

Nhập môn hệ quản trị cơ sở dữ liệu

LỜI NÓI ĐẦU

Trong sự phát triển của kinh tế tri thức hiện nay, không thể phủ nhận vai trò của công nghệ thông tin. Đặc biệt lĩnh vực cơ sở dữ liệu đã và đang được nhiều người quan tâm. Rất nhiều hệ quản trị cơ sở dữ liệu đã được ứng dụng như Foxpro, Access, SQL Server, Informix, Oracle,... Nhưng đơn giản và thân thiện hơn cả là hệ quản trị CSDL Access, mặc dù nó không có những công cụ hỗ trợ mạnh như Informix hay Oracle, nhưng nó cũng hỗ trợ tương đối tốt về mặt quản lý dữ liệu vừa và nhỏ, đồng thời nó cũng cung cấp công cụ cho phép chúng ta có thể lập trình được. Chính vì vậy mà nhiều khi ta còn gọi là ngôn ngữ Access. Microsoft Access cung cấp hệ thống chương trình ứng dụng rất mạnh, giúp người dùng mau chóng và dễ dàng tạo lập các trình ứng dụng từ bất kể nguồn dữ liệu nào thông qua Query, Form, Report kết hợp với việc sử dụng một số lệnh Access Basic hay còn gọi là lập trình VBA.

Microsoft Access đã trải qua nhiều phiên bản, từ phiên bản Microsoft Access đầu tiên phát hành vào năm 1992 đến nay đã qua nhiều lần phát hành với trên 130 triệu bản. Vì vậy có thể nói rằng Access là một trong những chương trình cơ sở dữ liệu phổ biến nhất thế giới. Hiện nay một số phiên bản đang được sử dụng rộng rãi là Microsoft Access 2000, Microsoft Access XP, Microsoft Access 2003, Microsoft Access 2005. Trong tài liệu này chúng tôi dùng phiên bản Access 2000 và 2003 để minh họa.

Xuất phát từ thực tế nhu cầu học tập của sinh viên chuyên ngành CNTT nói chung và sinh viên trường ĐH Giao Thông Vận tải nói riêng. Chúng tôi viết cuốn bài giảng bám sát theo đề cương môn học của trường, để phục vụ cho sinh viên năm thứ nhất theo học môn “Nhập môn hệ quản trị CSDL”.

Bài giảng được xây dựng gồm 7 chương:

Chương 1: Giới thiệu các khái niệm cơ bản của Microsoft Access

Chương 2: Trình bày cách tạo bảng và cách sử dụng bảng để tổ chức dữ liệu cho bài toán.

Chương 3: Trình bày cách tạo và sử dụng truy vấn để tổng hợp, sắp xếp và tìm kiếm dữ liệu.

Chương 4: Trình bày cách thiết kế và sử dụng mẫu biểu để tổ chức nhập dữ liệu cho một bảng, truy vấn hoặc các bảng có quan hệ với nhau.

Chương 5: Trình bày cách sử dụng báo biểu để tổ chức in dữ liệu của một bảng, truy vấn.

Chương 6: Giới thiệu về Macro và Menu dùng thiết kế giao diện chương trình.

Chương 7: Trình bày về lập trình trong Access.

Trong quá trình biên soạn bài giảng này, mặc dù chúng tôi đã cố gắng hết sức nhưng không tránh khỏi những sai sót. Chúng tôi rất mong nhận được ý kiến đóng góp của độc giả để cuốn sách hoàn thiện hơn.

Hà Nội, ngày 22 tháng 12 năm 2006

Nhóm tác giả

Th.S Đặng Thị Thu Hiền

Th.S Đỗ Thanh Thủy

CHƯƠNG 1

TỔNG QUAN VỀ MICROSOFT ACCESS

1. GIỚI THIỆU VỀ MICROSOFT ACCESS

Microsoft Access là một thành phần trong bộ Microsoft Office, là hệ quản trị cơ sở dữ liệu trên môi trường Windows, trong đó có sẵn các công cụ hữu hiệu và tiện lợi để dễ dàng xây dựng các chương trình cho một số các bài toán thường gặp trong quản lý, thống kê, kế toán...

Microsoft Access là phương tiện để trao đổi thông tin trên mạng Internet và Intranet, giúp người dùng nhanh chóng tạo và sử dụng các chương trình ứng dụng trên mạng. Access không những dễ dàng liên kết các thông tin liên quan mà còn có thể làm việc với nhiều nguồn dữ liệu khác: dBASE, Paradox, SQL trên nhiều loại máy tính: máy tính văn phòng (PC), máy chủ (Server) máy mini và máy lớn (mainframe). Access đảm nhận hai vai trò: phần mềm trên máy khách (client) hoặc máy chủ và dễ dàng truy cập các chương trình ứng dụng khác như Microsoft Word, Excel, PowerPoint.

Mỗi một tệp chương trình thường có một đuôi quy định, ví dụ các tệp của Microsoft Word có đuôi là .doc, tệp chương trình Pascal có đuôi là .pas ... Một tệp chương trình do Access tạo ra có đuôi là .MDB

Một hệ chương trình do Access tạo ra gọi là một Database (cơ sở dữ liệu). Trong các ngôn ngữ như C, Pascal, Foxpro, .. một hệ chương trình gồm các tệp chương trình và các tệp dữ liệu được tổ chức một cách riêng biệt. Nhưng trong Access toàn bộ chương trình và dữ liệu được chứa trong một tệp duy nhất có đuôi .MDB. Như vậy thuật ngữ hệ chương trình hay cơ sở dữ liệu được hiểu là tổ hợp bao gồm cả chương trình và dữ liệu. Để ngắn gọn nhiều khi ta gọi chương trình thay cho thuật ngữ hệ chương trình. Như vậy từ đây khi nói đến chương trình hay hệ chương trình hay cơ sở dữ liệu thì cũng có nghĩa đó là một hệ phần mềm gồm cả chương trình và dữ liệu do Access tạo ra.

2. LÀM VIỆC VỚI MICROSOFT ACCESS

2.1 Khởi động Access

Để khởi động (chạy) Microsoft Access bạn phải chắc chắn rằng Microsoft Office (trong đó có thành phần Microsoft Access) đã được cài đặt đúng cách trên máy tính.

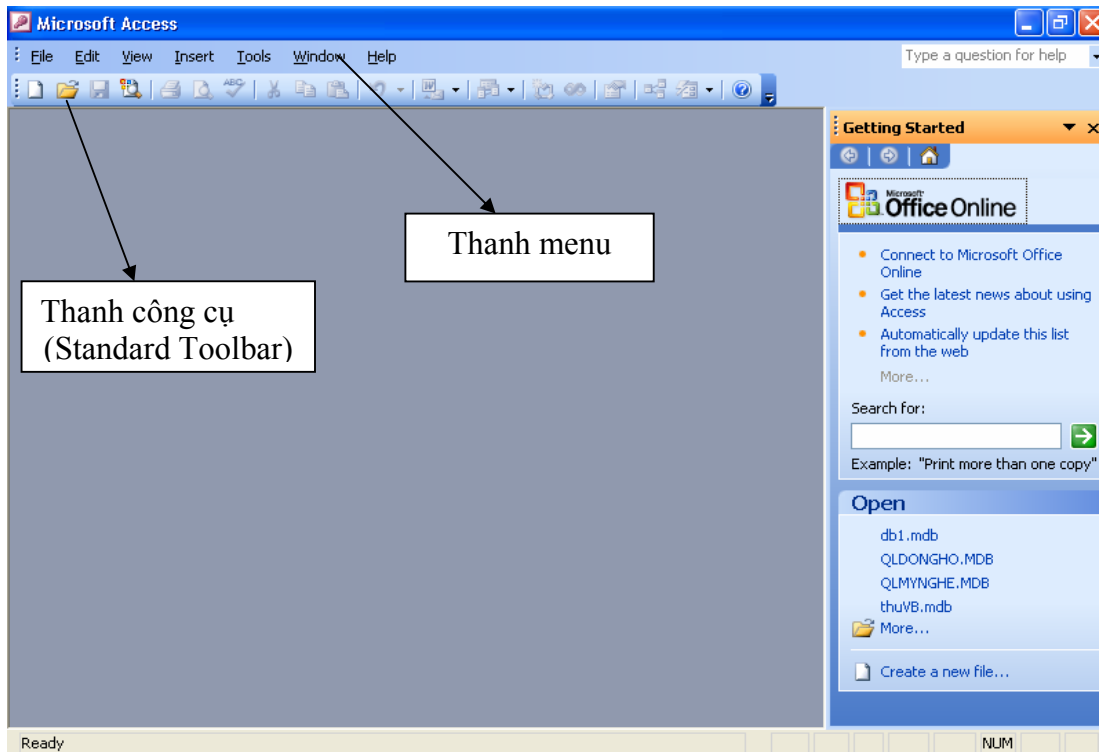
Có nhiều cách để khởi động Microsoft Access phụ thuộc vào máy tính của bạn

Cách 1: Chọn Start, Program, Microsoft Access

Cách 2: Chạy từ icon trên Desktop hoặc TaskBar

Cách 3: Trong trường hợp không tìm thấy shortcut của chương trình trên Desktop hoặc trong menu start ta có thể chạy trực tiếp tệp tin Msaccess trong thư mục đã cài đặt Office thông thường là "C:\Program Files\Microsoft Office\Office". Kết quả nhận được ở hình 1.1: Tại hình này:

- Nếu muốn mở một cơ sở dữ liệu đã có thì chọn trong danh sách ở mục Open.
- Nếu muốn tạo một cơ sở dữ liệu mới thì chọn Create a new file.



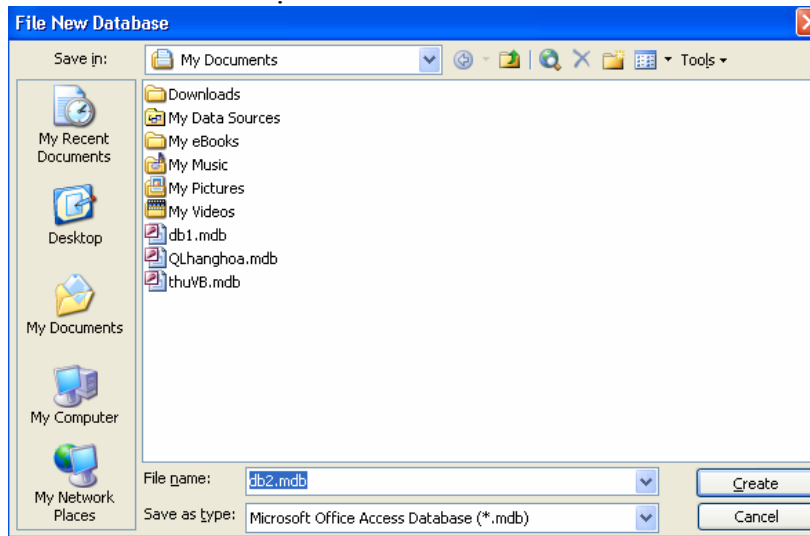
Hình 1.1: Cửa sổ Microsoft Access

2.2 Tạo một cơ sở dữ liệu mới

Trong Access tích hợp tất cả các đối tượng tạo thành file có đuôi là .MDB

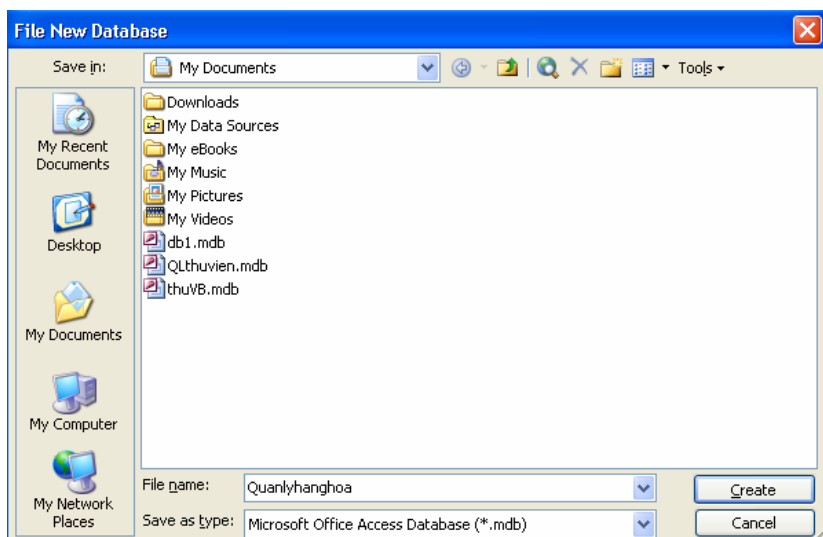
Để tạo 1 cơ sở dữ liệu mới ta làm như sau:

Từ cửa sổ Microsoft Access: chọn menu File, chọn chức năng Create a new file (hoặc kích chuột tại biểu tượng trên thanh công cụ). Tại cửa sổ tiếp theo ta bấm chuột tại biểu tượng Blank Database để hiện ra cửa sổ hình 1.2:



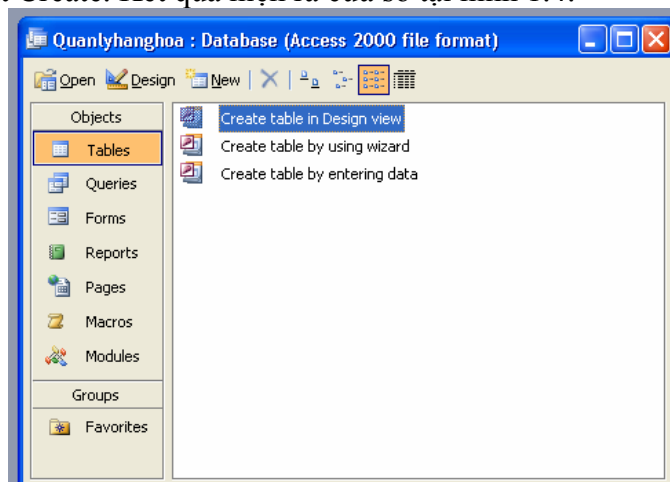
Hình 1.2

Chọn thư mục sẽ chứa tệp cơ sở dữ liệu cần tạo trong hộp Save in (ví dụ thư mục My documents) sau đó đặt tên cơ sở dữ liệu trong hộp File name (chẳng hạn: Quanlyhanghoa). Cửa sổ tương ứng khi đó có dạng hình 1.3



Hình 1.3

Kích chuột tại nút Create. Kết quả hiện ra cửa sổ tại hình 1.4:



Hình 1.4: Cửa sổ chính Database

Cửa sổ trên hình 1.4 được gọi là cửa sổ Database. Đây là một trong những cửa sổ rất quan trọng của Access. Cửa sổ bao gồm:

Hệ menu với các menu: Open, Design, New ...

Tiêu đề Database: **Quanlyhanghoa** là tên của cơ sở dữ liệu

Các đối tượng của một cơ sở dữ liệu trong Access gồm (các mục):

- + **Table**: Đây là các bảng của cơ sở dữ liệu
- + **Query**: là các truy vấn của cơ sở dữ liệu
- + **Form**: các form dùng để thiết kế giao diện chương trình
- + **Report**: Các báo cáo thống kê
- + **Macro**: các Macro dùng để thực thi các hành động trong CSDL.
- + **Module**: chứa các thủ tục được viết bằng Access Basic.

2.3 Làm việc với cơ sở dữ liệu đã tồn tại

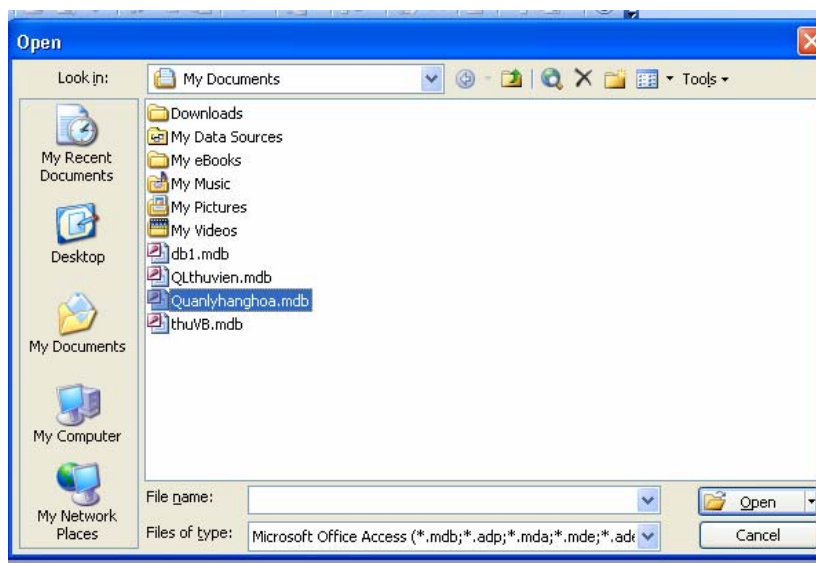
Giả sử đã có cơ sở dữ liệu **Quanlyhanghoa** trong thư mục C:\My Documents. Để làm việc (xem, bổ sung, thực hiện chương trình) với cơ sở dữ liệu trên ta lần lượt thao tác như sau:

Bước1: Từ cửa sổ Microsoft Access chọn menu File, Open Database... hoặc kích chuột tại biểu tượng Open trên thanh Standard Toolbar. Kết quả ở hình 1.5:

Bước 2: Chọn thư mục trong đó có chứa tệp cơ sở dữ liệu cần mở trong hộp Look in

Kết quả trong cửa sổ sẽ hiện ra danh sách các cơ sở dữ liệu trong thư mục này nếu giá trị tại hộp File of type là: Microsoft Access.

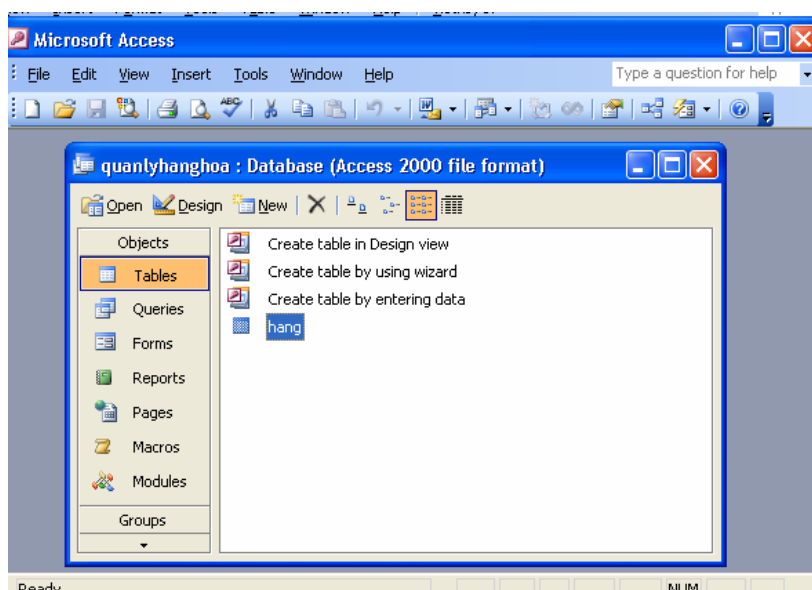
Trong trường hợp có quá nhiều cơ sở dữ liệu đã tồn tại để nhanh chóng tìm được cơ sở dữ liệu cần mở ta chú ý đến một số tùy chọn khác trên cửa sổ như: Last modified, text or property...



Hình 1.5

Bước 3: Chọn tên cơ sở dữ liệu muốn mở tại hộp File name (hoặc kích chuột tại tên của cơ sở dữ liệu cần mở hiện trong cửa sổ), ví dụ Quanlyhanghoa, sau đó kích chuột tại nút Open. Kết quả nhận được trong hình 1.6:

Khi mở xong một cơ sở dữ liệu, nếu muốn chọn mục nào thì ta bấm chuột vào mục đó. Như hình trên ta đang chọn mục Tables, mục này chứa danh sách các bảng của cơ sở dữ liệu **Quanlyhanghoa**. Nhìn vào mục Tables của cửa sổ Database của cơ sở dữ liệu ta thấy mới chỉ có bảng hang (hàng).



Hình 1.6

CHƯƠNG 2

BẢNG VÀ MỐI QUAN HỆ

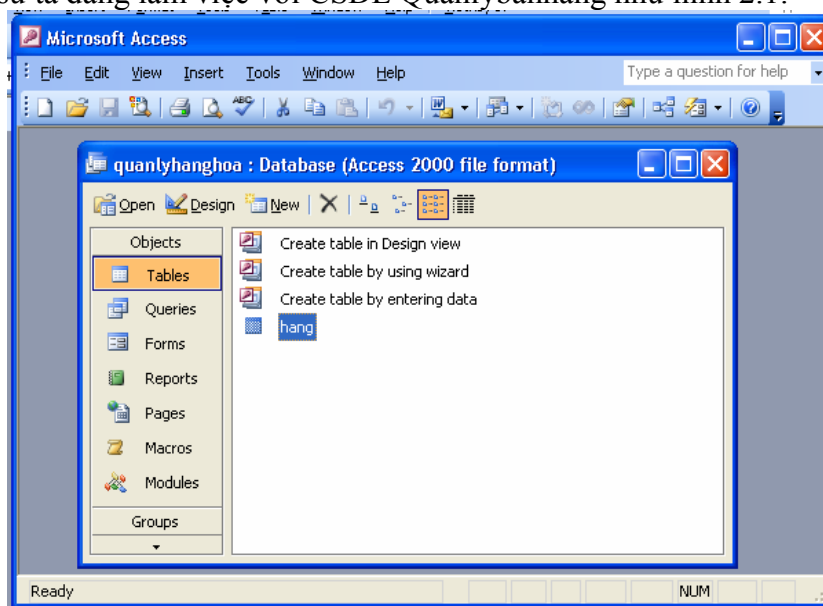
1. CẤU TRÚC CỦA MỘT BẢNG

- Các bảng là khối cấu trúc cơ bản nhất của một cơ sở dữ liệu. Một bảng trong cơ sở dữ liệu phải chứa thông tin thích hợp cho một kiểu cụ thể của thực thể tức là bảng là nơi chứa các thông tin về một chủ đề cụ thể, chẳng hạn như danh sách thư tín. Ta có thể có một hay nhiều bảng trong một cơ sở dữ liệu.
- Các bảng trong cơ sở dữ liệu có cấu trúc giống như các tờ bảng tính. Một bảng gồm nhiều trường/cột (field) và mẫu tin/dòng (record).
- Các dòng (mẫu tin, record) trong bảng tượng trưng cho một dạng mô tả các điển hình của các thực thể được lưu trữ trong bảng. Trật tự các dòng trong bảng không quan trọng, ta có thể sắp xếp chúng theo bất kỳ trật tự nào mà vẫn không thay đổi ý nghĩa thông tin trong bảng.
- Trường là một mục thông tin cụ thể. Các trường có thể có các kiểu dữ liệu khác nhau như: Text, Number, Date/Time,
- Các bảng trong một cơ sở dữ liệu thường có quan hệ với nhau.

2. THIẾT KẾ BẢNG

2.1 Tạo bảng

Giả sử ta đang làm việc với CSDL Quanlybanhang như hình 2.1:



Hình 2.1

Trình tự để tạo một bảng như sau:

a) Từ cửa sổ Database chọn Table

Kết quả sẽ mở cửa sổ Tables, trong đó có chứa danh sách các bảng đã tồn tại và 3 nút lệnh: Open, Design, New

Nút New dùng để tạo bảng mới

Nút Open dùng để mở nhập dữ liệu cho bảng được chọn

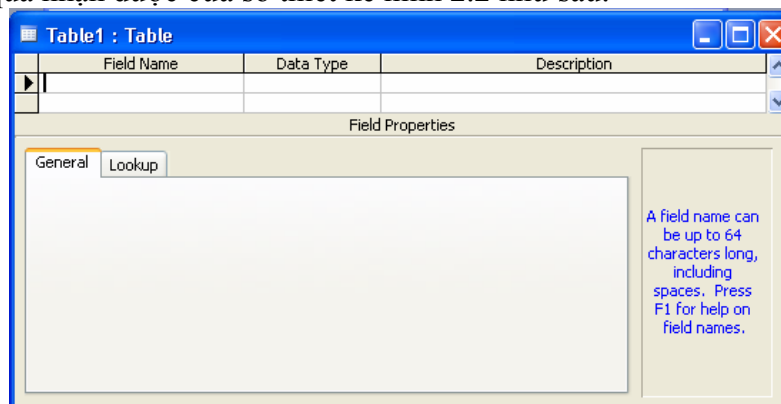
Nút Design dùng để xem, sửa cấu trúc của bảng được chọn

Khi tạo bảng mới ta có thể chọn một trong ba lựa chọn sau:

- + Create table in Table View thì bảng được thiết kế theo ý của người sử dụng. Khi xây dựng bảng mới ta thường sử dụng phương pháp này.
- + Create table by Using Wizard: bảng được xây dựng bằng công cụ Wizards (có sẵn) của Access và ta chỉ việc trả lời các câu hỏi mà Access yêu cầu.
- + Create table by Entering data: bảng được xây dựng theo các cột dữ liệu (thường ít sử dụng cách này).

b) Chọn nút Create table in Design View

Kết quả nhận được của sổ thiết kế hình 2.2 như sau:



Hình 2.2

Tên bảng được Access đặt mặc định là Table1. Sau này ta có thể thay tên bảng bằng một tên khác có ý nghĩa hơn.

Cửa sổ Design Table được chia làm 2 phần:

- + Phần trên gồm 3 cột: Field Name, Data Type, Description dùng để khai báo các trường của bảng, mỗi trường khai báo trên một dòng
- + Phần dưới dùng để quy định các thuộc tính của bảng.

Để di chuyển giữa hai phần ta có thể dùng chuột hoặc bấm phím F6.

Ba cột của phần trên có ý nghĩa như sau:

- Field Name (Tên trường): là một dãy không quá 64 kí tự, bao gồm chữ cái, chữ số, khoảng trống. Nói chung tên các đối tượng trong Access như tên bảng, tên trường, tên mẫu biểu, ... cho phép sử dụng dấu cách. Tuy nhiên tên một trường không cho phép bắt đầu bằng dấu cách
- Data Type (Kiểu dữ liệu): Trong cột Data Type ta có thể chọn 1 kiểu dữ liệu cho trường. Có 10 kiểu dữ liệu sau:

- Text	Ký tự	độ dài ≤ 255 byte
- Memo	Ký tự	độ dài ≤ 255 byte
- Number	Số nguyên, thực	đài: 1,2,3 hoặc 8 byte
- Date /Time	Ngày tháng /giờ	đài 8 byte
- Curreny	Tiền tệ	đài 8 byte
- AutoNumber	Số	đài 8 byte
- Yes/No	Boolean	1 Bit
- OLE Object	Đối tượng nhúng kết hình ảnh, nhị phân	1 Giga Byte
- Hyperlink	Ký tự hoặc kết hợp ký tự và số	
-Lookup Wizard	Cho phép chọn giá trị từ bảng khác	

- Description (Mô tả): Để giải thích cho rõ hơn một trường nào đó. Văn bản mô tả sẽ được hiển thị khi nhập số liệu cho các trường.

c) Khai báo các trường:

- + Đặt tên trường
- + Chọn kiểu dữ liệu
- + Mô tả trường nếu muốn
- + Quy định các thuộc tính cho trường (field properties). Ví dụ: muốn khai báo một trường là kiểu số nguyên, ta phải khai báo trong cột Datatype là Number, nhưng ở phía dưới tại thuộc tính FieldSize để chọn kiểu long Integer. (sẽ được trình bày chi tiết trong các phần sau)

d) Thay đổi thiết kế, chỉnh sửa cấu trúc bảng

Xoá một trường: Chọn trường cần xoá, bấm phím Delete hoặc chọn Delete từ menu Edit, Delete Rows.

Chú ý: Để xoá đồng thời nhiều trường ta có thể dùng các phím Shift và Ctrl cùng với nút trái chuột để chọn các trường cần xoá, rồi bấm phím Delete hoặc chọn Edit / Delete Row.


Để thay đổi nội dung (tên, kiểu, ..) của một trường, ta chỉ việc dùng bàn phím để thực hiện các thay đổi cần thiết.

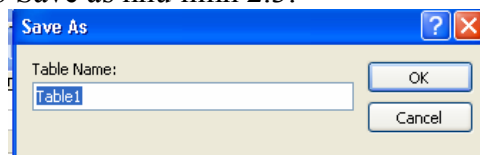
Để chèn thêm một trường mới ta làm như sau: Chọn trường mà trường mới sẽ được chèn vào trước, rồi chọn Edit, Insert Row.

Di chuyển trường: Chọn trường cần di chuyển, rồi kéo trường tới vị trí mới.

e) Lưu cấu trúc bảng

Sau khi hoàn chỉnh việc thiết kế, ta cần ghi cấu trúc bảng và đặt tên cho bảng. Thực hiện như sau:

- Chọn File, Save hoặc bấm chuột tại biểu tượng ghi . Khi bảng được ghi lần đầu thì Access sẽ hiện cửa sổ Save as như hình 2.3:



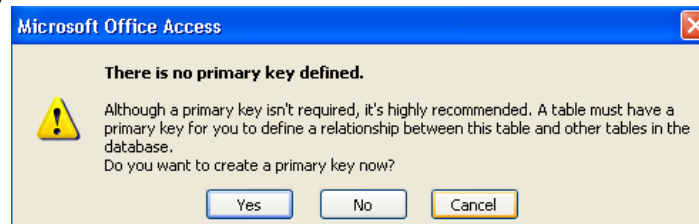
Hình 2.3

Trong hộp Table Name ta đưa vào tên bảng, rồi bấm OK.

Sau khi bấm OK nếu chưa đặt khoá chính thì Access sẽ có thông báo như hình 2.4:


Nếu chọn No thì cấu trúc của bảng được ghi như thiết kế (không có khóa chính)

Nếu chọn Yes thì Access tự động tạo thêm trường có tên ID, kiểu AutoNumber và dùng trường này làm khoá chính.



Hình 2.4

Chú ý:

- + Sau khi ghi xong thì Access trở lại cửa sổ thiết kế bảng
- + Để ra khỏi cửa sổ thiết kế và trở về cửa sổ Database, ta cần đóng cửa sổ thiết kế theo quy tắc của windows (bấm chuột tại biểu tượng  của cửa sổ cần đóng)

+ Khi đóng cửa sổ thiết kế bảng Access sẽ yêu cầu ghi các thay đổi trên thiết kế bảng.

2.2. Đặt khoá chính

+ Khoá chính là một hay nhiều trường xác định duy nhất một bản ghi. Mỗi bảng trong cơ sở dữ liệu nên có một khoá chính.

+ Lợi ích của khoá chính:

Access tự động tạo chỉ mục (Index) trên khoá nhằm tăng tốc độ truy vấn và các thao tác khác.

Khi xem mẫu tin (dạng bảng hay mẫu biểu), các mẫu tin sẽ được trình bày theo thứ tự khoá chính.


Khi nhập số liệu, Access kiểm tra sự trùng nhau trên khoá chính.

Access dùng khoá chính để tạo sự liên kết giữa các bảng.

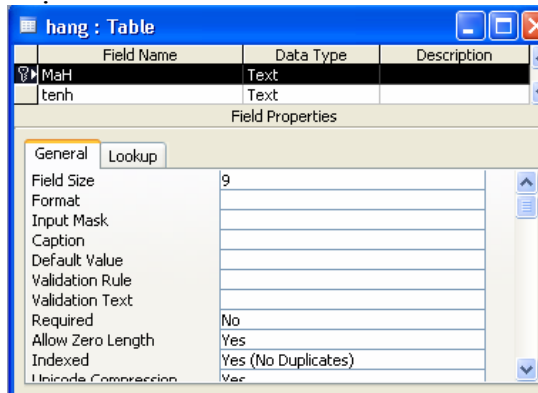
+ Đặt khoá chính

Thực hiện theo trình tự sau:

- Chọn các trường làm khoá chính.


