thay đổi.

ReDim DynArray(4 to 12)

Dim Matrix1() as integer

Sub CalcValuesNow()

-

ReDim Matrix1(19,29)

End sub

Mỗi lần gọi ReDim, tất cả các giá trị chưa trong mảng hiện hành bị mất. Vb

khởi tạo lại giá trị cho chúng (Empty đối với mảng Variant, 0 cho mảng số, chuỗi rỗng cho mảng chuỗi, hoặc nothing cho mảng các đối tượng). Cách này tiện lợi khi ta muốn thêm dữ liệu mới hoặc muốn xoá bớt vùng nhớ. Đôi khi, ta muốn thay đổi kích cỡ của mảng mà không mất dữ liệu. Ta dùng ReDim với từ khoá **Preserve**. Ví dụ, mở rộng mảng thêm một phần tử và không mất dữ liệu:

ReDim Preserve DynArray (UBound(DynArray)+1)

Hàm **UBound** trả về phần tử cuối của mảng(Upper Bound).

Hàm **LBound** trả về phần tử đầu tiên của mảng (Lower Bound).

Tuy nhiên chỉ có biên trên của chiều cuối cùng trong mảng được thay đổi khi ta dùng **Preserve**. Nếu thay đổi chiều khác, hoặc biên dưới của chiều cuối cùng VB sẽ báo lỗi.

2.3.2. Cấu trúc chọn

So sánh mặc định trong Visual basic mặc định là so sánh có phần biệt cỡ chữ. Nếu muốn tắt chế độ này, ta thêm dòng khai báo sau vào chương trình

Option Compare Text

Nếu muốn trả về trạng thái ban đầu, có 2 cách:

Đưa dòng khai báo:

Option Compare Binary

Chi cần xoá dòng khai báo “Option Compare Text”

Các biểu thức so sánh

Ký hiệu	Ý nghĩa
=	Bằng
<>	Khác
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng

2.3.2.1. If.. Then

QC1. Một dòng lệnh

If <điều kiện> Then <dòng lệnh>

QC2. Nhiều dòng lệnh

If <điều kiện> Then

<nhóm lệnh>

End if

Nếu điều kiện thỏa mãn thì VB sẽ thực hiện dòng lệnh trong QC1, hay là thực hiện một nhóm các lệnh trong QC2. Nếu điều kiện không thỏa mãn thì không thực hiện lệnh nào cả.

2.3.2.2. If.. Then...Else

If <điều kiện 1> Then

[Khối lệnh - 1]

ElseIf <điều kiện 2> Then

[Khối lệnh - 2]

.....

Else

[Khối lệnh - n]

End if

Cấu trúc nào cho phép thực hiện kiểm tra nhiều điều kiện. Nếu điều kiện 1 thỏa mãn thì đi thực hiện khối lệnh 1. Nếu không có điều kiện nào thỏa mãn thì thực hiện khối lệnh n紧跟 sau từ khóa Else cuối cùng.

2.3.2.3. Select Case

Giải quyết trường hợp có quá nhiều Elseif được dùng, giúp chương trình sáng sủa dễ đọc. Biểu thức để so sánh được tính toán một lần vào đầu cấu trúc. Sau đó Visual basic so sánh kết quả biểu thức với từng Case. Nếu bằng nó thi hành khối lệnh trong Case đó.

Select Case <bíểu thức kiểm tra>

Case <danh sách biểu thức 1>

 Khối lệnh 1

Case <danh sách biểu thức 2>

Khối lệnh 2

.....

Case else

Khối lệnh n

End Select

Mỗi danh sách biểu thức chứa một hoặc nhiều giá trị, các giá trị cách nhau bằng dấu phẩy. Mỗi khối lệnh có thể chứa từ 0 đến nhiều dòng lệnh. Nếu có hơn một Case thỏa mãn điều kiện thì Case đầu tiên được thực hiện. Case else không nhất thiết phải có, dùng trong trường hợp còn lại của các Case trước. Nếu danh sách các giá trị biến thiên liên tiếp, ta có thể biểu diễn bằng cách viết <giá trị đầu> to <giá trị cuối>

VD: Case 1 to 10

2.3.3. Cấu trúc lặp

2.3.3.1. Do..loop

Thi hành một khối lệnh với số lần lặp không định trước, trong đó, một biểu thức điều kiện dùng so sánh để quyết định vòng lặp có tiếp tục hay không. Điều kiện phải quy về False hoặc True.

Kiểu 1:

Do While <điều kiện>

<khối lệnh>

[Exit Do]

Loop

Kiểu 2: Vòng lặp luôn có ít nhất một lần thi hành khối lệnh

Do

<Khối lệnh>

[Exit Do]

Loop While <điều kiện>

Kiểu 3: Lặp trong khi điều kiện là False

Do until <điều kiện>

<khối lệnh>

[Exit Do]

Loop

Kiểu 4: Lặp trong khi điều kiện là False và có ít nhất một lần thi hành khối lệnh

Do

<khối lệnh>

[Exit Do]

Loop Until <điều kiện>

Khi gặp lệnh Exit Do lập tức vòng lặp sẽ kết thúc vô điều kiện.

2.3.3.2. For...Next

Biết trước số lần lặp. Ta dùng biến đếm tăng dần hoặc giảm dần trong vòng lặp.

For <bien dem> = <điểm đầu> To <điểm cuối> [Step <bước nhảy>]

<khối lệnh>

[Exit For]

Next [<Biến đếm>]

Biến đếm, điểm đầu, điểm cuối và bước nhảy là những giá trị số nguyên.

Bước nhảy có thể là âm hoặc dương. Nếu bước nhảy là dương, điểm đầu phải nhỏ hơn hoặc bằng điểm cuối, nếu bước nhảy là âm thì ngược lại.

Khi gặp Exit for vòng lặp kết thúc vô điều kiện.

2.3.3.3. For Each...Next

Tương tự vòng lặp For... Next, nhưng nó lặp khối lệnh theo số phần tử của một tập các đối tượng hay một mảng thay vì theo số lần lặp xác định. Vòng lặp này tiện lợi khi ta không biết chính xác bao nhiêu phần tử trong tập hợp.

For Each <phần tử> In <Nhóm>

<khối lệnh>

[Exit For]

Next <phần tử>

2.3.3.4. Vòng lặp While...Wend

Tương tự vòng lặp Do...While, nhưng ta không thể thoát vòng lặp bằng lệnh Exit. Vì vậy, vòng lặp kiểu này chỉ thoát khi biểu thức điều kiện sai.

```
While <điều kiện>
<khối lệnh>
Wend
```

2.4. Các thao tác cơ bản khi lập trình và sửa lỗi

Có lẽ khâu quan trọng nhất trong lập trình là thiết kế. Sau khi thiết kế giao diện, bạn cần thiết kế cấu trúc chương trình. Cách thiết kế khác nhau sẽ dẫn đến cách hoạt động khác nhau và bảo trì, theo đó cũng khác nhau.

Code trong VB được tổ chức theo dạng cây phân nhánh. Một ứng dụng thông thường chứa một hoặc nhiều mô-đun. Mỗi biểu mẫu có một mô-đun, có thể thêm những mô-đun chuẩn chứa những đoạn chương trình dùng chung, và cũng có thể có thêm mô-đun lớp.

2.4.1. Các thao tác thông dụng trong cửa sổ Code

a. Soạn thảo Code

Ngoài khả năng soạn thảo văn bản để viết chương trình, cửa sổ Code còn hỗ trợ một số chức năng khác như:

Đánh dấu (bookmarks)

Dùng đánh dấu các dòng chương trình trong cửa sổ Code để dễ dàng xem lại về sau. Để bật tắt khả năng này, cũng như tìm kiếm dấu hiện hành, chọn Bookmarks từ menu Edit, hoặc chọn từ thanh công cụ Edit.

Dùng phím trong cửa sổ Code

Chức năng	Phím tắt
Xem cửa sổ Code	F7
Xem cửa sổ Object Browser	F2
Tìm kiếm	CTRL + F
Thay thế	CTRL + H
Tìm tiếp	SHIFT + F4
Tìm ngược	SHIFT + F3
Chuyển đến thủ tục kế tiếp	CTRL + DOWN ARROW
Chuyển đến thủ tục trước đó	CTRL + UP ARROW

Xem định nghĩa	SHIFT + F2
Cuộn xuống 1 màn hình	CTRL + PAGE DOWN
Cuộn lên một màn hình	CTRL + PAGE UP
Nhảy về vị trí trước đó	CTRL + SHIFT + F2
Trở về đầu của mô-đun	CTRL + HOME
Đến cuối mô-đun	CTRL + END
Chức năng	Phím tắt
Dời con trỏ sang phải 1 từ	CTRL + RIGHT ARROW
Dời con trỏ sang trái 1 từ	CTRL + LEFT ARROW
Dời con trỏ về cuối dòng	END
Dời con trỏ về đầu dòng	HOME
Lấy lại hành động trước đó	CTRL + Z
Xoá dòng hiện hành	CTRL + Y
Xoá 1 từ	CTRL + DELETE
Canh trái	TAB
Bỏ hành động canh trái trước đó	SHIFT + TAB
Xoá tất cả các điểm dừng (break-points)	SHIFT + SHIFT + F9
Xem menu cảm ngữ cảnh	SHIFT + F10

b. Chức năng tự động kiểm tra cú pháp (Auto Syntax Check)

Từ menu Tools, chọn Option... Hộp thoại xuất hiện

Khi Auto Syntax Check không bật lên, nếu ta viết 1 dòng chương trình như sau:

Form1.left =

rồi nhấn phím Enter. VB sẽ hiển thị dòng chương trình sai với màu đỏ. Tuy nhiên, nó không giải thích thêm và ta có thể tiếp tục gõ chương trình. Nếu Auto Syntax Check được bật lên, khi ta vừa nhấn phím Enter, VB lập tức cho ta biết một số thông tin về lỗi và hiển thị con trỏ ngay dòng

chương trình sai để chờ ta sửa. Trong trường hợp này, VB cần một giá trị bên phải dấu bằng.

2.4.2. Gỡ rối chương trình

Không một chương trình nào là không có lỗi. Tuy nhiên, giảm khả năng lỗi đến mức tối thiểu là có thể làm được. Để có chương trình tốt, ta cần có thiết kế chặt chẽ. Sau đó, chương trình phải được viết sao cho có ít lỗi và nếu có thì dễ tìm.

2.4.2.1. Một số giải pháp giám lỗi

- Thiết kế cẩn thận, ghi ra các vấn đề quan trọng và cách giải quyết cho từng phần. Ghi ra từng thủ tục và mục đích của nó.
- Chủ thích rõ ràng trong chương trình
- Đối tượng có tham chiếu tường minh thay vì kiểu chung chung như Object, Control
- Tuân thủ Coding convention
- Một trong những nguyên nhân gây lỗi là gõ sai tên biến hoặc nhầm lẫn điều khiển. Dùng Option Explicit để tránh trường hợp này.

Đối tượng Err

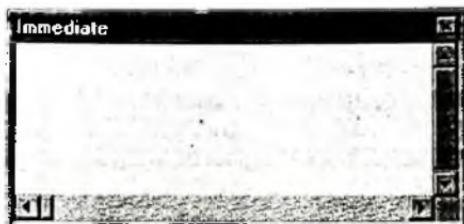
Là đối tượng do Visual basic cung cấp sẵn. Nó có vô số thuộc tính. Sau đây là những thuộc tính thông dụng:

Thuộc tính	Giải thích
Number	Giá trị mặc định, số hiệu lỗi
Description	Mô tả lỗi
Source	Tên đối tượng gây ra lỗi

Gỡ rối

Có thể tạm dừng chương trình bằng cách chọn Break từ menu Run hoặc nhấn trên thanh công cụ, hoặc nhấn trên tổ hợp phím Ctrl-Break. Ta cũng có thể đặt dòng lệnh Stop trong chương trình nhưng sẽ có cách khác tốt hơn.

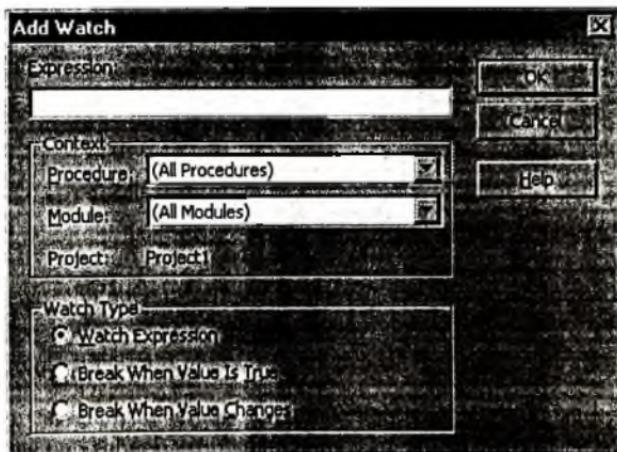
Cửa sổ Immediate



Cửa sổ này cho phép ta xem các giá trị của các biến trong form khi ta gõ rối.

Cửa sổ Watch

Hiển thị các giá trị của một biến, thuộc tính hay biểu thức bất kỳ. thậm chí có thể buộc chương trình tạm ngưng sau một số lần lặp.



Đi qua từng dòng chương trình

Thanh Debug



Thứ tự nút bấm từ trái sang phải như sau:

- Start: thi hành chương trình

- Break: tạm dừng chương trình
- End: Kết thúc chương trình
- BreakPoint: Điểm đánh dấu dòng lệnh để tạm dừng chương trình. Nút này được sử dụng để bật tắt chế độ breakpoint. Khi có lỗi xảy ra và ta chưa khoanh được khu vực nghi ngờ, thì Breakpoint là giải pháp tốt nhất để cô lập vùng chương trình bị lỗi.
- Step Into: Nếu dòng lệnh hiện hành đang gọi một thủ tục, nhấn F8 sẽ nhảy vào bên trong thủ tục.
- Step Over: Nếu dòng lệnh hiện hành đang gọi một thủ tục, nhấn Shift-F8 sẽ chạy qua thủ tục.
- Step Out: Nếu điểm dừng đang ở trong một thủ tục, nhấn Ctrl-Shift-F8 sẽ chạy hết thủ tục và dừng ở dòng kế tiếp sau lệnh gọi thủ tục
-

2.4.2.2. Bắt lỗi

Lệnh On Error

Lệnh On Error dùng trong hàm hay thủ tục báo cho Visual basic biết cách xử lý khi lỗi xảy ra.

On Error GoTo <Nhân>

Dùng On error Goto 0 tắt xử lý lỗi.

Nhưng nếu muốn lờ đi lỗi xảy ra và tiếp tục thực hiện chương trình ta viết

On Error Resume Next

CÂU HỎI ÔN TẬP CHƯƠNG 2

1. Các thành phần của một dự án phần mềm trong VisualBasic?
2. Các bước để xây dựng một dự án phần mềm trong VisualBasic?
3. Khái niệm đối tượng (control) và một số thuộc tính cơ bản của đối tượng trong VisualBasic?
4. Trình bày về chức năng và các thuộc tính cơ bản của hộp văn bản TextBox? Viết một modul chương trình ví dụ?
5. Trình bày về chức năng và các thuộc tính cơ bản của đối tượng nhãn Label? Viết một modul chương trình ví dụ?
6. Trình bày về chức năng và các thuộc tính cơ bản của hộp menu đồ xuống ComboBox? Viết một modul chương trình ví dụ?
7. Trình bày về chức năng và các thuộc tính cơ bản của hộp danh sách liệt kê ListBox? Viết một modul chương trình ví dụ?
8. Trình bày về chức năng và các thuộc tính cơ bản của hộp đánh dấu CheckBox? Viết một modul chương trình ví dụ?
9. Trình bày về chức năng và các thuộc tính cơ bản của nút tùy chọn OptionButton? Viết một modul chương trình ví dụ?
10. Trình bày về chức năng và các thuộc tính cơ bản của nút lệnh CommandButton? Viết một modul chương trình ví dụ?
11. Trình bày về chức năng và các thuộc tính cơ bản của đối tượng lưới DataGrid? Viết một modul chương trình ví dụ?
12. Trình bày về các kiểu dữ liệu thường dùng trong VisualBasic? Cho ví dụ?
13. Trình bày về quy cách, chức năng của câu lệnh rẽ nhánh (IF, SELECT CASE) trong VisualBasic? Cho ví dụ minh họa?
14. Trình bày về quy cách, chức năng của câu lệnh chu trình (FOR, DO WHILE) trong VisualBasic? Cho ví dụ minh họa?

CHƯƠNG 3

XÂY DỰNG GIAO DIỆN CỦA ỨNG DỤNG TRONG QUẢN LÝ

3.1. Menu

Menu hay còn gọi là thực đơn đó là một bảng chứa danh sách các công việc hay chức năng mà người dùng có thể lựa chọn trong quá trình khai thác phần mềm ứng dụng.



Có 2 loại menu:

- Menu thả xuống (Drop-down menu): là dạng menu thông dụng nhất.
- Menu bật ra (Popup menu): thường hiển thị khi ta nhấn nút phải chuột.

Menu cũng là một loại điều khiển, nhưng windows sẽ kiểm soát việc vẽ menu. Lập trình viên chỉ quản lý phần xử lý các sự kiện mà thôi.

3.1.1. Dùng trình soạn thảo menu để tạo menu

Menu không chứa trong hộp công cụ như những điều khiển khác, mà được thiết kế bằng trình soạn thảo menu. Từ menu tools, chọn Menu editor để mở rộng chương trình này hoặc dùng Ctrl + E hoặc nhấn biểu tượng trong menu của Visual basic.

Chú ý, nếu chưa có biểu mẫu thì biểu tượng này không xuất hiện trên thanh công cụ.

Các thuộc tính của menu

Thuộc tính của menu không chứa trong cửa sổ thuộc tính như các điều

khiển khác mà đặt trong trình soạn thảo menu.

Thuộc tính caption: Là chuỗi ký tự hiển thị trên menu.

Thuộc tính name: Phải được đặt duy nhất và dễ nhớ. Có 2 cách đặt tên:

- Nhóm các mục có cùng cha trên menu vào chung một dãy các điều khiển và dùng chung một tên. Cách này được Visual basic hết sức khuyến khích.
- Mỗi mục có một tên riêng, nhưng nên bắt đầu bằng mnu. ví dụ mnuFile

Thuộc tính index: Dùng với dãy các điều khiển menu. Trong đó, vì có nhiều mục cùng tên nên index được dùng cho phân biệt giữa chúng với nhau.

Thuộc tính shortcut: Người sử dụng có thể nhấn chuột để chọn menu theo cách bình thường, hoặc dùng phím tắt. VD: nhấn Ctrl+C thay vì chọn Copy.

Thuộc tính Windows list: dùng trong các ứng dụng MDI. Đây là những ứng dụng có một biểu mẫu chính và nhiều biểu mẫu con. Thuộc tính windowsList ra lệnh cho Visual basic hiển thị tiêu đề của các cửa sổ con trên menu.

Thuộc tính Checked: Nếu chọn thuộc tính này, trên menu sẽ hiển thị một dấu bên cạnh. Tuy nhiên, thuộc tính này không được gán cho những mục menu đang chứa menu con.

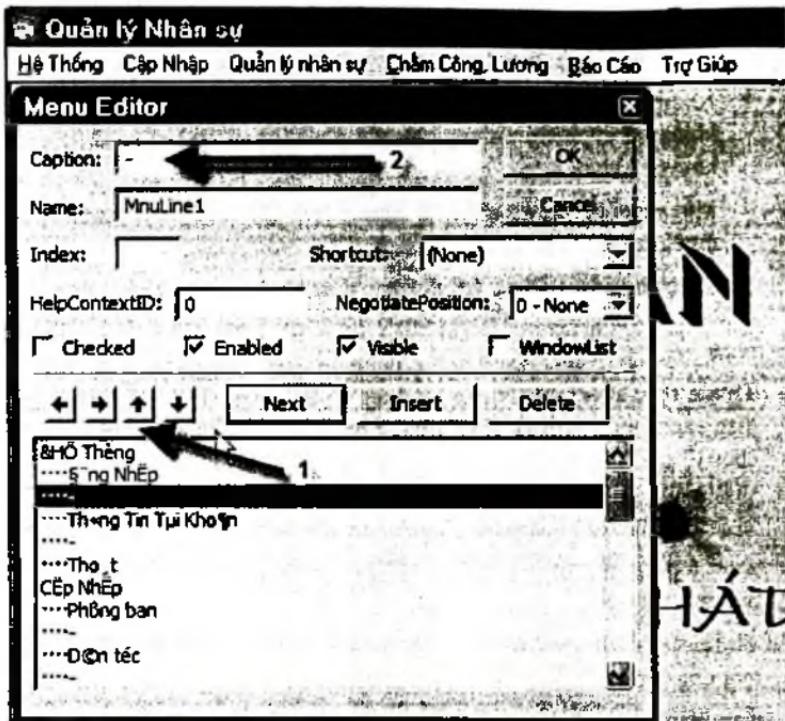
Thuộc tính enable: Nếu thuộc tính này không được chọn người sử dụng không thể chọn vào đó được.

Thuộc tính Visible: Nếu thuộc tính này không được chọn mục này sẽ biến mất khỏi màn hình.

Thuộc tính NegotiatePosition: Quản lý vị trí gắn menu trong trường hợp sử dụng các đối tượng ActivateX.

Tách nhóm menu: Nếu menu có khá nhiều mục, tốt nhất ta nên chia chúng thành nhiều nhóm nhỏ.

Chú ý:



+ Chúng ta sử dụng các nút mũi tên lên, xuống để di chuyển tới các menu được khai báo (1)

+ Ta sử dụng các nút mũi tên sang phải, sang trái để xác định cấp bậc cho menu. Menu cấp càng thấp thì càng về phía bên phải (1)

+ Muốn khai báo 1 đường kẻ để phân nhóm hệ thống menu: ta khai báo menu có caption = - (1 dấu gạch ngang) và vẫn phải khai báo tên cho nó. (2)

+ Nên khai báo hệ thống menu tại form chính (MDIForm)

3.1.2. Viết chương trình điều khiển menu

Để lập trình cho menu, ta mở cửa sổ thiết kế biểu mẫu và nhấp chuột lên mục mà ta muốn xử lý.

3.1.2.1. Pop-up menu

Ví dụ mẫu - tạo pop-up menu

Bạn tạo ra một menu file có Open, Save, Save as...

Mở cửa sổ code, trong sự kiện mouseUp của biểu mẫu, đưa vào dòng lệnh

```
Private Sub Form_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
If Button = vbRightButton Then
```

```
PopupMenu mnuFile, vbPopupMenuLeftAlign
```

```
End If
```

```
End Sub
```

Tắt thuộc tính Visible của menu file.

Chạy chương trình, khi ta nhấn chuột một menu pop-up sẽ hiển thị.

3.1.2.2. Menu thường

Viết lệnh để đáp ứng sự kiện được lựa chọn cho các menu cấp thấp nhất bằng cách tại cửa sổ thiết kế ta lựa chọn 1 dòng menu ta muốn lập trình. Xuất hiện cửa sổ lập trình, con trỏ soạn thảo đang nằm tại 1 thủ tục tương ứng với sự kiện menu được chọn. Tại đây ta sẽ viết lệnh đáp ứng sự kiện này.

3.2. Hộp thoại

Hộp thoại (dialog) là một cửa sổ làm việc của phần mềm ứng dụng mà thông qua đó ứng dụng gửi đến người dùng những thông điệp hoặc để người dùng cập nhật thông tin hay là trả lời những câu hỏi tình huống có liên quan.

Có 4 kiểu hộp thoại:

- Thông điệp
- Hộp nhập
- Các hộp thoại thông dụng
- Hộp thoại hiệu chỉnh.

3.2.1. Thông điệp (Message box)

Là dạng hộp thoại đơn giản nhất, gồm hai loại:

- Chỉ cung cấp thông tin
- Tương tác với người sử dụng.

Quy cách:

```
MsgBox(Prompt [Buttons As VbMsgBoxStyle = vbOKOnly], [Title], [HelpFile], [Context]) As VbMsgBoxResult
```

Prompt: Là chuỗi ký tự (thông điệp) sẽ hiện ra

Buttons: quy định loại thông điệp. Xem lại bảng thông điệp bên dưới

Title: Là tiêu đề của hộp thông điệp

Chiều dài thông điệp

Windows tự động cắt chuỗi khi nó quá dài, nhưng đôi khi việc này không đúng ý muốn của lập trình viên. Ta có thể làm bằng tay:

MsgBox "This is a multi-line" & chr\$(10) & "message."

Hàm chr\$() trả về ký tự có mã ASCII là tham số truyền vào chr\$(10) trả về dòng ký tự mới.

Các loại thông điệp

Hàng số	Thể hiện
vbOKOnly	OK
vbOKCancel	OK, Cancel
vbAbortRetryIgnore	Abort, Retry, Ignore
vbYesNoCancel	Yes, No, Cancel
vbYesNo	Yes, No
vbRetryCancel	Retry, Cancel

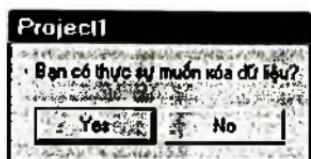
Hộp thoại có tương tác với người dùng

Ta có thể viết theo dạng

```
If MsgBox ("Ban co thuc su muon xoa du lieu?", vbYesNo) = vbYes Then
```

'Thuc hien lenh xoa du lieu

EndIf



3.2.2. Hộp nhập (Input box)

Input box ít được dùng. Lý do là:

- Không có cách nào để kiểm định dữ liệu mà người sử dụng đưa vào khi họ chưa nhấn Enter. Nếu dùng biểu mẫu do chính mình thiết kế, ta có thể đưa vào hộp văn bản và viết chương trình để xử lý sự kiện liên quan đến việc kiểm tra dữ liệu mà với Input box không thể làm được.
- InputBox chỉ cho người sử dụng nhập rất ít thông tin. Muốn lấy ra được nhiều thông tin, nên dùng biểu mẫu tự thiết kế.
- Quy cách

InputBox(Prompt, [Title], [Default], [XPos], [YPos], [HelpFile], [Context]) As String

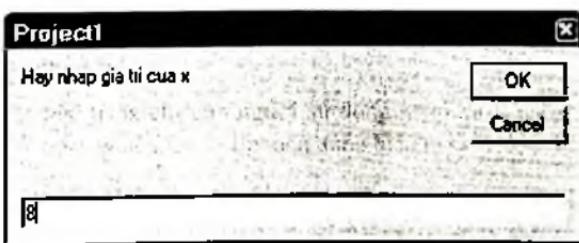
Prompt: Là chuỗi ký tự (thông điệp) sẽ hiện ra, để nghị nhập dữ liệu

Title: Là tiêu đề của hộp thông điệp

Default: Giá trị mặc định

Ví dụ:

X= InputBox("Hãy nhập giá trị của x")



3.2.3. Hộp thoại thông thường (common dialog)

Bạn có để ý thấy hầu như mọi ứng dụng trong Windows đều có cùng những dialogs để Open và Save files? Và hầu như tất cả programs đều có cùng dialogs để chọn màu, font chữ hay để in? Đó là vì các Dialogs thông dụng ấy thuộc về Common Dialog Library của MSWindows và cho phép các program gọi.

Muốn dùng các Dialogs ấy trong VB6 ta phải khai báo sử dụng thư viện điều khiển Comdlg32.ocx bằng IDE Menu command.

Project | Components... rồi chọn và Apply Microsoft Common Dialog Control 6.0.

Microsoft Common Dialog Control 6.0 cho ta sáu dạng Dialogs thông qua các method như sau:

Tên Method

Open File ShowOpen

Save File ShowSave

Color ShowColor

Font ShowFont

Print ShowPrinter

Help ShowHelp

- **Open và Save File Dialogs**

Bạn hãy mở một Project mới với một button tên CmdOpen trong Form1,

thêm điều khiển common dialog vào Form



và sử dụng

đoạn code sau để điều khiển mở hộp thoại Open.

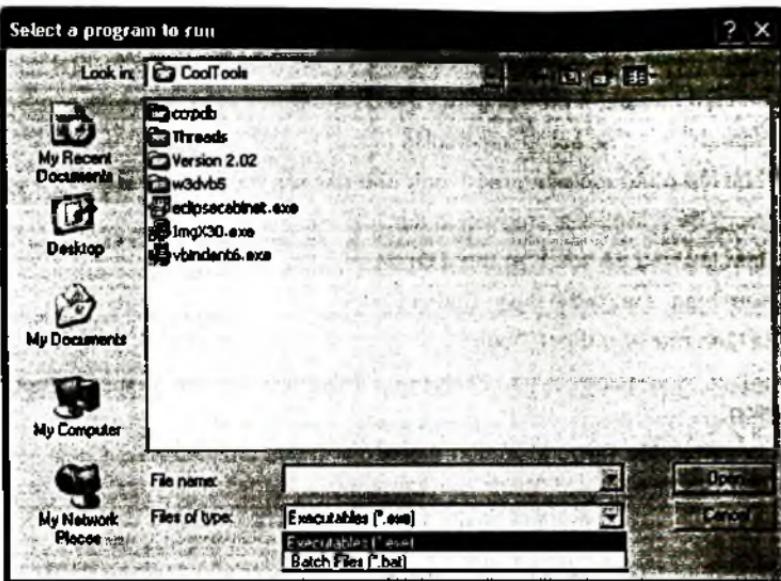
```
Private Sub CmdOpen_Click()
On Error GoTo DialogError
With CommonDialog1
.CancelError = True ' Generate Error number cdICancel if user click Cancel
.InitDir = "E:\VB6" ' Initial (i.e. default) Folder
```

```

'Thiết lập thư mục mặc định cho hộp thoại Open là E:\VB6
.Filter = "Executables (*.exe) | *.exe| Batch Files (*.bat)| *.bat"
'Thiết lập các phần mở rộng của tệp có thể được hiển thị trong cửa sổ
.FilterIndex = 1 ' Select ""Executables (*.exe) | *.exe" as default
' Thiết lập phần mở rộng mặc định cho các tệp hiển thị là *.exe
DialogTitle = "Select a program to run"
' Thiết lập tiêu đề của hộp thoại Open
ShowOpen ' Lauch the Open Dialog
' Ra lệnh mở hộp thoại Open
MsgBox "You selected " & .FileName, vbOKOnly + vbInformation, "Open Dialog"
' Lấy ra tên tệp được lựa chọn thông qua thuộc tính FileName của hộp
thoại
End With
Exit Sub
DialogError:
If Err.Number = cdlCancel Then
    MsgBox "You clicked Cancel!", vbOKOnly + vbInformation, "Open Dialog"
    Exit Sub
Else
    MsgBox "Error in Dialog's use: " & Err.Description, vbOKOnly + vbCritical,
    "Error"
    Exit Sub
End If
End Sub

```

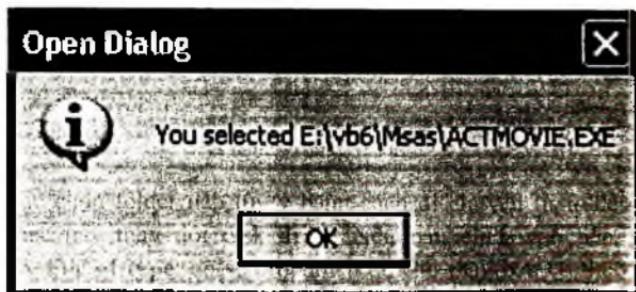
Bây giờ hãy chạy program lại và click button Open để hiển thị **OpenDialog**.



Bạn có thể chọn folder nào tùy ý bằng cách di chuyển từ folder này qua folder khác hay thay đổi disk drive. Nếu bạn click vào bên phải của combobox **File of type**, nó sẽ dropdown để cho thấy bạn có thể chọn một trong hai loại Files như liệt kê trong statement:

```
.Filter = "Executables (*.exe) | *.exe| Batch Files (*.bat)| *.bat"
```

Sau khi chọn một Filename có sẵn hay đánh một tên vào **File name** textbox, bạn click **Open**. Sau đó, CommonDialog1.Filename sẽ chứa tên file bạn đã chọn hay đánh vào.

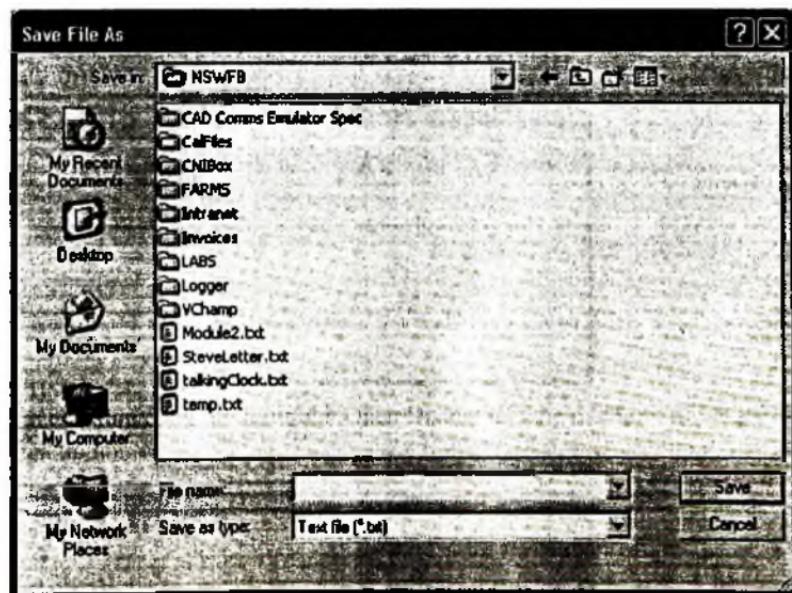


Vì ta cho **.CancelError = True** nên nếu user click Cancel program sẽ

generate một Error số 32755 (cdlCancel). Ở đây ta bắt Error ấy bằng cách dùng **On Error GoTo DialogError** và thử Err.Number= cdlCancel để hiển thị Error message dưới đây:

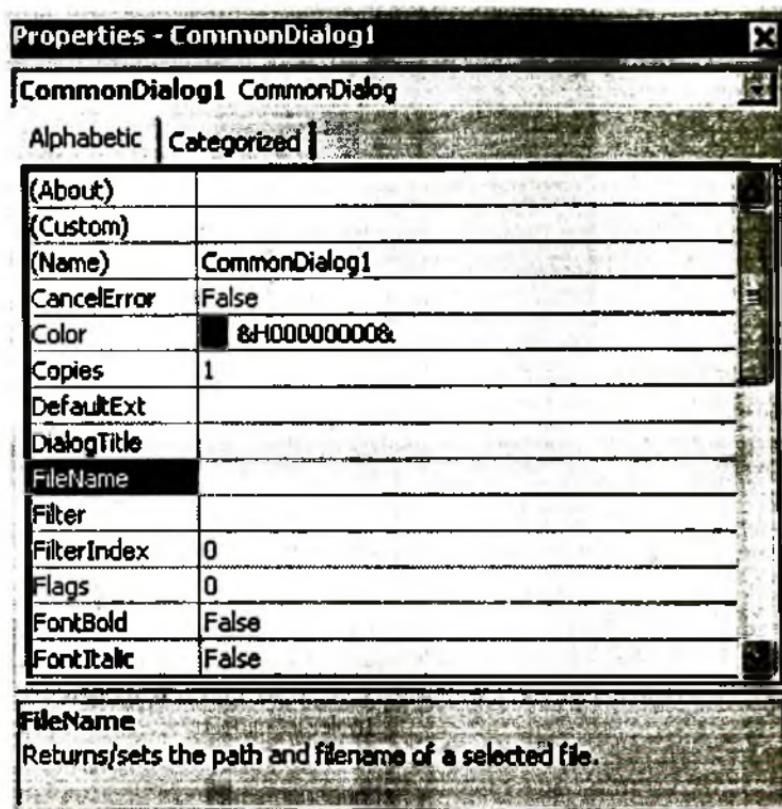


Save Dialog cũng tương tự như Open Dialog, ta dùng method **ShowSave** để hiển thị nó.

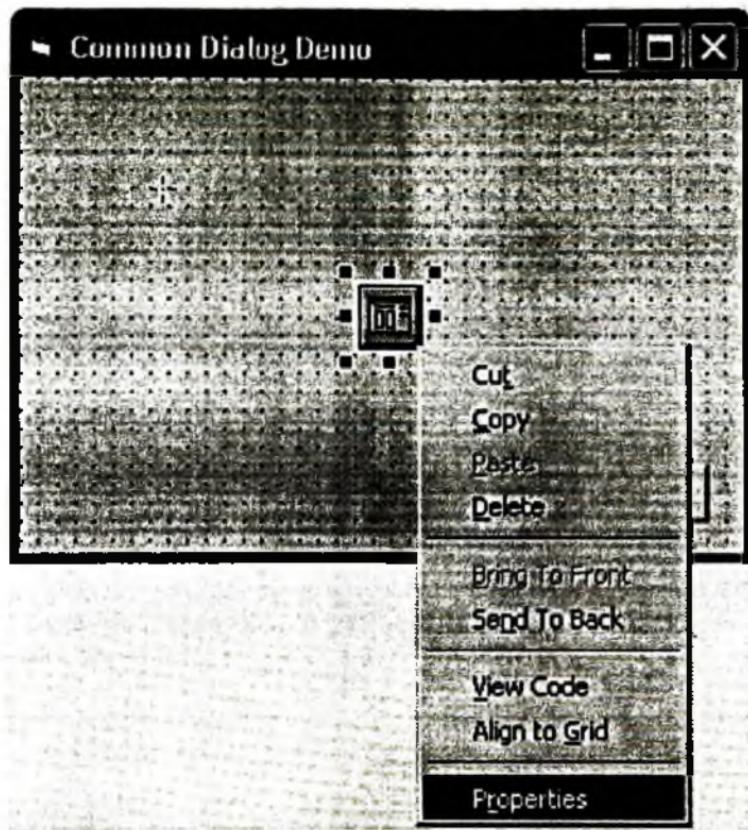


Trong thí dụ trên ta định nghĩa các properties của CommonDialog1 bằng code. Bạn cũng có thể dùng

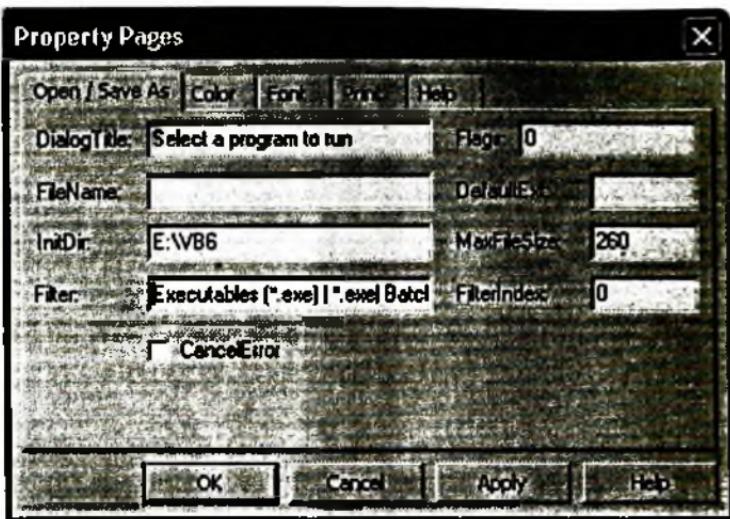
Properties Windows để định nghĩa chúng như dưới đây:



Ngoài ra, bạn cũng có thể dùng các trang Properties của CommonDialog1 để định nghĩa Properties lúc thiết kế bằng cách right click Commondialog1 trên Form1 rồi chọn Properties:

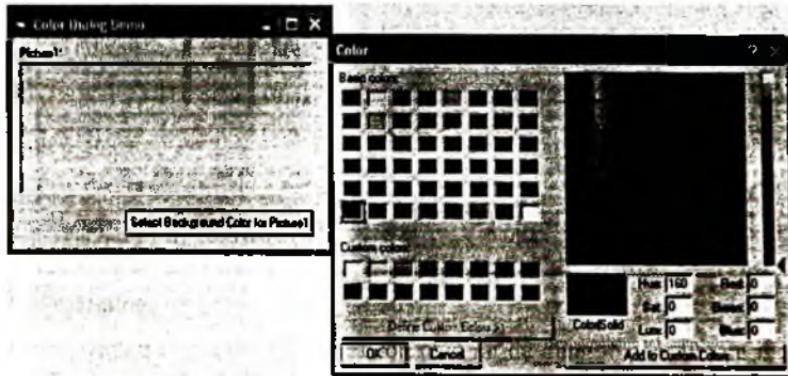


Properties Pages Dialog sẽ hiển thị với Tab Open/Save As có sẵn lúc đầu, bạn có thể đánh các tin tức như sau:



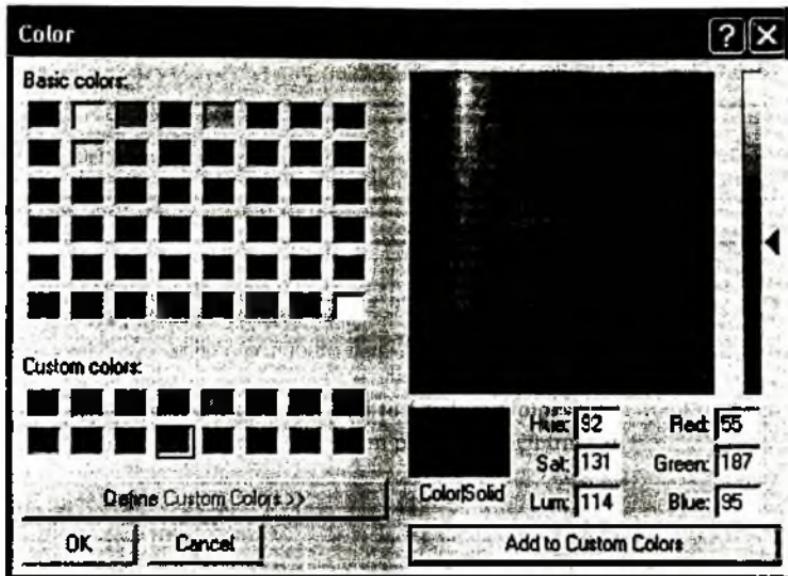
Color Dialog

Color Dialog cho user một cách chọn màu rất dễ dùng. Ngoài những màu có sẵn, user có thể tự tạo ra một màu rồi cho nó thêm vào trong bảng màu được cung cấp, gọi là **Windows Palette** bằng cách click button **Add to Custom Colors**.



Bạn tạo ra một màu bằng cách click chuột có màu theo ý trong bảng màu lớn hình vuông rồi nắm hình tam giác bên phải kéo lên, kéo xuống để thay đổi độ đậm của màu như hiển thị trong hộp vuông **Color|Solid**. Khi vừa ý với

màu hiển thị, bạn click button **Add to Custom Colors**, màu ấy sẽ được cho thêm vào nhóm **Custom Colors** nằm phía dưới, bên trái.



Ta dùng method **ShowColor** để hiển thị Color Dialog. Sau khi user đã chọn một màu rồi, ta có thể trực tiếp assign nó cho property **ForeColor** hay **BackColor** của một control. Trong thí dụ dưới đây cái màu mà user vừa chọn được assigned cho background của picturebox **Picture1**:

```
Private Sub CmdSelectColor_Click()
On Error GoTo NoColorChosen
With CommonDialog1
.CancelError = True
' Entire dialog box is displayed, including the Define Custom Colors
section
.Flags = cdiICCFullOpen
.ShowColor ' Launch the Color Dialog
Picture1.BackColor = .Color ' Assign selected color to background of
Picture1
```

```

Exit Sub
End With
NoColorChosen:
' Get here if user clicks the Cancel button
MsgBox "You did not select a color!", vbInformation, "Cancelled"
Exit Sub
End Sub

```

Font Dialog

Font Dialog cho ta chọn Font cho màn ảnh hay printer và chọn màu để dùng cho chữ của Font. Ta dùng method **ShowFont** để hiển thị **FontDialog**. Các chi tiết trình bày trong Font Dialog tùy thuộc vào trị số của Flags như sau:

Constant	Trị số	Hỗn quả
cdlCFScreenFonts	1	Hiển thị các Font của màn ảnh
cdlCFPrinterFonts	2	Hiển thị các Font của printer, chưa chắc tất cả đều được printer hỗ trợ
cdlCFBoth	3	Hiển thị các Font màn ảnh và printer
cdlCFScaleableOnly	&H20000	Hiển thị các scalable Font như TrueType fonts mà bạn đã chỉ vào máy

Nếu bạn muốn cho user có thể chọn màu cho font thì thêm 256 vào trị số của Flags. Dưới đây là code để cho user chọn Font và màu của Label1.

```

Private Sub CmdSelectFont_Click()
On Error GoTo NoFontChosen
CommonDialog1.CancelError = True
' Causes the dialog box to list only the screen fonts supported by the
system.
CommonDialog1.Flags = cdlCFScreenFonts + 256
' Add 256 to include Color option
CommonDialog1.ShowFont ' Launch the Font Dialog

```

Áp dụng các lựa chọn cho Font chữ của Label1

With Label1.Font

.Bold = CommonDialog1.FontBold

.Italic = CommonDialog1.FontItalic

.Name = CommonDialog1.FontName

.Size = CommonDialog1.FontSize

.Strikethrough = CommonDialog1.FontStrikethru

.Underline = CommonDialog1.FontUnderline

End With

Label1.ForeColor = CommonDialog1.Color

Label1.Caption = "Hello world!!!, this is a Font Dialog Demo"

Exit Sub

NoFontChosen:

MsgBox "No font was chosen!", vbInformation, "Cancelled"

Exit Sub

End Sub

Chú ý: Nếu bạn quên cho Flags một trong những hàng số nói trên program sẽ cho một Error message như sau:

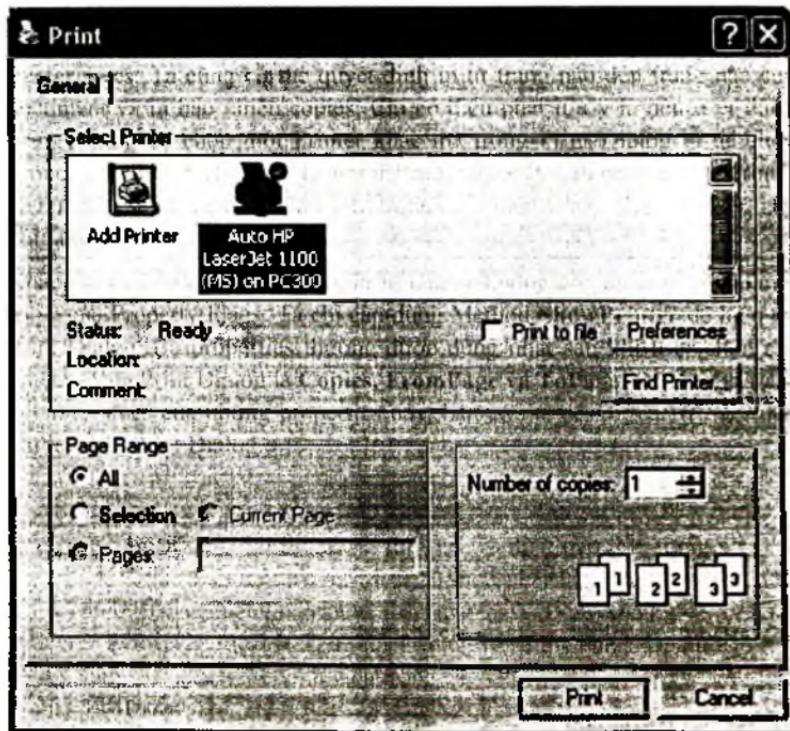


Print Dialog

Print Font cho ta một giao diện cũng giống như trong Microsoft Office để chọn những tùy chọn về việc in. Với Print Dialog ta có thể chọn printer nào với những đặc tính nào bằng cách click button **Properties** hay button

Preferences. Ta cũng có thể quyết định in từ trang nào đến trang nào của document và in bao nhiêu copies. Chỉ có điều phải lưu ý là nếu user dùng Print Dialog để chọn một Printer khác mà trong Print Dialog ta đã chọn **Property PrinterDefault = True** thì Printer ấy sẽ trở thành Default Printer và nó cũng sẽ có hiệu lực vĩnh viễn trong cả Windows cho đến khi user thay đổi lại.

Khác với Color và Font Dialogs, Print Dialog không đòi hỏi ta phải cho một trị số của Property Flags. Ta chỉ cần dùng Method **ShowPrinter** để hiển thị Print Dialog. Ba properties thường được dùng nhất sau khi user chọn các nhiệm vụ của Print Dialog là **Copies**, **FromPage** và **ToPage**. Để cho user các default values của những properties này, bạn có thể để sẵn các trị số trước khi hiển thị Print Dialog.



Dưới đây là code mẫu dùng print Dialog:

```
Private Sub CmdSelectPrinter_Click()
With CommonDialog1
    .FromPage = 1
    .ToPage = 1
    .Copies = 1
    .ShowPrinter
End With
End Sub
```

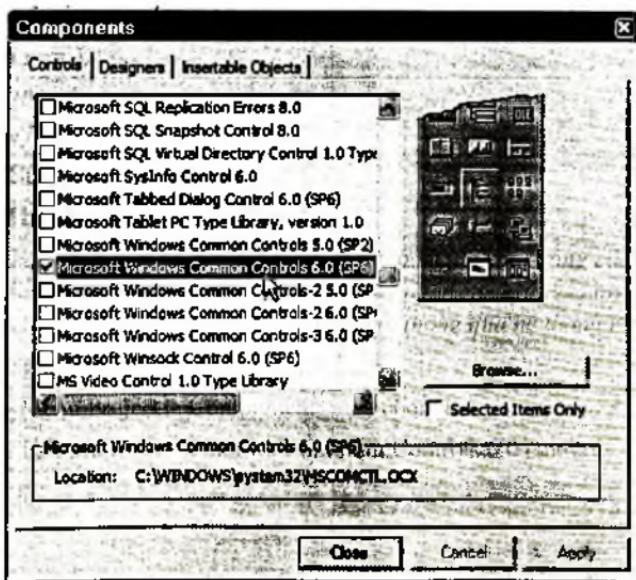
3.3. Thanh công cụ (Toolbar)

Thanh công cụ là một tập hợp các nút bấm, để khởi động các chức năng hay thực hiện một nhiệm vụ nào đó của phần mềm ứng dụng, mà người dùng có thể lựa chọn thay vì phải truy suất thông qua hệ thống menu.

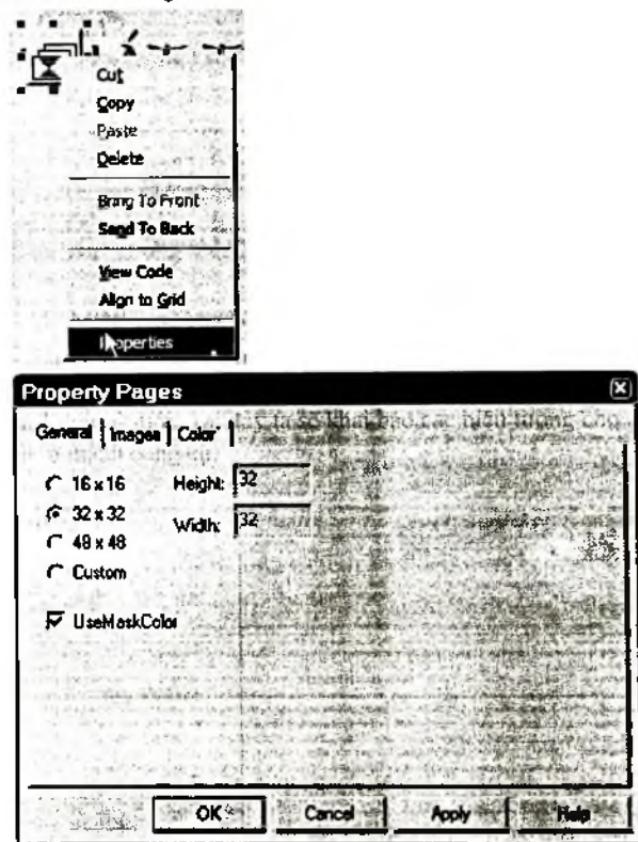
3.3.1. Thiết kế thanh công cụ

- Menu Project > Component > lựa chọn dòng:

Chọn Microsoft Windows Common Controls 6.0 (sp6)

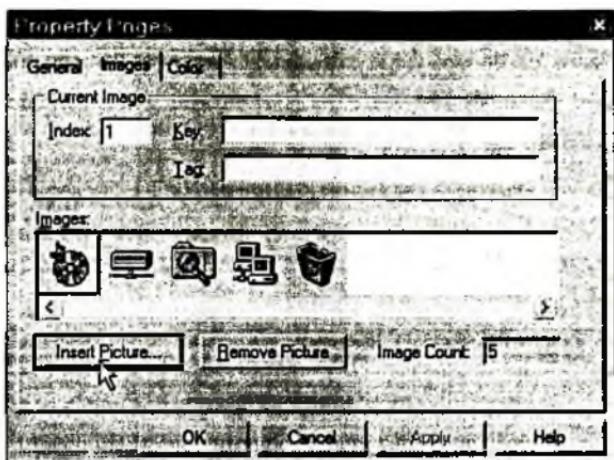


- Mở MDIForm, mở hộp Toolbox
- Lựa chọn biểu tượng ImageList ở trong ToolBox và thêm nó vào form chính. Nhấn phải chuột vào đối tượng ImageList => Properties làm hiện ra 1 cửa sổ mới. Tại đây ta sẽ khai báo các biểu tượng cho các nút lệnh sẽ có trên thanh công cụ.

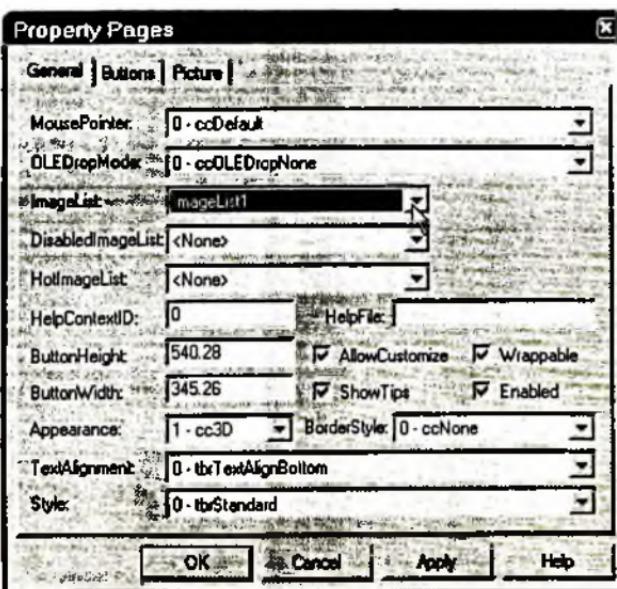


- + Tại thẻ General, chọn kích thước cho các biểu tượng 16x16, kích thước icon giống trên VB (nên chọn 32x32)
- + Thẻ Image, bấm chuột vào nút Insert Picture để tìm tới thư mục chứa danh sách các biểu tượng đang có trên máy, thông thường là C:\Program File\Microsoft Visual Studio\Commands\Graphic. Hoặc thư mục nào đó chứa biểu tượng trên máy. Tìm tệp “*.icon”, lấy đường dẫn.

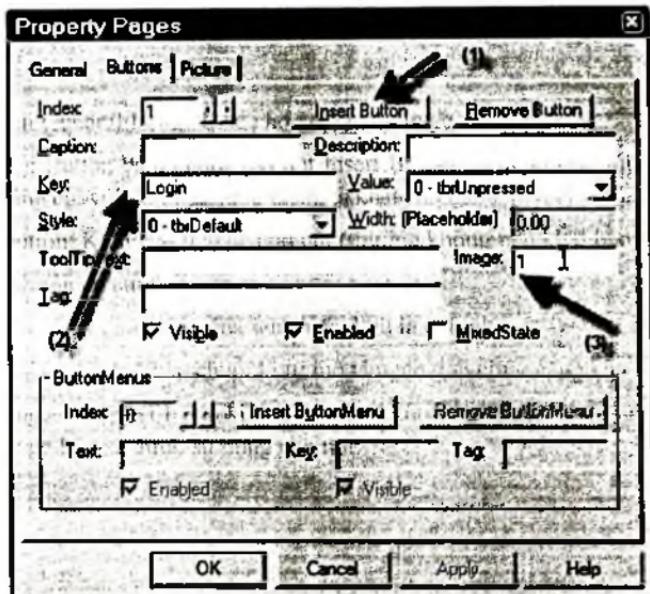
+ Lựa chọn các biểu tượng cần thêm vào dự án > open



- Nhấn phải chuột vào thanh công cụ vừa được thêm ở trước, chọn Properties. Khi vừa thêm có thẻ thành công cụ trắng. Properties cho phép khai báo các nút lệnh cùng với các thuộc tính khai báo



- Tại General/Image List, chọn tên của Image List
- Tại Button bấm chuột vào nút Insert Button để thêm các nút mới vào thanh công cụ, với mỗi nút được thêm, ta khai báo thuộc tính cho nó, gồm có:
 - Caption: Khai báo tiêu đề cho nút (thường không nên khai báo tiêu đề)
 - Key: tên của nút
 - Style: Lựa chọn kiểu của nút (mặc định là default)
 - ToolTipText: Nhập chuỗi kí tự mô tả ngắn gọn chức năng của nút
 - Image: Nhập STT cho hình ảnh của biểu tượng mà đã định nghĩa trong mục Image List sẽ được sử dụng cho nút.



- Nếu muốn tạo ra 1 đường | ngăn cách các nút thì tạo 1 nút mới có Style = 3 - Separator.



Ghi chú: tạo các nút Flat (phẳng):

Tại thẻ General > trong mục Style chọn = 1 - Flat > Apply



3.3.2. Lập trình cho thanh công cụ

- Viết lệnh để đáp ứng sự kiện nút được bấm: Nháy kép chuột vào thanh công cụ, thủ tục hiện thời là Button Click. Ta lập trình để đáp ứng sự kiện một nút nào đó được bấm theo cấu trúc như sau:

Select Case Button.Key

Case "Login"

MsgBox "Đăng nhập hệ thống"

Case "End"

End

Case "Report"

Form1.Show

End Select

Dựa vào thuộc tính .Key (tên của nút), đã được định nghĩa khi khai báo các nút trên thanh công cụ, cùng với cấu trúc Select Case ta đặt các lệnh cần thực hiện vào ngay sau mục kiểm tra tên của nút.

Như trong ví dụ trên, khi người dùng bấm nút có Key là "Login" thì câu lệnh *MsgBox "Đăng nhập hệ thống"* sẽ được thực hiện để đưa ra màn hình câu thông báo "Đăng nhập hệ thống".

Chú ý: Tên của mỗi nút lệnh (.Key) có phân biệt chữ hoa, chữ thường

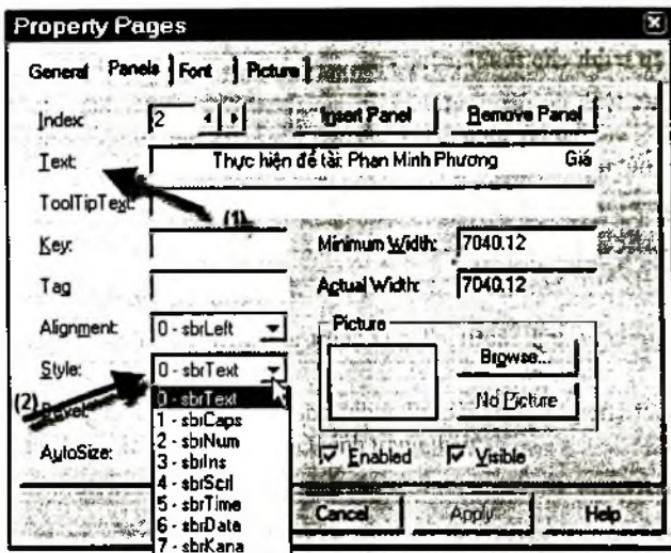
3.4. Thanh trạng thái (Statusbar)

Thanh trạng thái thường nằm phía dưới cùng của màn hình chính của ứng dụng được chia thành nhiều ô, mỗi ô có chứa các thông tin khác nhau thông báo về trạng thái ví dụ thời gian hiện tại, tên người dùng hiện thời, tên chức năng hiện thời, kết quả công việc vừa được thực hiện...

Khai báo

Menu Project / Component / chọn MS Common Control 6.0

- Mở MDIForm, ToolBox
- Chọn điều khiển StatusBar trong hộp ToolBox, định vị tại dòng cuối cùng của form chính
- Nháy phải chuột vào StatusBar, chọn Properties > Property Pages: cho phép khai báo các thành phần của thanh trạng thái



- Tại thẻ Panels:

Chọn nút Insert Panels để thêm một ô (panel) tại thanh trạng thái
Tại mục Style: chọn kiểu của ô tương ứng. = 0: hiển thị chuỗi kí tự

- = 5: tự động hiển thị thời gian
- = 6: tự động hiển thị ngày tháng

AutoSize: chọn = 2 để tự động điều chỉnh Size (Muốn hiển thị VD: Status > vào text)

- Để lập trình hiển thị chuỗi kí tự như mong muốn trên thanh trạng thái
StatusBar1.Panels(1).Text = "<chuỗi kí tự mong muốn>"

*	Thực hiện để tài Phan Minh Phương	Giáo viên hướng dẫn Ths. Tạ Văn Hải Sơn	14/10/2008
---	-----------------------------------	---	------------

CÂU HỎI ÔN TẬP CHƯƠNG 3

1. Các bước để khai báo và sử dụng menu trong VisualBasic? Cho ví dụ minh họa?
2. Các bước để khai báo và sử dụng thanh công cụ ToolBars trong VisualBasic? Cho ví dụ minh họa?
3. Nguyên tắc cơ bản trong thiết kế giao diện của các Form chức năng của một dự án phần mềm?
4. Nguyên tắc cơ bản trong thiết kế hệ thống menu của một dự án phần mềm?
5. Hãy thiết kế hệ thống menu chức năng, thanh công cụ và thanh trạng thái cho các phần mềm ứng dụng sau:
 - a. Quản lý việc mượn, trả sách tại thư viện
 - b. Quản lý nhập, xuất hàng hóa và công nợ khách hàng tại một công ty thương mại
 - c. Quản lý điểm của sinh viên tại trường ĐH KTQD hiện nay
 - d. Quản lý cán bộ công chức cho một cơ quan nhà nước
 - e. Quản lý cho thuê băng đĩa

Chương 4

LẬP TRÌNH LIÊN KẾT VÀ KHAI THÁC CƠ SỞ DỮ LIỆU

4.1. Các khái niệm liên quan đến cơ sở dữ liệu

- **Thực thể:** (entity) Là một đối tượng mà nhà quản lý muốn lưu trữ thông tin. Thông thường, tên thực thể sẽ trùng với tên bảng trong cơ sở dữ liệu.
- **Trường (Fields)** là các thuộc tính hay tính chất của đối tượng. Thông thường mỗi trường tương ứng với một cột trong bảng dữ liệu. Có các trường tự thân và cả các trường nhân tạo. Trường nhân tạo là trường được nhà quản lý gán thêm cho đối tượng để thuận tiện cho việc quản lý.
- **Khóa (Key)** là một hay nhiều trường mà giá trị của một hay nhiều trường đó xác định một cách duy nhất cá thể mà nó mô tả. Thông thường để thuận tiện, người ta tạo ra một trường nhân tạo cho đối tượng đó là trường mã số, đóng vai trò là khóa.
- **Bản ghi (Records)** là một tập hợp các thông tin phản ánh về một cá thể được lưu trong bảng cơ sở dữ liệu. Mỗi bản ghi thường tương ứng với một dòng trong bảng.
- **Bảng (table)** là tập hợp các thông tin lưu trữ về thực thể quản lý. Mỗi bảng thường được tổ chức theo cấu trúc gồm các dòng và các cột. Mỗi cột tương ứng với một trường còn mỗi dòng tương ứng với một bản ghi.
- **Cơ sở dữ liệu (Database)** là một hay nhiều bảng dữ liệu có quan hệ với nhau, phục vụ cho việc quản lý một tổ chức hay một nhóm đối tượng nào đó.

Ví dụ một cơ sở dữ liệu quan hệ

Bảng KHOA

MKTG	TENKHOA	DIENG
DHTG1	Khoa Toán cơ - Tin học	054622407
DHTG2	Khoa Công nghệ thông tin	0546225767
DHTG3	Khoa Vật lý	0546224467
DHTG4	Khoa Hóa học	0546224467
...

Bảng LOP

MALOP	TENLOP	MKTG	MAKHOA	S
C24101	Toán K24 24	Chinh quy 2000	5	DHTG1
C24102	Tin K24 24	Chinh quy 2000	6	DHTG2
C24103	Lý K24 24	Chinh quy 2000	7	DHTG3
C24101	Sinh K24 24	Chinh quy 2000	5	DHTG5

Bảng SINHVIEN

MSV	HOTEN	LBH	NAM sinh	NAMHIS	GIOTLINE	NGUON	MAKHOA	MALOP
0241010001	Ngô Thị Nhật	Ánh	Nov 27 1982	10	Quảng Ninh, Quảng Bình	C24101		
0241010002	Nguyễn Thị Ngọc	Ánh	Mar 21 1983	0	Tân Kỳ, Nghệ An	C24101		
0241010003	Ngô Việt	Bắc	May 11 1982	1	Yên Khánh, Ninh Bình	C24101		
0241010004	Nguyễn Định	Bình	Oct 6 1982	1	Huế	C24101		
0241010005	Rô Bàng	Chiến	Jan 20 1982	1	Phong Điền, TT Huế	C24101		
0241020001	Nguyễn Tuấn	Ánh	Jul 15 1979	1	Đo Lĩnh, Quảng Trị	C24102		
0241020002	Trần Thái Kim	Ánh	Nov 4 1982	0	Phong Điền, TT Huế	C24102		
0241020003	Võ Đức	Ánh	May 24 1982	1	Huế	C24102		
0241020004	Nguyễn Công	Binh	Jun 6 1979	1	Thăng Bình, Quảng Nam	C24102		
0241020005	Nguyễn Thành	Binh	Apr 24 1982	1	Huế	C24102		
...

Cơ sở dữ liệu trên gồm 3 bảng: bảng KHOA, bảng LOP và bảng SINHVIEN có quan hệ với nhau. Trong mỗi bảng đều có 1 trường khóa. Bảng KHOA có trường MAKHOA; bảng LOP có trường MALOP và bảng SINHVIEN có trường MASV. Bên cạnh đó còn có các trường tự thân khác phục vụ cho việc quản lý sinh viên.

Các bảng trong cơ sở dữ liệu không tồn tại độc lập với nhau mà có mối quan hệ ràng buộc lẫn nhau về mặt dữ liệu. Giá trị dữ liệu xuất hiện ở bảng này phải xuất hiện trước trong một bảng khác.

4.2. Ngôn ngữ truy vấn có cấu trúc SQL

4.2.1. Giới thiệu chung

SQL, viết tắt của Structured Query Language (ngôn ngữ hỏi có cấu trúc), do Viện tiêu chuẩn Quốc gia Mỹ (ANSI) tạo ra, là công cụ sử dụng để tổ chức, quản lý và truy xuất dữ liệu được lưu trữ trong các cơ sở dữ liệu. SQL là một hệ thống ngôn ngữ bao gồm tập các câu lệnh sử dụng để tương tác với cơ sở dữ liệu quan hệ.

Tên gọi ngôn ngữ truy vấn có cấu trúc phản ánh chúng ta liên tưởng đến một công cụ (ngôn ngữ) dùng để truy xuất dữ liệu trong các cơ sở dữ liệu. Thực sự mà nói, khả năng của SQL vượt xa so với một công cụ truy xuất dữ liệu, mặc dù đây là mục đích ban đầu khi SQL được xây dựng nên

và truy xuất dữ liệu vẫn còn là một trong những chức năng quan trọng của nó. SQL được sử dụng để điều khiển tất cả các chức năng mà một hệ quản trị cơ sở dữ liệu cung cấp cho người dùng bao gồm:

- **Định nghĩa dữ liệu:** SQL cung cấp khả năng định nghĩa các cơ sở dữ liệu, các cấu trúc lưu trữ và tổ chức dữ liệu cũng như mối quan hệ giữa các thành phần dữ liệu.
- **Truy xuất và thao tác dữ liệu:** Với SQL, người dùng có thể dễ dàng thực hiện các thao tác truy xuất, bổ sung, cập nhật và loại bỏ dữ liệu trong các cơ sở dữ liệu.
- **Điều khiển truy cập:** SQL có thể được sử dụng để cấp phát và kiểm soát các thao tác của người sử dụng trên dữ liệu, đảm bảo sự an toàn cho cơ sở dữ liệu.
- **Đảm bảo toàn vẹn dữ liệu:** SQL định nghĩa các ràng buộc toàn vẹn trong cơ sở dữ liệu nhờ đó đảm bảo tính hợp lệ và chính xác của dữ liệu trước các thao tác cập nhật cũng như các lỗi của hệ thống.

Như vậy, có thể nói rằng SQL là một ngôn ngữ hoàn thiện được sử dụng trong các hệ thống cơ sở dữ liệu và là một thành phần không thể thiếu trong các hệ quản trị cơ sở dữ liệu. Mặc dù SQL không phải là một ngôn ngữ lập trình như C, C++, Java,... song các câu lệnh mà SQL cung cấp có thể được nhúng vào trong các ngôn ngữ lập trình nhằm xây dựng các ứng dụng tương tác với cơ sở dữ liệu.

Khác với các ngôn ngữ lập trình quen thuộc như C, C++, Java,... SQL là ngôn ngữ có tính khai báo. Với SQL, người dùng chỉ cần mô tả các yêu cầu cần phải thực hiện trên cơ sở dữ liệu mà không cần phải chỉ ra cách thức thực hiện các yêu cầu như thế nào. Chính vì vậy, SQL là ngôn ngữ dễ tiếp cận và dễ sử dụng.

4.2.2. Vai trò của SQL

Bản thân SQL không phải là một hệ quản trị cơ sở dữ liệu, nó không tồn tại độc lập. SQL thực sự là một phần của hệ quản trị cơ sở dữ liệu, nó xuất hiện trong các hệ quản trị cơ sở dữ liệu với vai trò ngôn ngữ và là công cụ giao tiếp giữa người sử dụng và hệ quản trị cơ sở dữ liệu.

Trong hầu hết các hệ quản trị cơ sở dữ liệu quan hệ, SQL có những vai trò như sau:

- **SQL là ngôn ngữ hỏi có tính tương tác:** Người sử dụng có thể dễ

dàng thông qua các trình tiện ích để gửi các yêu cầu dưới dạng các câu lệnh SQL đến cơ sở dữ liệu và nhận kết quả trả về từ cơ sở dữ liệu.

- **SQL là ngôn ngữ lập trình cơ sở dữ liệu:** Các lập trình viên có thể nhúng các câu lệnh SQL vào trong các ngôn ngữ lập trình để xây dựng nên các chương trình ứng dụng giao tiếp với cơ sở dữ liệu.
- **SQL là ngôn ngữ quản trị cơ sở dữ liệu:** Thông qua SQL, người quản trị cơ sở dữ liệu có thể quản lý được cơ sở dữ liệu, định nghĩa các cấu trúc lưu trữ dữ liệu, điều khiển truy cập cơ sở dữ liệu,...
- **SQL là ngôn ngữ cho các hệ thống khách/chủ (client/server):** Trong các hệ thống cơ sở dữ liệu khách/chủ, SQL được sử dụng như là công cụ để giao tiếp giữa các trình ứng dụng phía máy khách với máy chủ cơ sở dữ liệu.
- **SQL là ngôn ngữ truy cập dữ liệu trên Internet:** Cho đến nay, hầu hết các máy chủ Web cũng như các máy chủ trên Internet sử dụng SQL với vai trò là ngôn ngữ để tương tác với dữ liệu trong các cơ sở dữ liệu.
- **SQL là ngôn ngữ cơ sở dữ liệu phân tán:** Đối với các hệ quản trị cơ sở dữ liệu phân tán, mỗi một hệ thống sử dụng SQL để giao tiếp với các hệ thống khác trên mạng, gửi và nhận các yêu cầu truy xuất dữ liệu với nhau.
- **SQL là ngôn ngữ sử dụng cho các cổng giao tiếp cơ sở dữ liệu:** Trong một hệ thống mạng máy tính với nhiều hệ quản trị cơ sở dữ liệu khác nhau, SQL thường được sử dụng như là một chuẩn ngôn ngữ để giao tiếp giữa các hệ quản trị cơ sở dữ liệu.

4.2.3. Các câu lệnh SQL thường dùng

a. Câu lệnh *SELECT*

Câu lệnh SELECT được sử dụng để truy xuất dữ liệu từ các dòng và các cột của một hay nhiều bảng, khung nhìn. Câu lệnh này có thể dùng để thực hiện phép chọn (tức là truy xuất một tập con các dòng trong một hay nhiều bảng), phép chiếu (tức là truy xuất một tập con các cột trong một hay nhiều bảng) và phép nối (tức là liên kết các dòng trong hai hay nhiều bảng để truy xuất dữ liệu). Ngoài ra, câu lệnh này còn cung cấp khả năng thực hiện các thao tác truy vấn và thống kê dữ liệu phức tạp khác.

Cú pháp chung của câu lệnh SELECT có dạng:

```
SELECT [ALL | DISTINCT][TOP n] danh_sách_chọn
[INTO tên_bảng_mới]
FROM danh_sách_bảng/khung_nhin
[WHERE điều_kiện]
[GROUP BY danh_sách_cột]
[HAVING điều_kiện]
[ORDER BY cột_sắp_xếp]
```

Điều cần lưu ý đầu tiên đối với câu lệnh này là các thành phần trong câu lệnh SELECT nếu được sử dụng phải tuân theo đúng thứ tự như trong cú pháp. Nếu không, câu lệnh sẽ được xem là không hợp lệ.

Câu lệnh SELECT được sử dụng để tác động lên các bảng dữ liệu và kết quả của câu lệnh cũng được hiển thị dưới dạng bảng, tức là một tập hợp các dòng và các cột.

Ví dụ: Kết quả của câu lệnh sau đây cho biết mã lớp, tên lớp và hệ đào tạo của các lớp hiện có.

```
SELECT malop,tenlop,hedaotao
```

```
FROM lop
```

MÃ KHOA	TÊN KHOA	ĐIỆN THOẠI
DHT01	Khoa Toán cơ - Tin học	054822407
DHT02	Khoa Công nghệ thông tin	054826767
DHT03	Khoa Vật lý	054823462
DHT04	Khoa Hóa học	054823951
DHT05	Khoa Sinh học	054822934
DHT06	Khoa Địa lý - Địa chất	054823837
DHT07	Khoa Ngữ văn	054821133
DHT08	Khoa Lịch sử	054823833
DHT09	Khoa Báo - Lê Nin	054825698
DHT10	Khoa Luật	054821135

Mệnh đề FROM

Mệnh đề FROM trong câu lệnh SELECT được sử dụng nhằm chỉ định các bảng và khung nhìn cần truy xuất dữ liệu. Sau FROM là danh sách tên của

các bảng và khung nhìn tham gia vào truy vấn, tên của các bảng và khung nhìn được phân cách nhau bởi dấu phẩy.