- Chọn Edit, Primary Key hoặc nhấp biểu tượng khoá 

Ví dụ: Trong hình 2.5 chọn MaH là khoá chính.



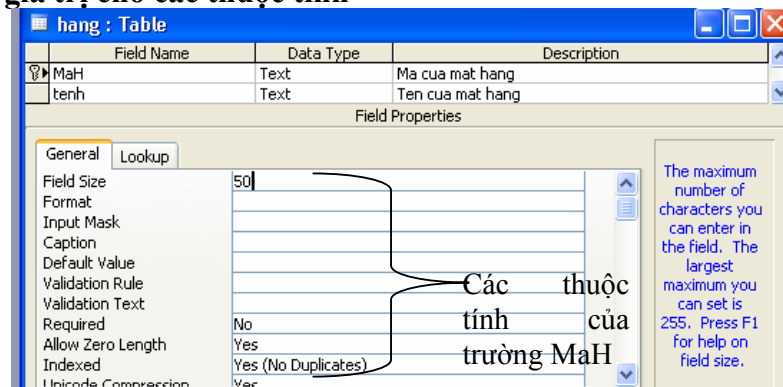
Hình 2.5

+ Đặt lại khoá chính: Để đặt lại khoá chính ta chọn các trường làm khoá rồi thực hiện như trên.

+ Xoá khoá chính: Chọn trường khoá, bấm nút . Hoặc bấm chuột phải tại trường khoá chọn lại Primary Key (nút này đã được chọn):

2.3 Thuộc tính của trường

a) Cách đặt giá trị cho các thuộc tính



Hình 2.6

Trong cửa sổ thiết kế bảng, mỗi khi chọn một trường ở nửa trên thì nửa dưới thể hiện các thuộc tính của trường vừa chọn. Mỗi thuộc tính nằm trên một dòng như hình 2.6.

Lúc đầu mỗi thuộc tính hoặc bỏ trống hoặc có giá trị mặc định ví dụ giá trị mặc định của thuộc tính FieldSize của trường kiểu Text là 50 (giá trị này ta có thể đặt lại bằng cách chọn Tools/Options /Tables/ Queries trong ô Default Field Sizes Chọn Text =giá trị mới)

Giá trị của thuộc tính có thể gõ trực tiếp từ bàn phím hoặc có thể chọn từ một danh sách của combo box.

b) Tổng quan về các thuộc tính

FieldSize: Số ký tự (Độ dài) của trường Text hoặc kiểu của trường Number

Format: Dạng hiển thị dữ liệu kiểu số và ngày

DecimalPlaces: Số chữ số thập phân trong kiểu number và Currency

InputMask: Quy định khuôn dạng nhập liệu (Mặt nạ nhập liệu)

Caption: Đặt nhãn cho trường. Nhãn sẽ được hiển thị khi nhập liệu thay vì tên trường (nhãn mặc định)

Default Value: Xác định giá trị mặc định của trường

Validation Rule: Quy tắc dữ liệu hợp lệ. Dữ liệu phải thoả mãn quy tắc này mới được nhập.

Required: Không chấp nhận chuỗi rỗng. Cần phải nhập một dữ liệu cho trường.

AllowZeroLength: Chấp nhận chuỗi rỗng trong trường Text, Memo

Indexed: Tạo chỉ mục để tăng tốc độ tìm kiếm trên trường này.

2.4 Thuộc tính của bảng

- **Description:** Mô tả những nét chung của bảng

- **Validation Rule:** Access kiểm tra quy tắc (điều kiện) này trước khi cho nhập một mẫu tin vào bảng

- **Validation Text:** Thông báo lỗi khi một bản ghi vi phạm quy tắc.

Mở các thuộc tính của bảng:

Mở bảng trong chế độ Design View.

Từ menu View chọn Properties để hiện các dòng thuộc tính của bảng.

Đưa các điều kiện vào dòng thuộc tính Validation Rule

Đưa các thông tin muốn thông báo khi dữ liệu nhập không thoả điều kiện Validation Rule

2.5 Cập nhật dữ liệu vào bảng

Khi chúng ta muốn thao tác nhập, sửa, xoá dữ liệu vào bảng thì bảng đó phải được mở (hay còn gọi là chế độ DataSheet).

Muốn vào chế độ này ta làm như sau: Trong cửa sổ Database chọn Tab Tables, chọn bảng muốn mở rồi bấm nút Open, hoặc nhấp đúp chuột.

- Để cập nhật dữ liệu vào bảng, ta mở bảng ở chế độ Datasheet

+ Ta có thể xem, sửa, bổ sung các bản ghi mới trong cửa sổ nhập liệu.

+ Để chọn Font chữ, cỡ chữ thích hợp, ta sử dụng chức năng Font của menu Format.

- Để xoá một hàng trong bảng ta đặt chuột tại hàng muốn xoá chọn menu Edit, Select Record, sau đó bấm chuột phải tại hàng đó chọn Delete Record, hoặc bôi đen và bấm phím Delete.

2.6 Xoá và đổi tên bảng

Để xoá một bảng, tại cửa sổ Database chọn tab Tables, chọn bảng muốn xoá. Nhấn phím Delete hoặc chọn menu Edit/ Delete.

Để đổi tên bảng, tại cửa sổ Database chọn tab Tables, chọn bảng muốn đổi tên. Dùng chức năng Edit/Rename để đổi tên bảng. Hoặc nháy chuột phải chọn Rename.

3. THIẾT LẬP QUAN HỆ GIỮA CÁC BẢNG

3.1 Mỗi quan hệ giữa các bảng

Khi tạo các bảng cho một ứng dụng ta nên xem xét (cân nhắc) mối quan hệ giữa chúng. Những mối quan hệ này làm cho một cơ sở dữ liệu quan hệ có nhiều sức mạnh. Có ba loại quan hệ giữa các bảng: quan hệ một - một, quan hệ một - nhiều, quan hệ nhiều - nhiều.

+ **Quan hệ một – một (1-1):** Trong quan hệ một-một, mỗi record(mẫu tin) trong bảng này tương ứng với một record trong bảng thứ hai. Kiểu quan hệ này có các lợi ích như sau: Lợi ích thứ nhất, bạn có thể đặt các field trong 2 bảng vào một bảng, thành bảng kết hợp. Lợi ích thứ 2 là nó có thể làm giảm thời gian cần thiết để mở một bảng lớn bằng cách đặt một số cột của bảng thành bảng thứ 2 riêng biệt. Cuối cùng quan hệ một-một có thể hỗ trợ bảo mật. Access áp dụng việc bảo mật mức người sử dụng ngay tại mức bảng. Bởi vậy, nếu có một tập con của các trường trong bảng cần bảo mật, bạn đặt chúng vào một bảng riêng biệt để ngăn cản ứng dụng truy xuất tới một số field nào đó. ứng dụng của bạn có thể liên kết với bảng bị giới hạn để trở về bảng chính qua mối quan hệ một-một để cho những người được phép mới có thể chỉnh sửa, xoá và thêm mới vào những Field này.

+ **Quan hệ một - nhiều(1-n):** Quan hệ một nhiều là mối quan hệ trong đó một record này tương ứng với một hay nhiều record của bảng kia. Đây là loại quan hệ khá thông dụng. Loại quan hệ này có thể tạo thành bước cơ bản cho mối quan hệ nhiều-nhiều.

+ **Quan hệ nhiều - nhiều(n-n):** Mối quan hệ nhiều nhiều chỉ tồn tại gián tiếp, nó được xây dựng trên mối quan hệ một – nhiều.


+ **Tính toàn vẹn tham chiếu:**

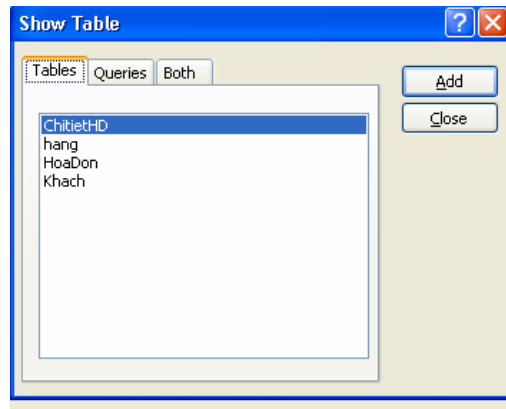
Ta có thể làm cho cơ sở dữ liệu của mình mạnh mẽ hơn bằng cách kết hợp chặt chẽ tính toàn vẹn tham chiếu, xoá và cập nhật theo các mức. Các quy tắc của tính toàn vẹn tham chiếu phải chắc chắn rằng mối quan hệ giữa các bảng là hợp lệ. Các quy tắc này còn ngăn ngừa việc thay đổi tình cờ dữ liệu.

Tính toàn vẹn tham chiếu không cho phép việc thêm dữ liệu ở bên nhiều của mối quan hệ nếu giá trị giá trị khoá không so khớp với bên một trong quan hệ một – nhiều.

Tính toàn vẹn tham chiếu còn giúp để tránh tình trạng các mẫu tin(record) bị “mò côi”. Các mẫu tin trong bảng bên nhiều của mối quan hệ không có mẫu tin nào tương ứng ở bên một. Để khắc phục điều này ta ngăn chặn việc xoá các mẫu tin bên một nếu vẫn còn có một hay nhiều mẫu tin tương ứng ở bên nhiều.

3.2 Cách tạo quan hệ

Bước 1: Kích chuột tại biểu tượng  hoặc chọn Relationship từ menu Tools để hiển thị cửa sổ Relationship cùng với cửa sổ Show Table, xem hình 2.7:

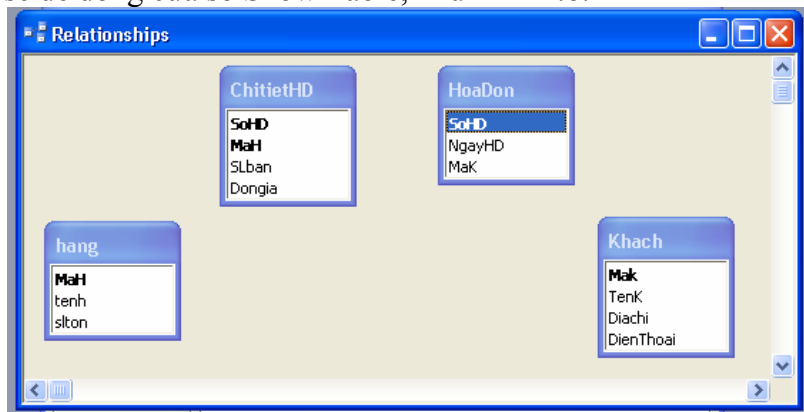


Hình 2.7

Tác dụng của 3 tab trên cửa sổ:

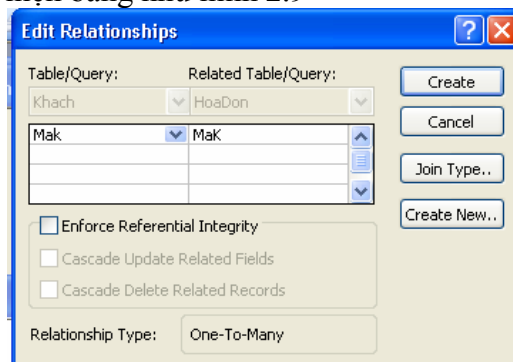
- + Tab Table để hiển thị các bảng có trong cơ sở dữ liệu
- + Tab Queries để hiển thị các truy vấn có trong cơ sở dữ liệu
- + Tab để hiển thị tất cả các bảng, các truy vấn có trong cơ sở dữ liệu

Bước 2: Chọn các bảng và truy vấn muốn tạo đưa vào quan hệ. Sử dụng các phím Ctrl hoặc Shift để chọn nhiều bảng hoặc truy vấn, sau đó bấm nút Add. Sau khi chọn xong bấm nút Close để đóng cửa sổ Show Table, như hình 2.8.



Hình 2.8

Bước 3: Chọn một trường từ bảng chính (Primary table) và kéo sang trường tương ứng của bảng quan hệ, xuất hiện bảng như hình 2.9

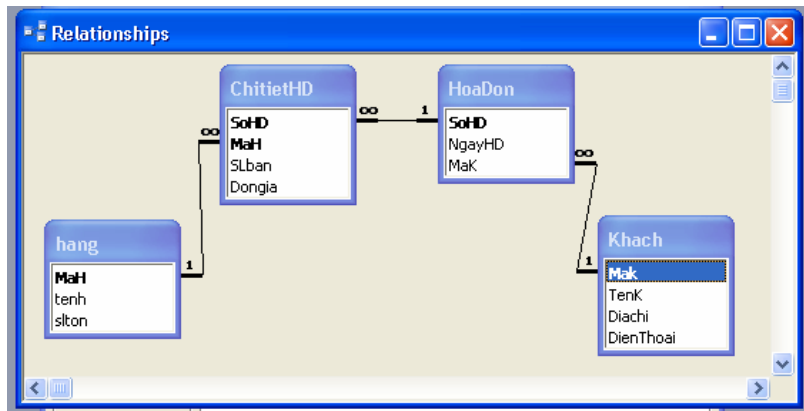


Hình 2.9

Nên chọn mục Enforce Refential Intergrity sau đó chọn tiếp 2 mục phía dưới để đảm bảo toàn vẹn dữ liệu (se nói chi tiết trong phần sau).

Có thể bấm nút Join Type để xác định lại kiểu kết nối.

Cuối cùng bấm chuột tại Create để tạo quan hệ. Khi đó sẽ có đường thẳng nối giữa hai trường biểu diễn quan hệ vừa tạo như hình 2.10:



Hình 2.10

3.3 Chỉnh sửa quan hệ

Xoá quan hệ: bấm chuột tại đường quan hệ rồi bấm phím Delete.

Thay đổi kiểu quan hệ, ta kích chuột tại đường quan hệ bấm chuột phải chọn Edit Relationship sau đó chọn nút **Join Type**, ta được hình 2.10:



Hình 2.11: Các kiểu quan hệ của Access

Lựa chọn 1: Chỉ những bản ghi có giá trị bằng nhau trên trường liên kết của hai bảng mới được liên kết với nhau. Thực tế hay sử dụng kiểu quan hệ này.

Lựa chọn 2: Tất cả những bản ghi của bảng chính (là bảng bắt đầu kéo chuột khi tạo quan hệ) đều được liên kết để tạo thành bản ghi mới. Với những bản ghi có giá trị bằng nhau trên trường liên kết giữa hai bảng sẽ được tạo ra, ngoài ra còn trường hợp những bản ghi của bảng chính không có giá trị của trường liên kết trên bảng quan hệ, khi đó bản ghi mới vẫn được tạo ra với những trường của bảng quan hệ đều có giá trị “trống”.

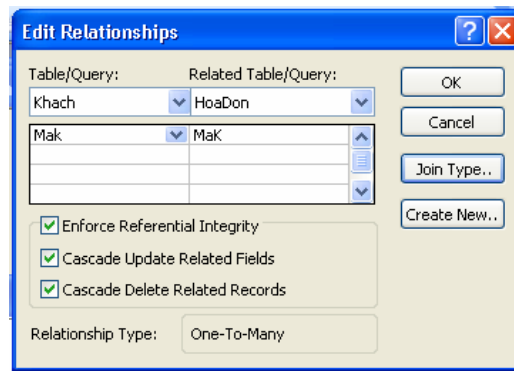
Lựa chọn 3: Tất cả những bản ghi của bảng quan hệ (bảng quan hệ là bảng kéo chuột trở đến sau) đều được liên kết để tạo thành bản ghi mới. Với những bản ghi có giá trị bằng nhau trên trường liên kết giữa hai bảng sẽ được tạo ra, ngoài ra còn trường hợp những bản ghi của bảng quan hệ không có giá trị của trường liên kết trên bảng chính, khi đó bản ghi mới vẫn được tạo ra với những trường của bảng chính đều có giá trị “trống”.

Lựa chọn 1 còn gọi là liên kết nội, lựa chọn 2 gọi là liên kết trái (bảng bên trái được bảo toàn) và lựa chọn 3 là liên kết phải (bảng bên phải được bảo toàn). Liên kết trái và phải được gọi là liên kết ngoại.

3.4 Tạo các toàn vẹn tham chiếu

Để đảm bảo tính toàn vẹn tham chiếu của các bảng chúng ta chọn như sau:

Chọn mục: Enforce Refential Intergrity trong cửa sổ tạo quan hệ (hình 2.12):



Hình 2.12

Nếu thoả mãn các điều kiện:

- + Trường quan hệ của bảng chính là khoá chính
- + Các trường quan hệ có cùng kiểu dữ liệu
- + Cả hai bảng thuộc cùng cơ sở dữ liệu

Thì Access luôn đảm bảo tính chất sau: Mỗi bản ghi trong bảng quan hệ phải có một bản ghi tương ứng trong bảng chính. Điều này ảnh hưởng đến các phép: xoá trên bảng chính và phép thêm trong bảng quan hệ.

Khi đã chọn **Enforce Referential Integrity** thì có thể sử dụng thêm các tùy chọn sau:

- + **Cascade Update Related Fields**: Khi sửa giá trị trường khoá trong bảng chính giá trị tương ứng của các bản ghi trong trường quan hệ sẽ bị sửa theo. Không cho phép thêm vào một bản ghi mới vào bảng quan hệ khi giá trị của trường quan hệ chưa có trong bảng chính. Trong hình 2.12 ta không thể thêm vào bảng HoaDon một bản ghi mà trong đó MaK chưa tồn tại trong bảng Khach.
- + **Cascade Delete Related Records**: Khi xoá một bản ghi trong bảng chính, các bản ghi tương ứng trong bảng quan hệ sẽ bị xoá. Trong hình 2.12 khi muốn xoá một mặt hàng nào đó trong bảng Khach thì các bản ghi tương ứng trong bảng HoaDon cũng bị xoá theo.

BÀI TẬP CHƯƠNG 2

Bài 1: Tạo bảng **Solieu** trong cơ sở dữ liệu **QUANLYDIEM** có cấu trúc như sau:

Tên trường	Kiểu dữ liệu	Mô tả
Socongto	Text	Số công tơ
Chuhoc	Text	Tên chủ hộ
Diachi	Text	Địa chỉ
Sotruoc	Long Interger	Số điện tháng trước
Sosau	Long Interger	Số điện tháng sau

Bài 2: Để quản lý thu chi của một cơ quan ta lập cơ sở dữ liệu **THUQUY**, dữ liệu được lưu trữ trong bảng **Thuchi** có cấu trúc như sau:

Tên trường	Kiểu dữ liệu	Mô tả
SoHD	Text	Số hoá đơn
Loai	yes/No	Loại hoá đơn yes là thu, No là chi
Lydo	Text	Lý do thu, chi
Ngay	Date/Time	Ngày viết HD
Sotien	Single	Số tiền
NguoiTC	Text	Người thu, chi
Ghichu	Text	Ghi chú

Bài 3: Để quản lý thi cử của một trường phổ thông người ta lập cơ sở dữ liệu **THICU** gồm các bảng sau:

Bảng: **Danhsach** có cấu trúc như sau:

Tên trường	Kiểu dữ liệu	Mô tả
<u>SoBD</u>	Text	Số báo danh
Hoten	Text	Họ và tên
Lop	Text	Lớp
Diachi	Text	Địa chỉ
Tongdiem	Single	Tổng điểm 3 môn

Bảng: **SoBD_Phach** có cấu trúc như sau:

Tên trường	Kiểu dữ liệu	Mô tả
SoBD	Text	Số báo danh
Phach1	Text	Phách môn 1
Phach2	Text	Phách môn 2
Phach3	Text	Phách môn 3

Bảng: **Phach_Diem_1** có cấu trúc như sau:

Tên trường	Kiểu dữ liệu	Mô tả
<u>Phach</u>	Text	Phách môn 1
Diem	Single	Điểm môn thứ 1

Bảng: **Phach_Diem_2** có cấu trúc như sau:

Tên trường	Kiểu dữ liệu	Mô tả
<u>Phach</u>	Text	Phách môn 2
Diem	Single	Điểm môn thứ 2

Bảng: **Phach_Diem_3** có cấu trúc như sau:

Tên trường	Kiểu dữ liệu	Mô tả
<u>Phach</u>	Text	Phách môn 3
Diem	Single	Điểm môn thứ 3

Ghi chú: các trường gạch chân là các trường khoá của các bảng.

Hãy tạo quan hệ giữa các bảng trên.

Bài 4: Trong bài toán quản lý công ty người ta xây dựng cơ sở dữ liệu **QUANLYCONGTY** gồm các bảng sau:

Bảng: **Chinhanh** có cấu trúc như sau:

Tên trường	Kiểu dữ liệu	Mô tả
<u>MaCN</u>	Text	mã chi nhánh
TenCN	Text	tên chi nhánh
Diachi	Text	địa chỉ chi nhánh

Bảng: **Duan** có cấu trúc như sau:

Tên trường	Kiểu dữ liệu	Mô tả
<u>MaDuan</u>	Text	mã dự án
TenDuan	Text	tên dự án
MaCN	Text	mã chi nhánh thực hiện

Bảng: **Nhanvien** có cấu trúc như sau:

Tên trường	Kiểu dữ liệu	Mô tả
<u>MaNV</u>	Text	mã nhân viên
TenNV	Text	họ tên nhân viên
Diachi	Text	Địa chỉ
Que	Text	quê quán
Ngaysinh	Date/Time	ngày sinh
MaCN	Text	chi nhánh mà NV đó làm việc

Bảng: **Duan_Nhanvien** có cấu trúc như sau:

Tên trường	Kiểu dữ liệu	Mô tả
<u>MaDuan</u>	Text	Mã dự án
<u>MaNV</u>	Text	mã nhân viên

Ghi chú: các trường gạch chân là các trường khoá của các bảng.

Hãy tạo quan hệ giữa các bảng trên.

CHƯƠNG 3

TRUY VẤN

1. TỔNG QUAN VỀ TRUY VẤN

Truy vấn(Query) là một công cụ mạnh của Access dùng để: tổng hợp, sắp xếp và tìm kiếm dữ liệu. Query cho phép thao tác trong các bảng dữ liệu.

Truy vấn thực chất là một câu lệnh SQL (Structured Query Language) được xây dựng nhờ công cụ của Access dùng để tổng hợp dữ liệu từ các bảng nguồn.

Ta có thể sử dụng chúng để chỉ định nội dung cho các Form và report. Query còn được dùng làm nguồn dữ liệu cho một trang Web.

Các loại truy vấn

1. Truy vấn chọn (Select): là loại truy vấn thông dụng nhất với các khả năng như sau:

- + Chọn bảng, query khác làm nguồn dữ liệu.
- + Chọn các trường hiển thị.
- + Thêm mới các trường là kết quả thực hiện các phép tính trên các trường của bảng nguồn.
- + Đưa vào các điều kiện tìm kiếm, lựa chọn.
- + Đưa vào các trường dùng để sắp xếp.

Sau khi truy vấn thực hiện, dữ liệu rút ra được tập hợp vào một bảng kết quả gọi là Dynaset, nó hoạt động như một bảng. Mỗi lần mở truy vấn, Access lại tạo một Dynaset gồm kết quả mới nhất của các bảng nguồn.

Loại Query này rất tiện lợi khi ta chỉ muốn thể hiện một số field trong một bảng có rất nhiều cột. Việc trích xuất một vài cột trong một bảng có thể làm tăng tốc độ thao tác của query.

Có thể chỉnh sửa, xoá, bổ sung thông tin vào các bảng nguồn thông qua Dynaset.

2. Paramter Query: Là truy vấn thông số, cho phép nhập các giá trị vào và hiển thị các dữ liệu thoả mãn điều kiện.

3. Crosstab Query: Truy vấn này tạo các bảng tổng hợp với dữ liệu dựa trên bảng hay Query.

4. Action Query: Thay vì trả về một tập hợp các dòng giống như một Select Query, các Action Query thực hiện một tác vụ dựa vào một hay nhiều bảng như tạo bảng mới, thêm, xoá sửa các mẫu tin trong bảng.

5. Union Query: là truy vấn kết hợp các trường tương ứng từ hai hay nhiều bảng.

6. Data-definition Query: Truy vấn được xây dựng từ một câu lệnh SQL dùng để tạo mới hoặc thay đổi cấu trúc bảng.

Chúng ta sẽ lần lượt tìm hiểu về các loại truy vấn trên trong các phần tiếp theo.

2. THAO TÁC TRÊN TRUY VẤN

2.1 Các bước tạo một truy vấn mới

Ta đã biết rằng khi thực hiện truy vấn, ta nhận được bảng tổng hợp gọi là bảng Dynaset. Nó không phải là bảng như đã xét trong chương 2, nó không được ghi vào đĩa và nó sẽ bị xoá ngay sau khi kết thúc (đóng) truy vấn.

Một truy vấn đã tạo có thể sử dụng để tạo các truy vấn khác.

Các bước chính để xây dựng một truy vấn gồm:

Bước 1: Trong cửa sổ Database mục Queries, chọn New hoặc các lựa chọn sau:

+ Create Query in Design view: Xây dựng truy vấn theo cách người sử dụng tự thiết kế

+ Create Query by wizard: Xây dựng truy vấn bằng công cụ Wizard

Bước 2: Chọn nguồn dữ liệu cho truy vấn mới: các bảng, các truy vấn đã tạo từ trước.

Bước 3: Tạo lập quan hệ giữa các bảng, truy vấn nguồn (nếu cần).

Bước 4: Chọn các trường từ các bảng, truy vấn nguồn để đưa vào truy vấn, mới.

Bước 5: Đưa vào các điều kiện để chọn lọc các mẫu tin thoả mãn các điều kiện đưa vào. Nếu không đưa vào các điều kiện để chọn lọc thì kết quả của truy vấn bao gồm tất cả các bản ghi từ các bảng, truy vấn nguồn.

Bước 6: Chọn các trường dùng để sắp xếp các mẫu tin trong Dynaset. Nếu không chọn trường sắp xếp thì các mẫu tin trong Dynaset được hiển thị theo thứ tự như trong các bảng nguồn.

Bước 7: Xây dựng các cột (trường) mới từ các trường đã có trong bảng, truy vấn nguồn. Giả sử trường mới là tổng, hiệu, tích, thương... của hai trường có sẵn nào đó. Khi xây dựng trường mới, có thể sử dụng các phép toán, các hàm chuẩn của Access và các hàm tự lập bằng ngôn ngữ Access Basic.

Ví dụ: Giả sử trong cơ sở dữ liệu **Quanlyhanghoa** gồm những bảng sau:

Hang(MaH, TenH, SLTon) để lưu trữ tất cả các mặt hàng có trong cửa hàng.

HoaDon(SoHD, NgayHD, MaK) dùng để lưu trữ các thông tin về việc bán hàng.

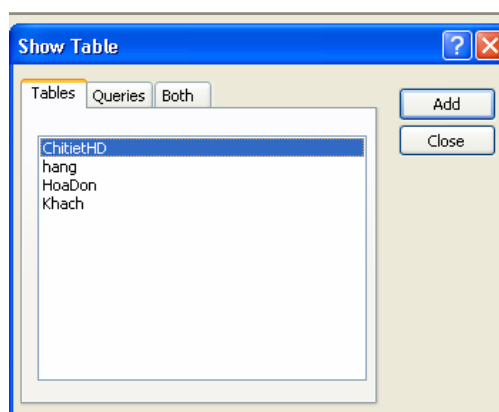
ChitietHD(SoHD, MaH, SLban, Dongia) dùng lưu trữ chi tiết về từng mặt hàng trong hoá đơn. Mỗi quan hệ giữa HoaDon và ChitietHD là một - nhiều (1-n) thông qua trường SoHD, giữa Hang và ChitietHD là 1-n thông qua trường MaH

Câu hỏi: Đưa ra thông tin về ngày bán, Tên hàng và số lượng hàng đã được bán.

Từ 3 bảng trên ta sẽ xây dựng truy vấn để trả lời câu hỏi theo trình tự sau:

Bước 1: Trong cửa sổ Database của cơ sở dữ liệu Quanlyhanghoa chọn mục Queries và chọn Create Query in Design view ta được hình 3.1

Bước 2: Trong hình 3.1 Access mở hộp Show Table cho ta chọn dữ liệu nguồn của truy vấn:



Hình 3.1

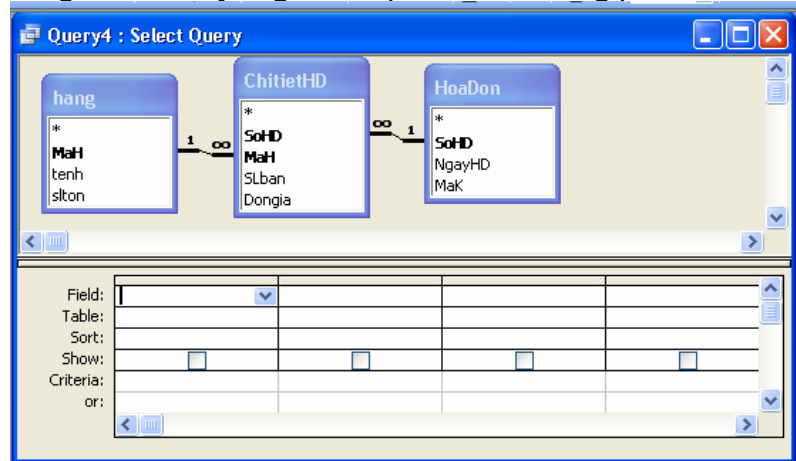
Tương tự trong phần tạo mối quan hệ tác dụng của 3 tab trên cửa sổ:

+ Tab Table để hiện các bảng có trong cơ sở dữ liệu

+ Tab Queries để hiện các truy vấn có trong cơ sở dữ liệu

+ Tab Both để hiện tất cả các bảng, các truy vấn có trong cơ sở dữ liệu. Chọn các bảng Hang, HoaDon, ChitietHD sau đó bấm nút Add. Rồi bấm Close để đóng cửa sổ Show Table (muốn hiện lại cửa sổ Show Table ta kích chuột tại biểu tượng trên thanh công cụ).

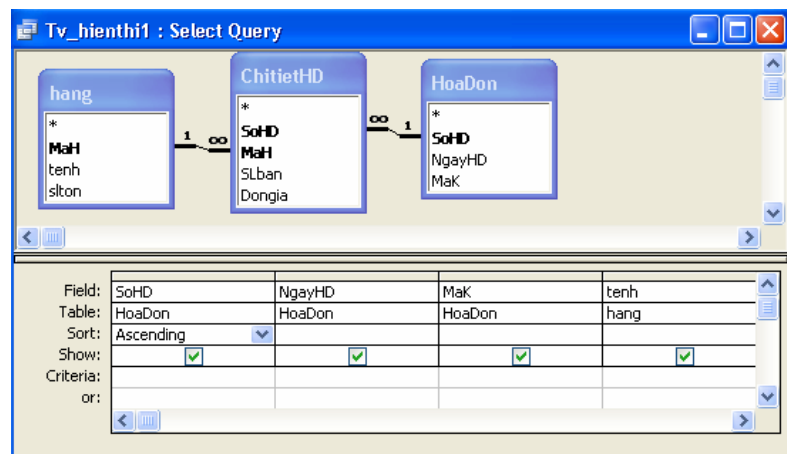
Kết quả ta nhận được của số Select Query gồm 2 phần (Hình 3.2). Phần trên hiện danh sách các bảng và truy vấn nguồn. Phần dưới gọi là QBE chứa các trường của truy vấn mới cần xây dựng. Để di chuyển giữa hai phần có thể dùng phím F6 hoặc chuột).