Ta có thể sử dụng các bí danh cho các bảng hay khung nhìn trong câu lệnh SELECT. Bí danh được gán trong mệnh đề FROM bằng cách chỉ định bí danh ngay sau tên bảng.

Ví dụ: câu lệnh sau gán bí danh là a cho bảng khoa

SELECT * FROM khoa a

Danh sách chọn trong câu lệnh SELECT

Danh sách chọn trong câu lệnh SELECT được sử dụng để chỉ định các trường, các biểu thức cần hiển thị trong các cột của kết quả truy vấn. Các trường, các biểu thức được chỉ định ngay sau từ khoá SELECT và phân cách nhau bởi dấu phẩy. Sử dụng danh sách chọn trong câu lệnh SELECT bao gồm các trường hợp sau:

- Chọn tất cả các cột trong bảng

Khi cần hiển thị tất cả các trường trong các bảng, sử dụng ký tự * trong danh sách chọn thay vì phải liệt kê danh sách tất cả các cột. Trong trường hợp này, các cột được hiển thị trong kết quả truy vấn sẽ tuân theo thứ tự mà chúng đã được tạo ra khi bảng được định nghĩa.

Ví dụ: Câu lệnh

SELECT * FROM lop

cho kết quả như sau:

MALOP	TENLOP	KHOA	HEDAO TÀO	NĂM HỌP HỌC	MÃ KHOA
C24101	Toán K24	24	Chinh quy	2000	DHT01
C24102	Tin K24	24	Chinh quy	2000	DHT02
C24103	Lý K24	24	Chinh quy	2000	DHT03
C24301	Sinh K24	24	Chinh quy	2000	DHT05
C25101	Toán K25	25	Chinh quy	2001	DHT01
C25102	Tin K25	25	Chinh quy	2001	DHT02
C25103	Lý K25	25	Chinh quy	2001	DHT03
C25301	Sinh K25	25	Chinh quy	2001	DHT05
C26101	Toán K26	26	Chinh quy	2002	DHT01
C26102	Tin K26	26	Chinh quy	2002	DHT02

- Chọn một số trường trong bảng

Trong trường hợp cần chỉ định cụ thể các cột cần hiển thị trong kết quả truy vấn, ta chỉ định danh sách các tên cột trong danh sách chọn. Thứ tự của các cột trong kết quả truy vấn tuân theo thứ tự của các trường trong danh sách chọn.

Ví dụ: Câu lệnh

```
SELECT malop,tenlop,namnhaphoc,khoa
```

```
FROM lop
```

cho biết mã lớp, tên lớp, năm nhập học và khoá của các lớp và có kết quả như sau:

MALOP	TENLOP	NAMNHAPHOC	KHOA
C24101	Toán K24	2000	24
C24102	Tin K24	2000	24
MALOP	TENLOP	NAMNHAPHOC	KHOA
C24103	Lý K24	2000	24
C24301	Sinh K24	2000	24
C25101	Toán K25	2001	25
C25102	Tin K25	2001	25
C25103	Lý K25	2001	25
C25301	Sinh K25	2001	25
C26101	Toán K26	2002	26
C26102	Tin K26	2002	26

Lưu ý: Nếu truy vấn được thực hiện trên nhiều bảng/khung nhìn và trong các bảng/khung nhìn có các trường trùng tên thì tên của những trường này nếu xuất hiện trong danh sách chọn phải được viết dưới dạng:

ten_bang.ten_truong

Ví dụ:

```
SELECT malop, tenlop, lop.makhoa, tenkhoa
```

```
FROM lop, khoa
```

```
WHERE lop.malop = khoa.makhoa
```

- Thay đổi tiêu đề các cột

Trong kết quả truy vấn, tiêu đề của các cột mặc định sẽ là tên của các trường tương ứng trong bảng. Tuy nhiên, để các tiêu đề trở nên thân thiện hơn, ta có thể đổi tên các tiêu đề của các cột. Để đặt tiêu đề cho một cột nào

đó, ta sử dụng cách viết:

`tiêu_dè_cột = tên_trường`

hoặc `tên_trường AS tiêu_dè_cột`

hoặc `tên_trường tiêu_dè_cột`

Ví dụ: Câu lệnh dưới đây:

```
SELECT 'Mã lớp'= malop,tenlop 'Tên lớp',khoa AS 'Khoa'
```

FROM lop

cho biết mã lớp, tên lớp và khoá học của các lớp trong trường. Kết quả của câu lệnh như sau:

Mã lớp	Tên lớp	Khoa
C24101	Tián K24	24
C24102	Tin K24	24
C24103	Lý K24	24
C24301	Sinh K24	24
C25101	Tián K25	25
C25102	Tin K25	25
C25103	Lý K25	25
C25301	Sinh K25	25
C26101	Tián K26	26
C26102	Tin K26	26

- Sử dụng cấu trúc CASE trong danh sách chọn

Cấu trúc CASE được sử dụng trong danh sách chọn nhằm thay đổi kết quả của truy vấn tùy thuộc vào các trường hợp khác nhau. Cấu trúc này có cú pháp như sau:

`CASE biểu_thức`

`WHEN biểu_thức_kiểm_tra THEN kết_quá`

`[...]`

`[ELSE kết_quá_của_else]`

`END`

hoặc:

`CASE WHEN điều_kiện THEN kết_quá`

`[...]`

[ELSE kết quả của else]

END

Ví dụ: Để hiển thị mã, họ tên và giới tính (nam hoặc nữ) của các sinh viên, ta sử dụng câu lệnh

SELECT masv,hodem,ten,

CASE gioitinh

WHEN 1 THEN 'Nam'

ELSE 'Nữ'

END AS gioitinh

FROM sinhvien

hoặc:

SELECT masv,hodem,ten,

CASE

WHEN gioitinh=1 THEN 'Nam'

ELSE 'Nữ'

END AS gioitinh

FROM sinhvien

Kết quả của hai câu lệnh trên đều có dạng như sau

MASV	HỌ TÊN	TIN	GIOITINH
0241010001	Ngô Thị Nhât	Anh	Nữ
0241010002	Nguyễn Thị Ngọc	Anh	Nữ
0241010003	Ngô Việt	Bắc	Nam
0241010004	Nguyễn Đinh	Bình	Nam
0241010005	Hồ Đăng	Chiến	Nam
0241020001	Nguyễn Tuấn	Anh	Nam
0241020002	Trần Thị Kim	Anh	Nữ
0241020003	Võ Đức	Àn	Nam
0241020004	Nguyễn Công	Bình	Nam
0241020005	Nguyễn Thanh	Bình	Nam
0241020006	Lê Thị Thanh	Châu	Nữ
0241020007	Bùi Định	Chiến	Nam
0241020008	Nguyễn Công	Chính	Nam
***	***	***	***

- Hằng và biểu thức trong danh sách chọn

Ngoài danh sách trường, trong danh sách chọn của câu lệnh SELECT còn có thể sử dụng các biểu thức. Mỗi một biểu thức trong danh sách chọn trở thành một cột trong kết quả truy vấn.

Ví dụ: câu lệnh dưới đây cho biết tên và số tiết của các môn học

SELECT tenmonhoc,sodvht*15 AS sotiet

FROM monhoc

TENMONHOC	SOTIET
Hoá đại cương	45
Tin hoc dai cuong	60
Ngôn ngữ C	75
Lý thuyết hệ điều hành	60
Cấu trúc dữ liệu và ...	60
Đại số tuyến tính	60
Giải tích 1	60
Bài tập Đại số	30
Bài tập Giải tích 1	30
Vật lý đại cương	45

Nếu trong danh sách chọn có sự xuất hiện của giá trị hằng thì giá trị này sẽ xuất hiện trong một cột của kết quả truy vấn ở tất cả các dòng.

Ví dụ: Câu lệnh

SELECT tenmonhoc,'Số tiết: ',sodvht*15 AS sotiet

FROM monhoc

cho kết quả như sau:

TENMONHOC	(No column name)	SOTIET
Hoá đại cương	Số tiết:	45
Tin hoc dai cuong	Số tiết:	60
Ngôn ngữ C	Số tiết:	75
Lý thuyết hệ điều hành	Số tiết:	60
Cấu trúc dữ liệu và giải thuật	Số tiết:	60
Đại số tuyến tính	Số tiết:	60
Giải tích 1	Số tiết:	60
Bài tập Đại số	Số tiết:	30
Bài tập Giải tích 1	Số tiết:	30
Vật lý đại cương	Số tiết:	45

- Loại bỏ các dòng dữ liệu trùng nhau trong kết quả truy vấn

Trong kết quả của truy vấn có thể xuất hiện các dòng dữ liệu trùng nhau. Để loại bỏ bớt các dòng này, ta chỉ định thêm từ khóa DISTINCT ngay sau từ khóa SELECT.

Ví dụ: Hai câu lệnh dưới đây

SELECT khoa FROM lop

và:

SELECT DISTINCT khoa FROM lop

có kết quả lần lượt như sau:

KHOA
24
24
24
24
25
25
25
25
26
26

KHOA
24
25
26

- Giới hạn số lượng dòng trong kết quả truy vấn

Kết quả của truy vấn được hiển thị thường sẽ là tất cả các dòng dữ liệu truy vấn được. Trong trường hợp cần hạn chế số lượng các dòng xuất hiện trong kết quả truy vấn, ta chỉ định thêm mệnh đề TOP ngay trước danh sách chọn của câu lệnh SELECT.

Ví dụ: Câu lệnh dưới đây hiển thị họ tên và ngày sinh của 5 sinh viên đầu tiên trong danh sách

SELECT TOP 5 hodem,ten,ngaysinh

FROM sinhvien

Ngoài cách chỉ định cụ thể số lượng dòng cần hiển thị trong kết quả truy vấn, ta có thể chỉ định số lượng các dòng cần hiển thị theo tỷ lệ phần trăm bằng cách sử dụng thêm từ khoá PERCENT như ở ví dụ dưới đây.

Ví dụ: Câu lệnh dưới đây hiển thị họ tên và ngày sinh của 10% số lượng sinh viên hiện có trong bảng SINHVIEN

```
SELECT TOP 10 PERCENT hodem,ten,ngaysinh
```

```
FROM sinhvien
```

b. Chỉ định điều kiện truy vấn dữ liệu

Mệnh đề WHERE trong câu lệnh SELECT được sử dụng nhằm xác định các điều kiện đối với việc truy xuất dữ liệu. Sau mệnh đề WHERE là một biểu thức logic và chỉ những dòng dữ liệu nào thỏa mãn điều kiện được chỉ định mới được hiển thị trong kết quả truy vấn.

Ví dụ: Câu lệnh dưới đây hiển thị danh sách các môn học có số đơn vị học trình lớn hơn 3

```
SELECT * FROM monhoc
```

```
WHERE sodvht > 3
```

Kết quả của câu lệnh này như sau:

TIMONHOC	TENMONHOC	SODVHT
TI-001	Tin học đại cương	4
TI-002	Ngôn ngữ C	5
TI-003	Lý thuyết hệ điều hành	4
TI-004	Cấu trúc dữ liệu và giải thuật	4
TO-001	Đại số tuyến tính	4
TO-002	Giải tích 1	4

Trong mệnh đề WHERE thường sử dụng:

- Các toán tử kết hợp điều kiện (AND, OR)
- Các toán tử so sánh
- Kiểm tra giới hạn của dữ liệu (BETWEEN/ NOT BETWEEN)
- Danh sách
- Kiểm tra khuôn dạng dữ liệu.
- Các giá trị NULL

- Các toán tử so sánh

Toán tử	ý nghĩa
=	Bằng
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng
\neq	Khác
!>	Không lớn hơn
!<	Không nhỏ hơn

Ví dụ: Câu lệnh:

SELECT masv,hodem,tenten,ngaysinh

FROM sinhvien

WHERE (tenten='Anh')

AND (YEAR(GETDATE())-YEAR(ngaysinh)<=20)

Cho biết mã, họ tên và ngày sinh của sinh viên có tên là Anh và có tuổi nhỏ hơn hoặc bằng 20.

MASV	HỘ MÃI	TÊN	NGÀY SINH
0261010001	Lê Hoàng Phượng	Anh	1984-03-04 00:00:00
0261010002	Lê Thị Vân	Anh	1984-10-14 00:00:00
0261020002	Lê Thúc Quốc	Anh	1984-12-04 00:00:00
0261020004	Nguyễn Thị Lan	Anh	1984-08-23 00:00:00
0261020005	Nguyễn Thị Lan	Anh	1984-07-25 00:00:00

- Kiểm tra giới hạn của dữ liệu

Để kiểm tra xem giá trị dữ liệu nằm trong (ngoài) một khoảng nào đó, ta sử dụng toán tử BETWEEN (NOT BETWEEN) như sau:

Cách sử dụng	Ý nghĩa
giá_trị BETWEEN a AND b	$a \leq \text{giá_trị} \leq b$
giá_trị NOT BETWEEN a AND b	(giá_trị < a) AND (giá_trị > b)

Ví dụ: Câu lệnh dưới đây cho biết họ tên và tuổi của các sinh viên có tên là Bình và có tuổi nằm trong khoảng từ 20 đến 22

```
SELECT hodem,ten,year(getdate())-year(ngaysinh) AS tuoi  
FROM sinhvien
```

WHERE ten='Bình' AND

YEAR(GETDATE())-YEAR(ngaysinh) BETWEEN 20 AND 22

- Danh sách (IN và NOT IN)

Từ khoá IN được sử dụng khi ta cần chỉ định điều kiện tìm kiếm dữ liệu cho câu lệnh SELECT là một danh sách các giá trị. Sau IN (hoặc NOT IN) có thể là một danh sách các giá trị hoặc là một câu lệnh SELECT khác.

Ví dụ: Để biết danh sách các môn học có số đơn vị học trình là 2, 4 hoặc 5, thay vì sử dụng câu lệnh

```
SELECT * FROM monhoc
```

```
WHERE sodvht=2 OR sodvht=4 OR sodvht=5
```

ta có thể sử dụng câu lệnh

```
SELECT * FROM monhoc
```

```
WHERE sodvht IN (2,4,5)
```

- Toán tử LIKE và các ký tự đại diện

Từ khoá LIKE (NOT LIKE) sử dụng trong câu lệnh SELECT nhằm mô tả khuôn dạng của dữ liệu cần tìm kiếm. Chúng thường được kết hợp với các ký tự đại diện sau đây:

Ký tự đại diện	ý nghĩa
%	Chuỗi ký tự bất kỳ gồm không hoặc nhiều ký tự
_	Ký tự đơn bất kỳ
[]	Ký tự đơn bất kỳ trong giới hạn được chỉ định (ví dụ [a-f]) hay một tập (ví dụ [abcdef])
[^]	Ký tự đơn bất kỳ không nằm trong giới hạn được chỉ định (ví dụ [^a-f] hay một tập (ví dụ [^abcdef]))

Ví dụ: Câu lệnh dưới đây

SELECT hodem,ten FROM sinhvien

WHERE hodem LIKE 'Lê%'

cho biết họ tên của các sinh viên có họ là Lê và có kết quả như sau:

HODEM	TEN
Lê Thị Thanh	Châu
Lê Thị	Ánh
Lê Văn Khoa	Bảo
Lê Thị	Đi
Lê Tát Uyên	Châu
Lê Hoàng Phương	Ánh
Lê Thị Vân	Ánh
Lê Đăng	Ánh
Lê Huy	Đan
Lê Thúc Quốc	Ánh

Câu lệnh:

SELECT hodem,ten FROM sinhvien

WHERE hodem LIKE 'Lê%' AND ten LIKE '[AB]%'

Có kết quả là:

HODEM	TEN
Lê Thị	Ánh
Lê Văn Khoa	Bảo
Lê Hoàng Phương	Ánh
Lê Thị Vân	Ánh
Lê Thúc Quốc	Ánh

- Giá trị NULL

Dữ liệu trong một cột cho phép NULL sẽ nhận giá trị NULL trong các trường hợp sau:

- Nếu không có dữ liệu được nhập cho cột và không có mặc định cho cột hay kiểu dữ liệu trên cột đó.
- Người sử dụng trực tiếp đưa giá trị NULL vào cho cột đó.
- Một cột có kiểu dữ liệu là kiểu số sẽ chứa giá trị NULL nếu giá trị được chỉ định gây tràn số.

Trong mệnh đề WHERE, để kiểm tra giá trị của một cột có giá trị NULL hay không, ta sử dụng cách viết:

WHERE tên_cột IS NULL

hoặc:

WHERE tên_cột IS NOT NULL

c. *Tạo mới bảng dữ liệu từ kết quả của câu lệnh SELECT*

Câu lệnh SELECT ... INTO có tác dụng tạo một bảng mới có cấu trúc và dữ liệu được xác định từ kết quả của truy vấn. Bảng mới được tạo ra sẽ có số cột bằng số cột được chỉ định trong danh sách chọn và số dòng sẽ là số dòng kết quả của truy vấn.

Ví dụ: Câu lệnh dưới đây truy vấn dữ liệu từ bảng SINHVIEN và tạo một bảng TUOISV bao gồm các trường HODEM, TEN và TUOI

```
SELECT hodem,ten, YEAR(GETDATE())-YEAR(ngaysinh) AS tuoi  
INTO tuoisv  
FROM sinhvien
```

Lưu ý: Nếu trong danh sách chọn có các biểu thức thì những biểu thức này phải được đặt tiêu đề.

d. *Sắp xếp kết quả truy vấn*

Mặc định, các dòng dữ liệu trong kết quả của câu truy vấn tuân theo thứ tự của chúng trong bảng dữ liệu hoặc được sắp xếp theo chi mục (nếu trên bảng có chi mục). Trong trường hợp muốn dữ liệu được sắp xếp theo chiều tăng hoặc giảm của giá trị của một hoặc nhiều trường, ta sử dụng thêm mệnh đề ORDER BY trong câu lệnh SELECT; Sau ORDER BY là danh sách các cột cần sắp xếp (tối đa là 16 cột). Dữ liệu được sắp xếp có thể theo chiều tăng (ASC) hoặc giảm (DESC), mặc định là sắp xếp theo chiều tăng.

Ví dụ: Câu lệnh dưới đây hiển thị danh sách các môn học và sắp xếp theo chiều giảm dần của số đơn vị học trình

```
SELECT * FROM monhoc  
ORDER BY sodvht DESC
```

MÃ KHÓA HỌC	TÊN KHÓA HỌC	SỐ ĐIỂM
TI-002	Ngôn ngữ C	5
TI-003	Lý thuyết hệ điều hành	4
TI-001	Tin học đại cương	4
TI-004	Cấu trúc dữ liệu và giải thuật	4
TO-001	Đại số tuyến tính	4
TO-002	Giải tích 1	4
HO-001	Hoá đại cương	3
VL-001	Vật lý đại cương	3
TO-004	Bài tập Giải tích 1	2
TO-003	Bài tập Đại số	2

Nếu sau ORDER BY có nhiều cột thì việc sắp xếp dữ liệu sẽ được ưu tiên theo thứ tự từ trái qua phải.

Ví dụ: Câu lệnh

```
SELECT hodem,ten,gioitinh,
YEAR(GETDATE())-YEAR(ngaysinh) AS tuoi
FROM sinhvien
WHERE ten='Binh'
ORDER BY gioitinh,tuoi
```

có kết quả là:

HỘ MÃ	TÊN	GIOITINH	tuoi
Nguyễn Thị	Binh 0	23	
Hoàng Văn	Binh 1	21	
Chau Văn Quốc	Binh 1	21	
Nguyễn Thành	Binh 1	22	
Nguyễn Định	Binh 1	22	
Nguyễn Công	Binh 1	25	

Thay vì chỉ định tên cột sau ORDER BY, ta có thể chỉ định số thứ tự của cột cần được sắp xếp. Câu lệnh ở ví dụ trên có thể được viết lại như sau:

```
SELECT hodem,ten,gioitinh,
YEAR(GETDATE())-YEAR(ngaysinh) AS tuoi
FROM sinhvien
WHERE ten='Binh'
ORDER BY 3, 4
```

e. Phép hợp

Phép hợp được sử dụng trong trường hợp ta cần gộp kết quả của hai hay nhiều truy vấn thành một tập kết quả duy nhất. SQL cung cấp toán tử UNION để thực hiện phép hợp. Cú pháp như sau

Câu_lệnh_1

UNION [ALL] Câu_lệnh_2

[UNION [ALL] Câu_lệnh_3]

...
[UNION [ALL] Câu_lệnh_n]

[ORDER BY cột_sắp_xếp]

[COMPUTE danh_sách_hàm_gộp [BY danh_sách_cột]]

Trong đó

Câu_lệnh_1 có dạng

SELECT danh_sách_cột

[INTO tên_bảng_mới]

[FROM danh_sách_bảng|khung_nhìn]

[WHERE điều_kiện]

[GROUP BY danh_sách_cột]

[HAVING điều_kiện]

và Câu_lệnh_i (i = 2,...,n) có dạng

SELECT danh_sách_cột

[FROM danh_sách_bảng|khung_nhìn]

[WHERE điều_kiện]

[GROUP BY danh_sách_cột]

[HAVING điều_kiện]

Ví dụ: Giả sử ta có hai bảng Table1 và Table2 lần lượt như sau:

câu lệnh

SELECT A,B FROM Table1

UNION

SELECT D,E FROM table2

Cho kết quả như sau:

A	B	C
a	1	10
b	2	20
c	3	30
d	4	40
a	5	50
b	6	60

D	E
a	1
b	2
d	3
e	4

Mặc định, nếu trong các truy vấn thành phần của phép hợp xuất hiện những dòng dữ liệu giống nhau thì trong kết quả truy vấn chỉ giữ lại một dòng. Nếu muốn giữ lại các dòng này, ta phải sử dụng thêm từ khoá ALL trong truy vấn thành phần.

Ví dụ: Câu lệnh

SELECT A,B FROM Table1

UNION ALL

SELECT D,E FROM table2

Cho kết quả như sau

	B	C
a	1	
a	5	
b	2	
b	6	
c	3	
d	3	
d	4	
e	4	

Khi sử dụng toán tử UNION để thực hiện phép hợp, ta cần chú ý các nguyên tắc sau:

- Danh sách cột trong các truy vấn thành phần phải có cùng số lượng.
- Các cột tương ứng trong tất cả các bảng, hoặc tập con bất kỳ các cột được sử dụng trong bản thân mỗi truy vấn thành phần phải cùng kiểu dữ liệu.

- Các cột tương ứng trong bản thân từng truy vấn thành phần của một câu lệnh UNION phải xuất hiện theo thứ tự như nhau. Nguyên nhân là do phép hợp so sánh các cột từng cột một theo thứ tự được cho trong mỗi truy vấn.
- Khi các kiểu dữ liệu khác nhau được kết hợp với nhau trong câu lệnh UNION, chúng sẽ được chuyển sang kiểu dữ liệu cao hơn (nếu có thể).
- Tiêu đề cột trong kết quả của phép hợp sẽ là tiêu đề cột được chỉ định trong truy vấn đầu tiên.
- Truy vấn thành phần đầu tiên có thể có INTO để tạo mới một bảng từ kết quả của chính phép hợp.
- Mệnh đề ORDER BY và COMPUTE dùng để sắp xếp kết quả truy vấn hoặc tính toán các giá trị thống kê chỉ được sử dụng ở cuối câu lệnh UNION. Chúng không được sử dụng ở trong bất kỳ truy vấn thành phần nào.
- Mệnh đề GROUP BY và HAVING chỉ có thể được sử dụng trong bản thân từng truy vấn thành phần. Chúng không được phép sử dụng để tác động lên kết quả chung của phép hợp.
- Phép toán UNION có thể được sử dụng bên trong câu lệnh INSERT.
- Phép toán UNION không được sử dụng trong câu lệnh CREATE VIEW.

f. Phép nối

Khi cần thực hiện một yêu cầu truy vấn dữ liệu từ hai hay nhiều bảng, ta phải sử dụng đến phép nối. Một câu lệnh nối kết hợp các dòng dữ liệu trong các bảng khác nhau lại theo một hoặc nhiều điều kiện nào đó và hiển thị chúng trong kết quả truy vấn.

Xét hai bảng sau đây:

Bảng KHOA

MÃ KHOA	TÊN KHOA	ĐIỆN THOẠI
DHT01	Khoa Toán cơ - Tin học	054822407
DHT02	Khoa Công nghệ thông tin	054826767
DHT03	Khoa Vật lý	054823462
DHT04	Khoa Hoá học	054823951
DHT05	Khoa Sinh học	054822934
DHT06	Khoa Địa lý - Địa chất	054823837

Bảng LOP

MALOP	TENLOP	KHOA	HỌC KÌ/HỌC	HỌC KÌ/HỌC	MALO
C24101	Toán K24	24	Chính quy	2000	DHT01
C24102	Tin K24	24	Chính quy	2000	DHT02
C24103	Lý K24	24	Chính quy	2000	DHT03
C24301	Sinh K24	24	Chính quy	2000	DHT05
C25101	Toán K25	25	Chính quy	2001	DHT01
C25102	Tin K25	25	Chính quy	2001	DHT02
C25103	Lý K25	25	Chính quy	2001	DHT03
C25301	Sinh K25	25	Chính quy	2001	DHT05
C26101	Toán K26	26	Chính quy	2002	DHT01
C26102	Tin K26	26	Chính quy	2002	DHT02

Giả sử ta cần biết mã lớp và tên lớp của các lớp thuộc Khoa Công nghệ Thông tin, ta phải làm như sau:

- Chọn ra dòng trong bảng KHOA có tên khoa là Khoa Công nghệ Thông tin, từ đó xác định được mã khoa (MAKHOA) là DHT02.
- Tìm kiếm trong bảng LOP những dòng có giá trị trường MAKHOA là DHT02 (tức là bảng MAKHOA tương ứng trong bảng KHOA) và đưa những dòng này vào kết quả truy vấn

MALO	TENKHOA	KHOA	MAKHOA
DHT01	Khoa Toán cơ - Tin học	054822407	
DHT02	Khoa Công nghệ thông tin	051926767	
DHT03	Khoa Vật lý	051923462	
DHT04	Khoa Hóa học	054823051	
DHT05	Khoa Điện lực	051922934	
DHT06	Khoa Địa lý - Địa chất	051922627	

MALO	TENKHOA	KHOA	HỌC KÌ/HỌC	MALOP
C24101	Toán K24	24	Chính quy	2000
C24102	Tin K24	24	Chính quy	2000
C24103	Lý K24	24	Chính quy	2000
C24301	Sinh K24	24	Chính quy	2000
C25101	Toán K25	25	Chính quy	2001
C25102	Tin K25	25	Chính quy	2001
C25103	Lý K25	25	Chính quy	2001
C25301	Sinh K25	25	Chính quy	2001
C26101	Toán K26	26	Chính quy	2002
C26102	Tin K26	26	Chính quy	2002

MALO	TENKHOA
C24103	Tin K24
C25102	Tin K25
C26102	Tin K26

Như vậy, để thực hiện được yêu cầu truy vấn dữ liệu trên, ta phải thực hiện phép nối giữa hai bảng KHOA và LOP với điều kiện nối là MAKHOA của KHOA bằng với MAKHOA của LOP. Câu lệnh sẽ được viết như sau:

SELECT malop,tenlop

FROM khoa,lop

WHERE khoa.makhoa = lop.makhoa AND

tenkhoa='Khoa Công nghệ Thông tin'

Sử dụng phép nối

Phép nối là cơ sở để thực hiện các yêu cầu truy vấn dữ liệu liên quan đến nhiều bảng. Một câu lệnh nối thực hiện lấy các dòng dữ liệu trong các bảng tham gia truy vấn, so sánh giá trị của các dòng này trên một hoặc nhiều cột được chỉ định trong điều kiện nối và kết hợp các dòng thỏa mãn điều kiện thành những dòng trong kết quả truy vấn.

Để thực hiện được một phép nối, cần phải xác định được những yếu tố sau:

- Những cột nào cần hiển thị trong kết quả truy vấn
- Những bảng nào có tham gia vào truy vấn.
- Điều kiện để thực hiện phép nối giữa các bảng dữ liệu là gì

Trong các yếu tố kể trên, việc xác định chính xác điều kiện để thực hiện phép nối giữa các bảng đóng vai trò quan trọng nhất. Trong đa số các trường hợp, điều kiện của phép nối được xác định nhờ vào mối quan hệ giữa các bảng cần phải truy xuất dữ liệu. Thông thường, đó là điều kiện bằng nhau giữa khoá chính và khoá ngoài của hai bảng có mối quan hệ với nhau. Như vậy, để có thể đưa ra một câu lệnh nối thực hiện chính xác yêu cầu truy vấn dữ liệu đòi hỏi phải hiểu được mối quan hệ cũng như ý nghĩa của chúng giữa các bảng dữ liệu.

Danh sách chọn trong phép nối

Một câu lệnh nối cũng được bắt đầu với từ khóa SELECT. Các cột được chỉ định tên sau từ khóa SELECT là các cột được hiển thị trong kết quả truy vấn. Việc sử dụng tên các cột trong danh sách chọn có thể là:

- Tên của một số cột nào đó trong các bảng có tham gia vào truy vấn. Nếu tên cột trong các bảng trùng tên nhau thì tên cột phải được viết dưới dạng tên_bảng.tên_cột
- Dấu sao (*) được sử dụng trong danh sách chọn khi cần hiển thị tất cả các cột của các bảng tham gia truy vấn.
- Trong trường hợp cần hiển thị tất cả các cột của một bảng nào đó, ta sử dụng cách viết: tên_bảng.*

Mệnh đề FROM trong phép nối

Sau mệnh đề FROM của câu lệnh nối là danh sách tên các bảng (hay khung nhìn) tham gia vào truy vấn. Nếu ta sử dụng dấu * trong danh sách chọn thì thứ tự của các bảng liệt kê sau FROM sẽ ảnh hưởng đến thứ tự các cột được hiển thị trong kết quả truy vấn.

Mệnh đề WHERE trong phép nối

Khi hai hay nhiều bảng được nối với nhau, ta phải chỉ định điều kiện để thực hiện phép nối ngay sau mệnh đề WHERE. Điều kiện nối được biểu diễn dưới dạng biểu thức logic so sánh giá trị dữ liệu giữa các cột của các bảng tham gia truy vấn.

Các toán tử so sánh dưới đây được sử dụng để xác định điều kiện nối

Phép toán Ý nghĩa

= Bằng

> Lớn hơn

>= Lớn hơn hoặc bằng

< Nhỏ hơn

<= Nhỏ hơn hoặc bằng

<> Khác

!= Không lớn hơn

!< Không nhỏ hơn

Ví dụ: Câu lệnh dưới đây hiển thị danh sách các sinh viên với các thông tin: mã sinh viên, họ và tên, mã lớp, tên lớp và tên khoa

SELECT masv,hodem,ten,sinhvien.malop,tenlop,tenkhoa

FROM sinhvien,lop,khoa

WHERE sinhvien.malop = lop.malop AND

lop.makhoa=khoa.makhoa

Trong câu lệnh trên, các bảng tham gia vào truy vấn bao gồm SINHVIEN, LOP và KHOA. Điều kiện để thực hiện phép nối giữa các bảng bao gồm hai điều kiện:

sinhvien.malop = lop.malop

và lop.malop = khoa.malop

Điều kiện nối giữa các bảng trong câu lệnh trên là điều kiện bằng giữa khoá ngoài và khoá chính của các bảng có mối quan hệ với nhau. Hay nói cách khác, điều kiện của phép nối được xác định dựa vào mối quan hệ giữa các bảng trong cơ sở dữ liệu.

Các loại phép nối

Phép nối bằng và phép nối tự nhiên

Một phép nối bằng (equi-join) là một phép nối trong đó giá trị của các cột được sử dụng để nối được so sánh với nhau dựa trên tiêu chuẩn bằng và tất cả các cột trong các bảng tham gia nối đều được đưa ra trong kết quả.

Ví dụ: Câu lệnh dưới đây thực hiện phép nối bằng giữa hai bảng LOP và KHOA

```
SELECT *
```

```
FROM lop,khoa
```

```
WHERE lop.makhoa=khoa.makhoa
```

Trong kết quả của câu lệnh trên, cột makhoa (mã khoa) xuất hiện hai lần trong kết quả phép nối (cột makhoa của bảng khoa và cột makhoa của bảng lop) và như vậy là không cần thiết. Ta có thể loại bỏ bớt đi những cột trùng tên trong kết quả truy vấn bằng cách chỉ định danh sách cột cần được hiển thị trong danh sách chọn của câu lệnh.

Một dạng đặc biệt của phép nối bằng được sử dụng nhiều là phép nối tự nhiên (natural-join). Trong phép nối tự nhiên, điều kiện nối giữa hai bảng chính là điều kiện bằng giữa khoá ngoài và khoá chính của hai bảng; Và trong danh sách chọn của câu lệnh chỉ giữ lại một cột trong hai cột tham gia vào điều kiện của phép nối

Để thực hiện phép nối tự nhiên, câu lệnh trong ví dụ trên được viết lại như sau

```
SELECT malop,tenlop,khoa,hedaotao,namnhaphoc,
```

```
siso,lop.makhoa,tenkhoa,dienthoai
```

```
FROM lop,khoa
```

```
WHERE lop.makhoa=khoa.makhoa
```

hoặc viết dưới dạng ngắn gọn hơn:

```
SELECT lop.*,tenkhoa,dienthoai  
FROM lop,khoa  
WHERE lop.makhoa=khoa.makhoa
```

Phép nối với các điều kiện bổ sung

Trong các câu lệnh nối, ngoài điều kiện của phép nối được chỉ định trong mệnh đề WHERE còn có thể chỉ định các điều kiện tìm kiếm dữ liệu khác (điều kiện chọn). Thông thường, các điều kiện này được kết hợp với điều kiện nối thông qua toán tử AND.

Ví dụ: Câu lệnh dưới đây hiển thị họ tên và ngày sinh của các sinh viên Khoa Công nghệ Thông tin

```
SELECT hodem,ten,ngaysinh  
FROM sinhvien,lop,khoa  
WHERE tenkhoa='Khoa Công nghệ Thông tin' AND  
sinhvien.malop = lop.malop AND  
lop.makhoa = khoa.makhoa
```

Phép tự nối và các bí danh

Phép tự nối là phép nối mà trong đó điều kiện nối được chỉ định liên quan đến các cột của cùng một bảng. Trong trường hợp này, sẽ có sự xuất hiện tên của cùng một bảng nhiều lần trong mệnh đề FROM và do đó các bảng cần phải được đặt bí danh.

Ví dụ: Để biết được họ tên và ngày sinh của các sinh viên có cùng ngày sinh với sinh viên Trần Thị Kim Anh, ta phải thực hiện phép tự nối ngay trên chính bảng sinhvien. Trong câu lệnh nối, bảng sinhvien xuất hiện trong mệnh đề FROM với bí danh là a và b. Bảng sinhvien với bí danh là a sử dụng để chọn ra sinh viên có họ tên là Trần Thị Kim Anh và bảng sinhvien với bí danh là b sử dụng để xác định các sinh viên trùng ngày sinh với sinh viên Trần Thị Kim Anh. Câu lệnh được viết như sau:

```
SELECT b.hodem,b.ten,b.ngaysinh  
FROM sinhvien a, sinhvien b  
WHERE a.hodem='Trần Thị Kim' AND a.ten='Anh' AND  
a.ngaysinh=b.ngaysinh AND a.masv<>b.masv
```

Phép nối không dựa trên tiêu chuẩn bằng

Trong phép nối này, điều kiện để thực hiện phép nối giữa các bảng dữ liệu không phải là điều kiện so sánh bằng giữa các cột. Loại phép nối này trong thực tế thường ít được sử dụng.

Phép nối ngoài (outer-join)

Trong các phép nối đã đề cập ở trên, chỉ những dòng có giá trị trong các cột được chỉ định thoả mãn điều kiện kết nối mới được hiển thị trong kết quả truy vấn, và được gọi là phép nối trong (inner join). Theo một nghĩa nào đó, những phép nối này loại bỏ thông tin chứa trong những dòng không thoả mãn điều kiện nối. Tuy nhiên, đôi khi ta cũng cần giữ lại những thông tin này bằng cách cho phép những dòng không thoả mãn điều kiện nối có mặt trong kết quả của phép nối. Để làm điều này, ta có thể sử dụng phép nối ngoài.

SQL cung cấp các loại phép nối ngoài sau đây:

- **Phép nối ngoài trái** (ký hiệu: *=): Phép nối này hiển thị trong kết quả truy vấn tất cả các dòng dữ liệu của bảng nằm bên trái trong điều kiện nối cho dù những dòng này không thoả mãn điều kiện của phép nối.
- **Phép nối ngoài phải** (ký hiệu: =*): Phép nối này hiển thị trong kết quả truy vấn tất cả các dòng dữ liệu của bảng nằm bên phải trong điều kiện nối cho dù những dòng này không thoả mãn điều kiện của phép nối.

Ví dụ: Giả sử ta có hai bảng DONVI và NHANVIEN như sau:

Bảng DONVI Bảng NHANVIEN

MADV	TENDV
1	Đại ngõe1
2	Hành chính
3	Kế toán
4	Kinh doanh

HOTEN	MADV
Thanh	1
Hoa	2
Nam	2
Vinh	1
Hung	5
Phuong	NULL

Câu lệnh:

SELECT *

FROM nhanvien,donvi

WHERE nhanvien.madv=donvi.madv

có kết quả là:

HOTEN	MADV	MADV	TENDV
Thanh	1	1	Doi ngoai
Hoa	2	2	Hanh chinh
Nam	2	2	Hanh chinh
Vinh	1	1	Doi ngoai

Nếu thực hiện phép nối ngoài trái giữa bảng NHANVIEN và bảng DONVI:

SELECT *

FROM nhanvien,donvi

WHERE nhanvien.madv*=donvi.madv

kết quả của câu lệnh sẽ là:

HOTEN	MADV	MADV	TENDV
Thanh	1	1	Doi ngoai
Hoa	2	2	Hanh chinh
Nam	2	2	Hanh chinh
Vinh	1	1	Doi ngoai
Hung	5	NULL	NULL
Phuong	NULL	NULL	NULL

Và kết quả của phép nối ngoài phải:

select *

from nhanvien,donvi

where nhanvien.madv=*donvi.madv

như sau:

HOTEN	MADV	MADV	TENDV
Thanh	1	1	Doi ngoai
Vinh	1	1	Doi ngoai
Hoa	2	2	Hanh chinh
Nam	2	2	Hanh chinh
NULL	NULL	3	Kinh doanh
NULL	NULL	4	Kinh doanh

Phép nối và các giá trị NULL

Nếu trong các cột của các bảng tham gia vào điều kiện của phép nối có các giá trị NULL thì các giá trị NULL được xem như là không bằng nhau.

Ví dụ: Giả sử ta có hai bảng TABLE1 và TABLE2 như sau:

TABLE1	
A	B
1	b1
NULL	b2
4	b3

TABLE2	
C	D
NULL	d1
4	d2

Câu lệnh:

```
SELECT *
FROM table1, table2
WHERE A *= C
```

Có kết quả là:

A	B	C	D
1	b1	NULL	NULL
NULL	b2	NULL	NULL
4	b3	4	d2

g. Thống kê dữ liệu với GROUP BY

Ngoài khả năng thực hiện các yêu cầu truy vấn dữ liệu thông thường (chiếu, chọn, nối,...) như đã đề cập như ở các phần trước, câu lệnh SELECT còn cho phép thực hiện các thao tác truy vấn và tính toán thống kê trên dữ liệu như: cho biết tổng số tiết dạy của mỗi giáo viên, điểm trung bình các môn học của mỗi sinh viên,...

Mệnh đề GROUP BY sử dụng trong câu lệnh SELECT nhằm phân hoạch các dòng dữ liệu trong bảng thành các nhóm dữ liệu, và trên mỗi nhóm dữ liệu thực hiện tính toán các giá trị thống kê như tính tổng, tính giá trị trung bình,...

Các hàm gộp được sử dụng để tính giá trị thống kê cho toàn bảng hoặc trên mỗi nhóm dữ liệu. Chúng có thể được sử dụng như là các cột trong danh sách chọn của câu lệnh SELECT hoặc xuất hiện trong mệnh đề HAVING, nhưng không được phép xuất hiện trong mệnh đề WHERE.

SQL cung cấp các hàm gộp dưới đây:

Hàm gộp	Chức năng
SUM([ALL DISTINCT] biểu_thức)	Tính tổng các giá trị.
AVG([ALL DISTINCT] biểu_thức)	Tính trung bình của các giá trị
COUNT([ALL DISTINCT] biểu_thức)	Đếm số các giá trị trong biểu thức.
COUNT(*)	Đếm số các dòng được chọn.
MAX(biểu_thức)	Tính giá trị lớn nhất
MIN(biểu_thức)	Tính giá trị nhỏ nhất

Trong đó:

- Hàm SUM và AVG chỉ làm việc với các biểu thức số.
- Hàm SUM, AVG, COUNT, MIN và MAX bỏ qua các giá trị NULL khi tính toán.
- Hàm COUNT(*) không bỏ qua các giá trị NULL.

Mặc định, các hàm gộp thực hiện tính toán thống kê trên toàn bộ dữ liệu. Trong trường hợp cần loại bỏ bớt các giá trị trùng nhau (chỉ giữ lại một giá trị), ta chỉ định thêm từ khoá DISTINCT ở trước biểu thức là đối số của hàm.

Thống kê trên toàn bộ dữ liệu

Khi cần tính toán giá trị thống kê trên toàn bộ dữ liệu, ta sử dụng các hàm gộp trong danh sách chọn của câu lệnh SELECT. Trong trường hợp này, trong danh sách chọn không được sử dụng bất kỳ một tên cột hay biểu thức nào ngoài các hàm gộp.

Ví dụ: Để thống kê trung bình điểm lần 1 của tất cả các môn học, ta sử dụng câu lệnh như sau:

`SELECT AVG(diemlan1)`

`FROM diemthi`

còn câu lệnh dưới đây cho biết tuổi lớn nhất, tuổi nhỏ nhất và độ tuổi trung bình của tất cả các sinh viên sinh tại Huế:

`SELECT MAX(YEAR(GETDATE())-YEAR(ngaysinh)),`

`MIN(YEAR(GETDATE())-YEAR(ngaysinh)),`

`AVG(YEAR(GETDATE())-YEAR(ngaysinh))`

FROM sinhvien

WHERE noisinh='Hué'

Thống kê dữ liệu trên các nhóm

Trong trường hợp cần thực hiện tính toán các giá trị thống kê trên các nhóm dữ liệu, ta sử dụng mệnh đề GROUP BY để phân hoạch dữ liệu vào trong các nhóm. Các hàm gộp được sử dụng sẽ thực hiện thao tác tính toán trên mỗi nhóm và cho biết giá trị thống kê theo các nhóm dữ liệu.

Ví dụ 2.36: Câu lệnh dưới đây cho biết số lượng sinh viên của mỗi lớp

SELECT lop.malop,tenlop,COUNT(masv) AS siso

FROM lop,sinhvien

WHERE lop.malop=sinhvien.malop

GROUP BY lop.malop,tenlop

và có kết quả là

MALOP	TENLOP	SISO
C24101	Toán K24	5
C24102	Tin K24	6
C24103	Lý K24	7
C24301	Sinh K24	5
C25101	Toán K25	5
C25102	Tin K25	6
C25103	Lý K25	6
C25301	Sinh K25	8
C26101	Toán K26	5
C26102	Tin K26	5

còn câu lệnh:

SELECT sinhvien.masv,hodem,ten,

sum(diemlan1*sodvht)/sum(sodvht)

FROM sinhvien,diemthi,monhoc

WHERE sinhvien.masv=diemthi.masv AND

diemthi.mamonhoc=monhoc.mamonhoc

GROUP BY sinhvien.masv,hodem,ten

cho biết trung bình điểm thi lần 1 các môn học của các sinh viên

Lưu ý: Trong trường hợp danh sách chọn của câu lệnh SELECT có cả các hàm gộp và những biểu thức không phải là hàm gộp thì những biểu thức này phải có mặt đầy đủ trong mệnh đề GROUP BY, nếu không câu lệnh sẽ không hợp lệ.

Ví dụ: Dưới đây là một câu lệnh sai

```
SELECT lop.malop,tenlop,COUNT(masv)
```

```
FROM lop,sinhvien
```

```
WHERE lop.malop=sinhvien.malop
```

```
GROUP BY lop.malop
```

do thiếu trường TENLOP sau mệnh đề GROUP BY.

Chỉ định điều kiện đối với hàm gộp

Mệnh đề HAVING được sử dụng nhằm chỉ định điều kiện đối với các giá trị thống kê được sản sinh từ các hàm gộp tương tự như cách thức mệnh đề WHERE thiết lập các điều kiện cho câu lệnh SELECT. Mệnh đề HAVING thường không thực sự có nghĩa nếu như không sử dụng kết hợp với mệnh đề GROUP BY. Một điểm khác biệt giữa HAVING và WHERE là trong điều kiện của WHERE không được có các hàm gộp trong khi HAVING lại cho phép sử dụng các hàm gộp trong điều kiện của mình.

Ví dụ: Để biết trung bình điểm thi lần 1 của các sinh viên có điểm trung bình lớn hơn hoặc bằng 5, ta sử dụng câu lệnh như sau:

```
SELECT sinhvien.masv,hodem,ten,
```

```
SUM(diemlan1*sodvht)/sum(sodvht)
```

```
FROM sinhvien,diemthi,monhoc
```

```
WHERE sinhvien.masv=diemthi.masv AND
```

```
diemthi.mamonhoc=monhoc.mamonhoc
```

```
GROUP BY sinhvien.masv,hodem,ten
```

```
HAVING sum(diemlan1*sodvht)/sum(sodvht)>=5
```

h. Truy vấn con (Subquery)

Truy vấn con là một câu lệnh SELECT được lồng vào bên trong một câu lệnh SELECT, INSERT, UPDATE, DELETE hoặc bên trong một truy vấn con khác. Loại truy vấn này được sử dụng để biểu diễn cho những truy vấn

trong đó điều kiện truy vấn dữ liệu cần phải sử dụng đến kết quả của một truy vấn khác.

Cú pháp của truy vấn con như sau:

```
(SELECT [ALL | DISTINCT] danh_sách_chọn  
FROM danh_sách_bảng  
[WHERE điều_kiện]  
[GROUP BY danh_sách_cột]  
[HAVING điều_kiện])
```

Khi sử dụng truy vấn con cần lưu ý một số quy tắc sau:

- Một truy vấn con phải được viết trong cặp dấu ngoặc. Trong hầu hết các trường hợp, một truy vấn con thường phải có kết quả là một cột (tức là chỉ có duy nhất một cột trong danh sách chọn).
- Mệnh đề COMPUTE và ORDER BY không được phép sử dụng trong truy vấn con.
- Các tên cột xuất hiện trong truy vấn con có thể là các cột của các bảng trong truy vấn ngoài.
- Một truy vấn con thường được sử dụng làm điều kiện trong mệnh đề WHERE hoặc HAVING của một truy vấn khác.
- Nếu truy vấn con trả về đúng một giá trị, nó có thể sử dụng như là một thành phần bên trong một biểu thức (chẳng hạn xuất hiện trong một phép so sánh bằng)

Phép so sánh đối với kết quả truy vấn con

Kết quả của truy vấn con có thể được sử dụng để thực hiện phép so sánh số học với một biểu thức của truy vấn cha. Trong trường hợp này, truy vấn con được sử dụng dưới dạng:

```
WHERE biểu_thức phép_toán_số_học [ANY|ALL]  
(truy_vấn_con)
```

Trong đó phép toán số học có thể sử dụng bao gồm: =, <, >, <=, >=; Và truy vấn con phải có kết quả bao gồm đúng một cột.

Ví dụ: Câu lệnh dưới đây cho biết danh sách các môn học có số đơn vị học trình lớn hơn hoặc bằng số đơn vị học trình của môn học có mã là TI-001

```
SELECT *
FROM monhoc
WHERE sodvht>=(SELECT sodvht
FROM monhoc
WHERE mamonhoc='TI-001')
```

Nếu truy vấn con trả về nhiều hơn một giá trị, việc sử dụng phép so sánh như trên sẽ không hợp lệ. Trong trường hợp này, sau phép toán so sánh phải sử dụng thêm lượng từ ALL hoặc ANY. Lượng từ ALL được sử dụng khi cần so sánh giá trị của biểu thức với tất cả các giá trị trả về trong kết quả của truy vấn con; ngược lại, phép so sánh với lượng từ ANY có kết quả đúng khi chỉ cần một giá trị bất kỳ nào đó trong kết quả của truy vấn con thỏa mãn điều kiện.

Ví dụ: Câu lệnh dưới đây cho biết họ tên của những sinh viên lớp Tin K25 sinh trước tất cả các sinh viên của lớp Toán K25

```
SELECT hodem,ten
FROM sinhvien JOIN lop ON sinhvien.malop=lop.malop
WHERE tenlop='Tin K25' AND
ngaysinh<ALL(SELECT ngaysinh
FROM sinhvien JOIN lop
ON sinhvien.malop=lop.malop
WHERE lop.tenlop='Toán K25')
```

và câu lệnh:

```
SELECT hodem,ten
FROM sinhvien JOIN lop on sinhvien.malop=lop.malop
WHERE tenlop='Tin K25' AND
year(ngaysinh)= ANY(SELECT year(ngaysinh)
FROM sinhvien JOIN lop
ON sinhvien.malop=lop.malop
WHERE lop.tenlop='Toán K25')
```

cho biết họ tên của những sinh viên lớp Tin K25 có năm sinh trùng với năm sinh của bất kỳ một sinh viên nào đó của lớp Toán K25.

Sử dụng truy vấn con với toán tử IN

Khi cần thực hiện phép kiểm tra giá trị của một biểu thức có xuất hiện (không xuất hiện) trong tập các giá trị của truy vấn con hay không, ta có thể sử dụng toán tử IN (NOT IN) như sau:

WHERE biểu_thức [NOT] IN (truy_vấn_con)

Ví dụ: Đề hiển thị họ tên của những sinh viên lớp Tin K25 có năm sinh bằng với năm sinh của một sinh viên nào đó của lớp Toán K25, thay vì sử dụng câu lệnh như ở ví dụ trên, ta có thể sử dụng câu lệnh như sau:

```
SELECT hodem,ten  
FROM sinhvien JOIN lop ON sinhvien.malop=lop.malop  
WHERE tenlop='Tin K25' AND  
year(ngaysinh)IN(SELECT year(ngaysinh)  
FROM sinhvien JOIN lop  
ON sinhvien.malop=lop.malop  
WHERE lop.tenlop='Toán K25')
```

Sử dụng lượng từ EXISTS với truy vấn con

Lượng từ EXISTS được sử dụng kết hợp với truy vấn con dưới dạng:

WHERE [NOT] EXISTS (truy_vấn_con)

để kiểm tra xem một truy vấn con có trả về dòng kết quả nào hay không. Lượng từ EXISTS (tương ứng NOT EXISTS) trả về giá trị True (tương ứng False) nếu kết quả của truy vấn con có ít nhất một dòng (tương ứng không có dòng nào). Điều khác biệt của việc sử dụng EXISTS với hai cách đã nêu ở trên là trong danh sách chọn của truy vấn con có thể có nhiều hơn hai cột.

Ví dụ: Câu lệnh dưới đây cho biết họ tên của những sinh viên hiện chưa có điểm thi của bất kỳ một môn học nào

```
SELECT hodem,ten  
FROM sinhvien  
WHERE NOT EXISTS(SELECT masv FROM diemthi  
WHERE diemthi.masv=sinhvien.masv)
```

Sử dụng truy vấn con với mệnh đề HAVING

Một truy vấn con có thể được sử dụng trong mệnh đề HAVING của một truy vấn khác. Trong trường hợp này, kết quả của truy vấn con được sử dụng để tạo nên điều kiện đối với các hàm gộp.

Ví dụ: Câu lệnh dưới đây cho biết mã, tên và trung bình điểm lần 1 của các môn học có trung bình lớn hơn trung bình điểm lần 1 của tất cả các môn học

```
SELECT diemthi.mamonhoc,tenmonhoc,AVG(diemlan1)
```

```
FROM diemthi,monhoc
```

```
WHERE diemthi.mamonhoc=monhoc.mamonhoc
```

```
GROUP BY diemthi.mamonhoc,tenmonhoc
```

```
HAVING AVG(diemlan1)>
```

```
(SELECT AVG(diemlan1) FROM diemthi)
```

i. Bổ sung, cập nhật và xoá dữ liệu

Các câu lệnh thao tác dữ liệu trong SQL không những chỉ sử dụng để truy vấn dữ liệu mà còn để thay đổi và cập nhật dữ liệu trong cơ sở dữ liệu. So với câu lệnh SELECT, việc sử dụng các câu lệnh để bổ sung, cập nhật hay xoá dữ liệu đơn giản hơn nhiều. Trong phần còn lại của chương này sẽ đề cập đến 3 câu lệnh:

- Lệnh INSERT
- Lệnh UPDATE
- Lệnh DELETE

Bổ sung dữ liệu

Dữ liệu trong các bảng được thể hiện dưới dạng các dòng (bản ghi). Để bổ sung thêm các dòng dữ liệu vào một bảng, ta sử dụng câu lệnh INSERT. Hầu hết các hệ quản trị CSDL dựa trên SQL cung cấp các cách dưới đây để thực hiện thao tác bổ sung dữ liệu cho bảng:

- Bổ sung từng dòng dữ liệu với mỗi câu lệnh INSERT. Đây là các sử dụng thường gặp nhất trong thao tác SQL.
- Bổ sung nhiều dòng dữ liệu bằng cách truy xuất dữ liệu từ các bảng dữ liệu khác.

Bổ sung từng dòng dữ liệu với lệnh INSERT

Để bổ sung một dòng dữ liệu mới vào bảng, ta sử dụng câu lệnh INSERT với cú pháp như sau:

```
INSERT INTO tên_bảng[(danh_sách_cột)]
```

```
VALUES(danh_sách_giá_trị)
```

Trong câu lệnh INSERT, danh sách cột ngay sau tên bảng không cần thiết phải chỉ định nếu giá trị các trường của bản ghi mới được chỉ định đầy đủ trong danh sách giá trị. Trong trường hợp này, thứ tự các giá trị trong danh sách giá trị phải bằng với số lượng các trường của bảng cần bổ sung dữ liệu cũng như phải tuân theo đúng thứ tự của các trường như khi bảng được định nghĩa.

Ví dụ: Câu lệnh dưới đây bổ sung thêm một dòng dữ liệu vào bảng KHOA
INSERT INTO khoa

```
VALUES('DHT10','Khoa Luật','054821135')
```

Trong trường hợp chỉ nhập giá trị cho một số cột trong bảng, ta phải chỉ định danh sách các cột cần nhập dữ liệu ngay sau tên bảng. Khi đó, các cột không được nhập dữ liệu sẽ nhận giá trị mặc định (nếu có) hoặc nhận giá trị NULL (nếu cột cho phép chấp nhận giá trị NULL). Nếu một cột không có giá trị mặc định và không chấp nhận giá trị NULL mà không được nhập dữ liệu, câu lệnh sẽ bị lỗi.

Ví dụ: Câu lệnh dưới đây bổ sung một bản ghi mới cho bảng SINHVIEN

```
INSERT INTO sinhvien(masv,hodem,ten,gioitinh,malop)
```

```
VALUES('0241020008','Nguyễn Công','Chính',1,'C24102')
```

câu lệnh trên còn có thể được viết như sau:

```
INSERT INTO sinhvien
```

```
VALUES('0241020008','Nguyễn Công','Chính',
```

```
NULL,1,NULL,'C24102')
```

Bổ sung nhiều dòng dữ liệu từ bảng khác

Một cách sử dụng khác của câu lệnh INSERT được sử dụng để bổ sung nhiều dòng dữ liệu vào một bảng, các dòng dữ liệu này được lấy từ một bảng khác thông qua câu lệnh SELECT. Ở cách này, các giá trị dữ liệu được bổ sung vào bảng không được chỉ định tường minh mà thay vào đó là một câu lệnh SELECT truy vấn dữ liệu từ bảng khác.

Cú pháp câu lệnh INSERT có dạng như sau:

INSERT INTO tên_bảng[(danh_sách_cột)] câu_lệnh_SELECT

Ví dụ: Giả sử ta có bảng LUUSINHVIEN bao gồm các trường HODEM, TEN, NGAYSINH. Câu lệnh dưới đây bổ sung vào bảng LUUSINHVIEN các dòng dữ liệu có được từ câu truy vấn SELECT:

INSERT INTO luusinhvien

SELECT hodem,tен,ngaysinh

FROM sinhvien

WHERE noisinh like '%Hué%'

Khi bổ sung dữ liệu theo cách này cần lưu ý một số điểm sau:

- Kết quả của câu lệnh SELECT phải có số cột bằng với số cột được chỉ định trong bảng đích và phải tương thích về kiểu dữ liệu.
- Trong câu lệnh SELECT được sử dụng mệnh đề COMPUTE ... BY

Cập nhật dữ liệu

Câu lệnh UPDATE trong SQL được sử dụng để cập nhật dữ liệu trong các bảng. Câu lệnh này có cú pháp như sau:

UPDATE tên_bảng

SET tên_cột = biểu_thức

[, ..., tên_cột_k = biểu_thức_k]

[FROM danh_sách_bảng]

[WHERE điều_kiện]

Sau UPDATE là tên của bảng cần cập nhật dữ liệu. Một câu lệnh UPDATE có thể cập nhật dữ liệu cho nhiều cột bằng cách chỉ định các danh sách tên cột và biểu thức tương ứng sau từ khoá SET. Mệnh đề WHERE trong câu lệnh UPDATE thường được sử dụng để chỉ định các dòng dữ liệu chịu tác động của câu lệnh (nếu không chỉ định, phạm vi tác động của câu lệnh được hiểu là toàn bộ các dòng trong bảng)

Ví dụ: Câu lệnh dưới đây cập nhật lại số đơn vị học trình của các môn học có số đơn vị học trình nhỏ hơn 2

UPDATE monhoc

SET sodvht = 3

WHERE sodvht < 2

Sử dụng cấu trúc CASE trong câu lệnh UPDATE

Cấu trúc CASE có thể được sử dụng trong biểu thức khi cần phải đưa ra các quyết định khác nhau về giá trị của biểu thức

Ví dụ: Giả sử ta có bảng NHATKYPHONG sau đây

SOPHONG	LOAIPHONG	SONGAY	TIENPHONG
101	A	5	
202	B	5	
101	A	2	
102	C	3	

Sau khi thực hiện câu lệnh:

UPDATE nhatkyphong

SET tienphong=songay*CASE WHEN loaiphong='A' THEN 100

WHEN loaiphong='B' THEN 70

ELSE 50

END

Dữ liệu trong bảng sẽ là:

SOPHONG	LOAIPHONG	SONGAY	TIENPHONG
101	A	5	500
202	B	5	350
101	A	2	200
102	C	3	150

Điều kiện cập nhật dữ liệu liên quan đến nhiều bảng

Mệnh đề FROM trong câu lệnh UPDATE được sử dụng khi cần chỉ định các điều kiện liên quan đến các bảng khác với bảng cần cập nhật dữ liệu. Trong trường hợp này, trong mệnh đề WHERE thường có điều kiện nối giữa các bảng.

Ví dụ: Giả sử ta có hai bảng MATHANG và NHATKYBANHANG như sau:



Câu lệnh dưới đây sẽ cập nhật giá trị trường THANHTIEN của bảng NHATKYBANHANG theo công thức THANHTIEN = SO LUONG * GIA
UPDATE nhatkybanhang

```
SET thanhtien = soluong*gia
FROM mathang
WHERE nhatkybanhang.mahang = mathang.mahang
```

Câu lệnh UPDATE với truy vấn con

Tương tự như trong câu lệnh SELECT, truy vấn con có thể được sử dụng trong mệnh đề WHERE của câu lệnh UPDATE nhằm chỉ định điều kiện đối với các dòng dữ liệu cần cập nhật dữ liệu.

Ví dụ: Câu lệnh ở trên có thể được viết như sau:

```
UPDATE nhatkybanhang
SET thanhtien = soluong*gia
FROM mathang
WHERE mathang.mahang =(SELECT mathang.mahang
FROM mathang
WHERE mathang.mahang=nhatkybanhang.mahang)
```

Xoá dữ liệu

Để xoá dữ liệu trong một bảng, ta sử dụng câu lệnh DELETE. Cú pháp của câu lệnh này như sau:

```
DELETE FROM tên_bảng
[FROM danh_sách_bảng]
[WHERE điều_kiện]
```

Trong câu lệnh này, tên của bảng cần xoá dữ liệu được chỉ định sau DELETE FROM. Mệnh đề WHERE trong câu lệnh được sử dụng để chỉ định điều kiện đối với các dòng dữ liệu cần xoá. Nếu câu lệnh DELETE không có mệnh đề WHERE thì toàn bộ các dòng dữ liệu trong bảng đều bị xoá.

Ví dụ: Câu lệnh dưới đây xoá khỏi bảng SINHVIEN những sinh viên sinh tại Huế

```
DELETE FROM sinhvien
```

WHERE noisinh LIKE '%Hué%'

Xoá dữ liệu khi điều kiện liên quan đến nhiều bảng

Nếu điều kiện trong câu lệnh DELETE liên quan đến các bảng không phải là bảng cần xóa dữ liệu, ta phải sử dụng thêm mệnh đề FROM và sau đó là danh sách tên các bảng đó. Trong trường hợp này, trong mệnh đề WHERE ta chỉ định thêm điều kiện nối giữa các bảng

Ví dụ: Câu lệnh dưới đây xoá ra khỏi bảng SINHVIEN những sinh viên lớp Tin K24

`DELETE FROM sinhvien`

`FROM lop`

`WHERE lop.malop=sinhvien.malop AND tenlop='Tin K24'`

Sử dụng truy vấn con trong câu lệnh DELETE

Một câu lệnh SELECT có thể được lồng vào trong mệnh đề WHERE trong câu lệnh DELETE để làm điều kiện cho câu lệnh tương tự như câu lệnh UPDATE.

Ví dụ: Câu lệnh dưới đây xoá khỏi bảng LOP những lớp không có sinh viên nào học

`DELETE FROM lop`

`WHERE malop NOT IN (SELECT DISTINCT malop`

`FROM sinhvien)`

Xoá toàn bộ dữ liệu trong bảng

Câu lệnh DELETE không chỉ định điều kiện đối với các dòng dữ liệu cần xoá trong mệnh đề WHERE sẽ xoá toàn bộ dữ liệu trong bảng. Thay vì sử dụng câu lệnh DELETE trong trường hợp này, ta có thể sử dụng câu lệnh TRUNCATE có cú pháp như sau:

`TRUNCATE TABLE tên_bảng`

Ví dụ: Câu lệnh sau xoá toàn bộ dữ liệu trong bảng diemthi:

`DELETE FROM diemthi`

có tác dụng tương tự với câu lệnh

`TRUNCATE TABLE diemthi`

4.3. Sử dụng điều khiển DATA để liên kết và khai thác cơ sở dữ liệu Access

4.3.1. Giới thiệu chung

Từ VB5, Visual Basic cho lập trình viên một control để truy cập cơ sở dữ liệu, đó là điều khiển Data. Như ta biết, có một cơ sở dữ liệu Microsoft gói kèm khi ta mua VB6 - đó là Jet Database Engine. Jet Database Engine là cái "phòng máy" của chính MS Access Database Management System.

Microsoft cho ra ba kỹ thuật chính để kết nối với cơ sở dữ liệu kể từ phiên bản VB5

- DAO (Data Access Objects): DAO là kỹ thuật bí truyền của Microsoft, chỉ để dùng với Jet Database Engine. Nó rất dễ dùng, hiệu năng và tiện, nhưng bị giới hạn trong phạm vi MS Access. Dù vậy, nó rất thịnh hành vì có lợi ích thực tiễn.
- ODBC (Open Database Connectivity): ODBC được thiết kế để cho phép users nối với dù loại databases mà chỉ dùng một method duy nhất. Điều này cắt bớt gánh nặng cho lập trình viên, để chỉ cần học một kỹ thuật lập trình duy nhất mà có thể làm việc với bất cứ loại database nào. Nhất là khi sau này nếu cần phải thay đổi loại database, như nâng cấp từ Access lên SQLServer chẳng hạn, thì sự sửa đổi về coding rất ít. Khi dùng ODBC chung với DAO, ta có thể cho Access Database nối với các databases khác.
- RDO (Remote Data Object): Một trong những lý do chính để RDO được thiết kế là giải quyết khó khăn về sự rắc rối của ODBC. Cách lập trình với RDO đơn giản như DAO, nhưng thật ra nó dùng ODBC nên cho phép users nối với nhiều databases. Tuy nhiên, RDO không được thịnh hành lắm.

VB6 tiếp tục hỗ trợ các kỹ thuật nói trên, và cho thêm một kỹ thuật truy cập database mới, rất quan trọng, đó là ADO (ActiveX Data Objects). Trong một bài tới ta sẽ học về ADO với những ưu điểm của nó. Tuy nhiên, vì DAO rất đơn giản và hiệu năng nên ta vẫn có thể tiếp tục dùng nó rất hữu hiệu trong hầu hết các ứng dụng.

Chú ý: Bạn cần phải cài bản sửa lỗi và nâng cấp Visual Basic Service Pack 5 trở lên nếu muốn ứng dụng có thể kết nối với cơ sở dữ liệu Access 2000 trở lên.

4.3.2. Sử dụng điều khiển Data để liên kết và khai thác cơ sở dữ liệu Access

a. Thiết lập liên kết

- Điều khiển Data mặc định có sẵn trong hộp Toolbox, ta mở Form cần thiết kế, lựa chọn điều khiển Data, xác định vị trí và kích thước của nó trên Form
- Mở cửa sổ thuộc tính của đối tượng Data (Properties)

+ Tại mục: Connect chọn Access 2000

+ Tại mục Database Name, nhấn chuột vào nút (...) để hiện cửa sổ cho phép tìm tới tệp CSDL Access. Ở đây ta chọn

E:\Program Files\Microsoft Visual Studio\VB98\BIBLIO.MDB, trong máy tính của bạn có thể nó nằm trên disk C hay D.

+ Tại mục RecordSource: lựa chọn tên bảng nằm trong CSDL đã được kết nối

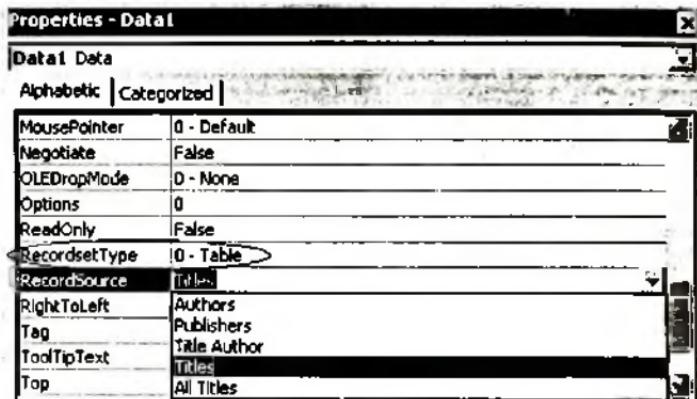
(tệp CSDL Access ở trên mục DatabaseName đã chọn)



Trong chương trình này ta muốn làm việc với table Titles của database BIBLIO.MDB, để xem và cập nhật các bản ghi.

Để ý property **DefaultType** của Data1 có trị số 2- UseJet, tức là dùng kỹ thuật DAO, thay vì dùng kỹ thuật ODBC. Khi bạn click lên property

Recordsource của Data1, rồi click lên cái tam giác nhỏ bên phải, một ComboBox sẽ mở ra cho ta thấy danh sách các tables trong database. Bạn hãy chọn **Titles**. Để ý **property RecordsetType** của Data1 có trị số là **0 - Table**:



Chú ý: Theo cách khai báo Database name như trên, đường dẫn đến tệp CSDL sẽ cố định là:

E:\ProgramFiles\MicrosoftVisual studio\VB98\BIBLIO.MDB

Điều này sẽ dẫn tới việc khi bạn cài đặt dự án sang một máy tính khác thì nếu trên máy tính đó không tồn tại đường dẫn giống như máy của bạn khi xây dựng dự án: E:\Program Files\Microsoft Visual studio\VB98\BIBLIO.MDB thì khi thi hành ứng dụng sẽ báo lỗi là không thể tìm thấy tệp CSDL tại đường dẫn: E:\Program Files\Microsoft Visual studio\VB98\BIBLIO.MDB.

Để khắc phục vấn đề này chúng ta sẽ tiến hành khai báo đường dẫn động tới tệp CSDL như sau:

Giả dụ ta muốn để database trong cùng một thư mục với chương trình đang chạy, ta có thể dùng **property Path** của Application Object App như sau:

```
Dim AppFolder As String  
Private Sub Form_Load()  
AppFolder = App.Path  
Data1.DatabaseName = AppFolder & "BIBLIO.MDB"
```

End Sub

Với cách code nói trên ta sẽ đảm bảo chương trình tìm thấy file database đúng chỗ, không cần biết người ta cài chương trình bạn ở đâu trong đĩa cứng của máy tính.

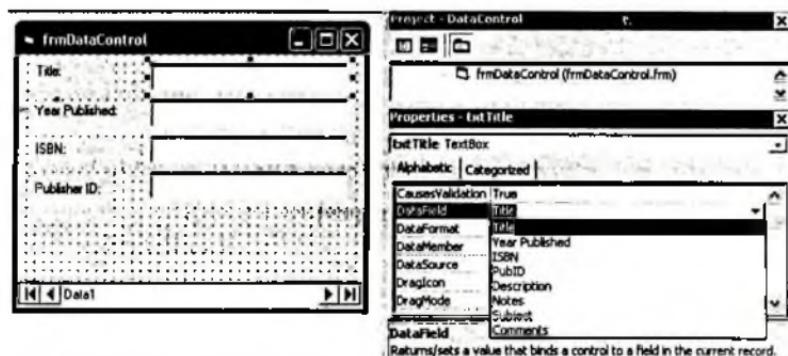
b. Liên kết các đối tượng trên form với CSDL

Bây giờ bạn hãy đặt lên Form 4 labels với captions: **Title**, **Year Published**, **ISBN** và **Publisher ID**.

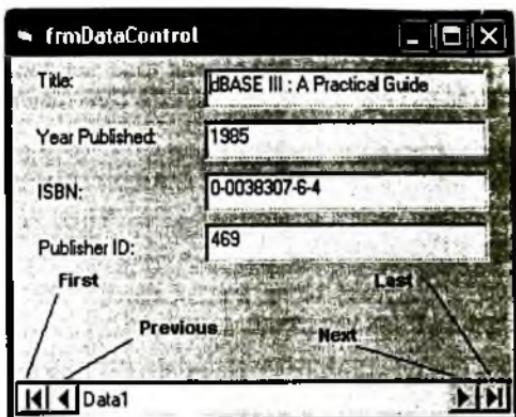
Kế đó cho thêm 4 textboxes tương ứng và đặt tên chúng là **txtTitle**, **txtYearPublished**, **txtISBN** và **txtPublisherID**.

Chọn textbox **txtTitle**, rồi set property **Datasource** của nó trong Properties Window thành **Data1**.

Khi click lên property **Datafield** của **txtTitle** và mở ComboBox ra bạn sẽ thấy liệt kê tên các Fields trong table **Titles**. Đó là vì Data1 được coi như trung gian lấy table **Titles** từ database. Ở đây ta sẽ chọn cột **Title**. Lặp lại công tác này cho 3 textboxes kia, và chọn các cột **Year Published** (năm xuất bản), **ISBN** (số lý lịch trong thư viện quốc tế), và **PubID** (số lý lịch nhà xuất bản) làm Datafield cho chúng.



Tới đây, mặc dù chưa viết một hàng code, ta có thể chạy chương trình được rồi. Nó sẽ hiển thị chi tiết của record đầu tiên trong table **Titles** như dưới đây:



Bạn có thể bấm các nút di chuyển Navigator Buttons để di đến các record đầu (**first**), trước (**previous**), kế (**next**) và cuối (**last**).

Mỗi lần bạn di chuyển đến một record mới là chi tiết của record ấy sẽ hiển thị. Nếu không dùng các Navigator Buttons, ta cũng có thể code để làm công tác tương đương bằng cách gọi các Recordset methods **MoveFirst**, **MovePrevious**, **MoveNext** và **MoveLast**.

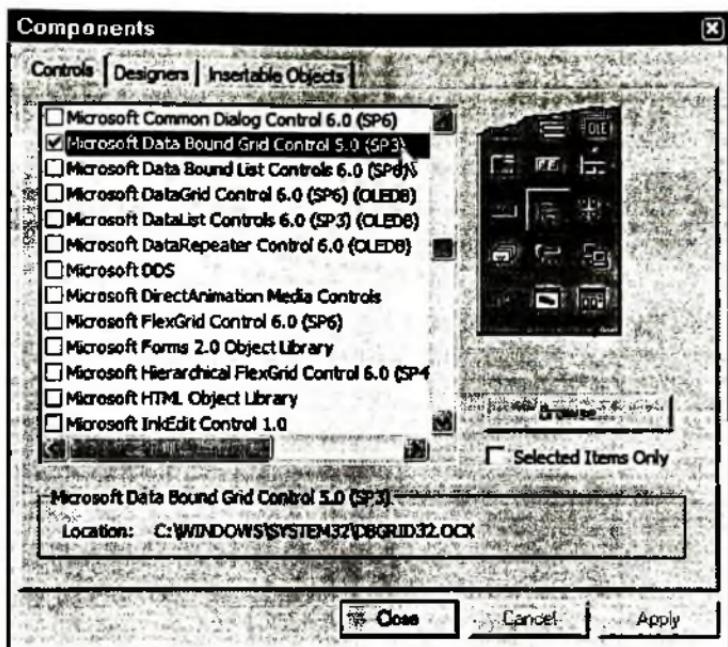
Khi record cuối của Recordset đang hiển thị, nếu ta gọi method **MoveLast** thì property **EOF** (End-Of-File) của Recordset trở thành True. Tương tự như vậy, khi record thứ nhất của Recordset đang hiển thị, nếu ta gọi method **MovePrevious** thì property **BOF** (Begin-Of-File) của Recordset trở thành True.

Nếu một Recordset không có chứa một record nào cả thì cả hai properties **EOF** và **BOF** đều là True.

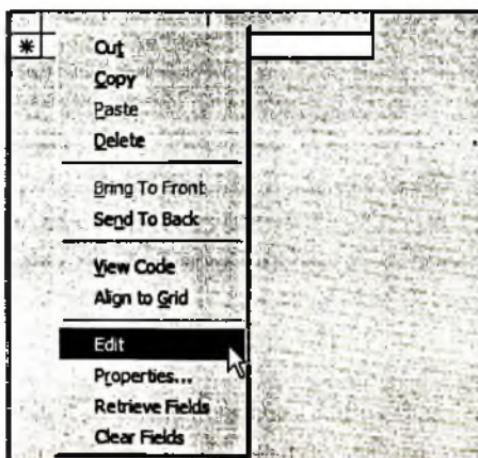
c. Liên kết với CSDL data với đối tượng ô lưới (Grid)

- Khai báo sử dụng đối tượng ô lưới bằng cách:

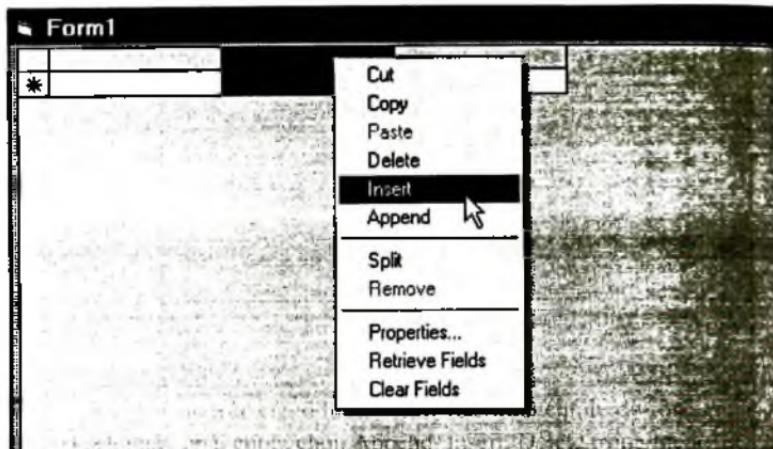
Chọn Menu Project/Component/tích vào dòng Microsoft Data Bound Grid Control 5.0



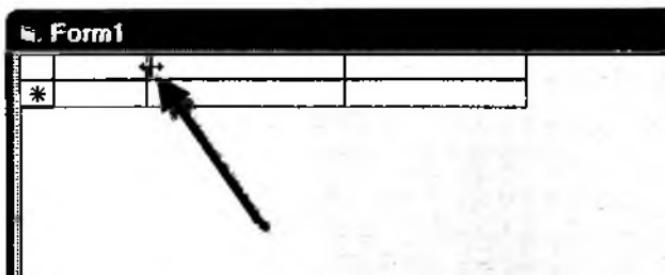
- Lựa chọn đối tượng DBGrid trong hộp ToolBox và vẽ lên Form
- Bấm chuột phải lên ô lưới và chọn Edit để bắt đầu chức năng thiết kế ô lưới



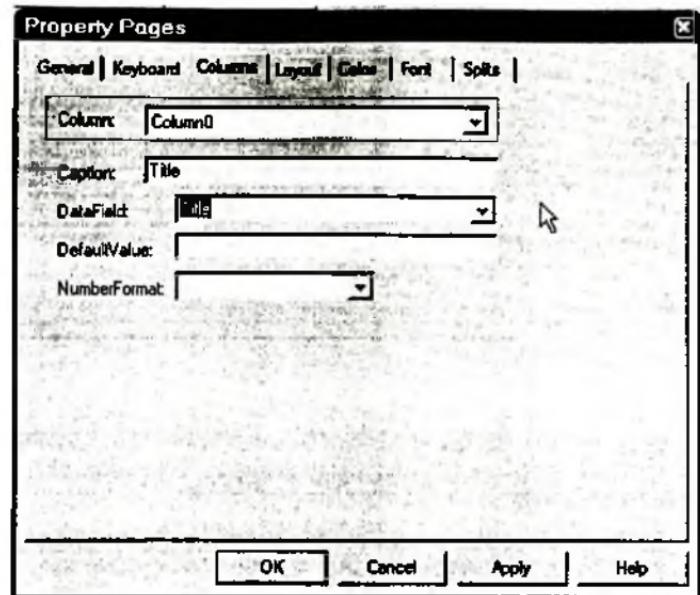
- Thiết kế cấu trúc của ô lưới: thêm, xoá, điều chỉnh, độ rộng cột bằng cách nháy phải chuột chọn Append, Insert, Delete trong menu hiện ra



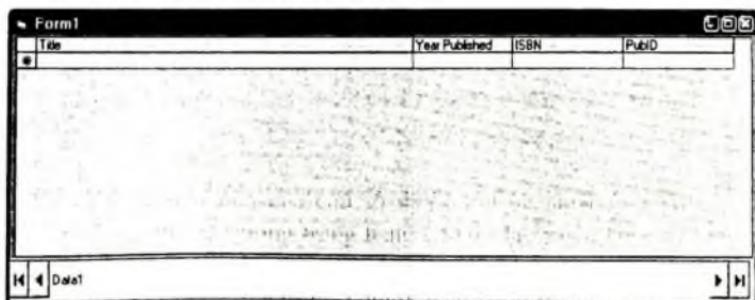
- Điều chỉnh độ rộng của cột bằng cách di chuột đến phần biên bên phải của cột, chơ chuột biến thành hình mũi tên 2 chiều; bấm giữ nút trái chuột trong khi điều chỉnh kích thước cột.



- Để nhập tiêu đề cột và khai báo tên trường liên kết với cột.
 - Trước hết bấm F4 để hiện cửa sổ thuộc tính của ô lưới, tại mục Data Source chọn tên của điều khiển Data muốn liên kết (Data1)
 - Nháy phải chuột vào ô lưới chọn Properties... để hiện ra một cửa sổ thuộc tính thứ 2.



- Tại thẻ Column của cửa sổ thuộc tính ta lần lượt khai báo các thông tin về trường trong bảng CSDL được liên kết cho từng cột bao gồm: Caption (tiêu đề cột); DataField (tên trường liên kết tới); DefaultValue (giá trị mặc định); NumberFormat (định dạng số).
- Tương tự ví dụ trên, ta cũng khai báo 4 cột trong ô lưới liên kết với 4 trường trong bảng Titles gồm: **Title**, **Year Published**, **ISBN** và **Publisher ID**.
- Ô lưới hoàn thành có dạng như sau:

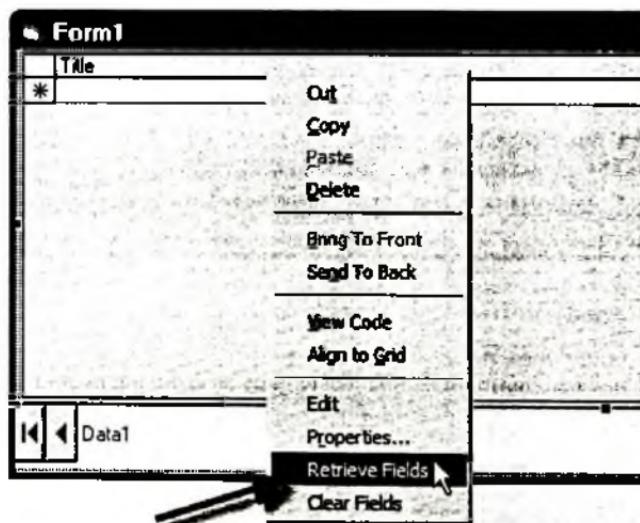


o Chạy thử Form ta có

Title	Year Published	ISBN	Publisher
cBASE III : A Practical Guide	1985	0-0303074-4	422
The PASCAL Programming Language	1986	0-03032574-8	423
cBASE III Plus	1987	0-03033764-X	423
Database Management - Developing Application Systems Using Oracle	1988	0-03139852-1	715
Wordstar 4.0 & Quick Reference Guide	1990	0-01336561-4	460
Oracle Triggers and Stored Procedure Programming	1996	0-01344363-3	715
Programming in Clipper	1998	0-20114563-3	8
Inside Macintosh	1994	0-20140573-3	9
Orms Online Database Directory	1993	0-2079320-9	135
Structured C for Engineering and Technology/Book and Diskette	1995	0-2210081-2	716
An Introduction to Assembly Language Programming for the Intel 8086 Micro	1995	0-0230362-0-6	716
Applied Calculus With Linear Programming : For Business, Economics, Life	1995	0-02305504-7	515
...[10]		0-02305511-2	515

Chú ý:

- Thay vì thực hiện tất cả các công việc khai báo như trên, nếu bạn muốn có bao nhiêu trường bảng CSDL hiện ra tất trong ô lưới. Nháy chuột phải vào ô lưới chọn **Retrieve Field**.



- Ô lưới của chúng ta giờ sẽ tự động được liên kết với bảng trong CSDL với số cột bằng đúng số trường của bảng

Form1

Title	Year Published	ISBN	PubID	Description	Notes	Subject
DBASE III Advanced	1985	0-03813074-4	483	225		
The dBASE Programer	1986	0-03813257-8	483	225	008.798520	
dBASE III Plus	1987	0-03813274X	483	285		
Database Management	1989	0-01319852-1	715	54	008.7419	
Wordstar 4.0 & DOS	1990	0-01226251-4	480	14.85		
Oracle Triggers and	1995	0-01344263-1	715	0	008.798520	
Programming in C/C++	1998	0-03011458-3	910	0	008.798520	2nd
Inside Macintosh	1994	0-02014057-3	910	99.01	008.778520	
Office Online Database	1995	0-01344263-1	158	0	025.0419025	
Structured C for Engineers	1995	0-02300812-1	715	54	008.265520	2nd Ed.& disk
An Introduction to Mac	1995	0-02303620-6	715	60	008.265520	Book & Disk
... A Practical Approach	1995	0-02303620-6	715	60	008.265520	Book & Disk

Data1

- Trong cửa sổ thuộc tính của ô lưới ta có thêm một số tùy chọn sau
 - o AllowAddnew (true/false) cho phép hoặc không việc người dùng có thể thêm bản ghi mới vào bảng CSDL. Mặc định là False.
 - o AllowDelete (true/false) cho phép hoặc không việc người dùng có thể xóa bản ghi trong ô lưới. Mặc định là False.
 - o AllowUpdate (true/false) cho phép hoặc không việc hệ thống tự động cập nhật nội dung bản ghi nếu có sự thay đổi. Mặc định là True.
- Khi ứng dụng hoạt động, muốn lấy ra hay cập nhật giá trị của một ô trong ô lưới ta sử dụng cấu trúc câu lệnh như sau:
 - o Lấy giá trị tại một cột xác định tương ứng với bản ghi hiện thời
 - o <biến>=DBGrid1.Columns(<STT cột>)
 - o Lấy giá trị tại ô hiện thời đang được lựa chọn
 - o <biến>=DBGrid1.Text

f. Lập trình với điều khiển Data

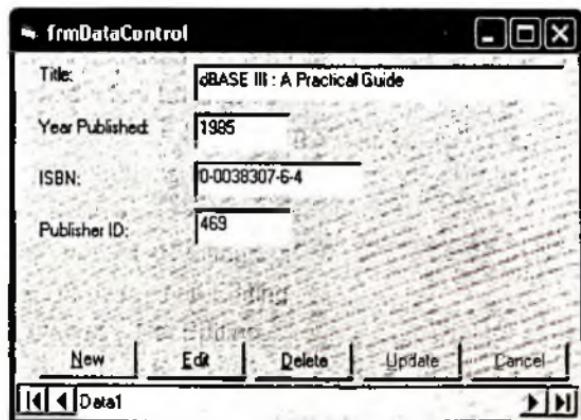
Thêm bớt các Records

Chương trình trên dùng cũng tạm được, nhưng nó không cho ta phương tiện để thêm (add), bớt (delete) các records. Vậy giờ bạn hãy để vào Form 5 buttons tên: cmdEdit, cmdNew, cmdDelete, cmdUpdate và cmdCancel. Mặc dù bạn không thấy, nhưng thật ra Control Data Data1 có một property Recordset và khi ta dùng Navigator buttons là di chuyển từ record này đến record khác trong Recordset ấy. Ta có thể nói đến nó bằng Notation (cách viết) Data1.Recordset, và mỗi lần muốn lấy Recordset mới nhất từ database ta dùng method Refresh như Data1.Recordset.Refresh. Lúc chương trình mới khởi động, user đang xem (browsing) các records thì hai

buttons **Update** và **Cancel** không cần phải làm việc. Do đó ta sẽ nhân tiện Lock (khóa) các textboxes và disable (làm cho bất lực) hai buttons này vì không cần dùng chúng. Trong Sub **SetControls** dưới đây, ta dùng một parameter gọi là **Editing** với trị số False hay True tùy theo user đang Browse hay Edit, ta gọi là **Browse mode** và **Edit mode**. Trong **Edit mode**, các Textboxes được unlocked (mở khóa) và các nút **cmdNew**, **cmdDelete** và **cmdEdit** trở nên bắt lực:

```
Sub SetControls(ByVal Editing As Boolean)
    ' Lock/Unlock textboxes
    txtTitle.Locked = Not Editing
    txtYearPublished.Locked = Not Editing
    txtISBN.Locked = Not Editing
    txtPublisherID.Locked = Not Editing
    ' Enable/Disable buttons
    CmdUpdate.Enabled = Editing
    CmdCancel.Enabled = Editing
    CmdDelete.Enabled = Not Editing
    cmdNew.Enabled = Not Editing
    CmdEdit.Enabled = Not Editing
End Sub
```

Trong **Browse mode**, Form có dạng như sau:



Sub SetControls được gọi trong **Sub Form_Load** khi chương trình khởi động và trong **Sub CmdEdit** khi user click nút **Edit** như sau:

```
Private Sub Form_Load()
    ' Fetch Folder where this program EXE resides
    AppFolder = App.Path
    ' make sure it ends with a back slash
    If Right(AppFolder, 1) <> "\" Then AppFolder = AppFolder & \
    ' Assign Full path database filename to Data1
    Data1.DatabaseName = AppFolder & "BIBLIO.MDB"
    ' Place controls in Browse Mode
    SetControls (False)
End Sub

Private Sub CmdEdit_Click()
    ' Place controls in Edit Mode
    SetControls (True)
End Sub
```

Khi ta Delete một record trong recordset, vị trí của record hiện tại (current record) vẫn không thay đổi. Do đó, sau khi delete một record ta phải **MoveNext**. Nhưng, nếu ta vừa delete record cuối của Recordset thì sau khi **MoveNext**, **property EOF** của Recordset sẽ thành True. Thành ra ta phải kiểm tra điều đó, nếu đúng vậy thì lại phải **MoveLast** để hiển thị record cuối của Recordset như trong code của **Sub cmdDelete_Click** dưới đây:

```
Private Sub CmdDelete_Click()
On Error GoTo DeleteErr
With Data1.Recordset
    Delete new record
    Delete
    Move to next record
    MoveNext
    If .EOF Then .MoveLast
```

```
Exit Sub  
End With  
DeleteErr:  
MsgBox Err.Description  
Exit Sub  
End Sub
```

Trong lúc code, ta cập nhật một bản ghi trong Recordset bằng method **Update**. Nhưng ta chỉ có thể gọi method **Update** của một Recordset khi Recordset đang ở trong **Edit hay AddNew mode**. Ta đặt một Recordset vào Edit mode bằng cách gọi method **Edit** của Recordset, thí dụ như **Data1.Recordset.Edit**. Tương tự như vậy, ta đặt một Recordset vào AddNew mode bằng cách gọi method **AddNew** của Recordset, thí dụ như **Data1.Recordset.AddNew**.

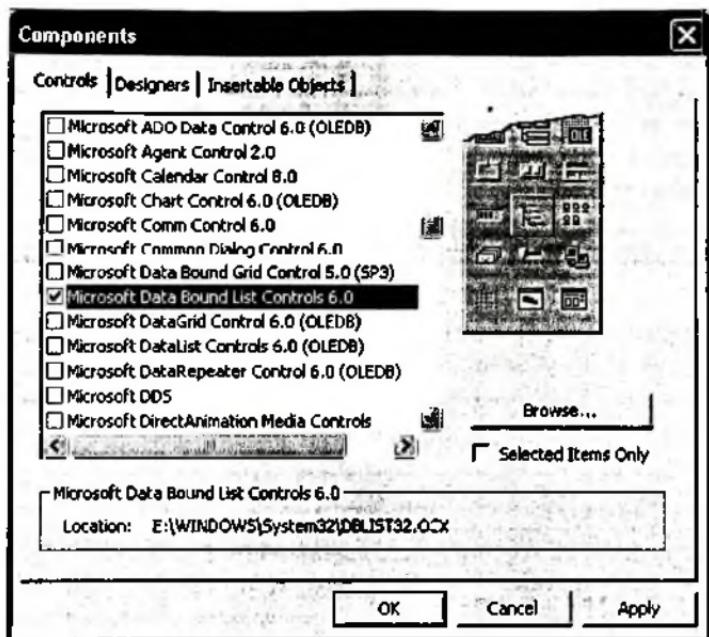
```
Private Sub cmdNew_Click()  
' Place Recordset into Recordset AddNew mode  
Data1.Recordset.AddNew  
' Place controls in Edit Mode  
SetControls (True)  
End Sub
```

Sau khi Recordset gọi method **Update** thì Recordset **离开** ra khỏi AddNew hay Edit modes. Ta cũng có thể tự thoát ra khỏi AddNew hay Edit modes, hay nói cho đúng hơn là hủy bỏ mọi pending (đang chờ đợi) Update bằng cách gọi method **CancelUpdate**, thí dụ như **Data1.Recordset.CancelUpdate**.

g. Sử dụng DataBound Combo (DB combo) với điều khiển Data

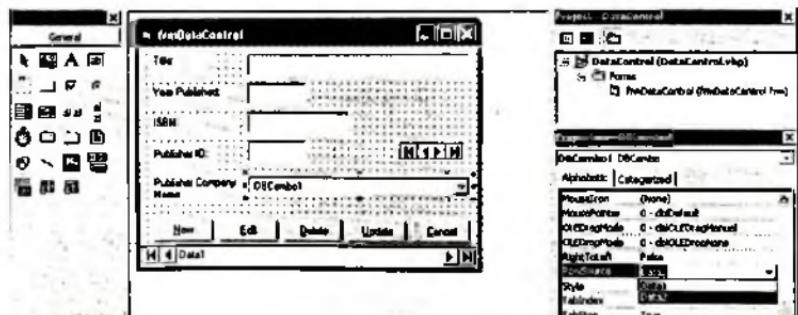
Trong chương trình hiện tại ta chỉ hiển thị lý lịch nhà xuất bản (PubID) của Title, chứ không có thêm chi tiết. Mặc dù chương trình lưu trữ PubID, nhưng hiển thị được Company Name của nhà xuất bản cho ta làm việc để không phải nhớ các con số thì hay quá.

Ta có thể thực hiện điều đó bằng cách dùng Control DBCombo (Data Bound Combo). Bạn hãy dùng IDE Menu Command Project | Components... để chọn **Microsoft Data Bound List Controls 6.0** rồi click Apply.



Kế đó, thêm một DBCombo tên **DBCombo1** vào Form. Vì ta cần một Recordset khác để cung cấp Table Publisher cho DBCombo1, nên bạn hãy thêm một control Data thứ nhì tên **Data2** vào Form. Cho Data2, hãy set property DatabaseName thành

E:\Program Files\Microsoft Visual Studio\VB98\BIBLIO.MDB và property RecordSource thành Publishers. Để không cho người ta thấy hình Data2 lúc run-time, bạn hãy set property **Visible** thành False.



Mục đích của chúng ta khi dùng DBCombo1 là hiển thị Company Name của nhà xuất bản, nhưng đăng sau lưng thì không có gì thay đổi, tức là ta vẫn làm việc với PubID cho các record Title của Data1. Khi user click lên DBCombo1 để chọn một nhà xuất bản, thì ta theo Company Name đó mà chứa PubID tương ứng trong record Title của Data1. Do đó ta phải sắp đặt cho DBCombo1 như sau:

Property	Value	Chú thích
RowSource	Data2	Đây là datasource của chính DBCombo1. Nó cung cấp table Publishers.
Listfield	Company Name	Khi RowSource phía trên đã được chọn rồi, Combo của property Listfield này sẽ hiển thị các fields của table Publishers. Company Name là field của RowSource mà ta muốn hiển thị trên DBCombo1.
DataSource	Data1	Đây là datasource của record mà ta muốn edit, tức là record của table Titles
Datafield	PubID	Field (của record Title) sẽ được thay đổi.
BoundColumn	PubID	Field trong RowSource (table Publishers) tương ứng với item user chọn trong DBCombo1 (Company Name).

Khi trong Edit mode user chọn một Company Name khác trong DBCombo1 rồi click nút Update bạn sẽ thấy Textbox txtPublisherID cũng đổi theo và hiển thị con số lý lịch PubID mới. Nếu trước khi Update bạn muốn thấy PubID mới hiển thị trong Textbox txtPublisherID thì bạn có thể dùng Event Click của DBCombo1 như sau:

```
Private Sub DBCombo1_Click(Area As Integer)
' Display new PubID
txtPublisherID.Text = DBCombo1.BoundText
End Sub
```

Property BoundText của DBCombo1 là trị số của BoundColumn mà ta có thể truy cập (viết hay đọc) được. Thí dụ như bạn muốn mỗi khi thêm một record Title mới thì default PubID là 324, tức là Company Name= "GLOBAL ENGINEERING". Bạn có thể assign trị số 324 vào property BoundText của DBCombo1 trong Sub cmdNew_Click như sau:

```
Private Sub cmdNew_Click()
```

```
' Place Recordset into Recordset AddNew mode  
Data1.Recordset.AddNew  
' Default Publisher is "GLOBAL ENGINEERING", i.e. PubID=324  
DBCombo1.BoundText = 324  
' Place controls in Edit Mode  
SetControls (True)  
End Sub
```

h. Lập trình nạp dữ liệu vào hộp combobox

Điều khiển ComboBox là điều khiển luôn có sẵn trong hộp Toolboxes. Khi lập trình CSDL nó sẽ không được thuận tiện như điều khiển DBCombo. Mặc dù vậy nếu muốn, chúng ta vẫn có thể chủ động nạp dữ liệu và lập trình với điều khiển ComboBox.

Giả sử đoạn chương trình dưới đây sẽ có chức năng nạp Danh mục Khoa vào hộp ComboBox có tên là Combo1 thông qua sự kiện người dùng chọn nút chức năng Nạp danh mục Khoa trên Form.

```
Private Sub CmdNapDMKhoa_Click ()  
    Data2.RecordSource = "Select * from DMKhoa"  
    Data2.Refresh  
    Data2.RecordSet.MoveFirst  
    Combo1.Clear  
    Do While Not Data2.Recordset.EOF()  
        With Data2.Recordset  
            Combo1.AddItem !MaKhoa & "_" & !TenKhoa  
            .MoveNext  
        EndWith  
    Loop  
End Sub
```

Trong đoạn chương trình này ta đã nạp vào hộp Combobox một danh sách bao gồm cả Mã khoa và Tên khoa cách nhau bởi dấu gạch chéo (Combo1.AddItem !MaKhoa & "_" & !TenKhoa). Khi ứng dụng thực hiện, người dùng có thể chọn một dòng bao gồm cả mã khoa và tên khoa trong

hộp combobox, ta có thể lấy ra giá trị được lựa chọn thông qua thuộc tính text của hộp combo (vd. Luachon=Combo1.Text). Và để có thể lấy ra Mã khoa ta sẽ viết một hàm chức năng riêng sử dụng đặc điểm là Mã khoa và tên khoa được cách nhau bởi 1 dấu gạch chân.

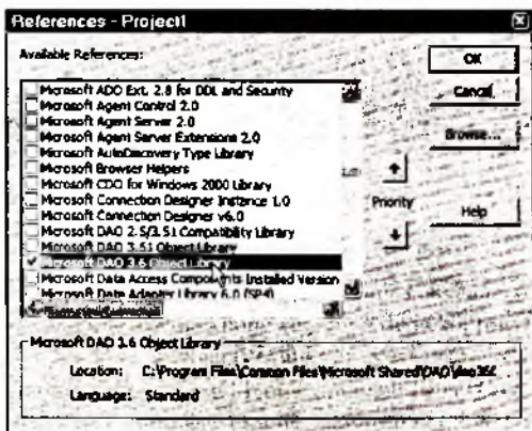
Hàm tách mã có thể viết như sau:

```
Public Function Tachma (st as string) as String  
Dim i As Integer  
i = 1  
Do While Mid (st, i, 1) <> "_"  
i = i + 1  
Loop  
Tachma = Left (st,i-1)  
End Function
```

4.4. Lập trình sử dụng thư viện DAO

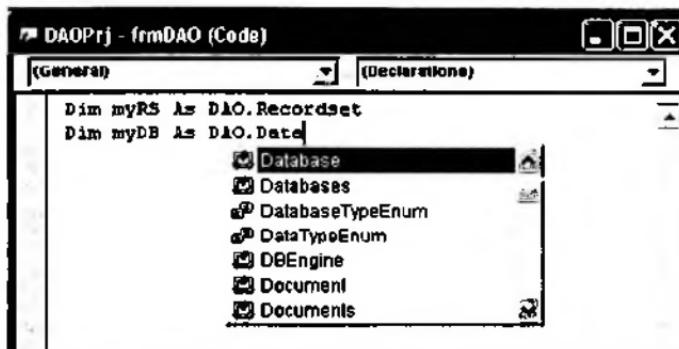
4.4.1. Tổng quát

- Trong bài này ta sẽ học những cách lập trình căn bản với MS Access database qua kỹ thuật DAO mà không cần dùng đến Control Data như trong bài trước. Ta sẽ cần đến vài Objects trong thư viện DAO, do đó nếu bạn mở một dự án VB6 mới thì hãy dùng Menu Command Project | References... để chọn Microsoft DAO 3.6 Object Library bằng cách click cái checkbox bên trái như trong hình dưới đây.



- Khai báo các đối tượng liên quan. Trong cửa sổ lập trình của Form chính chúng ta khai báo 2 biến như sau:

- Dim myDB as DAO.Database
- Dim myRS as DAO.Recordset



Biến myDB có nhiệm vụ liên kết (mở) một tệp cơ sở dữ liệu. Biến myRS có nhiệm vụ mở một bảng trong cơ sở dữ liệu hay là một câu lệnh truy vấn từ nhiều bảng.

- Để có thể ứng dụng tốt hơn, ta sẽ đi làm rõ khái niệm Recordset sử dụng trong lập trình cơ sở dữ liệu: Recordset là một **Set of records**, nó có thể chứa một số records hay không có record nào cả. Một record trong Recordset có thể là một record lấy từ một Table.

Trong trường hợp ấy có thể ta lấy về tất cả records trong table hay chỉ những records thỏa đúng một điều kiện, thí dụ như ta chỉ muốn lấy các records của những sách xuất bản trước năm 1990 (Year Published < 1990). Một Record trong Recordset cũng có thể là tập hợp các cột (columns) từ hai (hay ba) tables qua các mối liên hệ one-to-one và one-to-many. Thí dụ như khi lấy các records từ table Titles, ta muốn có thêm chi tiết tên công ty (Company Name) và điện thoại (Telephone) của nhà xuất bản (table Publishers) bằng cách dùng **Foreign Key PubID** trong table Titles làm **Primary Key** trong table Publishers để lấy các chi tiết ấy.

4.4.2. Thủ tục mở Database và mở Recordset

- ❖ Thủ tục mở Database

```
Set myDB = OpenDatabase(AppFolder & "BIBLIO.MDB")
```

Để ý chữ Set trong câu lệnh trên. Đó là vì myDB là một con trỏ (Pointer) đến một đối tượng (Object). Mặc dù từ nay về sau ta sẽ dùng myDB như một Database theo cách giống như bắt cứ biến thuộc kiểu dữ liệu nào khác, nhưng khi chỉ định lần đầu là nó từ đâu đến thì ta dùng chữ Set, để nói rằng thật ra myDB không phải là Object Database (giống như điều khiển Data), nhưng là Pointer đến Object Database.

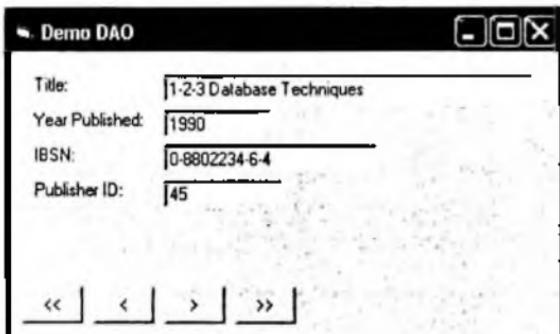
- ❖ Thủ tục mở một Recordset

```
Set myRS=myDB.OpenRecordset(<table name/ SQL>)
```

Tham số của thủ tục có thể là tên chính xác của một bảng trong CSDL, có thể là một câu lệnh truy vấn SQL lấy dữ liệu từ nhiều bảng.

4.4.3. Lập trình hiển thị nội dung một bản ghi trong Recordset

Ta thiết kế một Form như sau:



Ta lần lượt thêm các đoạn lệnh sau vào cửa sổ lập trình

```
Private Sub Form_Load()
```

```
' Fetch Folder where this program EXE resides
```

```
AppFolder = App.Path
```

```
' make sure it ends with a back slash
```

```
If Right(AppFolder, 1) <> "\" Then AppFolder = AppFolder & "\\"
```

```
' Open main database
```

```
Set myDB = OpenDatabase(AppFolder & "BIBLIO.MDB")
```

```
'Open recordset
```

```
Set myRS = myDB.OpenRecordset("Select * from Titles ORDER BY Title")
```

```
' if Recordset is not empty then display the first record
If myRS.RecordCount > 0 Then
myRS.MoveFirst ' move to first record
Displayrecord
'display details of current record
End If
End Sub
```

- Sau khi dùng **method MoveFirst** của Recordset để position **current record** ở Record đầu tiên, ta hiển thị trị số các fields của record bằng cách liên kết chúng vào các ô text của Form như sau:

```
Private Sub Displayrecord()
' Assign record fields to the appropriate textboxes
With myRS
' Assign field Title to textbox txtTitle
txtTitle.Text = .Fields("Title")
txtYearPublished.Text = .Fields("[Year Published]")
txtISBN.Text = .Fields("ISBN")
txtPublisherID.Text = .Fields("PubID")
End With
End Sub
```

Các nút di chuyển

Muốn có các nút Navigators tương đương với của một Control Data, bạn hãy đặt lên Form 4 buttons mang tên **CmdFirst**, **CmdPrevious**, **CmdNext** và **CmdLast** với captions: <<, <, >, >>.

Code cho các nút này cũng đơn giản, nhưng ta phải coi chừng khi user muốn di chuyển qua record cuối cùng hay record đầu tiên. Ta phải kiểm tra xem **EOF** có trở thành True khi user click **CmdNext**, hay **BOF** có trở thành True khi user click **CmdPrevious**:

```
Private Sub CmdNext_Click()
myRS.MoveNext ' Move to next record
'Display record details if has not gone past the last record
If Not myRS.EOF Then
```

```
Displayrecord ' display details of current record
Else
myRS.MoveLast ' Move back to last record
End If
End Sub

Private Sub CmdPrevious_Click()
myRS.MovePrevious ' Move to previous record
' Display record details if has not gone past the first record
If Not myRS.BOF Then
Displayrecord ' display details of current record
Else
myRS.MoveFirst ' Move back to first record
End If
End Sub

Private Sub CmdFirst_Click()
myRS.MoveFirst ' Move back to first record
Displayrecord ' display details of current record
End Sub

Private Sub CmdLast_Click()
myRS.MoveLast ' Move back to last record
Displayrecord ' display details of current record
End Sub
```

Khi chạy chương trình bạn sẽ thấy nó hiển thị chi tiết của Record đầu tiên khác với trong bài trước đây vì các records đã được sắp xếp (Set myRS = myDB.OpenRecordset("Select * from Titles ORDER BY Title")).

4.4.4. Lập trình thêm, bớt, tìm kiếm một bản ghi

Bây giờ bạn hãy đè vào Form 5 buttons tên:

cmdEdit, cmdNew, cmdDelete, cmdUpdate và cmdCancel.

Ta sẽ dùng lại Sub SetControls với parameter **Editing** có trị số False hay True tùy theo user đang Browse hay Edit. Trong **Browse mode**, các Textboxes bị Locked (khóa) và các nút **cmdUpdate** và **cmdCancel** trở nên bất lực. Trong **Edit mode**, các Textboxes được unlocked (mở khóa) và các

nút cmdNew, cmdDelete và cmdEdit trở nên bất lực.

Vì ở đây không có Data Binding nên đợi cho đến khi **Update (cập nhật hóa)** ta mới đặt Recordset vào **AddNew hay Edit mode**. Do đó ta chỉ cần nhớ là khi user edits là đang Edit một record hiện hữu hay thêm một Record mới. Ta chứa trị số Boolean ấy trong biến **AddNewRecord**. Nếu user sắp thêm một record mới thì AddNewRecord = True, nếu User sắp Edit một record hiện hữu thì AddNewRecord = False. Ngoài ra, khi User sắp thêm một record mới bằng cách click nút New thì ta phải tự clear (làm trống) hết các textboxes bằng cách gán ký tự trắng vào thuộc tính text của chúng như sau:

```
' If Editing existing record then AddNewRecord = False  
' Else AddNewRecord = true  
Dim AddNewRecord As Boolean  
Private Sub ClearAllFields()  
' Clear all the textboxes  
txtTitle.Text = ""  
txtYearPublished.Text = ""  
txtISBN.Text = ""  
txtPublisherID.Text = ""  
End Sub
```

Lập trình cho các nút bấm lần lượt như sau:

```
Private Sub cmdNew_Click()  
' Remember that this is Adding a new record  
AddNewRecord = True  
' Clear all textboxes  
ClearAllFields  
' Place controls in Edit Mode  
SetControls (True)  
End Sub  
  
Private Sub CmdEdit_Click()  
' Place controls in Edit Mode  
SetControls (True)  
' Remember that this is Editing an existing record
```

```
AddNewRecord = False
```

```
End Sub
```

Nếu user clicks Cancel trong khi đang edit các textboxes, ta không cần gọi **method CancelUpdate** vì Recordset chưa bị đặt vào AddNew hay Edit mode. Ở đây ta chỉ cần hiển thị lại chi tiết của current record, tức là hủy bỏ những gì user đang đánh vào:

```
Private Sub CmdCancel_Click()
' Cancel update
SetControls (False)
'Redisplay details or current record
Displayrecord
End Sub
```

Lúc user clicks Update, bạn có dịp để kiểm tra data xem có field nào bị bỏ trống (nhất là **Primary Key ISBN** bắt buộc phải có trị số) hay có gì không valid bằng cách gọi **Function GoodData**. Nếu GoodData trả lại một trị số False thì ta không xúc tiến với việc Update. Nếu GoodData trả về trị số True thì ta đặt Recordset vào AddNew hay Edit mode tùy theo trị số của Boolean variable AddNewRecord. Giống như khi hiển thị chi tiết của một Record ta phải assign từng Field vào textbox, thì bây giờ khi Update ta phải làm ngược lại, tức là assign property Text của từng textbox vào Record Field tương ứng. Sau cùng ta gọi **method Update** của recordset và cho các controls trở lại Browse mode:

```
Private Function GoodData() As Boolean
' Check Data here. If Invalid Data then GoodData = False
GoodData = True
End Function

Private Sub CmdUpdate_Click()
' Verify all data, if Bad then do not Update
If Not GoodData Then Exit Sub
' Assign record fields to the appropriate textboxes
With myRS
If AddNewRecord Then
```

```
.AddNew ' Place Recordset in AddNew Mode
Else
>Edit ' Place Recordset in Edit Mode
End If
' Assign text of txtTitle to field Title
.Fields("Title") = txtTitle.Text
.Fields("[Year Published]") = txtYearPublished.Text
.Fields("ISBN") = txtISBN.Text
.Fields("PubID") = txtPublisherID.Text
' Update data
.Update
End With
' Return controls to Browse Mode
SetControls (False)
End Sub
```

Cũng vì không có Data Binding, nên khi User Delete một record, sau khi di chuyển qua record kế tiếp ta phải tự hiển thị chi tiết của record đó như sau:

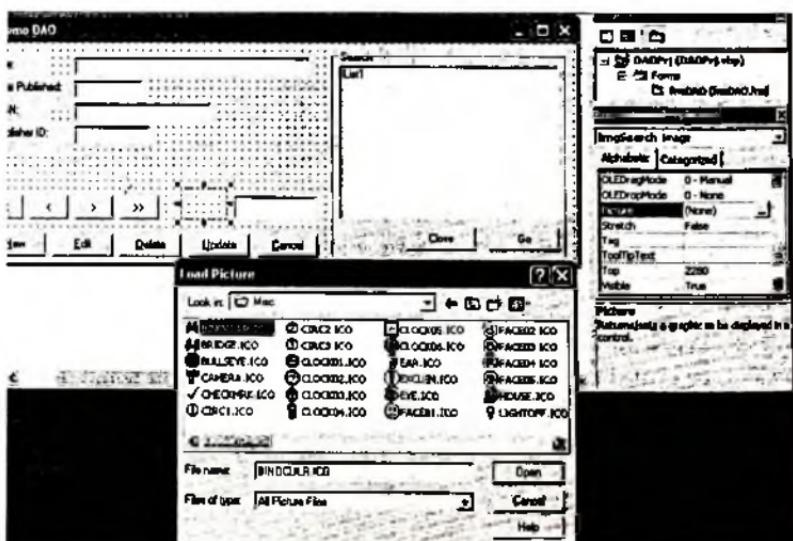
```
Private Sub CmdDelete_Click()
On Error GoTo DeleteErr
With myRS
'Delete new record
.Delete
'Move to next record
.MoveNext
If .EOF Then .MoveLast
'Display details of current record
Displayrecord
Exit Sub
End With
DeleteErr:
MsgBox Err.Description
```

Exit Sub

End Sub

4.4.5. Tìm một record

Tiếp theo đây, ta muốn liệt kê các sách có tiêu đề chứa một chữ hay câu nào đó, thí dụ như chữ "**Guide**". Kế đó user có thể chọn một sách bằng cách select tiêu đề sách ấy và click nút **Go**. Chương trình sẽ locate (tim ra) record của sách ấy và hiển thị chi tiết của nó. Bây giờ bạn hãy cho vào Form một textbox tên **txtSearch** và một Image tên **ImgSearch**. Kế đó đặt một frame tên **fraSearch** vào Form. Để lên frame này một listbox tên **List1** để hiển thị tiêu đề các sách, và hai buttons tên **CmdClose** và **CmdGo**, với caption Close và Go. Sau khi select một sách trong **List1**, user sẽ click nút **Go** để hiển thị chi tiết sách ấy. Nếu đổi ý, user sẽ click nút **Close** để làm biến mất frame **fraSearch**. Bình thường frame **fraSearch** chỉ hiện ra khi cần, nên lúc đầu hãy set **property Visible** của nó thành False. Ta sẽ cho **ImgSearch** hiển thị hình một ống đòn nên bạn hãy click vào bên phải **property Picture** trong Properties Window để chọn Icon **BINOCULR.ICO** từ folder **E:\Program Files\Microsoft Visual Studio\Common\Graphics\Icons\Misc:**



Cái Primary Key của table **Titles** là **ISBN**. Khi user select một sách ta muốn

bìêt ISBN của sách áy đê locate (định chô) nó trong Recordset myRS. Do đó trong khi thêm tiêu đê của một sách vào List1, ta đồng thời thêm ISBN của sách áy vào một Listbox thứ hai tên List2. Ta chỉ sê dùng List2 sau hâu trường, nên hñay set property Visible của nó thành False. Dưới đây là code đê load tiêu đê sách và ISBN vào các Listboxes:

```
Private Sub ImgSearch_Click()
    ' Show Search Frame
    fraSearch.Visible = True
    Dim SrchRS As DAO.Recordset
    Dim SQLCommand As String
    ' Define SQL statement
    SQLCommand = "Select * from Titles where Title LIKE "" & """ &
    txtSearch & """ & """
    ORDER BY Title"
    ' Fetch all records having Title containing the text pattern given by
    txtSearch
    Set SrchRS = myDB.OpenRecordset(SQLCommand)
    ' If Recordset is not Empty then list the books' titles in List1
    If SrchRS.RecordCount > 0 Then
        List1.Clear ' Clear List1
        ' We use List2 to contain the Primary Key ISBN corresponding to the
        books in List1
        List2.Clear ' Clear List2
        With SrchRS
            ' Iterate through the Recordset until EOF
            Do While Not SrchRS.EOF
                ' Display Title in List1
                List1.AddItem .Fields("Title")
                ' Store corresponding ISBN in List2
                List2.AddItem .Fields("ISBN")
            End While
        End With
    End If
End Sub
```

.MoveNext ' Move to next record in the Recordset

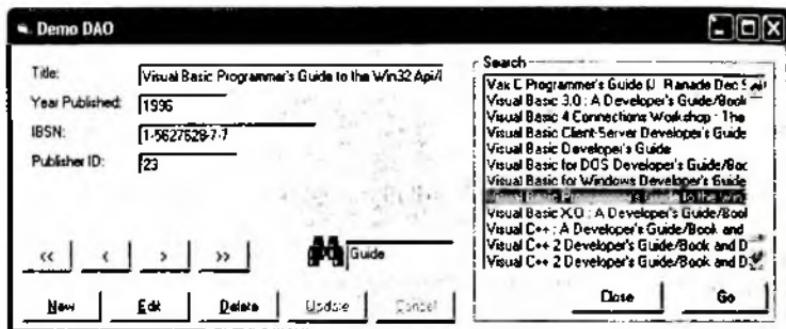
Loop

End With

End If

End Sub

Khi user Click ImgSearch với text pattern là chữ **Guide**, ta sẽ thấy hình dưới đây:



Trong SELECT statement bên trên ta dùng từ khóa **LIKE** trên text pattern, chữ **Guide**, có ký tự đại diện (*) ở hai bên. Wildcard character là chỗ có (hay không có) chữ gì cũng được. Trong trường hợp này có nghĩa là **hỗn** có chữ **Guide** trong tiêu đề sách là được, không cần biết nó nằm ở đâu. Ngoài ra, sự chọn lựa này **Không có CaseSensitive**, tức là chữ **guide**, **Guide** hay **GUIDE** đều được cả. Khi user clicks nút Go, ta sẽ dùng **method FindFirst** của Recordset myRS để định chỗ của record có trị số Primary Key là hàng text trong List2 tương ứng với tiêu đề được chọn trong List1 như sau:

```
Private Sub CmdGo_Click()
Dim SelectedISBN As String
Dim SelectedIndex As Integer
Dim Criteria As String
' Index of line selected by user in List1
SelectedIndex = List1.ListIndex
' Obtain corresponding ISBN in List2
```

```
SelectedISBN = List2.List(SelectedIndex)
' Define Search criteria - use single quotes for selected text
Criteria = "ISBN = " & SelectedISBN & ""
' Locate the record, it will become the current record
myRS.FindFirst Criteria
' Display details of current record
Displayrecord
' Make fraSearch disappeared
fraSearch.Visible = False
End Sub
```

Lưu ý là trong chuỗi điều kiện tìm kiếm, vì ISBN thuộc loại text, chứ không phải là một con số, nên ta phải đặt nó giữa hai dấu ngoặc đơn.

Bookmark

Khi di chuyển từ record này đến record khác trong Recordset, đôi khi ta muốn đánh dấu vị trí của một record để có dịp sẽ trở lại. Ta có thể thực hiện điều ấy bằng cách ghi nhớ **Bookmark** của Recordset. Thí dụ khi user clicks nút Go, ta muốn nhớ vị trí của record lúc ấy để sau này quay trở lại khi User clicks nút **Go Back**. Bạn hãy thêm vào Form một button tên **CmdGoBack** với Caption **Go Back**. Ta sẽ thêm một variable tên **LastBookmark** loại data type **Variant**:

Dim LastBookMark As Variant

Lúc đầu button **CmdGoBack** invisible, và chỉ trở nên visible sau khi user clicks nút Go.

Ta thêm các hàng codes sau vào **Sub CmdGo_Click()** như sau:

' Remember location of current record

LastBookMark = myRS.BookMark

CmdGoback.Visible = True

Dưới đây là code để quay trở lại vị trí current record trước đây trong Recordset:

Private Sub CmdGoback_Click()

' Reposition record to last position

myRS.BookMark = LastBookMark

```
' Redisplay details or current record
```

```
Displayrecord
```

```
End Sub
```

LastModified

LastModified là vị trí của record vừa mới được sửa đổi hay thêm vào trong Recordset. Để thử điều này bạn hãy thêm một button invisible tên **CmdLastModified** với caption là **Last Modified**. Button này chỉ hiện ra sau khi user clicks Update. Bất cứ lúc nào bạn Click nút **CmdLastModified**, record mới vừa được sửa đổi hay thêm vào sẽ hiển thị:

```
Private Sub CmdLastModified_Click()
```

```
' Reposition record to last position
```

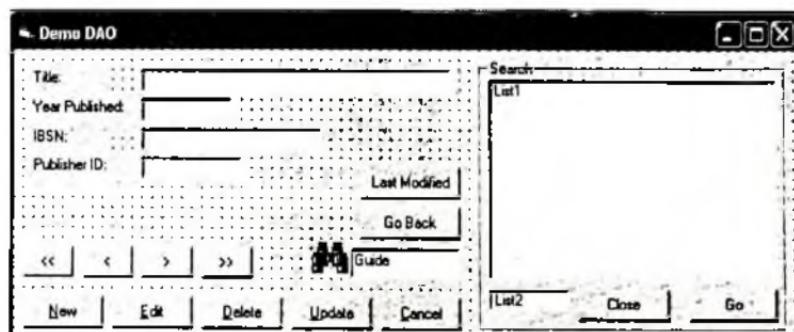
```
myRS.BookMark = myRS.LastModified
```

```
' Redisplay details or current record
```

```
Displayrecord
```

```
End Sub
```

Dưới đây là hình của Form lúc đang được thiết kế:

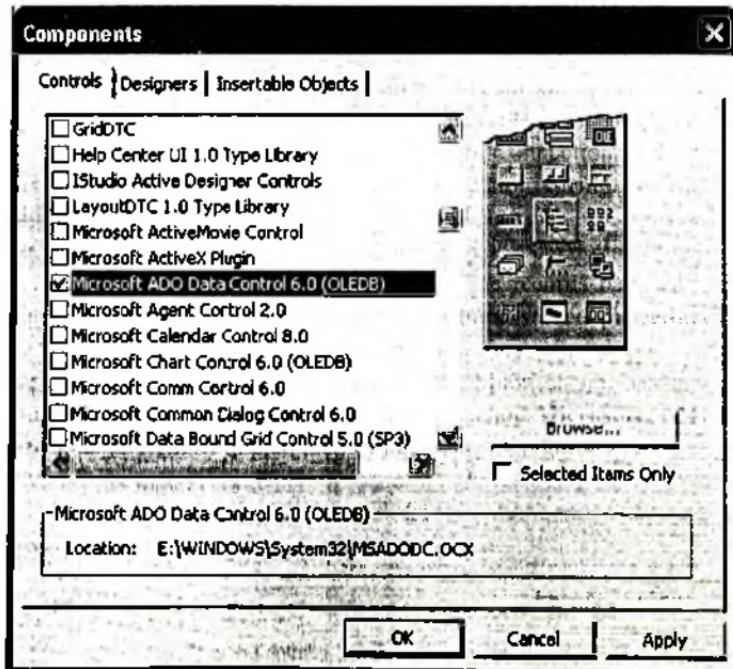


4.5. Sử dụng điều khiển ADO Data để liên kết và khai thác cơ sở dữ liệu Access

4.5.1. Khai báo sử dụng ADO

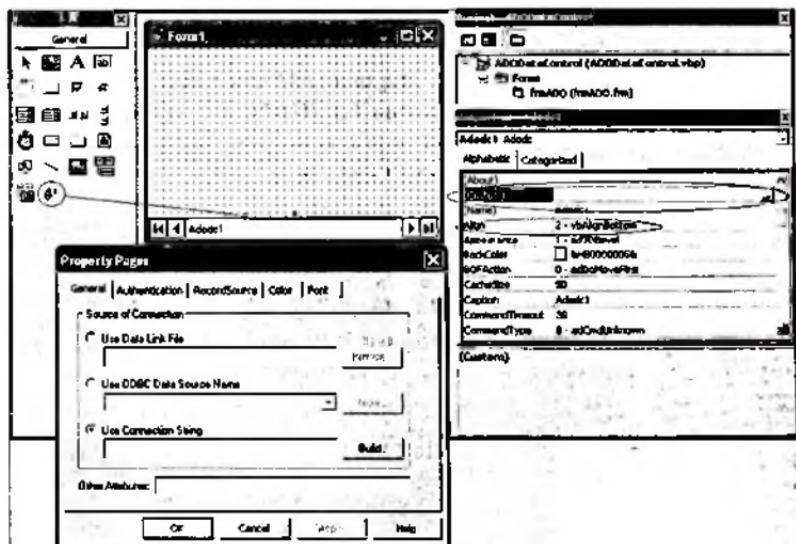
Visual Basic 6 cho ta sự lựa chọn về kỹ thuật khi lập trình với database, hoặc là dùng **DAO** như trong hai bài trước, hoặc là dùng **ADO (ActiveX**

Data Objects). Sự khác biệt chính giữa ADO và DAO là ADO cho phép ta làm việc với mọi loại nguồn dữ kiện (data sources), không nhất thiết phải là Access database hay ODBC. Nguồn dữ kiện có thể là danh sách các địa chỉ Email, hay một file text string, trong đó mỗi hàng là một record gồm những fields ngăn cách bởi các dấu phẩy (**comma separated values**). Nếu trong DAO ta dùng thẳng tên của MSAccess Database thì trong ADO cho ta **nối với (connect)** một database qua một Connection bằng cách chỉ định một **Connection String**. Trong Connection String có **Database Provider** (thí dụ như Jet, ISAM, Oracle, SQLServer..v.v.), tên Database, **UserName/Password** để logon một database ..v.v.. Sau đó ta có thể **lấy về (extract)** những recordsets, và cập nhật hóa các records bằng cách dùng những **lệnh SQL** trên các tables hay dùng những **stored procedures** bên trong database. Bình thường, khi ta mới khởi động một project VB6 mới, Control **Data ADO** không có sẵn trong IDE. Muốn có nó, bạn hãy dùng Menu **Command Project | Components...**, rồi chọn **Microsoft ADO Data Control 6.0 (OLEDB)** từ giao diện Components như dưới đây:



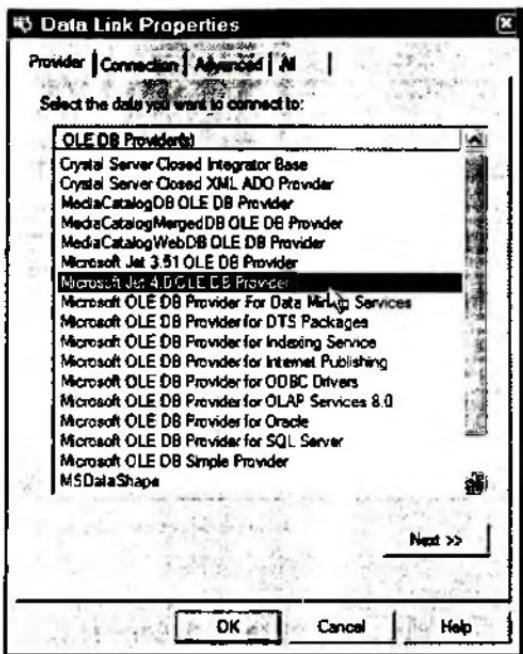
Tương tự với điều khiển Data như phần trước, bạn hãy bắt đầu một dự án VB6 mới, cho nó tên **ADODatControl** bằng cách click tên project trong Project Explorer bên phải rồi edit property Name trong Properties Window. Sửa tên của form chính thành **frmADO**, và đánh dấu **ADO DataControl Demo** vào Caption của nó. DoubleClick lên Icon của Control Data ADO trong Toolbox. Một Control Data ADO tên **Adoctl** sẽ hiện ra trên Form. Muốn cho nó nằm bên dưới Form, giống như một StatusBar, hãy set property Align của nó trong Properties Window thành 2 - **vbAlignBottom**. Click bên phải hàng property (**Custom**), kế đó click lên nút browse có ba chấm để giao thoại **Property Pages** hiện ra. Trong giao thoại này, trên Tab **General** chọn **Radio (Option) Button**

Use Connection String rồi click nút **Build...**.

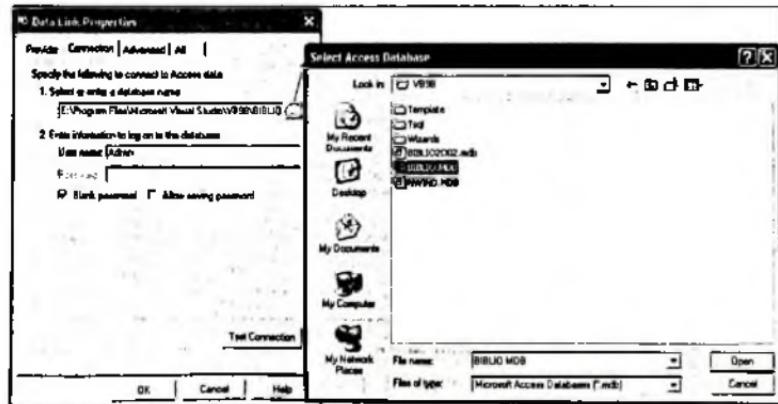


Trong giao thoại **Data Link Properties**, Tab **Provider**, chọn **Microsoft Jet 4.0 OLE DBProvider**, rồi click nút **Next >>** hay **Tab Connection**.

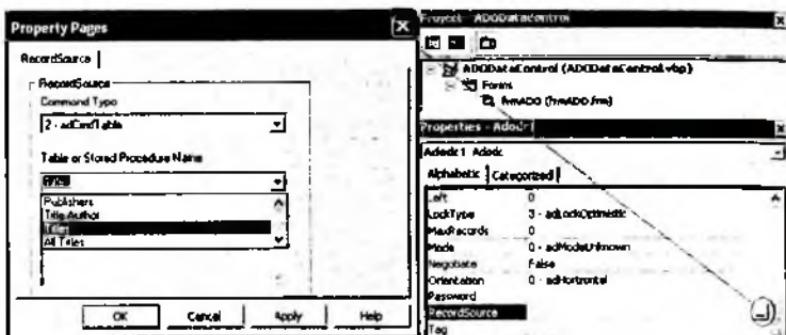
Chú ý: Bạn không nên chọn **Microsoft Jet 3.51 OLE DBProvider** bởi vì như vậy bạn sẽ chỉ có thể kết nối với CSDL Access97 trở về trước.



Ở chỗ Select or enter a database name ta chọn E:\Program Files\Microsoft Visual Studio\VB98\BIBLIO.MDB, trong computer của bạn có thể file ấy nằm trên disk C hay D. Sau đó, bạn có thể click nút Test Connection phía dưới để thử xem connection có được thiết lập tốt không.

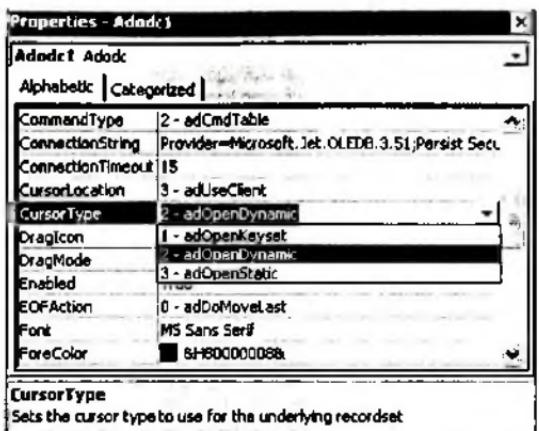


Lập connection xong rồi, ta chỉ định muốn lấy gì về làm Recordset bằng cách click property Recordsource của Adodc1. Trong giao diện Property Pages của nó chọn 2-adCmdTable làm Command Type, kế đó mở Combo box cho Table or Stored Procedure Name để chọn table Titles.



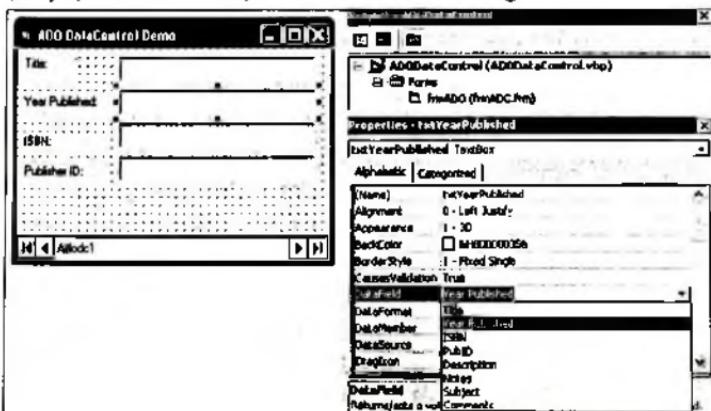
Tùy theo cách ta dùng Recordset trong ADO, nó có ba loại và được gọi là Cursor Type. Cursor chẳng qua là một tên khác của Recordset:

- Static Cursor: Static Cursor cho bạn một static copy (bản sao cứng ngắc) của các records. Trong lúc bạn dùng Static Cursor, nếu có ai khác sửa đổi hay thêm, bớt gì vào recordset bạn sẽ không thấy.
- Keyset Cursor: Keyset Cursor hon Static Cursor ở chỗ trong lúc bạn dùng nó, nếu có ai sửa đổi record nào bạn sẽ biết. Nếu ai delete record nào, bạn sẽ không thấy nó nữa. Tuy nhiên bạn sẽ không biết nếu có ai thêm một record nào vào recordset.
- Dynamic Cursor: Như chữ sống động (dynamic) hàm ý, trong lúc bạn đang dùng một Dynamic Cursor, nếu có ai khác sửa đổi hay thêm, bớt gì vào recordset bạn sẽ thấy hết.
- Bạn hãy chọn trị số 2-adOpenDynamic cho property Cursor Type của Adodc1:

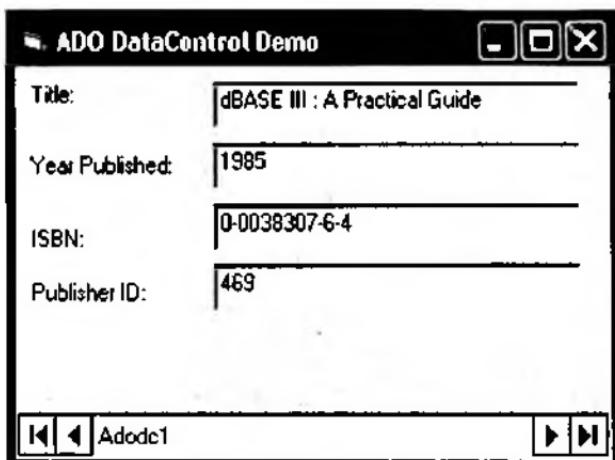


Bây giờ bạn hãy đặt lên Form 4 labels với captions: Title, Year Published, ISBN và Publisher ID. Kế đó cho thêm 4 textboxes tương ứng và đặt tên chúng là txtTitle, txtYearPublished, txtISBN và txtPublisherID.

Để thực hiện Data Binding, bạn hãy chọn textbox txtYearPublished (năm xuất bản), rồi set property Datasource của nó trong Properties Window thành Adodc1. Khi click lên property DataField của txtYearPublished và mở ComboBox ra bạn sẽ thấy liệt kê tên các Fields trong table Titles. Đó là vì Adodc1 được coi như trung gian lấy table Titles từ database. Ở đây ta sẽ chọn cột Year Published. Lặp lại công tác này cho 3 textboxes kia, và chọn các cột Title (Tiêu đề), ISBN (số lý lịch trong thư viện quốc tế), và PubID (số lý lịch nhà xuất bản) làm DataField cho chúng.



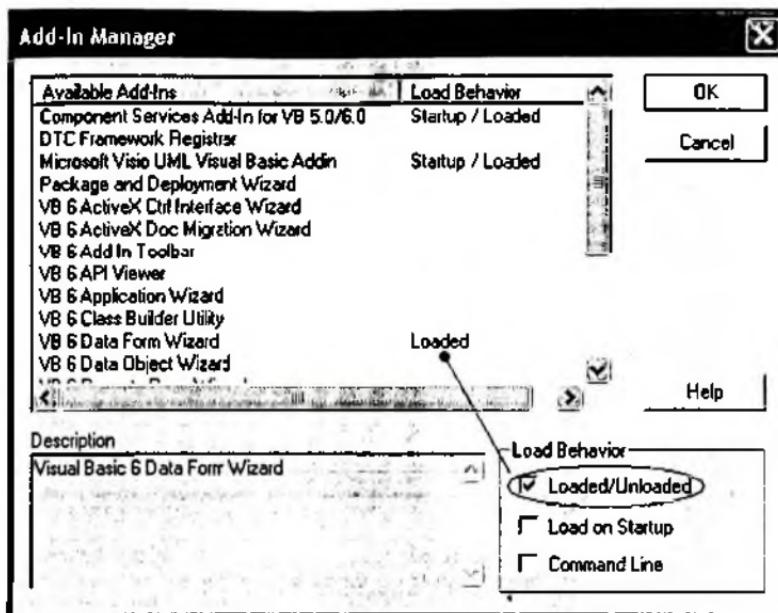
Đến đây, mặc dù chưa viết một hàng code nào, bạn có thể chạy chương trình và nó sẽ hiển thị như dưới đây:



4.5.2. Data Form Wizard

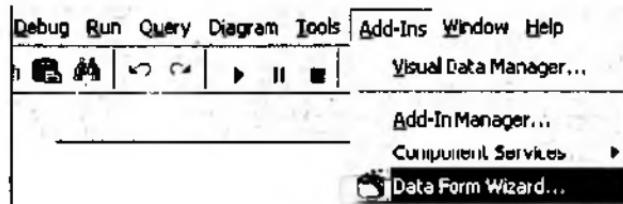
Để giúp lập trình viên thiết kế các data forms nhanh hơn, VB6 cho ta Data Form Wizard để generate (phát sinh) ra một form có hỗ trợ Edit, Add và Delete records. Bây giờ bạn hãy khởi động một standard project VB6 mới, tên ADOClass và copy MS Access file BIBLIO.MDB, tức là database, vào trong cùng folder của dự án mới này.

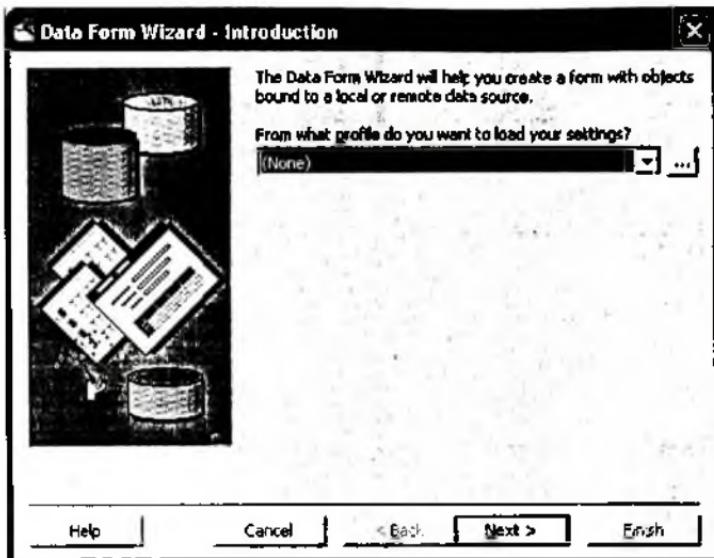
Muốn dùng Data Form Wizard, trước hết ta phải thêm nó vào môi trường phát triển (IDE) của VB6. Bạn hãy dùng IDE Menu Command Add-Ins | Add-In Manager.... Chọn VB6 DataForm Wizard trong giao thoại, rồi click Checkbox Loaded/Unloaded để chữ Loaded hiện bên phải hàng "VB6 Data Form Wizard" như trong hình dưới đây:



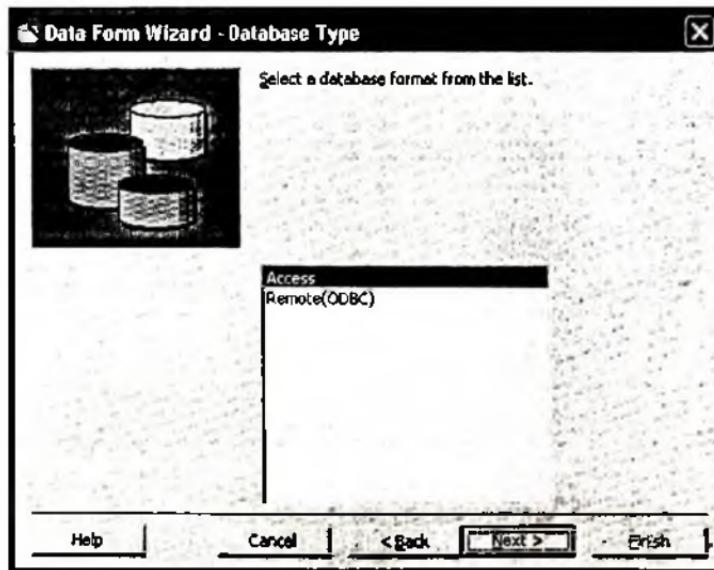
Nếu bạn muốn mỗi lần khởi động VB6 IDE là có sẵn Data Form Wizard trong menu Add-Ins thì ngoài option Loaded, bạn click thêm check box Load on Startup. Một Add-In là một menu Item mới mà ta có thể thêm vào một chương trình ứng dụng có sẵn. Thường thường, người ta dùng Add-Ins để thêm chức năng cho một chương trình, làm như là chương trình đã có sẵn chức năng ấy từ đầu. Bạn hãy khởi động Data Form Wizard từ IDE.

Menu Command mới Add-Ins | Data Form Wizard...

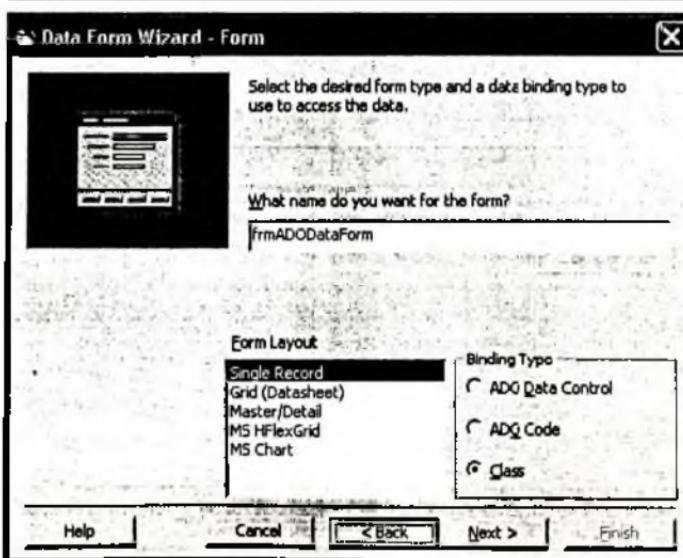
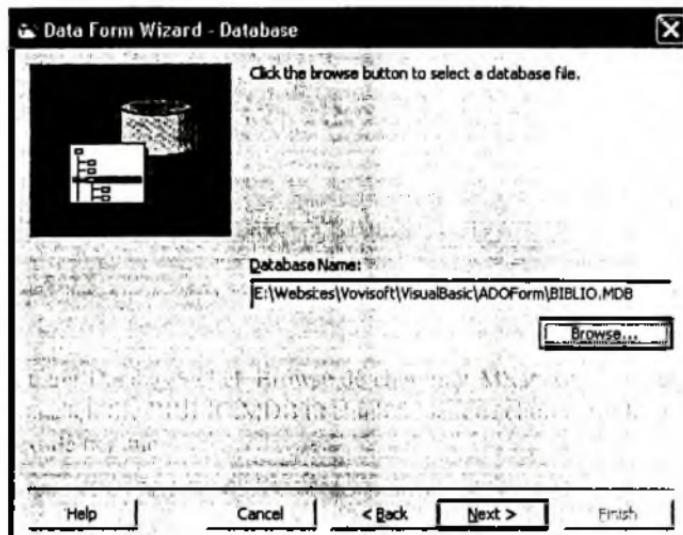




Trong trang kê đó chọn Access làm Database Type.

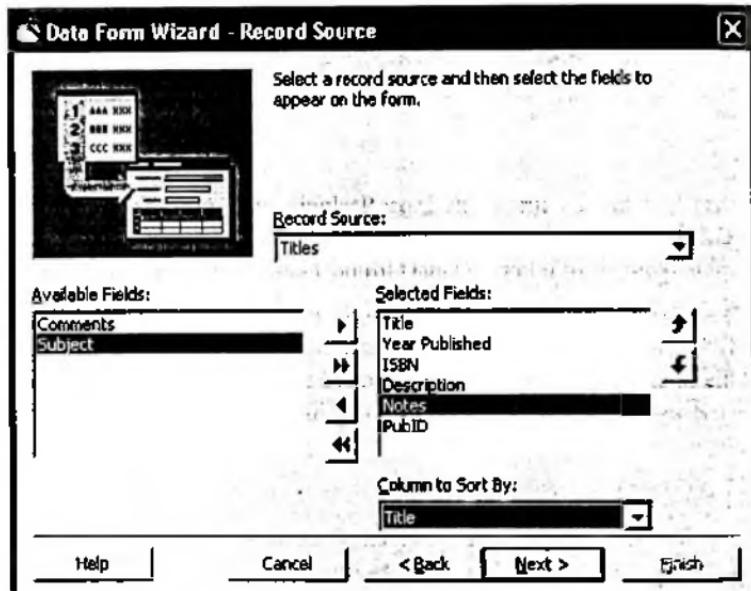


Trong trang Database, click Browse để chọn một MS Access database file. Ở đây ta chọn file BIBLIO.MDB từ chính folder của chương trình này. Bấm nút Next để tiếp tục.



Trong trang Form, ta chọn Single Record cho Form Layout và Class cho Binding Type. Đoạn click Next. Nếu ta chọn ADO Data Control thì kết quả sẽ giống giống như khi ta dùng Control Data DAO như trong một bài trước.

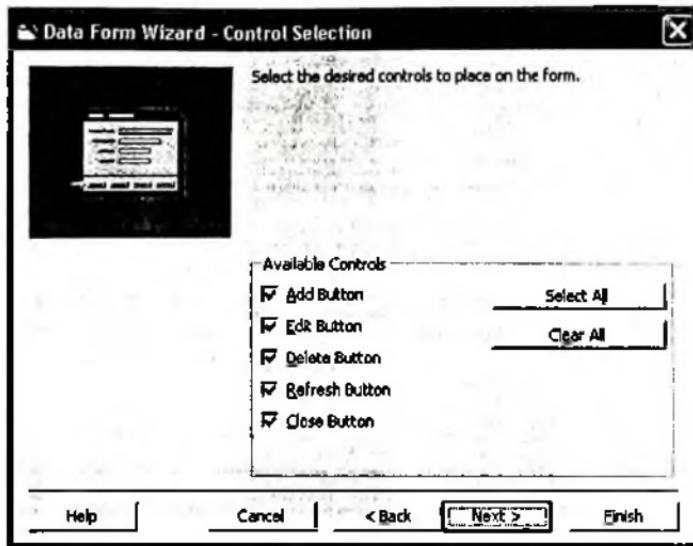
Trong trang Record Source ta chọn table Titles. Listbox của Available Fields sẽ hiển thị các fields của table Titles. Sau khi chọn một field bằng cách click lên tên field ấy trong Listbox, nếu ban click hình tam giác chỉ qua phải thì tên field ấy sẽ được dời qua nằm dưới cùng trong Listbox Selected Fields bên phải.



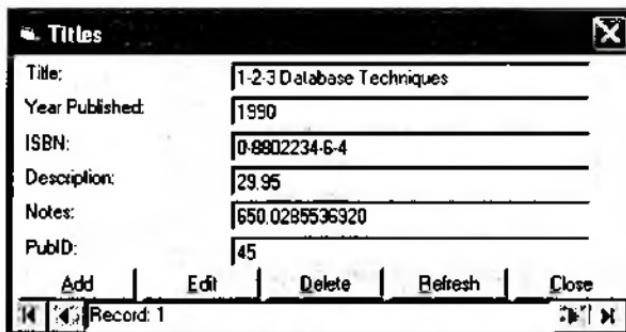
Nếu bạn click hình hai tam giác chỉ qua bên phải thì tất cả mọi fields còn lại bên trái sẽ được dời qua bên phải. Bạn cũng có thể sắp đặt vị trí của các selected fields bằng cách click lên tên field ấy rồi click hình mũi tên chỉ lên hay xuống để di chuyển field ấy lên hay xuống trong danh sách các fields.

Ngoài ra, bạn hãy chọn Title làm Column to Sort By trong cái Combobox của nó để các records trong Recordset được sắp xếp theo thứ tự ABC (alphabetical order) của field Tiêu đề (Title).

Trong trang Control Selection, ta sẽ để ý nguyên để có đủ mọi buttons. Bạn hãy click Next.



Khi Data Form Wizard chấm dứt, nó sẽ generate form frmADODataForm. Bạn hãy remove Form1 và dùng Menu Command Project | ADODataControl Properties... để đổi Startup Object thành frmADODataForm.



Chúng ta hãy quan sát Form và phần code được Data Form Wizard generated. Trong frmADODataForm, các textboxes làm thành một array tên txtFields. Mọi textbox đều có property DataField định sẵn tên field của table Titles. Thí dụ như txtFields(2) có DataField là ISBN. Form chính không dùng Control Data ADO nhưng dùng một Object của class clsTitles. Phần Initialisation của class clsTitles là Open một Connection và lấy về một

Dataset có tên DataMember là Primary như sau:

```
Private Sub Class_Initialize()
Dim db As Connection
Set db = New Connection
db.CursorLocation = adUseClient
' Open connection
db.Open "PROVIDER=Microsoft.Jet.OLEDB.3.51;Data
Source=E:\VisualBasic\ADOForm\BIBLIO.MDB;"
' Instantiate ADO recordset
Set adoPrimaryRS = New Recordset
' Retrieve data for Recordset
adoPrimaryRS.Open "select Title,[Year
Published],ISBN,Description,Notes,PubID from
Titles Order by Title", _
db, adOpenStatic, adLockOptimistic
' Define the only data member, named Primary
DataMembers.Add "Primary"
End Sub
```

Về vị trí của database, nếu bạn không muốn nó chết cứng ở một folder nào thì dùng App.Path để xác định mối liên hệ giữa vị trí của database và folder của chính chương trình đang chạy, thí dụ như:

```
db.Open "PROVIDER=Microsoft.Jet.OLEDB.3.6;Data Source=" & App.Path
& "\BIBLIO.MDB;"
```

Trong Sub Form_Load, ta có thể dùng For Each để đi qua hết các textboxes trong array txtFields. Vì property Datasource của textbox là một Object nên ta dùng keyword Set để chỉ nó đến Object PrimaryCLS. Đồng thời, ta cũng phải chỉ định tên của DataMember của mỗi textbox là Primary:

```
Private Sub Form_Load()
' Instantiate an Object of class clsTitles
Set PrimaryCLS = New clsTitles
Dim oText As TextBox
' Iterate through each textbox in the array txtFields
```

```
'Bind the text boxes to the data source, i.e. PrimaryCLS
For Each oText In Me.txtFields
oText.DataMember = "Primary"
' Use Set because property Datasource is an Object
Set oText.DataSource = PrimaryCLS
Next
End Sub
```

Khi sự di chuyển từ record này đến record khác chấm dứt, chính Recordset có raise Event MoveComplete. Event này được handled (giải quyết) trong class clsTitles bằng cách lại raiseEvent MoveComplete để nó được handled trong Form. Muốn handle Event trong clsTitles ta phải declare recordset adoPrimaryRS với WithEvents:

```
Dim WithEvents adoPrimaryRS As Recordset
Private Sub adoPrimaryRS_MoveComplete(ByVal adReason As ADODB.EventReasonEnum,
    ByVal pError As ADODB.Error, adStatus As ADODB.EventStatusEnum,
    ByVal pRecordset
    As ADODB.Recordset)
    ' Raise event to be handled by main form
    RaiseEvent MoveComplete
End Sub
```

Và trong Form ta cũng phải declare (object clsTitles) PrimaryCLS với WithEvents:

```
Private WithEvents PrimaryCLS As clsTitles
Trong Form, Event MoveComplete sẽ làm hiển thị vị trí tuyệt đối (Absolute Position) của record bằng code dưới đây:
```

```
Private Sub PrimaryCLS_MoveComplete()
    'This will display the current record position for this recordset
    lblStatus.Caption = "Record: " & CStr(PrimaryCLS.AbsolutePosition)
End Sub
```

Khi user clicks Refresh, các textboxs sẽ được hiển thị lại với chi tiết mới

nhất của record từ trong recordset, nhỡ khi có ai khác đã sửa đổi record. Method Requery của clsTitles lại gọi method Requery của Recordset như sau:

```
Private Sub cmdRefresh_Click()
    'This is only needed for multi user applications
    On Error GoTo RefreshErr
    ' fetch the latest copy of Recordset
    PrimaryCLS.Requery
    Exit Sub
    RefreshErr:
    MsgBox Err.Description
    End Sub
```

In Class clsTitles

```
Public Sub Requery()
    ' Fetch latest copy of record
    adoPrimaryRS.Requery
    DataMemberChanged "Primary"
End Sub
```

4.6. Lập trình sử dụng thư viện ADO

4.6.1. Khai báo biến liên quan

ADO có hai đối tượng chính là Connection và RecordSet. Connection dùng để kết nối cơ sở dữ liệu và thực thi câu lệnh SQL dạng hành động. Đối tượng RecordSet cho phép đọc tập dữ liệu từ cơ sở dữ liệu vào đối tượng này, sau đó trình bày trên các điều khiển.