Hình 3.2

Bước 3: Tạo mối quan hệ giữa các bảng: Nếu khi tạo bảng ta đã tạo quan hệ (Relationship) giữa các bảng rồi thì nó sẽ tự động xuất hiện quan hệ đó khi tạo truy vấn (như Hình 3.2). Nếu không ta phải tạo quan hệ trên cửa sổ Query, quan hệ tạo tại đây chỉ có tác dụng đối với truy vấn này. Khi ra khỏi truy vấn nó không còn hiệu lực nữa.

Bước 4: Chọn các trường đưa vào truy vấn, bằng cách kéo tên trường trong các bảng/truy vấn nguồn từ phần trên đặt xuống dòng Field của phần dưới (hoặc chỉ cần nháy đúp chuột vào trường cần hiển thị). Ta nháy đúp vào các trường SoHD, NgayHD, MaK, TenH, SLban, Dongia (Hình 3.3).



Hình 3.3

Bước 5: Định thứ tự sắp xếp: (kết quả theo một thứ tự nào đó)

Bấm chuột tại ô Sort của trường cần sắp xếp

Chọn chiều muốn sắp xếp là tăng dần (Ascending) hoặc giảm dần (Descending).

Nếu muốn sắp theo nhiều trường thì thứ tự ưu tiên từ trái sang phải.

Trong ví dụ này ta sắp xếp theo chiều tăng dần của SoHD.


Bước 6: Ghi truy vấn: Sau khi đã hoàn chỉnh việc thiết kế cần ghi cấu trúc của truy vấn. Chọn Save từ menu File hoặc bấm chuột tại nút đóng cửa sổ window: Access sẽ yêu cầu đặt tên với lần ghi đầu tiên.

Các dạng hiển thị truy vấn: truy vấn có thể hiển thị theo các dạng sau:

- + Design View (dạng thiết kế)
- + SQL View (câu lệnh SQL tương ứng)
- + Datasheet View (Bảng kết quả của truy vấn)

Khi đang thiết kế truy vấn ở bất kỳ dạng hiển thị nào có thể chuyển sang dạng khác bằng cách chọn menu View rồi chọn dạng muốn hiển thị tương ứng.

Thực hiện truy vấn:

- Trong cửa sổ thiết kế truy vấn kích chuột tại biểu tượng:  hoặc chọn menu View / Datasheet

Trong cửa sổ Database chọn tab Queries rồi chọn truy vấn muốn thực hiện, bấm nút Open

Kết quả của truy vấn trên hình 3.4



SoHD	NgayHD	MaK	tenh	SLban	Dongia
HD1	03/03/2006	K1	kẹo	2	5000
HD1	03/03/2006	K1	bánh	3	7000
HD2	06/07/2006	K2	kẹo	1	5000
HD2	06/07/2006	K2	bia	3	8000

Hình 3.4

Việc đưa điều kiện vào cho truy vấn và tạo các trường tính toán mới sẽ được trình bày trong phần tiếp theo.

2.2 Đặt điều kiện trong truy vấn

Trong khi tạo truy vấn, đôi khi ta chỉ cần quan tâm đến một số thông tin thoả mãn một số điều kiện nào đó. Ví dụ chỉ hiển thị những hàng có SLban>3 ta thực hiện như sau:

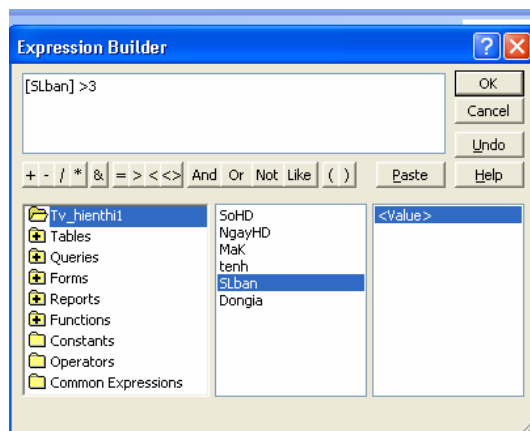
+ Đưa trực tiếp một biểu thức điều kiện vào ô Criteria của các trường cần đặt điều kiện. Tại dòng Criteria của ô SLban ta gõ điều kiện: >3.

Trong một truy vấn nếu có nhiều điều kiện của các trường cùng viết trên dòng Criteria, thì được hiểu là điều kiện và (có nghĩa là các bản ghi phải thoả mãn đồng thời các điều kiện đó mới được hiển thị ra). Còn nếu điều kiện đặt ở dòng OR thì hiểu là hoặc.

Ví dụ: nếu muốn hiển thị những hàng mà có SLban >3 và có Dongia>5000 thì tại dòng Criteria và cột SLban ta gõ >3, tại dòng Criteria và cột Dongia gõ >5000.

Nếu muốn hiển thị những hàng mà có SLban >3 hoặc có Dongia>5000 thì tại dòng Criteria và cột SLban ta gõ >3, còn tại dòng OR và cột Dongia gõ >5000.

+ Hoặc có thể dùng cửa sổ xây dựng biểu thức hoặc cửa sổ Zoom. Để mở cửa sổ xây dựng biểu thức cho ô Criteria của trường nào ta bấm phải chuột tại ô Criteria của trường, rồi chọn Build. Như hình 3.5



Hình 3.5

Tại đây ta có thể thiết lập biểu thức điều kiện bất kỳ cho truy vấn.

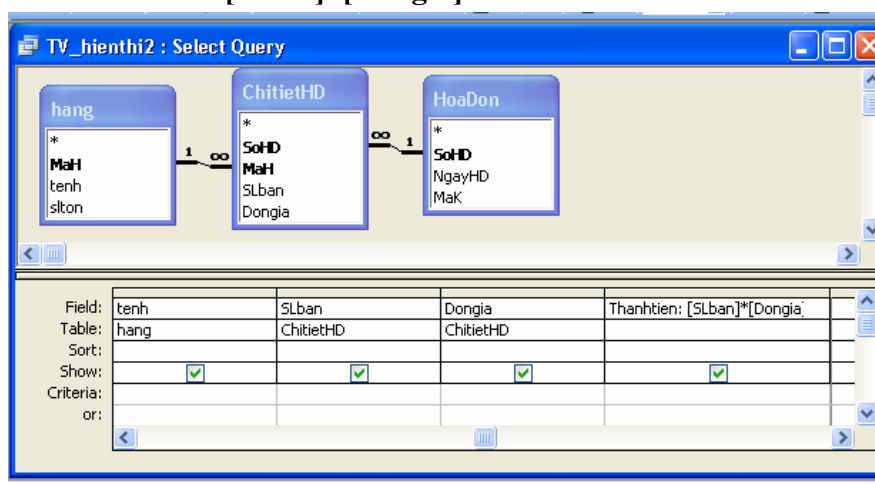
2.3 Tạo trường tính toán trong truy vấn

Để tạo thêm một trường mới vào truy vấn và giá trị của trường sẽ là kết quả tính toán của một biểu thức ta làm như sau:

- Tại cửa sổ thiết kế truy vấn. Chọn một ô còn trống, trên hàng Field đưa vào tên trường sau đó là dấu hai chấm: rồi đưa vào biểu thức. Trong biểu thức có thể dùng tên các trường, các hằng, các phép tính, các hàm chuẩn của Access và các hàm tự lập.

Trong ví dụ trên ta thêm trường mới: ThanhTien như sau:

ThanhTien: [SLban]*[Dongia] như trên hình 3.6



Hình 3.6

Thực hiện truy vấn ta có kết quả như hình 3.7:

	tenh	SLban	Dongia	ThanhTien
▶	kẹo	2	5000	10000
	bánh	3	7000	21000
	kẹo	1	5000	5000
	bia	3	8000	24000
*				

Hình 3.7

2.4 Thuộc tính của trường và truy vấn

Thuộc tính của trường:

Mỗi trường trong truy vấn đều có những thuộc riêng. Để mở bảng thuộc tính của trường ta chọn trường đó (bấm chuột tại ô Field của trường cần chọn) sau đó chọn Properties từ menu View hoặc bấm nút phải chuột và chọn Properties.

Thuộc tính của truy vấn:

Để mở bảng thuộc tính của truy vấn ta chọn cả truy vấn (bằng cách bấm chuột bên ngoài QBE và ngoài các các bảng/truy vấn nguồn ở phần trên) sau đó tiến hành như khi mở bảng thuộc tính của trường.

Sau khi đã mở bảng thuộc tính (của trường hay truy vấn) ta tiến hành chọn các giá trị thích hợp cho các thuộc tính (như khi đặt thuộc tính cho các trường của bảng). Theo mặc nhiên trường trong truy vấn kế thừa các thuộc tính trong bảng / truy vấn nguồn.

Vì vậy nếu không đặt lại các thuộc tính cho một trường trong truy vấn thì trường sẽ có các thuộc tính như trong bảng/truy vấn nguồn.

Hiển thị các trường của truy vấn:

Để hiển thị một trường nào đó trong kết quả truy vấn tại ô Show của trường ta chọn x. Nếu muốn một trường ẩn trong bảng kết quả ta bấm chuột tại ô show của trường đó để bỏ dấu chọn:

2.5 Các phép toán và một số hàm trong Access

Khi xây dựng các biểu thức điều kiện, ta có thể sử dụng các phép toán sau :

a) Các phép toán số học

Gồm các phép toán sau:

Kí hiệu	Tên	Ví dụ
^	Luỹ thừa	$2^3 = 8$
*	Nhân	$2*3 = 6$
/	Chia	$7 / 2 = 3.5$
\	Chia nguyên	$7 \setminus 2 = 3$
mod	Lấy phần dư	$7 \text{ mod } 3 = 1$
+	Cộng	$7 + 3 = 10$
-	Trừ	$7 - 3 = 10$

b) Phép toán ghép chuỗi:

Cú pháp như sau: chuỗi_1 & chuỗi_2 & & chuỗi_n
 hoặc chuỗi_1 + chuỗi_2 + + chuỗi_n

Sẽ cho một chuỗi mới bằng cách ghép các chuỗi thành phần lại. Các chuỗi_i có thể là các biểu thức chuỗi.

Ví dụ: “Chu Văn” & “ “ & “An”, cho chuỗi kết quả là “Chu Văn An”
 hoặc “Chu Văn” + “ “ + “An” cũng có kết quả vẫn là “Chu Văn An”

Chú ý: Phép & linh hoạt hơn phép +, vì phép & sẽ tự động chuyển các toán hạng từ dạng số sang dạng chuỗi rồi mới tiến hành ghép. Còn phép + chỉ cho phép ghép các toán hạng dạng chuỗi.

Ví dụ: “ Ha Tay” + 2006 sẽ sinh lỗi, nhưng “ Ha Tay” & 2006=”Ha Tay2006”

c) Các phép toán so sánh

- > Lớn hơn
- >= Lớn hơn hoặc bằng
- < Nhỏ hơn
- <= Nhỏ hơn hoặc bằng
- = Bằng nhau
- <> Khác nhau

Các phép toán so sánh cho kết quả là -1 (nếu đúng), là 0 (nếu sai).
 Trong Access Basic đã định nghĩa 2 hằng :

True = -1

False = 0

d) Các phép toán logic

And, Or, Not, Xor, Eqv (Equivalence), Imp (Implication)

Các phép toán có thể thực hiện trên các giá trị logic (-1 và 0), hoặc trên các giá trị nguyên (sẽ thực hiện trên từng bit) theo những quy tắc trong bảng sau:

x	Y	x and y	x or y	x xor y	x Eqv y	x Imp y
0	0	0	0	0	-1	-1
0	-1	0	-1	-1	0	-1
-1	0	0	-1	-1	0	0
-1	-1	-1	-1	0	-1	-1

e) Một số phép toán khác

Phép toán Between...And: Phép toán này dùng để biểu thị một khoảng giá trị. Ví dụ muốn tìm các mặt hàng đã bán có số lượng nằm từ 50 đến 100 ta đưa vào ô Criteria của trường SLban điều kiện sau: BETWEEN 100 AND 300

Điều kiện này có thể biểu diễn bằng cách khác như sau: >=100 AND <=300

Phép toán IN: Phép toán này dùng để kiểm tra xem giá trị của trường có nằm trong một tập hợp nào đó hay không. Cú pháp như sau:

IN(giá trị 1, giá trị 2,, giá trị n)

Ví dụ: Để tìm các mặt hàng có số lượng đã bán trong một hoá đơn nào đó thuộc một trong 3 giá trị: (20, 40, 60) ta đặt điều kiện sau vào ô Criteria của trường SLban: IN(110,120,21).

Phép toán LIKE: Phép toán này dùng đối với các trường kiểu Text và Date /Time để tìm các giá trị theo một mẫu nào đó.

Cách viết như sau: LIKE Mẫu

Mẫu là một chuỗi ký tự. Trong mẫu cho phép dùng các ký tự thế chân như sau:

- Ký tự * thay cho một dãy ký tự của trường kiểu Text hoặc một mục (ngày, tháng, năm, ...) của trường Date/Time.
- Ký tự ? thay cho một ký tự của trường kiểu Text hoặc ký tự (ngày, tháng, năm, ...) của trường Date/Time.

Chú ý: trong Access hằng ký tự được bao bởi cặp dấu nháy đơn ‘ ’, còn hằng xâu ký tự được bao bởi cặp dấu nháy kép “ ”, hằng ngày tháng được bao bởi cặp dấu thăng # #. Ta có hằng ký tự ‘H’, hằng xâu “Ha noi”, hằng ngày tháng # 01/02/2006#.

Ví dụ: Để tìm các mặt hàng có chữ B ở đầu ta đặt điều kiện sau vào ô Criteria của trường [TenH]: là Like “B*”

Để tìm các hoá đơn viết vào tháng 9 ta đặt điều kiện sau vào ô Criteria của trường NgayHD: like “*/09/*”

NULL và NOT NULL: Dùng để kiểm tra xem một trường đã có dữ liệu hay chưa.

Giá trị IS NULL tương đương với NULL, giá trị IS NOT NULL tương đương với NOT NULL.

Ví dụ: Để tìm các hoá đơn thiếu ngày viết hoá đơn ta đưa giá trị NULL vào ô Criteria của trường NgayHD.

Một số hàm trong Access:

Các hàm xử lý ngày:

Date(), cho ngày máy đang dùng

Now() cho kết quả là ngày và giờ hệ thống.

Day(date): cho ngày của tháng, có giá trị từ 1 đến 31

month(date): cho kết quả là tháng của date

year(date): cho kết quả là năm của date

Dateadd(đổi1, n, date) cho giá trị kiểu ngày bằng cách cộng thêm vào date n ngày hoặc tháng hoặc năm hoặc quý tùy theo đổi1.

Ví dụ: Dateadd("d",10,date()) cộng thêm 10 ngày so với ngày hiện tại.

DatePart(đổi1,date): Cho một thành phần là ngày hoặc tháng hoặc năm hoặc quý từ một giá trị kiểu ngày (date) tùy theo đổi1. Đổi1 có thể nhận các giá trị như trên. Đổi1 có thể nhận các giá trị sau: "d" – ngày, "m" – tháng, "q" – quý, "yyyy"- năm.

Ví dụ: Datepart("q",date()) cho quý hiện tại.

Các hàm xử lý chuỗi:

Len(St): cho độ dài xâu St

Trim(St): Xoá các dấu cách ở cả hai đầu của xâu st

Left(String,n) cho chuỗi con của string gồm n ký tự bên trái

Right(String,n) cho chuỗi con của string gồm n ký tự bên phải

Mid(String, start, n) Cho chuỗi con của String bắt đầu từ vị trí Start và gồm n ký tự.

Str(number) đổi số sang chuỗi.

Val(st) đổi xâu St thành số, nếu không thoả mãn thì cho giá trị 0.

Ví dụ: Left("Ha Noi",2)= "Ha", Right("Ha Noi",3)= "Noi",

Mid("Dang Thu Hien",6,3)= "Thu", Str(2006) kết quả là chuỗi "2006".

Ucase(Str): Biến đổi xâu Str thành chữ hoa.

Lcase(Str): Biến đổi xâu Str thành chữ thường.

Hàm điều kiện:

IFF(đk, bt1, bt2), hàm này sẽ cho giá trị là bt1 nếu biểu thức đk đúng, ngược lại cho giá trị là bt2. Bản thân mỗi bt1, bt2 lại có thể là một IFF.

Ví dụ: IFF(x>5,10,100), cho giá trị là 10 nếu x>5, trái lại (x<=5) thì cho giá trị là 100.

Ví dụ tổng hợp:

Giả sử có cơ sở dữ liệu **Quanlyhanghoa** gồm những bảng sau:

Hang(MaH, TenH, Loai, SLTon) để lưu trữ tất cả các mặt hàng có trong cửa hàng.

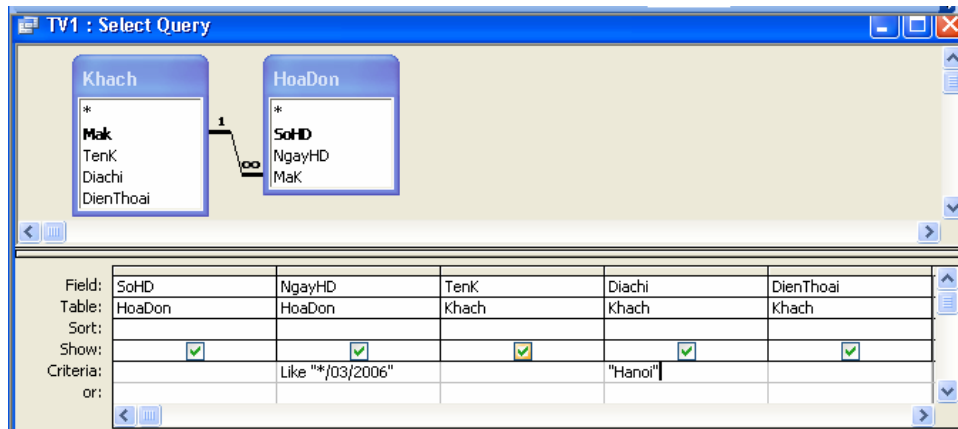
Khach(MaK, TenK, Diachi, DienThoai) lưu trữ những thông tin về khách hàng.

HoaDon(SoHD, NgayHD, MaK) dùng để lưu trữ các thông tin về việc bán hàng.

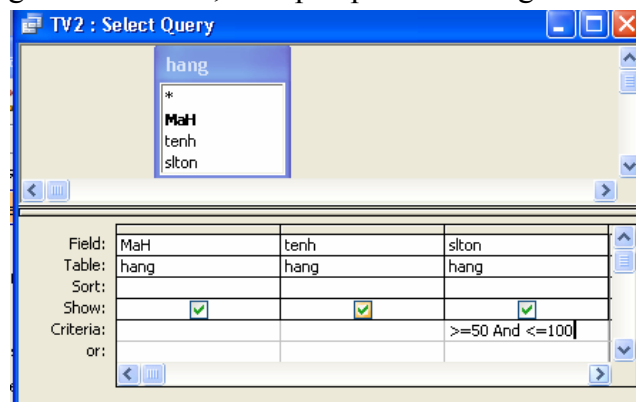
ChitietHD(SoHD, MaH, SLban, Dongia) dùng lưu trữ chi tiết về từng mặt hàng trong hoá đơn. Mỗi quan hệ giữa HoaDon và ChitietHD là một - nhiều (1-n) thông qua trường SoHD, giữa Hang và ChitietHD là 1-n thông qua trường MaH. Giữa Khách và HoaDon là 1-n thông qua trường MaK

Hãy viết các truy vấn trả lời các yêu cầu sau:

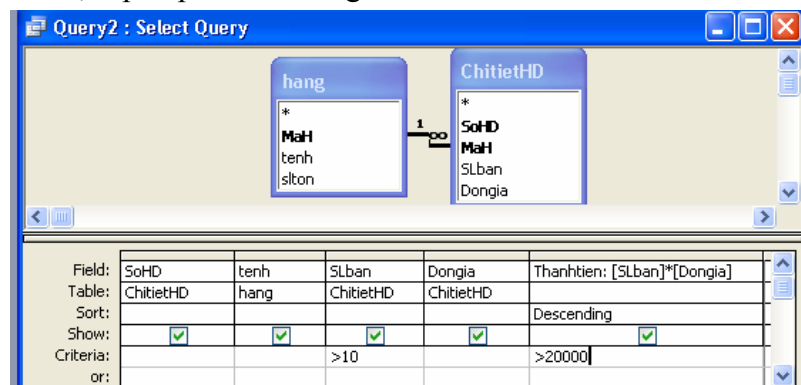
1) Hãy đưa ra bảng gồm các cột SoHD, NgayHD, TenK, Diachi và chỉ hiển thị những hoá đơn viết vào tháng 03 năm 2006 của những khách hàng ở Hanoi.



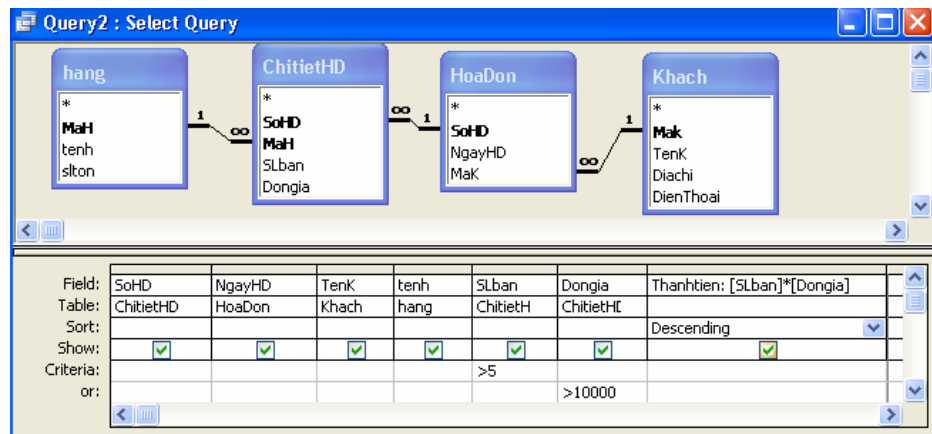
2) Đưa ra bảng gồm những cột MaH, TenH, Loai, SLton. Chỉ hiển thị những mặt hàng SLton trong khoảng từ 50 đến 100, và sắp xếp theo chiều giảm dần của SLton.



3) Đưa ra bảng gồm các cột SoHD, TenH, SLban, Dongia, ThanhTien trong đó ThanhTien=SLban*Dongia, và chỉ hiển thị những hàng có SLban>10 và có ThanhTien>20000, sắp xếp theo chiều giảm dần của ThanhTien.



4) Đưa ra bảng gồm những cột SoHD, NgayHD, TenK, TenH, SLban, Dongia, ThanhTien. Chỉ hiển thị những mặt hàng có SLban>5 hoặc Dongia>10000, và sắp xếp theo chiều giảm dần của ThanhTien.



3. CÁC LOẠI TRUY VẤN KHÁC

3.1 Truy vấn tính tổng (Totals)

Truy vấn này cho phép phân nhóm các bản ghi sau đó thực hiện các phép tính sau trên từng nhóm:

- Sum: Tính tổng các giá trị trên trường kiểu Number
- Avg: Tính giá trị trung bình của trường kiểu Number
- Min: Tìm giá trị nhỏ nhất của trường kiểu Number
- Max: Tìm giá trị lớn nhất của trường kiểu Number
- Count: Đếm số giá trị khác rỗng của trường
- First: Cho giá trị của trường mẫu tin đầu tiên trong nhóm
- last: Cho giá trị của trường mẫu tin cuối cùng trong nhóm

Các bước tạo truy vấn tính tổng

1. Cách tạo như truy vấn đơn giản. Chọn dữ liệu nguồn cho truy vấn.
2. Chọn mục Totals trong menu View. Sẽ xuất hiện dòng Total trên QBE
3. Chọn các trường:
 - Phân nhóm (Group by trên ô Total)
 - Điều kiện, tiêu chuẩn tham gia phân nhóm và tính tổng:
 - + Có chữ Where trên ô Total
 - + Có biểu thức điều kiện trên ô Criteria
 - Tính toán (chọn một hàm đặt vào ô Total, đổi tên trường cho có ý nghĩa tính toán)
 - Chọn thứ tự hiển thị (dùng ô Sort của trường Group by và trường tính toán)
 - Định tiêu chuẩn hiển thị kết quả (đưa điều kiện vào ô Criteria của các trường phân nhóm và các trường tính toán).

Cách thức làm việc của truy vấn tính tổng:

1. Dựa vào các điều kiện đặt trong các trường (có giá trị where trên ô total) để lọc ra các mẫu tin tham gia phân và tính toán.
2. Phân nhóm và sắp xếp các mẫu tin trong từng nhóm theo các trường phân nhóm (có giá trị Group by trên ô Total). Nhóm này là dãy các bản ghi có cùng giá trị trên các trường phân nhóm.

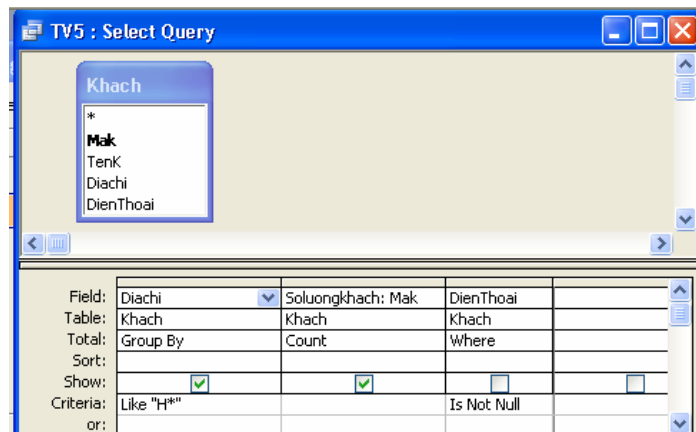
- Thực hiện phép tính theo từng nhóm trên các trường tính toán (có hàm Sum, Avg,...) trên ô Total.
- Nếu có các điều kiện trên các trường phân nhóm và tính toán thì chỉ các dòng thoả mãn điều kiện này mới được hiển thị.
- Nếu dùng thuộc tính Top Value của truy vấn thì chỉ một số dòng bên trên mới được hiển thị. Giả sử giá trị của Top Value là 2 thì chỉ có 2 dòng đầu của mỗi nhóm được hiển thị trong bảng kết quả.

Ví dụ: Cho CSDL **Quanlyhanghoa** như phần trên. Hãy đưa ra bảng tổng hợp số lượng khách hàng tại mỗi địa chỉ (ở đây hiểu địa chỉ của khách hàng chỉ gồm tên tỉnh), chỉ thống kê với những khách hàng có số điện thoại (nghĩa là cột số điện thoại không rỗng), và chỉ hiển thị các địa chỉ bắt đầu bằng chữ H.

Ta thực hiện các bước như sau:

- Chọn bảng Khách làm nguồn cho truy vấn
- Chọn các trường Diachi, Mak, Dienthoai
- Chọn mục Totals của menu view
- Tại trường Diachi:
 - Bấm chuột tại hàng Total và chọn Group by
 - Gõ điều kiện: like "H*" tại hàng Criteria
- Tại trường Mak: Chọn Count tại hàng Total và đổi tên là **Soluongkhach**
- Tại trường Dienthoai
 - Chọn Where tại ô Total
 - Gõ Is Not Null tại ô Criteria
 - Trường điều kiện sẽ không được hiển thị trong bảng kết quả.

Khi đó ta có cửa sổ thiết kế như hình 3.8



Hình 3.8

3.2 Truy vấn thông số (Parameter Query)

Các Parameter Query có tên như vậy bởi vì chúng yêu cầu nhập một tham số vào thời điểm trước khi query có thể thực hiện xong. Thay vì, chúng ta đưa những điều kiện cố định khi xây dựng truy vấn thì ta có thể đưa những điều kiện “động” hơn tại mỗi thời điểm chạy truy vấn. Tức là giá trị trong các biểu thức điều kiện không cần định trước mà sẽ được nhắc nhở việc nhập mỗi khi thực hiện truy vấn. Ta có thể nhắc nhở việc nhập một hoặc nhiều tham số.

Các bước xây dựng

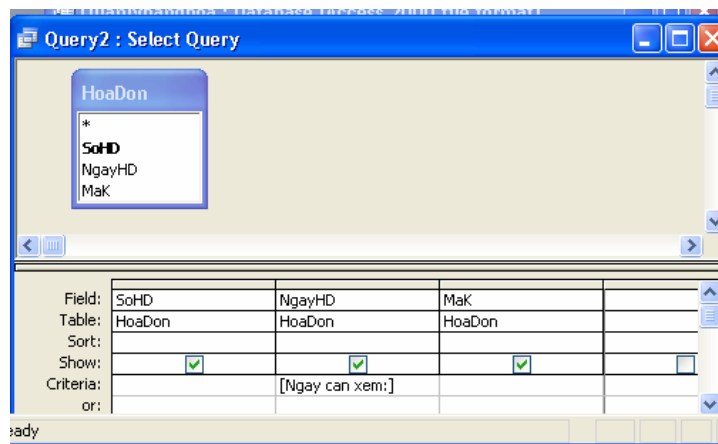
Khi tạo biểu thức điều kiện thay cho một giá trị cụ thể ta đưa vào một thông báo được bao trong hai dấu : []. Một thông báo như vậy gọi là một thông số (tham số).

Dòng thông báo sẽ xuất hiện như một chỉ dẫn khi nhập liệu thực hiện truy vấn.

Ví dụ minh họa

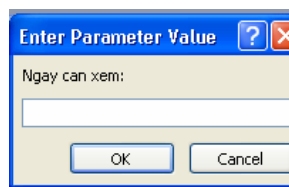
Giả sử với CSDL **Quanlyhanghoa** như trên, ta muốn thiết kế 1 truy vấn, để khi chạy có thể nhập vào một ngày bất kỳ và nó liệt kê ra toàn bộ những hoá đơn được lập trong ngày đó.

Ta thực hiện như sau: Tạo truy vấn mới với dữ liệu nguồn là bảng HoaDon, tại dòng Criteria của trường NgayHD ta đưa vào dòng thông báo: [Ngày cần xem:], Cửa sổ thiết kế như hình 3.9



Hình 3.9

Khi thực hiện truy vấn xuất hiện cửa sổ Hình 3.10



Hình 3.10

Ta nhập ngày cần xem rồi ấn OK, khi đó sẽ hiển thị toàn bộ những hoá đơn lập vào ngày vừa nhập.

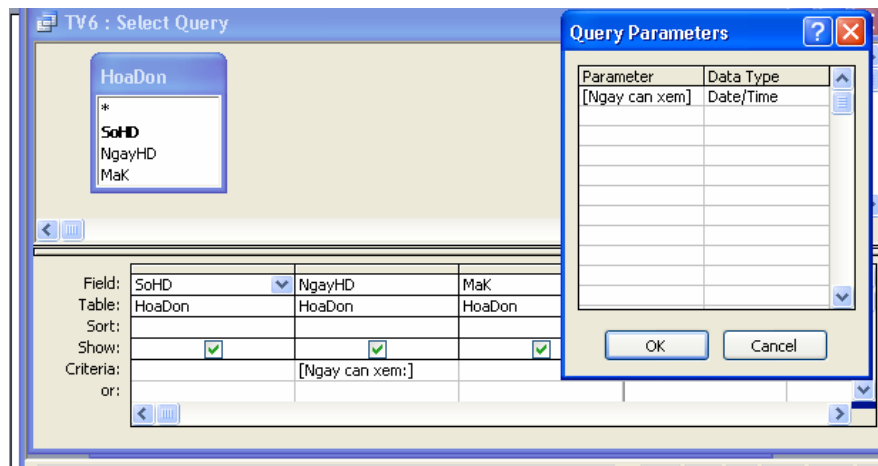
Để truy vấn thông số chạy ổn định và chính xác hơn thì người ta khuyến cáo rằng nên khai báo kiểu dữ liệu cho các thông số.