Có hai cách để khởi tạo đối tượng ADO

- Cách thức nhất: sử dụng hàm CreateObject
 - Cách thứ hai là khai báo và khởi tạo đối tượng ADO bằng từ khóa New
- ❖ Cú pháp khai báo và khởi tạo đối tượng ADO sử dụng hàm CreateObject:
 - Dim Cn, Rs
 - Set Cn = CreateObject("ADODB.Connection")

- Set Rs = CreateObject("ADODB.RecordSet")

Trong đó:

- cn và rs có kiểu là Variant
- từ khóa Set và hàm CreateObject để khởi tạo hai đối tượng Connection và RecordSet.

❖ Khai báo và khởi tạo đối tượng ADO sử dụng từ khóa New:

- Tham chiếu đến thư viện Microsoft ActiveX Data Object 2.5:
Project \ References \ Microsoft ActiveX Data Object 2.5.
- Cú pháp:

```
Dim Cn As New ADODB.Connection  
Dim Rs As New ADODB.RecordSet
```

4.6.2. Thiết lập kết nối với CSDL

Có hai phương pháp kết nối cơ sở dữ liệu:

- Cách 1:

```
Cn.Open "Chuỗi kết nối"
```

- Cách 2:

```
Cn.ConnectionString= "Chuỗi kết nối"
```

```
Cn.Open
```

❖ Cú pháp “Chuỗi kết nối” cơ sở dữ liệu Access 2000:

```
Provider = Microsoft.Jet.OLEDB.4.0;
```

```
Persist Security Info=False;
```

```
Data Source = PathFile
```

- Trong đó:

- Provider = Microsoft.Jet.OLEDB.4.0: phương thức kết nối.

- PathFile là tên tập tin cơ sở dữ liệu và đường dẫn của tập tin đó.

❖ Cú pháp “Chuỗi kết nối” cơ sở dữ liệu SQL Server 7.0, SQL Server 2000:

```
Provider = SQLOLEDB.1;
```

```
Persist Security Info=False;
```

```
UID = sa; PWD = sa;
```

- Initial Catalog = databasename;**
- Server = servername; port = 1433;**
- Trong đó:
 - Provider = SQLOLEDB.1: phương thức kết nối.
 - UID = sa; PWD = sa: tài khoản đăng nhập.
 - Initial Catalog = databasename: là tên cơ sở dữ liệu.
 - Server = servername tên server nơi cài đặt SQL Server.
 - port = 1433 là cổng truy cập cơ sở dữ liệu mặc định.
- Để mở một bảng dữ liệu có tên là tên Table, Query, View hay chuỗi SQL dạng Select. Ta dùng cú pháp sau:
Rs.Open “Table/Query/View>Select”, Cn
Trong đó:
Đối tượng Cn đã được kết nối với cơ sở dữ liệu.
- Một số thuộc tính chính của đối tượng RecordSet:
 - EOF: trả về giá trị là True nếu con trỏ ở sau mẫu tin cuối cùng.
 - BOF: trả về giá trị là True nếu con trỏ ở trên mẫu tin đầu tiên.
 - Fields: để gán hoặc lấy giá trị trường FieldName của mẫu tin theo cú pháp:
 - Rs(“FieldName”) = Value
 - X= Rs(“FieldName”)
 - RecordCount: trả về giá trị là số mẫu tin đang có trong đối tượng RecordSet
- Một số phương thức chính của đối tượng RecordSet:
 - MoveFirst, MoveLast, MoveNext, MovePrevious: các phương thức này để di chuyển con trỏ giữa các mẫu tin.
 - AddNew, Update, Delete: các phương thức này để thêm, cập nhật và xóa mẫu tin.

Đóng kết nối ADO ta dùng cú pháp sau

Rs.Close

Cn.Close

Ghi chú:

- Có thể mở lại tập dữ liệu với câu lệnh SQL hay tên đối tượng

Table, View, Query.

- Có thể kết nối cơ sở dữ liệu trở lại với chuỗi kết nối khác trong ConnectionString.

4.6.3. Lập trình thêm dữ liệu vào bảng sử dụng ADO

Có hai cách thêm dữ liệu:

- Cách thứ nhất: sử dụng phương thức Execute của đối tượng Connection kết hợp với phát biểu Insert của SQL.
- Cách thứ hai sử dụng phương thức AddNew và Update của đối tượng RecordSet.

❖ Phương thức Execute của đối tượng Cn:

```
Dim Cn As New ADODB.Connection  
Dim strCn As String      'Chuỗi kết nối cơ sở dữ liệu  
Dim strSQL As String     'Chuỗi phát biểu SQL  
Dim intRecords As Integer' Số bản ghi được thêm  
strCn = "Provider = ..."  
strSQL = "INSERT INTO ... VALUES ..."  
'Kết nối cơ sở dữ liệu  
Cn.Open strCn  
Cn.Execute strSQL, intRecords' Thêm dữ liệu  
Cn.Close  
Set Cn = Nothing
```

❖ Phương thức AddNew và Update của Rs:

```
Dim Cn As New ADODB.Connection  
Dim Rs As New ADODB.RecordSet  
Dim strCn As String      'Chuỗi kết nối cơ sở dữ liệu  
Dim strSQL As String     'Chuỗi phát biểu SQL  
strCn = "Provider = ..."  
strSQL = "SELECT * FROM ..."  
'Mở kết nối  
Cn.Open strCn  
'Mở cơ sở dữ liệu
```

```
Rs.Open strSQL, Cn, adOpenDynamic, adLockOptimistic  
'Thêm mới mẫu tin  
Rs.AddNew  
Rs("FieldName") = Value  
...  
'Cập nhật mẫu tin  
Rs.Update  
Rs.Close  
Cn.Close  
Set Rs = Nothing  
Set Cn = Nothing
```

4.6.4. Lập trình hiển thị các bản ghi trong bảng CSDL

```
Dim Cn As New ADODB.Connection  
Dim Rs As New ADODB.RecordSet  
Dim strCn As String      'Chuỗi kết nối cơ sở dữ liệu  
Dim strSQL As String     'Chuỗi phát biểu SQL  
strCn = "Provider = ..."  
strSQL = "SELECT * FROM ..."  
'Mở kết nối  
Cn.Open strCn  
'Mở cơ sở dữ liệu  
Rs.Open strSQL, Cn, adOpenDynamic, adLockOptimistic  
'Liệt kê mẫu tin  
If NOT Rs.EOF Then  
    Variable = Rs("FieldName")  
    ...  
End If  
Rs.Close  
Cn.Close  
Set Rs = Nothing  
Set Cn = Nothing
```

4.6.5. Cập nhật dữ liệu vào bảng CSDL

- ❖ Có hai cách cập nhật dữ liệu vào cơ sở dữ liệu sau khi đã thay đổi:
 - Cách thứ nhất sử dụng phương thức Execute của đối tượng Connection kết hợp với phát biểu UPDATE của SQL.
 - Cách thứ hai sử dụng phương thức Update của đối tượng RecordSet.
- ❖ Cập nhật dữ liệu sử dụng phương thức Execute:

'Đối tượng kết nối cơ sở dữ liệu

```
Dim Cn As New ADODB.Connection
```

```
Dim strCn As String
```

```
Dim strSQL As String
```

```
Dim intRecords As Integer
```

```
strCn = "Provider = ..."
```

```
strSQL = "UPDATE ... SET ..."
```

'Mở kết nối

```
Cn.Open strCn
```

'Mở cơ sở dữ liệu

```
Cn.Execute strSQL, intRecords
```

'Liệt kê mẫu tin

```
Cn.Close
```

```
Set Rs = Nothing
```

```
Set Cn = Nothing
```

- ❖ Cập nhật dữ liệu sử dụng phương thức Update:

'Đối tượng kết nối cơ sở dữ liệu

```
Dim Cn As New ADODB.Connection
```

```
Dim Rs As New ADODB.RecordSet
```

```
Dim strCn As String
```

```
Dim strSQL As String
```

```
strCn = "Provider = ..."
```

```
strSQL = "SELECT * FROM ..."
```

'Mở kết nối

```
Cn.Open strCn
'Mở cơ sở dữ liệu
Rs.Open strSQL, Cn, adOpenDynamic, adLockOptimistic
If NOT Rs.EOF Then
Rs("FieldName") = Value

...
Rs.Update
End If
Rs.Close
Cn.Close
Set Rs = Nothing
Set Cn = Nothing
```

4.6.6. Xóa dữ liệu trong CSDL

Có hai cách xóa dữ liệu trên cơ sở dữ liệu:

- Cách thứ nhất sử dụng phương thức Execute của đối tượng Connection kết hợp với phát biểu DELETE của SQL.
- Cách thứ hai sử dụng phương thức Delete của đối tượng RecordSet.

Xóa dữ liệu dùng phương thức Execute:

```
'Đối tượng kết nối cơ sở dữ liệu
Dim Cn As New ADODB.Connection
Dim strCn As String
Dim strSQL As String
Dim intRecords As Integer
strCn = "Provider = ..."
strSQL = "DELETE FROM ..."
Cn.Open strCn
'Mở cơ sở dữ liệu
Cn.Execute strSQL, intRecords
Cn.Close
Set Rs = Nothing
Set Cn = Nothing
```

❖ Xóa dữ liệu dụng phương thức Update:

'Đổi tượng kết nối cơ sở dữ liệu

```
Dim Cn As New ADODB.Connection
Dim Rs As New ADODB.RecordSet
Dim strCn As String
Dim strSQL As String
strCn = "Provider = ..."
strSQL = "SELECT * FROM ..."
'Mở kết nối
Cn.Open strCn
'Mở cơ sở dữ liệu
Rs.Open strSQL, Cn, adOpenDynamic, adLockOptimistic
'Xóa mẫu tin
If NOT Rs.EOF Then
    Rs.Delete
End If
Rs.Close
Cn.Close
Set Rs = Nothing
Set Cn = Nothing
```

4.6.7. Kết hợp ComboBox và Listbox với đối tượng ADO

Tương tự như lập trình với DAO ta có thể nạp danh sách vào điều khiển combobox hoặc Listbox sử dụng cấu trúc sau:

```
'Nạp nội dung vào hộp ComboBox
ComboBox.Clear
Do Until Rs.EOF
    ComboBox.AddItem Rs("FieldName")
    Rs.MoveNext
Loop
'Nạp nội dung vào ListBox
Listbox.Clear
```

```

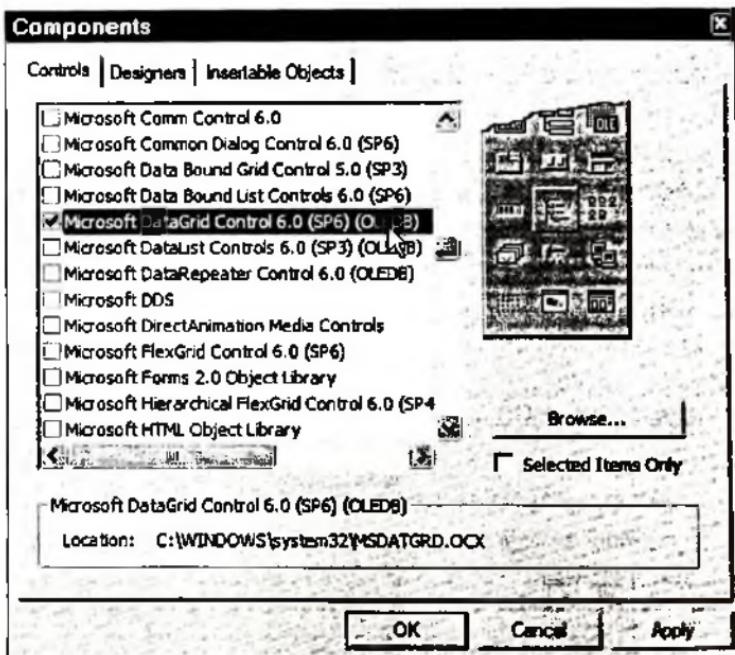
Do Until Rs.EOF
    ListBox.AddItem Rs("FieldName")
    Rs.MoveNext
Loop

```

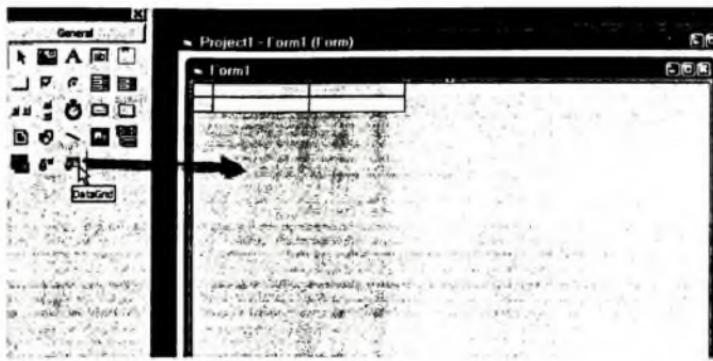
4.6.8. Kết hợp ô lưới DataGrid với ADO

Đối với điều khiển Data, chúng ta đã liên kết nó với ô lưới DBGrid. Tuy nhiên DBGrid lại không thể dùng với ADO mà ô lưới sử dụng với ADO phải là DataGrid.

Khai báo sử dụng bằng cách chọn Microsoft Data Grid control trong hộp thoại Component.



- Lựa chọn và thiết lập kết nối giữa DataGrid với ADO được làm tương tự như khi thiết kế DBGrid kết nối với DAO



Chú ý: Ta có thể truyền dữ liệu vào ô lưới khi ứng dụng hoạt động bằng câu lệnh: `Set DataGrid.DataSource = Res`

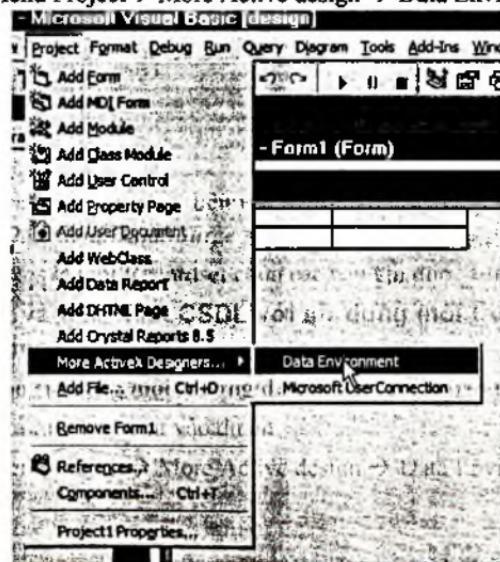
Trong đó Res là một Recordset chứa các bản ghi được truyền vào.

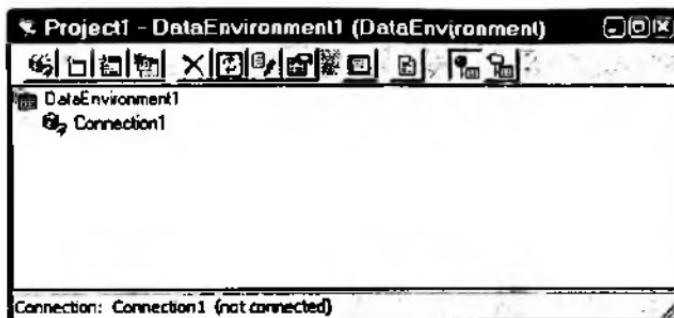
4.7. Liên kết và truy xuất CSDL với sử dụng môi trường dữ liệu Data Environment

4.7.1. Khai báo sử dụng môi trường dữ liệu Data Environment

Để thêm Data Environment vào dự án

- Menu Project → More ActiveX design → Data Environment

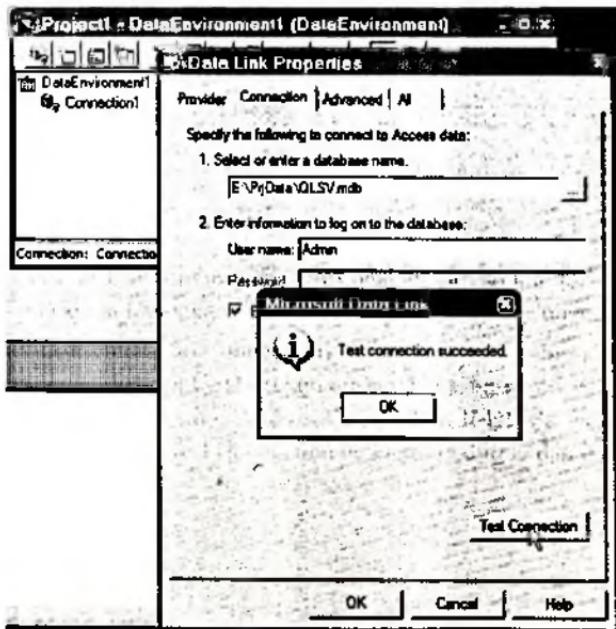




4.7.2. Liên kết với CSDL

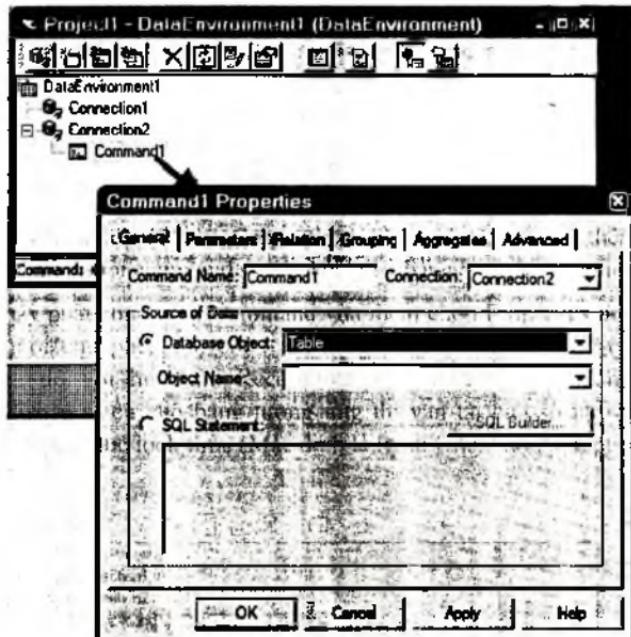
Trong cửa sổ Data Environment: **Ấn phải chuột vào tên đối tượng Data** → Add Connection làm xuất hiện Connection mới (Gọi là Connection1)

- Nháy phải chuột vào Connection1 chọn Properties hiện ra cửa sổ cho phép thiết lập liên kết với CSDL (chọn đường dẫn tới nơi có CSDL cần liên kết). Trong tab Provider chọn Microsoft Jet 4.0 OLE DB Provider → Next
- Chọn đường dẫn. **Ấn test Connection để kiểm tra đường dẫn đúng không.**



- Nháy phải chuột vào Connection1 → Add Command thêm vào 1 command mới (mỗi command sẽ liên kết với 1 bảng hoặc query trong CSDL).

- Nháy phải chuột vào Command vừa thêm chọn Properties làm hiện ra cửa sổ mới cho phép lựa chọn tên bảng thuộc CSDL được liên kết tới (Tương tự đối với thuộc tính RecordSource của ADO). Nếu muốn lập trình để cập nhật dữ liệu cho bảng tương ứng thì vẫn tại cửa sổ này trong thẻ Advance trong mục lock time (Mặc định là Read Only chọn là Optimistic)

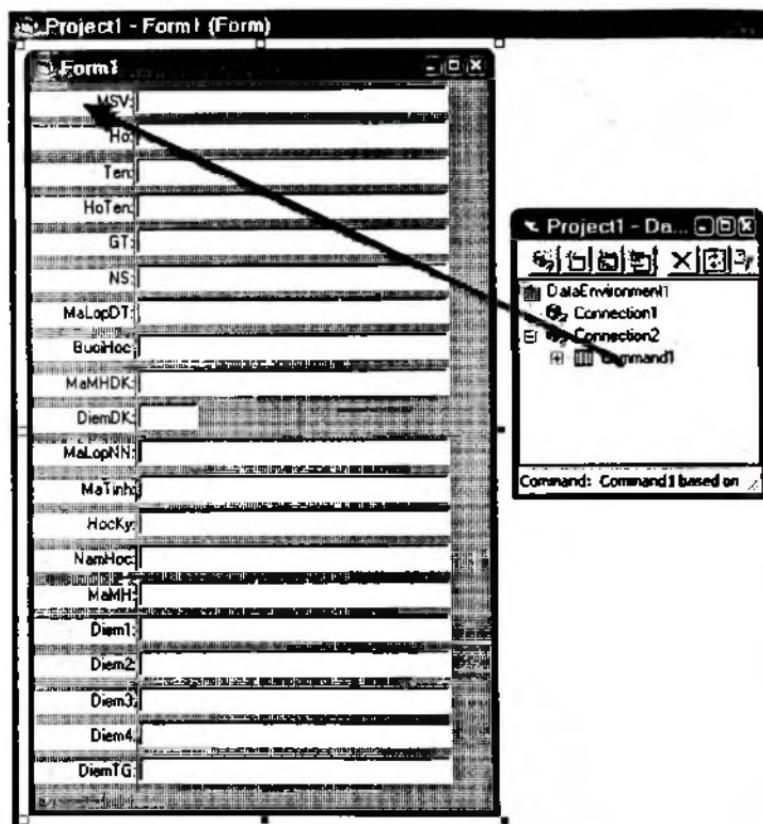


4.7.3. Tạo một giao diện người sử dụng với thiết kế DATAENVIRONMENT

Ta có thể tạo một giao diện người sử dụng nhanh chóng bằng cách dùng thiết kế DataEnvironment. Thiết kế kết hợp với cơ chế biểu mẫu của VB, cho phép ta dùng kỹ thuật kéo và thả để tạo một giao diện người sử dụng điều khiển bằng cơ sở dữ liệu. Để thực hiện điều này, ta theo các bước sau:

1. Mở biểu mẫu ta muốn dùng làm giao diện người sử dụng.
2. Chọn bảng trong thiết kế Data Environment.
3. Thả bảng vào biểu mẫu.

Một giao diện người sử dụng ràng buộc dữ liệu sẽ được tạo trên biểu mẫu.



Tạo giao diện người sử dụng ràng buộc dữ liệu.

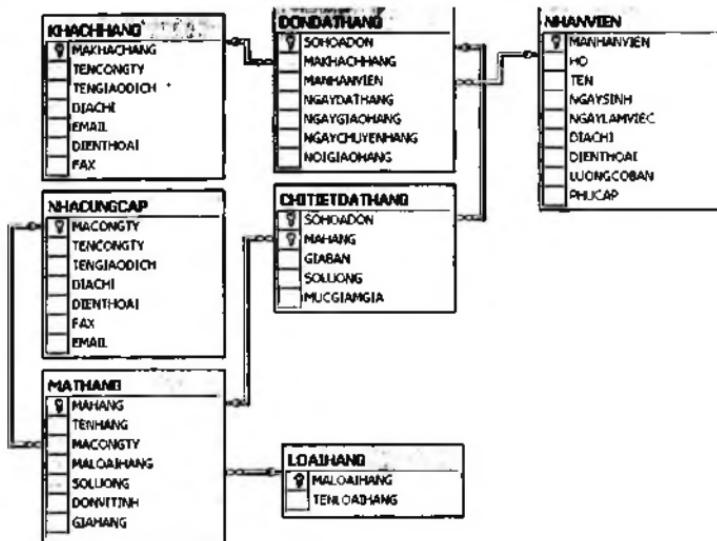
Thi hành ứng dụng để xem mẫu tin thứ nhất trong cơ sở dữ liệu. Tuy nhiên, không có chức năng duyệt từ mẫu tin này sang mẫu tin khác. Để thực hiện điều đó, ta phải lập trình hoặc dùng một điều khiển dữ liệu, mô tả trong phần sau.

Chú ý:

Với môi trường dữ liệu Data Environment ta có thể cùng lúc tạo nhiều liên kết (connection) với nhiều CSDL, trong mỗi liên kết thì lại có thể tạo nhiều Command kết nối với nhiều bảng dữ liệu trong CSDL.

CÂU HỎI ÔN TẬP CHƯƠNG 4

Bài tập tổng hợp 1: Cơ sở dữ liệu dưới đây được sử dụng để quản lý công tác giao hàng trong một công ty kinh doanh. Các bảng trong cơ sở dữ liệu này được biểu diễn trong sơ đồ dưới đây:



Trong đó:

Bảng NHACUNGCAP lưu trữ dữ liệu về các đối tác cung cấp hàng cho công ty.

Bảng MATHANG lưu trữ dữ liệu về các mặt hàng hiện có trong công ty.

Bảng LOAIHANG phân loại các mặt hàng hiện có.

Bảng NHANVIEN có dữ liệu là các thông tin về nhân viên làm việc trong công ty.

Bảng KHACHHANG được sử dụng để lưu giữ thông tin về khách hàng của công ty.

• Khách hàng đặt hàng cho công ty thông qua các đơn đặt hàng. Thông tin chung về các đơn đặt hàng được lưu trữ trong bảng DONDATHANG (Mỗi một đơn hàng phải do một nhân viên của công ty lập và do đó bảng này có quan hệ với bảng NHANVIEN)

Thông tin chi tiết của các đơn đặt hàng (đặt mua mặt hàng gì, số lượng, giá cả,...) được lưu trữ trong bảng CHITIETDATHANG. Bảng này có quan hệ với bảng DONATHANG và MATHANG

Hãy viết câu lệnh SQL để thực hiện những yêu cầu sau đây:

1. Cho biết danh sách các đối tác cung cấp hàng cho công ty?
2. Mã hàng, tên hàng và số lượng của các mặt hàng hiện có trong công ty?
3. Họ tên, địa chỉ và năm bắt đầu làm việc của các nhân viên trong công ty?
4. Địa chỉ và số điện thoại của nhà cung cấp VINAMILK là gì?
5. Cho biết mã và tên của các mặt hàng có giá lớn hơn 100000 và số lượng có ít hơn 50?
6. Cho biết mỗi mặt hàng trong công ty do ai cung cấp?
7. Công ty Việt Tiến đã cung cấp những mặt hàng nào?
8. Loại hàng thực phẩm do những công ty nào cung cấp và địa chỉ các công ty đó là gì?
9. Những khách hàng nào (tên giao dịch) đã đặt mua mặt hàng sữa hộp XYZ của công ty?
10. Đơn đặt hàng số 1 do ai đặt và do nhân viên nào lập, thời gian và địa điểm giao hàng ở đâu?
11. Hãy cho biết số tiền lương mà công ty phải trả cho mỗi nhân viên là bao nhiêu (lương=Lương cơ bản+phụ cấp)?
12. Trong đơn đặt hàng số 3 đặt mua những mặt hàng nào và số tiền mà khách hàng phải trả cho mỗi mặt hàng là bao nhiêu (số tiền phải trả được tính theo công thức SOLUONG×GIABAN - SOLUONG×GIABAN×MUCGIAMGIA/100)?
13. Hãy cho biết có những khách hàng nào lại chính là đối tác cung cấp hàng của công ty (tức là có cùng tên giao dịch)?
14. Trong công ty có những nhân viên nào có cùng ngày sinh?
15. Những đơn đặt hàng nào yêu cầu giao ngay tại công ty đặt hàng và những đơn đó là của công ty nào?
16. Cho biết tên công ty, tên giao dịch, địa chỉ và điện thoại của các khách hàng và các nhà cung cấp hàng cho công ty.
17. Những mặt hàng nào chưa từng được khách hàng đặt mua

18. Những nhân viên nào của công ty chưa từng lập bất kỳ một hóa đơn đặt hàng nào?
19. Những nhân viên nào của công ty có lương cơ bản cao nhất?
20. Tổng số tiền mà khách hàng phải trả cho mỗi đơn đặt hàng là bao nhiêu?
21. Trong năm 2003, những mặt hàng nào chỉ được đặt mua đúng một lần.
22. Hãy cho biết mỗi một khách hàng đã bỏ ra bao nhiêu tiền để mua hàng của công ty?
23. Mỗi một nhân viên công ty đã lập bao nhiêu đơn đặt hàng? (Nếu chưa hề lập một hóa đơn nào thì cho kết quả là 0)
24. Cho biết tổng số tiền hàng mà cửa hàng thu được trong mỗi tháng của năm 2003
25. Cho biết tổng số tiền lời mà công ty thu được từ mỗi mặt hàng trong năm 2003.
26. Hãy cho biết tổng số lượng hàng của mỗi mặt hàng mà công ty đã có (tổng số mặt hàng hiện có và đã bán)
27. Nhân viên nào của công ty bán được số lượng hàng nhiều nhất và là bao nhiêu?
28. Đơn đặt hàng nào có số lượng hàng đặt mua ít nhất?
29. Số tiền nhiều nhất mà mỗi khách hàng đã từng bỏ ra để đặt hàng trong các đơn đặt hàng là bao nhiêu?
30. Thông kê xem trong năm 2003, mỗi một mặt hàng trong mỗi tháng và trong cả năm bán được với số lượng là bao nhiêu? Yêu cầu: Kết quả được hiển thị dưới dạng bảng, hai cột đầu là mã hàng và tên hàng, các cột còn lại tương ứng với các tháng từ 1 đến 12 và cả năm. Như vậy, mỗi dòng trong kết quả cho biết số lượng hàng bán được mỗi tháng và trong cả năm của mỗi mặt hàng.

Bài tập tổng hợp 2

Hãy thiết kế mô hình CSDL cho 1 trung tâm thương mại với các thông tin ràng buộc như sau:

- Trung tâm kinh doanh nhiều loại mặt hàng, mỗi mặt hàng đều được quản lý với các thông tin: tên hàng, nước sản xuất, đơn giá, tỷ lệ thuế. Đồng thời trung tâm cũng có các danh sách các khách hàng của mình, mỗi khách

hàng được quản lý thông tin gồm tên khách hàng, địa chỉ, mã số thuế. Mỗi hoá đơn bán hàng tại trung tâm thì sẽ ghi nhận thông tin về 1 mặt hàng được bán.

Viết lệnh SQL để trả lời những câu hỏi sau:

1. Liệt kê những mặt hàng ngoại nhập hiện được bán tại trung tâm
2. Liệt kê danh sách những mặt hàng do Nhật Bản sản xuất và có đơn giá cao nhất.
3. Liệt kê danh sách của 10 mặt hàng có đơn giá thấp nhất.
4. Tính tổng doanh thu bán hàng của trung tâm trong quý 2 năm 2006
5. Tính tổng doanh thu của việc kinh doanh mặt hàng tivi tại trung tâm năm 2006
6. Liệt kê danh sách những mặt hàng có doanh thu cao nhất trong năm 2006.
7. Đếm tổng số lượng mặt hàng là tủ lạnh đã được bán trong quý I năm 2007.
8. Liệt kê danh sách những mặt hàng do VN sản xuất mà không bán được cái nào trong tháng 9 năm 2006.
9. Tạo bảng tổng hợp doanh thu bán hàng theo từng mặt hàng trong năm 2006.
10. Trung tâm muốn tặng quà cho 3 khách hàng tiêu tiêu tiền nhất trong việc mua hàng tại trung tâm trong nửa đầu năm 2007. Hãy tìm tên của 3 người đó.

Bài tập 3:

- Hãy viết câu lệnh kết nối (connection String) với cơ sở dữ liệu Access có tên là Data.mdb, đặt trong thư mục theo đường dẫn “C:\Database\data.mdb”, cơ sở dữ liệu có đặt mật khẩu là “ktqd”, sử dụng đối tượng dữ liệu Active Data Object.

- Các bước để liên kết các đối tượng cơ bản trên một Form chức năng của VisualBasic với một cơ sở dữ liệu?

Chương 5

TẠO BÁO CÁO ĐẦU RA CHO ỨNG DỤNG

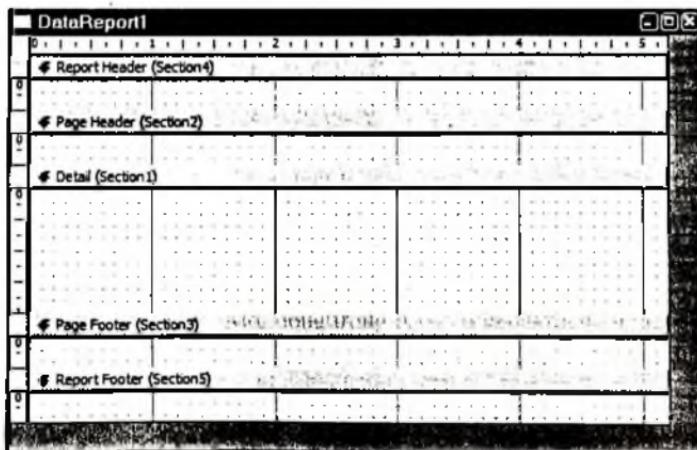
5.1. Giới thiệu chung

Mục đích cuối cùng của mỗi dự án phần mềm đó là tạo ra được các báo cáo đầu ra phục vụ công tác quản lý.

Báo cáo đầu ra có thể được chia thành các loại sau:

- Báo cáo chi tiết đầy đủ: là báo cáo liệt kê nội dung chi tiết của tất cả các bản ghi trong bảng CSDL. Số lượng các dòng trong báo cáo bằng đúng số dòng trong bảng CSDL.
- Báo cáo chi tiết có chọn lọc: là báo cáo liệt kê nội dung chi tiết của các bản ghi trong bảng CSDL thỏa mãn một điều kiện lọc nào đó. Số lượng các dòng trong báo cáo bằng đúng số bản ghi thỏa mãn điều kiện lọc trong bảng CSDL.
- Báo cáo tổng hợp: Mỗi dòng trong báo cáo là đại diện cho một nhóm các bản ghi trong bảng CSDL. Số dòng trong báo cáo bằng đúng số lượng các nhóm bản ghi trong bảng CSDL.

Một bản thiết kế báo cáo bất kỳ thì luôn có thể được chia thành các phần như sau:



- ReportHeader_Phần đầu báo cáo. Là nơi ta sẽ đặt các thiết lập mà nó sẽ chỉ hiển thị một lần tại phần đầu của trang đầu tiên của báo cáo.
- PageHeader_Phần tiêu đề trang báo cáo. Là nơi ta sẽ đặt các thiết lập mà nó sẽ hiển thị lặp lại tại đầu mỗi trang báo cáo.
- Detail_Phần chi tiết báo cáo. Là nơi ta sẽ đặt các thiết lập để hiển thị nội dung dữ liệu được truyền vào báo cáo.
- PageFooter_Phần cuối trang báo cáo. Là nơi ta sẽ đặt các thiết lập mà nó sẽ hiển thị lặp lại tại cuối mỗi trang báo cáo.
- ReportFooter_Phần cuối báo cáo. Là nơi ta sẽ đặt các thiết lập mà nó sẽ chỉ hiển thị một lần tại phần cuối cùng của trang báo cáo cuối cùng.

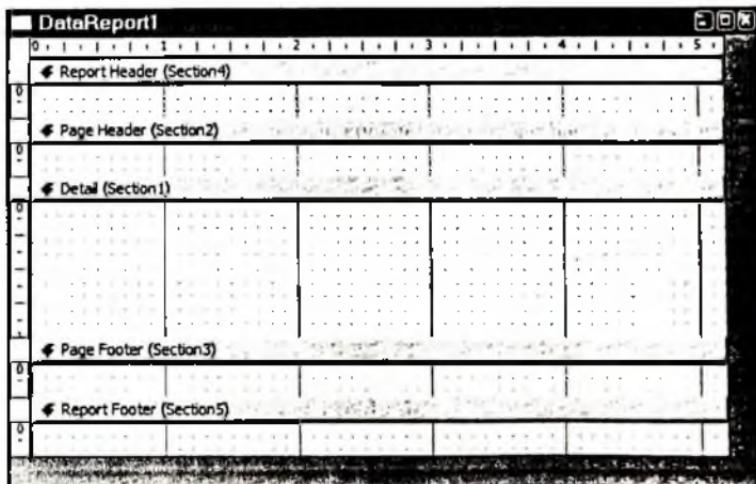
Tùy theo yêu cầu của công tác quản lý mà chúng ta sẽ thiết kế ra các mẫu báo cáo cho phù hợp.

Trong cuốn sách này chúng ta sẽ học cách tạo báo cáo bằng 2 công cụ là DataReport có sẵn trong Vb và CrystalReport, một công cụ tạo báo cáo chuyên nghiệp, độc lập.

5.2. Tạo báo cáo với công cụ DataReport

5.2.1. Thêm một Data Report vào dự án

Project → Add Data Report → 1 mẫu báo cáo mới được thêm vào dự án:



5.2.2. Thiết kế báo cáo

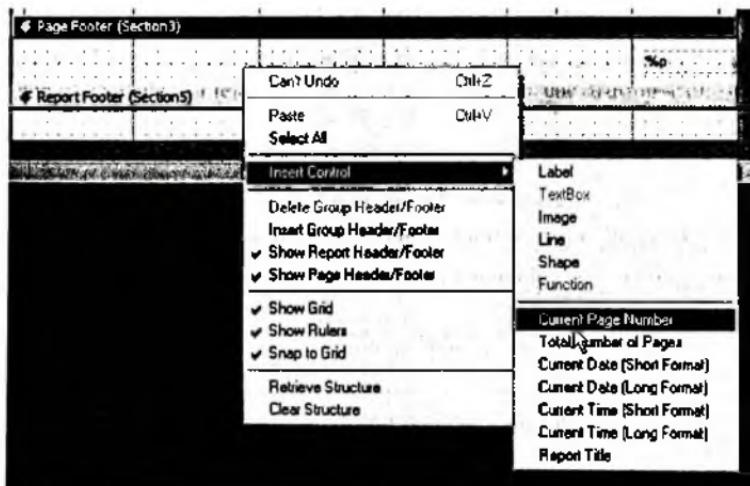
Trong hộp thoại Toolbox đã có sẵn một số công cụ cho phép ta thiết kế báo cáo gồm:

- RptLabel: cho phép viết các chuỗi ký tự trên báo cáo thường là tiêu đề báo cáo, tiêu đề cột hay là nơi để ta truyền một chuỗi ký tự từ ứng dụng vào báo cáo.

- RptTextbox: cho phép thiết kế nội dung báo cáo. Mỗi textbox liên kết với 1 trường trong bảng DL được xác định thông qua thuộc tính Data Member và Data - Field. Ta cũng có thể kéo, thả các trường trong một Command của môi trường dữ liệu Data Environment trực tiếp vào phần chi tiết (Detail) của báo cáo.

- RptImage Cho phép chèn hình ảnh vào báo cáo
- RptLine: Cho phép tạo ra các đường kẻ thường làm khung cho báo cáo
- RptShape: Cho phép tạo ra các hình vẽ (hình chữ nhật, vuông, tròn...)
- RptFunction: tạo kết quả tính toán thống kê trong phần cuối báo cáo (ví dụ số tổng cộng)

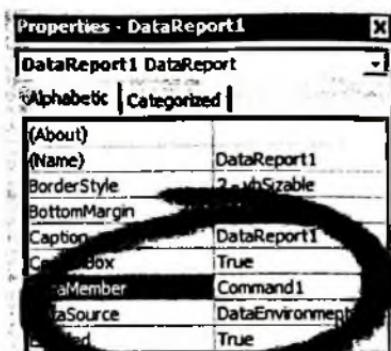
- Để thực hiện đánh số trang trong báo cáo. Tại phần PageFooter nháy phải chuột chọn InsertControl > Current Page Number



Để truyền dữ liệu vào báo cáo trong cửa sổ thuộc tính của báo cáo phải khai báo:

- Data Source: chọn tên đối tượng Data Environment cần liên kết

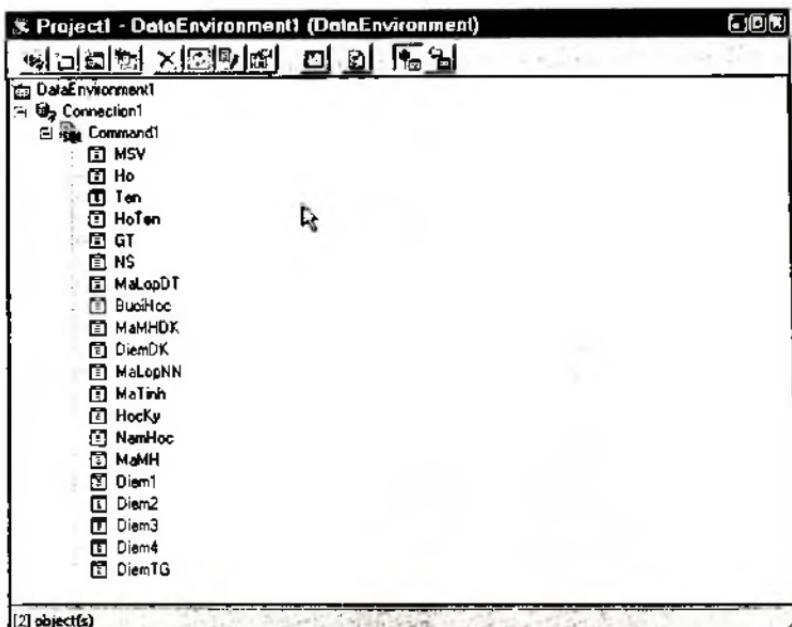
- Data member: chọn tên của Command liên kết với bảng dữ liệu mong muốn.



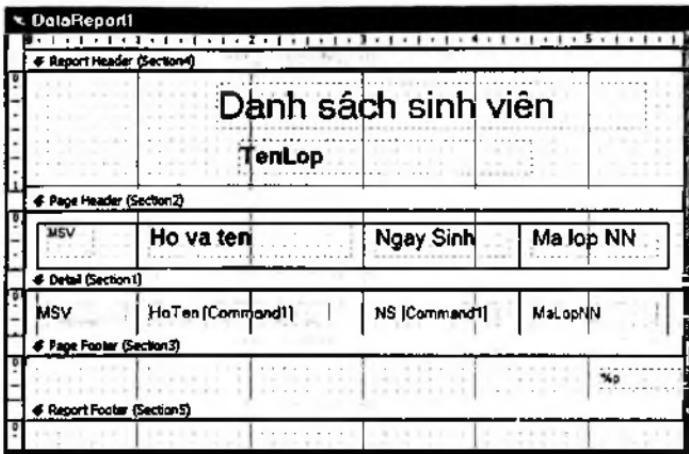
5.2.3. Gọi hiển thị báo cáo từ ứng dụng

- Ta sẽ thực hành tạo báo cáo In danh sách sinh viên theo lớp trong một ứng dụng Quản lý sinh viên.

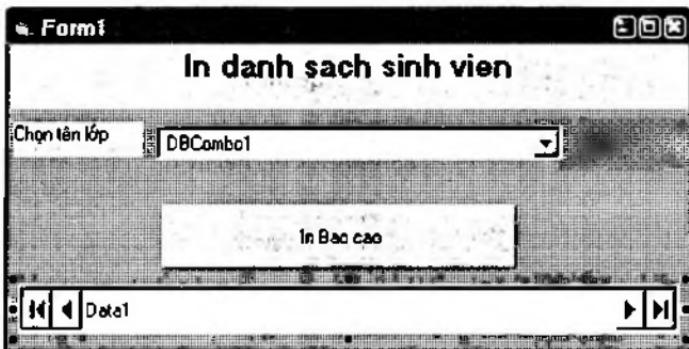
- Ta đã thiết lập môi trường dữ liệu DataEnvironment:



- Theo hướng dẫn ở trên ta đã tạo ra được báo cáo có mẫu như sau:



- Bây giờ để gọi hiển thị báo cáo trong ứng dụng, ta sẽ thiết kế một Form như sau:



Trong đó:

- + Điều khiển Data liên kết với tệp CSDL, thuộc tính RecordSource liên kết đến bảng DMLop phục vụ cho việc hiện tên lớp, lấy mã lớp cho hộp DBCombo1.
- + Hộp DBCombo1 với mục đích hiển thị danh sách tên lớp nhưng trả về mã lớp được thiết lập như sau:

Thuộc tính	Giá trị	Giải thích
Rowsource	Data1	Tên đối tượng Data kết nối
ListField	TenLop	Tên trường mà nội dung của nó sẽ được hiển thị
BoundColumn	MaLop	Tên trường mà giá trị sẽ được trả về khi người dùng chọn một tên lớp nào đó trong danh sách

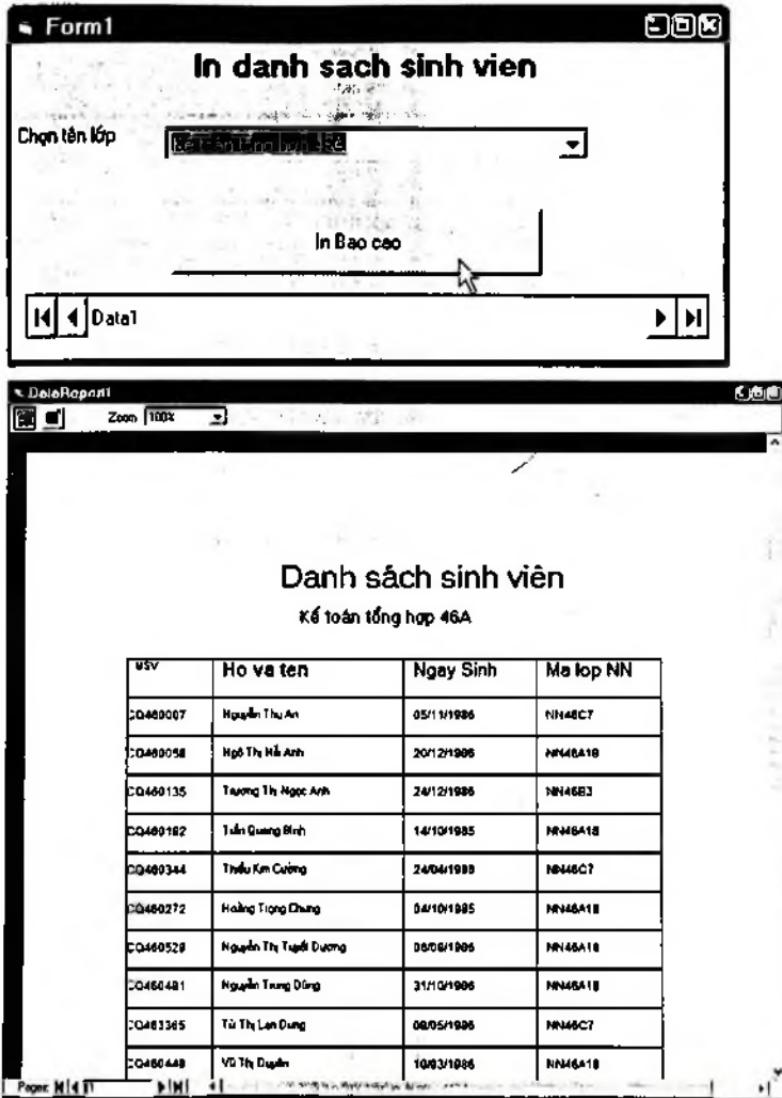
+ Nút InBaoCao được lập trình như sau:

```

Private Sub CmdInBC_Click()
Dim malop, tenlop, SQL As String
malop = DBCombo1.BoundText
tenlop = DBCombo1.Text
SQL = "Select * From dssv where malopdt="" & malop & """
'Kiem tra neu conection1 da duoc mo roi thi dong no lai
If DataEnvironment1.Connection1.State = 1 Then
    DataEnvironment1.rsCommand1.Close
End If
' Gan nguon du lieu cho Command1
DataEnvironment1.rsCommand1.Source = SQL
'Mo lai command1 voi nguon du lieu moi
DataEnvironment1.rsCommand1.Open
'truyen du lieu ten lop vao bao cao
DataReport1.Sections.Item("Section4").Controls.Item("lblTenlop").Caption
= tenlop
' goi hien thi bao cao
DataReport1.Show
End Sub

```

Khi thực hiện ứng dụng



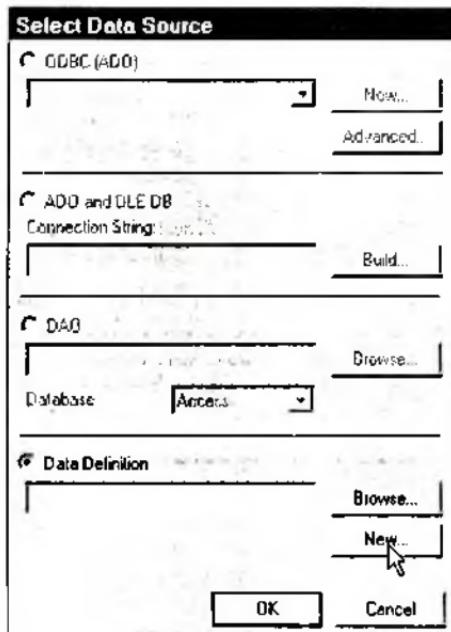
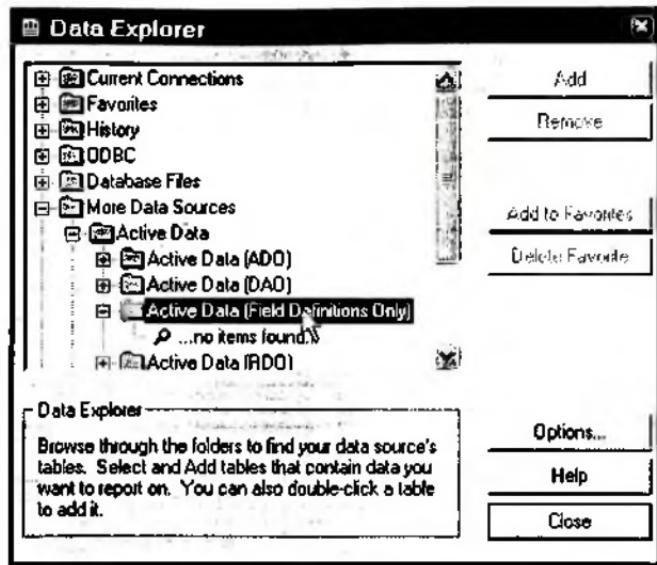
5.3. Tạo báo cáo với Crystal Report

5.3.1. Xây dựng bản mẫu báo cáo

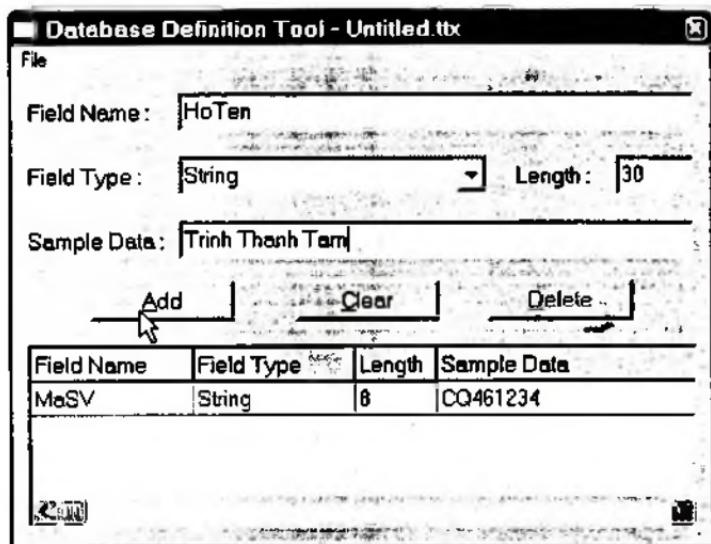
- Đầu tiên chọn Crystal Report \ Menu File \ New. Chọn Blank Report



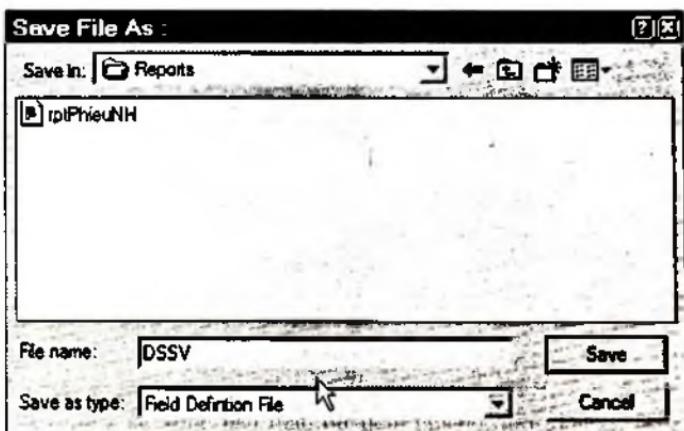
- Trong cửa sổ Data Explorer\ More Data Sources \ Active Data \ Active Data (Field Definition Only)\ Select Data Source. Nhấn Browse để chọn bản mẫu dữ liệu được thiết kế trước hoặc nút New để tạo mới một bản mẫu dữ liệu



- Trong cửa sổ thiết kế Data Definition Tools lần lượt khai báo thông tin các trường về File Name, Field Type, Sample Data



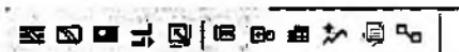
- Sau khi thiết kế xong bản mẫu dữ liệu, ghi nó dưới dạng một tệp (có phần mở rộng là .ttx) trong thư mục Report của dự án



- Bấm Add để thêm bản mẫu dữ liệu vào báo cáo mới. Bấm Close để đóng cửa sổ dữ liệu này lại.

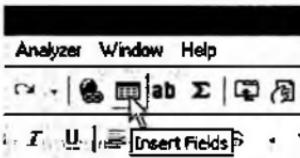
5.3.2. Thiết kế bản mẫu báo cáo

- Thêm các đối tượng báo cáo: trong hộp công cụ tại phía dưới, bên trái của màn hình. Ta chọn:

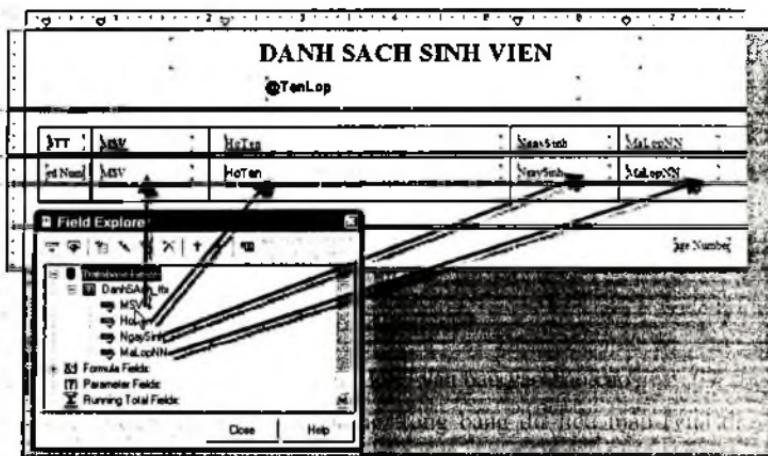


- + Insert line để thêm đường kẻ vào báo cáo
- + Insert box để kẻ khung trong báo cáo
- + Insert Picture để thêm hình ảnh vào báo cáo

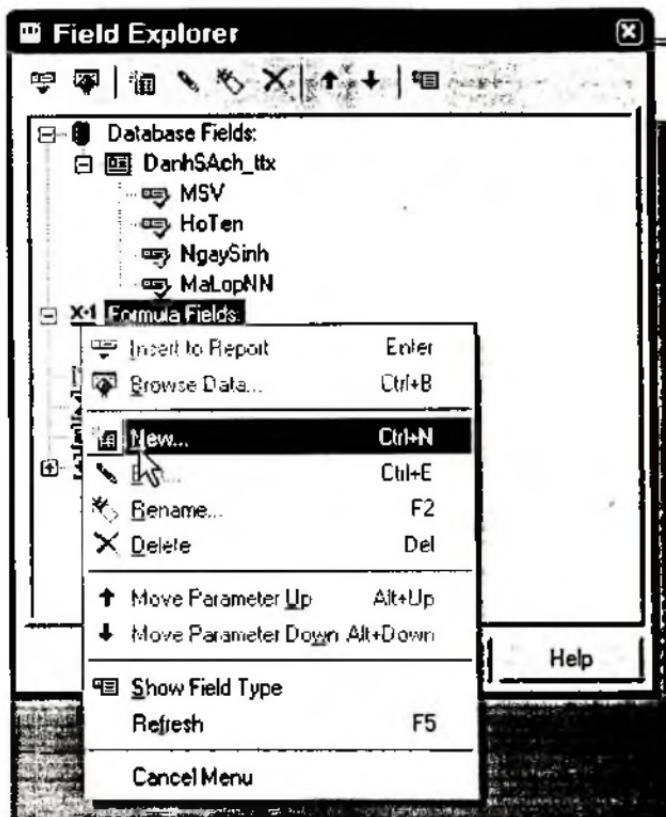
Trên thanh công cụ phía trên màn hình, ta chọn:



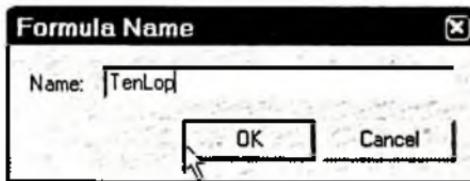
- + Insert Text Object để thêm các dòng văn bản vào báo cáo
- + InsertFields để thêm các trường trong bảng dữ liệu mẫu (vừa được khai báo) vào báo cáo.



- Để tạo ra một trường công thức (Formula Field) để cho phép truyền một giá trị dữ liệu vào báo cáo. Trong cửa sổ Field Explorer chọn mục Formula Field; nháy phải chuột chọn New



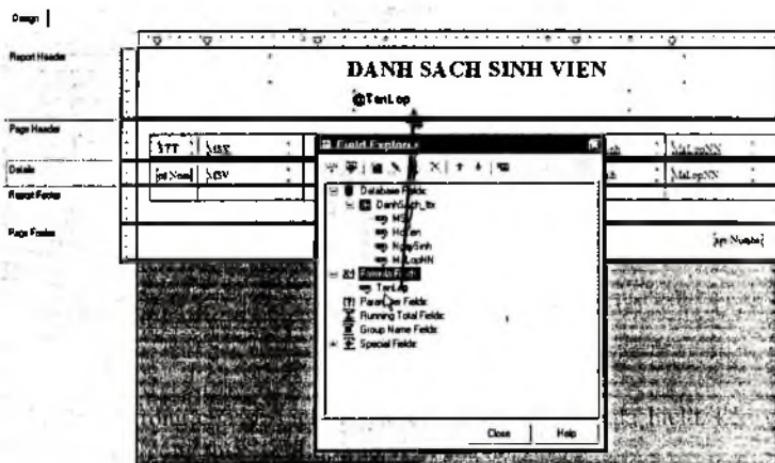
Trong cửa sổ hiện ra tiếp theo, đặt tên cho Formula là TenLop



Vì ta chỉ có mục đích để tạo ra một Formula Field để truyền vào báo cáo chuỗi tên lớp nên trong cửa sổ tiếp theo hiện ra ta chọn ngay nút "Save and Close (Ctrl+S)"



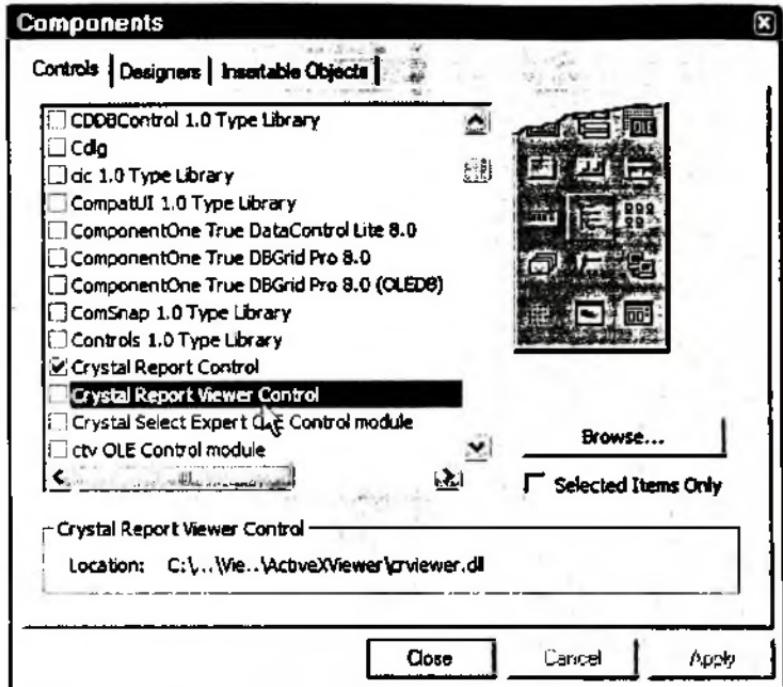
Cuối cùng trong cửa sổ Field Explorer chọn mục Formula Field, kéo và thả trường công thức TenLop vào vị trí của nó.



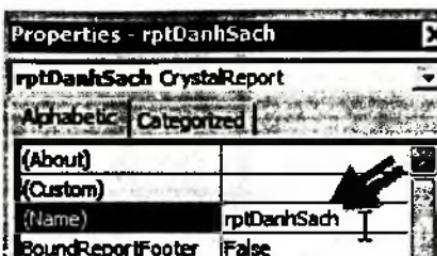
- Lưu báo cáo vào thư mục Report trong dự án với phần mở rộng Crystal Report. Ví dụ báo cáo có tên rptHSSV.rpt

5.3.3. Truyền dữ liệu và hiển thị báo cáo

- Trong Form in báo cáo ta đã tạo được phía trên, thêm vào một nút mới có tên là CrystalReport.
- Khai báo và thêm điều khiển CrystalReport vào Form



Đặt tên cho điều khiển là: rptDanhSach



Lập trình cho nút bấm cmdCrystalReport như sau:

```
Private Sub CmdCrystalReport_Click()
Dim mdb As DAO.Database
Dim res As DAO.Recordset
Dim SQL As String
Set mdb = OpenDatabase(App.Path & "\Database\qlsv2.mdb")
SQL = "Select MSV, HoTen, NS, MaLopNN From dssv where
malopDT=" & DBCCombo1.BoundText & ""
rptDanhSach.ReportFileName = App.Path & "\Report\Danhsach.rpt"
'Mo recordset chua noi dung du lieu se truyen vao bao cao
Set res = mdb.OpenRecordset(SQL)
'truyen du lieu vao bao cao
rptDanhSach.SetTablePrivateData 0, 3, res
'truyen chuoi ten lop vao truong cong thuc trong bao cao
rptDanhSach.Formulas(0) = "TenLop=" & DBCCombo1.Text & ""
'goi hien thi bao cao
rptDanhSach.PrintReport
End Sub
```

- Ứng dụng hoạt động

Form1

In danh sach sinh vien

Chọn tên lớp: Kế toán tổng hợp 46A

In Báo cáo **Crystal Report**

Danh sách sinh viên
Kế toán tổng hợp 46A

STT	MSSV	Họ Tên	Nhập sinh	Mã số CMND
1	CQ460002	Nguyễn Thị An	05/11/1986	NN46C1
2	CQ460058	Ngô Thị Hải Anh	20/12/1986	NN46A19
3	CQ460123	Trương Thị Ngọc Anh	24/12/1986	NN46B1
4	CQ460192	Trần Quang Bình	14/10/1985	NN46A29
5	CQ460342	Thiều Khanh Cường	24/04/1986	NN46C1
6	CQ460272	Hoàng Trung Chung	04/10/1985	NN46A18
7	CQ460529	Nguyễn Thị Puyên Dương	06/08/1986	NN46A10
8	CQ460383	Nguyễn Trung Dũng	31/10/1986	NN46A19
9	CQ461383	TU THỊ LÂM DŨNG	08/05/1986	NN46C7
10	CQ460448	VŨ THỊ DUYÊN	10/03/1986	NN46A18
11	CQ460872	Trương Thị Hường Giang	07/12/1986	NN46C7
12	CQ460091	Lương Thị Hàng	02/03/1986	NN46C7
13	CQ460540	Văn Thúy Hàng	04/03/1986	NN46A15

CÂU HỎI ÔN TẬP CHƯƠNG 5

1. Nguyên tắc cơ bản trong thiết kế báo cáo đầu của một dự án phần mềm?
Cho ví dụ?
2. Các bước để xây dựng một báo cáo truy vấn cơ sở dữ liệu trong VisualBasic?
3. Trình bày các bước để thiết kế và hiển thị báo cáo sử dụng DataReport
4. Trình bày các bước để thiết kế và hiển thị báo cáo sử dụng CRYSTAL REPORT (sử dụng Active data _Field definitions only)
5. Hãy thiết kế mẫu cho các báo cáo sau:
 1. Bảng kê hóa đơn bán hàng cho một trung tâm thương mại
 2. Bảng tổng hợp doanh thu bán hàng theo mặt hàng
 3. Bảng tổng hợp doanh thu bán hàng theo tháng
 4. Bảng tổng hợp nhập xuất tồn kho hàng hóa vật tư
 5. Bảng danh sách sinh viên phục vụ cho việc điểm danh
 6. Bảng điểm kiểm tra, chuyên cần sinh viên
 7. Bảng điểm học phần của sinh viên
 8. Bảng điểm cá nhân của mỗi sinh viên khi kết thúc khóa học
 9. Bảng kiểm kê sách trong kho của thư viện
 10. Bảng thống kê về tình hình mượn sách theo đầu sách

Chương 6

BIÊN SOẠN TÀI LIỆU HƯỚNG DẪN SỬ DỤNG

6.1. Các bước chuẩn bị

Chuẩn bị phần mềm:

- Hiện nay có rất nhiều phần mềm có thể giúp bạn tạo được một ebook .chm dễ dàng như Power CHM, CHM Maker,... nhưng để tối ưu hóa dung lượng và khả năng tương thích cao với các HĐH Windows bạn nên sử dụng phần mềm **HTML Help Workshop** được cung cấp bởi Microsoft và bạn có thể tải về và sử dụng miễn phí từ website của Microsoft www.microsoft.com.vn. HTML Help Workshop có thể giúp bạn tạo một file trợ giúp liên kết với phần mềm, khi đó bạn có thể quy định cho file trợ giúp của mình mở đúng phần trợ giúp tương ứng với phần cần trợ giúp trong phần mềm khi bạn nhấn phím F1.

- Máy bạn cũng cần phải cài một trong các phần mềm có thể tạo ra định dạng HTML như Dreamware, MS FrontPage,... vì file.chm được tạo từ các trang .html và bạn cũng cần có một ít kiến thức cơ bản về HTML để tạo ra được những trang web đơn giản.

- Ngoài ra bạn cũng cần phải cài một trong các phần mềm xử lý ảnh và biết một vài thao tác cơ bản về xử lý ảnh như thay đổi kích cỡ hình ảnh, làm nền của ảnh trong suốt... để chèn vào các trang tài liệu của bạn thêm phần sống động.

Chuẩn bị nội dung

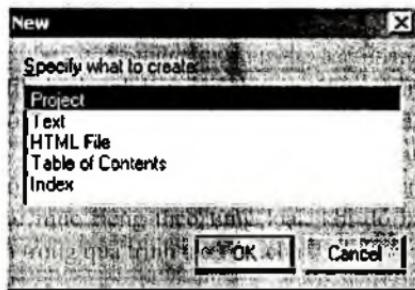
Trước tiên bạn phải tập trung tất cả các nội dung cần thiết cho tài liệu của mình và đưa chúng vào một thư mục riêng và chỉ dành cho việc tạo ebook .chm mà bạn dự định làm. Những tập tin nội dung có thể đưa vào .chm: trang html, hình ảnh, video, javascript, flash...

- Đối với các file html nên đặt tên tiếng Việt không dấu liền nhau, ngắn gọn, gợi nhớ đối với bạn để khi tiến hành làm bạn dễ dàng tìm và chỉ đến đúng file cần tìm.

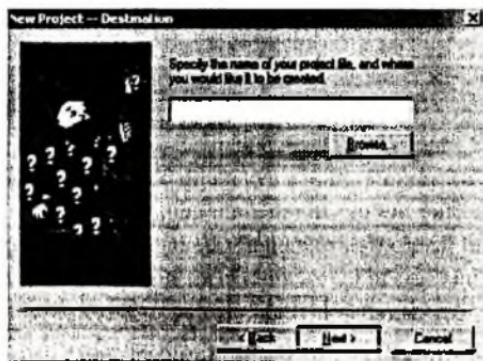
- Đối với các file hình ảnh bạn nên chuyển tất cả chúng về định dạng gif để có kích thước tối ưu, sau đó đưa tất cả các file hình ảnh vào một thư mục để dễ quản lý, thư mục chứa hình ảnh này là thư mục con của thư mục chính. Tương tự đối với các dạng dữ liệu khác không phải html bạn nên đưa chúng vào một thư mục riêng theo từng loại, khi đó bạn sẽ liên kết đến chúng dễ dàng hơn trong quá trình tạo file .chm.

6.2. Tạo mới một Help Project

Bước 1: Khởi động HTML Help Workshop từ Start - Programs - HTML Help Workshop. Nhấn nút New trên thanh công cụ hoặc File - New, trên màn hình chọn Project. Trong màn hình tiếp xuất hiện sau đó, không đánh dấu vào ô chọn bên dưới và nhấn Next để tiếp tục

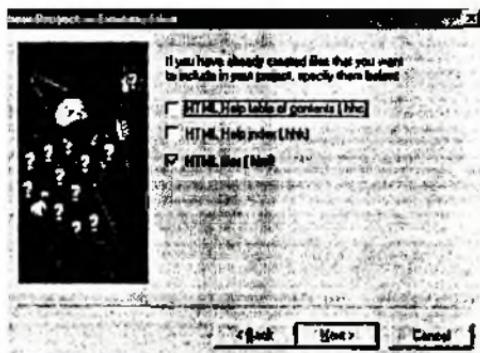


Bước 2: Xuất hiện màn hình chọn nơi đặt tên Project, khi đó nhấn nút Browse để chỉ vào thư mục mà bạn đã tạo riêng cho file .chm và đặt tên cho Project, tên project cũng sẽ là tên của file .chm sau khi hoàn tất.





Bước 3: Lưu ý bước này có một thao tác quan trọng mà bạn không nên bỏ qua, trên màn hình xuất hiện 3 mục chọn hỏi bạn đã có một trong các loại tập tin thành phần để cấu thành nên file .chm nào chưa:



- HTML help table of contents (.hhc) - là tập tin chứa các mục lục - tức các chương mục xuất hiện trong cửa sổ trái file trợ giúp và có cấu trúc cây. Trong trường hợp chúng ta tạo mới một project hoàn toàn thì file này chưa có nên không đánh dấu chọn mục này.

- HTML help index - là tập tin chứa chi mục, là danh sách từ khóa nằm trong cửa sổ bên trái của file trợ giúp mà khi bạn gõ vào ô chữ ở trên nếu trùng khớp với từ khóa có sẵn trong danh sách thì trong màn hình bên phải sẽ hiện lên nội dung trợ giúp tương ứng. Và nếu một project mới hoàn toàn thì file dạng này cũng không có sẵn, bạn không nên chọn mục này.

- HTML files (.html) - lưu ý bạn phải đánh dấu vào mục chọn này. Nhấn Next, kích nút ADD.

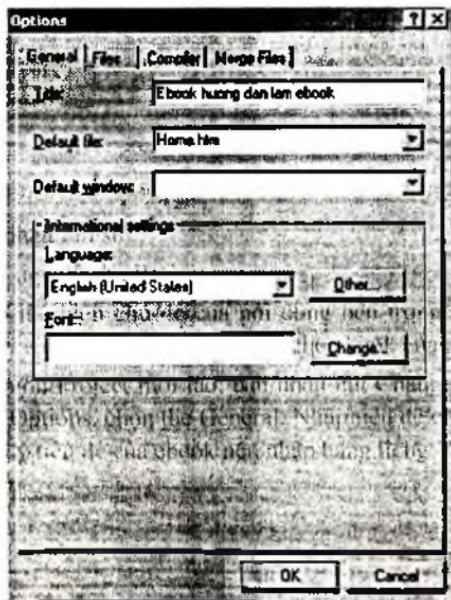


Nhấn Next, nhấn Finish

Bước 4: Tạo tiêu đề cho Project

Để đảm bảo thể hiện chủ đề của nội dung bên trong ebook cũng như tính thẩm mỹ của ebook bạn nên đặt tiêu đề cho ebook của mình.

Trong màn hình Project mới tạo, bạn nhấn nút Change project options. Trong hộp thoại Options, chọn thẻ General. Nhập tiêu đề cho ebook của bạn vào ô Title. Lưu ý tiêu đề của ebook nên nhập bằng tiếng Việt không dấu.

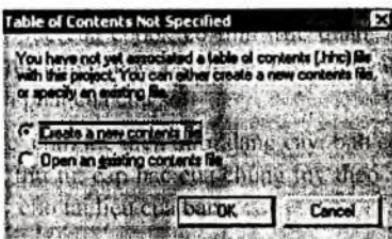


6.3. Tạo mục lục cho nội dung hướng dẫn

Mục đích: Mục lục của ebook tương tự như mục lục của sách thông thường, nhưng mục lục của ebook có tính trực quan, khi bạn muốn tham khảo thông tin liên quan đến mục nào đó chỉ cần kích vào mục lục sẽ hiện ra ngay trong phần bên phải của cửa sổ.

Mục lục của file .chm sẽ hiện dưới dạng cây, bạn có thể thêm bớt, xóa, sửa, thay đổi vị trí, thứ tự, cấp bậc của chúng tùy theo ý thích của mình để đảm bảo tính hợp lý cho tài liệu của bạn.

Thực hiện: Để tạo mục lục cho tài liệu, bạn kích vào thẻ Contents, trong hộp thoại xuất hiện tiếp đó chọn **Create a new contents file** nếu bạn chưa có file chứa các mục lục, chọn **Open an existing contents file** để mở file chứa mục lục có sẵn. Nhấn OK, nhập tên của file chứa mục lục. Lưu ý: Tất cả các file để tạo .chm nên đặt tên là tiếng Việt không dấu, ngắn gọn, gợi nhớ với bạn để phòng khi bạn muốn tái sử dụng để nâng cấp, thay đổi nội dung tài liệu được dễ dàng.



Các chức năng của thẻ Contents: Trong thẻ contents có các nút chức năng giúp bạn tạo mới, xóa, sửa các mục lục của mình.

Nút **Properties** dùng để thiết lập các lựa chọn riêng của bạn cho phần mục lục của tài liệu. Nếu bạn mới bắt đầu tạo .chm thì chưa nên sử dụng chức năng này.

Nút **Insert a heading** dùng để tạo mới một chủ đề chính, là những chương, mục trong tài liệu của bạn.

Nút **Insert a page** dùng để tạo mới một mục lục cho trang tài liệu, khi kích vào mục lục sẽ hiện nội dung của trang tài liệu tương ứng ở cửa sổ bên phải trong ebook.



Nút edit selection để chỉnh sửa, thay đổi một mục lục đang chọn.



Nút Delete selection để xóa một mục lục đang chọn.



Các nút mũi tên dùng để thay đổi vị trí, cấp bậc của các mục lục đang chọn.



Nút View HTML Source để xem mã HTML của trang tài liệu đang gắn với mục lục. Chức năng này không nên sử dụng vì bạn có thể làm rối nội dung bên trong tài liệu khi kích nhầm vào nút lưu tài liệu từ màn hình này.



Nút Save để lưu lại những thay đổi khi bạn kích vào nút View HTML Source. Không sử dụng chức năng này, vì khi bạn nhấn Save thì nếu trang tài liệu bạn có tiếng Việt thì tiếng Việt sẽ không hiển thị nữa và trang tài liệu của bạn hỏng hoàn toàn.

6.4. Hiển thị mục lục tiếng việt

Thực hiện các bước sau để có thể tạo được một mục lục tiếng Việt giúp người đọc ebook của bạn dễ dàng theo dõi, và mặc định HTML Help Workshop không hỗ trợ chúng ta nhập trực tiếp tiếng Việt Unicode để tạo ra các mục lục bằng tiếng Việt.

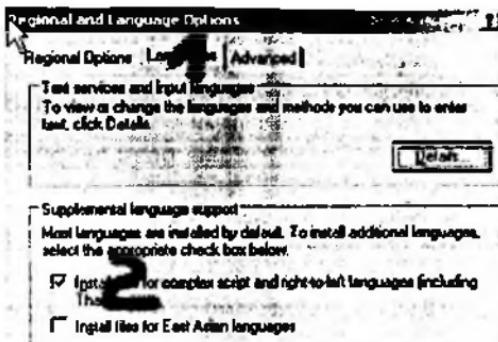
Thiết lập trong hệ thống: Tùy theo việc bạn sử dụng HDH nào chúng ta có các bước riêng để cấu hình hệ thống hỗ trợ tiếng Việt cho phần mục lục của file .chm

Đối với Windows 2000:

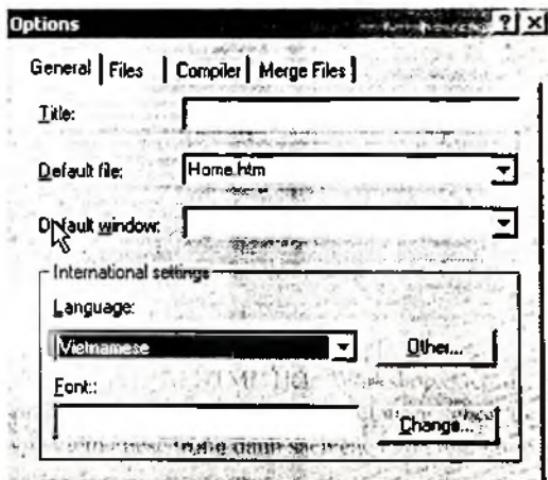
Vào Control Panel Regional Options. Chọn thẻ General, đánh dấu vào Vietnamese trong danh sách các ngôn ngữ và đưa đĩa cài đặt Windows 2000 vào. Sau đó khởi động lại máy để hoàn tất việc nạp các file hệ thống hỗ trợ tiếng Việt bằng mã Vietnamese locate CP 1258.

Đối với Windows XP:

Vào Control Panel, chạy Regional and Language Options. Chọn thẻ Language, đánh dấu vào mục Intall files for complex script and right-to-left languages (including Thailand) và nhớ phải đưa đĩa cài đặt Windows XP và ô CD để HDH nạp thêm phần hỗ trợ tiếng Việt vào HDH.



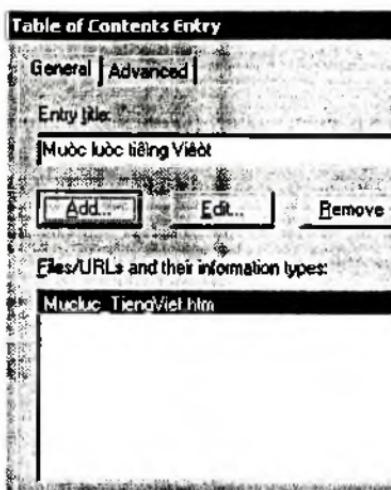
Thiết lập trong Project: Từ HTML Help Workshop, kích th调动 project options. Trong hộp thoại Options, chọn thẻ General trên hộp thoại, chọn Vietnamese trong danh sách các ngôn ngữ tại mục Language.



Chuẩn bị bộ gõ: Bạn nên dùng bộ gõ Unikey để dễ dàng chuyển đổi qua lại giữa các bảng mã. Để gõ được tiếng Việt cho mục lục, trong bảng điều khiển của Unikey chọn bảng mã là Vietnamese locate CP 1258.

Nhập mục lục bằng tiếng Việt: Trong ô nhập tên mục lục, bạn nhập tên mục lục tiếng Việt có dấu, nhưng có thể khi bạn nhập sẽ không ra dấu tiếng Việt ngay, và khi nhập xong trong màn hình thiết kế của Project cũng không hiển thị được tiếng Việt ở các mục lục, bạn đừng quá lo lắng và chỉ việc

nhập đúng chính tả là được. Tiếng Việt có dấu ở mục lục sẽ hiển thị khi bạn dịch ra thành .chm



Xem kết quả: Nhấn vào nút hình chiếc phiến để dịch ra ebooks. Nếu trong màn hình bên phải không hiện thông báo lỗi thì bạn nhấn nút View complete file (hình đôi mắt kiến) để xem file .chm của mình. Nếu sai chính tả ở mục lục thì lại nhấn nút Edit để chỉnh sửa.

6.5. Biên tập nội dung

Tất cả các nội dung của ebooks được biên tập thành các trang html. bạn có thể sử dụng một trong các phần mềm tạo được html như Dreamware, Frontpage, Notepad,... Dưới đây là một số lời khuyên dành cho bạn:

Về font chữ: Sử dụng font chữ thuộc bảng mã Unicode, kinh nghiệm cho thấy dùng font **Verdana** với size 2 (10pt) cho phần nội dung thì tệp hướng dẫn sử dụng của bạn rất thẩm mỹ, dễ đọc.

Về hình nền: bạn nên sử dụng những màu nền đơn giản, sáng rõ, có kích thước gọn nhẹ và dùng định dạng .gif; không dùng hình nền có quá nhiều màu, nhiều chi tiết vì khi mở trang đó ra một hình nền nhân thành nhiều hình mà các chi tiết không ghép nối được với nhau gây phản cảm. Ví dụ trang này sử dụng một hình nền có kích thước chỉ 1KB, sưu tầm từ www.pcworld.com.vn

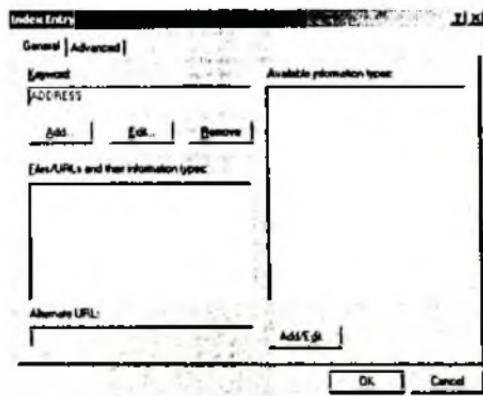
Bố cục nội dung: đề trang nội dung đưa vào ebook có độ chuyên nghiệp, tính thẩm mỹ bạn nên dùng một hoặc nhiều hình ảnh làm banner giống như một trang web thực sự làm cho người đọc thấy sự khác biệt, hoặc chủ đề của ebooks của bạn. Cuối mỗi trang có thể tạo thêm ấn tượng bằng cách ghi tên mình, địa chỉ email, ...

Biên tập nội dung: Những nội dung muốn thể hiện trong trang tài liệu của bạn tốt nhất nên đặt chúng vào bên trong các table, hầu hết các trang web người ta sử dụng table để cố định vị trí các nội dung. Đối với những nội dung bạn copy từ các nguồn khác không phải tự gõ thì nên xóa hết định dạng của chúng và định dạng lại theo cách của bạn để giảm thiểu kích thước, đảm bảo sự phù hợp với các trang tài liệu khác của bạn trong ebooks.

6.6. Tạo chỉ mục cho tài liệu hướng dẫn

Từ thẻ Index, nhấn nút Insert a keyword để thêm một từ khóa vào danh sách các từ khóa của chỉ mục.

Trong hộp thoại tạo mới từ khóa, nhập từ khóa và nhấn nút Add để chọn trang HTML chứa nội dung cần liên kết đến.



Trong hộp thoại để chọn trang HTML, nhấn nút Browse và chỉ đến trang HTML mà bạn muốn liên kết khi gõ từ khóa đó trong mục Index của ebook. Lưu ý: trang HTML này phải được lưu trong thư mục dành riêng cho ebook của bạn, nếu không khi bạn chép file .chm sau khi dịch xong đến một nơi khác nó sẽ không có nội dung khi bạn gõ từ khóa trên. Đây là điều bạn cần phải lưu ý nhất trong khi làm file .chm.



6.7. Sử dụng Bookmark

Thông thường, chúng ta thường sử dụng một từ khóa để liên kết với một nội dung rất nhỏ và việc liên kết mỗi từ khóa với một trang html chứa một nội dung nhỏ làm cho tài liệu của bạn có dung lượng lớn, nội dung phân tán và không logic.

Ứng dụng bookmark trong việc tạo một từ khóa trong danh sách chi mục để liên kết đến một nội dung được đánh dấu trong một trang html chứa nhiều nội dung. Và chúng ta có thể tập hợp được nhiều nội dung liên quan nhỏ và một trang html và sử dụng chức năng bookmark rồi tạo các từ khóa liên kết đến từng nội dung trong trang html đó.

Đặt bookmark trong trang html.

Chọn nội dung cần đánh dấu trong trang html, vào Insert - Bookmark và đặt tên cho vùng bạn chọn. Tên phải ngắn gọn, không có dấu, dễ nhớ với bạn.

Trong hình bên là danh sách các bookmark để đánh dấu các nội dung cho ebook "Thuật ngữ tin học" của tác giả đã thực hiện gồm các nội dung có cùng vấn với nhau vào một trang riêng rồi đánh dấu chúng.



Tạo từ khóa liên kết đến một bookmark.

Từ thẻ Index trong HTML Help Workshop, nhấn nút Insert a keyword như thao tác tạo một từ khóa thông thường. Bạn cũng nhập từ khóa, chọn đường dẫn đến trang html chứa nội dung cần liên kết.

Điều khác biệt duy nhất là sau khi bạn chọn trang chứa nội dung liên kết với từ khóa đã nhập thì nhập thêm dấu #<bookmark> vào sau đường dẫn của trang html đó.

Ví dụ hình bên sử dụng từ khóa AGP để liên kết đến một nội dung được đánh dấu bằng tên AGP bên trong trang html a.htm

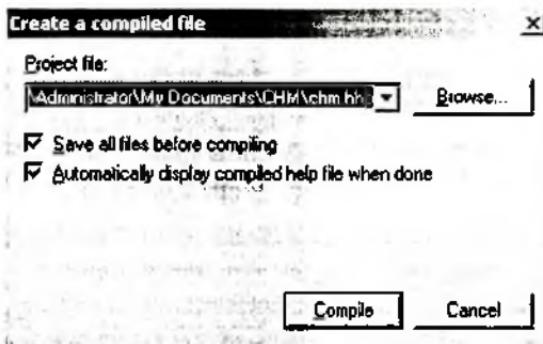


Bạn cũng có thể dùng bookmark cho phần mục lục tương tự như với từ khóa ở phần chi mục. Các bước tương tự như đã hướng dẫn ở trên.

6.8 Biên dịch help project thành tệp .chm

Đây là khâu cuối cùng để bạn biên dịch Project của mình thành .chm, tuy nhiên trong quá trình làm .chm bạn cũng nên thường xuyên sử dụng chức năng này để kiểm tra kết quả của mình.

Kích vào nút **Compile HTML files** trên thanh công cụ để biên dịch tất cả các thành phần của Project thành một file .chm duy nhất. Ngay sau đó một hộp thoại xuất hiện yêu cầu bạn chỉ đến tập tin Project cần dịch, mặc định sẽ chỉ đến Project đang mở.



Trong hộp thoại này bạn nên đánh dấu chọn mục **Save all files before compiling** - lưu tất cả các file liên quan đến Project trước khi tiến hành biên dịch. Và đánh dấu cho mục **Automatically display compiled help file when done** - hiển thị kết quả là file .chm sau khi dịch xong. Trong một số trường hợp chức năng tự động hiển thị kết quả có thể sẽ không có hiệu lực và bạn không nên quá lo lắng.

Chọn xong bạn nhấn nút **Compile** và trong màn hình bên phải sẽ xuất hiện nhật ký của quá trình dịch .chm, nó cho biết thời gian dịch .chm, có bao nhiêu chủ đề, bao nhiêu liên kết trong nội bộ các file, có bao nhiêu liên kết đến Internet, có bao nhiêu hình ảnh được sử dụng trong file .chm của bạn. Nếu sau khi nhấn nút dịch mà bạn không nhận được kết quả tương tự như bên dưới thì chắc chắn rằng quá trình tạo .chm của bạn đã bị lỗi.

Compile time: 0 minutes, 6 seconds

18 Topics
34 Local links
21 Internet links
42 Graphics

Created c:\Documents and Settings\Administrator\My Documents\CHM\chm.chm.

2,498,643 bytes

Compression decreased file by 165,043 bytes.

Lỗi thường gặp nhất trong quá trình tạo .chm là file .chm của bạn đã tạo ra lần trước còn dang mở, bạn phải tắt đi thì quá trình dịch .chm mới được thực hiện; lỗi thường gặp tiếp theo là bạn liên kết đến một file .html có tên là tiếng Việt có dấu hoặc nằm trong một đường dẫn có tên là tiếng Việt có dấu.

Nếu có quá trình dịch thành công - tức không có thông báo lỗi nào xuất hiện và .chm không tự mở như bạn chọn thì bạn nhấn nút xem kết quả hình đôi mắt kiếng màu đỏ bên cạnh nút dịch để xem kết quả. Nếu xuất hiện hộp thoại yêu cầu chọn .chm thì chọn đúng .chm trong thư mục của Project.

6.9. Kết nối tệp Help với ứng dụng

Tạo một module mới có tên là showhelp với nội dung như sau:

Type HH_IDPAIR

dwControlId As Long

dwTopicId As Long

End Type

Public ids(2) As HH_IDPAIR

Declare Function GetDlgCtrlID Lib "user32" _

 (ByVal hwnd As Long) As Long

Private Declare Function HtmlHelp Lib "hhctrl.ocx" Alias "HtmlHelpA" _

 (ByVal hwndCaller As Long, ByVal pszFile As String, _

 ByVal uCommand As Long, ByVal dwData As Long) As Long

Private Declare Function HTMLHelpTopic Lib "hhctrl.ocx" Alias

 "HtmlHelpA" _

```
(ByVal hwndCaller As Long, ByVal pszFile As String, _
    ByVal uCommand As Long, ByVal dwData As String) As Long
Private Declare Function HtmlHelpSearch Lib "hhctrl.ocx" Alias
    "HtmlHelpA" _
        (ByVal hwndCaller As Long, ByVal pszFile As String, _
            ByVal uCommand As Long, dwData As HH_FTS_QUERY) As
    Long
Private Declare Function HtmlHelpIndex Lib "hhctrl.ocx" Alias
    "HtmlHelpA" _
        (ByVal hwndCaller As Long, ByVal pszFile As String, _
            ByVal uCommand As Long, dwData As HH_AKLINK) As Long
Private Declare Function SendMessage Lib "user32" Alias
    "SendMessageA" _
        (ByVal hwnd As Long, ByVal wMsg As Long, _
            ByVal wParam As Long, ByVal lParam As Any) As Long

' HH API const
'-----
Public Const HH_DISPLAY_TOPIC = &H0
' select last opened tab, [display a specified topic]
Public Const HH_DISPLAY_TOC = &H1
' select contents tab, [display a specified topic]
Public Const HH_DISPLAY_INDEX = &H2
' select index tab and searches for a keyword
Public Const HH_DISPLAY_SEARCH = &H3
' select search tab (perform a search is not working)

Private Const HH_SET_WIN_TYPE = &H4
Private Const HH_GET_WIN_TYPE = &H5
Private Const HH_GET_WIN_HANDLE = &H6
```

```
Private Const HH_KEYWORD_LOOKUP = &HD
' Searches for one or more keywords

Private Const HH_DISPLAY_TEXT_POPUP = &HE
' Display string resource ID or

Private Const HH_HELP_CONTEXT = &HF
' display mapped numeric value in dwData

Private Const HH_TP_HELP_CONTEXTMENU = &H10
' Text pop-up help, similar to WinHelp's HELP_CONTEXTMENU.

Private Const HH_TP_HELP_WM_HELP = &H11
' text pop-up help, similar to WinHelp's HELP_WM_HELP.

' HH_DISPLAY_SEARCH Command Related Structures and Constants
' -----
Public Const HH_FTS_DEFAULT_PROXIMITY = -1

Public Type HH_FTS_QUERY      ' UDT for accessing the Search tab
    cbStruct        As Long      ' Sizeof structure in bytes.
    fUniCodeStrings As Long      ' TRUE if all strings are unicode.
    pszSearchQuery As String     ' String containing the search query.
    iProximity       As Long      ' Word proximity.
    fStemmedSearch   As Long      ' TRUE for StemmedSearch only.
    fTitleOnly       As Long      ' TRUE for Title search only.
    fExecute         As Long      ' TRUE to initiate the search.
    pszWindow        As String     ' Window to display in
End Type

' HH_ALINK Command Related Structures and Constants
' -----
Public Type HH_AKLINK
```

```
cbStruct      As Long  
fReserved     As Boolean  
pszKeywords   As String  
pszUrl        As String  
pszMsgText    As String  
pszMsgTitle   As String  
pszWindow      As String  
fIndexOnFail  As Boolean  
End Type
```

Public Function HFile(ByVal i_HFile As Integer) As String

'..... Set the string variable to include the application path of helpfile

Select Case i_HFile

Case 1

'--- Applications main help file

HFile = App.Path & "\help\Tên File Help của bạn.chm"

Case 2

'----- [optional] special help window definition

HFile = App.Path & "\help\ Tên File Help của bạn.chm >wndTopic"

Case 3

'----- Place other Help file paths in successive case statements

HFile = App.Path & "\help\CHM-other-language.chm"

End Select

End Function

Public Sub ShowContents(ByVal intHelpFile As Integer)

HtmlHelp hwnd, HFile(intHelpFile), HH_DISPLAY_TOC, 0

End Sub

Public Sub ShowIndex(ByVal intHelpFile As Integer)

```
    HtmlHelp hwnd, HFile(intHelpFile), HH_DISPLAY_INDEX, 0
End Sub

Public Sub ShowTopic(ByVal intHelpFile As Integer, strTopic As String)
    HTMLHelpTopic hwnd, HFile(intHelpFile), HH_DISPLAY_TOPIC,
    strTopic
End Sub

Public Sub ShowTopicID(ByVal intHelpFile As Integer, IdTopic As Long)
    HtmlHelp hwnd, HFile(intHelpFile), HH_HELP_CONTEXT, IdTopic
End Sub

Để hiển thị nội dung trợ giúp, tại những nơi cần thiết bạn chỉ cần đe câu
lệnh gọi các thủ tục tương ứng. Ví dụ như
ShowContents 1

....
```

Chương 7

TẠO BỘ CÀI ĐẶT CHO ỨNG DỤNG

Một công đoạn không thể thiếu trong quy trình sản xuất phần mềm hoàn chỉnh đó chính là việc tạo bộ cài đặt cho phần mềm.

Bộ cài đặt phần mềm là một tập hợp bao gồm tệp thi hành của phần mềm cùng với các tệp thành phần khác có liên quan như tệp cơ sở dữ liệu, tệp báo cáo, tệp thư viện... được bố trí để có thể cài đặt vào những thư mục phù hợp trên máy của người sử dụng sao cho phần mềm ứng dụng có thể hoạt động độc lập mà không cần đến môi trường phát triển của nó.

Một phần mềm ứng dụng, được tạo bởi ngôn ngữ lập trình bậc cao, khi hoạt động thường sử dụng, truy xuất đến rất nhiều tệp liên quan:

- Các tệp cơ sở dữ liệu
- Các tệp chứa mẫu báo cáo
- Các tệp hình ảnh (icons, pictures, wall papers...)
- Tệp thư viện (dll, ocx...)
- Tệp cấu hình

Tại máy tính của nhà sản xuất phần mềm, các tệp liên quan trên được cài đặt đầy đủ khi cài đặt môi trường phát triển phần mềm. Tuy nhiên, tại máy tính của người sử dụng cuối thì thường là không có sẵn hệ thống các tệp thư viện và các tệp cấu hình cần thiết để cho phần mềm ứng dụng có thể hoạt động. Hơn nữa, các tệp thư viện, tệp cấu hình lại thường không nằm cùng hệ thống thư mục với tệp thi hành của ứng dụng (trong c:\program files) mà chúng thường được đặt trong thư mục riêng do hệ điều hành quản lý (ví dụ: c:\windows\system32). Đây cũng chính là lý do tại sao khi người dùng sao chép toàn bộ một hệ thống thư mục của một phần mềm ứng dụng nào đó (trong c:\Program files) về máy của mình nhưng phần mềm ứng dụng vẫn không thể hoạt động được.

Trong chương này, chúng ta sẽ học cách tạo bộ cài đặt cho phần mềm ứng dụng viết bằng visual basic sử dụng tiện ích có sẵn trong bộ visual studio là **Package and Deployment** và một công cụ tạo bộ cài đặt chuyên nghiệp khác là **InstallShield**.

7.1. Tạo bộ cài đặt với công cụ có sẵn **Package and Deployment**

7.1.1. Giới thiệu chung

Đây là công cụ có sẵn trong bộ visual studio, nó cho phép người dùng tạo ra bộ cài đặt cho ứng dụng của mình chỉ thông qua một số bước trả lời câu hỏi.

Bạn khởi động ứng dụng trong mục Microsoft Visual Studio 6.0 Tools => Package and Deployment Wizard nằm trong mục Microsoft Visual Studio 6.0 trong Programs của menu START.

Trước khi tạo bộ cài đặt bạn cần biên dịch ứng dụng tạo ra tệp thi hành (*.exe).

Package and Deployment wizard cho phép 2 tùy chọn để tạo ra bộ cài đặt ứng dụng.

- Bộ cài đặt tiêu chuẩn được thiết kế để chạy trực tiếp trên máy tính của người dùng cuối thông qua tệp setup.exe;
- Bộ cài đặt thông qua mạng internet thiết kế để được tải về từ website dưới dạng một tệp thi hành duy nhất.
- Một tập hợp các tệp tin liên quan (Dependency files) để có thể thực hiện việc cài đặt thủ công trên máy người dùng (chú ý: sao chép, đăng ký các tệp thành phần vào các vị trí phù hợp trên hệ thống).

Các thành phần của bộ cài đặt hoàn thành được để trong một tệp nén (*.cab), bạn có thể lựa chọn tạo ra nhiều tệp nén để có thể chứa trong các phương tiện lưu trữ dung lượng thấp.

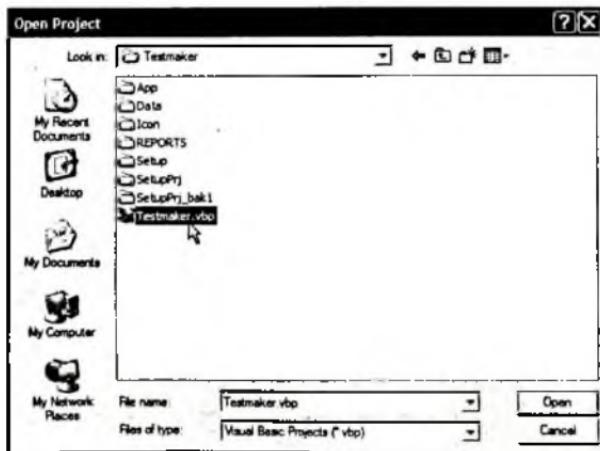
7.1.2. Thực hành tạo bộ cài đặt

Chúng ta sẽ cùng nhau tạo bộ cài đặt cho phần mềm TestMaker 2008.

7.1.2.1. Khởi động tiện ích



Chọn nút **Browse...** và tìm đến thư mục chứa tệp dự án (*.vbp) của ứng dụng muốn tạo bộ cài đặt.



Và bây giờ đường dẫn tới tệp dự án đã được điền vào ô select project



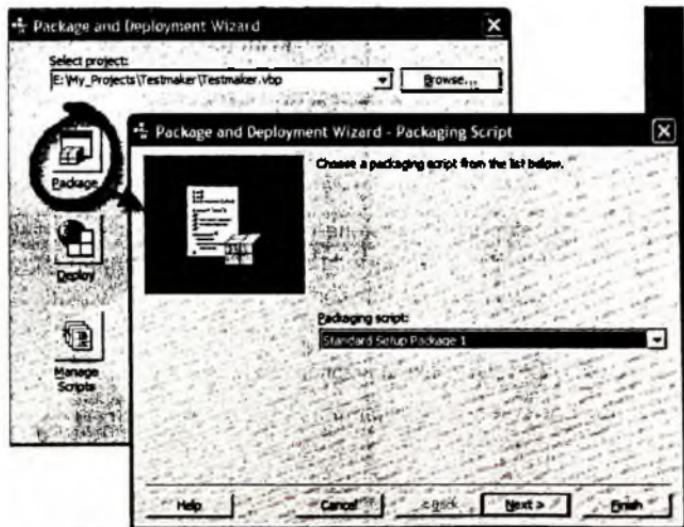
7.1.2.2. Biên dịch lại phần mềm

Lựa chọn nút Package sẽ đưa bạn đến bước đầu tiên trong quá trình đóng gói phần mềm của bạn. Khi kết thúc quá trình đóng gói bạn có thể quay lại cửa sổ giao tiếp này chọn nút Deploy để triển khai việc phân phối bộ cài đặt phần mềm vừa được tạo ra.

Nút Deploy có chức năng phân phối bộ cài đặt phần mềm vừa được tạo ra. Bạn có thể lựa chọn việc phân phối bộ cài đặt qua đĩa mềm, qua mạng cục bộ, qua một thư mục trên máy hoặc là qua internet.

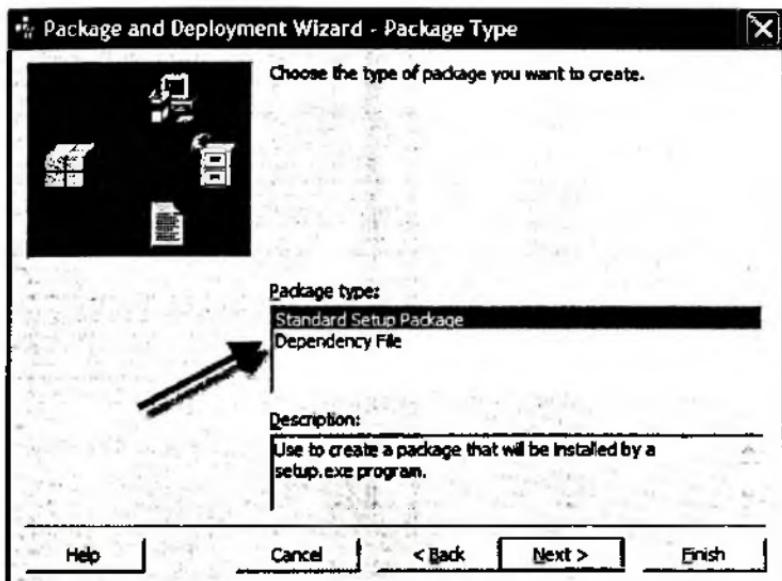
Nút Manager Script có chức năng hiển thị các Scripts Quản lý hộp thoại mà bạn có thể đổi tên, nhân bản, hoặc biên dịch lại các kịch bản triển khai.

Trong ví dụ này, chúng ta sẽ chọn nút Package để bắt đầu đi làm bộ cài đặt cho phần mềm.



Trong cửa sổ tiếp theo hiện ra, hệ thống đề nghị bạn lựa chọn một kịch bản đóng gói phần mềm (packaging script) đã được lưu trữ lại trong những lần tạo bộ cài đặt trước đây hoặc nếu không thì bạn sẽ lựa chọn kịch bản mặc định được đề nghị (standard setup package 1). Bấm nút Next để tiếp tục.

7.1.2.3. Lựa chọn thể loại của bộ cài đặt sẽ được tạo



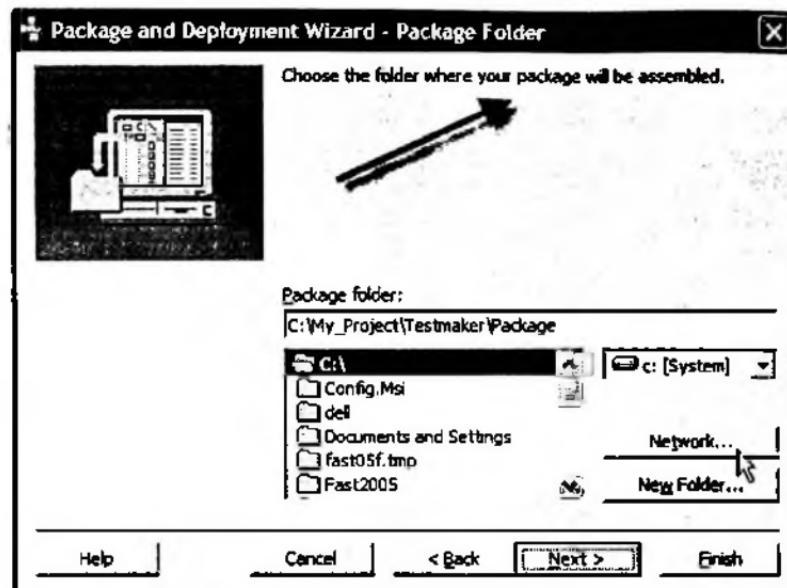
Trong mục Package type, bạn lựa chọn:

Standard Setup Package để tạo ra một bộ cài đặt tiêu chuẩn với tệp Setup.exe để thi hành

Dependency file để chỉ tạo ra một tập hợp các tệp liên quan đến ứng dụng và việc cài đặt ứng dụng sẽ được thực hiện chủ động thông qua việc sao chép các tệp liên quan vào các vị trí thích hợp trên hệ thống.

Trong cửa sổ này, bạn chọn Standard Setup Package và bấm nút Next để tiếp tục.

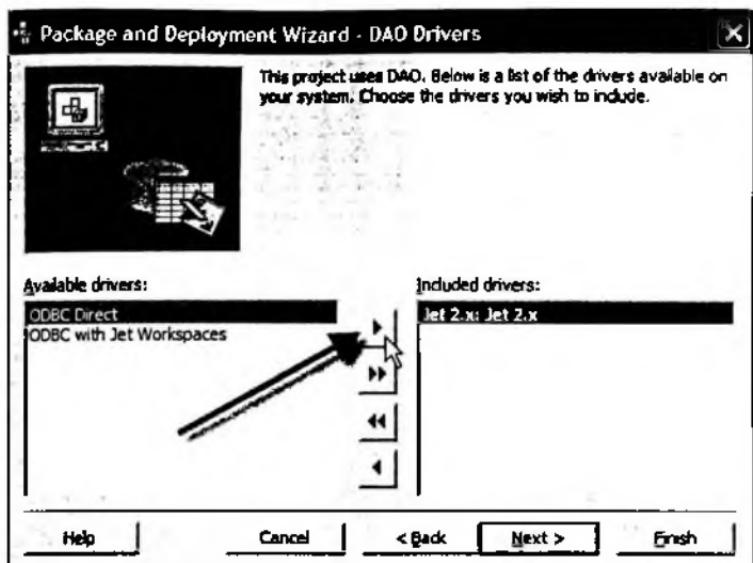
7.1.2.4. Lựa chọn thư mục nơi sẽ chứa bộ cài đặt được tạo ra



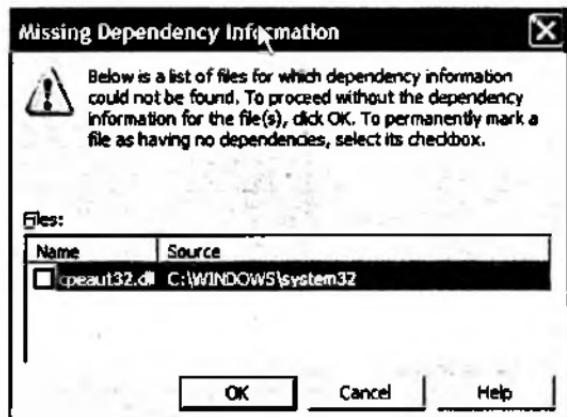
Bấm nút Next để tiếp tục.

7.1.2.5. Lựa chọn các gói thư viện có sẵn mà sẽ bao gồm trong bộ cài đặt

Vì phần mềm ứng dụng của chúng ta có sử dụng phương thức kết nối và khai thác cơ sở dữ liệu DAO (Data access object) cho nên hệ thống sẽ đề nghị lựa chọn một hoặc nhiều gói các tệp thư viện có sẵn sẽ được thêm vào bộ cài đặt. Trong tình huống này, hệ thống đã tự chọn cho chúng ta gói thư viện Jet 2.x; ta có thể chọn thêm 1 hay 2 gói ở cửa sổ bên trái, bấm nút > để thêm vào phần cửa sổ bên phải. Bấm nút Next để tiếp tục.



7.1.2.6. Hoàn thiện hồ sơ bộ cài đặt



Một cửa sổ liệt kê danh sách các tệp thư viện liên quan đến ứng dụng nhưng mà lại không được tìm thấy trong môi trường phát triển phần mềm. Để tiếp tục tạo bộ cài đặt mà không có các tệp bị thiếu được liệt kê ta bấm nút OK.

Nếu bạn biết rằng một hay nhiều các tệp được liệt kê thực sự không liên quan (không cần thiết) với phần mềm ứng dụng thì hãy đánh dấu lựa chọn vào tên các tệp không liên quan được liệt kê.

7.1.2.7. Kiểm tra và lập danh sách các tệp liên quan

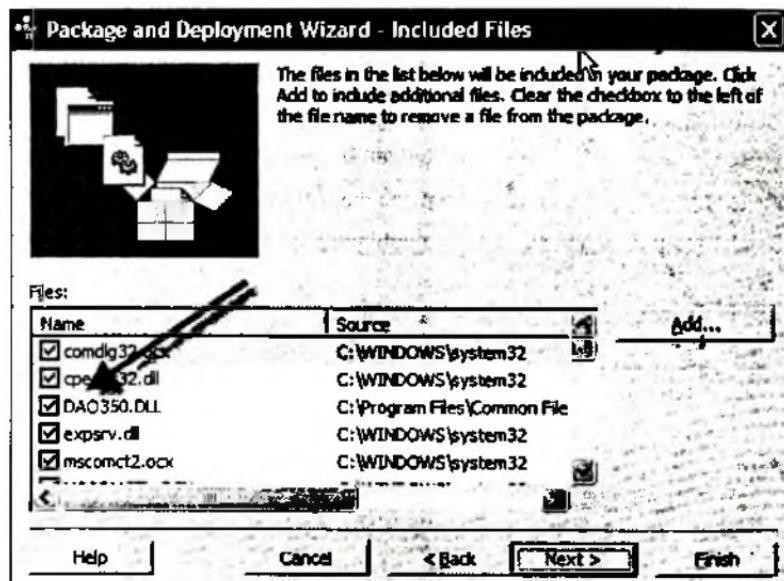
Máy tính sẽ tự động phân tích ứng dụng của bạn để xác định một danh sách các tệp liên quan bao gồm:

Tệp thi hành (*.exe)

Tệp cơ sở dữ liệu

Tệp trợ giúp (help file)

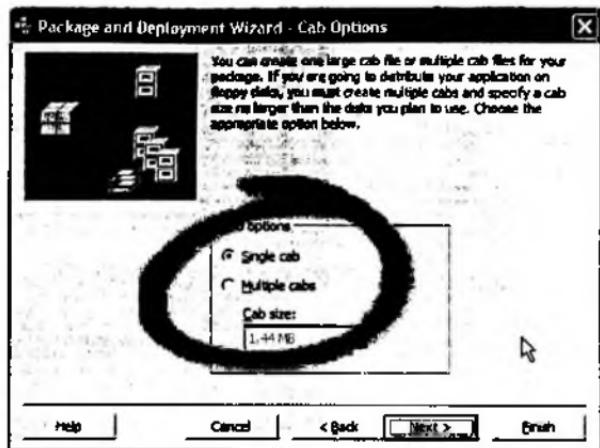
Tệp thư viện...



Một danh sách các tệp liên quan được liệt kê, bạn cần kiểm tra lại để hủy bỏ sự đánh dấu với những tệp được liệt kê nhưng không cần thiết cho ứng dụng hoặc bấm nút Add... để thêm vào các tệp liên quan cần thiết nhưng còn thiếu, chưa có trong danh sách.

Bấm nút Next để tiếp tục.

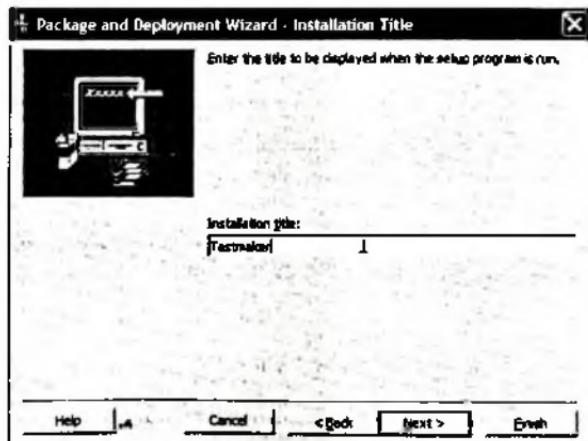
7.1.2.8. Lựa chọn hình thức của tệp nén (cab) sẽ được tạo ra



- Bạn chọn là Single cab để chỉ tạo ra một tệp nén duy nhất có dung lượng to hay nhỏ tùy thuộc vào quy mô của phần mềm.
- Bạn chọn là Multiple cab để tạo ra nhiều tệp nén có kích thước giới hạn để có thể chứa trong đĩa mềm (1.44mb) hay trên đĩa CD (650mb).

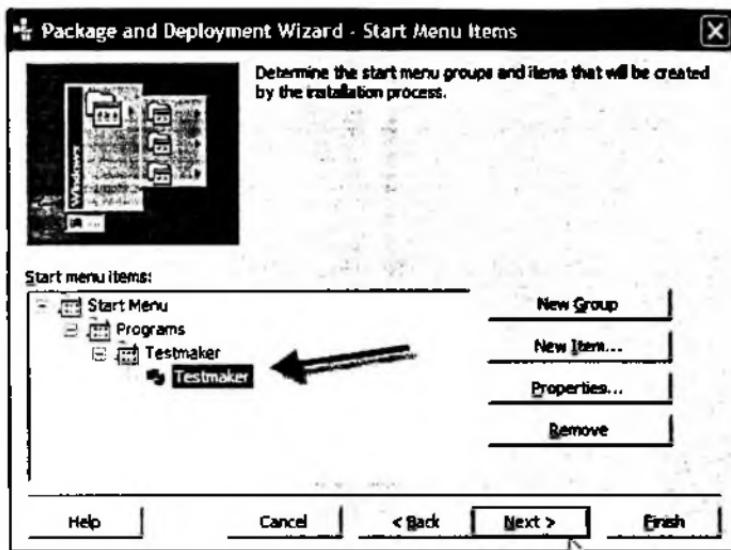
*Bấm Next để tiếp tục.

7.1.2.9. Nhập tiêu đề cho bộ cài đặt



Bấm Next để tiếp tục.

7.1.2.10. Thiết kế shortcut cho tệp thi hành



Xác định nhóm thực đơn (menu groups) và các đè mục (short cut) trong nhóm (items) của menu START sẽ được tạo ra trên máy của người dùng cuối khi tiến hành cài đặt phần mềm.

Như trong cửa sổ trên, sau khi người dùng cài đặt thành công phần mềm ứng dụng thì sau đó có thể khởi động phần mềm bằng cách lựa chọn theo đường dẫn:

Menu START=> Programs => Testmaker => Testmaker.

Với các nút chức năng ở phần cửa sổ bên phải, bạn có thể điều chỉnh mọi thứ theo ý mình:

Nút New Group để tạo ra một Group mới trong menu START

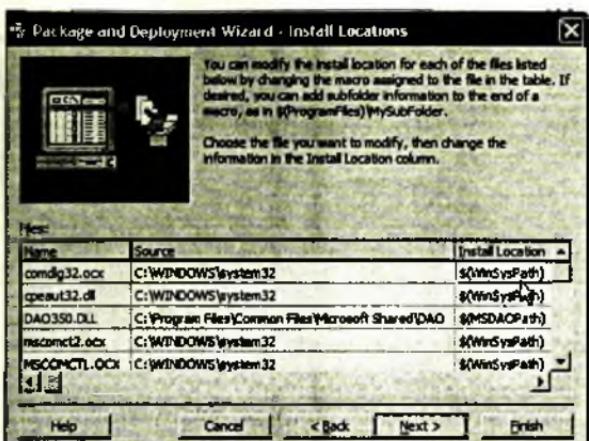
Nút New Item để tạo ra một shortcut mới trong một nhóm lựa chọn

Nút Properties để hiện cửa sổ thiết lập thuộc tính cho shortcut

Nút Remove để xóa một nhóm lựa chọn hay một shortcut trong nhóm.

Bấm Next để tiếp tục.

7.1.2.11. Lựa chọn vị trí cho các tệp thư viện sẽ được cài đặt trên máy người dùng.

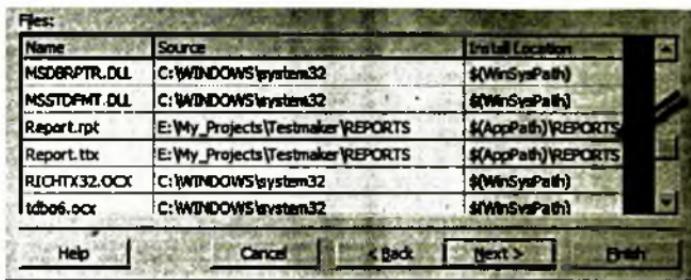


Với mỗi tệp thư viện trong danh sách, tại cột Install Location ta biết được vị trí của nó trên máy người dùng khi phần mềm được cài đặt.

Ta có thể thay đổi lại vị trí cài đặt này thông qua việc lựa chọn các giá trị trong danh sách:

- WinSysPath chỉ thư mục System32 trong window (C:\WINDOWS\system32)
- WinPath chỉ thư mục Windows trên ổ đĩa hệ thống (C:\WINDOWS)
- CommonFiles chỉ thư mục CommonFiles trên ổ đĩa hệ thống (C:\Program Files\Common Files)
- CommonFilesSys chỉ thư mục System trong thư mục CommonFiles trên ổ đĩa hệ thống (C:\Program Files\Common Files\System)
- ProgramFiles chỉ thư mục ProgramFiles (C:\Program Files)
- AppPath chỉ thư mục của ứng dụng sẽ được cài đặt trên máy người dùng. Mặc định thì thư mục được đề nghị sẽ là C:\Program Files\[Tên ứng dụng] ví dụ như C:\Program Files\Testmaker và người dùng hoàn toàn có thể thay đổi đường dẫn này nếu muốn. Thường thì trong thư mục cài đặt của ứng dụng có thêm 2 thư mục con là Data để chứa các tệp cơ sở dữ liệu và REPORTS để chứa các tệp mẫu báo cáo. Vì vậy các tệp cơ sở dữ liệu hay

tệp báo cáo của ứng dụng sẽ được chúng ta thiết lập cài đặt ở vị trí được khai báo AppPath\DATA và AppPath\REPORTS

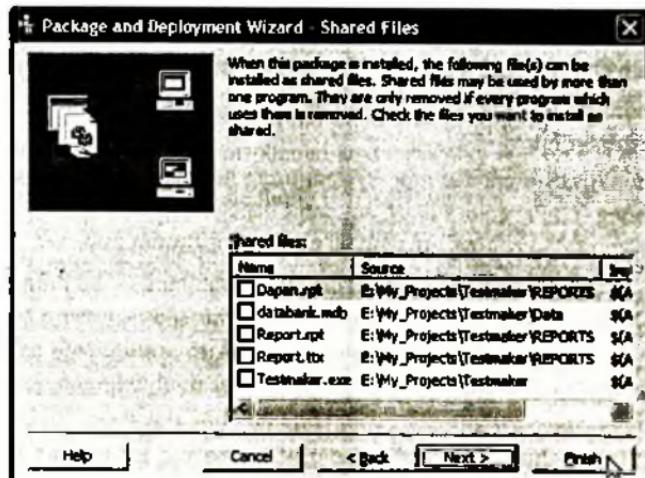


- MSDAOPath chỉ thư mục chứa thư viện DAO của hệ thống (C:\Program Files\Common Files\Microsoft Shared\DAO)

- Font chỉ thư mục Font của hệ thống (C:\WINDOWS\Fonts)

7.1.2.12. Xác định danh sách các tệp có thể chia sẻ

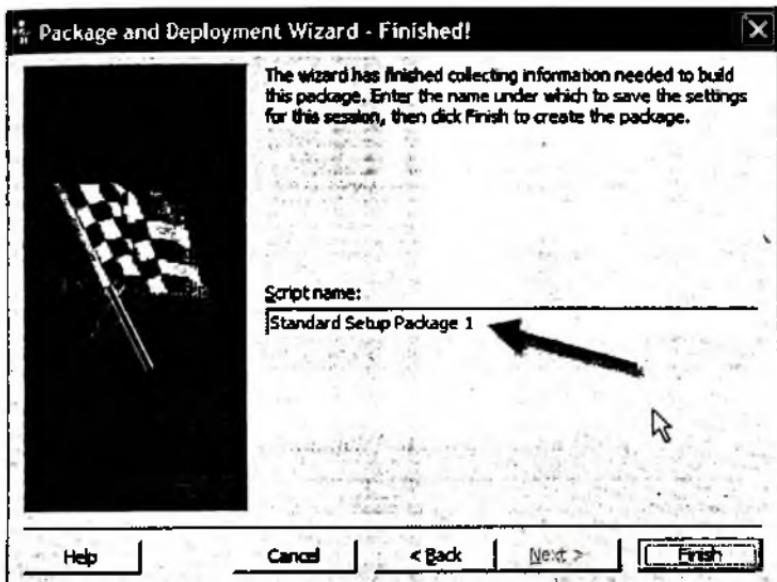
Máy tính liệt kê một danh sách các tệp bao gồm: tệp thi hành, tệp cơ sở dữ liệu; tệp báo cáo của ứng dụng và đề nghị chúng ta đánh dấu lựa chọn vào những tệp sẽ được cài đặt dưới dạng là tệp chia sẻ (shared files). Tệp chia sẻ là tệp có thể được sử dụng bởi nhiều hơn 1 phần mềm ứng dụng và nó sẽ chỉ bị gỡ bỏ nếu người dùng gỡ bỏ tất cả những ứng dụng cùng chia sẻ tệp này.



Trong tình huống này chúng ta sẽ không chọn tệp nào cả.

Bấm nút Next để tiếp tục.

7.1.2.13. Đặt tên cho tệp chứa kịch bản của quá trình tạo bộ cài đặt

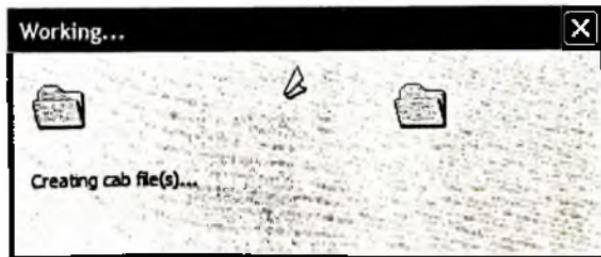


Nếu muốn bạn có thể đặt tên cho kịch bản cài đặt vừa được thiết lập tại ô script name để kịch bản đó sẽ có thể được sử dụng lại khi có nhu cầu tạo một bộ cài đặt cho một ứng dụng khác tương tự.

Bấm Finish để hoàn tất.

7.1.2.14. Tạo bộ cài đặt

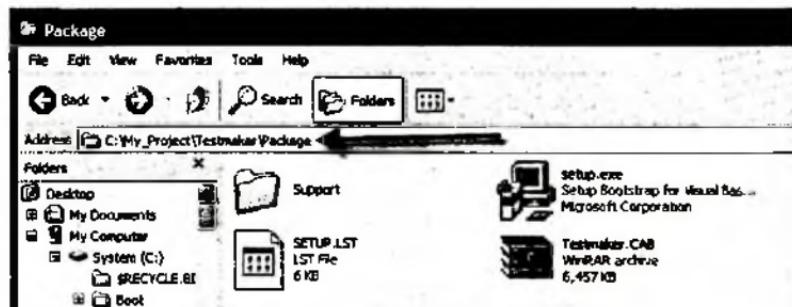
Quá trình tạo bộ cài đặt diễn ra



Và kết thúc khi trên màn hình hiện bảng thông báo



Theo đường dẫn được nhắc lại trong báo cáo, ta có bộ cài đặt đã được hoàn thành.



Nhận xét:

- Công cụ Package and Deployment có sẵn trong bộ chương trình Visual Studio cho phép chúng ta tạo ra bộ cài đặt cho phần mềm ứng dụng rất đơn giản tuy nhiên bộ cài đặt được tạo ra sẽ chỉ hoạt động tốt (ứng dụng được cài đặt và hoạt động tốt) nếu phần mềm chỉ được tạo ra sử dụng những công

cụ, điều khiển có sẵn của bộ Visual Studio. Nếu phần mềm có sử dụng những công cụ, điều khiển được cung cấp bởi một hãng phần mềm thứ 3 thì sẽ dẫn đến việc bị thiếu các tệp thư viện cần thiết trong bộ cài đặt (ứng dụng được cài đặt sẽ không thể hoạt động được).

- Việc tạo bộ cài đặt là một việc làm đòi hỏi sự công phu mà nếu chỉ làm một lần thì thường sẽ không thành công. Theo kinh nghiệm, chúng ta sẽ thử nghiệm bộ cài đặt được tạo ra trên một máy tính mới chỉ được cài hệ điều hành (chưa được cài đặt phần mềm ứng dụng nào). Nếu thất bại thì ghi chép tên các tệp thư viện bị thiếu được thông báo để làm cơ sở điều chỉnh lại bộ cài đặt và tiến hành thử nghiệm lần sau. Cứ làm như vậy cho đến khi phần mềm ứng dụng được cài đặt và hoạt động thành công.

7.2. Tạo bộ cài đặt với công cụ chuyên nghiệp InstallShield

7.2.1. Giới thiệu công cụ InstallShield 2008 Premiere

InstallShield 2008 Premiere - Phiên bản phổ biến, nhiều chức năng và mạnh mẽ nhất cho các nhà phát triển các bộ cài đặt. Nó cho phép các nhà sản xuất phần mềm mọi qui mô tạo các bộ cài đặt chất lượng cao trên nền Windows, bao gồm Windows Vista™ và Win7, đồng thời mở rộng ra cả các thiết lập tinh chỉnh cơ sở dữ liệu trên máy chủ, các dịch vụ web và các thiết bị di động.

Theo IBM, 28% lỗi ứng dụng liên quan đến quá trình cài đặt và chỉnh sửa lỗi. Tuy nhiên nhiều công ty phần mềm vẫn tiếp tục lệ thuộc vào các bộ cài đặt riêng đã lỗi thời; điều này làm tăng chi phí hỗ trợ kỹ thuật và làm cho người dùng bối rối.

Được triển khai ở hơn 500 triệu máy tính cá nhân toàn thế giới, công nghệ InstallShield là sự kết hợp sức mạnh, tính linh hoạt và dễ sử dụng. Nó cung cấp một phương thức đơn giản cho phép tạo ra các bộ cài đặt chuyên nghiệp và đáng tin cậy cho cả những ứng dụng phức tạp nhất, đảm bảo cho người dùng luôn an tâm và giảm chi phí hỗ trợ kỹ thuật. InstallShield 2008 cho ra mắt 3 phiên bản: Premier, Professional và Express. Phần mềm hỗ trợ 3 ngôn ngữ: tiếng Anh, tiếng Đức và tiếng Nhật.

Một số ưu điểm nổi bật:

Tối ưu hóa bộ cài đặt cho Windows 7: InstallShield 2008 hỗ trợ tương thích với UAC, Restart Manager, quá trình kiểm tra bàn quyền của Windows Vista, Windows Installer 4.0, mở rộng khả năng đóng dấu điện tử và hơn thế nữa.

Tương thích với các công nghệ tiên tiến nhất từ Microsoft: Bao gồm .NET Framework 3.0, Visual Studio 2005, IIS 7.0, SQL Server 2005, Windows Server "Longhorn", Direct X 9.0c, Windows Mobile platforms gồm Windows Embedded CE 6.x,..

Giảm thời gian phát triển ứng dụng:

- Công nghệ IDE mạnh mẽ, tiến trình cài đặt tự động, các khuôn mẫu và các vật thể tạo sẵn giúp người dùng tiết kiệm đáng kể thời gian và công sức.
- Công cụ trợ giúp hướng dẫn người dùng từng bước qua quá trình xây dựng một bộ cài đặt, vừa tăng hiệu quả làm việc vừa giúp giảm thiểu lỗi.
- Bộ InstallShield Best Practices Validation giúp người dùng tránh được những vấn đề thường gặp có thể gây trì trệ.

Giảm thiểu chi phí hỗ trợ kĩ thuật với các bộ cài đặt đáng tin cậy: InstallShield 2008 với các bản vá lỗi hay nâng cấp được đơn giản hóa, cơ chế hỗ trợ hệ thống được dụng riêng biệt, bộ máy dò lỗi và các công cụ Build Report chắc chắn sẽ làm tăng chất lượng cho bộ cài đặt.

Khai thác tối đa hiệu quả làm việc nhóm với chức năng InstallShield Collaboration: InstallShield 2008 còn hỗ trợ chức năng InstallShield Collaboration, cho phép bạn thu lại, kiểm tra, bảo trì và sử dụng các yêu cầu cài đặt trong suốt quá trình phát triển.

InstallShield 2008 phiên bản Premier cho phép các nhà phát triển có thể:

- Tránh những vấn đề thường gặp trong quá trình cài đặt - các bộ kiểm tra (Validators) trong bộ chức năng InstallShield Best Practices validation mới cảnh báo cho bạn nếu có xung đột trong bộ cài đặt.
- Bảo trì một hệ thống gọn gàng - Với bộ máy Standalone Build, giờ đây bạn chỉ cần sử dụng thành phần của InstallShield được dùng để xây dựng bộ cài đặt.
- Tăng cường trải nghiệm cho người dùng cuối - với các thiết kế giao diện người dùng cuối được làm mới hoàn toàn.
- Dễ dàng chuyển đổi các bộ cài đặt cũ thành các gói cài đặt chuẩn Windows Installer thông qua công cụ InstallShield Repackager
- Tùy chọn ngôn ngữ cho bộ cài đặt - InstallShield hỗ trợ đa ngôn ngữ cho phép tạo các bộ cài đặt cho 33 ngôn ngữ chỉ trong một phiên cài đặt.
- Nhanh chóng phát hành bộ cài đặt mới - Với các cơ sở dữ liệu trên mạng, giờ đây thật dễ dàng xây dựng một cơ sở dữ liệu gồm các thành phần

phổ biến mà nhiều nhà phát triển có thể truy xuất và dùng lại trong các kế hoạch khác.

- Tạo cái bản thử nghiệm mà không phải sửa chữa mã nguồn - Với tùy chọn try-and-die của cơ chế tình hình ứng dụng thử nghiệm đơn giản, bạn dễ dàng cung cấp cho những người dùng tiềm năng phiên bản thử nghiệm với đầy đủ tính năng của sản phẩm (Sau một thời gian định trước bản thử nghiệm sẽ tự động hết hiệu lực trên máy tính của người dùng cuối).

- Khai thác triệt để hiệu quả hoạt động nhóm - thông qua sự trợ giúp của công nghệ InstallShield Collaboration, là một plug-in cho môi trường phát triển Visual Studio.

Các chức năng chính:

- Bộ InstallShield Best Practices Validation: giúp tránh các vấn đề thường gặp trong quá trình cài đặt bằng cách gửi cảnh báo từ bộ kiểm tra, nếu có xung đột trong bộ cài đặt.

- Cải thiện giao diện tự động hóa: dùng script để thêm tập tin mới, thêm và tháo bỏ các tính năng, khởi động quá trình xây dựng và thay đổi tên sản phẩm, mã nâng cấp, các hiệu chỉnh, thông tin tóm tắt, đánh dấu phiên bản và hơn thế nữa.

- Cải thiện các yêu cầu hệ thống trước khi cài đặt, người dùng có nhiều tùy chọn hơn trong việc xác lập chính xác những điều kiện cần có trước khi cài đặt trên từng phiên bản hệ điều hành khác nhau.

- Hỗ trợ XML, tiết kiệm thời gian thông qua việc kiểm tra nhanh chóng những thay đổi hiệu chỉnh XML trong quá trình cài đặt.

- Đòi hỏi người dùng đọc cam kết EULA, “ép buộc” người dùng phải cuộn xuống cuối cam kết EULA trước khi tiếp tục quá trình cài đặt.

- Tiến hành nhanh hơn, tăng cường khả năng xử lý số lượng tập tin lớn Cải thiện đáng kể hiệu suất khi làm việc với số lượng tập tin lớn thông qua trình Direct Editor hay với String Tables.

- Giao diện người dùng cuối được làm mới cho các bộ cài MSI cơ bản. Các thiết kế giao diện được làm mới cho các bộ cài MSI cơ bản đem đến nhiều lựa chọn mới mẻ cho giao diện người dùng của bộ cài đặt, cải thiện trải nghiệm người dùng cuối, bao gồm 19 giao diện cho phiên bản Premier và 5 cho Professional.

- Tăng cường hỗ trợ Windows Vista và Win7, hỗ trợ tự động cho Windows Vista và các công nghệ liên quan như UAC, Restart Manager, và

hơn thế nữa; đồng thời nhận hỗ trợ cho Windows Vista thông qua chức năng InstallScript và One-Click Install projects.

- Kiến trúc IDE View, dễ dàng tìm và thay đổi các thiết lập ở chế độ Releases view, giờ đây được sắp xếp lại từ chế độ duyệt lè đơn hay duyệt theo tab.

- Hỗ trợ 32bit và 64-Bit .NET Framework 2.0 và 3.0, nhận hỗ trợ cho tất cả phiên bản 32 và 64-bit của .NET 2.0 và .NET 3.0 trong cả các bộ cài MSI cơ bản và InstallScript MSI.

- Hỗ trợ công nghệ mới nhất của nền Windows Mobile, nhận hỗ trợ cho những công nghệ mới nhất trên nền Windows Mobile (bao gồm Windows Embedded CE 6.x) cho bộ cài Basic MSI (phiên bản Premier, Professional), InstallScript MSI (phiên bản Premier, Professional), Smart Device (phiên bản Premier, Professional, Express), và Express (phiên bản Express).

- Hỗ trợ các cập nhật cho Direct X 9.0 Object, giờ đây người dùng có thể sử dụng các object trên nền Windows Installer-based cho bộ cài Basic MSI và InstallScript MSI trong các bộ cài đặt nén hay cài ngầm.

- Hỗ trợ UAC, InstallShield cho phép tạo các bộ cài đặt có thể thông qua quá trình kiểm tra UAC yêu cầu xác nhận của người dùng bằng cách yêu cầu quyền truy cập 1 lần duy nhất cho tất cả bộ cài đặt và sử dụng quyền này xuyên suốt quá trình cài đặt.

- Hỗ trợ chữ ký điện tử, tiết kiệm thời gian bằng cách thêm chữ ký điện tử vào tất cả các tập tin cùng lúc trong quá trình tạo.

- Cải thiện khả năng sử dụng cho IDE. Dễ dàng xem các thông tin thêm vào và kiểm soát chặt chẽ hơn IDE và các giao diện tùy biến.

- Hỗ trợ SaaS (công nghệ IIS 7 và SSL), dễ dàng triển khai các ứng dụng web dựa trên nền Windows với cải thiện hỗ trợ cho SQL và các công nghệ web như IIS 7 và SSL.

Yêu cầu hệ thống:

- * Các máy tính Pentium III (500 MHz hay cao hơn)
- * 256 MB RAM (đề nghị: 512 MB)
- * Dung lượng đĩa cứng trống 500 MB
- * Màn hình thiết kế với độ phân giải XGA 1024 × 768 hay cao hơn
- * Hệ điều hành Windows 2000 SP3 hay mới hơn
- * Windows XP

- * Windows Server 2003
- * Windows Vista
- * Windows 7
- * Trình duyệt Microsoft Internet Explorer 5.01 (đè nghi: IE 5.5 hay mới hơn)
- * Quyền quản trị hệ thống
- * Chuột Microsoft IntelliMouse hay các thiết bị tương thích khác

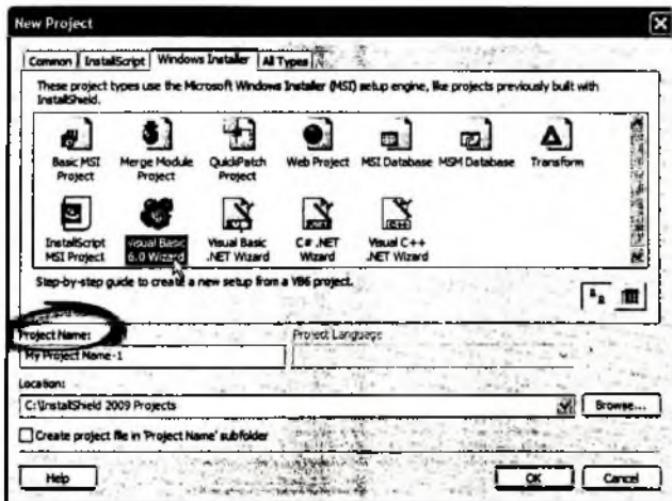
7.2.2. Thực hành tạo bộ cài đặt cho ứng dụng viết bằng Visual Basic 6

Tương tự như phần trên, chúng ta cũng sẽ tạo bộ cài đặt cho phần mềm TestMaker bằng công cụ InstallShield.

7.2.2.1. Khởi động và bắt đầu

- Khởi động InstallShield

Vào menu File chọn NEW... hoặc chọn mục chức năng "create a New Project" trong khung Project Task ở lề bên trái của màn hình.

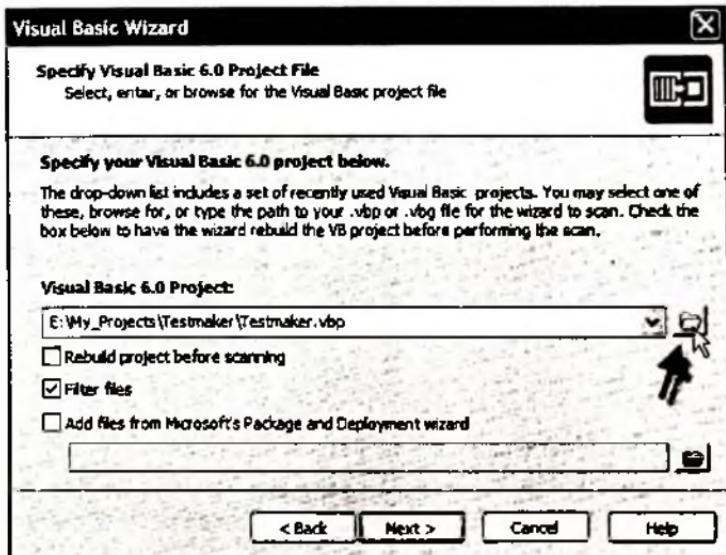


- Trong cửa sổ hiện ra, chọn thẻ Windows Installer và chọn Visual Basic 6.0 Wizard
- Tại mục Project name nhập tên của dự án phần mềm, đây sẽ chính là tên của thư mục chứa bộ cài đặt sẽ được tạo ra sau này trong thư mục chính là: C:\InstallShield 2009 Projects.

- Bấm OK và chúng ta sẽ bắt đầu các bước tạo ra bộ cài đặt cho phần mềm TestMakek với sự trợ giúp, hướng dẫn của Install shield



7.2.2.2. Khai báo đường dẫn đến thư mục chứa mã nguồn



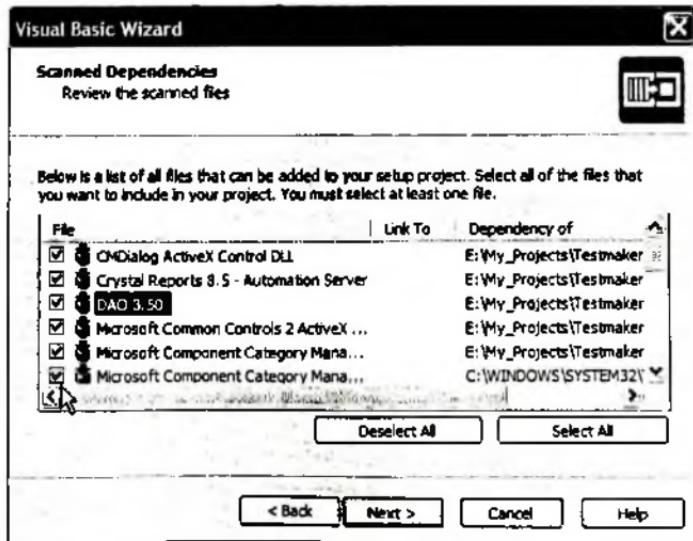
Bấm nút Browse để tìm đến thư mục chứa tệp dự án (*.vbp) của ứng dụng

- Tùy chọn “rebuild project before scanning” nghĩa là hệ thống sẽ tạo lại tệp thi hành (Testmaker.exe) cho dự án trước khi tìm kiếm và lập danh sách về các tệp thư viện có liên quan (nếu trước đó ta đã tạo tệp thi hành mới nhất cho dự án thì không cần chọn tùy chọn này).
- Tùy chọn “Filter file” liệt kê các tập tin phụ thuộc mà bạn không muốn thêm vào trình cài đặt. Ví dụ, tập tin hệ thống mà đã được cài sẵn trên máy người dùng thường là không cần phải cài đặt lại. Để tránh có những tập tin này được bổ sung vào bộ cài đặt của bạn ta sẽ chọn tùy chọn này.
- Tùy chọn “Add file from Microsoft....” để hệ thống sẽ tự động thêm danh sách các tệp liên quan mà đã được lập bằng công cụ có sẵn của bộ Visual Studio “Microsoft Package and Deployment wizard”.

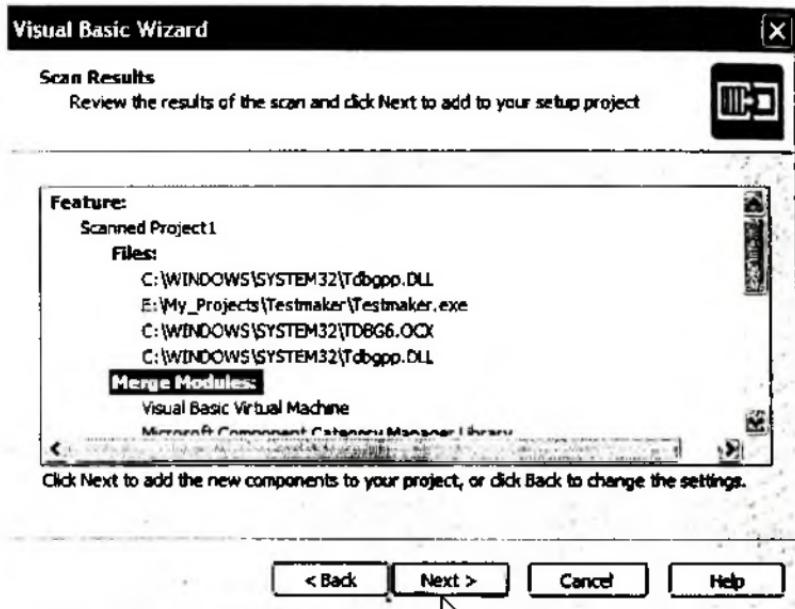
Bấm Next để hệ thống chuyển sang bước tiếp theo.

7.2.2.3. Lựa chọn nhóm các tệp liên quan sẽ được thêm vào bộ cài đặt

Sau khi thực hiện kiểm tra và lập danh sách, hệ thống hiển thị cửa sổ tiếp theo chứa danh sách các tệp liên quan đến dự án mà bạn có thể lựa chọn thêm vào bộ cài đặt.



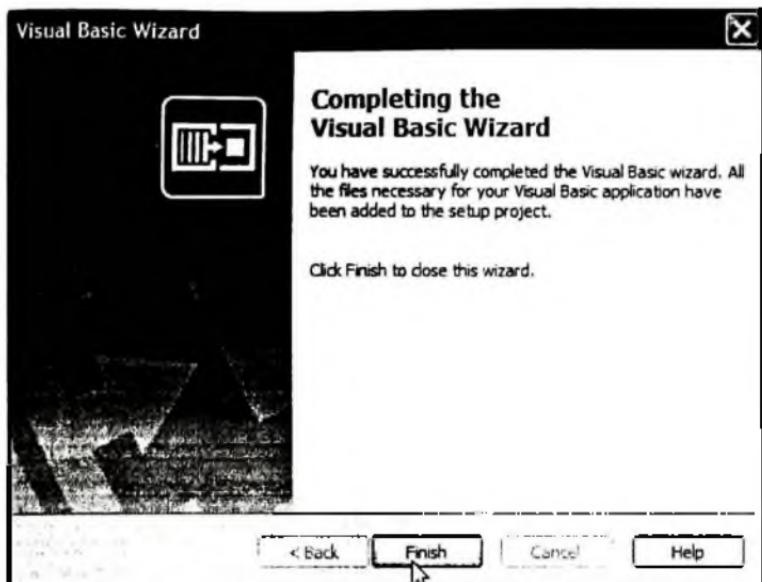
- Trong lần đầu tiên, khi chưa có kinh nghiệm, chúng ta có thể chọn thêm tất cả (Select All) các tệp liên quan trong danh sách vào bộ cài đặt (“thừa còn hơn thiếu !”)



Hệ thống hiển thị cửa sổ chứa báo cáo về những tệp liên quan đã được thêm vào bộ cài đặt.

Hoàn tất việc khai báo thông tin về dự án muốn tạo bộ cài đặt.

Theo lý thuyết thi đến bước này chúng ta đã có đủ thông tin để tạo ra bộ cài đặt cho ứng dụng. Trên thực tế, vì ứng dụng của ta thường được tạo ra với một hệ thống các thư mục con chứa các thành phần liên quan (cơ sở dữ liệu, mẫu báo cáo, các tệp hình ảnh,...) cùng với việc sử dụng một số công cụ của hãng thứ 3 (true grid, Spread, XP form,...) trong việc thiết kế giao diện. Vì vậy tiếp theo đây chúng ta sẽ chủ động thực hiện một số khai báo, thiết lập dựa trên hồ sơ đã được tạo ra qua các bước trên.



7.2.2.4 Khai báo thông tin tổng quát về ứng dụng



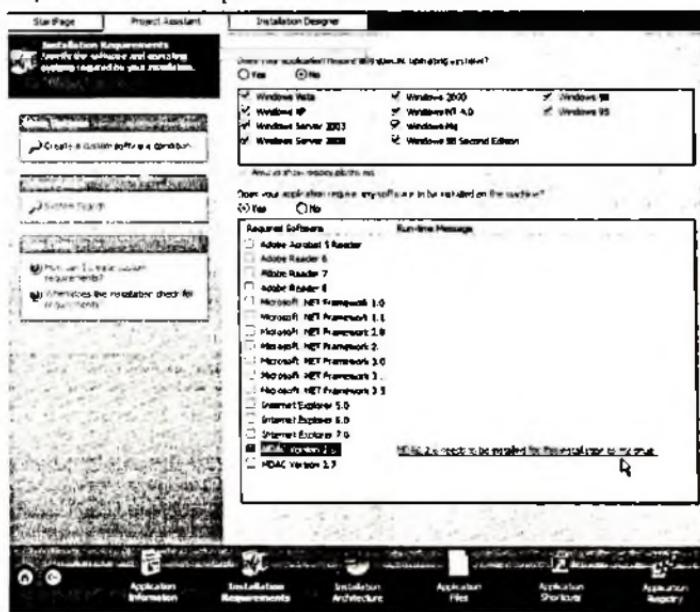
Tại thanh công cụ chỉ dẫn phía bên dưới màn hình, chúng ta sẽ lần lượt khai báo và thiết lập các thông tin cho bộ cái đặt hoàn thiện.

Trước hết, lựa chọn mục “Application Information” ta có màn hình như trên để thực hiện khai báo các thông tin về ứng dụng.

- Specify your company name: Nhập tên công ty của bạn
- Specify your application name: Nhập tên ứng dụng
- Specify your application version: nhập số phiên bản của ứng dụng
- Specify your company web address: Nhập địa chỉ website của công ty bạn
- Hai tùy chọn còn lại:
 - o “Bạn có muốn thông báo cho người dùng cuối mỗi khi bạn tạo ra bản cập nhật mới cho phần mềm?” Yes/ No
 - o “Lựa chọn biểu tượng sẽ hiển thị cùng ứng dụng của bạn trong phần quản lý “add or remove programs” của windows”, nếu muốn bạn có thể bấm nút Browse... và lựa chọn một biểu tượng mới thay thế cho biểu tượng mặc định.