Qui định kiểu dữ liệu cho thông số

Để quy định kiểu dữ liệu cho các thông số ta làm như sau :

- Mở truy vấn thông số ở chế độ Design View
- Chọn mục Parameters từ menu Query để mở cửa sổ Query Parameters.
- Lần lượt đưa vào tên thông số và chọn kiểu dữ liệu thích hợp và ấn OK.

Khi đó cửa sổ Query Parameters như hình 3.11:



Hình 3.11

3.3 Truy vấn Crosstab

a) Công dụng:

- Truy vấn Crosstab được dùng để tóm lược dữ liệu và trình bày kết quả theo dạng cô đọng như một bảng tính, với một hình thức rất dễ đọc. Vì vậy nó thường được dùng để tạo nên số liệu để so sánh và tiên đoán hướng phát triển của dữ liệu.

- Có 2 phần việc chính trong truy vấn Crosstab là :

+ Thực hiện phân nhóm và tính toán (như truy vấn tính tổng)

+ Sau đó trong mỗi nhóm ta lại chia thành các nhóm con, thực hiện phép toán trên mỗi nhóm con và trình bày kết quả theo từng cột (mỗi nhóm con ứng với một cột)

Ví dụ : Giả sử trong bảng **Hang**, bảng **ChitietHD** của CSDL **Quanlyhanghoa** gồm các trường MaH, TenH, Loai, SoHD, MaH, SLban ta có thể dùng truy vấn Crosstab để đưa ra bảng thống kê xem trong mỗi hoá đơn có bao nhiêu loại hàng và mỗi loại hàng đó có số lượng như thế nào?:

SoHD	Tổng SL hàng bán	A	B	C
HD1	15	3	7	2
HD2	17		10	5
HD3	16	1	15	
HD4	20		10	4

b) Cấu trúc của truy vấn:

A. Gồm các trường như truy vấn Total:

1. Các trường phân nhóm (Total: Group By)
2. Các trường tính toán (Total: Sum hoặc Avg)
3. Các trường điều kiện (Total: Where)

B. Thêm hai trường

1. Trường phân nhóm con theo cột:
 - Ô Total: Group by
 - Ô Crosstab: Column Heading

2. Trường tính toán trên các nhóm con
 - Ô Total: Hàm (Sum, Count, Avg....)
 - Ô Crosstab: Value

Chú ý : Nội dung của ô Crosstab đối với các trường của nhóm A như sau:

1. Trường điều kiện: Để trống
2. Các trường phân nhóm và tính toán, nếu:
 - Để trống: không hiển thị
 - Row heading: Hiển thị các giá trị theo hàng

c) Cách xây dựng truy vấn:

Bước 1: Chọn các bảng/ truy vấn nguồn để xây dựng truy vấn mới.

Bước 2: Chọn mục Crosstab Query trong menu Query

Kết quả: Trong QBE xuất hiện thêm hai hàng là: Total và Crosstab.

Bước 3: Chọn các trường

1. Dùng làm tiêu chuẩn lựa chọn các bản ghi:
 - Total: Where
 - Crosstab: để trống
 - Criteria: điều kiện chọn
2. Phân nhóm và hiển thị theo hàng
 - Total: Group by
 - Crosstab: Row heading
 - Dòng Sort, Criteria, Or có thể dùng
3. Tính theo nhóm và hiển thị theo hàng
 - Total: Group by
 - Crosstab: Row heading
 - Dòng Sort, Criteria, Or có thể dùng
4. Một trường dùng để phân nhóm con và hiển thị theo cột
 - Total: Group by
 - Crosstab: Column Heading
5. Một trường dùng để tính toán trên các nhóm con và hiển thị kết quả theo bảng
 - Total: Hàm (Sum, Count, Avg, ...)
 - Crosstab :Value

Bước 4: Ghi và đặt tên cho truy vấn.

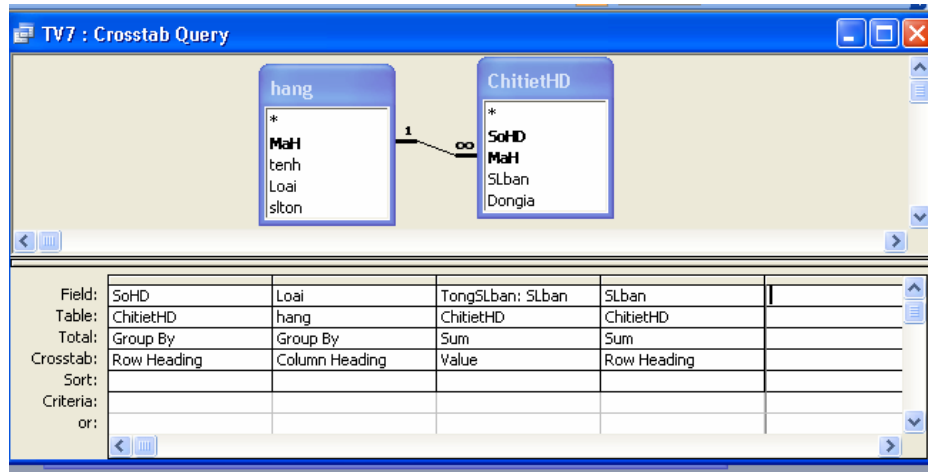
Ví dụ 1: Giả sử có bảng **Hang, ChitietHD** của CSDL **Quanlyhanghoa** gồm các trường MaH, TenH, Loai, SoHD, MaH, SLban. Hãy đưa ra bảng thống kê trên.

Các bước thực hiện:

1. Chọn bảng **Hang, ChitietHD** làm dữ liệu nguồn
2. Chọn kiểu truy vấn Crosstab
3. Đưa vào truy vấn các trường sau:
 - Trường phân nhóm là: **SoDH**

- Trường tính toán trên nhóm là **SLban** (để tính tổng số lượng hàng bán của mỗi hoá đơn).
- Trường phân nhóm con là trường **Loai** (mỗi loại thống kê xem có số lượng bán như thế nào?)
- Trường tính toán trên nhóm con là trường **SLban**(để tính tổng số lượng hàng bán của mỗi loại)

Màn hình thiết kế như hình 3.12



Hình 3.12

Bước 4: Ghi lại thiết kế truy vấn và chạy khi đó ta được kết quả như hình 2.13

SoHD	SumOfSLban	A	B	C
HD1	5	2	3	
HD2	4	1		3

Hình 3.13

Ví dụ 2: giả sử có bảng **HOCSINH (KHOI, LOP, TO, LOAI, HOTEN)**

Yêu cầu đặt ra là xây dựng bảng tổng hợp theo mẫu như sau, chỉ thống kê với khối 9.

Lớp	Tổ	Sĩ số	Kém	Đạt	Khá	Giỏi
9A	1	10	1	4	2	3
9A	2	12		5	5	2
9A	1	9	2	7		
9B	2	14		5	4	1

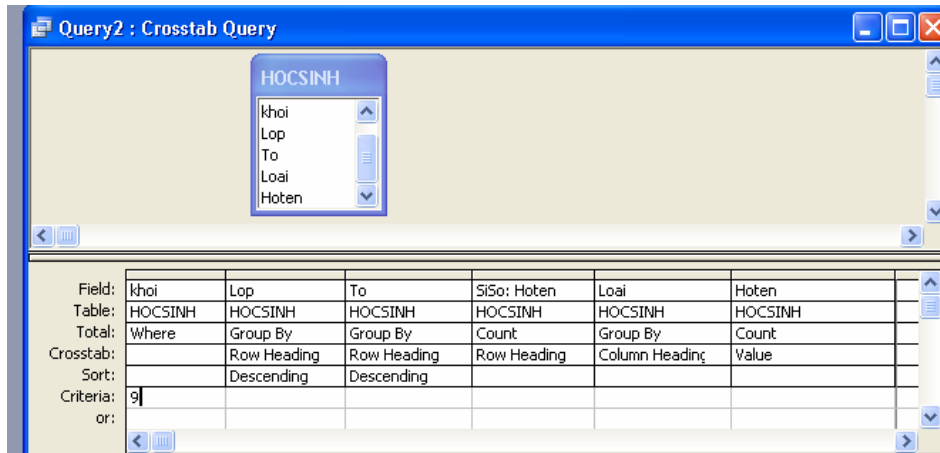
Trong ví dụ này chúng ta có các trường phân nhóm, đó là LOP, TO và trường LOAI sẽ là phân nhóm nhỏ để hiển thị theo cột.

Các bước làm như sau:

1. Chọn bảng HOCSINH làm nguồn cho truy vấn
2. Chọn kiểu truy vấn Crosstab
3. Đưa vào truy vấn các trường sau:
 - Trường điều kiện là trường KHOI
 - Các trường phân nhóm là: LOP và TO
 - Trường tính toán (để tính sĩ số theo tổ) là trường HOTEN

- Trường phân nhóm con là trường LOAI (mỗi tổ thống kê xem có bao nhiêu học sinh giỏi, khá, trung bình...).
- Trường tính toán trên nhóm con là trường HOTEN (đếm số học sinh từng loại).

Cửa sổ thiết kế truy vấn có dạng hình 3.14:



Hình 3.14

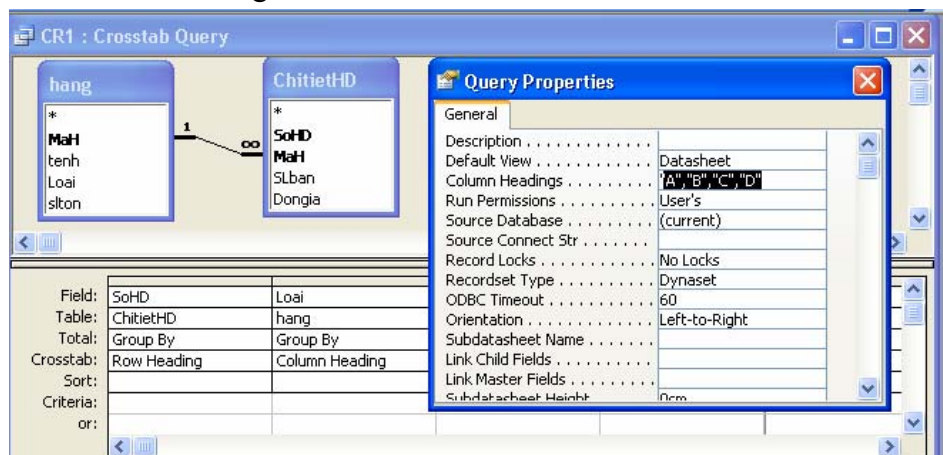
d) Thứ tự hiển thị các cột

Qua bảng kết quả của ví dụ 1 ta thấy, các cột “A”, “B”, “C”, được sắp xếp theo thứ tự ABC của tiêu đề cột. Nếu không có mặt hàng nào loại “D” thì không có cột “D” trong bảng kết quả.

Ta có thể sắp xếp các cột kết quả theo ý muốn và thêm các cột rỗng, ta cần dùng thuộc tính Column Heading của truy vấn. Thực hiện như sau:

- + Mở truy vấn trong chế độ Design
- + Mở bảng thuộc tính của truy vấn
- + Trong dòng thuộc tính Column Heading ta đặt tiêu đề các cột cần hiển thị theo thứ tự từ trái sang phải.

Trong ví dụ trên ta có bảng thuộc tính như hình 3.15:



Hình 3.15

Thực hiện truy vấn ta thu được kết quả như hình 3.16: Ta thấy ở đây cột “D” không có dữ liệu nhưng vẫn được hiển thị.

	SoHD	SumOfSLban	A	B	C	D
HD1		5	2	3		
HD2		4	1		3	

Hình 3.16

3.4 Truy vấn tạo bảng (Make Table Query)

Công dụng:

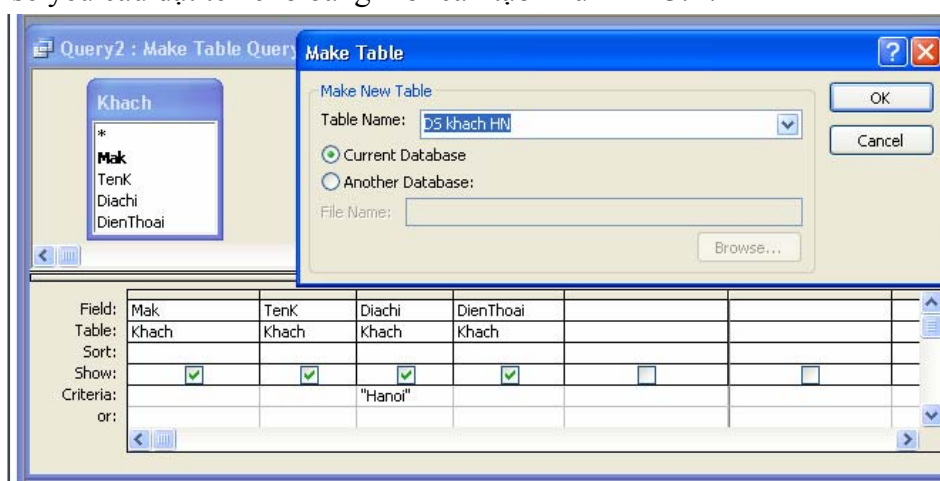
Truy vấn này cho phép tạo ra một bảng mới dựa trên bảng nguồn đã được cung cấp. Khi thực hiện truy vấn, kết quả của truy vấn được ghi ra đĩa dưới dạng một bảng mới.

Các bước xây dựng:

Bước 1. Tiến hành thiết kế như đối với một truy vấn chọn thông thường.

Bước 2. Sau đó chọn menu Query/Make Table Query:

Access sẽ yêu cầu đặt tên cho bảng mới cần tạo như hình 3.17



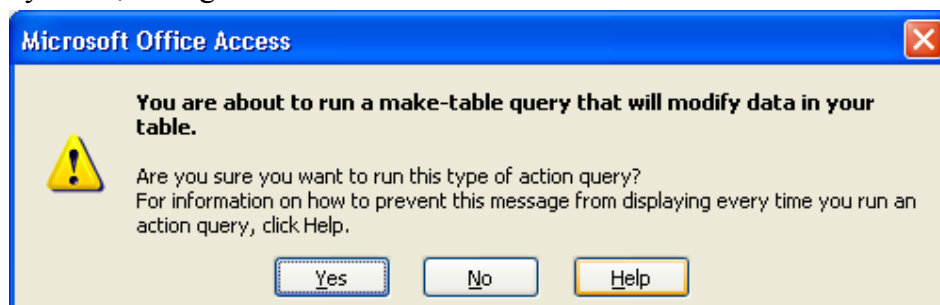
Hình 3.17

Tại đây ta có thể xác định bảng mới cần tạo nằm trong CSDL hiện tại hoặc có thể nằm trong CSDL khác, khi đó ta bấm chọn “Another Database” và đưa vào tên file CSDL đó.

Bước 3. Đặt tên bảng vào ô Table Name và chọn OK để trở về cửa sổ thiết kế truy vấn.

Bước 4. Ghi truy vấn và thực hiện truy vấn để nhận bảng mới.

Chú ý: Do tính nguy hiểm của truy vấn hành động nên khi thực hiện một truy vấn hành động Access sẽ ra thông báo yêu cầu sự khẳng định. Nội dung của thông báo trong truy vấn tạo bảng như sau:



Nếu muốn bỏ sự nhắc nhở này của Access thì ta làm như sau:

Chọn menu Tools, Option

Chọn Edit/ Find

Bỏ đánh dấu tại mục: Confirm/ Action Queries

Ví dụ: Giả sử ta có CSDL **Quanlyhanghoa** gồm 2 bảng:

HoaDon(SoHD,NgayHD,MaK), ChitietHD(SoHD,MaH,SLban,Dongia)

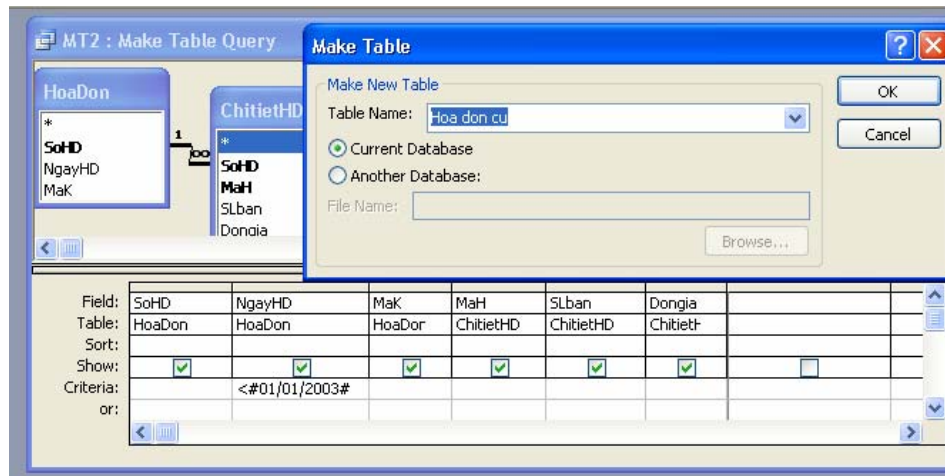
Hãy xây dựng truy vấn để tạo một bảng mới có tên là [Hoa don cu], dùng để lưu những hoá đơn bán hàng trước ngày 1/1/2003.

Các bước thực hiện như sau:

Bước 1: Tạo như truy vấn thông thường, bảng dữ liệu nguồn là HoaDon, ChitietHD. Điều kiện tại trường NgayHD là: <#01/01/2003#.

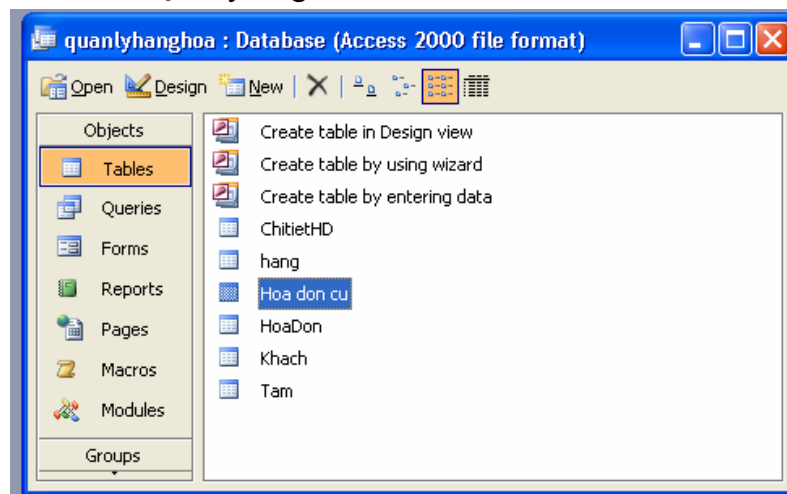
Bước 2: Chọn menu Query/Make Table Query.

Bước 3: Đặt tên mới là [Hoa don cu] và ấn OK. xem hình 3.18



Hình 3.18

Bước 4: Đặt tên truy vấn là MT2, chạy truy vấn này ta được 1 bảng mới nằm trong phần Tables của CSDL Quanlyhanghoa. Như hình 3.19



Hình 3.19

3.5 Truy vấn bổ sung (Append Query)

Công dụng:

Bổ sung dữ liệu từ một bảng/truy vấn vào cuối một bảng đã có.

Các bước xây dựng:

Bước 1: Tiến hành như đối với một truy vấn chọn thông thường.

Bước 2: Chọn menu Query/Append Query. Access sẽ yêu cầu nhập tên bảng cần bổ sung dữ liệu (bảng đó cần phải tồn tại).

Bước 3: Nhập tên bảng cần bổ sung dữ liệu.

Bước 4: Trong hàng Field: Đưa các trường của bảng/truy vấn nguồn.

Bước 5: Trong hàng Append to: Đưa các trường tương ứng của bảng nhận (bảng cần bổ sung dữ liệu).

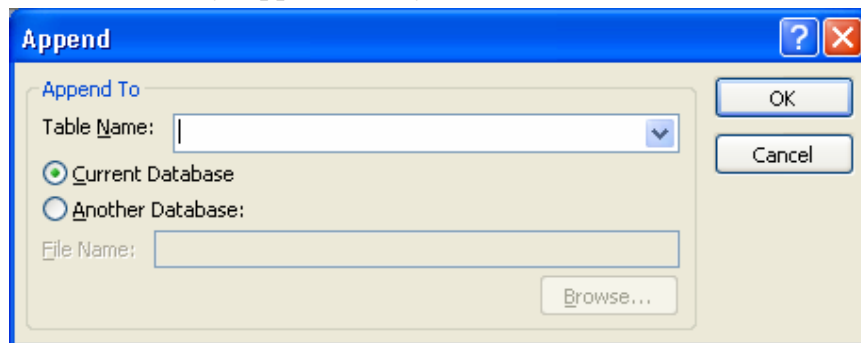
Chú ý: Nếu bảng/truy vấn nguồn có cùng cấu trúc như bảng nhận (cùng số trường, thứ tự, tên và kiểu) thì có thể kéo dấu * của bảng/truy vấn nguồn vào hàng Field. Nếu bảng nhận không có các trường như ở bảng nguồn thì các trường đó tự động được bỏ qua khi ta chọn dấu *.

Ví dụ: Giả sử trong CSDL **Quanlyhanghoa** có bảng:

Khach(MaK,TenK,Diachi,Dienthoai), và cũng có một bảng **KhachHCM** có cấu trúc giống như bảng **Khach** nhưng chứa tất cả những khách hàng có địa chỉ ở phía nam. Bây giờ cần bổ sung toàn bộ những khách hàng trong bảng **KhachHCM** vào cuối của bảng **Khach**. Ta thực hiện như sau:

Bước 1: Tạo một truy vấn thông thường với dữ liệu nguồn là bảng **KhachHCM** (là bảng lấy dữ liệu ra để bổ sung)

Bước 2: Chọn menu Query/Append Query, ta có được như hình 3.19



Hình 3.19

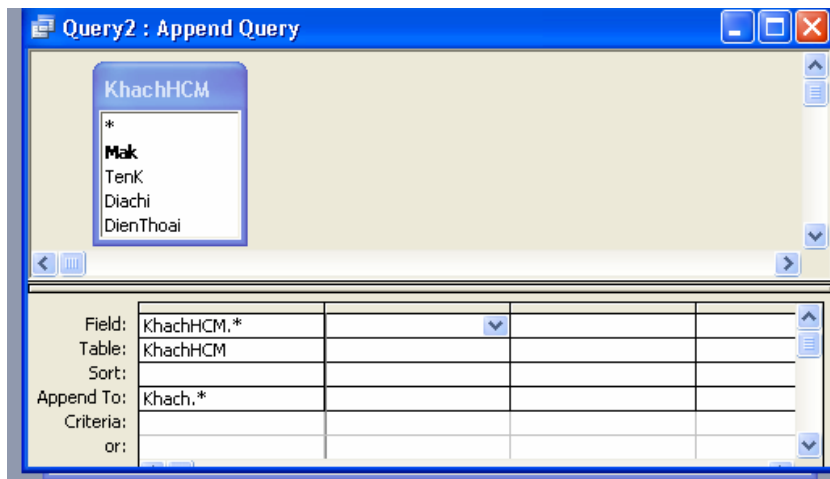
Bước 3: Trong hộp Table Name, ta đưa vào tên bảng **Khach** (là bảng sẽ được dữ liệu bổ sung vào) rồi bấm OK. Kết quả sẽ xuất hiện thêm dòng Append to trong vùng QBE

Bước 4: Kéo dấu * của bảng nguồn (**KhachHCM**) vào 1 ô của hàng field. Kết quả

- Trong ô được kéo xuống hiện nội dung: **KhachHCM.***

- Trong ô tương ứng trên hàng Append to xuất hiện nội dung: **Khach.***

Cửa sổ thiết kế như hình 3.20



Hình 3.20

Bước 5: Ghi và thực hiện truy vấn, ta sẽ thấy toàn bộ phần dữ liệu của bảng **KhachHCM** đã được bổ sung vào cuối bảng **Khach**.

3.6 Truy vấn xoá (Delete Query)

Công dụng: Xoá các mẫu tin từ bảng nguồn. Ta nên thận trọng khi thực hiện truy vấn

Cách xây dựng:

Bước 1: Chọn các bảng/truy vấn nguồn.

Bước 2: Chọn menu Query/Delete Query. Kết quả sẽ hiện thêm hàng Delete trong nửa dưới của cửa sổ thiết kế truy vấn.

Bước 3: Kéo dấu * của bảng muốn xoá vào ô Field. Khi đó từ From xuất hiện tại ô tương ứng trên dòng Delete.

Chú ý: Chỉ được xoá các bản ghi ở một bảng trong số các bảng nguồn.

Bước 4: Kéo các trường dùng để lập điều kiện vào dòng Field. Từ Where xuất hiện trên dòng Delete.

Chú ý: Có thể sử dụng trường biểu thức để lập điều kiện. Khi đó cần kéo thêm một số trường dùng để tạo ra trường biểu thức.

Bước 5: Ghi và thực hiện truy vấn.

Ví dụ:

Giả sử trong CSDL **Quanlyhanghoa** có bảng:

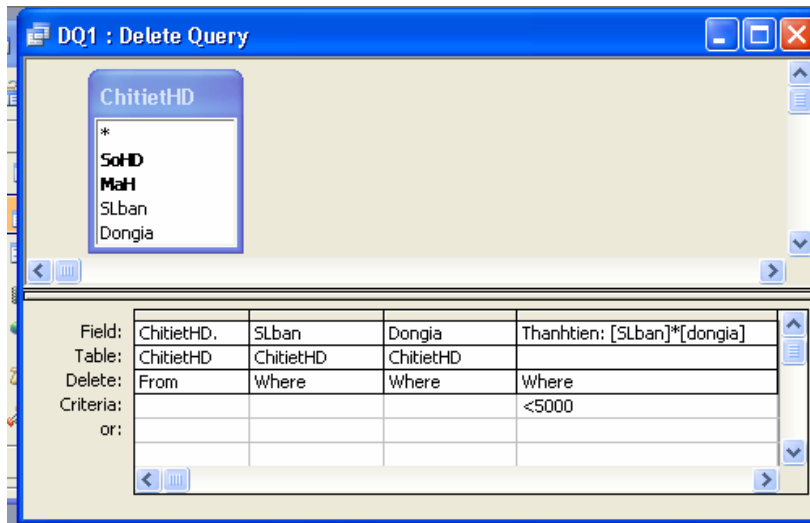
ChitietHD(SoHD, MaH, SLban, Dongia). Ta cần xoá bỏ những bản ghi trên bảng này mà có $ThanhTien = SLban * Dongia < 5000$. Ta thực hiện như sau:

Bước 1: Chọn bảng ChitietHD làm dữ liệu nguồn.

Bước 2: Chọn menu Query/Delete Query. Kết quả sẽ hiện thêm hàng **Delete** trong nửa dưới của cửa sổ thiết kế truy vấn.

Bước 3: Kéo dấu * của bảng ChitietHD ô Field.

Bước 4: Kéo các trường SLban, Dongia vào hàng Field để tạo trường ThanhTien. Tạo trường Tongtien: $SLban * Dongia$. Nếu không thiết lập điều kiện thì đồng nghĩa với việc xoá tất cả các bản ghi của bảng đó. Cửa sổ thiết kế truy vấn trên hình 3.21



Hình 3.21

Thực ra trong ví dụ này có thể không cần thiết kéo các trường SLban, Dongia vào, bởi đây chính là những trường thuộc bảng cần xóa.

Bước 5: Ghi và thực hiện truy vấn. Sau khi chạy truy vấn Access còn đưa ra những lời cảnh báo để nhắc nhở người thực hiện. Thực hiện truy vấn này ta sẽ xóa hết những bản ghi trong bảng ChitietHD mà có ThanhTien<5000.

3.7 Truy vấn chỉnh sửa (Update Query)

Công dụng: Cho phép chỉnh sửa dữ liệu trong một hay nhiều bảng.

Cách xây dựng:

Bước 1: Chọn các bảng/truy vấn nguồn.

Bước 2: Chọn menu Query/Update Query. Kết quả hiện thêm hàng **Update to** trong phần QBE.

Bước 3: Kéo các trường cần sửa và các trường dùng để thiết lập điều kiện vào hàng Field. Cùng một trường có thể vừa là trường cần sửa vừa là trường đặt điều kiện.

Bước 4: Đặt các biểu thức vào ô **Update To** của các trường cần sửa. Giá trị của biểu thức sẽ là giá trị mới của các trường này. Trong biểu thức có thể sử dụng các trường của bảng/truy vấn nguồn, các hàm chuẩn của Access và các hàm tự lập.

Bước 5: Đặt các biểu thức điều kiện vào các ô Criteria của các trường điều kiện.

Bước 6: Ghi và thực hiện truy vấn.

Ví dụ 1:

Giả sử trong CSDL **Quanlyhanghoa** có bảng:

ChitietHD(SoHD, MaH, SLban, Dongia). Giả sử vì một lý do nào đó, cần giảm 5%Dongia cho những mặt hàng có SLban>100. Ta thực hiện như sau:

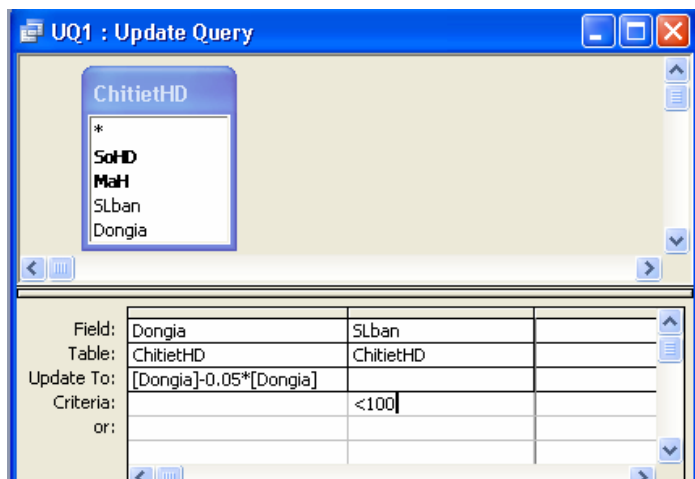
Bước 1: Chọn bảng **ChitietHD** làm dữ liệu nguồn.

Bước 2: Chọn menu Query/Update Query. Kết quả hiện thêm hàng **Update to** trong phần QBE.

Bước 3: Kéo trường **Dongia, SLban** vào hàng Field.

Bước 4: Trong ô **Update To** của trường **Dongia** đưa vào biểu thức sau: [Dongia]-0.05*[Dongia].

Bước 5: Trong ô Criteria của **SLban** đặt vào điều kiện: >100. Cửa sổ thiết kế như hình 3.22



Hình 3.22

Bước 6: Ghi và thực hiện truy vấn ta thấy giá của những mặt hàng trong bảng ChitietHD đã được thay đổi nếu nó có SLbán tại mỗi lần >100.

Ví dụ 2:

Kết quả trong một cuộc thi Olympic Tin học được chứa trong hai bảng:

Thisinh(Hoten, Phach, Tentruong), Ketqua(Phach, Diem)

Vì kết quả hơi thấp nên ban tổ chức quyết định cộng thêm cho mỗi thí sinh 2 điểm.

Xây dựng truy vấn để thực hiện yêu cầu trên, đồng thời cũng sửa toàn bộ trường **Hoten** thành chữ hoa. Ta thực hiện như sau:

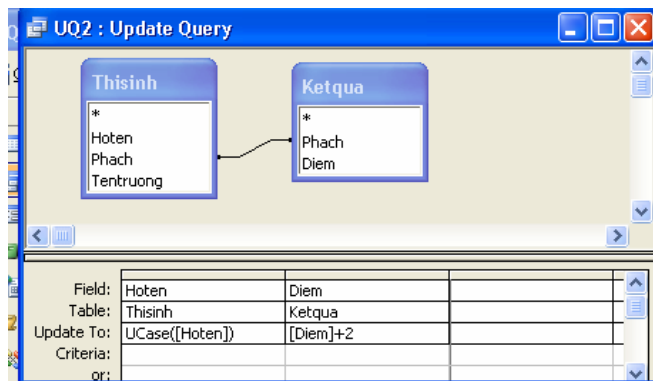
Bước 1: Chọn bảng **Thisinh** và **Ketqua** làm dữ liệu nguồn.

Bước 2: Chọn menu Query/Update Query. Kết quả hiện thêm hàng **Update to** trong phần QBE.

Bước 3: Kéo trường **Hoten, Diem** vào hàng Field.

Bước 4: Trong ô **Update To** của trường **[Hoten]** đưa vào biểu thức sau: UCASE([Hoten]). Trong ô **Update To** của trường **[Diem]** đưa vào biểu thức: [Diem]+2.

Cửa sổ thiết kế như hình 3.23



Hình 3.23

Bước 5: Ghi và thực hiện truy vấn ta thấy giá của cả hai bảng đều có sự thay đổi.

Ví dụ 3:

Giả sử trong CSDL **QuanlySinhvien** có bảng

Danhsach(MaSV, Hoten, Ngaysinh, Que, DTB, Hocbong)

Có 2 mức học bổng cho sinh viên là loại A=350.000đ cho các sinh viên có điểm trung bình đạt từ 8.0 trở lên. Loại B= 250.000đ cho các sinh viên có điểm trung bình thuộc vào đoạn mở [7.0,8.0). Để cập nhật dữ liệu cho trường **Hocbong**, ta xây dựng truy vấn như sau:

Bước 1: Chọn bảng **Danhsach** làm dữ liệu nguồn.

Bước 2: Chọn menu Query/Update Query. Kết quả hiện thêm hàng **Update to** trong phần QBE.

Bước 3: Kéo trường **Hocbong** vào hàng Field.

Bước 4: Trong ô **Update To** của trường **Hocbong** đưa vào biểu thức sau: $IIF([DTB] \geq 8.0, 350.000, IIF([DTB] \geq 7.0 \text{ and } [DTB] < 8.0), 250.0000, " ")$.

Bước 5: Ghi và thực hiện truy vấn ta thấy giá của trường **Hocbong** đã được cập nhật.

BÀI TẬP CHƯƠNG 3

Bài 1: Sử dụng cơ sở dữ liệu QUANLYDIEM trong phần bài tập chương 2 xây dựng truy vấn hiện tất cả dữ liệu có trong bảng Solieu.

Bài 2: Sử dụng cơ sở dữ liệu THICU trong phần bài tập chương 2. Thiết kế truy vấn hiện các thông tin: Số báo danh, Họ tên, Địa chỉ, Điểm trung bình 3 môn của tất cả các thí sinh.

Bài 3: Dựa trên cơ sở dữ liệu QUANLYCONGTY trong phần bài tập chương 2, thiết kế truy vấn hiển thị các thông tin Họ tên, địa chỉ, các dự án, tên chi nhánh của một nhân viên trong công ty khi biết mã nhân viên.

Bài 4: Hãy lập bảng báo cáo như sau:

Loại hoá đơn	Giá trị
Yes	300.000
No	423.000

Dựa trên bảng Thuchi của cơ sở dữ liệu THUQUY trong phần bài tập chương 2.

Bài 5: Lập bảng tổng hợp như sau:

Dự án	chi nhánh 1	chi nhánh 2	Chi nhánh n
Dự án A	3	2	...	2
Dự án B	4	0	...	4
Dự án C	1	1	...	1

Dựa trên các bảng trong cơ sở dữ liệu QUANLYCONGTY trong phần bài tập chương 2.

Bài 6: Cho CSDL Quanlyvattu gồm các bảng sau:

VATTU(mavattu, tenvattu, donvitinh, dongia, chungloai)

KHACH(makhach, tenkh, diachikh, dienthoai)

KHO(makho, diachikho)

PHIEUNHAP(sophieu, ngaynhap, makhach, makho)

DONGVATTU(sophieu, mavattu, soluong).

Hãy xây dựng truy vấn trả lời các yêu cầu sau:

1. Xây dựng truy vấn đặt tên là TV1 gồm những trường Sophieu, ngaynhap, tenkhach, makho, diachikho, tenvattu, soluong, dongia, ThanhTien(=soluong*dongia).
2. Xây dựng truy vấn đặt tên là TV2 lấy dữ liệu nguồn là TV1 và chỉ hiển thị những trường sau: Sophieu, ngaynhap, tenkhach, tenvattu, soluong, dongia, ThanhTien. Yêu cầu chỉ hiển thị những phiếu nhập vào 05/2005 và có soluong>100, sắp xếp theo chiều giảm dần của sophieu.
3. Xây dựng truy vấn thông số đặt tên là TV3, cho biết một ngày bất kỳ có những phiếu nhập nào được lập.
4. Xây dựng truy vấn tính tổng đặt tên là TV4, để đưa ra bảng thống kê số lượng khách tại mỗi địa chỉ như sau: Và chỉ hiển thị những địa chỉ có số khách >2.

Địa chỉ	Số lượng khách
---------	----------------

5. Xây dựng truy vấn Crosstab đặt tên là TV5, đưa ra bảng thống kê về chủng loại vật tư như sau:

Số phiếu	Tổng SL vật tư	Chủng loại 1	Chủng loại 2	Chủng loại 3
N100	20	10	4	6
N200	10	0	7	3

6. Xây dựng truy vấn tạo bảng có tên là TV6, dùng để tạo ra bảng có tên là TONGHOP có dữ liệu là toàn bộ TV1 với những bản ghi có Thanhtien>50000.
7. Giả sử có bảng PHIEUNHAP2 có cấu trúc như bảng PHIEUNHAP. Xây dựng truy vấn bổ sung có tên là TV7, dùng để bổ sung dữ liệu trong bảng PHIEUNHAP2 vào cuối của bảng PHIEUNHAP.
8. Xây dựng truy vấn xoá có tên là TV8, dùng để xoá những phiếu nhập có ngày nhập trước 1/1/2000 trong bảng PHIEUNHAP.
9. Xây dựng truy vấn sửa có tên là TV9, dùng để tăng đơn giá của vật tư lên 10%Dongia cũ trong bảng VATTU.

Bài 7: Cho CSDL **SINHVIENT** như sau:

Bảng **HOSO** để lưu hồ sơ các sinh viên gồm các trường:

- MaSV: Text, rộng 10, khoá chính, lưu mã sinh viên.
- Ho: Text, rộng 15, lưu phân họ và tên đệm của SV.
- Ten: Text, rộng 6, lưu phần tên của SV.
- NgaySinh: Date/Time, lưu ngày sinh
- GioiTinh: Text, rộng 3, lưu giới tính là “Nam” hay “Nu”
- Khoa: Text, rộng 20, tên khoa sinh viên học.
- QueQuan: Text, rộng 50, ghi quê quán của SV gồm huyện và tỉnh.
- DiaChi: Text, rộng 50, địa chỉ cư trú hiện tại của SV
- Dienthoai: Text, rộng 12, điện thoại của SV nếu có
- Giaitich, Daiso, Tinhoc, Triethoc, Anhvan: Number, Double, lưu điểm thi các môn học kỳ I năm thứ nhất của SV.

Bảng **GIADINH** để lưu nhưng thông tin người thân trong gia đình có trách nhiệm trực tiếp đối với SV gồm các trường:

- MaSV: Text, rộng 10, mã sinh viên, dùng để liên kết với bảng HOSO.
- Hoten: Text, rộng 25, lưu họ và tên người thân của SV
- Namsinh: Number, long Integer, lưu năm sinh.
- Quanhe: Text, rộng 12, quan hệ với SV.
- Diachi: Text, rộng 50.
- Dienthoai: Text, rộng 12, số điện thoại liên hệ nếu có.

Mối quan hệ giữa bảng **HOSO** và bảng **GIADINH** là 1-n thông qua trường MaSV.

Tạo các truy vấn sau:

1. Query 1: dữ liệu nguồn là bảng HOSO tạo một truy vấn để hiển thị các trường MaSV, Ho, Ten, Khoa, GiaiTich, DaiSo, TinHoc, Triethoc, Anhvan, DiemTB (bảng điểm trung bình của cả 5 môn học). Danh sách sắp xếp theo hai trường: Khoa tăng dần, và DiemTB giảm dần.
2. Query 2: Dùng truy vấn Make Table tạo một bảng mới KhoCNTT gồm tất cả các trường của bảng HOSO nhưng chỉ giữ lại các SV thuộc khoa CNTT

3. Query 3: Từ hai bảng HOSO và GIADINH đưa ra bảng dữ liệu gồm các cột MaSV, Họ tên sinh viên, Họ tên người thân trong gia đình, Quan hệ người thân với SV. Danh sách sắp xếp theo chiều tăng dần của Tên SV.
4. Query 4: Từ bảng HOSO đưa ra danh sách cá SV phải thi lại (một trong các môn dưới 5 điểm) gồm các cột: MaSV, Ho, Ten, Khoa, Giatich, Daiso, Tinhoc, Triethoc, Anhvan. Danh sách sắp xếp theo chiều tăng dần của trường Ten.
5. Query 5: dùng truy vấn thông số đưa ra danh sách các SV có cùng độ tuổi (tuổi nhập vào từ bàn phím) gồm các trường: MaSV, Ho, Ten, Ngaysinh, GioiTinh, Khoa.

CHƯƠNG 4

MẪU BIỂU

Mẫu biểu (Form) dùng để hiển thị và cập nhật dữ liệu. Tổ chức giao diện chương trình. Người ta nói rằng Form chính là “bộ mặt” của các ứng dụng, bởi giao diện người dùng của một ứng dụng hầu như đều thông qua các Form, nên việc thiết kế và thao tác Form là rất quan trọng.

Mẫu biểu gồm các ô điều khiển thuộc các thể loại khác nhau và có công dụng khác nhau. Chẳng hạn điều khiển Label (Nhãn) dùng để thể hiện các dòng văn bản, điều khiển Text Box (hộp văn bản) dùng để thể hiện giá trị các trường, các biểu thức hoặc giá trị gõ từ bàn phím....

Access luôn cho phép kết hợp form và dữ liệu một cách dễ dàng và đơn giản. Đây là một trong những lý do chính tại sao nó là môi trường phát triển ứng dụng nhanh.

Nguồn dữ liệu (nếu có) của mẫu biểu là một bảng hoặc một truy vấn. Khi có nguồn dữ liệu, mẫu biểu dùng để thể hiện và cập nhật dữ liệu cho các trường nguồn. Mẫu biểu không có nguồn dữ liệu (Unbound) thường dùng để tổ chức giao diện chương trình.

1. TẠO MẪU BIỂU BẰNG AUTOFORM VÀ WIZARD

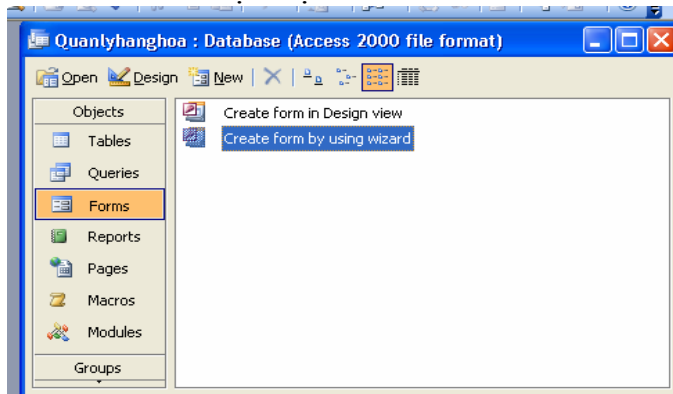
Nếu chỉ dùng biểu mẫu để thể hiện và cập nhật dữ liệu thì cách nhanh nhất là dùng cách tạo tự động hoặc bằng Wizard.

1.1 Tạo mẫu biểu bằng AutoForm

AutoForm giúp chúng ta tự động tạo được một Form nhanh chóng để biểu hiện thị thông tin từ một bảng/truy vấn trong cơ sở dữ liệu.

Các bước thực hiện như sau:

Bước 1: Trong cửa sổ Database chọn mục Forms. Xem hình 4.1



Hình 4.1

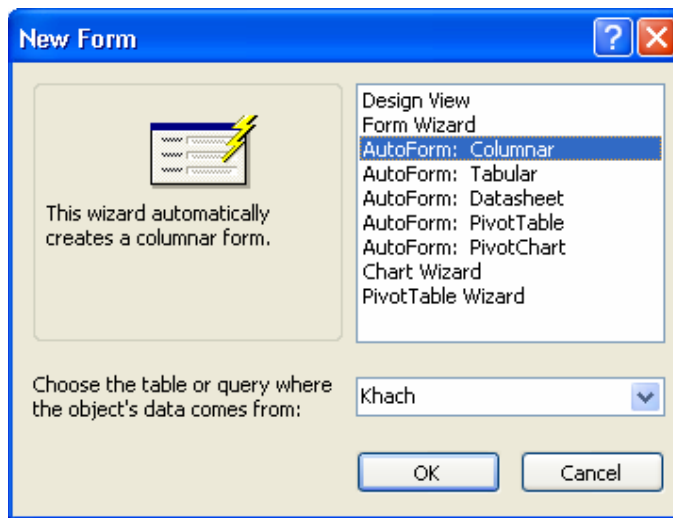
Nút New dùng để tạo mẫu biểu mới.

Nút Design dùng mở cửa sổ thiết kế mẫu biểu

Nút Open dùng để thực hiện mẫu biểu.

Ngoài ra tại khung bên phải ta còn thấy xuất hiện lựa chọn “Create Form in Design view”, “Create form using wizard”, đây là hai lựa chọn thường được sử dụng cho phép chúng ta tạo form “tất” nhanh hơn mà không phải bấm vào New nữa.

Bước 2: Bấm nút New ta nhận được cửa sổ hình 4.2



Hình 4.2

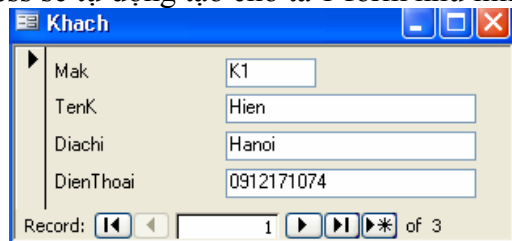
Ta thấy có rất nhiều loại AutoForm, ta có thể chọn một loại tùy ý, nhưng mỗi loại có những thể hiện khác nhau như:

- + Columnar: Hiển thị từng mẫu tin riêng biệt
- + Tabular : Hiển thị nhiều mẫu tin cùng một lúc
- + Datasheet: Hiển thị nhiều mẫu tin cùng một lúc tương tự như một bảng trong chế độ hiển thị Datasheet.

Giả sử ta chọn AutoForm: Columnar.

Bước 3: Chọn bảng/truy vấn làm dữ liệu nguồn cho Form. Có thể không chọn nếu Form đó không có dữ liệu nguồn.

Bước 4: Bấm OK Access sẽ tự động tạo cho ta 1 form như hình 4.3



Hình 4.3

Bước 5: Đặt tên cho mẫu biểu

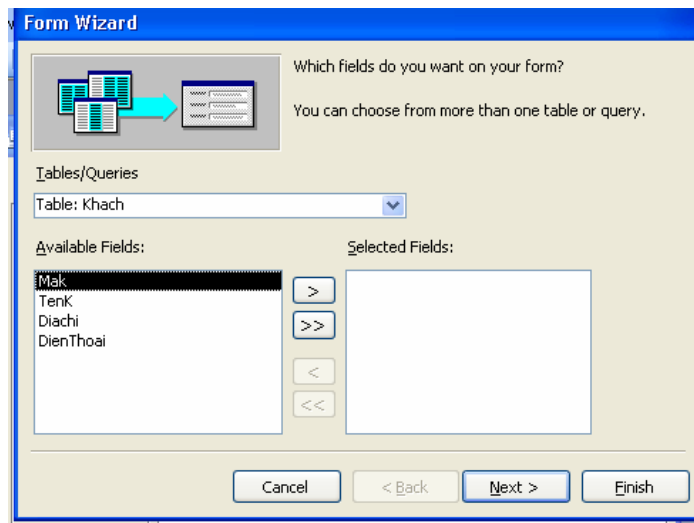
1.2 Tạo mẫu biểu bằng Form Wizard

Tạo Form bằng Wizard là tạo Form bằng cách làm theo từng bước do Access qui định sẵn. Các bước thực hiện như sau:

Bước 1: Trong cửa sổ Database chọn mục Forms. sau đó ta có thể chọn luôn “Create form using wizard” bên phải. Xem hình 4.4.

Bước 2: Chọn bảng làm dữ liệu nguồn cho Form (trong ví dụ này chọn bảng **Khach**). Chọn các trường muốn đưa vào mẫu biểu tại cột Available Field, rồi bấm vào nút > để đưa sang cột Selected Fields. Nếu muốn đưa sang tất cả các trường ta bấm nút >>. Nếu muốn bỏ trường nào đã chọn trong cột Select Field thì ta bấm <, muốn bỏ tất cả bấm vào nút: <<. Bấm nút **Next** để tiếp tục.

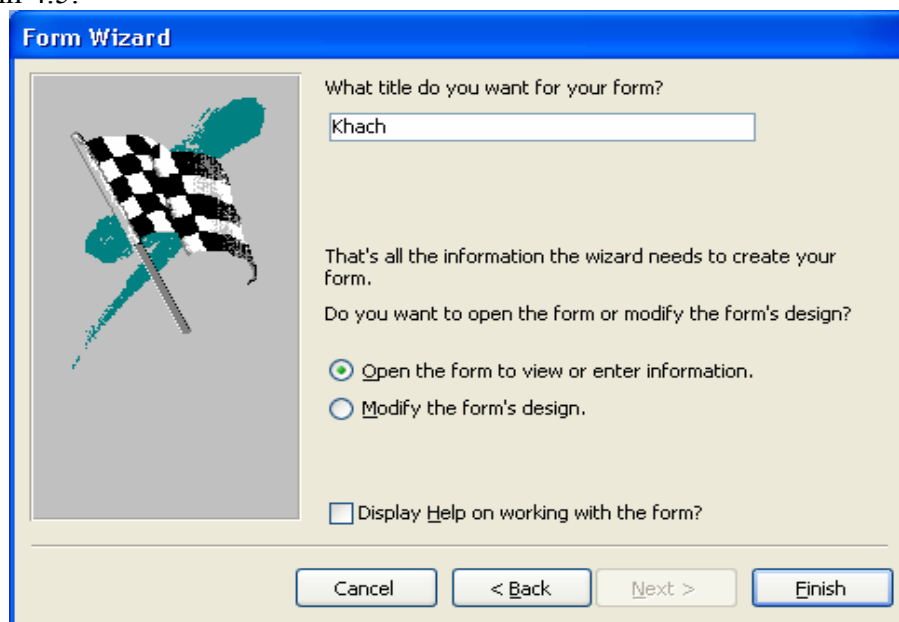
Bước 3: Chọn dạng thể hiện của Form. Cửa sổ cho phép chọn một trong 4 dạng thể hiện là **Columnar**, **Tabular**, **Datasheet**, **Justified**, **PivotTable**, **PivotChart**. (trong ví dụ này chọn Columnar). Tiếp tục bấm **Next**.



Hình 4.4

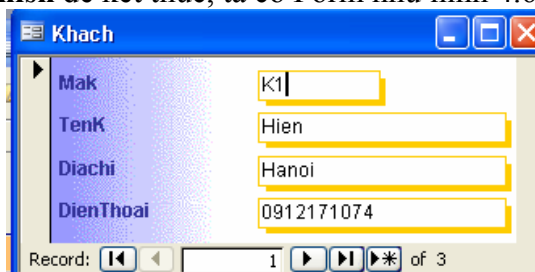
Bước 4: Chọn một trong các dạng trình bày mẫu biểu trong cửa sổ. Giả sử ở đây ta chọn kiểu **Blends**. Sau đó bấm **Next**.

Bước 5: Đặt tên cho biểu mẫu tại hộp màu trắng. Access dùng tên bảng/truy vấn nguồn làm tên mẫu biểu mặc định. Ta có thể giữ nguyên hoặc thay đổi theo ý muốn. Xem hình 4.5.



Hình 4.5

Bước 6: Bấm phím **Finish** để kết thúc, ta có Form như hình 4.6



Hình 4.6

Nhận xét: Trong cả hai cách tạo Form là Auto và Wizard đều cho phép tạo mẫu biểu dưới dạng đồ thị (GRAPH), khi đó ta chọn PivotChart. Sau khi sử dụng Auto hoặc Wizard ta có thể mở Form ở chế độ Design để có thể thiết kế thêm theo ý mình.

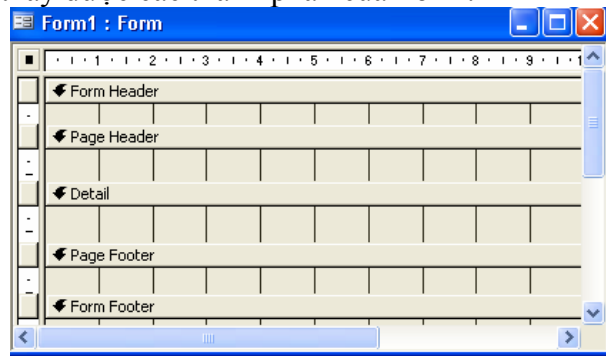
2. THIẾT KẾ FORM Ở CHẾ ĐỘ DESIGN

2.1 Các thành phần trong một Form

Form (mẫu biểu) gồm 5 thành phần:

- Đầu biểu (Form Header)
- Đầu trang (Page Header)
- Thân (Detail)
- Cuối trang (Page Footer)
- Cuối biểu (Form Footer)

Xem hình 4.7 ta sẽ thấy được các thành phần của Form:



Hình 4.7

Để hiện(ẩn) các thành phần đầu/cuối biểu hoặc đầu /cuối trang ta chọn menu View, Form Header/Footer hoặc Page Header/Footer.

Ý nghĩa:

- Đầu biểu, Cuối biểu (Form Header / Form Footer):
 - + Dùng để trình bày tiêu đề, hướng dẫn sử dụng các nút lệnh
 - + Có thể không in đầu biểu bằng cách đặt giá trị Screen Only vào thuộc tính Display when1
 - + Có thể không cho hiển thị đầu biểu (khi thực hiện Form) bằng thuộc tính: Visible: No
- Đầu trang, cuối trang (Page Header, Page Footer): chỉ xuất hiện khi in
- Thân biểu (Detail): Chứa các ô điều khiển buộc vào các trường nguồn.

Thuộc tính của mẫu biểu và các thành phần của nó:

Từng thành phần (trong 5 thành phần của mẫu biểu) có bảng thuộc tính riêng. Mẫu biểu cũng có bảng thuộc tính. Muốn mở bảng thuộc tính của đối tượng nào ta làm như sau:

- + Đầu tiên chọn đối tượng cần mở bảng thuộc tính
- + Sau đó: hoặc chọn Properties trong menu View hoặc bấm nút phải chuột chọn Properties.

2.2 Các bước thiết kế Form

2.2.1 Các bước thiết kế

Bước 1: Tại cửa sổ Database chọn Forms và bấm nút New, chọn Design View. Tại đây ta có thể chọn bảng/truy vấn làm dữ liệu nguồn cho Form.

Bước 2: Thiết kế nội dung của Form nhờ hộp công cụ. Tại đây ta có thể đưa vào Form những điều khiển, căn chỉnh, trang trí Form,...

Bước 3: Ghi và đặt tên cho Form.

Các loại điều khiển thông dụng

Hộp căn bản (Text Box): Dùng để nhập và hiển thị dữ liệu

Nhãn (Label): Dùng để viết tiêu đề của Form và của các hộp văn bản

Hộp lựa chọn (Combo box): Dùng để lựa chọn dữ liệu có sẵn

Hộp danh sách (List Box): Dùng chọn dữ liệu có sẵn trong một danh sách

Nút lệnh (Command Button): Dùng để thực thi một lệnh nào đó

Nhóm lựa chọn (Option Group): Thể hiện một nhóm các lựa chọn

Cách dùng các điều khiển:

+ Bound: Điều khiển buộc vào một trường, dùng để truy xuất tới trường đó

+ Unbound: Điều khiển không buộc, không liên quan đến trường nào cả, dùng để: Nhập dữ liệu từ bàn phím, Trình bày tiêu đề, trang trí hình vẽ, hộp đường ...

+ Calculated: điều khiển tính toán, dùng để thể hiện giá trị của một biểu thức

Cách tạo một điều khiển

- Chọn loại điều khiển: Bấm vào nút tương ứng trên hộp công cụ.

- Di chuyển đến vị trí cần đặt điều khiển vào biểu mẫu và bấm chuột.

- Thực hiện các sửa đổi cần thiết.

2.2.2 Chọn đồng thời các ô điều khiển

Có thể chọn đồng thời các ô điều khiển theo các cách sau:

+ Kết hợp giữa phím Shift và chuột

+ Bấm chuột tại một vị trí trên thước dọc mép trái mẫu biểu để chọn các điều khiển trên cùng một hàng chứa vị trí bấm.

+ Bấm chuột một vị trí trên thước ngang (mép trên biểu mẫu) để chọn các điều khiển trên cùng một cột chứa vị trí bấm.

+ Có thể dùng phím Shift và cách bấm trên thước để chọn đồng thời các điều khiển nằm trên nhiều hàng, nhiều cột.

+ Để chọn các điều khiển nằm trên các hàng liên tiếp, ta di chuột tới hàng đầu tại thước dọc, rồi kéo chuột theo thước tới hàng cuối.

+ Để chọn các điều khiển nằm trên các cột liên tiếp, ta di chuột tới cột đầu tại thước ngang, rồi kéo chuột theo thước tới cột cuối.

2.2.3 Thuộc tính của điều khiển

Để mở bảng thuộc tính của một ô điều khiển ta làm như sau: Chọn điều khiển cần mở bảng thuộc tính. Chọn View/Properties, hoặc bấm đúp chuột tại điều khiển, hoặc bấm nút phải chuột rồi chọn Properties.

Bảng thuộc tính có dạng sau:

Mỗi loại ô điều khiển có thể có một số thuộc tính khác nhau. Tại đây trình bày một số thuộc tính hay dùng.

Name: tên điều khiển

Caption: tiêu đề

Control Source: nguồn dữ liệu

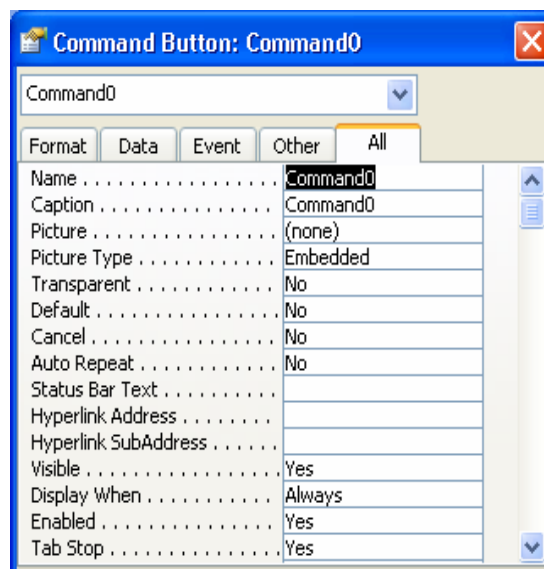
Format: quy định hình thức hiển thị

Default Value: giá trị mặc định

Validation Value: quy tắc dữ liệu hợp lệ

Validation Text: văn bản hợp lệ

Input Mask: mặt nạ nhập liệu



Status Bar Text: hướng dẫn sử dụng điều khiển (hướng dẫn này sẽ hiện tại dòng trạng thái)

Để di chuyển các ô/nhóm điều khiển ta chọn ô/nhóm cần di chuyển, sau đó kéo đến vị trí mới.

2.2.4 Định vị các ô điều khiển

Có thể định vị các ô theo các khả năng sau:

- + Gióng thẳng theo mép trái của điều khiển đứng bên trái nhất (căn trái)
- + Gióng thẳng theo mép phải của điều khiển đứng bên phải nhất (căn phải)
- + Gióng thẳng theo mép trên của điều khiển đứng bên trên nhất (căn trên)
- + Gióng thẳng theo mép dưới của điều khiển đứng bên dưới nhất (căn dưới)

Thao tác như sau:

Chọn các ô điều khiển cần định vị, chọn Form/Align và chọn:

- + Left để căn trái
- + Right để căn phải
- + Top để căn trên
- + Bottom để căn dưới

Cách đều các ô điều khiển

- + Cách đều theo hàng: Chọn Format/ Horizontal Spacing/ Make Equal

Kết quả các ô đầu hàng và cuối hàng không thay đổi vị trí. Các ô khác được dịch chuyển sao cho khoảng cách giữa hai ô trong hàng bằng nhau.

- + Cách đều theo cột: Chọn Format/Vertical Spacing/ Make Equal

Kết quả các ô đầu cột và cuối cột không thay đổi vị trí. Các ô khác được dịch chuyển sao cho khoảng cách giữa hai ô trong cột bằng nhau.

- + Tăng, giảm khoảng cách giữa các điều khiển trong hàng:

Chọn Format, Horizontal Spacing

Sau đó chọn Increase để tăng hoặc chọn Decrease để giảm.

2.2.5 Thay đổi kích thước ô điều khiển

Chọn các ô cần thay đổi kích thước. Chọn Format/Size

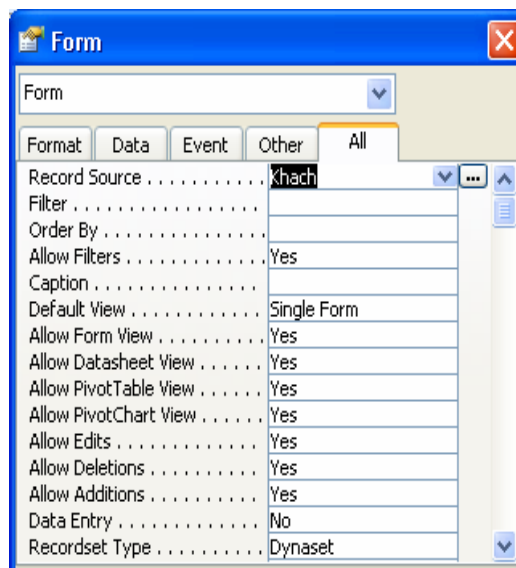
- To Fit: các ô sẽ có kích thước vừa đủ thể hiện nội dung trong nó.
- To Tallest: các ô sẽ có chiều cao bằng chiều cao của ô cao nhất.
- To Shortest: các ô sẽ có chiều cao bằng chiều cao của ô thấp nhất.
- To Widest: các ô sẽ có chiều rộng bằng chiều rộng của ô rộng nhất.
- To Narrowest: các ô sẽ có chiều rộng bằng chiều rộng của ô hẹp nhất.

Ta có thay đổi Font và cỡ chữ cho một nhóm ô điều khiển bằng cách chọn những Font chữ, cỡ chữ cần thay đổi.