7.2.2.5. Khai báo các môi trường và nền tảng cần có để ứng dụng hoạt động

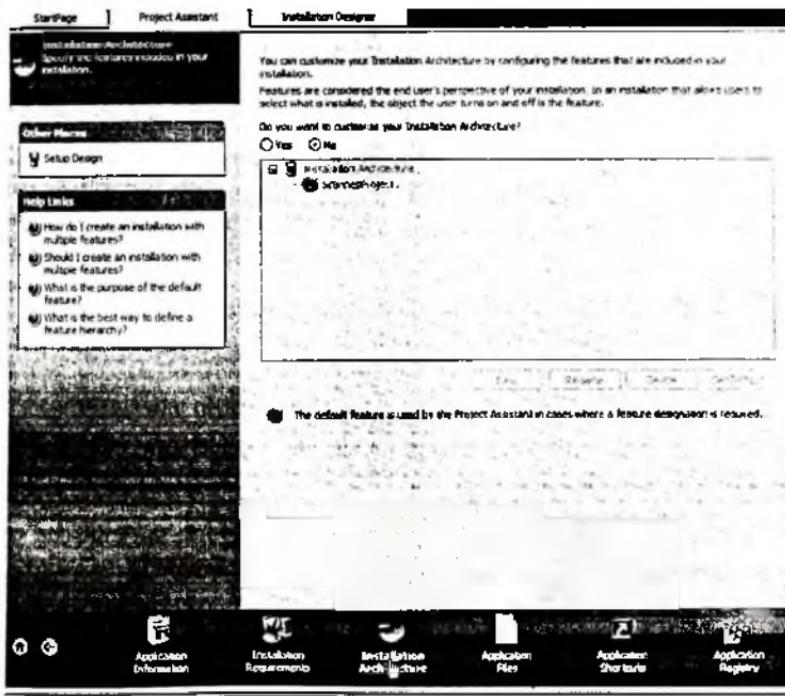
Lựa chọn Installation Requirements ta có màn hình



- Nếu ứng dụng của bạn chỉ có thể chạy trên một hệ điều hành cụ thể nào đó thì ta sẽ lựa chọn hệ điều hành tương ứng trong danh sách, nếu không có gì đặc biệt thì cứ để mặc định là chọn tất.
- Nếu ứng dụng của bạn được phát triển dựa trên nền tảng công nghệ nào đó thì ta phải lựa chọn nó trong danh sách “required software” như là một điều kiện bắt buộc cần phải được cài đặt trước trên máy người dùng cuối trước khi có thể cài đặt ứng dụng.

Với ứng dụng được viết bởi visual basic 6.0 thi ta chỉ cần đòi hỏi trên máy người dùng cuối phải có sẵn “MDAC Version 2.6” (bộ công cụ truy suất dữ liệu, có sẵn nếu hệ thống có cài đặt offices 2003)

7.2.2.6. Tùy chọn về kiến trúc của hồ sơ tạo bộ cài đặt

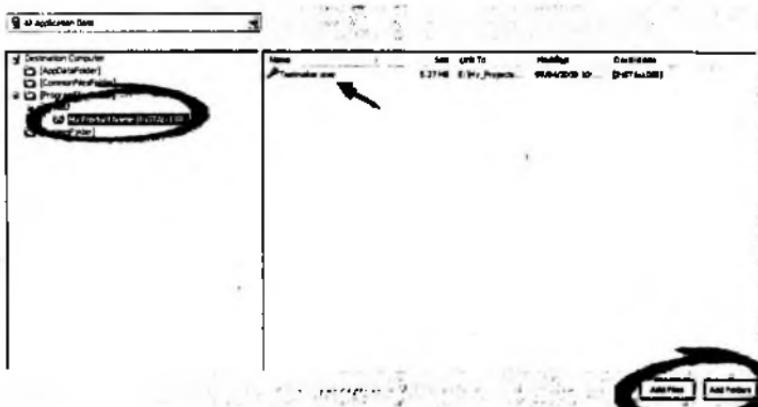


- Trong mục thứ 3 “Installation Architecture” hệ thống cho phép chúng ta lựa chọn thay đổi cấu trúc của hồ sơ tạo bộ cài đặt.

- Với ứng dụng viết trong vb6 chúng ta giữ nguyên tùy chọn mặc định trong mục này.

7.2.2.7. Xây dựng cấu trúc thư mục và khai báo các tệp liên quan

Tại mục thứ 4 “Application Files”

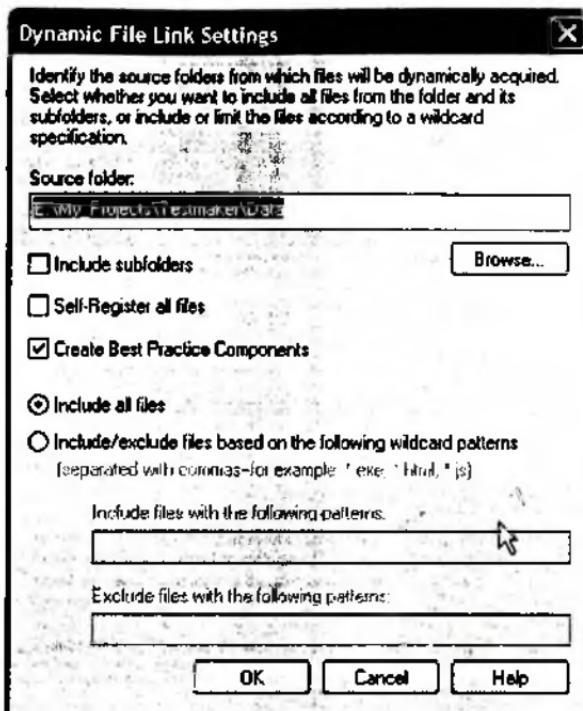


Trong hình trên chúng ta thấy thư mục chứa ứng dụng sẽ được cài đặt tại máy tính người dùng cuối (INSTALLDIR) mới chỉ có duy nhất một tệp thi hành của ứng dụng (testmaker.exe). Thường thì ứng dụng của chúng ta bao gồm một hệ thống thư mục con phía dưới chứa các tệp liên quan:



Trong bước này, chúng ta sẽ lần lượt tạo ra hệ thống các thư mục con và thêm danh sách các tệp liên quan và các thư mục con tương ứng.

- Tại góc phía dưới bên phải màn hình, chọn nút Add Folders để thêm vào hồ sơ tạo bộ cài đặt một thư mục cùng với các tệp chứa trong nó.



Như hình trên, ta đã chọn thêm thư mục Data tại đường dẫn E:\My_Projects\TestMaker vào hồ sơ tạo bộ cài đặt. Hệ thống đề nghị sử dụng cơ chế “Liên kết files linh hoạt” để bảo đảm sau này khi tạo lại bộ cài đặt thì mọi thay đổi đã được tạo ra đối với thư mục Data sẽ được cập nhật vào trong hồ sơ.

- Tùy chọn **Include subfolders** để cho phép thêm vào hồ sơ các tệp và cả những thư mục con chứa trong thư mục được chỉ định.
- Tùy chọn **Self-Register all files** để cho phép tự đăng ký vào registry của hệ thống tất cả các tệp trong thư mục (ta chỉ chọn tùy chọn này nếu trong thư mục được thêm vào hồ sơ chứa các tệp thư viện của hãng thứ 3 hoặc thư viện do người dùng tạo ra).
- Tùy chọn “**Create Best Practice Components**” là tùy chọn mặc định để áp dụng những cách thức cài đặt an toàn, không gây xung đột khi cài đặt thư mục cùng ứng dụng.

- Tùy chọn “Include all files” để bao gồm tất cả các tệp chứa trong thư mục sẽ được thêm vào hồ sơ dự án cài đặt
- Tùy chọn “Include/exclude....” Cho phép chỉ định phần mở rộng của những tệp sẽ được thêm/hoặc không được thêm vào thư mục trong hồ sơ cài đặt.

Sau khi hoàn thành việc khai báo thêm các thư mục con của ứng dụng vào hồ sơ dự án, màn hình như sau:



7.2.2.8. Sử dụng công cụ “Dynamic scan” và “Static scan”

Một trong những vấn đề hay gặp khi làm bộ cài đặt đó là chúng ta không khai báo đủ các tệp thư viện cần thiết trong bộ cài khiến cho sau khi cài đặt mặc dù vẫn khởi động được ứng dụng nhưng khi thi hành một chức năng nào đó của ứng dụng mà có sử dụng đến tệp thư viện bị thiếu thì lập tức sẽ bị báo lỗi và thoát ra ngoài.

Công cụ “Static scan” sẽ tiến hành kiểm tra tệp thi hành của ứng dụng để liệt kê các tệp thư viện có liên quan, cần thiết cho việc thi hành ứng dụng. Thực ra công cụ này đã được thi hành ngay sau bước đầu tiên khi chúng ta khai báo đường dẫn đến thư mục chứa mã nguồn của ứng dụng.

Công cụ “Dynamic Scan” sẽ kiểm tra và lập danh sách các tệp thư viện liên quan đến ứng dụng bằng cách cho ứng dụng hoạt động (người sử dụng cần thiết phải mở mọi mục chức năng của ứng dụng) và ghi nhận những tệp thư viện được gọi vào bộ nhớ trong suốt quá trình ứng dụng hoạt động. Quá trình Dynamic Scan sẽ chấm dứt khi người dùng kết thúc ứng dụng.

Ở bước này, chúng ta sẽ chỉ sử dụng công cụ Dynamic scan để tìm kiếm những tệp thư viện cần thiết còn thiếu.

Vào menu Project chọn “perform Dinamic scan”



Dynamic Scanning Wizard

Filter Files

Choose to filter files



You have the option to filter (exclude) files automatically during the scan of your project. By selecting the check box below, the files specified in filters.xml will be excluded from the project.

The files listed in filters.xml are common system files, such as Kernel32.dll, GDI32.dll, and Ole32.dll. It can be edited to filter additional files.

Filtering files using the preconfigured filters.xml is recommended.

Filter files

< Back

Next >

Cancel

Help

Dynamic Scanning Wizard

Specify the executable

Select an EXE from the project or a new EXE



I would like to select an executable file from my project (recommended)

Choose this option to have the wizard associate all dependency files with your executable's feature.

I would like to select a new executable

Choose this option to have a new feature created. This feature will contain your executable and all of its dependencies.

Click Next to proceed or Cancel to exit.

< Back

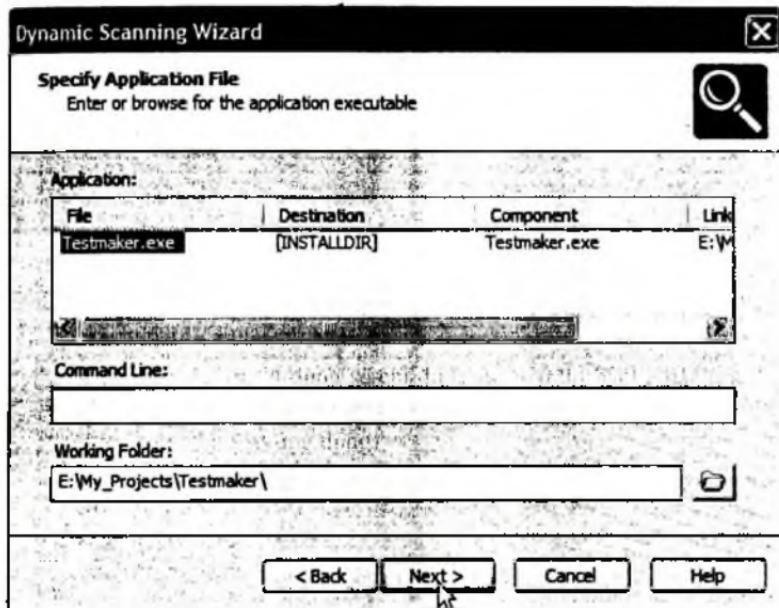
Next >

Cancel

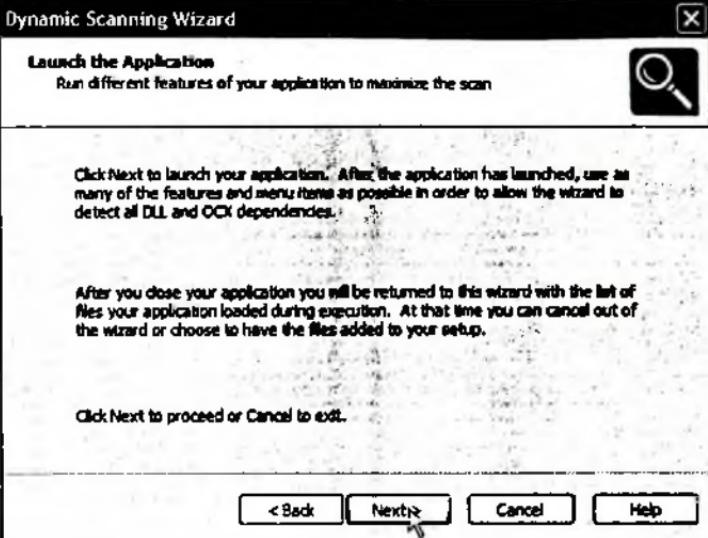
Help

Màn hình chứa 2 tùy chọn để xác định tệp thi hành của ứng dụng sẽ được khởi động ngay sau đây.

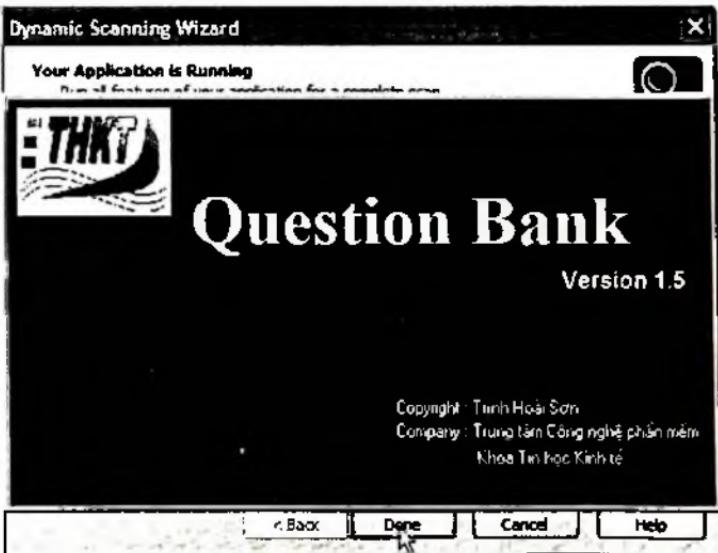
- Tùy chọn 1: Sử dụng ngay tệp thi hành đã được khai báo trong hồ sơ bộ cài đặt (testmaker.exe)
- Tùy chọn 2: Người dùng sẽ chỉ định một tệp thi hành khác (chi lựa chọn tùy chọn này nếu vừa có thêm chức năng hay thuộc tính mới được cập nhật vào ứng dụng)

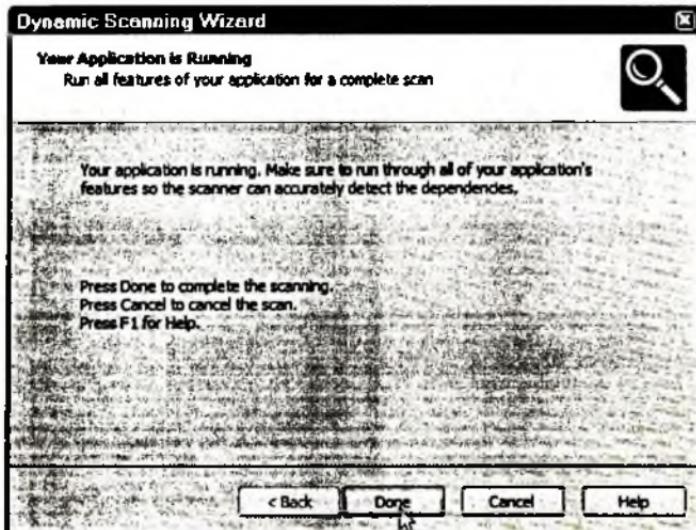
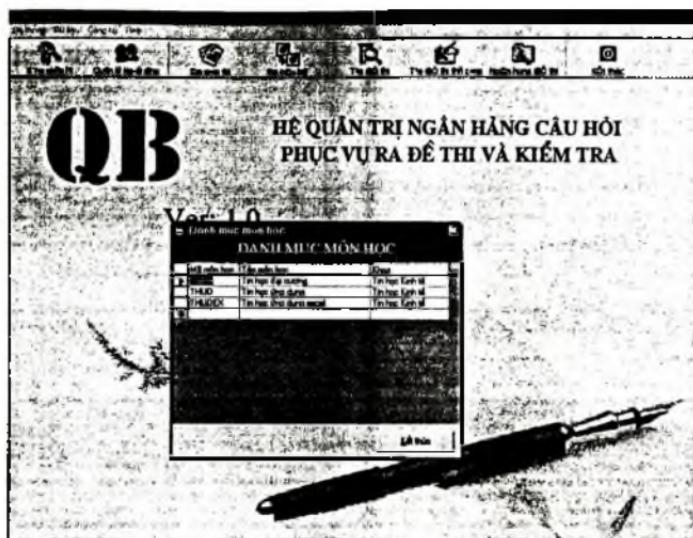


Kiểm tra và xác nhận đường dẫn đến tệp thi hành của ứng dụng

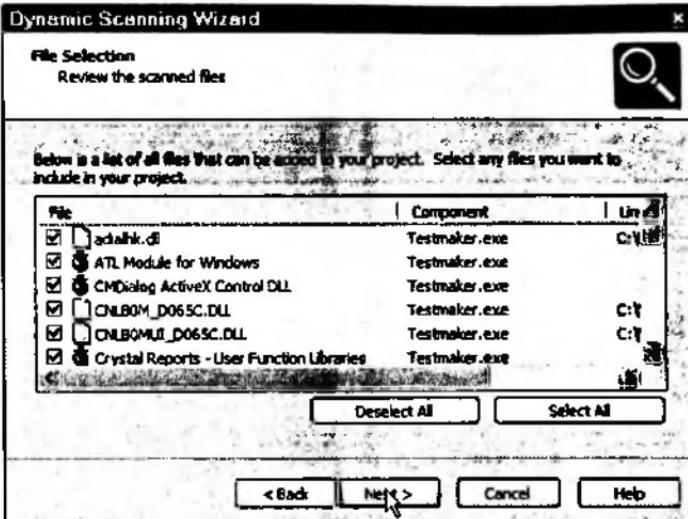


Sau bước này ứng dụng sẽ được khởi động cùng với gợi ý là bạn nên cho thi hành mọi chức năng của ứng dụng để có kết quả tốt nhất.





Sau khi chạy xong một lượt tất cả các chức năng của ứng dụng, bạn kết thúc ứng dụng và quay trở lại màn hình giao tiếp của Installshield. Bấm vào nút Done để chính thức kết thúc quá trình theo dõi (scan).



Trong cửa sổ tiếp theo, hệ thống liệt kê danh sách tất cả các tệp thư viện đã được gọi đến trong suốt quá trình chạy ứng dụng lúc trước.

Bạn đánh dấu vào những thư viện mong muốn thêm vào hồ sơ tạo bộ cài đặt.

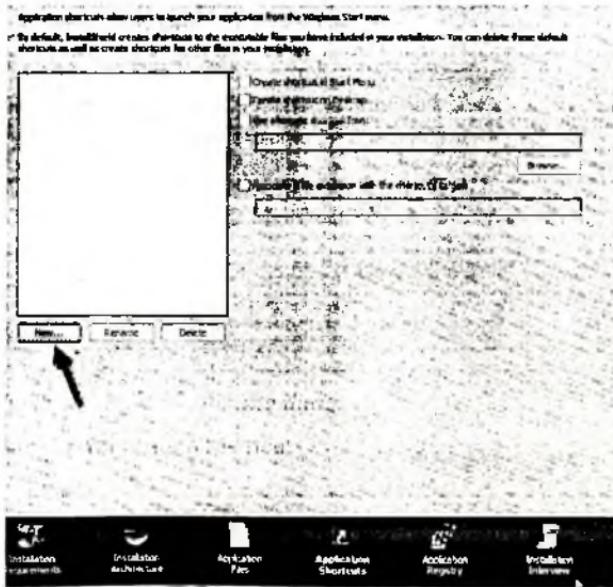


Một cửa sổ hiện ra chứa báo cáo về những tệp thư viện đã được lựa chọn.

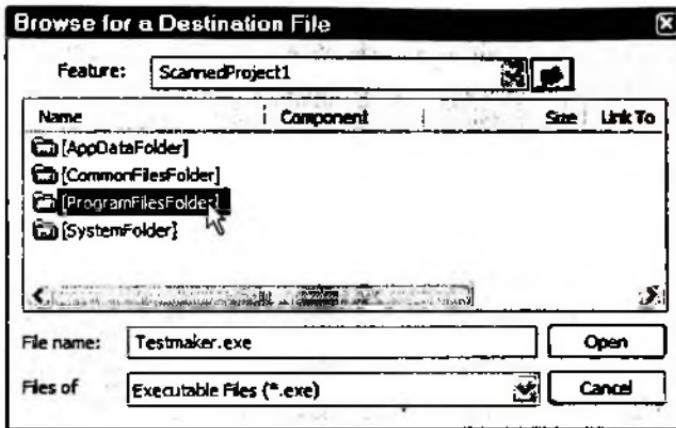


7.2.2.9. Thiết kế shortcut cho ứng dụng

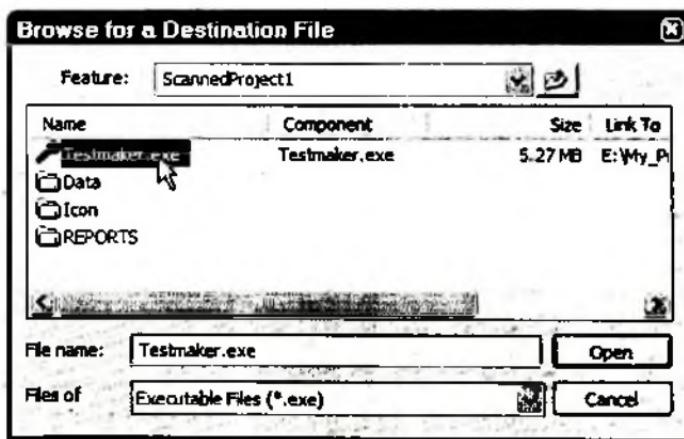
Tại mục thứ 5: Application Shortcut



Chọn nút New... để khai báo shortcut cho ứng dụng.



Cửa sổ hiện ra đề nghị chúng ta chỉ vị trí của tệp thi hành của ứng dụng. Mặc định thì bạn sẽ luôn tìm nó trong mục “ProgramFilesFolder”.



Tìm đến thư mục chứa ứng dụng đã được khai báo từ trước, chọn tệp thi hành muốn tạo shortcut. Bấm nút Open để hoàn tất việc lựa chọn.



Với tệp thi hành vừa chọn, ta có thể thiết lập các tùy chọn:

- Create shortcut in Start Menu: để tạo shortcut của ứng dụng trong StartMenu của hệ điều hành Window
 - Create shortcut on Desktop: để tạo shortcut của ứng dụng ngay trên màn hình Desktop
 - Use alternate shortcut icon: Nếu bạn muốn chỉ ra một biểu tượng (icon) mới thay thế cho biểu tượng mặc định của ứng dụng.



Lựa chọn mục Application Registry để thiết lập những đăng ký của ứng dụng trong Registry của Windows.

Với ứng dụng có phần đơn giản của chúng ta thì tốt nhất nên giữ nguyên những thiết lập mặc định.

7.2.2.10. Thiết lập những giao tiếp với người dùng cuối trong khi cài đặt ứng dụng

The Installation Interview is the collection of dialogs that your users will see at run time when they run your installation. By answering these simple questions you can configure the most common elements of the Installation Interview.

Do you want to display a Licence Agreement Dialog?

Yes No

Specify the path to your End User Licence Agreement file:

`<SPproductFolder>\Readme\DEMO\Eula.rtf`

[Browse...](#)

Do you want to prompt users to enter their Company Name and User Name?

Yes No

Do you want your users to be able to modify the installation location of your application?

Yes No

Do you want users to be able to selectively install only certain parts of your application?

Yes No

Do you want to give users the option to launch your application when the installation completes?

Yes No

Browse to the main executable to launch:

[Browse...](#)

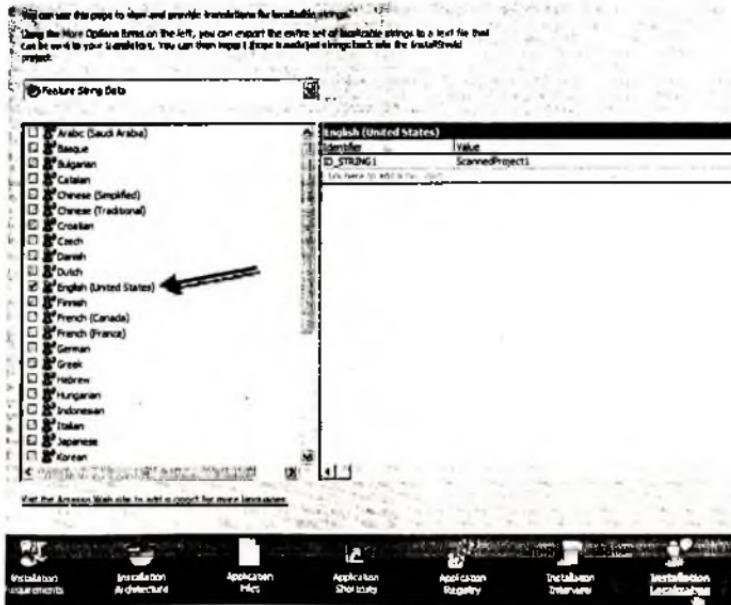


Tại mục Installation Interview ta có những tùy chọn sau:

- Do you want to display Licence Agreement file: Nếu bạn muốn hiển thị Thỏa thuận về bản quyền khi cài đặt ứng dụng thì chọn nút "Browse" và chỉ đường dẫn đến tệp rtf (*.rtf) chứa thỏa thuận bản quyền của riêng bạn hay bạn có thể sửa chữa ngay nội dung của tệp Eula.rtf mặc định của hệ thống.

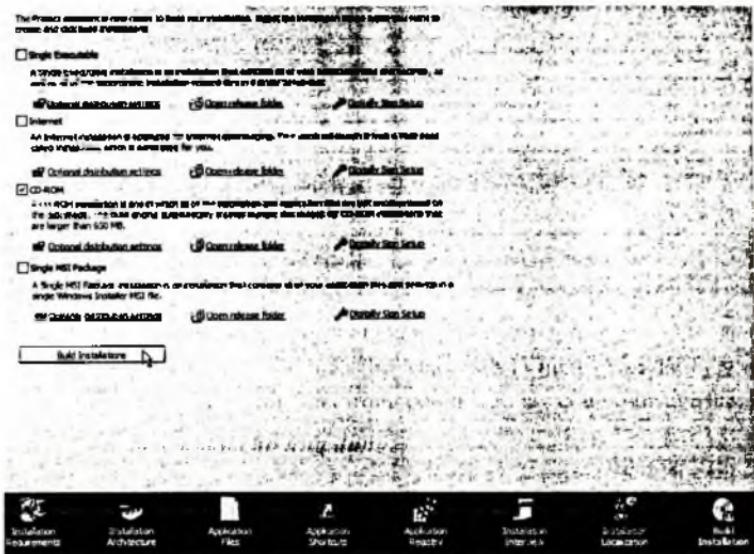
- Do you want prompt...: chọn YES nếu bạn muốn hiện cửa sổ để hỏi người dùng nhập tên cơ quan và tên người sử dụng.
- Do you want your user to be able...: chọn YES nếu bạn muốn người dùng cuối có thể thay đổi vị trí của thư mục chứa ứng dụng không theo vị trí mặc định.
- Do you want user to be able...: Chọn YES nếu bạn muốn người dùng được quyền lựa chọn cài đặt một phần hay là tất cả các thành phần của ứng dụng (tùy chọn này chỉ có ý nghĩa nếu ứng dụng được cài đặt có nhiều tệp thi hành cho nhiều thành phần khác nhau, giống như khi cài đặt bộ offices nếu muốn ta có thể lựa chọn cài đặt một hay một số phần mềm ứng dụng của bộ offices)
- Do you want to give user...: Chọn YES nếu bạn muốn cho phép người dùng có thể khởi chạy ứng dụng ngay sau khi bấm nút Finish kết thúc quá trình cài đặt ứng dụng. Khi đó bạn sẽ còn phải chỉ đường dẫn tới tệp thi hành của ứng dụng sẽ được khởi chạy.

7.2.2.11. Lựa chọn ngôn ngữ hỗ trợ



Tại mục Installation Localization ta lựa chọn những ngôn ngữ mà bộ cài đặt ứng dụng sẽ hỗ trợ trong việc hiển thị. Mặc định là tiếng Anh đã được lựa chọn, rất tiếc là chưa có tiếng Việt trong phiên bản này của bộ installshield.

7.2.2.12. Tạo bộ cài đặt cho ứng dụng



Vậy là đã đến mục cuối cùng: Build Installation. Chúng ta thiết lập theo các tùy chọn:

- Single Executable: để tạo ra bộ cài đặt với chỉ một tệp duy nhất.
- Internet: để tạo ra bộ cài đặt được tối ưu cho việc download và cài đặt trực tiếp từ trên mạng
- CD Rom: để tạo ra bộ cài đặt chứa trên một hay nhiều đĩa CD (nếu có kích thước lớn hơn 650Mb)
- Single MSI package: để tạo ra một tệp window Installer Msi file duy nhất.

Chọn nút “Build Installations” để bắt đầu quá trình tạo bộ cài đặt.

```
B) AdvUISequence table successfully built  
InstallExecuteSequence table successfully built  
InstallUISequence table successfully built  
Directory table successfully built  
Feature table successfully built  
FeatureComponents table successfully built  
Component table successfully built  
Loading File table  
Building File table  
Merging modules...  
Merging ATL Module for Windows: C:\Program Files\InstallShield\2009\Modules\386\ATL.MSM
```

Build ▶ Validate ▶ Results ▶ Compile ▶ Tasks /

Merging ATL Module for Windows: C:\Program Files\InstallShield\2009\Modules\386\ATL.MSM

Bạn có thể theo dõi quá trình tạo bộ cài đặt ở góc dưới bên trái màn hình.

Kết thúc quá trình tạo bộ cài đặt ta thấy báo cáo hiện ra

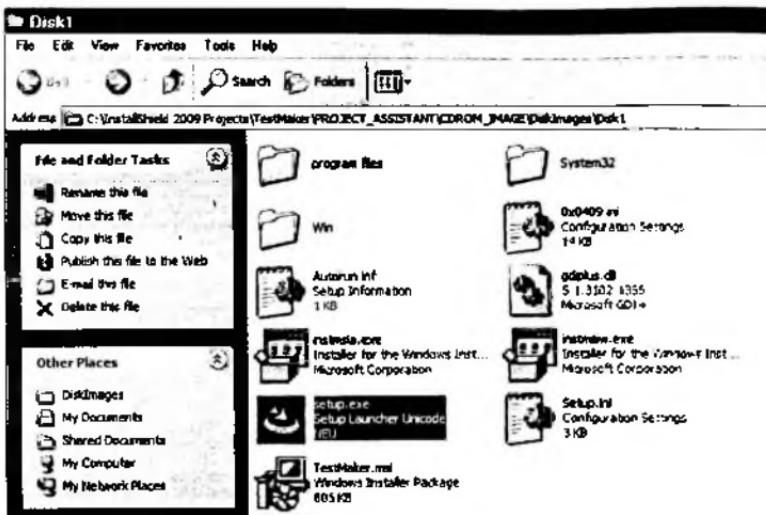
```
Building Standard setup for language English (United States) (English)  
Building strategy:  
 1. Runupgrade (English United States) built  
 2. Patch built  
 3. Setup: warning - Only if you intend to distribute this upgrade as a patch, you should author the "Patch Configuration" setting on the Advanced Settings panel of the Release Wizard.  
 4. Feature file built  
 5. Feature file created  
 6. Feature file updated (no patches detected)  
 7. 000001_ATLModule10_100001_0.msi (1. 1 warnings)  
 8. The last item was created: C:\InstallShield\2009\Modules\386\ATL.MSI  
Build ▶ Validate ▶ Results ▶ Compile ▶ Tasks /
```

Như hình trên thì có nghĩa là Bộ cài đặt của ứng dụng đã được tạo ra thành công tuy nhiên có 8 lỗi và một cảnh báo đối với công việc này.

Description	Error Code
File not found. An error occurred merging Module Crystal Reports - User Function Libraries.672F6A12_FD29_42D4_A... -4075	
File not found. An error occurred merging Module CrystalReports85_AutomationServer.041700E4_BA00_4964_B147... -4075	
File not found. An error occurred merging Module CrystalReports85_CrystalReport10_1A30C3F4_3B4F_480C_AC43_Da86E... -4075	
File not found. An error occurred merging Module CrystalReports85_CrystalReport10_1A30C3F4_3B4F_480C_AC43_Da86E... -4075	
File not found. An error occurred merging Module CrystalReports85_CrystalReport10_1A30C3F4_3B4F_480C_AC43_Da86E... -4075	
File not found. An error occurred merging Module CrystalReports85_DrctAcc_ActiveData.3F486363_B976_4E7A_B21... -4075	
File not found. An error occurred merging Module CrystalReports85_Importer.FF9C50F5_F1D0_4E99_9E46_0F4363... -4075	
File not found. An error occurred merging Module DAO150_68800B00_1C62_4580_B4D8_3E5927F50571 for Feature... -4075	

Trong thẻ Task ta có được mô tả về những lỗi xảy ra khi tạo bộ cài đặt.

Dựa vào thông báo nhận được, ta sửa lỗi và sau đó thực hiện lại việc tạo bộ cài đặt.



Bộ cài đặt đã được tạo ra chứa trong thư mục tại đường dẫn:

C:\InstallShield 2009 Projects\TestMaker\PROJECT_ASSISTANT\CDROM_IMAGE\DiskImages\Disk1

7.2.2.13. Nhận xét

Tạo bộ cài đặt là công đoạn cuối cùng trước khi chuyển giao phần mềm ứng dụng đến người dùng cuối. Có rất nhiều công cụ cho phép chúng ta tạo ra bộ cài đặt cho phần mềm ứng dụng. Mỗi công cụ có một hình thức trợ giúp, hỗ trợ khác nhau trong việc khai báo và thiết lập hồ sơ tạo bộ cài đặt. Tuy nhiên thực tế cho thấy là rất khó để tạo được bộ cài đặt hoàn chỉnh chỉ thông qua một lần thực hiện.

Để có một bộ cài đặt hoàn chỉnh cho phần mềm ứng dụng người ta thường thực hiện theo các bước lặp đi lặp lại Làm- Thử - Sai sửa cho đến khi nào có được bộ cài đặt hoàn thiện thì thôi.

PHỤ LỤC 1

Đề bài tập lớn thực hành phát triển ứng dụng trong quản lý

Đề tài 1:

Phần mềm Tạo và quản lý ngân hàng đề thi

Mô tả: Trước mỗi đợt thi học kỳ, thầy trưởng hoặc phó bộ môn sẽ biên soạn 6 đề thi cho ca thi. Đề thi thường được biên soạn theo những cách sau:

- Lựa chọn trong một ngân hàng đề thi đã được chuẩn hóa từ trước
- Tạo mới đề thi từ các câu hỏi được lựa chọn từ các đề thi trước đó hoặc trong một ngân hàng các câu hỏi có sẵn
- Tạo mới hoàn toàn hoặc là thực hiện sửa chữa từ các đề thi hay câu hỏi đã được lưu trữ từ trước.
- Cơ cấu một đề thi thường được chia làm 2 phần: Phần lý thuyết thường chiếm từ 4-6 điểm (tùy từng môn học) và còn lại là phần bài tập.
- Số lượng các câu hỏi mỗi phần thường là 2-3 câu (tùy từng môn)
- Một đề thi được trình bày trong 1 trang giấy A4

Đề tài 2:

Phần mềm Quản lý công việc của giảng viên

Mô tả: Vào đầu năm học thầy trưởng bộ môn sẽ nhận được tài liệu về kế hoạch học tập của các lớp, các khóa, các hệ đào tạo (Chính quy, Tại chức, VB2) trong năm học trước mắt.

Dựa vào kế hoạch học tập nhận được thầy trưởng bộ môn sẽ tiến hành phân công lịch giảng cho từng giảng viên dựa theo những tiêu thức sau:

- Mỗi giảng viên sẽ có một khối lượng giờ giảng tương đương nhau

- Các lớp giảng của một giảng viên có thời khóa biểu không trùng nhau
- Những yêu cầu đặc biệt từ giảng viên sẽ được xem xét để đáp ứng được nhiều nhất
-=> Nói chung là việc phân công này được thực hiện một cách chủ quan (không thể làm tự động được) dưới sự trợ giúp của máy tính (trước đây là Excel).

Lịch phân công được hoàn thành sau đó sẽ được in ra cho mỗi giảng viên một bộ giống nhau trong đó có lịch của tất cả mọi người, được nhóm theo hình thức đào tạo (Chinh quy, TC, B2), môn học và thời gian (sẽ có mẫu cụ thể để bạn xem).

Vào cuối năm học, mỗi giáo viên sẽ được in một bảng tổng hợp tình hình giảng dạy trong năm.

Trong năm thỉnh thoảng lại có sự kiện một giáo viên vì lý do nào đó đã đổi lớp cho giáo viên khác=> báo cáo để trường bộ môn điều chỉnh trong lịch phân công.

Công việc của giáo viên còn bao gồm việc chấm thi.

- Trường bộ môn sẽ lưu lại những thông tin về việc phân công chấm thi cho từng giáo viên: tên giáo viên, mã số túi bài thi, ngày phân công, hạn hoàn thành cuối cùng.
- Bài thường hoặc theo định kỳ trường bộ môn có thể in bảng liệt kê công việc chấm bài của từng giáo viên trong một khoảng thời gian bất kỳ.

Bề tài 3:

Phần mềm Quản lý tuyển sinh đại học tại chức

Hàng năm Khoa Tại chức ĐH KTQD thường tiến hành nhiều đợt tuyển sinh đại học tại chức học tại trường kinh tế và tại các địa phương liên kết đào tạo.

- Khi bán hồ sơ, người ta đã lưu thông tin về họ tên, ngày sinh, nơi sinh của thí sinh.
- Khi thu hồ sơ đăng ký thi, người ta lưu nốt các thông tin còn lại (sẽ có bộ hồ sơ tuyển sinh mẫu cho bạn).

- Ngoài ra người ta sẽ cập nhật các bảng danh mục sau:
 - o Danh mục cơ sở liên kết đào tạo
 - o Danh mục chuyên ngành đào tạo
 - o Danh mục địa điểm thi (đối với hội đồng tuyển sinh tại trường ĐH KTQD)
 - o Danh mục chỉ tiêu đào tạo cho mỗi chuyên ngành (tương ứng với mỗi đợt thi tuyển sinh)
- Sau khi hoàn tất việc cập nhật hồ sơ người ta sẽ:
 - o Đánh số báo danh tự động, phân phòng thi tương ứng với mỗi địa điểm thi, in danh sách thí sinh theo phòng thi, in danh sách ảnh, in giấy báo dự thi....
 - o Điều chỉnh lại hồ sơ dự tuyển theo định chính sai sót.
- Sau khi có kết quả chấm điểm thi 3 môn người ta sẽ:
 - o Cập nhật điểm cho thí sinh theo phòng thi
 - o Tính điểm chuẩn dựa theo tổng số chỉ tiêu tuyển sinh
 - o Tính điểm chuẩn cho từng chuyên ngành dựa vào nguyện vọng đăng ký và số chỉ tiêu được phân bổ
 - o In giấy báo điểm thi
 - o Phân lớp thí sinh trúng tuyển

Đề tài 4:

Phần mềm Quản lý bán hàng

Một trung tâm thương mại để thực hiện việc quản lý của mình có theo dõi các bảng danh mục sau:

- Danh mục hàng hóa, lượng tồn kho hiện thời
- Danh mục kho hàng
- Danh mục khách hàng, nhà cung cấp

Việc mua bán tại cửa hàng luôn được thanh toán ngay (nên không có công nợ!), trên một hóa đơn có thể bán nhiều mặt hàng, tỷ lệ thuế cho hàng bán ra là 10%. Khách hàng được hưởng chiết khấu trực tiếp trên hóa đơn bán hàng

theo quy định sau:

- Nếu tổng giá trị hóa đơn từ 1 triệu đến 3 triệu thì chiết khấu 5%
- Nếu trên 3 triệu thì chiết khấu 8%

Mỗi mặt hàng sẽ được bán nếu số lượng tồn hiện tại trong kho lớn hơn yêu cầu mua.

Để có hàng bán cửa hàng phải nhập thêm hàng, thể hiện ở các hóa đơn nhập hàng được cập nhật.

Định kỳ hay bất thường, cửa hàng có thể in các báo cáo sau:

- Báo cáo tổng hợp hàng nhập trong một khoảng thời gian xác định
- Báo cáo tổng hợp hàng bán trong một khoảng thời gian xác định
- Báo cáo tổng hợp nhập, xuất, tồn kho
- Báo cáo doanh thu theo từng khách hàng và theo từng mặt hàng
- Báo cáo chi phí theo từng nhà cung cấp và theo từng mặt hàng
- Báo cáo kết quả kinh doanh của cửa hàng.

Đề tài 5:

Phần mềm Quản lý thư viện

Thư viện trường kinh tế có quản lý các bảng danh mục sau:

- Danh mục nhà xuất bản
- Danh mục chủ đề
- Danh mục sách (NXB, chủ đề, số lượng...)
- Danh mục bạn đọc

Mỗi bạn đọc sẽ được mượn 2 cuốn sách trên 1 phiếu mượn, và phải trả sách tối đa là 3 ngày sau. Yêu cầu sẽ được đáp ứng nếu:

- Bạn đọc đã trả những cuốn sách mượn lần trước
- Số lượng cuốn sách tương ứng vẫn còn trong kho
- Bạn đọc không thuộc danh sách những bạn đọc bị phạt vì trả sách quá thời hạn (phạt 1 tuần không được mượn sách)

Định kỳ hay bất thường thư viện có thể in các báo cáo sau:

- Báo cáo kiểm kê sách trong kho
- Báo cáo tên 10 đầu sách được mượn nhiều nhất
- Báo cáo tên các đầu sách không được mượn lần nào
- Danh mục những cuốn sách đang được mượn
- Danh mục những cuốn sách đã quá hạn trả (mượn quá 3 ngày)
- Danh sách bạn đọc bị phạt vì trả quá hạn
- Bảng thống kê số lần mượn sách theo tên sách và theo tên bạn đọc

Đề tài 6:

Phần mềm quản lý điểm thi của sinh viên

Mỗi sinh viên có mã sinh viên, được quản lý theo lớp với các thông tin (họ đệm, tên, ngày sinh, giới tính, lớp, chuyên ngành). Với mỗi môn học trong suốt khóa học, cán bộ quản lý sẽ cập nhật điểm chuyên cần, điểm kiểm tra và điểm thi. Điểm trung bình môn học được tính theo quy định chung. Sinh viên thi trượt sẽ phải thi lại 1 lần, không có học lại.

Đầu vào của phần mềm quản lý điểm sinh viên:

- **Danh mục:**
 - Chuyên ngành
 - Lớp
 - Sinh viên
 - Môn học
- **Điểm phát sinh**
 - Điểm lần đầu (chuyên cần, kiểm tra, thi)
 - Điểm thi lại

Đầu ra:

- Bảng danh sách sinh viên theo lớp (danh sách điểm danh, danh sách chấm điểm kiểm tra, danh sách chấm thi, danh sách sinh viên thi lại theo từng môn học)
- Bảng điểm tổng hợp của sinh viên theo từng lớp, từng môn học

Đề tài 7:

Phần mềm quản lý điểm học sinh

Mỗi học sinh có mã học sinh, được quản lý theo lớp với các thông tin (họ đệm, tên, ngày sinh, giới tính, lớp). Với mỗi môn học trong suốt khóa học, cán bộ quản lý sẽ cập nhật điểm miệng, điểm kiểm tra và điểm thi. Điểm trung bình môn học được tính theo quy định chung (điểm thi tính hệ số 2).

Đầu vào của phần mềm quản lý điểm học sinh

- Danh mục:
 - o Lớp
 - o Học sinh
 - o Môn học
- Điểm phát sinh
 - o Điểm kiểm tra miệng (1 điểm), điểm kiểm tra viết (2 điểm) và điểm thi

Đầu ra:

- Bảng danh sách học sinh theo lớp (danh sách điểm danh, danh sách chấm điểm kiểm tra, danh sách chấm thi)
- Bảng điểm tổng hợp của học sinh theo từng lớp, từng môn học
- Bảng điểm của từng học sinh với tất cả các môn học trong toàn khóa học

Minh họa các bước phân tích thiết kế và xây dựng một phần mềm ứng dụng trong quản lý

Đề bài:

PHÂN TÍCH, THIẾT KẾ, XÂY DỰNG PHẦN MỀM QUẢN LÝ NHÂN SỰ VÀ TÍNH LƯƠNG CHO CÔNG TY TRÁCH NHIỆM HỮU HẠN KONA

NỘI DUNG CƠ BẢN

1. KHẢO SÁT HIỆN TRẠNG VÀ LẬP MÔ HÌNH HOẠT ĐỘNG HỆ THỐNG.....	279
1.1 Một số thông tin chung về công ty TNHH KONA	279
1.2 Phân tích nghiệp vụ quản lý nhân sự và tính lương tại công ty TNHH KONA	279
1.2.1 Quản lý hồ sơ nhân viên.....	279
1.2.2 Quản lý khen thưởng kỷ luật.....	280
1.2.3 Công tác chấm công.....	280
1.2.4 Công tác tính lương.....	280
1.3 Lập mô hình hoạt động hệ thống.....	283
1.3.1 Sơ đồ chức năng BFD	283
1.3.2 Sơ đồ luồng dữ liệu DFD	284
1.3.3 Sơ đồ luồng thông tin IFD.....	290
2. PHÂN TÍCH THIẾT KẾ	293
2.1 Thiết kế kiến trúc hệ thống.....	293
2.2 Thiết kế cơ sở dữ liệu.....	294
2.2.1 Sơ đồ quan hệ thực thể ERD	294
2.2.2 Thiết kế chi tiết các tệp cơ sở dữ liệu.....	295
3. LẬP TRÌNH.....	300

3.1 Cấu trúc một dự án	301
3.2 Giới thiệu một số công cụ hỗ trợ lập trình	302
3.2.1 Công cụ hỗ trợ hiển thị dữ liệu.....	302
3.2.2 Công cụ hỗ trợ giao diện	303
3.3 Lập trình giao diện chính	303
3.3.1 Lập trình thanh thực đơn Menu.....	304
3.3.2 Lập trình thanh công cụ Toolbar	304
3.3.3 Lập trình Menu chuột phải	308
3.4 Lập trình Sub main()	308
3.5 Lập trình Form Đăng nhập hệ thống	310
3.6 Lập trình Form Danh mục phòng ban	312
3.7 Lập trình Form Danh mục nhân viên	321
3.8 Lập trình Form Tìm kiếm nhân viên	335
3.9 Lập trình Form Quản lý chứng từ khen thưởng.....	344
3.10 Lập trình Form Cập nhật danh sách tính lương.....	350
3.11 Lập trình Form Tổng hợp vật tư nhập kho	353
3.12 Lập trình Form Cập nhật ngày công và tính lương	363
3.13 Lập trình From Báo cáo thâm niên công tác và sinh nhật.....	363
3.13.1 Xây dựng bản mẫu báo cáo	364
3.13.2 Thiết kế bản mẫu báo cáo.....	364
3.13.3 Lập trình cho Form Báo cáo thâm niên công tác và sinh nhật..	365

1. KHẢO SÁT HIỆN TRẠNG VÀ LẬP MÔ HÌNH HOẠT ĐỘNG HỆ THỐNG

1.1. Một số thông tin chung về công ty TNHH KONA

Tên tiếng Việt: CÔNG TY TRÁCH NHIỆM HỮU HẠN KONA

Tên giao dịch quốc tế: KONA COMPANY LIMITED

Địa chỉ: Bình Đà - Bình Minh - Thanh Oai - Hà Nội

Điện thoại: 0343 876 940 - 0343 876 937 Fax: 0343 876 947

Website: <http://www.konaco.com.vn> Email: kona@hn.vnn.vn

Công ty TNHH KONA là công ty sản xuất và kinh doanh sản phẩm chăn, ga, gối, đệm mang thương hiệu KORE. Sản phẩm mang thương hiệu KORE do công ty KONA sản xuất đã khẳng định được uy tín và chất lượng sản phẩm vượt trội, vẫn là chất liệu E, Cotton, Solid truyền thống được sản xuất trên dây chuyền công nghệ hiện đại, nguyên liệu 100% nhập từ Hàn Quốc luôn mang đến cho người sử dụng cảm giác thoải mái và nhẹ nhàng.

1.2. Phân tích nghiệp vụ quản lý nhân sự và tính lương tại công ty TNHH KONA

1.2.1. Quản lý hồ sơ nhân viên

Sau mỗi đợt tuyển dụng, ứng viên trúng tuyển sẽ nộp tiền đặt cọc, trải qua thời gian thử việc, nếu đạt yêu cầu sẽ được ký hợp đồng lao động và trở thành nhân viên chính thức của công ty.

Do công ty chưa có phần mềm quản lý nhân sự nên cán bộ kế toán tiền lương kiêm luôn nhiệm vụ xây dựng và bảo trì cơ sở dữ liệu về nhân sự trên Excel.

Khi có nhân viên mới vào công ty, thì hồ sơ của nhân viên sẽ được nhân viên của phòng hành chính chuyển cho cán bộ kế toán tiền lương để cập nhật vào máy tính.

Khi có yêu cầu sửa đổi thông tin từ các phòng ban khác hay giám đốc, cán bộ kế toán tiền lương sẽ tiến hành sửa đổi cho phù hợp.

Khi có yêu cầu tra cứu, lọc dữ liệu để đưa ra các thông tin quan trọng, cán bộ kế toán tiền lương phải sử dụng khá nhiều hàm Excel phức tạp để thực hiện. Với những yêu cầu phức tạp quá thì việc sử dụng Excel không đáp ứng được.

Khi có nhân viên bị sa thải, giám đốc sẽ ký quyết định thôi việc. Sau đó,

cán bộ tiền lương tiến hành xoá hồ sơ nhân viên khỏi cơ sở dữ liệu.

Cuối tháng, trước khi tính lương, cán bộ tiền lương tiến hành kiểm tra danh sách nhân viên, cập nhật cơ sở dữ liệu, sau đó in danh sách nhân viên và những thông tin cần thiết khác theo yêu cầu của giám đốc và các phòng ban khác.

1.2.2. Quản lý khen thưởng kỷ luật

Trưởng các phòng ban, bộ phận trong công ty có nhiệm vụ theo dõi khen thưởng, kỷ luật của nhân viên.

Khi có danh sách đề nghị khen thưởng kỷ luật được gửi lên từ các phòng ban bộ phận khác trong công ty, phòng hành chính tiến hành xét duyệt. Sau đó, tùy theo từng vụ việc cụ thể, chứng từ khen thưởng hay kỷ luật sẽ được lập áp dụng cho từng nhân viên. Cán bộ kế toán tiền lương lưu thông tin số tiền thưởng hay phạt vào Excel.

Số tiền thưởng, phạt đã được quy định sẵn trong quy chế của công ty.

Cuối tháng, những chứng từ khen thưởng hay kỷ luật sẽ được gửi lên giám đốc. Cán bộ kế toán tiền lương dựa vào chứng từ khen thưởng hay kỷ luật để làm cơ sở cộng hay trừ vào lương.

1.2.3. Công tác chấm công

Do số lượng cán bộ công nhân viên nhiều, nên quản đốc, trưởng các phòng ban sẽ chịu trách nhiệm chấm công từng phòng ban, bộ phận của mình.

Hàng ngày, sau khi chấm công, bảng chấm công và các thông tin nghỉ việc sẽ được gửi lên giám đốc để Giám đốc có các quyết định cần thiết.

Công ty thực hiện chấm công theo thời gian, chấm công theo sản phẩm. Ngoài ra còn có bảng tính hệ số phân bổ sản phẩm theo nhân viên, hay chính là để tính điểm sản phẩm chấm cho từng công nhân.

Cuối tháng, bảng chấm công thời gian, bảng chấm công sản phẩm cùng với bảng chấm điểm sản phẩm cả tháng sẽ được gửi sang cho cán bộ kế toán tiền lương của phòng kế toán để làm cơ sở quan trọng cho việc tính lương.

1.2.4. Công tác tính lương

1.2.4.1. Một số quy định chung

Lương cơ bản của một nhân viên được quy định là cố định, dùng để làm cơ sở tính lương. Không áp dụng hệ số lương. Việc thay đổi mức lương cơ bản do giám đốc quyết định. Ví dụ: Mức lương cơ bản là 3.000.000 đ.

Số công để tính lương hàng tháng theo quy ước là: 26.

Số công làm việc của nhân viên/tháng = 26 - số ngày nghỉ.

Số giờ làm việc trong ngày: 8 giờ. Quy định thời gian giờ hành chính là 7h30 - 16h30. Sáng từ 7h30 đến 12h, chiều từ 13h đến 16h30 phút. Ngoài thời gian trên nếu nhân viên làm việc thì tính làm thêm giờ.

Lương làm thêm giờ = Lương cơ bản / 26/8 * Số giờ làm thêm * 1.5

Lương ngày chủ nhật, lễ Tết =

Lương cơ bản / 26/8 * Số giờ làm thêm * 2.0

Phụ cấp. Mức tiền phụ cấp và việc thay đổi mức phụ cấp do Giám đốc công ty quyết định.

Ăn ca: Một bữa ăn được tính 7.000 đ

Khoản bù lương: Khi có bù lương thì sẽ được tính vào lương. (Có thể được bù do bị tinh thiếu lương, hoặc các trường hợp cụ thể khác)

Tạm ứng: KONA không chỉ trả tạm ứng mà trả lương một lần duy nhất vào đầu tháng.

Bảo hiểm xã hội: Quỹ bảo hiểm xã hội = 20% lương nhân viên. Công ty chi trả 15%, nhân viên chi trả 5%.

Bảo hiểm y tế: Quỹ bảo hiểm y tế = 3% lương nhân viên. Công ty chi trả 2%, nhân viên chi trả 1%.

Bảo hiểm thất nghiệp: Quỹ bảo hiểm thất nghiệp = 2% lương nhân viên. Nhân viên chi trả 1% lương, 1% do công ty chi trả.

Việc tính bảo hiểm đã từng được áp dụng nhưng mấy tháng gần đây công ty được hoãn và chưa tính khoản bảo hiểm thất nghiệp này vào lương.

Kinh phí công đoàn: Kinh phí công đoàn = 2% lương nhân viên. Công ty chịu toàn bộ kinh phí này.

Lương thực lĩnh = Lương cơ bản/26* số ngày làm việc + Lương làm thêm giờ + Ăn ca + Phụ cấp + Bù lương + Lương sản phẩm - Bảo hiểm xã hội - Bảo hiểm y tế + Tiền thường - Tiền phạt.

Công ty TNHH KONA thực hiện tính lương theo hai hình thức là tính lương theo sản phẩm và tính lương theo thời gian. Việc tính lương theo thời gian thực hiện với bộ phận văn phòng, kho, còn đối với bộ phận sản xuất (padding, quilting, lồng đệm, Ball, may, cát, phụ may) thì áp dụng cả hai

hình thức. Việc chấm công thời gian hay sản phẩm do quản đốc quyết định dựa trên tình hình công việc thực tế.

Vào cuối tháng, các trưởng phòng ban tiến hành nộp bảng chấm công cho phòng kế toán làm cơ sở tính lương.

1.2.4.2. Việc tính lương theo thời gian:

Cán bộ tiền lương tiến hành nhập các con số thống kê về mức lương cơ bản, số liệu ngày công thời gian, ngoài giờ thời gian, chủ nhật thời gian vào bảng tính Excel đã được lập sẵn.

Lương thời gian = Lương cơ bản/26 * số ngày làm việc + lương làm thêm thời gian.

1.2.4.3. Việc tính lương theo sản phẩm:

Việc tính lương sản phẩm được thực hiện theo 4 bước như sau:

Bước 1. Tính tổng giá trị vật tư nhập kho cho từng tổ sản xuất

Tổng giá trị vật tư nhập kho = \sum Số lượng vật tư nhập kho * Đơn giá tính lương 1 vật tư.

Trong đó:

- Số lượng vật tư nhập kho của một tổ sản xuất được lấy từ phần mềm kế toán Fast Accounting dưới dạng file Excel.

- Đơn giá tính lương đối với từng vật tư sẽ là cố định do công ty quy định, được lưu trong bảng tính Excel.

Bước 2. Tính tổng giờ lương sản phẩm của các nhân viên

Cán bộ kế toán tiền lương tiến hành cập nhật số công sản phẩm, ngoài giờ sản phẩm, chủ nhật sản phẩm, điểm cho từng nhân viên vào bảng tính Excel. Điểm của từng nhân viên do quản đốc phân xưởng quyết định. (Điểm có thể gọi là Hệ số phân bổ sản phẩm theo nhân viên)

Tổng giờ lương sản phẩm = \sum (Số công sản phẩm * 8 + Ngoài giờ sản phẩm * 1.5 + Chủ nhật sản phẩm * 1.5) * Điểm từng nhân viên.

Bước 3. Tính hệ số lương sản phẩm đơn vị

Hệ số lương sản phẩm đơn vị = Tổng giá trị vật tư nhập kho /Tổng giờ lương sản phẩm

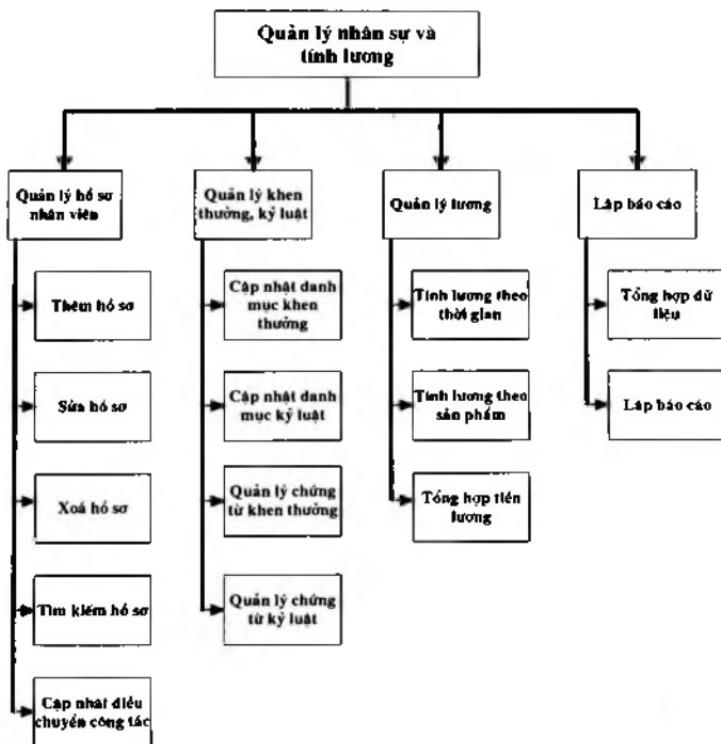
Bước 4. Tính lương sản phẩm của từng nhân viên

Lương sản phẩm của từng nhân viên = (Số công sản phẩm * 8 + Ngoài giờ sản phẩm * 1.5 + Chủ nhật sản phẩm * 1.5) * Điểm từng nhân viên * Hệ số lương sản phẩm đơn vị.

Sau khi tính được lương sản phẩm, cùng với việc tính lương thời gian sẽ tính được tổng số lương từng nhân viên. Tổng số lương này khấu trừ bảo hiểm, công tiền thường và trừ tiền phạt sẽ tính được lương thực lĩnh của từng nhân viên.

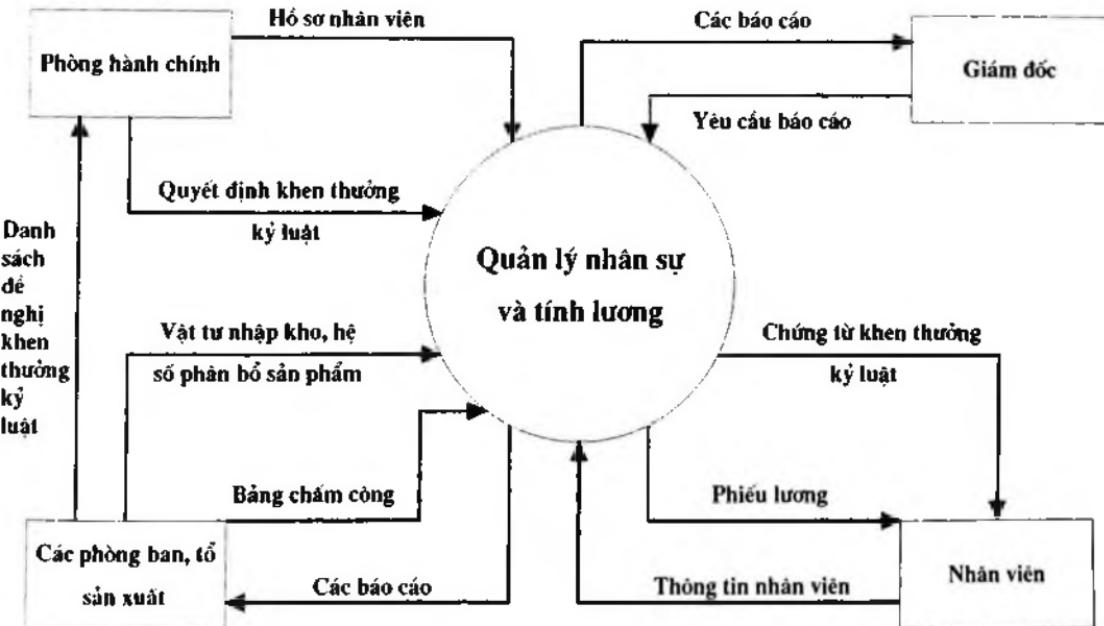
1.3. Lập mô hình hoạt động hệ thống

1.3.1. Sơ đồ chức năng BFD

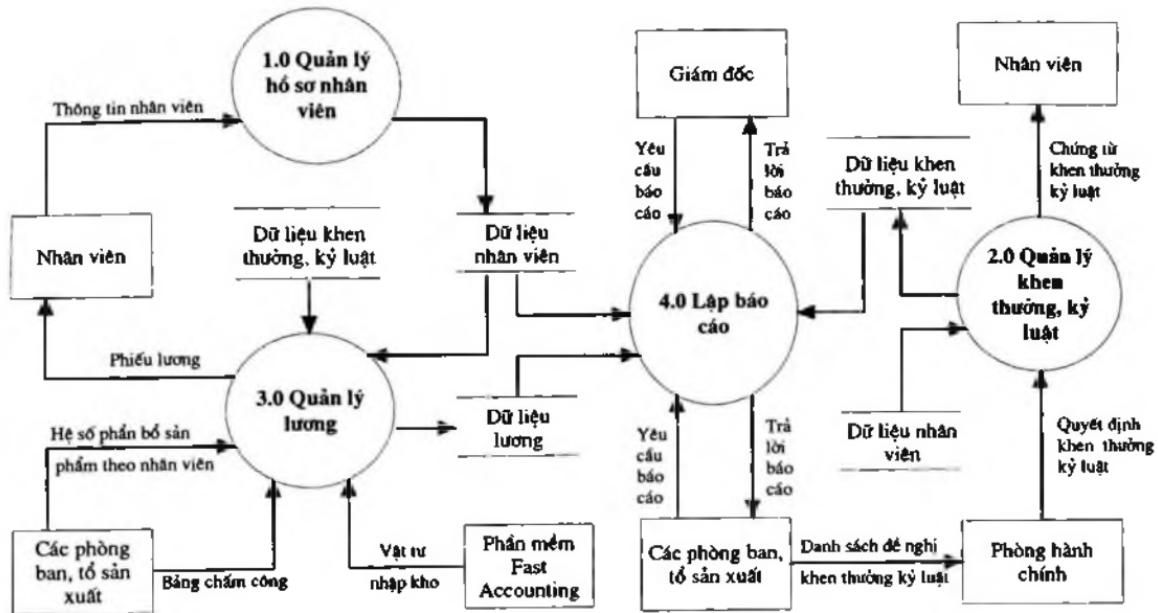


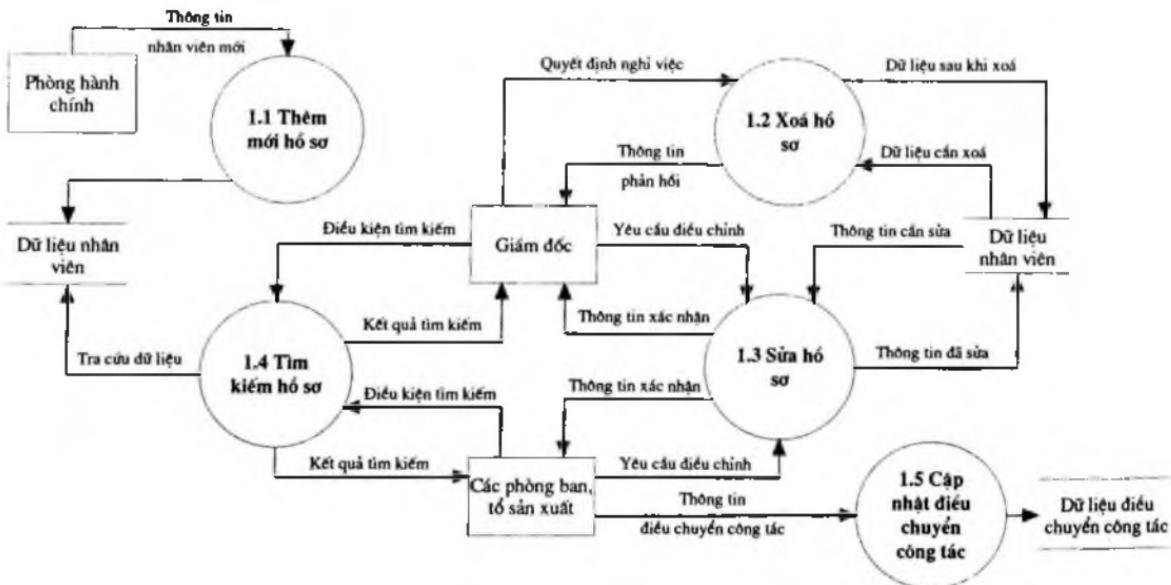
Hình 1. Sơ đồ chức năng BFD

1.3.2. Sơ đồ luồng dữ liệu DFD

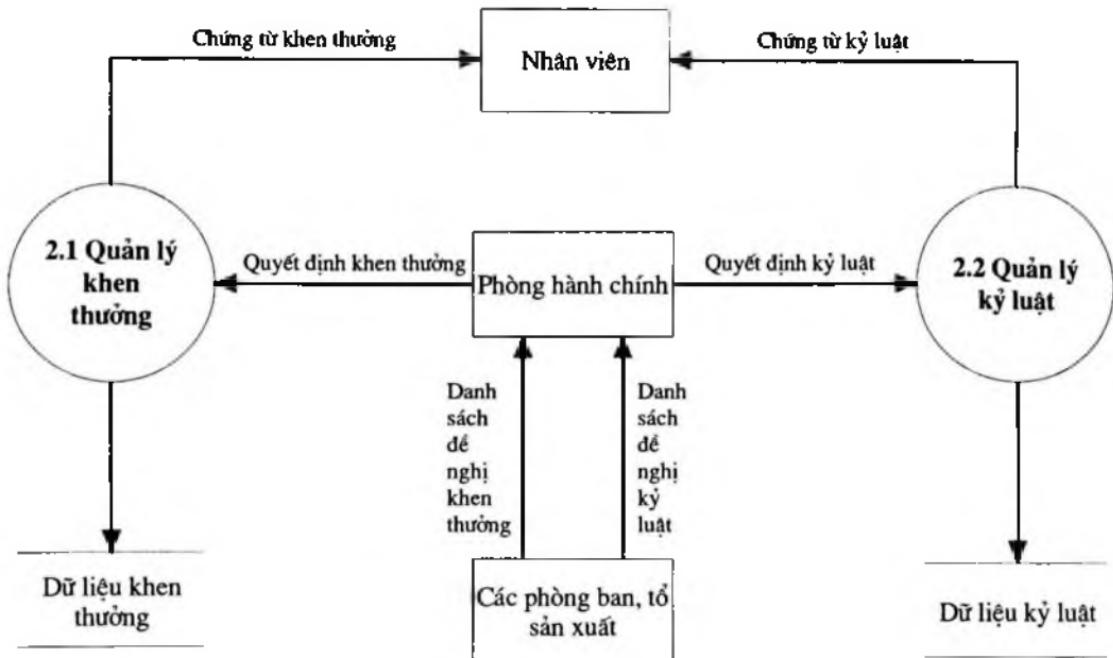


Hình 2. Sơ đồ DFD mức ngữ cảnh của hệ thống

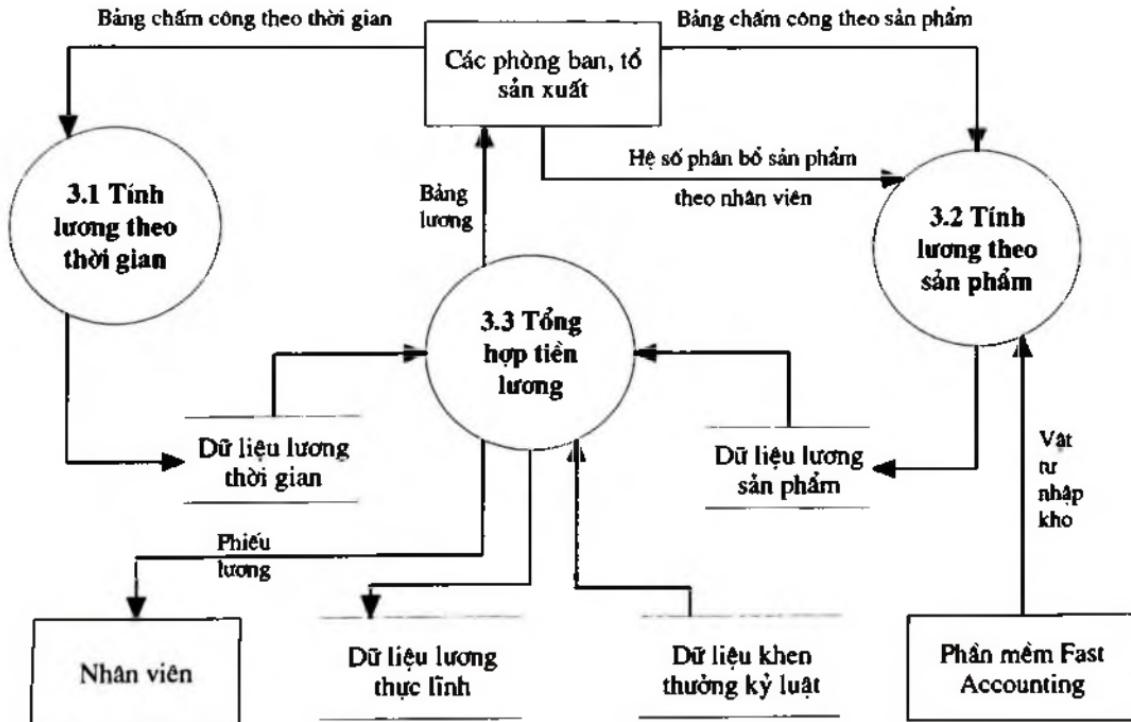




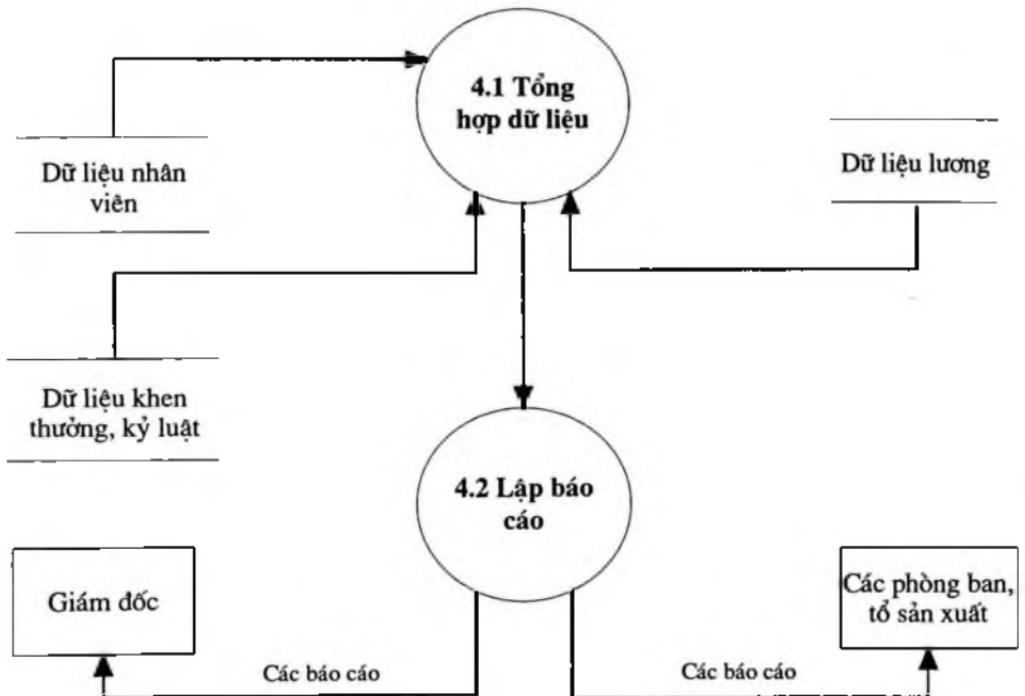
Hình 4. Sơ đồ DFD mức I của chức năng quản lý hồ sơ nhân viên



Hình 5. Sơ đồ DFD mức 1 của chức năng quản lý khen thưởng kỷ luật



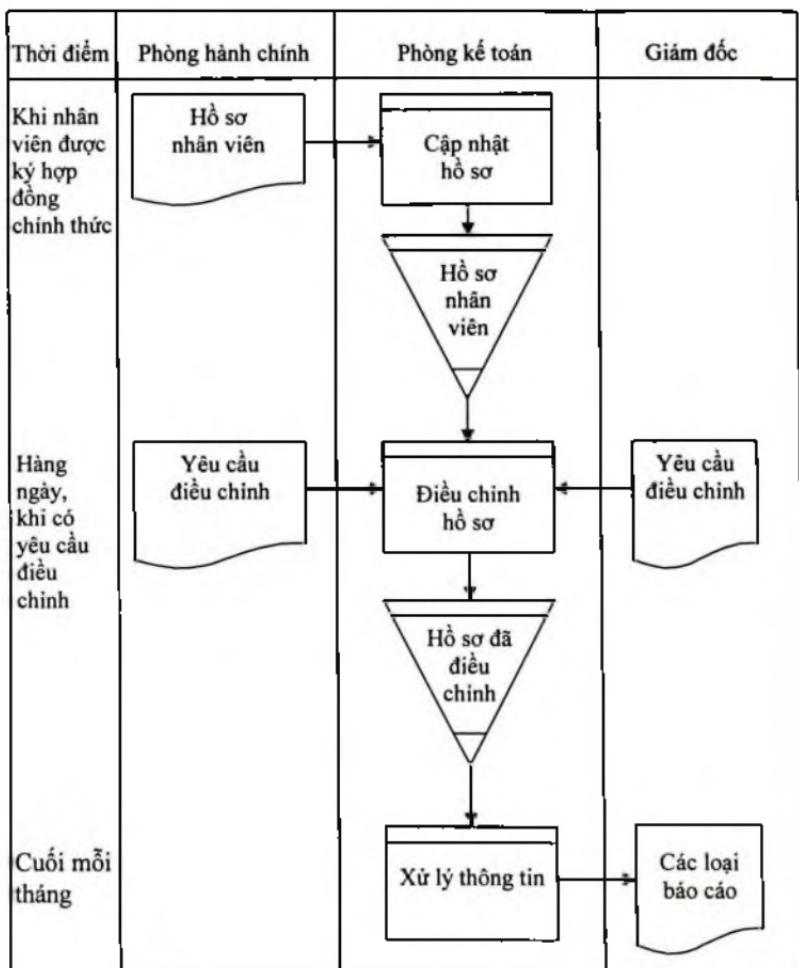
Hình 6. Sơ đồ DFD mức 1 của chức năng quản lý lương



Hình 7. Sơ đồ DFD mức I của chức năng Lập báo cáo

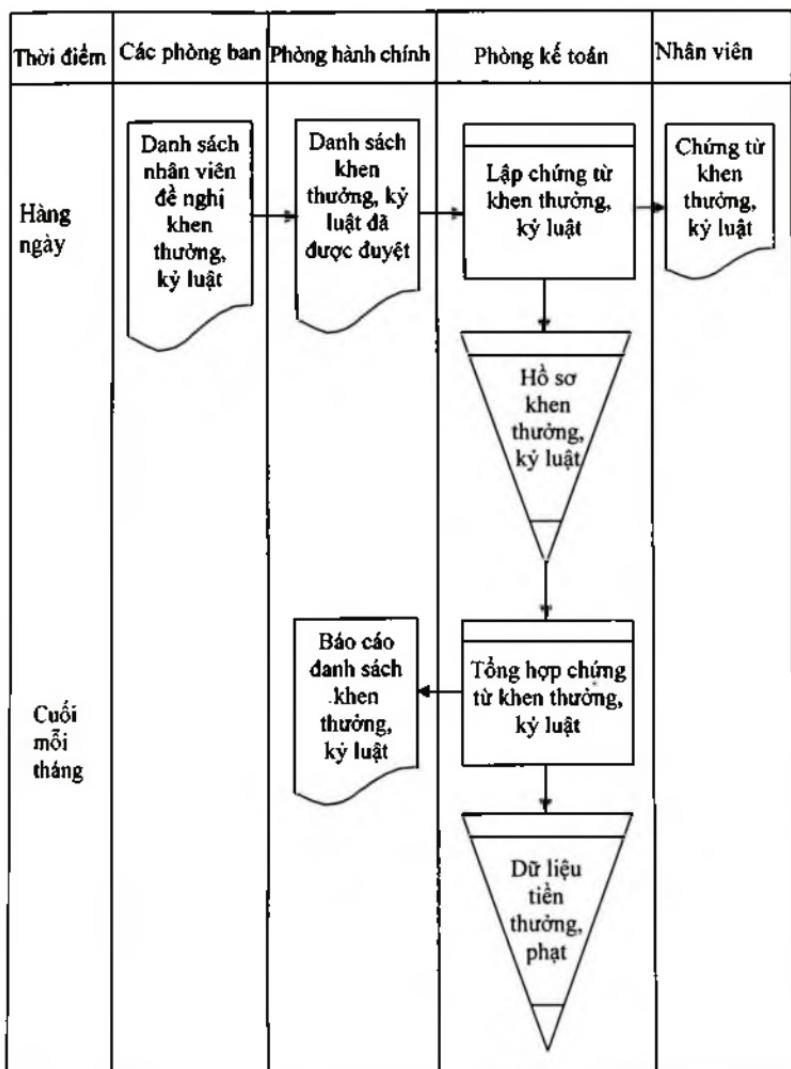
1.3.3. Sơ đồ luồng thông tin IFD

1.3.3.1. Sơ đồ IFD Quản lý hồ sơ nhân viên



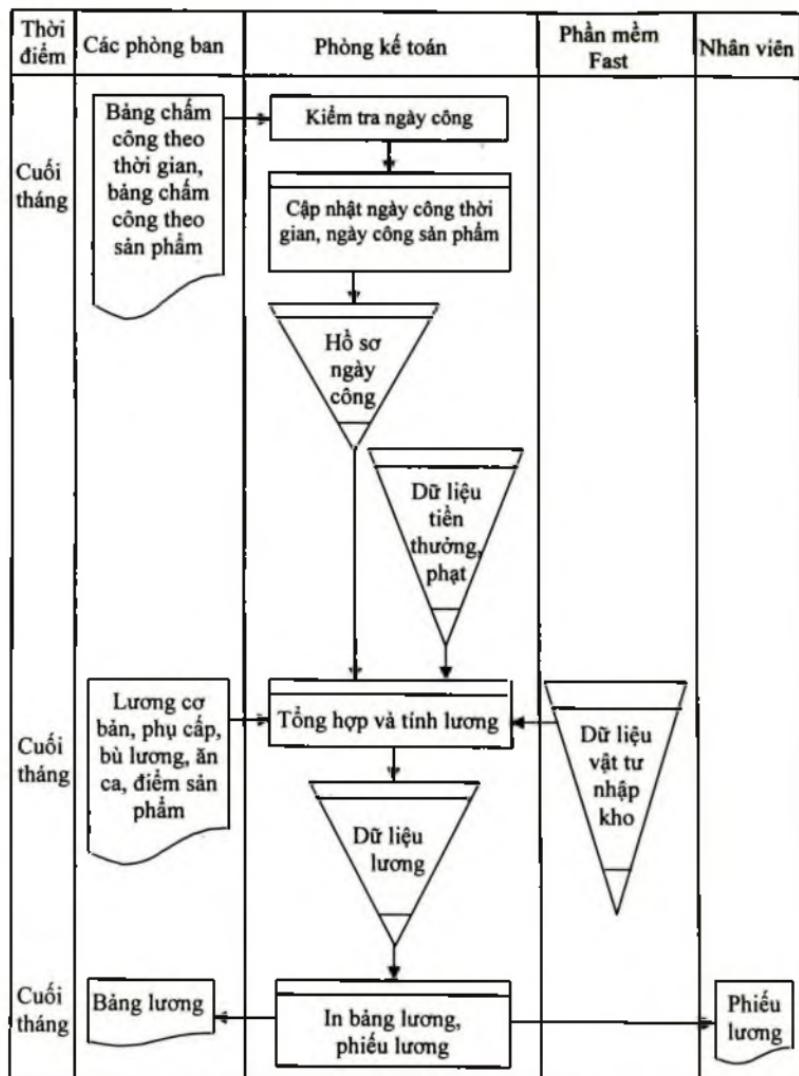
Hình 8. Sơ đồ IFD Quản lý hồ sơ nhân viên

1.3.3.2. Sơ đồ IFD Quản lý khen thưởng, kỷ luật



Hình 9. Sơ đồ IFD Quản lý khen thưởng, kỷ luật

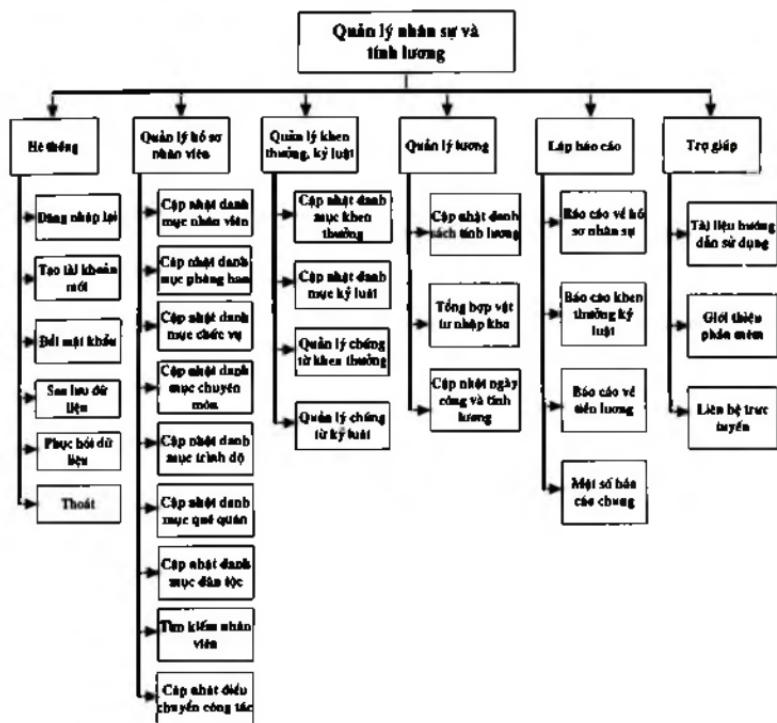
1.3.3.3. Sơ đồ IFD Quản lý lương



Hình 10. Sơ đồ IFD Quản lý lương

2. Phân tích thiết kế

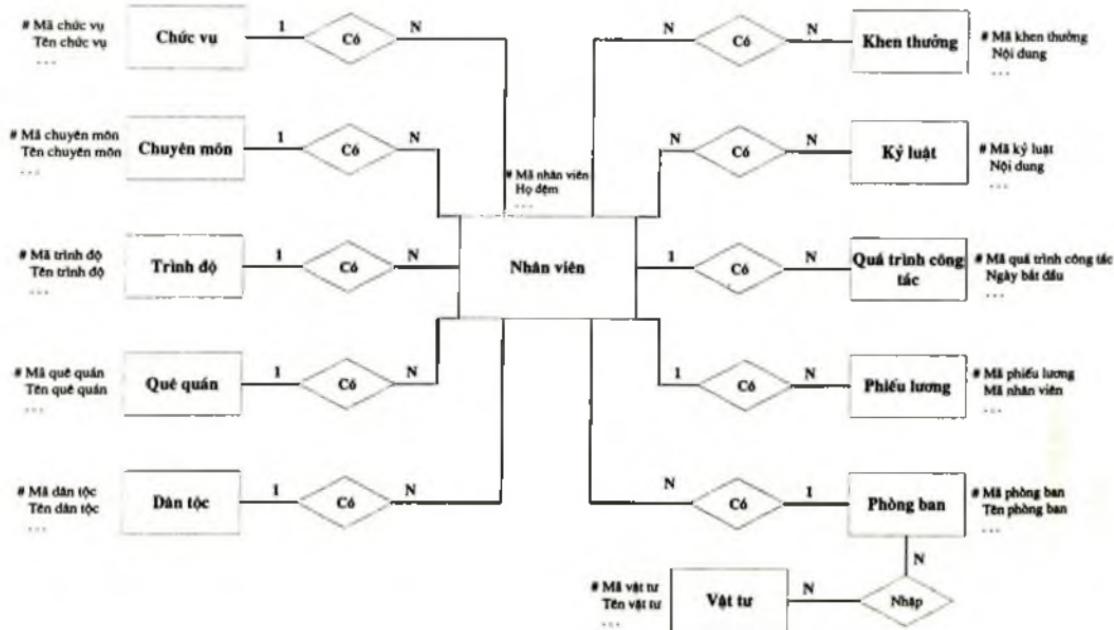
2.1. Thiết kế kiến trúc hệ thống



Hình 11. Thiết kế kiến trúc hệ thống

2.2. Thiết kế cơ sở dữ liệu

2.2.1. Sơ đồ quan hệ thực thể ERD



Hình 12. Sơ đồ Quan hệ thực thể ERD

2.2.2. Thiết kế chi tiết các tệp cơ sở dữ liệu

1. Tệp Phòng ban - PhongBan

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Điển giải
1	#MaPhongBan	Text	15	Mã phòng ban
2	TenPhongBan	Text	30	Tên phòng ban
3	DiaDiem	Text	25	Địa điểm
4	DienThoai	Text	15	Điện thoại

2. Tệp Chức vụ - ChucVu

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Điển giải
1	#MaChucVu	Text	15	Mã chức vụ
2	TenChucVu	Text	30	Tên chức vụ

3. Tệp Quê quán - QueQuan

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Điển giải
1	#MaQueQuan	Text	15	Mã quê quán
2	TenQueQuan	Text	30	Tên quê quán

4. Tệp dân tộc - DanToc

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Điển giải
1	#MaDanToc	Text	15	Mã dân tộc
2	TenDanToc	Text	30	Tên dân tộc

5. Tệp Chuyên môn - ChuyenMon

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Điển giải
1	#MaChuyenMon	Text	15	Mã chuyên môn
2	TenChuyenMon	Text	30	Tên chuyên môn

6. Tập Trình độ - TrinhDo

Số thứ tự	Tên trường	Kiểu dữ liệu	Độ rộng	Điều kiện
1	#MaTrinhDo	Text	15	Mã trình độ
2	TenTrinhDo	Text	30	Tên trình độ

7. Tập Nhân viên - NhanVien

Số thứ tự	Tên trường	Kiểu dữ liệu	Độ rộng	Điều kiện
1	# MaNhanVien	Text	15	Mã nhân viên
2	HoDem	Text	25	Họ đệm
3	Ten	Text	25	Tên
4	NgaySinh	Date/Time	dd/mm/yyyy	Ngày sinh
5	GioiTinh	Text	3	Giới tính
6	MaQueQuan	Text	15	Mã quê quán
7	MaDanToc	Text	15	Mã dân tộc
8	DienThoai	Text	15	Điện thoại
9	Email	Text	30	Địa chỉ thư điện tử
10	DiaChi	Text	40	Nơi ở hiện tại
11	SoCMTND	Text	15	Số chứng minh thư
12	MaPhongBan	Text	15	Mã phòng ban
13	MaChucVu	Text	15	Mã chức vụ
14	MaTrinhDo	Text	15	Mã trình độ
15	MaChuyenMon	Text	15	Mã chuyên môn
16	SoHDLaoDong	Text	15	Số hợp đồng lao động
17	LuongCoBan	Number	Long Integer	Lương cơ bản
18	NgayVaoCongTy	Date/time	dd/mm/yyyy	Ngày vào công ty
19	GhiChu	Text	50	Ghi chú

8. Tập quá trình công tác - QuaTrinhCongTac

Số	Tên trường	Kiểu dữ liệu	Độ rộng	Điển giải
1	#MaQTCongTac	AutoNumber	Long Integer	Mã quá trình công tác
2	MaNhanVien	Text	15	Mã nhân viên
3	NgayBatDau	Date/Time	dd/mm/yyyy	Ngày bắt đầu làm
4	NgayKetThuc	Date/Time	dd/mm/yyyy	Ngày kết thúc làm
5	MaPhongBan	Text	15	Mã phòng ban
6	MaChucVu	Text	15	Mã chức vụ
7	MucLuongCoBan	Number	Long Integer	Mức lương cơ bản

9. Tập danh mục khen thưởng - KhenThuong

Số	Tên trường	Kiểu dữ liệu	Độ rộng	Điển giải
1	#MaKhenThuong	Text	15	Mã khen thưởng
2	NoiDung	Text	15	Nội dung khen thưởng
3	TienThuong	Number	Long Integer	Tiền thưởng
4	GhiChu	Text	50	Ghi chú

10. Tập danh mục kỷ luật - KyLuat

Số	Tên trường	Kiểu dữ liệu	Độ rộng	Điển giải
1	#MaKyLuat	Text	15	Mã kỷ luật
2	NoiDung	Text	15	Nội dung kỷ luật
3	TienPhat	Number	Long Integer	Tiền phạt
4	GhiChu	Text	50	GhiChu

11. Tập chứng từ khen thưởng - ChungTuKhenThuong

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Diễn giải
1	SoChungTuKT	Text	15	Số chứng từ khen thưởng
2	#MaNhanVien	Text	15	Mã nhân viên
3	#MaKhenThuong	Text	15	Mã khen thưởng
4	#NgayChungTuKT	Date/Time	dd/mm/yyyy	Ngày quyết định
5	GhiChu	Text	50	Ghi chú

12. Tập chứng từ kỷ luật - ChungTuKyLuat

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Diễn giải
1	SoChungTuKL	Text	15	Số chứng từ kỷ luật
2	#MaNhanVien	Text	15	Mã nhân viên
3	#MaKyLuat	Text	15	Mã kỷ luật
4	#NgayChungTuKL	Date/Time	dd/mm/yyyy	Ngày quyết định
5	GhiChu	Text	50	Ghi chú

13. Tập Nhập vật tư - NhapVatTu

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Diễn giải
1	#MaVatTu	Text	15	Mã vật tư
2	SoLuongNhap	Number	Long Integer	Số lượng nhập
3	#NgayNhapLieu	Date/time	dd/mm/yyyy	Ngày nhập liệu
4	#MaPhongBan	Text	15	Mã phòng ban

14. Tập Vật Tư - VatTu

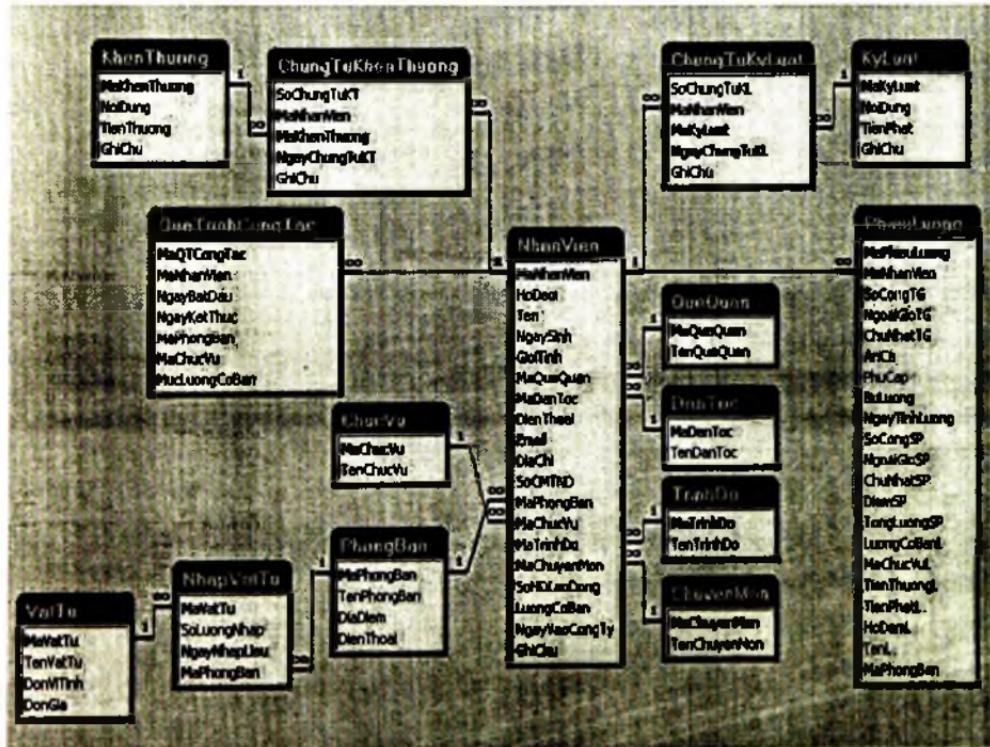
STT	Tên trường	Kiểu dữ liệu	Độ rộng	Diễn giải
1	#MaVatTu	Text	15	Mã vật tư
2	TenVatTu	Text	30	Tên vật tư
3	DonViTinh	Text	10	Đơn vị tính
4	DonGia	Number	Long Integer	Đơn giá

15. Tập Phiếu lương - PhieuLuong

Số	Tên trường	Kiểu dữ liệu	Độ rộng	Diễn giải
1	#MaPhieuLuong	Text	15	Mã phiếu lương
2	MaNhanVien	Text	15	Mã nhân viên
3	SoCongTG	Number	Double	Số công tính theo thời gian
4	NgoaiGioTG	Number	Double	Ngoài giờ tính theo thời gian
5	ChuNhatTG	Number	Double	Chủ nhật tính theo thời gian
6	AnCa	Number	Long Integer	Ăn ca
7	PhuCap	Number	Long Integer	Phụ cấp
8	BuLuong	Number	Long Integer	Bù lương
9	NgayTinhLuong	Date/time	dd/mm/yyyy	Ngày tính lương
10	SoCongSP	Number	Double	Số công tính theo sản phẩm
11	NgoaiGioSP	Number	Double	Ngoài giờ tính theo sản phẩm
12	ChuNhatSP	Number	Double	Chủ nhật tính theo sản phẩm
13	DiemSP	Number	Double	Hệ số phân bổ sản phẩm
14	LuongCoBanL	Number	Long Integer	Lương CB vào tháng tính lương
15	MaChucVuL	Text	15	Chức vụ vào tháng tính lương
16	MaPhongBan	Text	15	Phòng ban vào tháng tính lương
17	TienThuongL	Text	Long Integer	Tiền thưởng vào tháng tính lương
18	TienPhatL	Number	Long Integer	Tiền phạt vào tháng tính lương

16. Tập đăng ký người dùng - DangKy

Số	Tên trường	Kiểu dữ liệu	Độ rộng	Diễn giải
1	#TenUser	Text	25	Tên User
2	Password	Text	25	Password
3	Quyen	Text	10	Quyền
4	HoVaTen	Text	25	Họ và tên
5	NgaySinh	Date/time	dd/mm/yyyy	Ngày sinh
6	DienThoai	Text	10	Điện thoại
7	SoCMDN	Text	10	Số chứng minh nhân dân
8	ChucDanh	Text	20	Chức danh

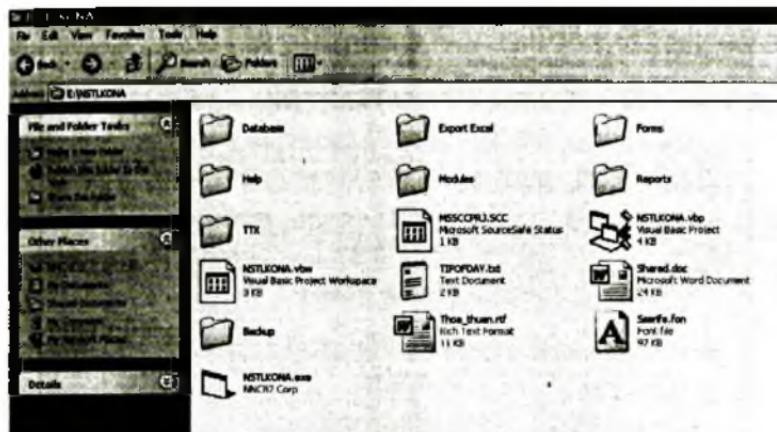


Hình 13. Bảng quan hệ giữa các thực thể

3. Lập trình

Dựa trên các sơ đồ thu được từ giai đoạn phân tích thiết kế, ta tiến hành lập trình các chức năng của phần mềm.

3.1. Cấu trúc một dự án



Để bắt đầu một dự án, bạn hãy phân chia một thư mục riêng cho dự án trên máy tính, ví dụ E:\NSTLKONA, bên trong thư mục này có các thư mục con lưu trữ các tệp tin cần thiết.

Database: Lưu trữ tệp cơ sở dữ liệu của dự án

Forms: Lưu trữ các Form thiết kế của dự án

Help: Lưu trữ tệp Hướng dẫn sử dụng của dự án

Modules: Lưu trữ các Module được sử dụng trong dự án

TTX: lưu trữ các bản mẫu báo cáo được tạo lập nhờ Crystal Report.

Reports: lưu trữ các báo cáo của dự án

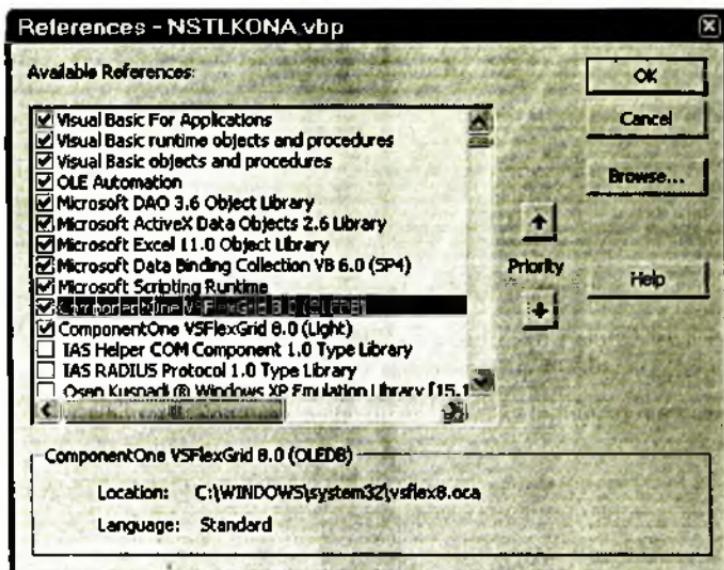
Backup: Lưu trữ tập tin sao lưu của dự án

NSTLKONA.vbp: Đây là tên dự án, nên được lưu ở thư mục gốc như hình vẽ

Tạo lập một cấu trúc rõ ràng như trên sẽ giúp bạn thuận tiện trong việc lập trình, sửa lỗi, cũng như tạo bộ cài đặt cho phần mềm sau này. Hãy tuân thủ theo đúng cấu trúc nhu trên.

- Sau khi đã tạo lập được cấu trúc như trên, bạn hãy bắt đầu một dự án mới

- Chọn một số thư viện cho dự án. Sau đây là một số thư viện hay được dùng



Sau đó hãy save dự án mới vào đúng thư mục đã tạo ra như ở phần trên.

Chú ý: Trong dự án dưới đây, ngoài Form *Tìm kiếm nhân viên* có sử dụng công nghệ *Data Access Objects - DAO*, tất cả các Form còn lại đều được lập trình duy nhất với công nghệ *Active X Data Objects - ADO*. Do đó, các điều khiển sử dụng, các thư viện liên quan đều tương thích và hỗ trợ công nghệ *ADO*.

3.2. Giới thiệu một số công cụ hỗ trợ lập trình

3.2.1. Công cụ hỗ trợ hiển thị dữ liệu

- Lưới *VS FlexGridPro 8.0* của hãng *ComponentOne*
- Lưới *Speard 8.0* của hãng *Farpoint*

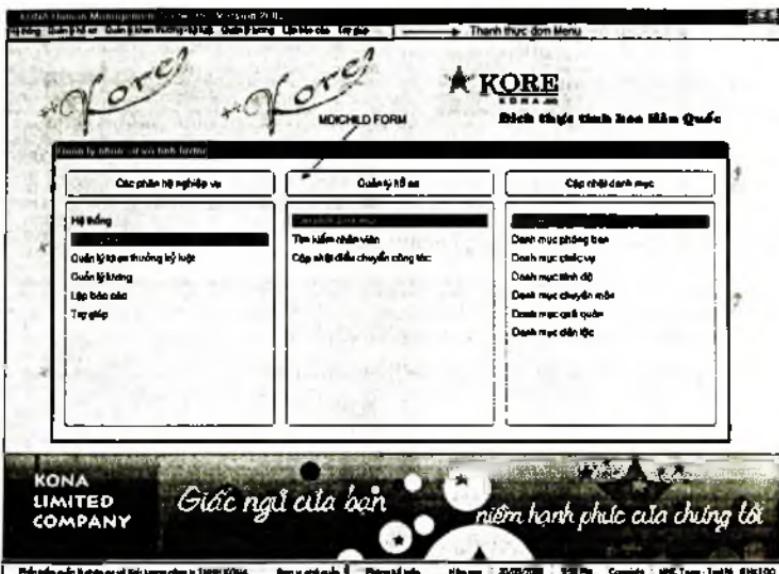
Đây là hai lưới làm việc với dữ liệu rất chuyên nghiệp dành cho VB6, cho phép bạn dễ dàng hiển thị, sửa đổi dữ liệu với giao diện đẹp, dễ sử dụng

cho người dùng, cũng như hỗ trợ tối đa người lập trình để giảm bớt gánh nặng của công việc lập trình.

3.2.2. Công cụ hỗ trợ giao diện

Có nhiều công cụ hỗ trợ giao diện khi lập trình với VB6 nhưng OsenXPSuite, OsenVistaSuite là một bộ công cụ toàn diện, hỗ trợ toàn bộ các Control phổ biến của VB6. Các control trở nên chuyên nghiệp với phong cách XP, Vista, hỗ trợ Unicode, xoá bỏ những hạn chế mà các control mặc định của VB6 mắc phải. OsenXPSuite và OsenVistaSuite thật sự hữu ích, giúp cho chương trình của bạn trở nên chuyên nghiệp và cuốn hút hơn.

3.3. Lập trình giao diện chính



Giao diện chính của một phần mềm thường có thanh thực đơn Menu, thanh công cụ Toolbar (như trong MS Word), thanh trạng thái Status, và một Form truy cập nhanh (như Fast Accounting) - để hiển thị một Form như vậy thì chỉ cần chọn thuộc tính MDIChild của Form đó là True.

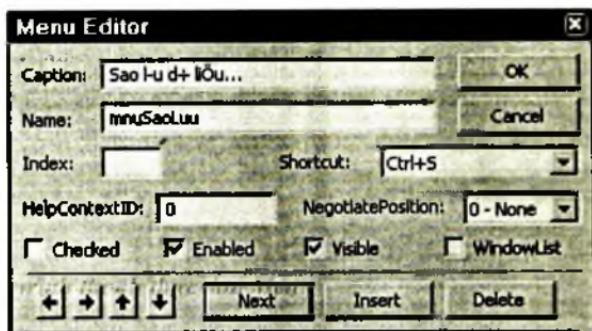
MDI Form chính là Form chứa tất cả các thành phần trên.

- Để thêm MDIForm: Project \ Add MDI Form

3.3.1. Lập trình thanh thực đơn Menu

Ở phần dưới đây, bạn sẽ học cách tạo lập thực đơn cho chương trình của mình.

- Mở MDI Form
- Tools \ Menu Editor



Caption: Khai báo tiêu đề cho menu: Ở ví dụ là “Sao lưu dữ liệu”

Name: Tên của Menu, ở ví dụ là mnuSaoLuu

Index: Trong trường hợp trùng tên menu, dùng Index để phân biệt

Shortcut: tạo phím truy cập nhanh gán cho menu

- Sau khi soạn thảo xong, bạn lập trình cho từng Menu khi được Click như sau

```
Private Sub mnuSaoLuu_Click()
    Frm_backup.Show
End Sub
```

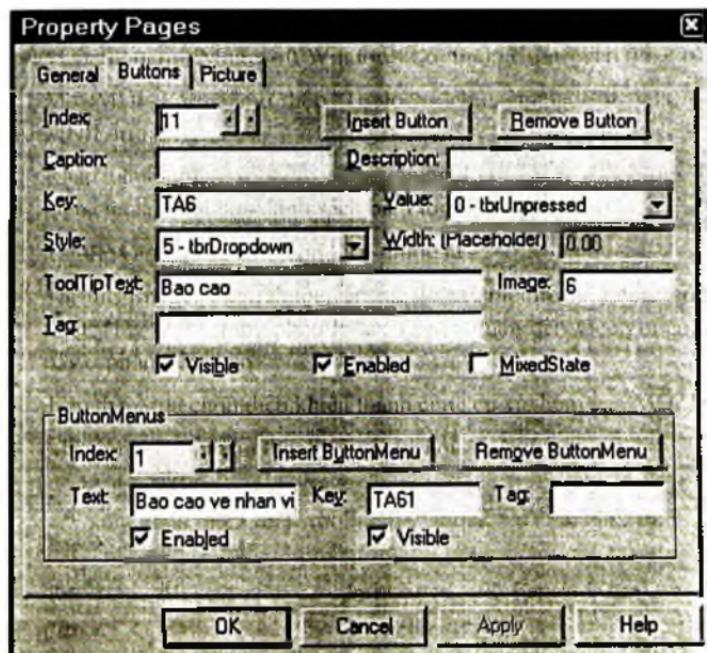
3.3.2. Lập trình thanh công cụ Toolbar

Ở phần dưới đây, bạn sẽ học cách lập trình cho thanh công cụ Toolbar, chứa những nút truy cập nhanh đến các chức năng hay dùng nhất của chương trình.



- Khai báo sử dụng điều khiển (MSCOMCTL.OCX)

- Mở MDI Form, mở hộp Toolbox, chọn ImageList và vẽ vào MDIForm
- Nhấn chuột phải vào đối tượng ImageList, chọn Properties Page, hiện cửa sổ khai báo danh sách biểu tượng sử dụng
- Tại thẻ General ta lựa chọn kích thước là 32*32
- Tại thẻ Image, bấm chuột vào nút Insert Picture để chọn icon cần thiết. Hãy chú ý rằng phần Index sẽ liên quan đến việc gọi các Icon này ở phần tiếp theo.
- Tiếp theo, lựa chọn điều khiển thanh công cụ và thêm vào MDI Form
- Nhấp phải chuột vào thanh công cụ chọn Properties



- Trong cửa sổ hiện ra, tại thẻ General, trong mục ImageList, ta chọn tên ImageList muốn liên kết sử dụng, thường là ImageList1
- Tại thẻ Buttons, ta bấm chuột lần lượt vào nút Insert Button để khai

báo thêm các nút mới trong thanh công cụ. Với mỗi nút thêm vào, ta khai báo các thông tin sau:

- Caption: khai báo tiêu đề (thường không khai báo cái này)
- Key: tên của nút, đây là phần quan trọng, vì nó liên quan đến việc lập trình, bạn hãy chú ý đặt tên gợi nhớ.
- Style: chọn kiểu của nút (thường là default). Muốn tạo ra một đường kẻ để ngăn cách giữa các nhóm nút, chọn Style là 3 - Separator
- Image: Nhập số thứ tự cho hình ảnh của biểu tượng (mà đã được định nghĩa trong mục ImageList lúc trước) sẽ được sử dụng cho nút.

Để viết lệnh đáp ứng sự kiện nút được Click, ta nháy kép chuột vào thanh công cụ làm hiện ra cửa sổ lệnh với thủ tục hiện thời là Button Click.

CODE

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Key
Case "TA1"
    frmLogin.Show ' Hiện Form Đăng nhập hệ thống
Case "TA2"
    frmNhanVien.Show ' Hiện Form Danh mục nhân viên
Case "TA3"
    FrmTongHopHNK.Show ' Hiện Form Tổng hợp vật tư nhập kho
Case "TA4"
    FrmLuongSP.Show ' Hiện Form Cập nhật ngày công và tính lương
Case "TA5"
    FrmTKNhanVien.Show ' Hiện Form Tìm kiếm nhân viên
Case "TA7"
    Frm_backup.Show ' Hiện Form Sao lưu dữ liệu
Case "TA8"
    FrmLinkWeb.Show ' Hiện Form Liên hệ trực tuyến
Case "TA9"
    ShowContents 1      ' Hiện Sách hướng dẫn sử dụng
```

Case "TA10"

```
Dim traloi  
traloi = MsgBox("Bạn muốn kết thúc chương trình thật không?",  
vbYesNo, "Thông báo")  
If traloi = vbYes Then  
    End  
End If  
End Select  
End Sub
```

- Để làm nút bấm có hình mũi tên trỏ xuống, cho phép chọn nhiều hơn một lệnh khi ấn vào nút trên thanh Toolbar (như hình vẽ), ta làm như sau:



+ Tại phần ButtonMenus của Button, ta Insert ButtonMenus tùy theo số lượng muốn thêm. Ở ví dụ này là 3 ButtonMenus. Chú ý đặt key cho các ButtonMenus như TA61, TA62, TA63.

+ Viết code cho các ButtonMenus này như sau:

```
Private Sub Toolbar1_ButtonMenuClick(ByVal ButtonMenu As  
MSComctlLib.ButtonMenu)
```

```
Select Case ButtonMenu.Key
```

```
Case "TA61"
```

```
    FrmBCTuyChon.Show ' Hiện báo cáo về nhân viên
```

```
Case "TA62"
```

```
    FrmBCKTKLTrongThang.Show ' Hiện báo cáo khen thưởng, kỷ luật
```

```
Case "TA63"
```

```
    FrmBCLuong.Show ' Hiện báo cáo về tiền lương
```

```
End Select
```

```
End Sub
```

3.3.3. Lập trình Menu chuột phải

Để khi ấn chuột phải vào giao diện chính, một Menu hiện ra cho phép bạn chọn những chức năng hay dùng nhất, ta làm như sau:

- Tạo một Menu như thông thường, nhưng ở ô Check Visible, bạn bỏ chọn để Menu này không hiện ra trên thanh thực đơn (nên đặt tên là MnuChuotPhai)

- Viết code để hưởng ứng sự kiện Click phải chuột như sau:

```
Private Sub MDIForm_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
If Button = 2 Then
```

```
    Me.PopupMenu MnuChuotPhai
```

```
End If
```

```
End Sub
```

3.4. Lập trình Sub main()

Ở phần dưới đây, bạn sẽ học cách lập trình cho module sẽ chạy đầu tiên khi chương trình của bạn bắt đầu. Đó là Sub main(). Bạn cũng làm quen với khái niệm “kết nối mềm”, chuỗi kết nối đến cơ sở dữ liệu, cách khai báo các biến, thủ tục hay dùng nhất trong chương trình.

Sub main() là module chạy đầu tiên của dự án. Chuột phải vào tên dự án, chọn Properties, cửa sổ Project Properties hiện ra, tại phần Startup Object chọn Sub main().

Tại Sub main() ta có thể khai báo các biến, các thủ tục hay được sử dụng trong dự án, như vậy khi thiết kế các Form trong dự án ta không cần phải khai báo các biến hay viết lại các thủ tục đó nữa.

CODE

```
Global STRSQL As String
```

```
Global path As String
```

```
Global Const pass = ";pwd=tin47aneu!"
```

```
Global Cn As New ADODB.Connection
```

```
Global rs As New ADODB.Recordset
```

```

Dim rsKONA As New ADODB.Recordset
Dim sqlKONA As String
Public Sub Main()
path = App.path & "\Database\NSKONA.mdb"
STRSQL = " Provider=Microsoft.Jet.OLEDB.4.0;" & _
    " Data Source= " & App.path & "\Database\NSKONA.mdb;" & _
    " Jet OLEDB:Database Password=tin47aneu!;"
Cn.ConnectionString = STRSQL
Cn.Open
rs.ActiveConnection = Cn
rs.CursorType = adOpenStatic
rs.LockType = adLockOptimistic
frmLogin.Show
End Sub

Public Sub DiChuyen(KeyCode As Integer)
If KeyCode = vbKeyReturn Then
    SendKeys "{Tab}"
End If
End Sub

```

Tại Module trên, ta đã khai báo các biến sẽ sử dụng trong khi lập trình, trong đó có 1 biến Connection và 1 biến Recordset.

STRSQL là chuỗi kết nối đến cơ sở dữ liệu khi lập trình với ADO. Chuỗi kết nối này sẽ được sử dụng thường xuyên trong chương trình.

Tại STRSQL, ta thấy có chuỗi "& App.path & "\Database\NSKONA.mdb;" - đây chính là phần kết nối mềm, thay cho kết nối cứng thường thấy như "E:\NSTLKONA\Database\NSKONA.mdb;"

Việc tạo lập kết nối mềm rất thuận lợi ở chỗ mã nguồn sẽ chạy được trên bất cứ vị trí nào của máy (khi copy mã nguồn sang máy khác), hoặc là cơ sở chính xác để tạo bộ cài đặt khi phân phối phần mềm về sau.

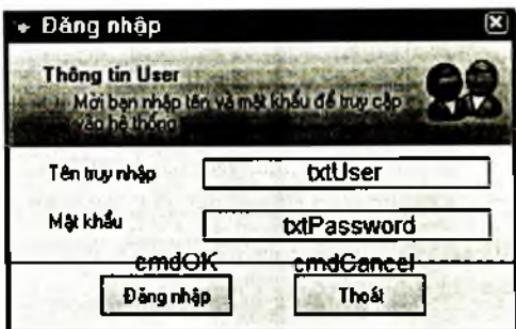
Từ nay về sau bạn hãy lập trình sử dụng chuỗi kết nối mềm với bất kỳ control hay thủ tục nào để phần mềm có thể chạy ổn định.

Đoạn STRSQL trên là chuỗi kết nối với cơ sở dữ liệu tên là NSKONA.mdb với password của cơ sở dữ liệu đó là *tin47aneu!* Để kết nối với cơ sở dữ liệu không đặt password, ta viết chuỗi kết nối như sau:

```
STRSQL = "Provider=Microsoft.Jet.OLEDB.4.0;" & _  
        " Data Source= " & App.path & "\Database\NSKONA.mdb;" & _  
        " Persist Security Info=False"
```

3.5. Lập trình Form Đăng nhập hệ thống

Ở phần dưới đây, bạn sẽ học cách viết Form đăng nhập hệ thống. Bạn cũng học cách làm quen với “lập trình hướng sự kiện” với sự kiện hay dùng nhất là Click chuột.



Form Đăng nhập hệ thống được sử dụng để tăng tính bảo mật cho chương trình.

Tại đoạn code dưới đây, chương trình sẽ kiểm tra xem Username và Password được nhập vào có tồn tại trong bảng DangKy của cơ sở dữ liệu hay không, nếu có thì sẽ đăng nhập vào chương trình, nếu không tồn tại tài khoản chương trình sẽ báo lỗi đăng nhập sai, nếu đăng nhập sai quá 3 lần thì chương trình sẽ tự thoát.

CODE

```
Dim Dem As Integer ' Khai báo biến đếm số lần đăng nhập  
Dim SQL_dangnhap As String ' Khai báo biến chuỗi  
  
Private Sub cmdCancel_Click() ' Khi Click vào nút Thoát  
End ' Thoát khỏi chương trình  
End Sub
```

```

Private Sub cmdOK_Click() ' Khi Click vào nút Đăng nhập
On Error GoTo Loi ' Bắt lỗi nếu có lỗi xảy ra
SQL_dangnhap = " select * from DangKy where TenUser=" & txtUser.Text
& " and Password = " & txtPassword.Text & ""
rs.Open SQL_dangnhap, Cn ' Mở recordset
If rs.EOF = False Then ' Nếu tồn tại tài khoản, recordset không rỗng thì
đăng nhập thành công
    Unload Me ' Đóng Form đăng nhập
    frmMenu.Show ' Mở Form giao diện chính
Else
    Dem = Dem + 1
    If Dem > 2 Then
        MsgBox "Đăng nhập sai quá 3 lần. Chương trình tự thoát",
vbInformation, "KONA HMS"
        End
    Else
        MsgBox "Đăng nhập sai ! Bạn còn " & 3 - Dem & " lần đăng
nhập", vbInformation, "KONA HMS"
        txtUser.SetFocus
    End If
End If
rs.Close ' Đóng Recordset đã mở ở trên.
Loi: ' Để khi xảy ra lỗi mà chương trình không tự thoát ra, khi gặp lỗi ta cần
Exit Sub.
If err.Number <> 0 Then
    MsgBox "Lỗi là " & err.Description
    Exit Sub
End If
End Sub
' Đoạn code dưới đây giúp cho người dùng ấn Enter là coi như click vào nút
Đăng nhập
Private Sub txtPassword_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then

```

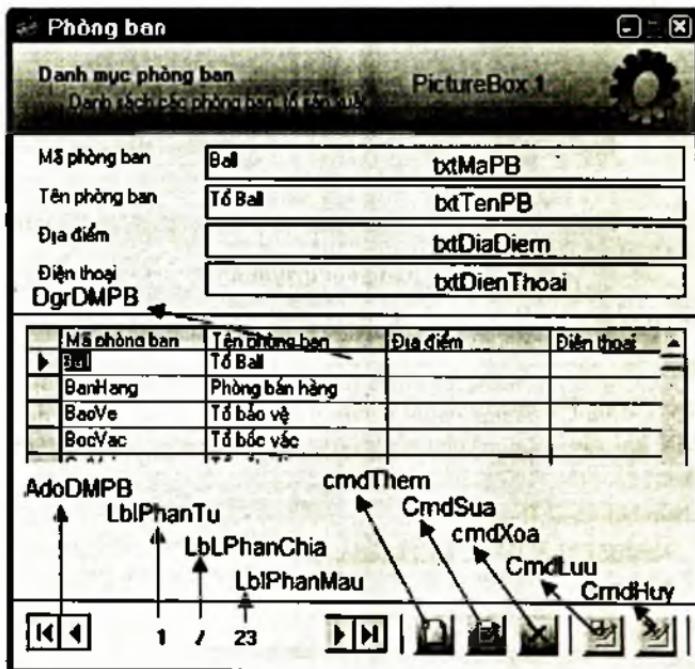
```

    cmdOK_Click
    keyascii = 0
End If
End Sub

```

3.6. Lập trình Form Danh mục phòng ban

Ở phần dưới đây, bạn sẽ học cách lập trình loại Form cập nhật danh mục, loại form xuất hiện rất nhiều tại mỗi chương trình. Bạn cũng học cách lập trình với điều khiển ADO, điều khiển lưới DataGridView, cách sử dụng hàm và thủ tục trong Form, và một số cách bắt lỗi cơ bản cho chương trình chạy ổn định.



Form Danh mục phòng ban được dùng để quản lý thông tin liên quan đến các phòng ban trong công ty. Dữ liệu sẽ được cập nhật vào Bảng PhongBan tại cơ sở dữ liệu.

Form Danh mục phòng ban gồm có các chức năng Thêm, Sửa, Xóa,

Lưu, Huỷ để cập nhật bản ghi. Hiện bản ghi hiện thời / tổng số bản ghi. Các bản ghi hiện trên một lưới để dễ theo dõi. Dữ liệu được nhập vào trên các Textbox.

Đây là một Form dạng Cập nhật danh mục khá hoàn thiện, bạn có thể ứng dụng để làm tất cả các Form liên quan đến việc cập nhật danh mục khác. Đó là các Form Danh mục nhân viên, Danh mục chức vụ, Danh mục chuyên môn, Danh mục trình độ, Danh mục Quê quán, Danh mục dân tộc, Danh mục khen thưởng, Danh mục kỷ luật.

Trong Form có sử dụng điều khiển adodc. Đây là một điều khiển liên kết dữ liệu sử dụng công nghệ ADO, là điều khiển liên kết dữ liệu ta thường dùng. Để sử dụng điều khiển này ta thêm MSADODC.ocx vào dự án. Ta add ocx theo đường dẫn sau:

Project \ Components \ Microsoft Ado Data Control 6.0 (OLEDB)

Một điều khiển mới mang tên adodc được thêm vào hộp Toolboxes. Ta chọn vẽ control adodc vào Form đang thiết kế, đặt tên là AdoDMPB.

Ở Form này có sử dụng một điều khiển nữa là DataGridView. Đây là điều khiển giúp ta hiển thị dữ liệu trên Form để người dùng tiện theo dõi. Để sử dụng điều khiển này ta thêm MSDATGRD.ocx vào dự án. Ta add ocx theo đường dẫn sau:

Project \ Components \ Microsoft DataGridView Controls 6.0 (OLEDB)

Một điều khiển mới mang tên DataGridView được thêm vào hộp Toolboxes. Ta chọn vẽ control DataGridView vào Form đang thiết kế, đặt tên là DgrDMPB.

Để thiết kế Form Danh mục phòng ban, bạn hãy thiết kế và đặt tên các control như hình vẽ trên đây.

CODE:

```
Private AddStatus As Boolean ' Biến xác định trạng thái là Thêm mới  
Private EditStatus As Boolean ' Biến xác định trạng thái là Sửa  
  
Private Sub Lock() ' Khoá các Control theo ý muốn của người lập trình  
    TxtMaPB.Locked = True  
    TxtTenPB.Locked = True  
    TxtDiaDiem.Locked = True  
    TxtDienThoai.Locked = True  
  
End Sub
```

```

Private Sub UnLock() 'Mở khoá các Control theo ý muốn của người lập trình
    TxtMaPB.Locked = False
    TxtTenPB.Locked = False
    TxtDiaDiem.Locked = False
    TxtDienThoai.Locked = False
End Sub

Private Sub SetValue() ' Thiết lập các Control hiển thị giá trị của trường liên
    quan trong CSDL
    If AddStatus = False Then
        If AdoDMPB.Recordset.EOF = False And AdoDMPB.Recordset.BOF =
        False Then
            If AddStatus = False Then
                TxtMaPB.Text = AdoDMPB.Recordset(0)
                ' Hoặc có thể viết   TxtMaPB.Text =
                AdoDMPB.Recordset.Field("MaPhongBan")
                ' Khi chỉ rõ text box liên kết với trường nào trong CSDL thì ta không cần
                phai nhớ vị trí 0, 1, 2 hay 3 của các trường MaPhongBan, TenPhongBan,
                DiaDiem, DienThoai trong CSDL nữa, và không cần phải viết các Text box
                theo đúng thứ tự các trường trong CSDL.
                TxtTenPB.Text = AdoDMPB.Recordset(1)
                TxtDiaDiem.Text = AdoDMPB.Recordset(2)
                'Nếu như trường giá trị liên kết trong CSDL là rỗng thì ô text sẽ hiện rỗng,
                ngược lại thì hiện giá trị. Cách làm này sẽ bắt lỗi Invalid Use of Null thường
                hay xảy ra làm thoát chương trình.
                If IsNull(AdoDMPB.Recordset(3)) = True Then
                    TxtDienThoai.Text = ""
                Else
                    TxtDienThoai.Text = AdoDMPB.Recordset(3)
                End If
            End If
        Else
            ResetForm

```

```

    End If
End If
End Sub

Private Sub Form_Load() ' Khi Form được load - Thủ tục sẽ chạy đầu tiên
khi mở Form
    AdoDMPB.ConnectionString = STRSQL ' STRSQL là chuỗi kết nối đã
được khai báo ở Sub main()
    AdoDMPB.CommandType = adCmdUnknown
    AdoDMPB.RecordSource = "Select * from PhongBan order by
MaPhongBan"
    Set dgrDMPB.DataSource = AdoDMPB
    Lock ' Khoá các ô Textbox, không cho sửa đổi, đảm bảo an toàn dữ liệu
    AddStatus = False ' Trạng thái Thêm mới = False
    EditStatus = False ' Trạng thái Sửa = False
    SetValue 'Gọi thủ tục SetValue
End Sub

Private Sub Form_Unload(Cancel As Integer) ' Khi ấn vào nút X trên cửa sổ
    frmMenu.Show ' Hiện Form giao diện chính
End Sub

Private Sub ResetForm() ' Xoá trắng các ô Textbox để người dùng nhập giá
trị mới
    TxtMaPB.Text = ""
    TxtTenPB.Text = ""
    TxtDiaDiem.Text = ""
    TxtDienThoai.Text = ""
End Sub

Private Sub SetStatus() ' Thủ tục thiết lập trạng thái của các nút lệnh, các
textbox, điều khiển ADO trên Form. Trạng thái này sẽ tương ứng với các nút
lệnh khi được Click, đảm bảo trạng thái các nút được Enabled hay không
một cách linh động, thuận tiện nhất cho người dùng.
If AddStatus = False Then

```

```
If EditStatus = False Then
    cmdThem.Enabled = True
    cmdSua.Enabled = True
    cmdXoa.Enabled = True
    cmdLuu.Enabled = False
    cmdHuy.Enabled = False
    AdoDMPB.Enabled = True
Else
    cmdThem.Enabled = False
    cmdSua.Enabled = False
    cmdXoa.Enabled = False
    cmdLuu.Enabled = True
    cmdHuy.Enabled = True
    AdoDMPB.Enabled = False
End If

Else
    If EditStatus = False Then
        cmdThem.Enabled = False
        cmdSua.Enabled = False
        cmdXoa.Enabled = False
        cmdLuu.Enabled = True
        cmdHuy.Enabled = True
        AdoDMPB.Enabled = False
    Else
        AddStatus = False
        EditStatus = False
        cmdThem.Enabled = True
        cmdSua.Enabled = True
        cmdXoa.Enabled = True
        cmdLuu.Enabled = False
    End If
End If
```

```

cmdHuy.Enabled = False
AdoDMPB.Enabled = True
End If
End If
End Sub

Private Sub cmdThem_Click() ' Khi người dùng ấn vào nút Thêm
    UnLock ' Mở khoá các ô Textbox, cho phép người dùng tác động
    ResetForm ' Xoá trắng các ô Textbox
    TxtMaPB.SetFocus ' Con trỏ nhấp nháy tại ô này
    AddStatus = True ' Đặt trạng thái Thêm mới = True
    SetStatus ' Gọi thủ tục SetStatus
End Sub

Private Sub cmdSua_Click() ' Khi người dùng ấn vào nút Sửa
If AdoDMPB.Recordset.RecordCount < 1 Then
    MsgBox "Không có bản ghi nào để sửa!", vbOKOnly, "KONA HMS"
Else
    UnLock ' Mở khoá các ô Textbox, cho phép người dùng tác động
    TxtMaPB.SetFocus ' Con trỏ nhấp nháy tại ô này
    EditStatus = True ' Đặt trạng thái Sửa = True
    SetStatus ' Gọi thủ tục SetStatus
    SetValue ' Gọi thủ tục SetValue
End If
End Sub

Private Sub cmdXoa_Click() ' Khi người dùng ấn vào nút Xóa
On Error GoTo Lỗi ' Bắt lỗi vì nút Xóa có thể xảy ra rất nhiều tình huống
If AdoDMPB.Recordset.EOF = False And AdoDMPB.Recordset.BOF =
False Then
    SetValue
    If rs.State = 1 Then
        rs.Close

```

```
End If

rs.Open "Select MaPhongBan from NhanVien Where MaPhongBan
= " & AdoDMPB.Recordset(0) & "", Cn

If rs.RecordCount > 0 Then
    MsgBox "Loại phòng ban này đã có trong bảng Nhân viên, không
    được xóa!", vbOKOnly, "KONA HMS"
Else
    If MsgBox("Bạn thực sự muốn xóa?", vbYesNo, "KONA HMS") =
    vbYes Then
        AdoDMPB.Recordset.Delete adAffectCurrent
        If AdoDMPB.Recordset.RecordCount < 1 Then
            ResetForm
        Else
            AdoDMPB.Recordset.MoveFirst
        End If
    End If
    End If
End If

If rs.State = 1 Then
    rs.Close
End If

Else
    MsgBox "Bạn chưa chọn loại phòng ban cần xóa!", vbOKOnly, "KONA
    HMS"
End If

Loi:
    If err.Number <> 0 Then
        MsgBox "Lỗi là " & err.Description, vbCritical, "KONA HMS"
        Exit Sub
    End If
End Sub
```

```

Private Function CheckForm() As Boolean ' Hàm kiểm tra dữ liệu được nhập
vào Textbox
Dim Flag As Boolean ' Khai báo một cờ
Flag = False
If Trim$(TxtMaPB.Text) = "" Then
    MsgBox "Bạn chưa nhập mã phòng ban!", vbOKOnly, "KONA HMS"
    TxtMaPB.SetFocus
ElseIf Trim$(TxtTenPB.Text) = "" Then
    MsgBox "Bạn chưa nhập tên phòng ban!", vbOKOnly, "KONA HMS"
    TxtTenPB.SetFocus
Else
    Flag = True
End If
CheckForm = Flag
End Function

Private Sub cmdLuu_Click() ' Khi người dùng ấn vào nút Lưu
On Error GoTo Loi ' Bắt lỗi vì nút Lưu có nhiều trường hợp xảy ra
If AddStatus = True Then ' Tình huống khi trạng thái là Thêm mới
    If CheckForm = True Then
        AdoDMPB.Recordset.AddNew
        AdoDMPB.Recordset(0) = TxtMaPB.Text
        AdoDMPB.Recordset(1) = TxtTenPB.Text
        AdoDMPB.Recordset(2) = txtDiaDiem.Text
        AdoDMPB.Recordset(3) = TxtDienThoai.Text
        AdoDMPB.Recordset.Update
        AddStatus = False
        Lock
        SetStatus
        SetValue
    End If

```

```
End If
If editStatus = True Then ' Tình huống khi trạng thái là Sửa
    If AdoDMPB.Recordset.EOF = False And AdoDMPB.Recordset.BOF =
False Then
        If checkForm = True Then
            AdoDMPB.Recordset(0) = TxtMaPB.Text
            AdoDMPB.Recordset(1) = TxtTenPB.Text
            AdoDMPB.Recordset(2) = txtDiaDiem.Text
            AdoDMPB.Recordset(3) = TxtDienThoai.Text
            AdoDMPB.Recordset.Update
            EditStatus = False
            Lock
            SetStatus
            SetValue
        End If
    Else
        MsgBox "Bạn chưa chọn phòng ban cần sửa !", vbOKOnly,
"KONA HMS"
    End If
End If
Loi:
If err.Number = -2147467259 Then
    MsgBox " Trùng khoá chính. Hãy nhập khoá khác", vbCritical, "KONA
HMS"
Exit Sub
ElseIf err.Number <> 0 Then
    MsgBox "Lỗi bất thường xảy ra. Hãy liên hệ tác giả phần mềm", vbCritical,
"KONA HMS"
Exit Sub
End If
End Sub
```

```

Private Sub cmdHuy_Click() ' Khi người dùng ấn vào nút Huỷ
If AddStatus = True Then
    AddStatus = False
End If
If EditStatus = True Then
    EditStatus = False
End If
Lock
SetStatus
SetValue
End Sub

Private Sub dgrDMPB_RowColChange(lastRow As Variant, ByVal lastCol
As Integer)
    ' Thủ tục này để cho Lưới trên Form luôn cập nhật các bản ghi, hiển thị
    tương ứng với các Textbox khi người dùng ấn vào nút hoa tiêu trên điều
    khiển AdoDMPB
    If EditStatus = False And AddStatus = False Then
        SetValue
    End If
End Sub

Private Sub AdoDMPB_MoveComplete(ByVal adReason As
ADODB.EventReasonEnum, ByVal pError As ADODB.Error, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
    ' Thủ tục này giúp việc hiện bản ghi hiện thời / tổng bản ghi luôn chính xác
    theo nút hoa tiêu.
    LblPhanTu.Caption = AdoDMPB.Recordset.AbsolutePosition
    LblPhanMau.Caption = AdoDMPB.Recordset.RecordCount
End Sub

```

3.7. Lập trình Form Danh mục nhân viên

Ở phần dưới đây, bạn sẽ học cách cài đặt và sử dụng các điều khiển không thuộc những điều khiển được cung cấp bởi Microsoft, để hỗ trợ công

việc lập trình, đại diện là điều khiển lưới VS FlexGrid Pro 8.0. Bạn cũng học cách xử lý với các nút lệnh F2, F3 ... khi được ấn trên Form.

Nhân viên								
Mã nhân viên	Họ đệm	Tên	Ngày sinh	Giới tính	Mã quê quán	Mã dân tộc	Điện thoại	Email
1-DungT	Nguyễn	Tiến	Đóng	27/1/2/1995	Nam	Hà Nội	Kinh	
2-DungD	Phạm	Duy	Bép	10/1/2/1990	Nam	Hà Nội	Kinh	
3-DungD	Nguyễn	Đại	Công	10/1/2/1997	Nam	Hà Nội	Kinh	
4-CanhT	Hoàng	Thị	Cẩm	21/6/1/1976	Nữ	Hà Nội	Kinh	
5-ChanhT	Tạ	Trí	Châm	02/10/1/1986	Nữ	Hà Nội	Kinh	
6-ChanhN	Nguyễn	Ngọc	Chiêm	19/8/2/1997	Nam	Hà Nội	Kinh	08041658888 ng_hidden_chane@yahoo.com
7-ChanhNV	Nguyễn	Văn	Chinh	02/12/2/1997	Nam	Hà Nội	Kinh	
8-DungDT	Bản	Trung	Đặng	15/6/2/1990	Nam	Nghệ An	Kinh	
9-DungND1	Nguyễn	Bắc	Đặng	20/8/2/1977	Nam	Hà Nội	Kinh	
10-DungND2	Nguyễn	Bắc	Đặng	01/10/2/1995	Nam	Hà Nội	Kinh	

Ở Form Nhân viên, dữ liệu sẽ được hiển thị trên lưới. Người sử dụng sẽ ấn phím liên quan F2, F3, ..., để thực hiện các chức năng tương ứng. Khi ấn phím thì tùy trường hợp mà chương trình sẽ đáp ứng theo đúng yêu cầu. Giả sử khi người dùng ấn F2, chương trình sẽ mở Form sau:

Cập nhật thông tin nhân viên

Cập nhật thông tin nhân viên
Cập nhật các thông tin cần thiết cho nhân viên

Mã nhân viên	txtMaNhanVien		
Họ đệm	txtHoDem	Tên	txtTen
Ngày sinh	DtpNgaySinh	Giới tính	cboGioiTinh
Quê quán	DcbMaQueQuan	Dân tộc	DcbMaDanToc
Điện thoại	txtDienThoai	Email	txtEmail
Địa chỉ hiện tại	txtDiaChi		
Số CMND	txtSoCMTND		
Phòng ban	DcbMaPhongBan	Chức vụ	DcbMaChucVu
Chuyên môn	DcbMaChuyenMon	Tình độ	DcbMaTinhDo
Số hợp đồng LD	txtSoHDLaoDong	Lương cơ bản	txtLuongCoBan
Ngày vào công ty	DTPNgayVaoCongTy		
Ghi chú	txtGhiChu		
Cập nhật		Huỷ bỏ	cmdLuu → cmdHuy

Sau đây chúng ta sẽ bắt đầu thiết kế và lập trình 2 form trên để thực hiện chức năng cập nhật danh mục nhân viên.

Ở Form Nhân viên có sử dụng một điều khiển mới là VsFlexGrid. Đây là điều khiển hiển thị dữ liệu trên lưới đã được giới thiệu ở các phần trước. Sau khi Download, cài đặt bình thường vào máy, VB6 sẽ tự nhận điều khiển mới này trong phần Components. Để sử dụng điều khiển này ta thêm vsflex8.ocx vào dự án. Ta add ocx theo đường dẫn sau:

Project \ Components \ ComponentOne VsFlexGrid 8.0 (OLEDB)

Một điều khiển mới được thêm vào hộp Toolboxes dạng hình lưới có số 8. Ta chọn vẽ control VSFlexGrid vào Form đang thiết kế, đặt tên là Vsg. Ngoài ra, hãy thiết kế thêm một control Adodc để kết nối dữ liệu, đặt tên là adodc1.

Để tùy biến điều khiển Vsg, ta ấn chọn Vsg. Trong khung Properties, chọn Custom. Trang Properties Page hiện ra, tại đây bạn có thể Insert dù các Column tương ứng với các cột dữ liệu bạn muốn hiển thị. Chú ý hãy đặt tên Header của Column một cách có thứ tự (Ví dụ Mã nhân viên, Họ đệm, Tên, Ngày sinh ...)

Với Vsg, ta cần thêm một số tùy chọn trong Properties để khi ấn vào tiêu đề của từng cột, dữ liệu sẽ được sắp xếp theo cột đó, cũng như không cho phép update trực tiếp thông tin từ lưới vào CSDL.

- Tại mục DataMode chọn là 0 - flexDMFree
- Tại mục Editable chọn là 0 - flexEDNone
- Tại mục AutoSearch chọn là 1 - flexSearchFromTop
- Tại mục ExploreBar chọn là 7 - flexExSortShowAndMove

Ở Form Cập nhật thông tin nhân viên FrmNhanVien, bạn hãy thiết kế và đặt tên các control như hình vẽ trên. Trong Form FrmNhanVien, ta có sử dụng một điều khiển mới là Datacombo. Để sử dụng điều khiển này ta thêm msdatlst.ocx vào dự án. Ta add ocx theo đường dẫn sau: Project Components \ Microsoft DataList Controls 6.0 (OLEDB)

Hai điều khiển DataList và DataCombo được thêm vào hộp Toolboxes. Ta chọn vẽ control DataCombo vào Form đang thiết kế, đặt tên theo hướng dẫn ở hình trên.

DataCombo là một hộp Combo box, cho phép người dùng lựa chọn giá

trị tương ứng trong bảng danh mục và trả về giá trị khác (trường khác), tương ứng với giá trị được lựa chọn.

Ở Form Cập nhật danh mục nhân viên, để khi ta chọn vào Data Combo quê quán, trên danh sách hiện ra là tên quê quán, nhưng khi cập nhật vào CSDL lại là mã quê quán (chọn tên lấy mã), ta thiết lập như hình vẽ sau.



Với các Datacombo còn lại làm tương tự.

CODE

Việc lập trình 2 Form Nhân viên và Cập nhật danh mục nhân viên có quan hệ chặt chẽ với nhau. Bạn hãy chú ý cách viết để xác định rõ một đối tượng trong một tuyển tập Forms, một tuyển tập Controls (ví dụ frmNhanVien.txtTen.text)

Code cho Form Nhân viên - Tên Form này là FrmVSGDMNV

```
Public NV_Flag As Long ' Phân biệt trạng thái Thêm mới hay Sửa  
Private Sub resetForm(frm As Form) ' Xoá trắng các Control  
frm.txtMaNhanVien.Text = ""  
frm.TxtHoDem.Text = ""  
frm.txtTen.Text = ""  
frm.cboGioiTinh.Text = ""  
frm.DcbMaQueQuan.BoundText = ""  
frm.DcbMaDanToc.BoundText = ""  
frm.txtDienThoai.Text = ""  
frm.TxtEmail.Text = ""
```

```
frm.txtDiaChi.Text = ""
frm.TxtSoCMTND.Text = ""
frm.DcbMaPhongBan.BoundText = ""
frm.DcbMaChucVu.BoundText = ""
frm.DcbMaTrinhDo.BoundText = ""
frm.DcbMaChuyenMon.BoundText = ""
frm.txtSoHDLaoDong.Text = ""
frm.txtLuongCoBan.Text = ""
frm.txtGhiChu.Text = ""
frm.txtMaNhanVien.SetFocus
End Sub
```

```
' Đoạn code sau dùng để gán các lệnh khi người dùng ấn phím
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
If KeyCode = vbKeyEscape Then
    Unload Me
ElseIf KeyCode = vbKeyF2 Then
    FrmNhanVien.Show
    NV_Flag = 1
Call resetForm(FrmNhanVien)
ElseIf KeyCode = vbKeyF5 Then
    MsgBox "Click chuột vào tiêu đề cột bạn muốn sắp xếp",
vbInformation, "KONA HMS"
ElseIf KeyCode = vbKeyF6 Then
    FrmTKNhanVien.Show
ElseIf KeyCode = vbKeyF3 Then
    i = VSg.row
    NV_Flag = 2
Call NhanVienSua(FrmNhanVien)
FrmNhanVien.Show
FrmNhanVien.txtMaNhanVien.Enabled = False
```

```
FrmNhanVien.DcbMaChucVu.Enabled = False  
FrmNhanVien.DcbMaPhongBan.Enabled = False  
FrmNhanVien.DTPNgayVaoCongTy.Enabled = False  
FrmNhanVien.txtLuongCoBan.Enabled = False  
ElseIf KeyCode = vbKeyF4 Then  
    i = VSg.row  
    Call XoaBanGhi  
    Me.Show  
End If  
End Sub
```

```
Private Sub XoaBanGhi()  
    Dim sqlXoa As String  
    i = VSg.row  
    On Error GoTo Loi  
    If MsgBox("Bạn thực sự muốn xóa nhân viên có mã là " &  
        VSg.TextMatrix(i, 1) & "?", vbYesNo, "KONA HMS") = vbYes Then  
        sqlXoa = "Delete * from NhanVien where MaNhanVien=" &  
        VSg.TextMatrix(i, 1) & """  
        Cn.Execute sqlXoa  
        Unload Me  
        MsgBox "Bản ghi đã bị xoá", vbInformation, "KONA HMS"  
    End If  
    Loi:  
    If err.Number <> 0 Then  
        MsgBox "Lỗi là " & err.Description & " Hãy liên hệ tác giả để sửa chữa.",  
        vbCritical, "KONA HMS"  
    Exit Sub  
End If  
End Sub
```

```

Private Sub Form_Load()
With Me.VSg ' Đoạn code này để hiển thị số thứ tự của bản ghi
Dim r
For r = 1 To .Rows - 1
    .TextMatrix(r, 0) = r
Next
.SelectionMode = flexSelectionListBox
End With
With Adodc1
.ConnectionString = STRSQL
.CommandType = adCmdText
.RecordSource = "Select * from NhanVien order by Ten"
End With
Set VSg.DataSource = Adodc1 ' Gán nguồn dữ liệu cho lưới VSG.
End Sub

Public Sub NhanVienSua(frm As Form)
Dim k
k = VSg.row
Dim sqlsua As String
Dim rssua As ADODB.Recordset
sqlsua = "Select * from NhanVien where MaNhanVien ='" &
VSg.TextMatrix(k, 1) & "'"
Set rssua = New ADODB.Recordset
rssua.Open sqlsua, Cn, adOpenDynamic, adLockOptimistic
With rssua
    frm.txtMaNhanVien.Text = .Fields("MaNhanVien")
    frm.TxtHoDem.Text = .Fields("HoDem")
    frm.txtTen.Text = .Fields("Ten")
    frm.DTPNgaySinh.Value = .Fields("NgaySinh")
    frm.cboGioiTinh.Text = .Fields("GioiTinh")
End With

```

```
frm.DcbMaQueQuan.BoundText = .Fields("MaQueQuan")
frm.DcbMaDanToc.BoundText = .Fields("MaDanToc")
frm.txtDienThoai.Text = .Fields("DienThoai")
frm.TxtEmail.Text = .Fields("Email")
frm.txtDiaChi.Text = .Fields("DiaChi")
frm.TxtSoCMTND.Text = .Fields("SoCMTND")
frm.DcbMaPhongBan.BoundText = .Fields("MaPhongBan")
frm.DcbMaChucVu.BoundText = .Fields("MaChucVu")
frm.DcbMaChuyenMon.BoundText = .Fields("MaChuyenMon")
frm.DcbMaTrinhDo.BoundText = .Fields("MaTrinhDo")
frm.txtSoHDLaoDong.Text = .Fields("SoHDLaoDong")
frm.txtLuongCoBan.Text = .Fields("LuongCoBan")
frm.DTPNgayVaoCongTy.Value = .Fields("NgayVaoCongTy")
frm.txtGhiChu.Text = .Fields("GhiChu")
```

End With

End Sub

Code cho Form Cập nhật thông tin nhân viên - tên Form là FrmNhanVien

Private Sub cmdHuy_Click()

Unload Me

End Sub

Private Sub cmdLuu_Click()

Dim sqlnv As String

Dim rsnv As ADODB.Recordset

Set rsnv = New ADODB.Recordset

Dim sqlsua As String

On Error GoTo Loi

If FrmVSGDMNV.NV_Flag = 1 Then ' Nút lệnh ở trạng thái Thêm mới

If checkForm = True Then

AdoDMNhanVien.Recordset.AddNew

AdoDMNhanVien.Recordset.Fields("MaNhanVien") =

txtMaNhanVien.Text

AdoDMNhanVien.Recordset.Fields("HoDem") = TxtHoDem.Text

AdoDMNhanVien.Recordset.Fields("Ten") = txtTen.Text

AdoDMNhanVien.Recordset.Fields("NgaySinh") = DTNgaySinh.Value

AdoDMNhanVien.Recordset.Fields("GioiTinh") = cboGioiTinh.Text

AdoDMNhanVien.Recordset.Fields("MaQueQuan") =

DcbMaQueQuan.BoundText

AdoDMNhanVien.Recordset.Fields("MaDanToc") =

DcbMaDanToc.BoundText

AdoDMNhanVien.Recordset.Fields("DienThoai") = txtDienThoai.Text

AdoDMNhanVien.Recordset.Fields("Email") = TxtEmail.Text

AdoDMNhanVien.Recordset.Fields("DiaChi") = txtDiaChi.Text

AdoDMNhanVien.Recordset.Fields("SoCMTND") =

TxtSoCMTND.Text

AdoDMNhanVien.Recordset.Fields("MaPhongBan") =

DcbMaPhongBan.BoundText

AdoDMNhanVien.Recordset.Fields("MaChucVu") =

DcbMaChucVu.BoundText

AdoDMNhanVien.Recordset.Fields("MaChuyenMon") =

DcbMaChuyenMon.BoundText

AdoDMNhanVien.Recordset.Fields("MaTrinhDo") =

DcbMaTrinhDo.BoundText

AdoDMNhanVien.Recordset.Fields("SoHDLaoDong") =

txtSoHDLaoDong.Text

AdoDMNhanVien.Recordset.Fields("LuongCoBan") =

txtLuongCoBan.Text

AdoDMNhanVien.Recordset.Fields("NgayVaoCongTy") =

DTNgayVaoCongTy.Value

AdoDMNhanVien.Recordset.Fields("GhiChu") = txtGhiChu.Text

AdoDMNhanVien.Recordset.Update

----- Cập nhật quá trình công tác đầu tiên vào bảng quá trình công tác
sqlInv = "Select * from QuaTrinhCongTac"

```

rsnv.Open sqlnv, Cn, adOpenDynamic, adLockOptimistic
rsnv.AddNew
rsnv.Fields("MaNhanVien") = txtMaNhanVien.Text
rsnv.Fields("NgayBatDau") = DTPNgayVaoCongTy.Value
rsnv.Fields("MaPhongBan") = DcbMaPhongBan.BoundText
rsnv.Fields("MaChucVu") = DcbMaChucVu.BoundText
rsnv.Fields("MucLuongCoBan") = txtLuongCoBan.Text
rsnv.Update
'----- Hiện dữ liệu vừa cập nhật trên lưới dữ liệu Nhân viên
Unload FrmVSGDMNV
Unload Me
Load FrmVSGDMNV
FrmVSGDMNV.Show
End If
End If
If FrmVSGDMNV.NV_Flag = 2 Then 'Nút lệnh ở trạng thái Sửa
If checkForm = True Then
sqlsua = "Update NhanVien SET MaNhanVien=" &
txtMaNhanVien.Text & ", HoDem ='" & TxtHoDem.Text & "', ten=''" &
txtTen.Text & "', NgaySinh ='" & DTPNgaySinh.Value & "', GioiTinh ='" &
cboGioiTinh.Text & "', MaQueQuan ='" & DcbMaQueQuan.BoundText & "'",
MaDanToc ='" & DcbMaDanToc.BoundText & "', DienThoai ='" &
txtDienThoai.Text & "', Email ='" & TxtEmail.Text & "', DiaChi ='" &
txtDiaChi.Text & "', SoCMTND ='" & TxtSoCMTND.Text & "'",
MaPhongBan ='" & DcbMaPhongBan.BoundText & "', MaChucVu ='" &
DcbMaChucVu.BoundText & "', MaTrinhDo ='" &
DcbMaTrinhDo.BoundText & "', Machuyenmon ='" &
DcbMaChuyenMon.BoundText & "', SoHDLaoDong ='" &
txtSoHDLaoDong.Text & "', LuongCoban ='" & txtLuongCoBan.Text & "'",
NgayVaoCongTy ='" & DTPNgayVaoCongTy.Value & "'",
GhiChu ='" & txtGhiChu.Text & "' Where manhanvien=" & txtMaNhanVien.Text & "'"

```

Cn.Execute sqlsua ' Sử dụng phương pháp Execute của đối tượng Connection

```

FrmVSGDMNV.Adodc1.Refresh
FrmVSGDMNV.VSg.Refresh
Unload FrmVSGDMNV
MsgBox "Bản ghi đã được sửa", vbInformation, "KONA HMS"
Unload Me
Load FrmVSGDMNV
FrmVSGDMNV.Show

End If
End If
Loi:
If err.Number = -2147467259 Then
    MsgBox "Trùng khoá chính. Hãy nhập khoá khác", vbCritical, "KONA
HMS"
Exit Sub
ElseIf err.Number = -214746259 Then
    MsgBox "Bạn không thể thay đổi hay thêm một bản ghi vì một bản ghi liên
quan cần phải có trong các bảng con.", vbInformation, "KONA HMS"
Exit Sub
ElseIf err.Number <> 0 Then
    MsgBox "Lỗi bất thường xảy ra. Hãy liên hệ tác giả phần mềm", vbCritical,
"KONA HMS"
Exit Sub
End If
End Sub

```

```

Private Function checkForm() As Boolean
Dim flag As Boolean
flag = False
If Trim$(txtMaNhanVien.Text) = "" Then
    MsgBox "Bạn chưa nhập mã nhân viên!", vbOKOnly, "Thông báo !"
    txtMaNhanVien.SetFocus

```

```
ElseIf Trim$(TxtHoDem.Text) = "" Then
    MsgBox "Bạn chưa nhập họ đệm !", vbOKOnly, "Thông báo !"
    TxtHoDem.SetFocus
ElseIf Trim$(TxtHoDem.Text) = "" Then
    MsgBox "Bạn chưa nhập tên nhân viên !", vbOKOnly, "Thông báo !"
    txtTen.SetFocus
ElseIf Trim$(cboGioiTinh.Text) = "" Then
    MsgBox "Bạn chưa nhập giới tính !", vbOKOnly, "Thông báo !"
    cboGioiTinh.SetFocus
ElseIf Trim$(txtDiaChi.Text) = "" Then
    MsgBox "Bạn chưa nhập địa chỉ !", vbOKOnly, "Thông báo !"
    txtDiaChi.SetFocus
ElseIf Trim$(DcbMaQueQuan.BoundText) = "" Then
    MsgBox "Bạn chưa nhập quê quán !", vbOKOnly, "Thông báo !"
    DcbMaQueQuan.SetFocus
ElseIf Trim$(DcbMaDanToc.BoundText) = "" Then
    MsgBox "Bạn chưa nhập dân tộc !", vbOKOnly, "Thông báo !"
    DcbMaDanToc.SetFocus
ElseIf Trim$(DcbMaPhongBan.BoundText) = "" Then
    MsgBox "Bạn chưa nhập phòng ban !", vbOKOnly, "Thông báo !"
    DcbMaPhongBan.SetFocus
ElseIf Trim$(DcbMaChucVu.BoundText) = "" Then
    MsgBox "Bạn chưa nhập chức vụ !", vbOKOnly, "Thông báo !"
    DcbMaChucVu.SetFocus
ElseIf Trim$(DcbMaChuyenMon.BoundText) = "" Then
    MsgBox "Bạn chưa nhập chuyên môn !", vbOKOnly, "Thông báo !"
    DcbMaChuyenMon.SetFocus
ElseIf Trim$(DcbMaTrinhDo.BoundText) = "" Then
    MsgBox "Bạn chưa nhập trình độ !", vbOKOnly, "Thông báo !"
    DcbMaTrinhDo.SetFocus
```

```

ElseIf Trim$(txtLuongCoBan.Text) = "" Then
    MsgBox "Bạn chưa nhập mức Lương cơ bản!", vbOKOnly, "Thông báo !"
    txtLuongCoBan.SetFocus
ElseIf Not IsNumeric(txtLuongCoBan.Text) Then
    MsgBox "Lương cơ bản phải là số!", vbOKOnly, "Thông báo !"
    txtLuongCoBan.SetFocus
ElseIf TxtSoCMTND.Text <> "" And Not IsNumeric(TxtSoCMTND.Text)
Then
    MsgBox "Số chứng minh thư phải là số!", vbOKOnly, "Thông báo !"
    TxtSoCMTND.SetFocus
ElseIf txtDienThoai.Text <> "" And Not IsNumeric(txtDienThoai.Text) Then
    MsgBox "Số điện thoại phải là số!", vbOKOnly, "Thông báo !"
    txtDienThoai.SetFocus
Else
    flag = True
End If
checkForm = flag
End Function

Private Sub Form_Load()
' Tại Form_load này, ta thiết lập các kết nối đến điều khiển adodc liên quan
được sử dụng trong databound combo tương ứng.
AdoDMNhanVien.ConnectionString = STRSQL ' Chuỗi kết nối
AdoDMNhanVien.CommandType = adCmdUnknown
AdoDMNhanVien.RecordSource = "Select * from NhanVien order by Ten"
Set DgrDMNhanVien.DataSource = AdoDMNhanVien
AdoQueQuan.ConnectionString = STRSQL
AdoQueQuan.CommandType = adCmdText
AdoQueQuan.RecordSource = "Select * from QueQuan Order by
MaQueQuan"
AdoQueQuan.Refresh

```

```
Do While Not AdoQueQuan.Recordset.EOF
AdoQueQuan.Recordset.MoveNext
Loop

AdoDanToc.ConnectionString = STRSQL
AdoDanToc.CommandType = adCmdTable
AdoDanToc.RecordSource = "DanToc"
AdoDanToc.Refresh
Do While Not AdoDanToc.Recordset.EOF
AdoDanToc.Recordset.MoveNext
Loop

AdoPhongBan.ConnectionString = STRSQL
AdoPhongBan.CommandType = adCmdText
AdoPhongBan.RecordSource = "Select * from PhongBan order by
TenPhongBan"
AdoPhongBan.Refresh
Do While Not AdoPhongBan.Recordset.EOF
AdoPhongBan.Recordset.MoveNext
Loop

AdoChucVu.ConnectionString = STRSQL
AdoChucVu.CommandType = adCmdText
AdoChucVu.RecordSource = "Select * from ChucVu order by
TenChucVu"
AdoChucVu.Refresh
Do While Not AdoChucVu.Recordset.EOF
AdoChucVu.Recordset.MoveNext
Loop

AdoChuyenMon.ConnectionString = STRSQL
AdoChuyenMon.CommandType = adCmdText
AdoChuyenMon.RecordSource = "Select * from ChuyenMon order by
TenChuyenMon"
```

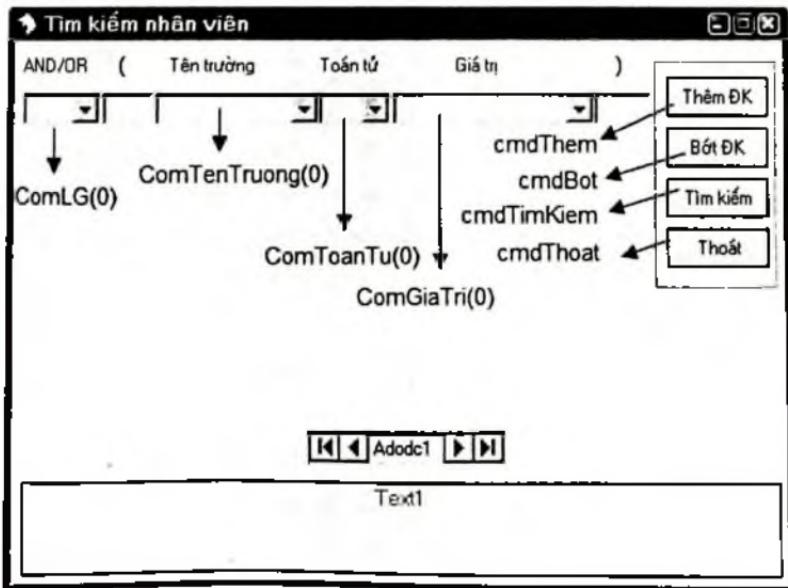
```

AdoChuyenMon.Refresh
Do While Not AdoChuyenMon.Recordset.EOF
AdoChuyenMon.Recordset.MoveNext
Loop
AdoTrinhDo.ConnectionString = STRSQL
AdoTrinhDo.CommandType = adCmdText
AdoTrinhDo.RecordSource = "Select * from TrinhDo order by
TenTrinhDo"
AdoTrinhDo.Refresh
Do While Not AdoTrinhDo.Recordset.EOF
AdoTrinhDo.Recordset.MoveNext
Loop
End Sub

```

3.8. Lập trình Form Tìm kiếm nhân viên

Ở phần dưới đây bạn sẽ học cách sử dụng mảng biến, cách viết câu lệnh truy vấn SQL để sử dụng trong Form tìm kiếm.



Đây là Form tìm kiếm tổng quát, bạn có thể áp dụng để viết các Form tìm kiếm khác.

Form tìm kiếm này có thể cho phép người dùng tùy chọn trường hiển thị, tùy chọn toán tử, tùy chọn giá trị tìm kiếm. Chương trình cho phép thêm, bớt đến 9 điều kiện. Ô text box cho phép hiện câu lệnh tìm kiếm đã được người dùng chọn.