2.2.6 Thuộc tính của Form

Trong chế độ Design View nhấp chuột vào nút chọn Form (ô vuông góc bên trái của Form), sau đó chọn View/Properties. Hoặc có thể nhấp đúp chuột vào nút chọn Form. Sau đó sẽ hiển thị hộp thoại Form, chọn All, hộp thoại hiện tất cả 107 thuộc tính của Form như sau:

Sau đây là một số thuộc tính hay sử dụng:



- **Record Source:** lựa chọn bảng/truy vấn nguồn(bảng/truy vấn có các trường cần đặt lên Form).
- **Default View:** lựa chọn một trong các cách thể hiện của Form
- **Scroll Bars:** Bật/tắt các thanh cuộn Form
- **Record Selections:** bật/tắt công cụ chọn bản ghi phía trái Form
- **Navigation Buttons:** bật/tắt các nút di chuyển và thêm bản ghi ở phía dưới Form
- **Dividing Lines:** bật/tắt các đường lưới ngăn cách các thành phần của Form
- **Control Box:** Bật/tắt các nút Min, Max, Close điều khiển trên Form.

Ví dụ: Giả sử tạo Form để nhập dữ liệu vào bảng **Khach** trong CSDL **Quanlyhanghoa**.

Bước 1: Chọn phần Forms trên cửa sổ Database, nhấp nút New chọn Design view, chọn tên bảng dữ liệu nguồn là Khach, nhấp Ok. Xuất hiện cửa sổ thiết kế Form còn trống.

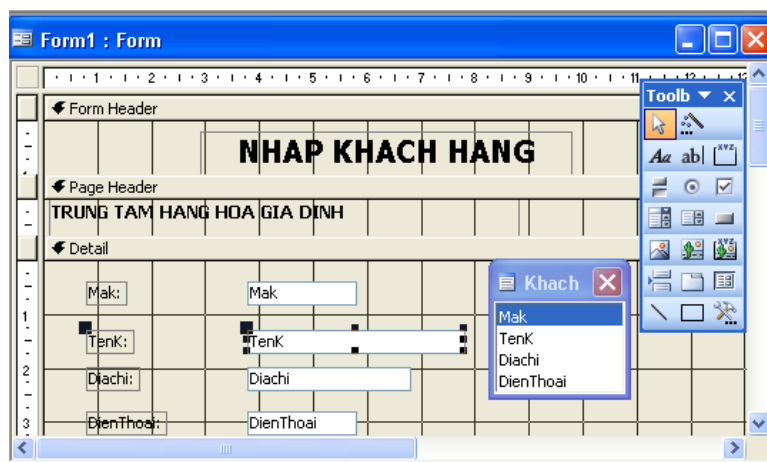
Bước 2: Dùng lệnh View/Form Header-Footer để hiển thị đầu Form và cuối Form. View/Page Header-Footer để hiển thị đầu trang và cuối trang

Bước 3: Dùng lệnh View/Field list và View/Toolbox để hiển thị danh sách các trường của bảng Khach và hộp công cụ.

Bước 4: Lần lượt đưa các trường vào Form, bằng cách kéo tên trường từ danh sách các trường vào Form. Mỗi trường đưa vào Form sẽ tạo nên hai điều khiển liên kết nhau: Điều khiển bên phải là một Text Box(tại vị trí này sẽ hiển thị dữ liệu của trường ứng với một bản ghi), điều khiển thứ hai là một Label-nhãn(Access mặc định lấy tên trường làm nhãn, ta có thể thay đổi tên trường cho thân thiện hơn).

- Kéo các trường Mak, TenK, Diachi, dienthoai vào Form
- Nhấp vào nút Label trên hộp Toolbox, khoanh vùng chữ nhật dải Form Header để xác định một nhãn và nhập xâu ký tự “NHAP KHACH HANG”.
- Tương tự trong dải Page Header tạo một nhãn “TRUNG TAM HANG HOA GIA DINH”.

Bước 5: Ghi thiết kế Form: bấm File/Save hoặc bấm vào nút đóng cửa sổ



Hình 4.8

3. HỘP CÔNG CỤ TOOLBOX

Hộp công cụ là phương tiện chính để xây dựng các mẫu biểu. Mỗi nút trên hộp công cụ ứng với một loại điều khiển và dùng để tạo các ô điều khiển thuộc loại tương ứng.



Hộp công cụ có các nút như sau:

Select Objects(dùng để lựa chọn các điều khiển trên vùng chữ nhật), Control Wizard, Label(tạo nhãn), Text Box, Option Group, Toggle Button, Check Box, List Box, Command Button, Image(tạo đối tượng hình ảnh), Unbound Object Frame, Bound Object Frame, Page Break(tạo ngắt trang), Tab Cotrl, Subform/SubReport, Line(tạo đường thẳng), Rectangle (tạo khung chữ nhật), More Controls(liệt kê nhiều loại điều khiển khác có thể tạo).

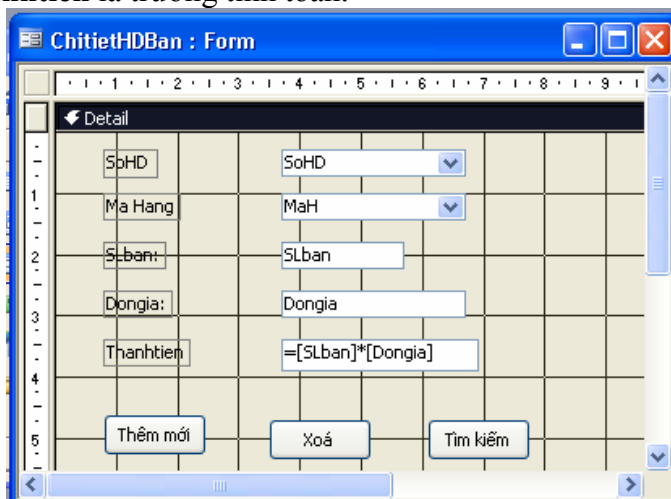
Sau đây chúng ta trình bày một số điều khiển thường dùng:

3.1 Điều khiển Text box

Điều khiển Text box dùng để:

- Hiện thị dữ liệu của các trường kiểu văn bản, số, và kiểu memo. Cũng có thể dùng để nhập dữ liệu từ bàn phím vào.
- Dùng để tạo một trường tính toán trên Form:

Ví dụ: Trong Form nhập dữ liệu cho bảng **ChitietHD** của CSDL **Quanlyhanghoa** có dạng như hình 4.9. Có nút **SLban**, **Dongia** là một Text box, **SoHD**, **MaH** là một Combo box, **ThanhTien** là trường tính toán.



Hình 4.9

- **Cách Text box SLban.** Kéo trường SLban từ Field List đặt lên Form, máy tự động tạo một Text Box có tên là **SLban** và thuộc tính nguồn dữ liệu Control Source là **SLban**, nhấn liên kết với Text box này có thuộc tính Caption là “SLban”.

- **Cũng tương tự ta tạo được Text Box Dongia.**

- **Tạo Text Box ThanhTien:**

Nháy nút Text Box trên thanh Toolbox, nháy và kéo trên Form để xác định vùng chữ nhật dùng làm Text Box.

Bấm View/Properties hiện thuộc tính của Text Box hoặc nháy trực tiếp vào ô này nhập các thuộc tính sau:

Control Source: = [SLban]*[Dongia]

Name: ThanhTien.

Mở thuộc tính của Label liên kết với Text Box, dòng thuộc tính Caption nhập “ThanhTien”, hoặc nháy chuột gõ trực tiếp tại Label.

3.2 Điều khiển Combo Box

Combo Box và List Box có thể dùng để tạo danh sách lựa chọn cho một trường bất kỳ. Danh sách trong List Box luôn được thể hiện, còn trong Combo Box chỉ được mở khi kích chuột tại nút mũi tên bên phải.

Mỗi một Combo Box đều phải có một nguồn dữ liệu để xây dựng bảng chọn. Nguồn dữ có thể là:

1. Các bản ghi của một bảng/truy vấn
2. Dãy các giá trị đưa vào khi tạo Combo Box (value list)
3. Tên các trường của bảng/truy vấn (field list)
4. Các bản ghi của một truy vấn tạo bởi câu lệnh SELECT
5. Danh sách tạo từ kết quả trả về của một hàm trong Access Basic.

Tương ứng ta xét ba kiểu Combo Box thường dùng đó là:

- Combo Box có giá trị lấy từ các trường của một bảng (ví dụ như Combo MaH lấy dữ liệu từ trường MaH của bảng Hang trong VD trên)
- Combo Box có giá trị gõ vào.
- Combo Box để tìm kiếm.

Khi tạo Combo Box ta có thể dùng Wizard hoặc không, sau đây sẽ trình bày hai cách tạo này.

3.2.1 Tạo Combo Box dùng wizard

Công cụ Wizard có thể tạo được 2 loại Combo Box sau: đó là Combo Box có nguồn dữ liệu là một bảng/truy vấn hoặc là một dãy giá trị đặt vào thuộc tính Row Source, hoặc xây dựng Combo cho phép tìm kiếm.

Cách tạo như sau:

Bước 1: Chọn chức năng Control Wizard bằng cách bấm vào biểu tượng Control Wizard trên thanh công cụ (nếu nó chưa bật).

Bước 2: Chọn nút Combo Box trên thanh công cụ

Bước 3: kéo một trường của bảng/truy vấn nguồn vào mẫu biểu. Access sẽ tạo một Combo Box buộc vào trường này.

Bước 4: Lần lượt trả lời các câu hỏi do Wizard đặt ra để xác định nguồn dữ liệu và các đặc trưng khác của Combo Box. Sau khi trả lời một câu hỏi bấm Next để sang câu hỏi tiếp theo, cuối cùng bấm Finish.

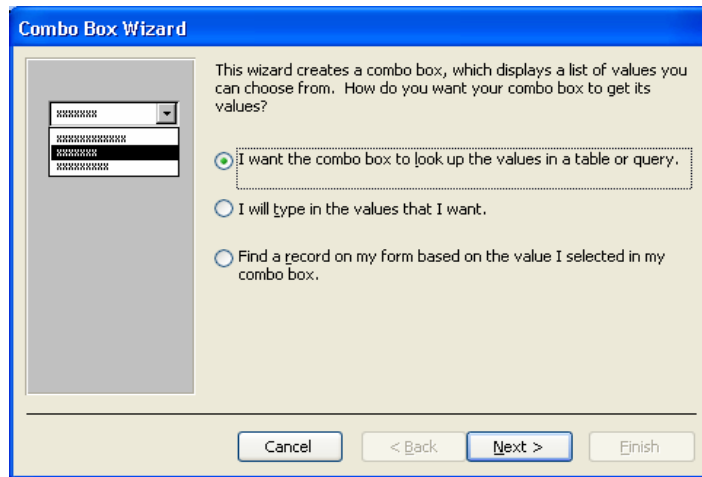
Ví dụ 1: Tạo Combo có các giá trị lấy từ một bảng/truy vấn:

Giả sử tạo nút Combo Box **MaH** trong ví dụ trên bằng cách dùng Wizard. Dữ liệu nguồn của Combo Box **MaH** là cột **MaH**, **TenH** của bảng **Hang** nhưng chỉ buộc vào **MaH**, Giả sử đang ở màn hình thiết kế Form (nguồn dữ liệu là bảng **ChitietHD**). Ta thực hiện như sau:

Bước 1: Bật chức năng Control Wizard

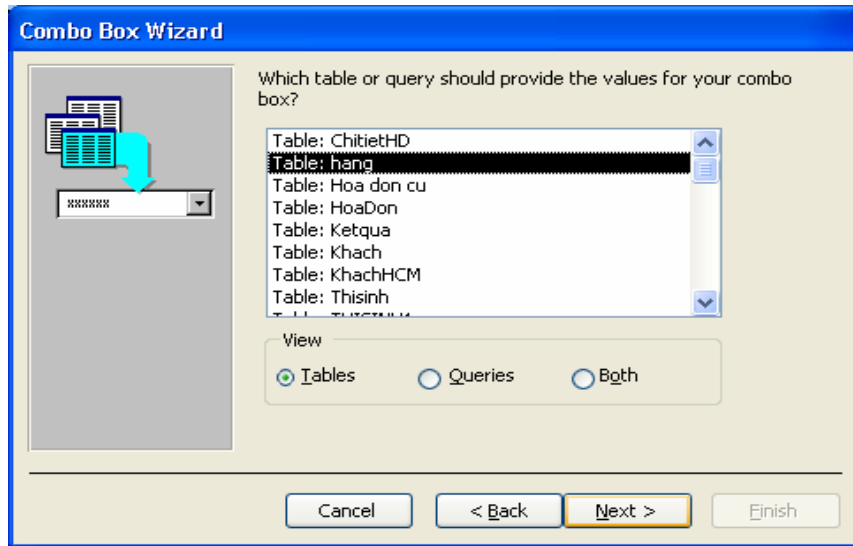
Bước 2: chọn Combo Box trên hộp công cụ

Bước 3: Kéo trường [MaH] từ hộp Field List vào Form. kết quả sẽ hiện ra trong hình 4.8



Hình 4.8

Bước 4. Chọn lựa chọn thứ nhất, rồi bấm Next, sẽ hiện ra danh sách các bảng/truy vấn nguồn của Combo Box. Hình 4.9



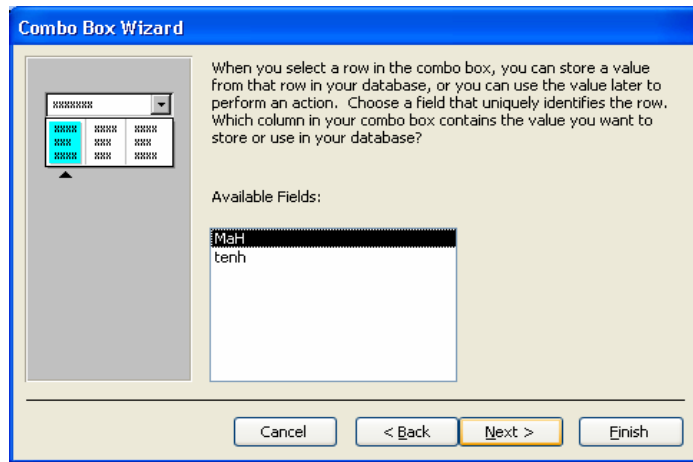
Hình 4.9

Bước 5. Chọn bảng Hang làm nguồn dữ liệu, tiếp tục bấm Next. Kết quả sẽ hiện ra danh sách các trường của bảng/truy vấn vừa chọn.

Bước 6. Chọn các trường đưa vào danh sách chọn của Combo Box. Ở đây ta chọn hai trường MaH, TenH, tiếp tục bấm Next.

Bước 7. Access hiển thị ra màn hình cho phép lựa chọn cách sắp xếp trong các trường theo chiều tăng dần hay giảm dần. Ở đây sắp xếp theo chiều tăng dần của TenH, tiếp tục bấm Next.

Bước 8. Bước này ta dùng chuột để điều chỉnh độ rộng các trường. Hoặc có thể bấm chọn “Hide key column” để ẩn đi trường khoá không xuất hiện trong bảng chọn. Trong ví dụ này ta kéo cho giữ nguyên cả hai cột. Kết quả như hình 4.10



Hình 4.10

Bước 9. Theo thiết kế Combo Box sẽ có hai cột. Bây giờ cần chọn cột để lấy giá trị đưa vào trường gắn với Combo Box. Ta chọn trường MaH, tiếp tục bấm Next.

Bước 10. Xác định trường gắn với Combo Box. Vì ta chủ động kéo trường MaH của bảng ChitietHD vào Form, nên mặc định trường này được chọn. Tiếp tục bấm Next

Bước 11. Sửa đổi tiêu đề của nhãn đi kèm với Combo Box, rồi bấm Finish. Kết quả tạo được Combo Box MaH.

Khi mở Form để chạy ta có được kết quả như hình 4.11



Hình 4.11

Ví dụ 2: Tạo Combo Box có các giá trị gõ vào

Giả sử có CSDL tên là **QuanlyGiaovien**, gồm các bảng sau:

DSGV(MaGV,Hoten, Matinh, chucvu, gioitinh, hocvi).

DSTINH(Matinh, tentinh)

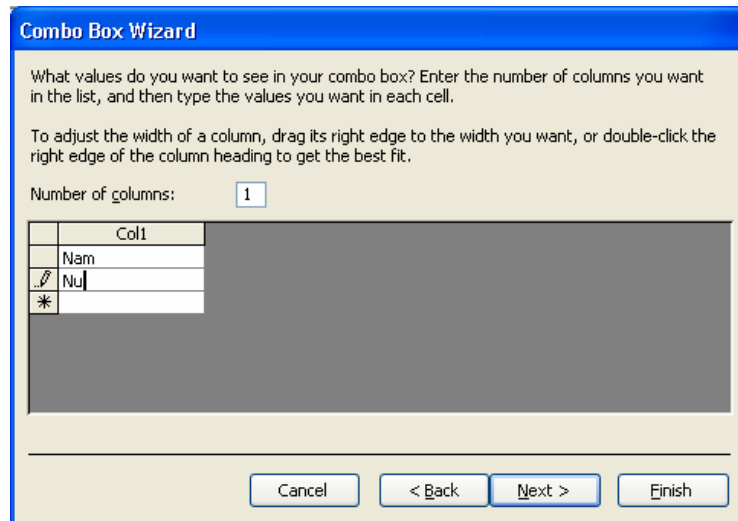
Ta muốn tổ chức nhập số liệu cho bảng DSGV.

Với trường **GioiTinh**: Yêu cầu là một Combo Box có 2 giá trị chọn là Nam, Nu.

Ta thực hiện như sau:

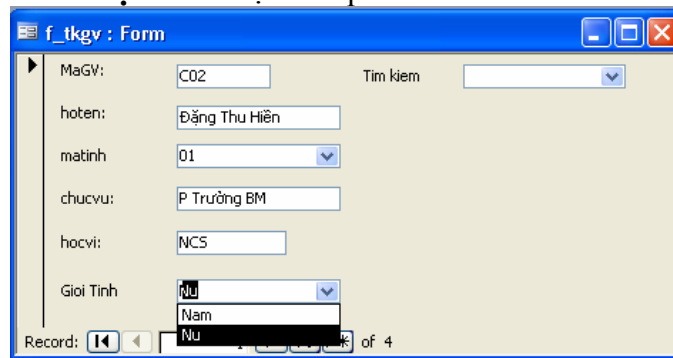
Bước 1,2,3 như trong ví dụ 1.

Bước 4: chọn nút thứ 2 “I will type in the values that I want” (tôi sẽ gõ các giá trị tôi muốn), nhấn nút Next. Kết quả như hình 4.12



Hình 4.12

Bước 5: Xuất hiện hộp thoại, trong cột Col1 cho phép gõ vào các giá trị mà sau này sẽ hiển thị trên Combo Box. Ta gõ (Nam, Nu), Bấm Next. Tiếp theo sẽ đến các lựa chọn như **Bước 10, 11** của **ví dụ 1**. Ta được kết quả như hình 4.12



Hình 4.12

Ví dụ 3: Tạo Combo Box tìm kiếm bản ghi

Trong Form hình 4.12 ta thấy có một Combo Box nằm ở góc bên phải dùng để tìm kiếm một giảng viên theo **MaGV** và **họ tên GV**. Khi chọn một GV nào đó trong Combo Box tìm kiếm thì nội dung bản ghi tương ứng của GV này sẽ hiển thị trên Form.

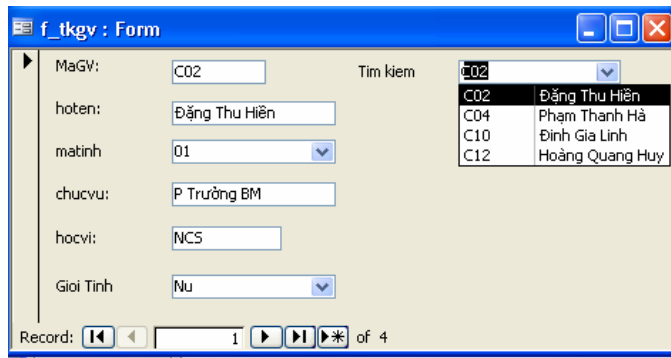
Các bước thực hiện:

Bước 1,2 như trong ví dụ 1, riêng **bước 3** không kéo các trường vì tạo Combo Box không buộc vào trường nào.

Bước 4: chọn nút thứ 3 “Find a record om my...”, nhảy nút Next.

Tiếp tục cho phép ta chọn các trường dùng để tìm kiếm (như **bước 6** trong ví dụ 1). Ở đây ta chọn 2 trường là **MaGV, Hoten**, nhảy nút Next.

Tiếp tục cho phép điều chỉnh độ rộng, ẩn các trường trong Combo Box (như **bước 8** VD1). Tại đây ta để hiển thị cả 2 trường. Nháy nút Next. Tiếp đến sửa lại nhãn cho Combo Box (như **bước 11** trong VD1). Ta được kết quả như hình 4.13



Hình 4.13

3.2.2 Tạo Combo Box không dùng wizard

- Để tạo Combo Box không dùng wizard. Trước hết ta phải tắt Control Wizard.
- Sử dụng các thuộc tính sau để xác định nguồn dữ liệu của Combo Box.

Loại (nguồn dữ liệu)	Thuộc tính Row Rource type	Thuộc tính Row Source
Các bản ghi của một bảng/truy vấn	Table/query	Tên bảng/truy vấn
Các bản ghi từ câu lệnh Select	Table/query	Câu Lệnh SELECT
Danh sách giá trị tự đặt	Value list	Danh sách các giá trị cách nhau bởi dấu phẩy.
Tên các trường của một bảng/truy vấn	Field List	Tên bảng truy vấn
Các giá trị trả về của một hàm trong Access Basic	Tên hàm đó	Để trống

Ngoài hai thuộc tính quan trọng là Row Rource type, Row Source dùng để xác định nguồn dữ liệu cho Combo Box. Chúng ta cần sử dụng thêm một số thuộc tính sau:

- **Control Source:** dùng để xác định trường gắn với Combo Box của bảng dữ liệu nguồn (đối với những Combo Box dùng để tìm kiếm thì thuộc tính này bỏ trống).
- **Column Count:** Số cột trong danh sách, tính từ trái sang phải. VD: Nếu đặt giá trị của thuộc tính này là 2, thì bảng chọn sẽ gồm 2 cột chứa giá trị hai trường đầu của nguồn dữ liệu.
- **Bound Column:** Cột được chọn làm nguồn dữ liệu của danh sách. VD: Nếu đặt là 2 thì giá trị cột 2 sẽ được chọn, các cột khác chỉ đóng vai trò hiển thị.
- **Column widths:** 2;1.5(độ rộng các cột, nếu viết như trên thì độ rộng cột một là 2cm, cột hai là 1.5cm). Nếu cho độ rộng bằng 0 thì cột không hiển thị.
- **Limit To List:** No/Yes(Yes: chỉ chọn trong danh sách, No: có thể nhập giá trị ngoài danh sách)
- **Width:** 3.5cm(bề rộng của danh sách=tổng bề rộng các cột)
- **Hieght:** 5cm(chiều cao của danh sách. Danh sách có thể nhiều hàng nhưng chỉ cho hiện ra các hàng trong phạm vi 5cm).

Ví dụ 3:

Giả sử có CSDL tên là **QuanlyGiaovien**, gồm các bảng sau:

DSGV(MaGV,Hoten, Matinh, chucvu, gioitinh, hocvi).

DSTINH(Matinh, tentinh)

Ta muốn tổ chức nhập số liệu cho bảng DSGV như Form hình 4.13 không dùng Wizard.

Trong Form này gồm:

- Trường **MaGV, Hoten, chucvu, hocvi** gõ trực tiếp từ bàn phím
- Trường **Matinh** là một Combo Box, với dữ liệu nguồn từ bảng **DSTINH**

- Trường **Gioitinh** chọn từ một Combo box với hai giá trị **Nam, Nu**
 Sau đây chỉ trình bày cách xây dựng hai Combo Box.
 Giả sử đang ở màn hình thiết kế Form với bảng dữ liệu nguồn là DSGV.

Combo Box MaTinh:

Bước 1: Bấm nút Combo Box trên thanh ToolBox kéo và thả vào Form

Bước 2: Nháy chuột phải chọn Properties để mở bảng thuộc tính của Combo Box.

Xác định giá trị các thuộc tính như sau:

Name: đặt tên là CBTinh

Control Source: Matinh

Row Source type: Table/query

Row Source: DSTinh

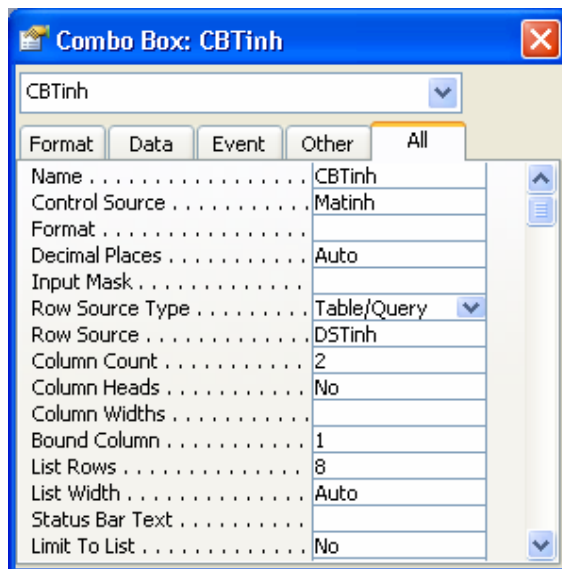
Column Count: 2, hiển thị cả 2 cột Matinh, TenTinh

Bound Column: 1. Buộc vào cột Matinh

Column widths: để trống thì Access sẽ lấy bằng độ rộng của trường đó.

Limit To List: No

Bước 4: nháy chuột chọn vào nhãn đi kèm để đặt tên nhãn là: MaTinh



Combo Box Gioitinh

Bước 1: Bấm nút Combo Box trên thanh ToolBox kéo và thả vào Form

Bước 2: Nháy chuột phải chọn Properties để mở bảng thuộc tính của Combo Box.

Xác định giá trị các thuộc tính như sau:

Name: đặt tên là CBGioitinh

Control Source: Gioitinh

Row Source type: Values List

Row Source: Nam;Nu

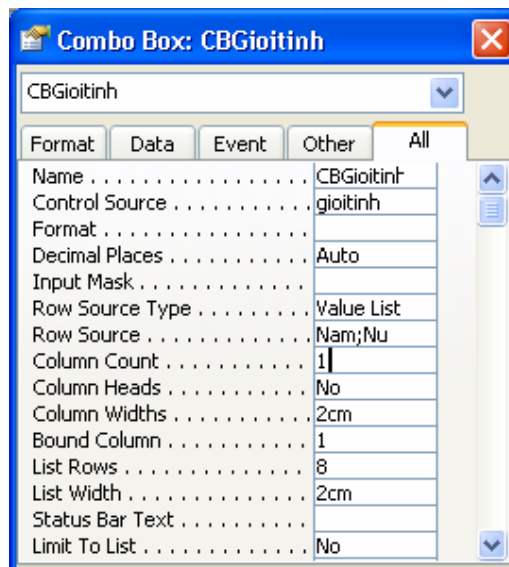
Column Count: 1

Bound Column: 1

Column widths: 2cm

Limit To List: No

Bước 4: nháy chuột chọn vào nhãn đi kèm để đặt tên nhãn là: Gioitinh



3.3 Điều khiển Command Button

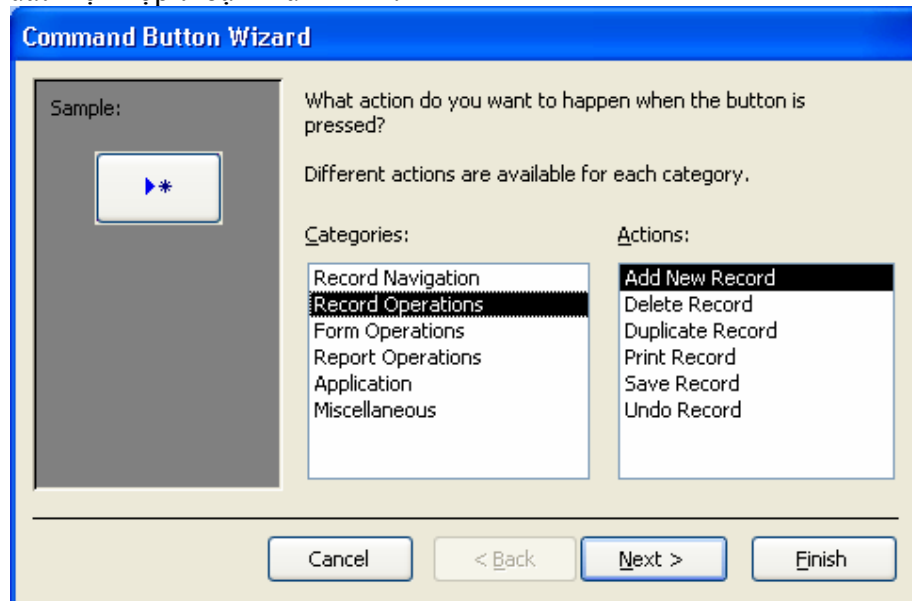
Nút lệnh (Command Button) là các nút nằm trên Form, khi ta nháy chuột lên nút lệnh này máy sẽ thực hiện một công việc nào đó. Trên Form ở Hình 4.11 có ba nút lệnh “**Thêm mới**” dùng để bổ xung bản ghi mới, “**Xóa**” dùng để xóa bản ghi hiện hành, “**Tìm kiếm**” dùng để tìm kiếm bản ghi theo trường hiện tại.

Khi tạo các nút lệnh ta cũng có thể dùng Wizard hoặc không dùng Wizard. Sau đây trình bày cách tạo bằng Wizard.

Quá trình tạo nút lệnh “Thêm mới”:

Bước 1: Nháy chuột vào nút Control Wizards.

Bước 2: Nháy chuột vào nút Command Button trên thanh Toolbox và kéo và đặt vào Form. Xuất hiện hộp thoại như hình 4.14



Hình 4.14

Bước 3: Trong khung **Categories** chọn “**Record Operations**”(các thao tác về bản ghi), trong khung **Actions** chọn “**Add New Record**”, nháy nút Next.

Bước 4: Tại đây ta chọn mục Text, nhập văn bản sẽ hiển thị trên nút lệnh là “**Thêm mới**” thay cho văn bản ngầm định “**Add Record**” (ta cũng có thể chọn một biểu tượng đồ họa hiện trên nút lệnh), nháy nút Next.

Bước 5: Đặt tên cho nút lệnh là Themmoi thay cho tên ngầm định, nháy nút Finish.

Ta thấy trong hình 4.14 cho phép ta lựa chọn rất nhiều hành động mà nút lệnh có thể thực hiện:

- Nhóm **Record Navigation** (di chuyển bản ghi có các chức năng): **Find next, Find Record, Go to first record, Go to last record, Go to next record, Go to previous record.**

- Nhóm **Record Operations** (các thao tác trên bản ghi): **Add new record, Delete record, Duplicate record, Print record, Save record, Undo record.**

- Nhóm **Form Operations** (các thao tác về mẫu biểu): **Apply Form Filter, Close Form, Edit Form Filter, Open Form, Open Page, Print a Form, Print Current Form, Refresh Form Data.**

- Nhóm **Report Operations** (các thao tác về báo cáo): **Mail report, Preview report, Print report, Sent report.**

- Nhóm **Application** (chương trình ứng dụng): **Quit Application, Run Application, Run MS Excel, Run MS Word.**

- Nhóm **Miscellaneous** (hỗn hợp): **Auto Dialer, Print Table, Run Macro, Run Query.**

Như vậy việc tạo nút “**Xoá**” cũng tương tự, tại bước 3 ta chọn như sau: trong khung **Categories** chọn “**Record Operations**”, trong khung **Actions** chọn “**Delete Record**”.