```
Dim db As DAO.Database
Dim res As DAO.Recordset
Dim res1 As DAO.Recordset
Dim sql As String
Dim SDK As Integer
Dim i As Integer
Public sqlDK As String
Public Sub DiChuyen(KeyCode As Integer)
If KeyCode = vbKeyReturn Then
    SendKeys "{Tab}"
End If
End Sub

Sub VaoToanTu(ByVal cbo As ComboBox)
    cbo.Text = "="
    cbo.AddItem "="
    cbo.AddItem ">"
    cbo.AddItem "<"
    cbo.AddItem ">="
    cbo.AddItem "<="
    cbo.AddItem "<>"
    cbo.AddItem "Like"
End Sub
```

```
Sub VaoLG(ByVal cbo As ComboBox)
    cbo.Text = "AND"
    cbo.AddItem "OR"
    cbo.AddItem "AND"
End Sub

Sub VaoTenTruong(ByVal cbo As ComboBox)
    cbo.AddItem "HoDem"
    cbo.AddItem "Ten"
    cbo.AddItem "NgaySinh"
    cbo.AddItem "GioiTinh"
    cbo.AddItem "MaPhongBan"
    cbo.AddItem "MaChucVu"
    cbo.AddItem "MaTrinhDo"
    cbo.AddItem "MaChuyenMon"
    cbo.AddItem "MaQueQuan"
    cbo.AddItem "MaDanToc"
    cbo.AddItem "DienThoai"
    cbo.AddItem "SoCMTND"
    cbo.AddItem "SoHDLaoDong"
    cbo.AddItem "LuongCoBan"
    cbo.AddItem "NgayVaoCongTy"
    cbo.Text = "Ten"
End Sub

Private Sub cmdbot_Click()
If SDK > 1 Then
    Unload comLG(SDK - 1)
    Unload txtngoaicM(SDK - 1)
    Unload comTentruong(SDK - 1)

```

```
Unload comtoantu(SDK - 1)
Unload comgiatri(SDK - 1)
Unload txtngoaCD(SDK - 1)
SDK = SDK - 1

End If
Call VietSQL
End Sub

Private Sub cmdThoat_Click()
sqlDK = "KT"
Unload Me
frmMenu.Show
End Sub

Private Sub cmdTimKiem_Click()
On Error GoTo Loi
Dim sqlHT As String
Call VietSQL
sqlDK = "Select * From NhanVien Where " & txtDK.Text
Adodc1.ConnectionString = STRSQL
Adodc1.CommandType = adCmdText
Adodc1.RecordSource = sqlDK
Adodc1.Refresh
If Adodc1.Recordset.RecordCount > 0 Then
Set FrmVSGDMNV.VSg.DataSource = Me.Adodc1 'Hiện kết quả tìm kiếm
lên Form Nhân viên - Vừa dễ theo dõi - Vừa có thể sửa đổi thông tin cần
thiết.
FrmVSGDMNV.Show
Else
MsgBox "Không tìm thấy kết quả nào! ", vbInformation, "KONA HMS"
End If
```

```

Loi:
If err.Number = 13 Or err.Number = 3075 Then
    MsgBox "Có lỗi trong câu lệnh truy vấn. Hãy kiểm tra lại điều kiện
tìm kiếm.", vbCritical, "KONA HMS"
    Exit Sub
ElseIf err.Number <> 0 Then
    MsgBox err.Description
    Exit Sub
End If
End Sub

Private Sub cmdThem_Click()
If SDK < 9 Then
    SDK = SDK + 1
    Load comLG(SDK - 1)
    comLG(SDK - 1).Left = comLG(0).Left
    comLG(SDK - 1).Top = comLG(SDK - 2).Top + 400
    comLG(SDK - 1).Visible = True
    Call VaoLG(comLG(SDK - 1))
    Load txtngoacM(SDK - 1)
    txtngoacM(SDK - 1).Left = txtngoacM(0).Left
    txtngoacM(SDK - 1).Top = txtngoacM(SDK - 2).Top + 400
    txtngoacM(SDK - 1).Text = ""
    txtngoacM(SDK - 1).Visible = True
    Load comTentruong(SDK - 1)
    comTentruong(SDK - 1).Left = comTentruong(0).Left
    comTentruong(SDK - 1).Top = comTentruong(SDK - 2).Top + 400
    comTentruong(SDK - 1).Visible = True
    comTentruong(SDK - 1).Clear

```

```
Call VaoTenTruong(comTentruong(SDK - 1))

Load comtoantu(SDK - 1)
    comtoantu(SDK - 1).Left = comtoantu(0).Left
    comtoantu(SDK - 1).Top = comtoantu(SDK - 2).Top + 400
    comtoantu(SDK - 1).Visible = True
    Call VaoToanTu(comtoantu(SDK - 1))

Load comgiatri(SDK - 1)
    comgiatri(SDK - 1).Left = comgiatri(0).Left
    comgiatri(SDK - 1).Top = comgiatri(SDK - 2).Top + 400
    comgiatri(SDK - 1).Visible = True

Load txtngoacD(SDK - 1)
    txtngoacD(SDK - 1).Left = txtngoacD(0).Left
    txtngoacD(SDK - 1).Top = txtngoacD(SDK - 2).Top + 400
    txtngoacD(SDK - 1).Visible = True
    txtngoacD(SDK - 1).Text = ""

Else
    MsgBox "Không thể thêm được nữa. Phức tạp quá rồi !", vbInformation,
    "KONA HMS"
End If

Call VietSQL
End Sub

Private Sub comgiatri_KeyDown(Index As Integer, KeyCode As Integer,
Shift As Integer)
    Call DiChuyen(KeyCode)
End Sub

Private Sub comgiatri_LostFocus(Index As Integer)
    Call VietSQL
End Sub
```

```
Private Sub comLG_KeyDown(Index As Integer, KeyCode As Integer, Shift As Integer)
Call DiChuyen(KeyCode)
End Sub

Private Sub comTentruong_Change(Index As Integer)
sql = comTentruong(Index).Text
comTentruong(Index).ToolTipText = comTentruong(Index).Text
End Sub

Private Sub comTentruong_Click(Index As Integer)
sql = comTentruong(Index).Text
If sql <> "Noidung" And sql <> "Loigiai" Then
Call Vaogiatri(comgiatri(Index), sql)
Else
comgiatri(Index).Clear
End If
comTentruong(Index).ToolTipText = comTentruong(Index).Text
Call ChuThich(Index)
End Sub

Private Sub comTentruong_KeyDown(Index As Integer, KeyCode As Integer, Shift As Integer)
Call DiChuyen(KeyCode)
End Sub

Private Sub comtoantu_KeyDown(Index As Integer, KeyCode As Integer, Shift As Integer)
Call DiChuyen(KeyCode)
End Sub

Private Sub comtoantu_LostFocus(Index As Integer)
If (res1.Fields(comTentruong(SDK - 1).Text).Type <> dbText) And
```

```

(comtoantu(SDK - 1).Text = "Like") And (res1.Fields(comTentruong(SDK - 1).Text).Type <> dbMemo) Then
    MsgBox "Không sử dụng toán tử Like với kiểu dữ liệu không phải dạng Text."
    comtoantu(SDK - 1).Text = "="
End If

End Sub

Private Sub Form_Load()
Set db = OpenDatabase(path, False, False, pass)
SDK = 1
Set res1 = db.OpenRecordset("Select * From NhanVien")
Call VaoTenTruong(comTentruong(SDK - 1))
Call VaoToanTu(comtoantu(SDK - 1))
comLG(SDK - 1).Visible = False
sql = comTentruong(Index).Text
If sql <> "Noidung" And sql <> "Loigiai" Then
    Call Vaogiatri(comgiatri(0), sql)
Else
    comgiatri(0).Clear
End If
End Sub

Public Sub VietSQL()
On Error GoTo err:
txtDK = ""
For i = 0 To SDK - 1
    txtDK = txtDK & " " & comLG(i).Text & " " & txtngoacM(i).Text &
    comTentruong(i).Text & " " & comtoantu(i).Text & " "
    If (res1.Fields(comTentruong(i).Text).Type = dbText Or
        res1.Fields(comTentruong(i).Text).Type = dbMemo) And comtoantu(i).Text

```

```

= "Like" And res1.Fields(comTentruong(i).Text).Type <> dbDate Then
    txtDK = txtDK & "***" & comgiatri(i).Text & "***"
End If

If (res1.Fields(comTentruong(i).Text).Type = dbText Or
res1.Fields(comTentruong(i).Text).Type = dbMemo) And comtoantu(i).Text
<> "Like" And res1.Fields(comTentruong(i).Text).Type <> dbDate Then
    txtDK = txtDK & "##" & comgiatri(i).Text & "##"
End If

If res1.Fields(comTentruong(i).Text).Type <> dbText And
res1.Fields(comTentruong(i).Text).Type <> dbMemo And
comtoantu(i).Text <> "Like" And res1.Fields(comTentruong(i).Text).Type
<> dbDate Then
    txtDK = txtDK & comgiatri(i).Text
End If

If res1.Fields(comTentruong(i).Text).Type = dbDate Then
    txtDK = txtDK & "#" & comgiatri(i).Text & "#"
End If

txtDK = txtDK & txtngoaocD(i).Text
Next
Exit Sub
err:
MsgBox err.Description
End Sub

Private Sub txtngoaocD_KeyDown(Index As Integer, KeyCode As Integer,
Shift As Integer)
Call DiChuyen(KeyCode)
End Sub

Private Sub txtngoaocD_LostFocus(Index As Integer)
Call VietSQL

```

```

End Sub

Private Sub txtIngoacM_KeyDown(Index As Integer, KeyCode As Integer,
Shift As Integer)
Call DiChuyen(KeyCode)
End Sub

Public Sub Vaogiatri(ByVal cbo As ComboBox, Truong As String)
On Error GoTo err:
cbo.Clear
Dim TTruong, Ssql As String
Ssql = ""
TTruong = Truong
Ssql = "Select distinct " & Truong & " From NhanVien Order by " & Truong
Set res = db.OpenRecordset(Ssql)
res.MoveFirst
Do While Not res.EOF
cbo.AddItem IIf(IsNull(res.Fields(0).Value), "", res.Fields(0).Value)
res.MoveNext
Loop
res.Close
Exit Sub
err:
MsgBox err.Description
End Sub

```

3.9. Lập trình Form Quản lý chứng từ khen thưởng

Ở phần dưới đây, bạn sẽ học cách lập trình để khi người dùng ấn vào một mã nhân viên hay mã khen thưởng trong cơ sở dữ liệu thì những thông tin còn lại liên quan đến mã nhân viên và mã khen thưởng sẽ tự động hiện ra. Bạn cũng học cách để lập trình cho số chứng từ tự động tăng thêm một khi người dùng thêm mới một chứng từ.

Quản lý chứng từ khen thưởng

Lập chứng từ

Số chứng từ	4	txtSoChungTu
Ngày lập chứng từ	31/03/2009	DtpNgayCTKT
Thông tin nhân viên		
Mã nhân viên	ChiếnNN	cboMaNhanVien
Họ và tên	Nguyễn Ngọc Chiến	txtHoVaTen
Phòng ban	Phòng kế toán	txtPhongBan
Chức vụ	Kế toán viên	txtChucVu
Thông tin khen thưởng		
Mã khen thưởng	KT001	cboMaKhenThuong
Nội dung	Hoàn thành vượt 5 - 20% kế hoạch	
Tiền thưởng	50000	txtTienThuong
Ghi chú	txtGhiChu	

cmdInChungTu

[Print] [Back] [Forward] [Home] [Search] [Exit]

Chức năng của Form Quản lý chứng từ khen thưởng là lập và theo dõi các chứng từ khen thưởng của nhân viên vào mỗi tháng. Giả sử trên một chứng từ khen thưởng gồm có các thông tin sau: số chứng từ, ngày lập chứng từ, mã nhân viên, họ tên, phòng ban, chức vụ, mã khen thưởng, nội dung, tiền thưởng, ghi chú.

Để lập trình cho form này, ta sẽ lấy thông tin nhân viên tại bảng NhanVien, lấy thông tin khen thưởng tại bảng KhenThuong trong cơ sở dữ liệu. Dữ liệu để quản lý chứng từ khen thưởng sẽ được lưu vào bảng ChungTuKhenThuong.

Về cơ bản, việc lập trình Form này tương tự như việc viết Form cập nhật danh mục, chỉ khác là có thêm chức năng in chứng từ. Do vậy phần code dưới đây chỉ tập trung vào những sự khác biệt tại Form này so với

dang Form cập nhật danh mục. Cách thức lập trình in báo cáo sẽ được nói tới ở các phần sau.

Áp dụng cách làm tương tự với Form Quản lý chứng từ kỷ luật.

Ở Form này có sử dụng một điều khiển mới là DTPicker. Đây là điều khiển giúp cho việc chọn ngày tháng có tính chất đồ họa. Để sử dụng điều khiển này ta thêm mscomct2.ocx vào dự án. Ta add ocx theo đường dẫn sau:

Project \ Components \ Microsoft Windows Common Controls-2 6.0 (sp4)

Một số điều khiển mới được thêm vào hộp Toolboxes. Ta chọn vẽ control DTPicker vào Form đang thiết kế, đặt tên là DTPNgayCTKT

CODE

' Hàm Tạo mã tự động sau đây được dùng để số chứng từ của Form sẽ tự động tăng lên một khi người dùng ấn vào nút Thêm. Tại nút Thêm ta chỉ cần cho txtSoChungTu.text = Tao_ma_tu_dong là được. Chủ ý trong CSDL tại bảng ChungTuKhenThuong hãy nhập trước vào một bản ghi.

```
Private Function Tao_ma_tu_dong() As Integer
Dim ma As Integer
Dim strNNC1 As String
Dim rsNNC1 As ADODB.Recordset
strNNC1 = "select SoChungTuKT from ChungTuKhenThuong"
Set rsNNC1 = New ADODB.Recordset
rsNNC1.Open strNNC1, Cn, adOpenDynamic, adLockOptimistic
rsNNC1.MoveFirst
ma = rsNNC1!SoChungTuKT
Do Until rsNNC1.EOF
If rsNNC1!SoChungTuKT > ma Then
ma = rsNNC1!SoChungTuKT
End If
rsNNC1.MoveNext
Loop
ma = ma + 1
```

```
Tao_ma_tu_dong = ma
```

```
rsNNC1.Close
```

```
End Function
```

‘ Thủ tục dưới đây cho phép hiện mã nhân viên vào Combo box CboMaNhanVien. Chú ý do biến sqlKONA và rsKONA đã được khai báo ở Sub main() nên ta không cần khai báo lại chúng trong Form nữa.

```
Private Sub HienMaNhanVien()
```

```
sqlKONA = "Select MaNhanVien from NhanVien "
```

```
Set rsKONA = New ADODB.Recordset
```

```
rsKONA.Open sqlKONA, Cn, adOpenDynamic, adLockOptimistic
```

```
cboMaNhanVien.Clear
```

```
Do Until rsKONA.EOF
```

```
cboMaNhanVien.AddItem rsKONA!manhanvien
```

```
rsKONA.MoveNext
```

```
Loop
```

```
rsKONA.Close
```

```
End Sub
```

‘ Thủ tục dưới đây cho phép hiện mã khen thưởng vào Combo box CboMaKhenThuong.

```
Private Sub HienMaKhenThuong()
```

```
sqlKONA = "Select MaKhenThuong from KhenThuong "
```

```
Set rsKONA = New ADODB.Recordset
```

```
rsKONA.Open sqlKONA, Cn, adOpenDynamic, adLockOptimistic
```

```
CboMaKhenThuong.Clear
```

```
Do Until rsKONA.EOF
```

```
CboMaKhenThuong.AddItem rsKONA!MaKhenThuong
```

```
rsKONA.MoveNext
```

```
Loop
```

```
rsKONA.Close
```

```
End Sub
```

' Thủ tục dưới đây cho phép hiện tự động họ tên, phòng ban, chức vụ của nhân viên khi người dùng ấn chọn mã nhân viên.

```
Private Sub cboMaNhanVien_Click()
```

```
sqlKONA = "SELECT NhanVien.HoDem +' '+ NhanVien.Ten AS HoVaTen, GioiTinh, ChucVu.TenChucVu, PhongBan.TenPhongBan" & _  
" FROM PhongBan INNER JOIN (ChucVu INNER JOIN NhanVien ON  
ChucVu.MaChucVu = NhanVien.MaChucVu) ON PhongBan.MaPhongBan  
= NhanVien.MaPhongBan" & _
```

```
" where MaNhanVien ='" & cboMaNhanVien.Text & "'"
```

```
Set rsKONA = New ADODB.Recordset
```

```
rsKONA.Open sqlKONA, Cn, adOpenDynamic, adLockOptimistic
```

```
If rsKONA.EOF = False And rsKONA.BOF = False Then
```

```
    TxtHoVaTen.Text = rsKONA!HovaTen
```

```
    TxtChucVu.Text = rsKONA!tenchucvu
```

```
    TxtPhongBan.Text = rsKONA!tenphongban
```

```
Else
```

```
    Exit Sub
```

```
End If
```

```
rsKONA.Close
```

```
End Sub
```

' Thủ tục dưới đây cho phép hiện tự động Nội dung, tiền thưởng khi người dùng ấn vào chọn mã khen thưởng.

```
Private Sub CboMaKhenThuong_Click()
```

```
sqlKONA = "SELECT KhenThuong.NoiDung, KhenThuong.TienThuong  
FROM KhenThuong where MaKhenThuong ='" &  
CboMaKhenThuong.Text & "'"
```

```
Set rsKONA = New ADODB.Recordset
```

```
rsKONA.Open sqlKONA, Cn, adOpenDynamic, adLockOptimistic
If rsKONA.EOF = False And rsKONA.BOF = False Then
    TxtNoiDung.Text = rsKONA!NoiDung
    TxtTienThuong.Text = rsKONA!TienThuong
Else
    Exit Sub
End If
rsKONA.Close
End Sub
```

```
Private Sub Form_Load()
    HienMaNhanVien ' Gọi thủ tục Hiện mã nhân viên
    HienMaKhenThuong ' Gọi thủ tục Hiện mã khen thưởng.
    AdoCTKhenThuong.ConnectionString = STRSQL
    AdoCTKhenThuong.CommandType = adCmdUnknown
    AdoCTKhenThuong.RecordSource = "Select * from
    ChungTuKhenThuong order by NgayChungTuKT Desc" ' Bang du lieu
End Sub
```

Chú ý:

' Tại đoạn code trên, hàm tạo mã tự động áp dụng với mã chỉ có toàn số. Để tạo mã có cả phần chữ và số tự động tăng thêm, ta viết như đoạn code sau:

```
Private Sub NapTuDong()
Dim str As String
Adodc2.ConnectionString = STRSQL
Adodc2.CommandType = adCmdUnknown
Adodc2.RecordSource = "select * from KhenThuong order by
MaKhenThuong"
Adodc2.Refresh
Adodc2.Recordset.MoveLast
```

```

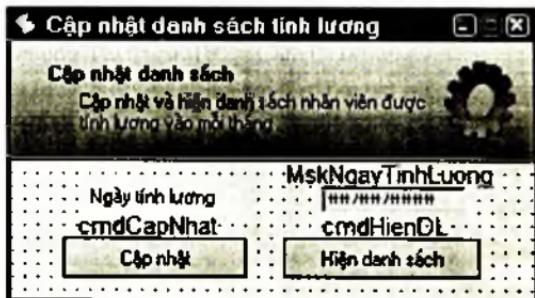
str = Adodc2.Recordset.Fields("MaKhenThuong").Value
Dim HD As Integer
HD = CInt(Right(str, Len(str) - 2)) + 1
If HD < 10 Then
    TxtMaKhenThuong.Text = "KT00" + CStr(HD)
End If
If HD > 10 And HD < 100 Then
    TxtMaKhenThuong.Text = "KT0" + CStr(HD)
End If
If HD > 100 Then
    TxtMaKhenThuong.Text = "KT" + CStr(HD)
End If
End Sub

```

Mã khen thưởng có dạng KT001, khi tự động tăng mã sẽ là KT002 ...

3.10. Lập trình Form Cập nhật danh sách tinh lương

Ở phần dưới đây, bạn sẽ học cách lập trình để chuyển dữ liệu từ một bảng này sang một bảng khác trong cơ sở dữ liệu.



Form Cập nhật danh sách tinh lương được dùng để cập nhật số lượng nhân viên được tính lương vào tháng tinh lương. Danh sách tinh lương được lấy từ bảng NhanVien, và nó được cập nhật vào PhieuLuong để làm cơ sở tinh lương hàng tháng. Mỗi đầu tháng sẽ tiến hành cập nhật danh sách tinh lương vì danh sách nhân viên vào mỗi tháng là khác nhau.

Ở Form này có sử dụng một điều khiển mới là MaskEdbox. Đây là điều khiển giúp cho việc điền ngày tháng theo định dạng mà người lập trình mong muốn. Để sử dụng điều khiển này ta thêm msmsk32.ocx vào dự án. Ta add ocx theo đường dẫn sau:

Project \ Components \ Microsoft Masked Edit Control 6.0

Một điều khiển mới được thêm vào hộp Toolboxs. Ta chọn vẽ control MaskEdbox vào Form đang thiết kế, đặt tên là MskNgayTinhLuong. Tại phần Properties của MskNgayTinhLuong, chọn Mask là #/#/#/#/#/# để nhập dữ liệu theo mẫu định dạng ngày tháng Việt Nam

Code

```
Dim sql1 As String
Dim sql2 As String
Dim rs1 As ADODB.Recordset
Dim rs2 As ADODB.Recordset

Private Sub CmdCapNhat_Click()
If IsDate(MskNgayTinhLuong.Text) = False Then
    MsgBox "Định dạng ngày sai. Hãy nhập lại", vbCritical, "KONA HMS"
    Exit Sub
End If

'---- Kiểm tra nhập khác tháng tính lương theo ngày hiện tại
If Month(Date) <> Month(MskNgayTinhLuong.Text) Or Year(Date) <>
Year(MskNgayTinhLuong.Text) Then
    MsgBox "Bây giờ là tháng " & Right(Date, 7) & ". Bạn lại cập nhật
danh sách cho tháng " & Right(MskNgayTinhLuong.Text, 7) & ". Hãy kiểm
tra lại", vbInformation, "KONA HMS"
    Exit Sub
End If

'---- Kiểm tra trùng ngày tháng tính lương
sql1 = "select NgayTinhLuong from PhieuLuong where
month(NgayTinhLuong) = " & Mid(MskNgayTinhLuong.Text, 4, 2) & "
```

```
and year(ngaytinhluong) = " & Right(MskNgayTinhLuong.Text, 4)
Set rs1 = New ADODB.Recordset
rs1.Open sql1, Cn, adOpenDynamic, adLockOptimistic
If rs1.BOF = False And rs1.EOF = False Then ' Kiểm tra nếu nhập trùng
ngày tháng
MsgBox "Đã cập nhật danh sách nhân viên để tính lương cho tháng này.
Bạn không cập nhật lại nữa.", vbCritical, "KONA HMS"
Exit Sub
End If
rs1.Close

'-----Cập nhật danh sách nhân viên để tính lương hàng tháng
sql1 = "Select * from NhanVien"      ' Mở bảng NhanVien
Set rs1 = New ADODB.Recordset
rs1.Open sql1, Cn, adOpenDynamic, adLockOptimistic
sql2 = "select * from PhieuLuong"      ' Mở bảng PhieuLuong
Set rs2 = New ADODB.Recordset
rs2.Open sql2, Cn, adOpenDynamic, adLockOptimistic
If IsDate(MskNgayTinhLuong.Text) = True Then
    rs1.MoveFirst ' Nhảy đến bản ghi đầu tiên của bảng NhanVien
    Do Until rs1.EOF ' Thực hiện cho tới bản ghi cuối cùng của bảng
NhanVien
        rs2.AddNew ' Thêm mới vào bảng PhieuLuong
        rs2!MaNhanVien = rs1!MaNhanVien
        rs2!HoDemL = rs1!HoDem
        rs2!TenL = rs1!Ten
        rs2!LuongCoBanL = rs1!LuongCoBan
        rs2!MaChucVuL = rs1!MaChucVu
        rs2!MaPhongban = rs1!MaPhongBan
        rs2!NgayTinhLuong = MskNgayTinhLuong.Text
```

```
    rs2.Update ' Cập nhật vào bảng PhieuLuong
    rs1.MoveNext ' Nhảy đến bản ghi tiếp theo trong bảng NhanVien
    rs2.MoveNext ' Nhảy đến bản ghi tiếp theo trong bảng PhieuLuong
    Loop
Else
    MsgBox "Bạn nhập sai định dạng ngày. Hãy nhập lại", vbInformation, "KONA HMS"
    Exit Sub
End If
rs1.Close
rs2.Close
MsgBox "Đã cập nhật xong danh sách nhân viên để tính lương cho tháng này.", vbInformation, "KONA HMS"
End Sub
```

```
Private Sub CmdHienDL_Click()
FrmVSFlex8_2.Show ' Hiện danh sách nhân viên được tính lương vừa cập nhật
End Sub

Private Sub Form_Load()
MskNgayTinhLuong.Text = Date ' Chọn ngày tính lương bằng ngày hiện tại
End Sub
```

3.11. Lập trình Form T嚩ng hợp vật tư nhập kho

Ở phần dưới đây, bạn sẽ học cách lập trình để lấy dữ liệu từ bảng tính Excel chuyển vào Access. Bạn cũng học cách đóng bộ hoá dữ liệu giữa hai bảng trong cơ sở dữ liệu bằng câu lệnh SQL, sử dụng điều khiển CommonDialog để Open và Save các định dạng File khác nhau.

Tổng hợp vật tư nhập kho

Tổng hợp vật tư nhập kho

Cập nhật dữ liệu

Đường dẫn:	<input name="txtDuongDan" type="text" value=""/>
Chọn Sheet:	<input name="btnChonSheet" type="button" value="..."/> Chọn sheet của File Excel <input name="cboSheet" type="text" value=""/>
Bắt đầu từ dòng:	<input name="txtDong" type="text" value="9"/>
Nhập mã vật tư	Bắt đầu từ cột: <input name="txtCol1" type="text" value="B"/>
Nhập số lượng	Bắt đầu từ cột: <input name="txtCol2" type="text" value="E"/>
Ngày nhập liệu	<input name="DTPicker1" type="text" value="12/03/2009"/>
Phòng ban	<input name="cboPhongBan" type="text" value="—"/>

Form Tổng hợp vật tư nhập kho được dùng để tổng hợp vật tư hàng tháng cho các tờ sản xuất có tính lương sản phẩm. Cách tính lương sản phẩm đã được nêu kỹ tại phần phân tích nghiệp vụ. Sau đây, ta chỉ quan tâm đến khía cạnh lập trình cho Form này.

Mục đích của Form này là cập nhật số lượng vật tư và mã vật tư vào bảng Nh?p Vat Tu trong CSDL từ bảng tính Excel có 4 thông tin là mã vật tư, tên vật tư, đơn vị tính, số lượng. Bảng tính Excel này được lấy từ phân hệ kế toán hàng tồn kho của phần mềm Fast Accounting được sử dụng tại phòng kế toán của công ty. (Kế toán hàng tồn kho \ Báo cáo hàng nhập \ Tổng hợp hàng nhập kho).

A	B	C	D	E
2	CÔNG TY TNHH KONA			
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				
39				
40				
41				
42				
43				
44				
45				
46				
47				
48				
49				
50				
51				
52				
53				
54				
55				
56				
57				
58				
59				
60				
61				
62				
63				
64				
65				
66				
67				
68				
69				
70				
71				
72				
73				
74				
75				
76				
77				
78				
79				
80				
81				
82				
83				
84				
85				
86				
87				
88				
89				
90				
91				
92				
93				
94				
95				
96				
97				
98				
99				
100				
101				
102				
103				
104				
105				
106				
107				
108				
109				
110				
111				
112				
113				
114				
115				
116				
117				
118				
119				
120				
121				
122				
123				
124				
125				
126				
127				
128				
129				
130				
131				
132				
133				
134				
135				
136				
137				
138				
139				
140				
141				
142				
143				
144				
145				
146				
147				
148				
149				
150				
151				
152				
153				
154				
155				
156				
157				
158				
159				
160				
161				
162				
163				
164				
165				
166				
167				
168				
169				
170				
171				
172				
173				
174				
175				
176				
177				
178				
179				
180				
181				
182				
183				
184				
185				
186				
187				
188				
189				
190				
191				
192				
193				
194				
195				
196				
197				
198				
199				
200				
201				
202				
203				
204				
205				
206				
207				
208				
209				
210				
211				
212				
213				
214				
215				
216				
217				
218				
219				
220				
221				
222				
223				
224				
225				
226				
227				
228				
229				
230				
231				
232				
233				
234				
235				
236				
237				
238				
239				
240				
241				
242				
243				
244				
245				
246				
247				
248				
249				
250				
251				
252				
253				
254				
255				
256				
257				
258				
259				
260				
261				
262				
263				
264				
265				
266				
267				
268				
269				
270				
271				
272				
273				
274				
275				
276				
277				
278				
279				
280				
281				
282				
283				
284				
285				
286				
287				
288				
289				
290				
291				
292				
293				
294				
295				
296				
297				
298				
299				
300				
301				
302				
303				
304				
305				
306				
307				
308				
309				
310				
311				
312				
313				
314				
315				
316				
317				
318				
319				
320				
321				
322				
323				
324				
325				
326				
327				
328				
329				
330				
331				
332				
333				
334				
335				
336				
337				
338				
339				
340				
341				
342				
343				
344				
345				
346				
347				
348				
349				
350				
351				
352				
353				
354				
355				
356				
357				
358				
359				
360				
361				
362				
363				
364				
365				
366				
367				
368				
369				
370				
371				
372				
373				
374				
375				
376				
377				
378				
379				
380				
381				
382				
383				
384				
385				
386				
387				
388				
389				
390				
391				
392				
393				
394				
395				
396				
397				
398				
399				
400				
401				
402				
403				
404				
405				
406				
407				
408				
409				
410				
411				
412				
413				
414				
415				
416				
417				
418				
419				
420				
421				
422				
423				
424				
425				
426				
427				
428				
429				
430				
431				
432				
433				
434				
435				
436				
437				</

Như vậy, ta cần phải lấy được dữ liệu từ ô B9 và E9 trả về để cập nhật mã vật tư và số lượng vào bảng NhapVatTu.

Ở Form này có sử dụng một điều khiển mới là CommonDialog. Đây là điều khiển giúp cho ta Open hoặc Save một tệp tin liên quan đến dự án từ nguồn dữ liệu bên ngoài. Để sử dụng điều khiển này ta thêm comdlg32.ocx vào dự án. Ta add ocx theo đường dẫn sau:

Project \ Components \ Microsoft Common Dialog Control 6.0 (sp6)

Một điều khiển mới được thêm vào hộp Toolboxes. Ta chọn vđ control CommonDialog vào Form đang thiết kế, đặt tên là dlgBox.

Ngoài ra, để thực hiện việc trao đổi dữ liệu với Microsoft Excel, ta cần thêm thư viện Microsoft Excel 11.0 Object Library vào dự án. Để thêm thư viện này ta làm như sau: Project \ References \ Microsoft Excel 11.0 Object Library

CODE

Dim file As String

Dim sheet As String

```
Dim sql2 As String
```

```
Dim rs2 As ADODB.Recordset
```

```
Private Sub CboSheet_Click() ' Thủ tục khi ấn vào Combo box Sheet
```

```
sheet = cboSheet.Text
```

End Sub

```

Private Sub CmdDuongDan_Click()
With dlgBox ' Đây là tên của hộp thoại Common Dialog
    .DialogTitle = "Mở tệp Excel"           ' Tên hộp thoại là Mở tệp Excel
    .InitDir = "D:\\"                      ' Ó dia đầu tiên mặc định được mở ra là Ó D
    .Filter = "Tệp tin Excel (*.xls)/*.xls" ' Dạng File được chọn là .xls
    .ShowOpen                         ' Hiện hộp thoại Open File
End With
If dlgBox.FileName = "" Then
    'Người dùng ấn Cancel trên hộp thoại Open
Else
    file = dlgBox.FileName
End If
TxtDuongDan.Text = file
'***** Nhập số Sheet tương ứng của bảng tính Excel vào ô Combo box
sheets
If file <> "" Then ' Nếu đã chọn được File rồi thì thực hiện
    Excel.Workbooks.Open file
    Dim j As Variant
    For Each j In Excel.ActiveWorkbook.Worksheets
        cboSheet.AddItem (j.Name)
    Next
End If
If Text1.Text <> "" Then ' Thực hiện mở khoá các control vì trước khi người
    dùng chọn File Excel cần cập nhật thì ta không cho phép người dùng sử
    dụng các control, nhằm đảm bảo an toàn dữ liệu. (Ta có thể khoá các control
    khi thiết kế hoặc tại thủ tục Form_Load)
    cmdCapNhat.Enabled = True
    cboPhongBan.Locked = False
    cboSheet.Locked = False

```

```

txtDong.Locked = False
txtCot1.Locked = False
txtCot2.Locked = False
End If
End Sub

Private Sub CmdCapNhat_Click()
On Error GoTo Loi:
Dim sqlnvt As String
Do While (cboSheet.Text = "" Or cboSheet.Text = "Chọn Sheet của File
Excel")
    MsgBox "Bạn chưa chọn sheet chứa vật tư trong file Excel",
vbInformation, "KONA HMS"
    cboSheet.SetFocus
    Exit Sub
Loop
Do While CboPhongBan = ""
    MsgBox "Bạn chưa nhập phòng ban", vbInformation, "KONA HMS"
    CboPhongBan.SetFocus
    Exit Sub
Loop
'-----
Dim i As Integer
i = Val(TxtDong.Text)
Set RsKoNa = New ADODB.Recordset
sqlnvt = "Select * from NhaphVatTu"
RsKoNa.Open sqlnvt, Cn, adOpenDynamic, adLockOptimistic

```

' Đoạn code sau sẽ cập nhật dữ liệu từ bảng tính Excel vào CSDL. Chủ ý là
giữa các dòng trong bảng tính Excel thì giá trị của Mã vật tư và Số lượng
không được rỗng.

```
Excel.Workbooks.Open file
With Excel.Workbooks.Application.ActiveWorkbook.Worksheets(sheet)
Do While (.Range(txtCot1.Text & i) <> "")
    RsKoNa.AddNew
    RsKoNa.Fields("MaVatTu").Value = .Range(txtCot1.Text & i)
    RsKoNa.Fields("SoLuongNhap").Value = .Range(txtCot2.Text & i)
    RsKoNa.Fields("NgayNhapLieu").Value = DTPicker1.Value
    RsKoNa.Fields("MaPhongBan").Value =
LayMaPhongBan(CboPhongBan.Text)

    RsKoNa.Update
    i = i + 1
Loop
MsgBox "Cập nhật số lượng vật tư thành công vào cơ sở dữ liệu",
vbInformation, "KONA HMS"
End With

Loi:
If err.Number <> 0 Then
    MsgBox "Bạn đã nhập vật tư cho phòng ban này rồi. Không nhập lại nữa",
vbCritical, "KONA HMS"
    MsgBox "Lỗi là " & err.Description, vbInformation, "KONA HMS", 2
Exit Sub
End If
End Sub

Private Sub CmdHienDuLieu_Click() 'Hiện dữ liệu về Vật tư được nhập
thành công vào CSDL
    FrmVSFlex8.Show
End Sub

Private Sub CmdSuaDoiMaVT_Click() ' Hiện Form cho phép sửa đổi mã vật
tư của bảng VatTu trong CSDL.
```

```

FrmVSFlex8_3.Show
End Sub

Private Sub Form_Load()
CboPhongBan.Locked = True
cmdCapNhat.Enabled = False
cboSheet.Locked = True
TxtDong.Locked = True
txtCot1.Locked = True
txtCot2.Locked = True
DTPicker1.Value = Date

'Doan code duoi day cho phép tại Combobox cboPhongBan, dữ liệu hiện ra
có cả mã phòng ban và tên phòng ban.

sql2 = "SELECT MaPhongBan + ' -- ' + TenPhongBan as VField,
MaPhongBan FROM PhongBan order by TenPhongBan"
Set rs2 = New ADODB.Recordset
rs2.Open sql2, Cn, adOpenDynamic, adLockOptimistic
Do While Not rs2.EOF
    CboPhongBan.AddItem rs2!vfield
    rs2.MoveNext
Loop
rs2.Close
End Sub

```

'Đoạn code sau đây là hàm lấy mã phòng ban, vì khi cập nhật vào CSDL, ta chỉ cập nhật mã phòng ban, nên phải cắt được chuỗi chỉ có mã phòng ban từ combobox cboPhongBan.

```

Private Function LayMaPhongBan(Kytu As String) As String
Dim ChuoiTG As String
Dim i As Long
LayMaPhongBan = ""

```

i = 1

Do While Mid(Kytu, i, 1) <> "

ChuoiTG = ChuoiTG & Mid(Kytu, i, 1)

i = i + 1

Loop

LayMaPhongBan = Trim(ChuoiTG)

End Function

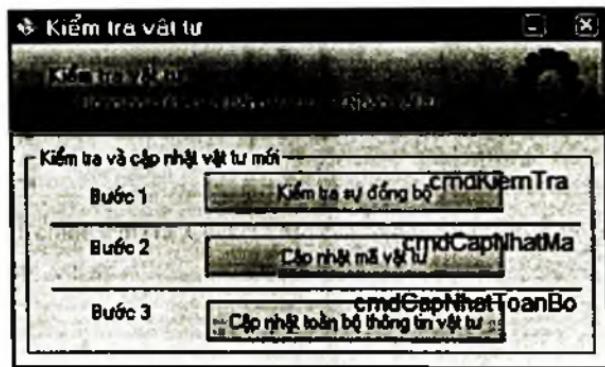
Private Sub cmdKiemTra_Click() ' Mở cửa sổ đóng bộ mã vật tư

FrmKiemTraVT.Show

End Sub

* Phân tích Form FrmKiemTraVT

Đây là Form giúp đóng bộ vật tư tại bảng NhậpVatTu và VatTu trong CSDL.



Do dữ liệu được nhập vào bảng Nhập vật tư hàng tháng có thể có những mã vật tư mới, mà tại bảng VatTu của cơ sở dữ liệu chưa có, do vậy ta phải tiến hành kiểm tra và đóng bộ mã vật tư tại 2 bảng này. Vì bảng VatTu lưu trữ mã vật tư, tên vật tư, đơn vị tính và đơn giá tính lương. Còn bảng NhậpVatTu lưu trữ số mã vật tư, số lượng, ngày nhập liệu, và mã phòng ban. Muốn nhân $SoLuong * DonGia$ thì mã vật tư tại 2 bảng phải được đóng bộ, nếu không chương trình sẽ gặp lỗi.

Trong Form này, bước 1 ta sẽ thực hiện kiểm tra sự đồng bộ, sử dụng câu lệnh sql. Ở bước 2 ta thực hiện cập nhật những mã vật tư chưa trong bảng Vật tư từ bảng Nhập vật tư. Bước 3, ta thực hiện cập nhật những thông tin còn lại của mã vật tư tương ứng cho bảng Vật tư, thông tin này nhập từ ngoài vào, chứ không tự động cập nhật như ở bước 2. Ở bước 3, ta thực hiện cập nhật trên lưới VSFlex Grid, và được lưu thông tin trực tiếp vào cơ sở dữ liệu.

CODE

```
Dim sqlkt As String
Dim rskt As ADODB.Recordset
Private Sub cmdKiemTra_Click()
    sqlkt = " SELECT MaVatTu From NhapVatTu " & _
    " WHERE not exists (Select * from VatTu Where VatTu.MaVatTu = 
NhapVatTu.MaVatTu);"
    Set rskt = New ADODB.Recordset
    rskt.Open sqlkt, Cn, adOpenDynamic, adLockOptimistic
    If rskt.EOF = False Then
        MsgBox " Mã vật tư trong Danh mục Vật tư còn thiếu. Bạn hãy cập nhật 
chúng !", vbInformation, "KONA HMS"
    Else
        MsgBox "Danh mục vật tư đã đầy đủ.", vbInformation, "KONA HMS"
    End If
End Sub

Private Sub cmdCapNhatMa_Click()
    Dim sqlkt1 As String
    Dim rskt1 As ADODB.Recordset
    sqlkt = " SELECT MaVatTu From NhapVatTu " & _
    " WHERE not exists (Select * from VatTu Where VatTu.MaVatTu = 
NhapVatTu.MaVatTu);"
    Set rskt = New ADODB.Recordset
    rskt.Open sqlkt, Cn, adOpenDynamic, adLockOptimistic
```

```

If rskt.EOF = True Then
    MsgBox "Mã vật tư đã đầy đủ. Hãy cập nhật các thông tin khác", vbInformation, "KONA HMS"
    Exit Sub
End If

sqlt1 = "Select MaVatTu from VatTu"
Set rskt1 = New ADODB.Recordset
rskt1.Open sqlt1, Cn, adOpenDynamic, adLockOptimistic
rskt.MoveFirst

' Đoạn code để chuyển những mã chưa có từ bảng Nhập vật tư sang bảng
' Vật tư.

Do Until rskt.EOF
    rskt1.AddNew
    rskt1!MaVatTu = rskt!MaVatTu
    rskt1.Update
    rskt1.MoveNext
    rskt.MoveNext
Loop

MsgBox "Đã cập nhật xong mã vật tư. Hãy chuyển sang bước 3.", vbInformation, "KONA HMS"
End Sub

Private Sub CmdCapNhatToanBo_Click()
    FrmVSFlexCapNhatToanBo.Show
End Sub

```

Tại Form FrmVSFlexCapNhatToanBo ta chỉ cần thiết kế 1 lưới VSFlex Grid với 4 cột là Mã vật tư, Tên vật tư, Đơn vị tính, Đơn giá. Đặt tên control là VSG. Để có thể khi đánh dữ liệu lên lưới, dữ liệu được lưu thẳng vào bảng Vật tư trong CSDL, ta cần chọn chế độ sau cho lưới có tên VSG. Trong Properties Page của Vsg, ta chọn các chế độ sau:

- Tại mục DataMode chọn là 1 - flexDBBound
- Tại mục Editable chọn là 1 - flexEDKbd

CODE

```
Private Sub Form_Load()
With Me.VSg
    Dim r
    For r = 1 To .Rows - 1
        .TextMatrix(r, 0) = r
    Next
    .SelectionMode = flexSelectionListBox
End With

Dim sql1 As String
Dim rs1 As ADODB.Recordset
sql1 = "SELECT MaVatTu, TenVatTu, DonViTinh, DonGia from VatTu" &
-
"Where isnull(donggia) = true or isnull(tenVattu) = true or isnull(DonViTinh) = true "
Set rs1 = New ADODB.Recordset
rs1.Open sql1, Cn, adOpenDynamic, adLockOptimistic
Set VSg.DataSource = rs1 'Thiết lập nguồn dữ liệu cho lưới
End Sub
```

3.12. Lập trình Form Cập nhật ngày công và tính lương

Ở phần này, bạn áp dụng các nội dung đã học để có thể update trực tiếp các số liệu thông qua lưới vào CSDL. Do có rất nhiều số liệu cần nhập nên việc sử dụng lưới nhập VSFlex Grid là nhanh và hiệu quả nhất. Ngoài ra, bạn cũng thực hành tính toán để đưa ra kết quả về lương, lưu dữ liệu tính toán vào CSDL để tiện cho việc báo cáo. Do Form này không có vấn đề mới nên chúng tôi khuyến khích bạn tự viết.

3.13. Lập trình Form Báo cáo thăm niêm công tác và sinh nhật

Ở phần dưới đây, bạn sẽ học cách tạo lập báo cáo cho chương trình. Đây là một phần quan trọng của chương trình vì kết quả của chương trình chính là các báo cáo đầu ra. Ta sẽ học cách lập trình báo cáo với công cụ Crystal

Report. Crystal Report là công cụ tạo lập báo cáo chuyên nghiệp, dễ sử dụng, có khả năng tùy biến cao. Bạn hãy cài đặt Crystal Report 8.5 vào máy tính. Để thực hiện lập trình báo cáo, ta làm theo những bước sau: xây dựng bản mẫu báo cáo (tệp .tx), thiết kế format của báo cáo (tệp .rpt), và truyền dữ liệu vào báo cáo.

3.13.1. Xây dựng bản mẫu báo cáo

- Khởi đầu chọn Crystal Report \ Menu File \ New. Chọn Blank Report

- Trong cửa sổ Data Explorer\ More Data Sources \ Active Data \ Active Data (Field Definition Only)\ Select Data Source. Nhấn Browse để chọn bản mẫu dữ liệu được thiết kế trước hoặc nút New để tạo mới một bản mẫu dữ liệu

- Trong cửa sổ thiết kế Data Definition Tools lần lượt khai báo thông tin các trường về File Name, Field Type, Sample Data

- Sau khi thiết kế xong bản mẫu dữ liệu, ghi nó dưới dạng một tệp (có phần mở rộng là .tx) trong thư mục Report của dự án

- Bấm Add để thêm bản mẫu dữ liệu vào báo cáo mới. Bấm Close để đóng cửa sổ dữ liệu này lại.

3.13.2. Thiết kế bản mẫu báo cáo

- Mẫu báo cáo gồm: Report Header, Detail, Footer

- Thêm các đối tượng báo cáo:

Insert line để thêm đường kẻ vào báo cáo

Insert box để kẻ khung trong báo cáo

Insert Picture để thêm hình ảnh vào báo cáo

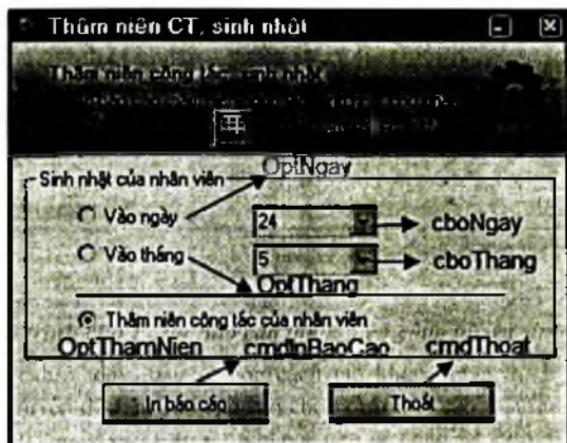
Insert Text Object để thêm các dòng văn bản vào báo cáo

- Để thêm các trường vào mục chi tiết của báo cáo chọn Insert\ Field Object hoặc Insert File

- Lưu báo cáo vào thư mục Report trong dự án với phần mở rộng Crystal Report. Ví dụ báo cáo có tên rptHSSV.rpt

3.13.3. Lập trình cho Form Báo cáo thăm niêm công tác và sinh nhật

Ở phần dưới đây, bạn sẽ học cách tạo khuôn mẫu dữ liệu, thiết kế report, và truyền dữ liệu vào báo cáo cho Form Báo cáo thăm niêm công tác và sinh nhật.

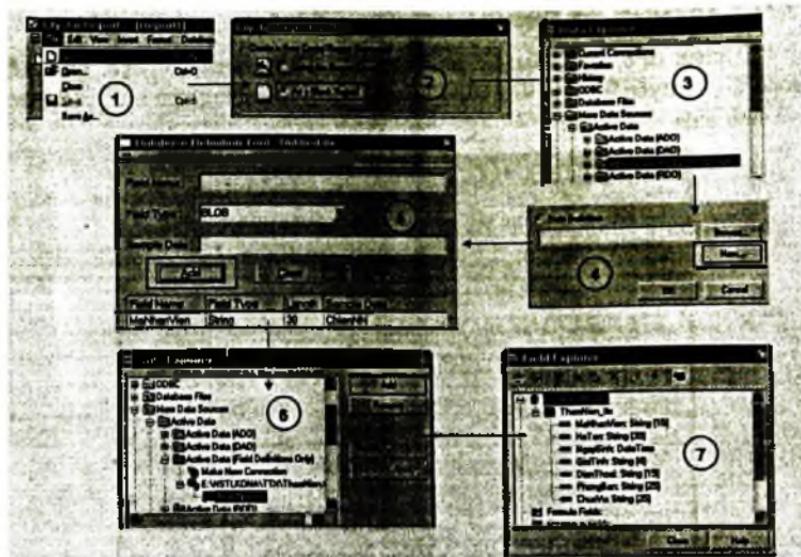


Ở Form báo cáo Tham niêm công tác và sinh nhật, mục đích là in ra danh sách những nhân viên có sinh nhật vào một ngày, một tháng được chọn, hoặc in ra tham niêm công tác theo nhân viên.

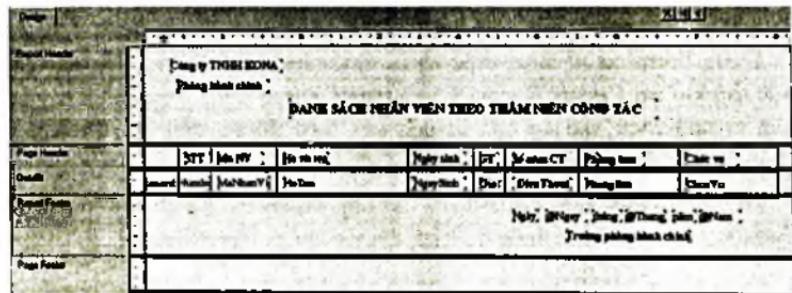
Trong Form có sử dụng điều khiển Crystal Report. Điều khiển này có được sau khi cài Crystal Report 8.5 vào máy. Để sử dụng điều khiển này ta thêm crystl32.ocx vào dự án. Ta add ocx theo đường dẫn sau: Project \ Components \ Crystal Report Control.

Một điều khiển mới được thêm vào hộp Toolboxes. Ta chọn vẽ control CrystalReport vào Form đang thiết kế, đặt tên là CryThamNienSN.

Để tạo khuôn mẫu báo cáo, ta thiết kế theo các bước như sau



Sau đó ta thiết kế báo cáo như sau, đặt tên là ThamNienCT.rpt. Chú ý, từ ThamNienCT.rpt ta có thể copy và đổi tên thành SinhNhatNgay.rpt, SinhNhatThang.rpt rồi viết lại tựa đề cho phù hợp với nội dung báo cáo.



Để hiện ngày tháng năm theo ngày hiện thời, ta tạo 3 Formular Field để hiện dữ liệu trên đó bằng cách nhấp chuột phải vào Formular Field trên cửa sổ Field Explorer, chọn New, sau đó đặt tên tương ứng là Ngay, Thang, Năm.

Tại cửa sổ Formular Editor, với Formular Ngày ta đánh day(currentdate), với Formular Tháng ta đánh month(currentdate), với Formular Năm ta đánh year(currentdate), sau đó save lại. Kéo thả các Formular này vào báo cáo như hình vẽ trên.

PHỤ LỤC 2

Xu hướng phát triển ứng dụng trong quản lý tại các doanh nghiệp ở Việt Nam hiện nay

Sự phát triển của công nghệ thông tin nói chung trên thế giới và Việt Nam đã kéo theo sự phát triển và thay đổi của các hệ thống thông tin ứng dụng trong quản lý doanh nghiệp. Ở các doanh nghiệp Việt Nam hiện nay, hệ thống thông tin quản lý được xây dựng theo hướng tổng hợp và toàn diện (ERP) trong đó bao gồm nhiều phân hệ chức năng được quản lý thống nhất cùng nhau chia sẻ các nguồn tài nguyên của hệ thống và chia sẻ dữ liệu lẫn nhau.

Trên thế giới, có nhiều hệ thống quản trị doanh nghiệp toàn diện(ERP) rất nổi tiếng ví dụ như: SAP Business One, Oracle E-Business Suite, hay Microsoft Dynamics. Các ứng dụng này gồm nhiều module nhỏ cấu thành để phục vụ các nghiệp vụ kế toán, tài chính, nhân sự, khách hàng,... của doanh nghiệp:

- *CRM (Quản lý khách hàng)*.
- *Financials (Quản lý kế toán – tài chính)*
- *HRMS (Quản lý nhân sự)*
- *Order Management (Quản lý đơn hàng)*
- *Asset Management (Quản lý tài sản)*
- *Transportation Management (Quản lý vận chuyển hàng)*
- *Warehouse Management (Quản lý kho hàng)*
- ...

Mặc dù rất nổi tiếng nhưng việc ứng dụng các hệ thống trên vào thực tế Việt Nam lại không được phù hợp bởi:

- Các phần mềm ERP hầu như là đóng gói dựa trên các quy trình quản lý doanh nghiệp chuẩn của các nước phương Tây. Trong khi đó yêu cầu quản lý ở các doanh nghiệp Việt Nam hiện nay đòi hỏi phải có sự chỉnh sửa cho phù hợp với thực tế của doanh nghiệp.

- Nhân lực tham gia vào quá trình vận hành ERP của doanh nghiệp chưa đồng bộ.
- Năng lực tư vấn và triển khai ERP của các đơn vị đã triển khai ERP còn hạn chế.
- Thêm nữa một nguyên nhân không thể không nhắc đến đó là giá thành của các hệ thống này rất đắt, vượt quá khả năng của nhiều doanh nghiệp.

Xu thế chung của các doanh nghiệp Việt Nam hiện nay đó là lựa chọn một giải pháp ERP được phát triển bởi các công ty phần mềm trong nước để tận dụng được những ưu điểm về sự phù hợp, chất lượng dịch vụ và giá thành hợp lý.

Một trong những hệ thống quản trị doanh nghiệp tiêu biểu, có rất nhiều ưu điểm của Việt Nam có thể kể đến *Phần mềm kế toán quản trị Meliasoft 2010*. Đây là sản phẩm mới ra đời nhưng được đánh giá là sản phẩm đã kế thừa được rất nhiều ý tưởng độc đáo từ các sản phẩm trước, đồng thời Công ty Meliasoft (www.meliasoft.com) cũng đã đưa vào nhiều tính năng mới, có tính đột phá về mặt hữu dụng.



Theo thông tin từ Meliasoft, việc phát triển Phần mềm *Meliasoft 2010* là hướng đến *sự phù hợp* cho doanh nghiệp, nó sẽ giải quyết những bất cập và cứng nhắc mà doanh nghiệp đang gặp phải khi sử dụng các phần mềm ERP của nước ngoài hay các phần mềm quản lý của các nhà cung cấp trong nước khác.

Giải pháp Công ty Phần mềm Meliasoft đưa ra là nhằm giảm thiểu chi phí về tài chính, nhân sự và thời gian cho doanh nghiệp, tăng độ chính xác, tính ổn định và tăng khả năng tự duy quản trị logic cho lãnh đạo doanh nghiệp.

Những phân hệ chính của Phần mềm Meliasoft 2010

- *Quản lý tài chính kế toán*
- *Quản lý mua hàng*
- *Quản lý bán hàng*
- *Quản lý hàng tồn kho*
- *Quản lý sản xuất và tính giá thành sản phẩm*
- *Quản lý hợp đồng*

- *Quản lý bảo hành*
- *Quản lý thiết bị*
- *Quản lý chứng từ gốc*
- *Quản lý chiết khấu, khuyến mãi mua, bán hàng*
- *Quản lý công nợ theo hạn thanh toán*
- *Quản lý hạn mức tín dụng*
- *Quản lý tuổi kho (Lô hàng)*
- *Quản lý vận tải*
- *Quản lý dự án, công trình (Hạng mục), dịch vụ, vụ việc.*

Những tính năng tiêu biểu của MeliaSoft

- Chức năng cho phép nhà quản trị lập Kế hoạch hoạt động sản xuất kinh doanh của doanh nghiệp
- Khả năng truy vết thông tin theo đối tượng quản trị
- Khả năng phân luồng dữ liệu và tích hợp quyền theo từng cấp
- Chức năng tìm kiếm nhanh các danh mục, báo cáo
- Chức năng trích rút thông tin đa chiều theo yêu cầu của nhà quản trị
- Chức năng đồng bộ hóa dữ liệu giữa các chi nhánh của Công ty
- Khả năng xử lý nghiệp vụ thông minh
- Chức năng tự động khôi phục lại dữ liệu khi máy tính gặp sự cố (Auto Recovery):

Trong điều kiện hội nhập kinh tế quốc tế của nền kinh tế đất nước, các doanh nghiệp không ngừng làm mới, hiện đại hóa hệ thống thông tin quản lý của mình. Việc xây dựng một hệ thống thông tin quản trị toàn diện đó chính là xu hướng tất yếu cho các doanh nghiệp việt nam hiện nay.

Tài liệu tham khảo

1. Bài giảng “Lập trình trong quản lý”, khoa Tin học Kinh tế
2. Giáo trình Hệ thống thông tin quản lý_ TS Trương Văn Tú, TS Trần Thị Song Minh_NXB Thống kê 2000
3. Giáo trình Cấu trúc dữ liệu và giải thuật_ PGS.TS Hàn Việt Thuận_ NXB Thống kê 2005
4. Giáo trình Cơ sở dữ liệu 1,2_ ThS Trần Công Uẩn_NXB Thống Kê 2005
5. Lập trình Visual Basic 6.0 Cơ bản _ TS Đặng Quế Vinh _ NXB KHKT 2005
6. Sách điện tử và mã nguồn sưu tầm trên mạng Internet

MỤC LỤC

Lời nói đầu.....	3
Chương 1. HỆ THỐNG THÔNG TIN QUẢN LÝ VỚI TỔ CHỨC DOANH NGHIỆP	5
1.1. Lợi ích và thực trạng của việc tin học hoá công tác quản lý tại các doanh nghiệp Việt Nam.....	5
1.1.1. Lợi ích của việc tin học hoá công tác quản lý.....	5
1.1.2. Các giai đoạn phát triển của ứng dụng tin học trong một tổ chức trên thế giới nói chung và ở Việt Nam nói riêng.....	8
1.1.3. Các lý do xây dựng một HTTT mới	11
1.2. Khái niệm phần mềm và những vấn đề liên quan	12
1.2.1. Khái niệm phần mềm.....	13
1.2.2. Phân loại phần mềm.....	13
1.2.3. Đặc điểm riêng có của phần mềm so với các sản phẩm khác	15
1.2.4. Các tiêu chí để đánh giá 1 phần mềm tốt.....	15
1.3. Các bước tổng quát của quy trình phân tích, thiết kế và xây dựng phần mềm	16
1.4. Các thành viên chính của dự án phát triển hệ thống thông tin	19
1.5. Vai trò phân tích viên hệ thống	20
1.6. Một số nguy cơ đối mặt trong quá trình sản xuất phần mềm.....	21
Câu hỏi ôn tập chương 1.....	23
Chương 2. NGÔN NGỮ LẬP TRÌNH VISUAL BASIC 6.0.....	24
2.1. Giới thiệu chung	24
2.1.1. Giới thiệu Visual Basic.....	24
2.1.2. Giao diện làm việc của VB.....	25
2.1.3. Các thành phần của dự án phần mềm (Project).....	32
2.1.4. Một số thao tác cơ bản trong VB.....	33
2.2. Đối tượng và thuộc tính	43
2.2.1. Khái niệm	43
2.2.2. Một số thuộc tính thường gặp của đối tượng	44
2.2.3. Một số đối tượng thường dùng để thiết kế Form.....	45

2.3. Các cấu trúc điều khiển trong VB.....	52
2.3.1 Biến, hằng và các kiểu dữ liệu trong VisualBasic.....	52
2.3.2 Cấu trúc chọn.....	62
2.3.3 Cấu trúc lặp.....	64
2.4 Các thao tác cơ bản khi lập trình và sửa lỗi.....	66
2.4.1 Các thao tác thông dụng trong cửa sổ Code.....	66
2.4.2 Gỡ lỗi chương trình	68
Câu hỏi ôn tập chương 2.....	71
CHƯƠNG 3. XÂY DỰNG GIAO DIỆN CỦA ỨNG DỤNG TRONG QUẢN LÝ.....	72
3.1 Menu	72
3.1.1 Dùng trình soạn thảo menu để tạo menu.....	72
3.1.2 Viết chương trình điều khiển menu.....	74
3.2 Hộp thoại	75
3.2.1. Thông điệp(Message box).....	76
3.2.2. Hộp nhập(Input box)	77
3.2.3. Hộp thoại thông thường (common dialog)	78
3.3. Thanh công cụ (Toolbar).....	89
3.3.1. Thiết kế thanh công cụ.....	89
3.3.2. Lập trình cho thanh công cụ.....	93
3.4. Thanh trạng thái (Statusbar)	93
Câu hỏi ôn tập chương 3.....	95
Chương 4. LẬP TRÌNH LIÊN KẾT VÀ KHAI THÁC CƠ SỞ DỮ LIỆU.....	96
4.1. Các khái niệm liên quan đến cơ sở dữ liệu	96
4.2. Ngôn ngữ truy vấn có cấu trúc SQL.....	97
4.2.1. Giới thiệu chung.....	97
4.2.2 Vai trò của SQL.....	98
4.2.3 Các câu lệnh SQL thường dùng	99
4.3. Sử dụng điều khiển DATA để liên kết và khai thác cơ sở dữ liệu Access ..	136
4.3.1 Giới thiệu chung.....	136
4.3.2 Sử dụng điều khiển Data để liên kết và khai thác cơ sở dữ liệu Access	137
4.4. Lập trình sử dụng thư viện DAO	152
4.4.1. Tổng quát.....	152
4.4.2. Thủ tục mở Database và mở Recordset.....	153

<i>4.4.3. Lập trình hiển thị nội dung một bản ghi trong Recordset</i>	<i>154</i>
<i>4.4.4. Lập trình thêm, bỏ, tìm kiếm một bản ghi</i>	<i>156</i>
<i>4.4.5. Tìm một record</i>	<i>160</i>
4.5. Sử dụng điều khiển ADO Data để liên kết và khai thác cơ sở dữ liệu Access	164
<i>4.5.1. Khai báo sử dụng ADO</i>	<i>164</i>
<i>4.5.2. Data Form Wizard</i>	<i>170</i>
4.6. Lập trình sử dụng thư viện ADO	178
<i>4.6.1. Khai báo biến liên quan</i>	<i>178</i>
<i>4.6.2. Thiết lập kết nối với CSDL</i>	<i>179</i>
<i>4.6.3. Lập trình thêm dữ liệu vào bảng sử dụng ADO</i>	<i>181</i>
<i>4.6.4. Lập trình hiển thị các bản ghi trong bảng CSDL</i>	<i>182</i>
<i>4.6.5. Cập nhật dữ liệu vào bảng CSDL</i>	<i>183</i>
<i>4.6.6. Xóa dữ liệu trong CSDL</i>	<i>184</i>
<i>4.6.7. Kết hợp ComboBox và Listbox với đối tượng ADO</i>	<i>185</i>
<i>4.6.8. Kết hợp ô lưới DataGrid với ADO</i>	<i>186</i>
4.7. Liên kết và truy xuất CSDL với sử dụng môi trường dữ liệu Data Invironment	187
<i>4.7.1. Khai báo sử dụng môi trường dữ liệu Data Environment</i>	<i>187</i>
<i>4.7.2. Liên kết với CSDL</i>	<i>188</i>
<i>4.7.3. Tạo một giao diện người sử dụng với thiết kế DATAENVIRONMENT</i>	<i>189</i>
Câu hỏi ôn tập chương 4	191
Chương 5. TẠO BÁO CÁO ĐẦU RA CHO ỨNG DỤNG	195
5.1. Giới thiệu chung	195
5.2. Tạo báo cáo với công cụ DataReport	196
<i>5.2.1. Thêm một Data Report vào dự án</i>	<i>196</i>
<i>5.2.2. Thiết kế báo cáo</i>	<i>197</i>
<i>5.2.3. Gọi hiển thị báo cáo từ ứng dụng</i>	<i>198</i>
5.3. Tạo báo cáo với Crystal Report	202
<i>5.3.1. Xây dựng bản mẫu báo cáo</i>	<i>202</i>
<i>5.3.2. Thiết kế bản mẫu báo cáo</i>	<i>205</i>
<i>5.3.3. Truyền dữ liệu và hiển thị báo cáo</i>	<i>207</i>
Câu hỏi ôn tập chương 5	211

Chương 6. BIÊN SOẠN TÀI LIỆU HƯỚNG DẪN SỬ DỤNG.....	212
6.1. Các bước chuẩn bị	212
6.2. Tạo mới một Help Project	213
6.3. Tạo mục lục cho nội dung hướng dẫn.....	216
6.4. Hiển thị mục lục tiếng Việt.....	217
6.5. Biên tập nội dung.....	219
6.6. Tạo chỉ mục cho tài liệu hướng dẫn.....	220
6.7. Sử dụng Bookmark	221
6.8. Biên dịch help project thành tệp .chm	223
6.9. Kết nối tệp Help với ứng dụng	224
Chương 7. TẠO BỘ CÀI ĐẶT CHO ỨNG DỤNG	229
7.1. Tạo bộ cài đặt với công cụ có sẵn Package and Deployment	230
7.1.1. Giới thiệu chung	230
7.1.2. Thực hành tạo bộ cài đặt.....	231
7.2. Tạo bộ cài đặt với công cụ chuyên nghiệp InstallShield	243
7.2.1. Giới thiệu công cụ InstallShield 2008 Premiere	243
7.2.2. Thực hành tạo bộ cài đặt cho ứng dụng viết bằng Visual Basic 6	247
Phụ lục 1	271
Phụ lục 2	367
Tài liệu tham khảo	370

Giáo trình
PHÁT TRIỂN ỨNG DỤNG TRONG QUẢN LÝ

NHÀ XUẤT BẢN ĐẠI HỌC KINH TẾ QUỐC DÂN

Địa chỉ: 207 Đường Giải Phóng, Hà Nội

Website: <http://nxb.neu.edu.vn> - Email: nxb@neu.edu.vn

Điện thoại: (04) 38696407 - 36282486 - 36282483

Fax: (04) 36282485

www.nxb.edu.vn

Chủ trách nhiệm xuất bản:

GS.TS. NGUYỄN THÀNH ĐỘ

Chủ trách nhiệm nội dung:

ThS. TRỊNH HOÀI SƠN

Biên tập kỹ thuật:

NGỌC LAN - TRỊNH QUYÊN

Chế bản vi tính:

QUANG KẾT

Thiết kế bìa:

MAI HOA

Sửa bản in và đọc sách mẫu:

NGỌC LAN - TRỊNH QUYÊN

In 500 cuốn, khổ 16 x 24cm tại Xưởng in Đại học Kinh tế quốc dân
Mã số ĐKXB: 311 - 2010/CXB/02 - 281/DHKQTQD và ISBN: 978-604-909-313-5.
In xong và nộp lưu chiểu quý II năm 2010.