Việc tạo nút “**Tim kiếm**” tại bước 3 ta chọn như sau: khung **Categories** chọn “**Record Navigation**”, trong khung **Actions** chọn “**Find Record**”.

3.4 Thiết kế dạng Main – Sub Form

Phần này trình bày cách tổ chức nhập, hiển thị dữ liệu đồng thời trên nhiều bảng có quan hệ với nhau, trong đó có một bảng chính và mỗi bảng phụ là quan hệ một nhiều. Cách tổ chức như sau:

+ Xây dựng cho mỗi bảng một mẫu biểu. Mẫu biểu ứng với bảng chính gọi là mẫu biểu chính, các mẫu biểu còn lại gọi là mẫu biểu phụ.

+ Trên mẫu biểu chính tạo các điều khiển Sub Form buộc vào các mẫu biểu phụ. Sử dụng các thuộc tính của điều khiển Sub Form để thiết lập mối quan hệ giữa mẫu biểu chính và mẫu biểu phụ, sao cho trên điều khiển **Sub Form** chỉ thể hiện các bản ghi của mẫu biểu phụ có liên quan đến bản ghi đang xét trên mẫu biểu chính.

+ Trong một mẫu biểu chính có thể có nhiều mẫu biểu phụ (gọi là mẫu biểu phụ cấp một). Số mẫu biểu phụ cấp một là không hạn chế.

+ Trong mỗi mẫu biểu phụ cấp một cho phép đặt các mẫu biểu phụ khác (gọi là mẫu biểu phụ cấp hai). Access chỉ cho phép tổ chức đến các mẫu biểu phụ cấp hai.

Cách tổ chức các mẫu biểu chính và phụ như sau:

Bước 1: Tạo một mẫu biểu chính và các mẫu biểu phụ một cách độc lập. Lúc đó giữa chúng chưa có quan hệ gì với nhau.

Bước 2: Sử dụng các thuộc tính của mẫu biểu để quy định dạng trình bày của các mẫu biểu phụ:

Thuộc tính	Dạng trình bày
Default View	
Datasheet	Dạng bảng
Single/Continuos Form	Dạng biểu
Pivot Table/Pivot Chart	Dạng tổng hợp theo bảng/đồ thị

Đối với mẫu biểu phụ ta thường đặt giá trị Datasheet cho thuộc tính View Allowed

Bước 3: Mở mẫu biểu chính trong chế độ Design. Bấm phím F11 để xuất hiện đồng thời cửa sổ Database.

Bước 4: Kéo một mẫu biểu phụ từ cửa sổ Database vào trong cửa sổ thiết kế của mẫu biểu chính. Access tạo một điều khiển SubForm buộc vào mẫu biểu phụ và một nhãn đi kèm.

Tên của điều khiển SubForm được đặt mặc định là tên của mẫu biểu phụ được kéo.

Tiêu đề của nhãn đi kèm cũng được đặt mặc định là tên của mẫu biểu phụ được kéo.

Để sửa thiết kế của mẫu biểu phụ, ta bấm đúp nút trái chuột tại điều khiển SubForm tương ứng. Khi đó sẽ nhận được cửa sổ thiết kế của mẫu biểu phụ.

Bước 5: Tạo sự liên kết giữa mẫu biểu chính và mẫu biểu phụ

+ Thuộc tính **LinkmasterFields** ghi các trường liên kết của mẫu biểu chính.

+ Thuộc tính **LinkchildFields** ghi các trường liên kết của mẫu biểu phụ.

Tác dụng của liên kết: Trên mẫu biểu phụ chỉ thể hiện các bản ghi có cùng giá trị trên các trường liên kết với bản ghi hiện hành của mẫu biểu chính.

Ví dụ 1: Tạo Subform không dùng Wizard

Giả sử cơ sở dữ liệu **Quanlyhanghoa** gồm các bảng sau:

Hang (MaH, TenH, Loai, SLTon)

Khach(MaK, TenK, Diachi, Dienthoai)

HoaDon(SoHD, NgayHD, MaK)

ChitietHD(SoHD, MaH, SLban, dongia)

Giữa bảng **HoaDon** và **ChitietHD** có quan hệ một nhiều theo trường **SoHD**. Nghĩa là nhiều mặt hàng có thể được bán và ghi trong một hoá đơn.

Ta có thể tổ chức nhập dữ liệu như sau:

- Khi đang xét một hoá đơn trong bảng **HoaDon** thì các thông tin của hoá đơn đó trong bảng **ChitietHD** cũng được thể hiện.

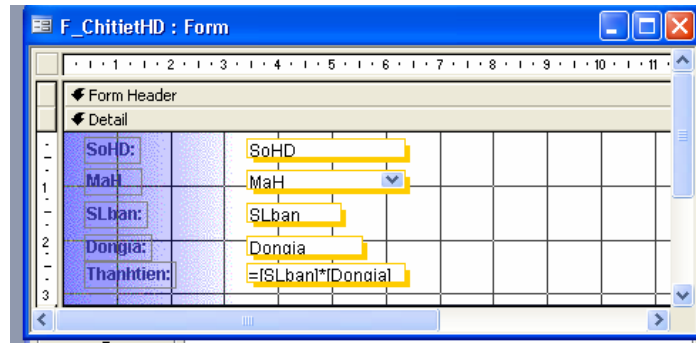
Trong trường hợp này ta tạo một mẫu biểu chính gắn với bảng **HoaDon** còn mẫu biểu phụ gắn với bảng **ChitietHD**. Hai mẫu biểu này gắn với nhau thông qua trường **SoHD**.

Các bước thực hiện:

Bước 1: Tạo mẫu biểu phụ **ChitietHD**, Nguồn dữ liệu của mẫu biểu này là bảng **ChitietHD**. Ta nên đặt thuộc tính của Form như sau **View Allowed: Datasheet**

Trong Form này ta thiết kế một Combo Box cho trường **MaH**, Combo Box này lấy dữ liệu từ bảng **Hang** và hiển thị hai cột **MaH**, **TenH** nhưng buộc vào cột **MaH**. Textbox **ThanhTien**=**[SLban]*[Dongia]**

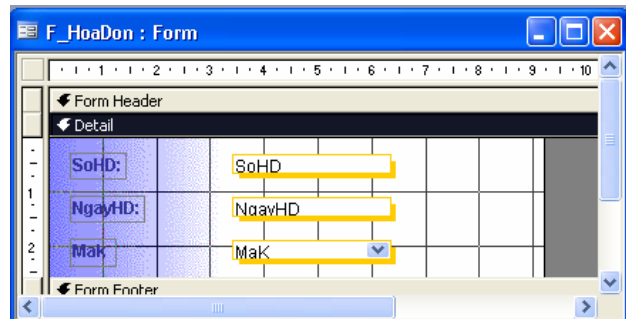
Cửa sổ thiết kế như hình 4.15



Hình 4.15

Bước 2: Tạo mẫu biểu chính **HoaDon**. Với dữ liệu nguồn của Form này là bảng **HoaDon**. Trên Form có 1 Combo Box **MaK**, dùng để chọn **MaK** trong bảng **Khach**. Combo Box có 2 cột **MaK**, **TenK** nhưng buộc vào cột **MaK**.

Thiết kế Form HoaDon như hình 4.16



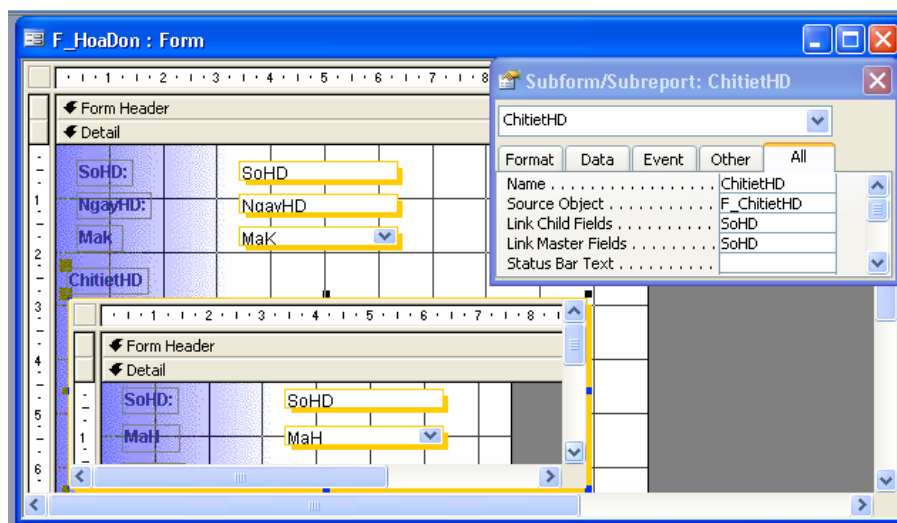
Hình 4.16

Bước 3: Mở Form **HoaDon** ở chế độ Design. Bấm phím F11 để hiện đồng thời cửa sổ Database. Kéo Form **ChitietHD** vào Form **HoaDon**. Kết quả tạo được một điều khiển SubForm buộc vào Form ChitietHD và một nhãn đi kèm. Tiếp đó ta mở thuộc tính của điều khiển SubForm và đưa vào các trường sau:

Link Child Fields: SoHD

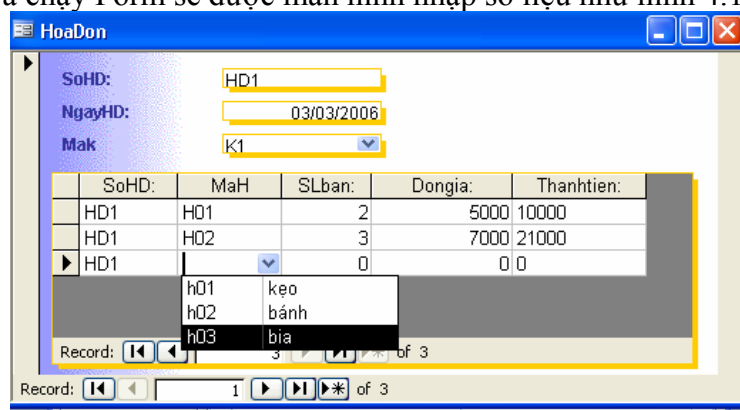
Link Master Fields: SoHD

Màn hình thiết kế Form như hình 4.17



Hình 4.17

Bước 4: Ghi và chạy Form sẽ được màn hình nhập số liệu như hình 4.18



Hình 4.18

Ví dụ 2: Tạo Subform dùng Wizard

Cũng với CSDL **Quanlyhanghoa** như trong ví dụ 1: Yêu cầu xây dựng Form nhập dữ liệu gồm 3 Form lồng nhau như sau:

- Form “**ChitietHD**” có dữ liệu nguồn là bảng **ChitietHD**
- Form “**HoaDon**” có dữ liệu nguồn là bảng **HoaDon** và chứa Form con “ChitietHD”
- Form “**Khach**” có dữ liệu nguồn là bảng **Khach** và chứa Form con “HoaDon”

Các bước thực hiện như sau:

Bước 1: Xây dựng các Form “**ChitietHD**”, “**HoaDon**” như trong VD1. Trong Form “**HoaDon**” không tạo MaK bằng Combo Box mà tạo bằng Textbox. Chọn chế độ hiển thị cho Form này là **View Allowed: Single Form**.

Bước 2: Mở Form “**HoaDon**” ở chế độ Design. Chọn nút **Subform/Subreport** trên hộp công cụ Toolbox, nháy và kéo trên Form để xác định vùng chữ nhật sẽ chứa Form con. Xuất hiện hộp thoại và chọn “**Use an Existing Form**”, chọn Form con đã xây dựng ở bước 1 là “**ChitietHD**”, nháy nút **Next**.

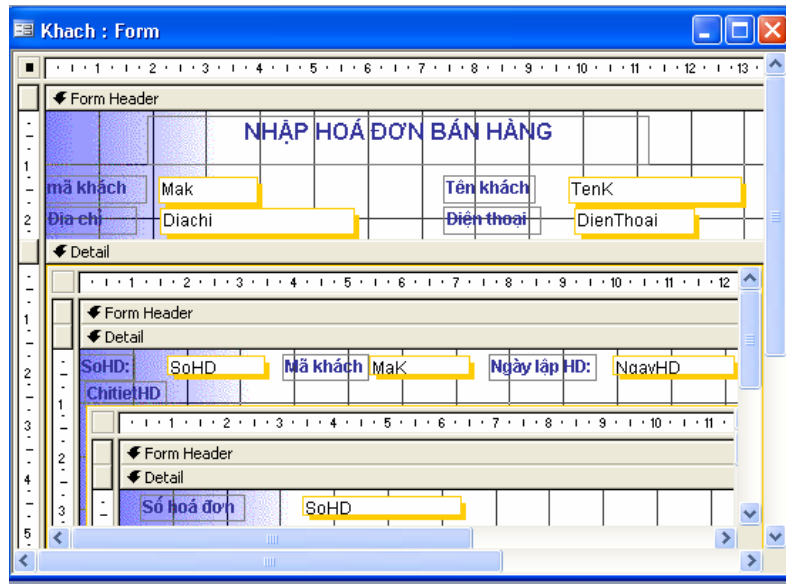
Bước 3: Trong hộp thoại tiếp theo chọn mục “**Show ChitietHD for each record in HoaDon using SoHD**”, nháy nút **Next**. Hộp thoại tiếp theo cho phép đặt tên cho Form con trong Form đang xây dựng, ta đặt là “**ChitietHD**”. Nháy **Finish** để trở về cửa sổ thiết kế Form.

Bước 4: Chỉnh sửa lại kích thước của Form con nếu cần. Ghi và đặt tên cho Form vừa thiết kế là “**HoaDon**”.

Bước 5: Tạo Form **Khach** với dữ liệu nguồn là bảng **Khach**. Đưa các trường của bảng này vào phần đầu của Form.

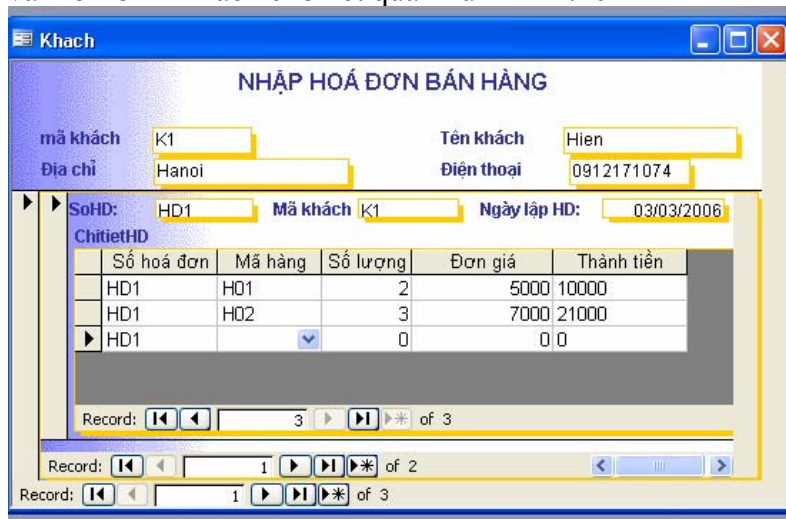
Bước 6: Chọn nút **Subform/Subreport** trên hộp công cụ Toolbox, nhấp và kéo trên Form để xác định vùng chữ nhật sẽ chứa Form con. Xuất hiện hộp thoại và chọn **“Use an Existing Form”**, chọn Form con đã xây dựng ở bước trước là **“HoaDon”**, nhấp nút **Next**.

Bước 7: Trong hộp thoại tiếp theo chọn mục **“Show HoaDon for each record in Khach using MaK”**, nhấp nút **Next**. Hộp thoại tiếp theo cho phép đặt tên cho Form con trong Form đang xây dựng, ta đặt là **“HoaDon”**. Nhấp **Finish** để trở về cửa sổ thiết kế Form. Cửa sổ thiết kế như hình 4.19



Hình 4.19

Bước 8: Ghi và mở Form **Khach** cho kết quả như hình 4.20



Hình 4.20

BÀI TẬP CHƯƠNG 4

Bài 1: Tạo Form nhập dữ liệu cho bảng **Thuchi** trong cơ sở dữ liệu **THUQUY** ở phần bài tập chương 2.

Bài 2: Tạo Form nhập dữ liệu đồng thời cho 2 bảng Danhsach và SoBD_Phach trong cơ sở dữ liệu **THICU** ở phần bài tập chương 2.

Bài 3: Cho CSDL **SINHVIEN** như sau:

Bảng **HOSO** để lưu hồ sơ các sinh viên gồm các trường:

- MaSV: Text, rộng 10, khoá chính, lưu mã sinh viên.
- Ho: Text, rộng 15, lưu phần họ và tên đệm của SV.
- Ten: Text, rộng 6, lưu phần tên của SV.
- NgaySinh: Date/Time, lưu ngày sinh
- GioiTinh: Text, rộng 3, lưu giới tính là “Nam” hay “Nu”
- Khoa: Text, rộng 20, tên khoa sinh viên học.
- QueQuan: Text, rộng 50, ghi quên quán của SV gồm huyện và tỉnh.
- DiaChi: Text, rộng 50, địa chỉ cư trú hiện tại của SV
- Dienthoai: Text, rộng 12, điện thoại của SV nếu có
- Giaitich, Daiso, Tinhoc, Triethoc, Anhvan: Number, Double, lưu điểm thi các môn học kỳ I năm thứ nhất của SV.

Bảng **GIADINH** để lưu nhưng thông tin người thân trong gia đình có trách nhiệm trực tiếp đối với SV gồm các trường:

- MaSV: Text, rộng 10, mã sinh viên, dùng để liên kết với bảng HOSO.
- Hoten: Text, rộng 25, lưu họ và tên người thân của SV
- Namsinh: Number, long Integer, lưu năm sinh.
- Quanhe: Text, rộng 12, quan hệ với SV.
- Diachi: Text, rộng 50.
- Dienthoai: Text, rộng 12, số điện thoại liên hệ nếu có.

Mối quan hệ giữa bảng **HOSO** và bảng **GIADINH** là 1-n thông qua trường MaSV.

Xây dựng các Form như sau:

1. Form 1: dạng Autoform: Columnar để nhập dữ liệu cho bảng HOSO
2. Form 2: dạng Autoform: Tabular để nhập dữ liệu cho bảng HOSO
3. Form 3: dạng Autoform: Datasheet để nhập dữ liệu cho bảng HOSO
4. Form 4: dùng wizard tạo Form nhập dữ liệu cho hai bảng HOSO và GIADINH
5. Tạo một Form để xem các bản ghi trong Query1 thuộc bài tập 7 chương 3.
6. Tạo **“Form hỏi đáp”** để tìm kiếm các SV trong một khoa có ngày sinh lớn hơn hoặc bằng một ngày nào đó. Form có 4 điều khiển: dùng một Combo Box tên là (CBkhoa) để nhập tên khoa, dùng một Text Box tên là (tbNgaysinh) để nhập ngày sinh, dùng nút lệnh “Tìm kiếm” để chạy một truy vấn, dùng nút lệnh “Đóng Form” để đóng Form.

Gợi ý: Tạo truy vấn “query Khoa và NgaySinh” từ bảng HOSO đưa các trường MaSV, Ho, Ten, Ngaysinh, Khoa vào lưới thiết kế. Dòng Criteria của cột Ngaysinh đưa vào công thức \geq [Forms]![Form hỏi đáp]![tbNgaysinh] (công thức có nghĩa trường Ngaysinh phải lớn hơn hay bằng giá trị lưu trong điều khiển tbNgaysinh của biểu mẫu “Form hỏi đáp”, “Form hỏi đáp” là một đối tượng của tập hợp Forms), dòng Criteria của cột Kho nhập công thức = [Forms]![Form hỏi đáp]![CBKhoa].

7. Tạo một Form lọc dữ liệu DSKhoa: cho phép chọn tên khoa từ một Combo Box, danh sách SV hiện trong biểu mẫu con DSKhoa_Subform.

Gợi ý: Tạo mẫu biểu con trước dùng Design view, mở hộp thoại thuộc tính Form, nháy vào nút Build trên dòng thuộc tính Record Source để xuất hiện cửa sổ thiết kế truy vấn Query Build, chọn bảng HOSO, đưa vào lưới thiết kế truy vấn các trường Ho, Ten, Ngaysinh, Gioitinh, Khoa. Dòng Criteria trên cột Khoa nhập [CBKhoa] (tức là tiêu chuẩn lấy từ combo box).

Trong mẫu biểu chính DSkhoa, tạo combo Box có thuộc tính Name là CBKhoa.

CHƯƠNG V

BÁO BIỂU

Báo biểu (Report) là công cụ thuận tiện để tổ chức in dữ liệu của một bảng, truy vấn. Mặc dù có thể in dữ liệu từ bảng, truy vấn và mẫu biểu, nhưng báo biểu cung cấp nhiều khả năng linh hoạt để điều khiển công việc in ấn. Report không hỗ trợ các điều khiển tương tác nhưng ta có thể đưa vào report các điều khiển để hiển thị dữ liệu như hộp văn bản, các hộp kiểm tra... Report cũng có thể chứa hình ảnh đồ họa. Report còn cho phép tổ chức và trình bày dữ liệu theo nhóm, tính toán các tổng nhóm, thống kê dữ liệu theo nhóm.

Trong chế độ Design View cấu trúc của một Report giống với Form gồm 5 thành phần: Report Header (phần đầu báo biểu, in một lần ở trang đầu tiên), Page Header (phần đầu trang, chứa tiêu đề các cột), Detail (nơi đặt các trường của bảng, truy vấn, các bản ghi sẽ được in ở phần này), Page Footer (phần cuối mỗi trang in, chứa số trang), Report Footer (phần cuối của báo biểu, in một lần ở trang cuối, chứa các thông tin tổng hợp của báo biểu). Nếu Report có gộp nhóm thì còn thêm 2 phần là đầu nhóm (in tiêu đề nhóm, tiêu đề cột) và phần cuối nhóm (thống kê tổng hợp về nhóm, tổng các cột, trung bình các cột).

Việc tạo Report dùng Wizard và Design View hoàn toàn giống với cách tạo một Form. Chức năng và cách dùng của hộp công cụ Toolbox và Field List vẫn giữ nguyên.

1. TẠO BÁO BIỂU BẢNG AUTOREPORT

Chọn một trong hai kiểu báo biểu tự động (có từ AutoReport) và nhấn OK.

1.1 Tạo báo biểu tự động kiểu Tabular

Trong AutoReport: Tabular (báo biểu dạng hàng cột) các trường thuộc một bản ghi xuất hiện trên một dòng với các nhãn hiện một lần tại phần đầu của báo biểu.

Ta xét cách tạo AutoReport: Tabular thông qua ví dụ sau:

Ví dụ: giả sử trong CSDL **Quanlyhanghoa**, có bảng **Khach**, ta tạo Report cho bảng này.

Bước 1: nhấn nút New trong phần Report trên cửa sổ Database, chọn bảng Khách và chọn mục “**AutoReport: Tabular**”, nhấn Ok. Kết quả ta được báo biểu ở chế độ Print Preview như hình 5.1

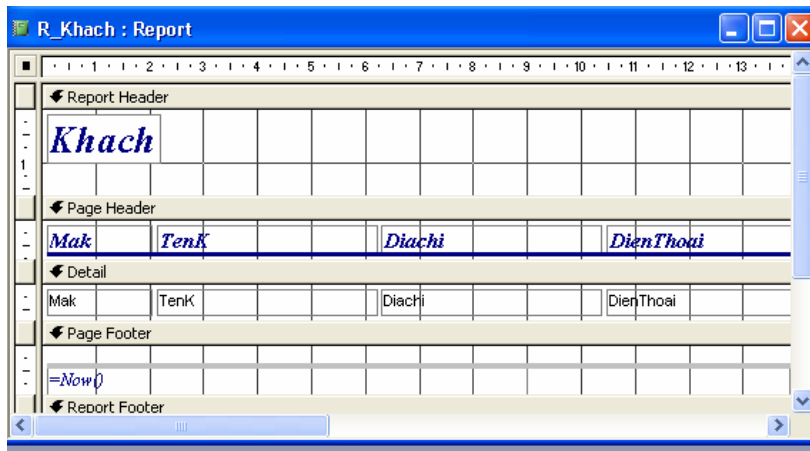


<i>Mak</i>	<i>TenK</i>	<i>Diachi</i>	<i>DienThoai</i>
K1	Hien	Hanoi	0912171074
K2	Linh	Hanoi	
K3	Thang	Hatay	034881764

Hình 5.1

Bước 2: Ghi và đặt tên cho báo biểu là R_Khach

Nếu cách trình bày của báo biểu in ra chưa vừa ý ta có thể chuyển sang chế độ Design View để sửa chữa: thay đổi kích thước và vị trí các điều khiển, định lại phông chữ và màu chữ, màu nền, thêm các đường kẻ khung cho báo biểu. Report đã mở ở chế độ Design View như hình 5.2



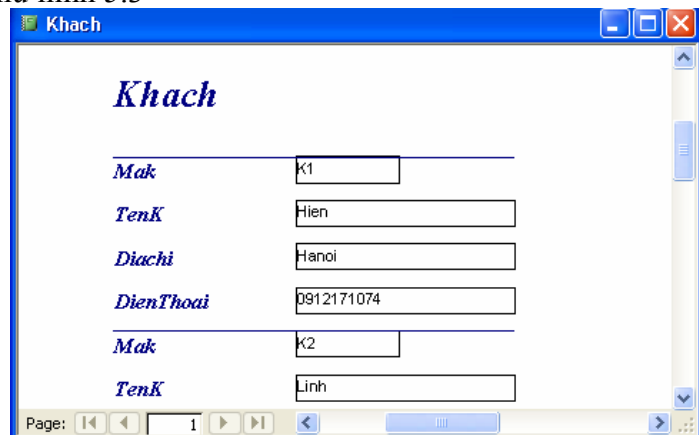
Hình 5.2

1.2 Tạo báo biểu tự động kiểu Columnar

Trong AutoReport: Columnar (báo biểu dạng cột) giá trị mỗi trường của bản ghi xuất hiện trên một dòng tách biệt cùng với nhãn ở bên trái nó.

Ta xét cách tạo AutoReport: Columnar thông qua ví dụ trên:

Bước 1: nhấn nút New trong phần Report trên cửa sổ Database, chọn bảng Khách và chọn mục “**AutoReport: Columnar**”, nhấn Ok. Kết quả ta được báo biểu ở chế độ Print Preview như hình 5.3



Hình 5.1

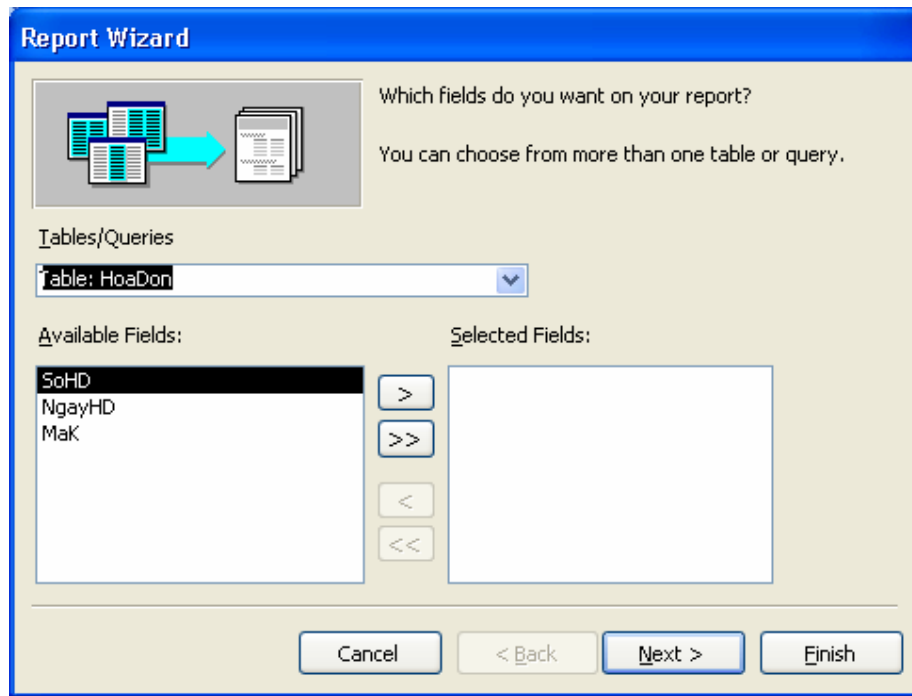
Bước 2: Ghi và đặt tên cho báo biểu

2. TẠO BÁO BIỂU DÙNG REPORT WIZARD

Report Wizard giúp ta nhanh chóng tạo một báo biểu từ một hay nhiều bảng, cho ta lựa chọn gộp nhóm và sắp xếp dữ liệu. Ta xem cách tạo Report qua ví dụ sau:

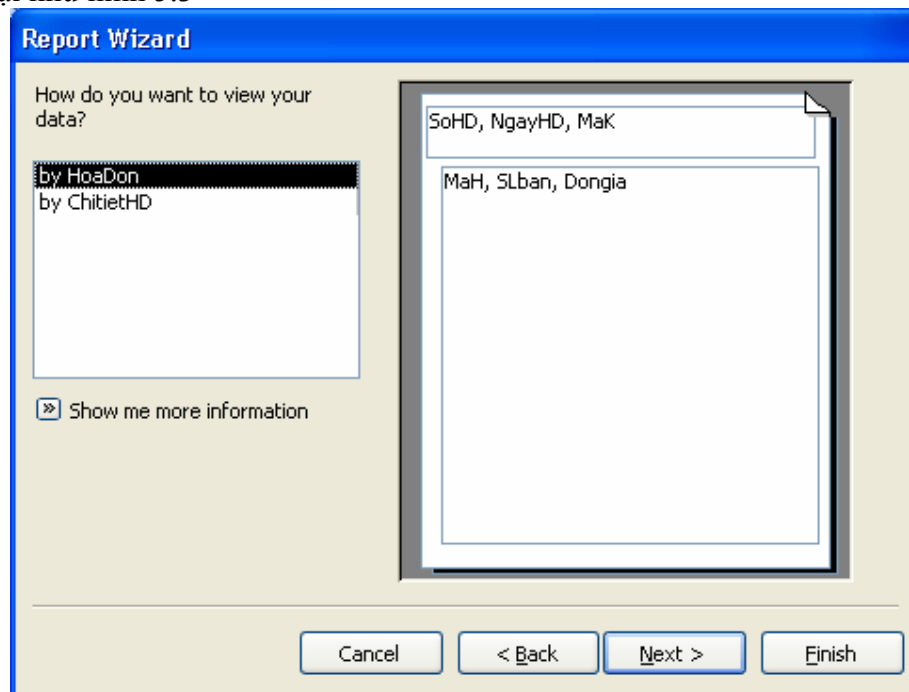
Ví dụ: Giả sử cần lập một báo cáo về chi tiết từng hoá đơn bán hàng, dữ liệu lấy từ bảng **HoaDon** và **ChitietHD** của CSDL **Quanlyhanghoa** như ở chương trước.

Bước 1: nhấn nút New trong phần Report trên cửa sổ Database, Report Wizard, nhấn OK (hoặc chọn Create Report by using wizard ở màn hình Database). Xuất hiện hộp thoại của Report Wizard như hình 5.2



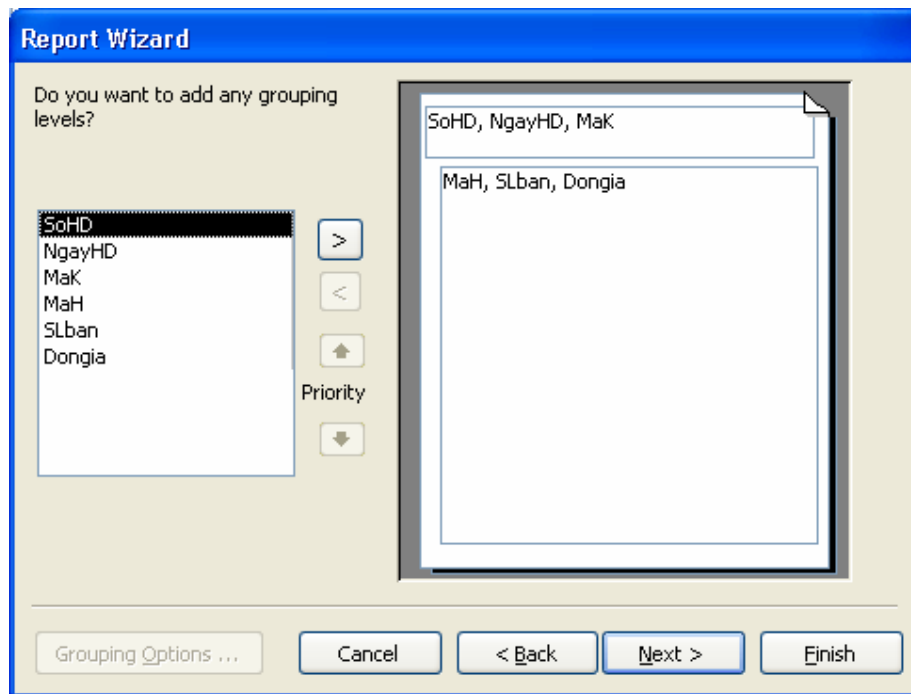
Hình 5.2

Bước 2: Chọn bảng **HoaDon** trong mục Table/Queries, đưa các trường **SoHD**, **MaK**, **NgàyHD** sang khung Selected fields. Chọn tiếp bảng thứ hai là **ChitietHD** đưa các trường **MaH**, **SLban**, **Dongia** sang khung Selected fields, nhấn nút **Next**. Xuất hiện hộp thoại như hình 5.3



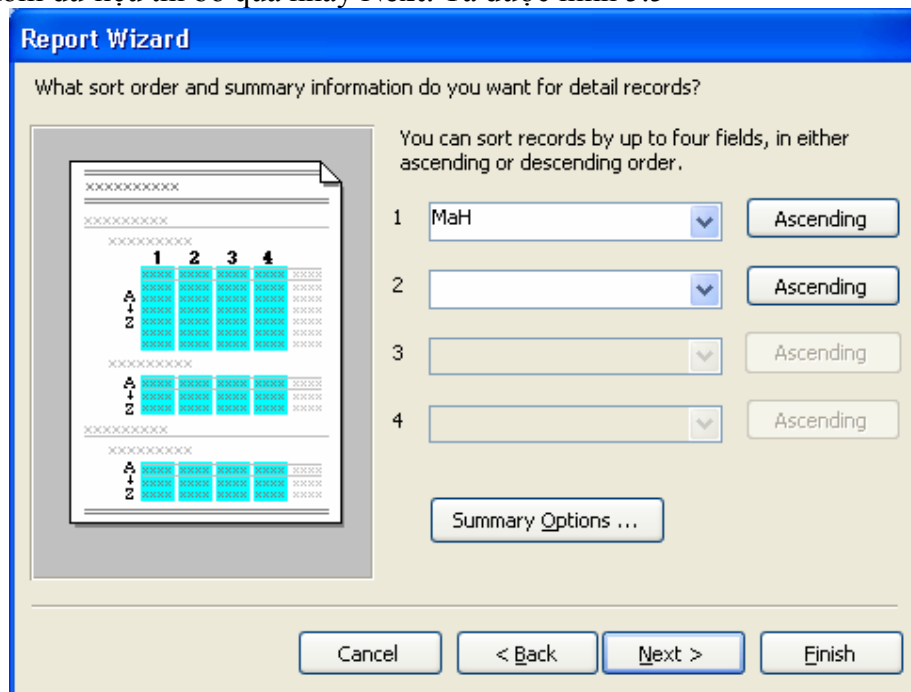
Hình 5.3

Bước 3: Ta chọn mục By HoaDon, nhấn nút **Next** Ta được hình 5.4



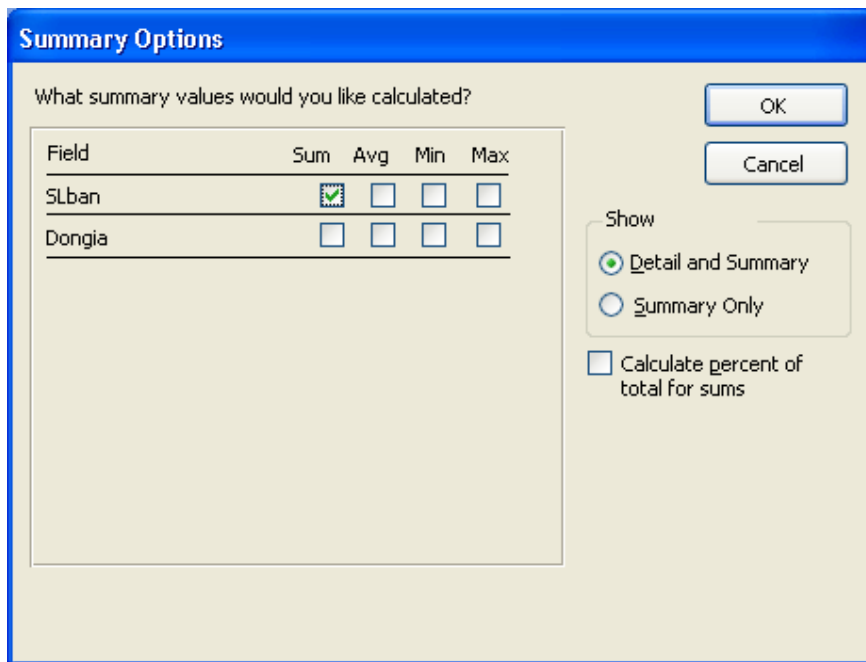
Hình 5.4

Bước 4: Trên hình 5.4 cho phép chọn trường phân nhóm dữ liệu. Ta có thể không phân nhóm dữ liệu thì bỏ qua nhấn **Next**. Ta được hình 5.5

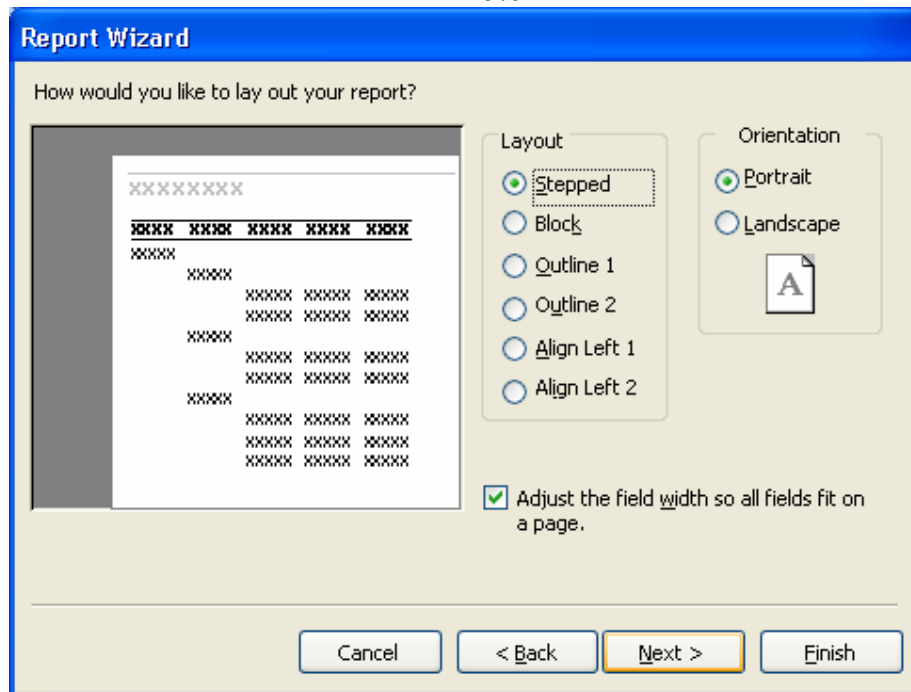


Hình 5.5

Bước 5: Hình 5.5 cho phép lựa chọn kiểu sắp xếp, tại đây ta chọn sắp xếp theo chiều tăng dần của **MaH**. Nhấn nút **Summary Options** để xác định các trường tính toán trong Report, trong ví dụ này ta chọn trường **SLban** như trong hình 5.6. Bấm OK trở lại và bấm **Next**. xuất hiện hộp thoại hình 5.7

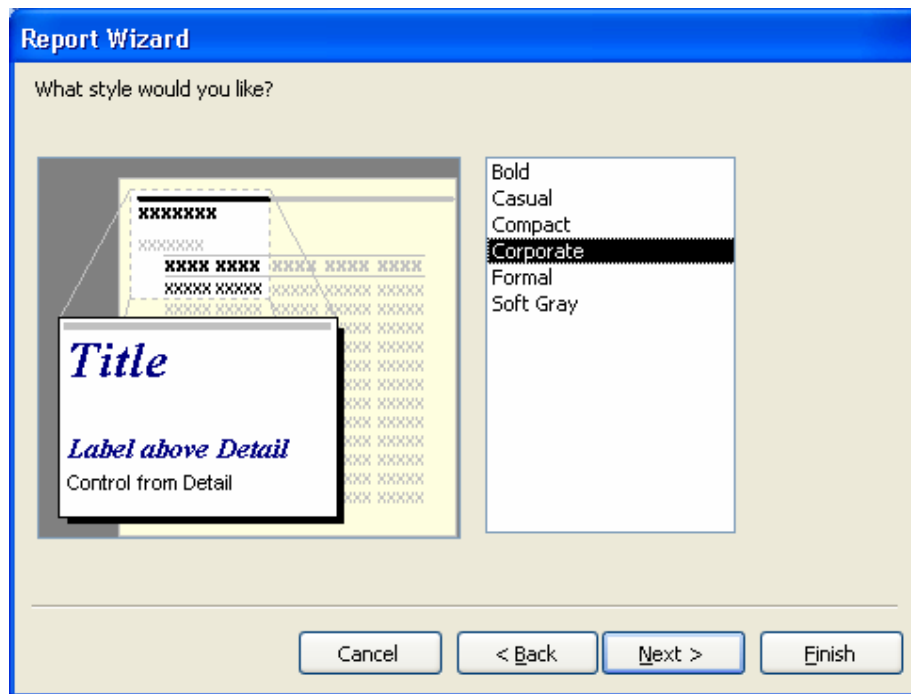


Hình 5.6



Hình 5.7

Bước 6: Trong hình 5.7 chọn **Stepped**, **Portrait**, nhấn nút **Next**. Ta có giao diện như hình 5.8



Hình 5.8

Bước 7: Chọn kiểu **Corporate**, nhấn nút **Next**

Bước 8: Đặt tên cho Report là **R_HoaDon**, nhấn nút **Finish**. Kết quả được báo biểu dạng Print Preview như hình 5.9

<i>SoHD</i>	<i>NgayHD</i>	<i>MaK</i>	<i>MaH</i>	<i>SLban</i>	<i>Dongia</i>
HD1	03/03/2006	K1			
			H01	2	5000
			H02	3	7000
Summary for 'SoHD' = HD1 (2 detail records)					
Sum				5	
HD2	06/07/2006	K2			
			H01	1	5000
			H03	3	8000
Summary for 'SoHD' = HD2 (2 detail records)					
Sum				4	

Hình 5.9

3. TẠO BÁO BIỂU DÙNG DESIGN VIEW

Ở chế độ này chúng ta tự thiết kế Report theo ý muốn. Ta xét cách tạo thông qua ví dụ sau:

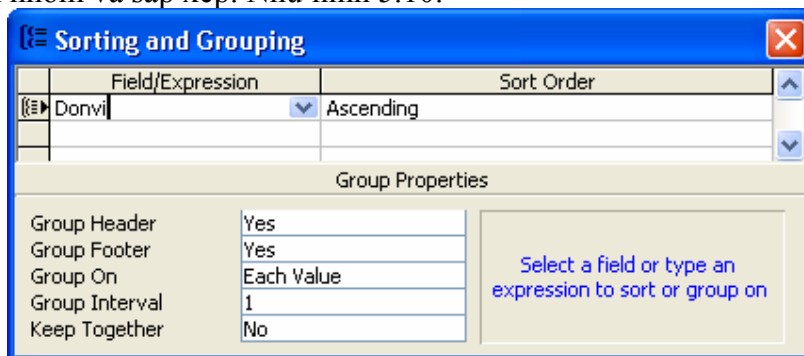
Ví dụ: Giả sử từ bảng **Nhanvien(MaNV, Hoten, Ngaysinh, Gioitinh, Diachi, Dienthoai, Donvi, Luong)** của CSDL **QLNHANVIEN** ta cần lập một bảng lương với các yêu cầu sau:

- Bảng gồm các cột: Số thứ tự, Họ và tên, Ngày sinh, Lương cơ bản, Bảo hiểm xã hội (bằng 5% Lương). Bảo hiểm y tế (bằng 1% lương), Tổng số tiền lĩnh (bằng lương chính trừ đi bảo hiểm xã hội và bảo hiểm y tế, cũng bằng 94% Lương).

- Trong bảng lương các nhân viên được nhóm theo từng đơn vị, đầu mỗi nhóm có ghi tên đơn vị, cuối mỗi nhóm có tính Tổng lương, Tổng bảo hiểm xã hội, Tổng bảo hiểm y tế, Tổng số tiền lĩnh của nhóm. Trường dùng để phân nhóm là trường DonVi.
- Cuối báo cáo có tính Tổng lương, Tổng bảo hiểm xã hội, Tổng bảo hiểm y tế, Tổng số tiền lĩnh của toàn bộ công ty.

Các bước thực hiện như sau:

Bước 1: Trong cửa sổ Database nhảy vào mục Reports chọn New, chọn bảng Nhanvien và chọn mục Design view, nhấn OK. Xuất hiện cửa sổ thiết kế Report với 3 phần: Page Header, Detail, Page Footer. Bấm View/Report Header- Footer để hiển thị phần Report Header và Report Footer. Bấm View/Sorting and Grouping để chọn các trường phân nhóm và sắp xếp. Như hình 5.10.



Hình 5.10

Bước 2: Chọn trường phân nhóm là Donvi trong cột Field/Expression, cột Sort Order chọn là Ascending (tức là nhóm các bản ghi theo từng đơn vị, các đơn vị xếp theo chiều tăng dần). Tại mục Group Header và Group Footer chọn là Yes. Khi đó trên cửa sổ thiết kế Report xuất hiện thêm hai phần là **Donvi Header** và **Donvi Footer**.

Bước 3: Thiết kế phần Report Header. Dùng nút Label của Toolbox để tạo một đối tượng Label trên vùng này, Văn bản trên nhãn là “**Bảng lương khoa CNTT tháng 10 năm 2006**”.

Bước 4: Thiết kế phần Detail:

-Tạo Text Box STT(số thứ tự): Chọn nút Text Box trên Toolbox, kéo vào vùng Detail, xoá nhãn liên kết với với Text Box này, các thuộc tính như sau:

Name: **STT** Control Source: =1

Running Sum: **Over Group** (đánh số thứ tự theo từng nhóm)

- Kéo ba trường **HoTen, Ngaysinh, Luong** từ Field List vào phần Detail để tạo ba Text Box, xoá các nhãn đi kèm với các Text Box.

- Tạo **Text Box BHXH**: chọn nút Text Box trên Toolbox và kéo vào vùng Detail, xoá bỏ nhãn đi kèm và điền các thuộc tính như sau:

Name: **BHXH** Control Source=[**Luong**]*0.05

Format: **fix** Deceimal Places:2

- Tạo **Text Box BHYT**: chọn nút Text Box trên Toolbox và kéo vào vùng Detail, xoá bỏ nhãn đi kèm và điền các thuộc tính như sau:

Name: **BHYT** Control Source=[**Luong**]*0.01

Format: **fix** Deceimal Places:2

- Tạo **Text Box Tienlinh**: tương tự ta xác định các thuộc tính:

Name: **TienLinh** Control Source=[**Luong**]-[**BHXH**]-[**BHYT**]

Format: **fix** Deceimal Places:2

Chọn tất cả các điều khiển trên phần Detail, định dạng cùng chiều cao, dóng theo phía trên, cách đều nhau, cùng Font chữ cỡ chữ. Chuyển sang chế độ Print Preview xem thử kết quả, chuyển lại chế độ Design View.

Bước 5: Thiết kế phần Donvi Header: Tạo một nhãn kèm văn bản là “**Đơn vị**”. Kéo trường Donvi từ Fields List vào để tạo **Text Box Donvi**. Như vậy ở đầu mỗi nhóm sẽ in ra tên của đơn vị.

Bước 6: Thiết kế phần Donvi Footer: Tạo một nhãn kèm văn bản “**Tổng cộng của đơn vị**”, tạo 4 Text Box như sau:

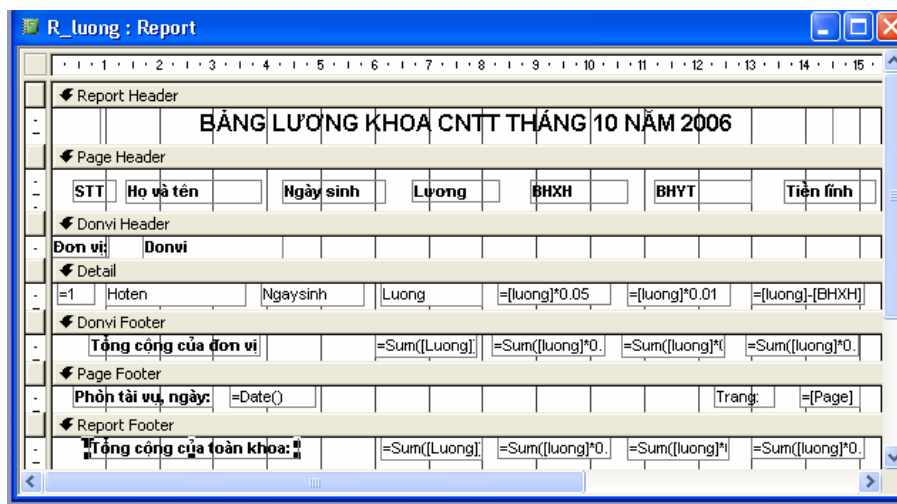
Name: TongLuong	Control Source= Sum([Luong])
Format: fix	Deceimal Places: 2
Name: TongBHXH	Control Source= Sum([Luong]*0.05)
Format: fix	Deceimal Places: 2
Name: TongBHYT	Control Source= Sum([Luong]*0.01)
Format: fix	Deceimal Places: 2
Name: TongTienLinh	Control Source= Sum([Luong]*0.94)
Format: fix	Deceimal Places: 2

Bước 7: Thiết kế phần Report Footer: Sao chép tất cả các đối tượng trong phần Donvi Footer xuống phần này và sửa chữa văn bản trên nhãn thành “**Tổng cộng của toàn Khoa:**”

Bước 8: Thiết kế phần Page Header

Tạo 7 nhãn làm tiêu đề các cột STT, Họ và tên, Ngày sinh, Lương, BHXH, BHYT, Tiền lĩnh. Vị trí và kích thước các nhãn này phải tương ứng với các Text Box thuộc dải Detail.

Bước 9: Thiết kế phần Page Footer: Tạo một nhãn với văn bản là “**Phòng tài vụ ngày**”. Tạo một Text Box chứa công thức=**Date()**. Tạo một nhãn với văn bản là “**Trang:**”. Tạo một Text Box chứa công thức=**[Page]**. Màn hình thiết kế như hình 5.11



Hình 5.11

Bước 10: Định dạng lại các đối tượng cho đẹp, có thể dùng lệnh File/Page Setup để chọn khổ giấy in, căn lề in báo cáo. Ta có in báo biểu ra máy in dùng lệnh File/Print. Kết quả ta được bảng lương như hình 5.12

STT	Họ và tên	Ngày sinh	Lương	BHXH	BHYT	Tiền lĩnh
Đơn vị BM CNPM						
1	Cao Thi Luyen	01/02/1979	1000000	50000.00	10000.00	940000.00
2	Nguyen Duc Du	03/02/1979	1000000	50000.00	10000.00	940000.00
3	Bui Hong Nhung	02/05/1979	1000000	50000.00	10000.00	940000.00
Tổng cộng của đơn vị:			3000000.00	150000.00	30000.00	2820000.00
Đơn vị BM Mạng&HTTT						
1	Ngo Thuy Linh	03/07/1981	700000	35000.00	7000.00	658000.00
2	Nguyen Quoc Tuan	04/05/1978	900000	45000.00	9000.00	846000.00
3	Do Thanh Thuy	05/06/1979	1000000	50000.00	10000.00	940000.00
4	Dang Thu Hien	18/10/1976	1200000	60000.00	12000.00	1128000.00
Tổng cộng của đơn vị:			3800000.00	190000.00	38000.00	3572000.00
Tổng cộng của toàn khoa:			6800000.00	340000.00	68000.00	6392000.00

Hình 5.12

Các hàm và hằng thường dùng trong thiết kế báo biểu:

Các hàm thống kê: **Count, Sum, Avg, Max, Min, StDev, Var, First, Last Date()** dùng để chèn ngày tháng năm hiện hành.

Now() dùng để chèn ngày tháng năm và giờ phút giây hiện hành.

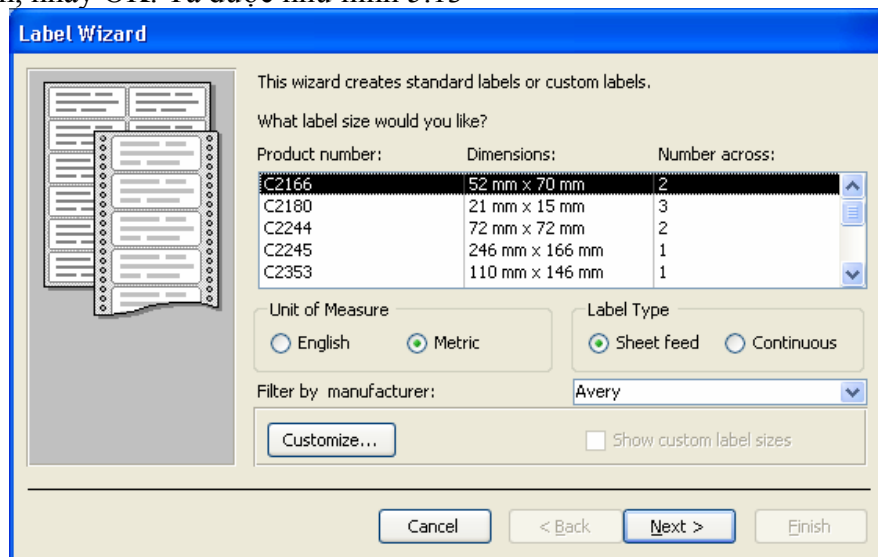
[Page] chèn số trang.

[Pages] chèn tổng số trang của Report.

4. TẠO CÁC NHÃN DÙNG LABEL WIZARD

Vì CSDL thường được dùng lưu trữ thông tin về khách hàng hay hàng hoá, do đó một trong các tiện ích hay được sử dụng nhất với các thông tin trong CSDL là tạo nhãn. Access có thể tạo các loại nhãn dùng để dán thư, dán trên hàng hoá, tạo các phiếu báo điểm thi. Giả sử ta cần tạo nhãn ghi thông tin tóm tắt về nhân viên từ bảng **Nhanvien(MaNV, Hoten, Ngaysinh, Gioitinh, Diachi, Dienthoai, Donvi, Luong)** trong CSDL **QLNHANVIEN**. Ta dùng Label Wizard thực hiện các bước sau:

Bước 1: Trong cửa sổ Database, nhấn nút New và chọn mục Label Wizard, chọn bảng NhanVien, nhấn OK. Ta được như hình 5.13

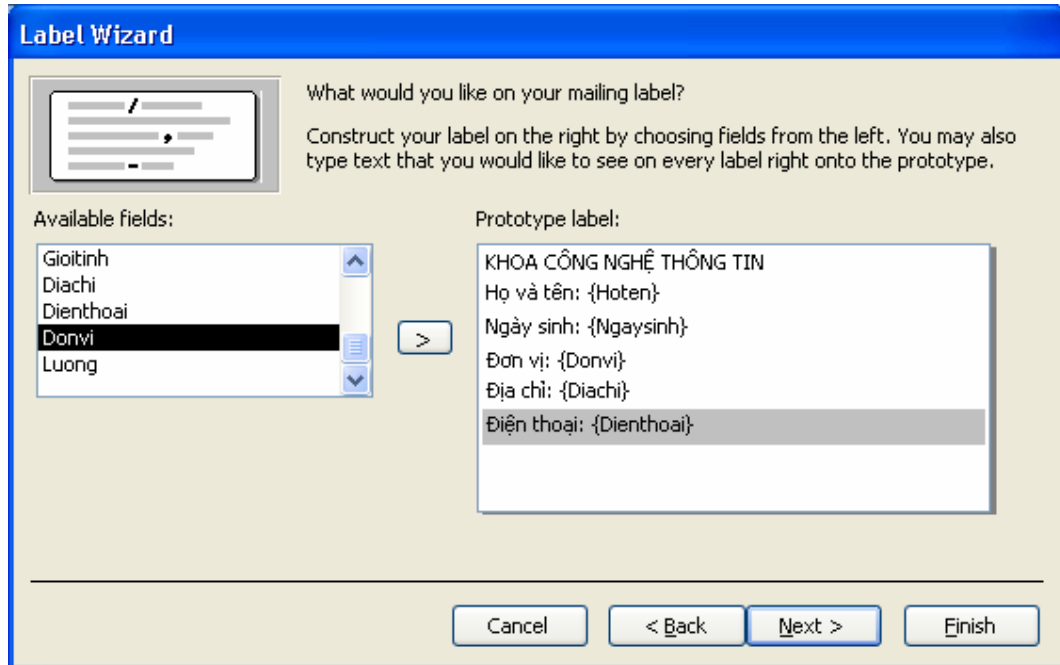


Hình 5.13

Bước 2: Tại hình 5.13 chọn đơn vị đo là Metric, chọn kiểu nhãn là Sheet feed, tên nhãn là C2166 của nhà sản xuất Avery(kích thước là 52mm x 70mm). Số 2 tổng cột

Number across có nghĩa là bố trí in 2 nhãn theo chiều ngang giấy. Dùng nút Customize của hộp thoại này ta có thể tự thiết kế kích thước nhãn, bao nhiêu nhãn xếp theo chiều ngang, khoảng cách giữa các nhãn, khoảng cách giữa nhãn với lề trang giấy, đặt tên cho kiểu tự thiết kế để sau này có thể dùng lại. Nháy nút Next.

Bước 3: Hộp thoại tiếp theo cho phép chọn Font chữ, nháy nút Next, ta có hình 5.14

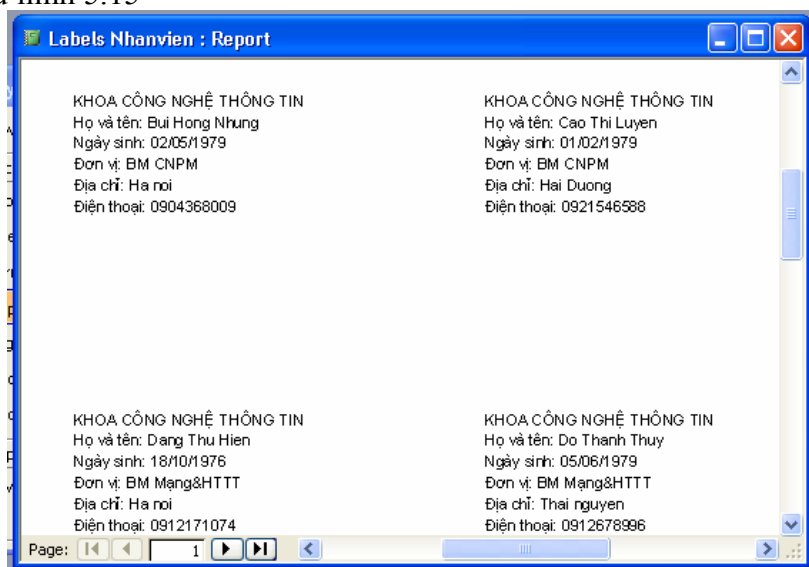


Hình 5.14

Bước 4: Đưa các trường Hoten, Ngaysinh, Donvi, Diachi, Dienthoai sang khung Prototype Label và gõ thêm dòng thuyết minh bên cạnh tên trường, nháy nút Next.

Bước 5: Chọn trường dùng để sắp xếp cá nhân, chọn trường Hoten, nháy nút Next.

Bước 6: Đặt tên cho nhãn vừa xây dựng. Ta chọn là “Labels Nhanvien”, chọn “See the Labels as they will look printed”, nháy Finish, kết quả ta được nhãn ở chế độ Print preview như hình 5.15



Hình 5.15

BÀI TẬP CHƯƠNG 5

Bài 1. Dùng CSDL **Quanlyhanghoa** mô tả trong chương trước như sau:

Hang(**MaH**, **TenH**, **Loai**, **SLTon**) để lưu trữ tất cả các mặt hàng có trong cửa hàng.

Khach(**MaK**, **TenK**, **Diachi**, **DienThoai**) lưu trữ những thông tin về khách hàng.

HoaDon(**SoHD**, **NgayHD**, **MaK**) dùng để lưu trữ các thông tin về việc bán hàng.

ChitietHD(**SoHD**, **MaH**, **SLban**, **Dongia**). Thực hiện các yêu cầu sau:

1. Report1: dùng Wizard tạo báo cáo chi tiết từng hoá đơn bán hàng, dữ liệu lấy từ hai bảng **HoaDon** và **ChitietHD**.

2. Report2: Dùng Design View thiết kế báo biểu về các hoá đơn xuất hàng có gộp nhóm, dữ liệu lấy từ cả bảng **HoaDon**, **ChitietHD**, **Hang**, **Khach**. Mẫu báo cáo có dạng.

THỐNG KÊ CÁC HOÁ ĐƠN BÁN HÀNG

Số hoá đơn:....

Tên khách hàng:...

Ngày bán:...

STT	Tên hàng	Số lượng	Đơn giá	Thành tiền
1	Xxxx	Xxx	Xxxx	Xxx
2	Xxxx	Xxx	Xxxx	Xxx

Tổng giá trị hàng của hoá đơn: xxxx

Số hoá đơn:...

.....

Tổng giá trị toàn bộ hàng đã bán: xxxxxx

3. Report3: Báo cáo từng khách hàng đã nhận từng mặt hàng với số lượng bao nhiêu và tiền là bao nhiêu. Báo cáo phân nhóm theo khách hàng, phân nhóm con theo tên hàng. Mẫu báo cáo có dạng:

THỐNG KÊ KHÁCH HÀNG - TÊN HÀNG

Tên khách hàng:

STT	Tên hàng	Số lượng	Tiền
1	Xxxxxx	Xxxxx	Xxxxx
2	Xxxxxx	Xxxxx	xxxxx

Tổng tiền: xxx

Tên khách hàng:....

Bài 2: Dùng CSDL **QLNhanvien** như mô tả trong phần trên. Tạo các Report:

1. Report1: Từ bảng **Nhanvien** thiết kế bảng lương có phân nhóm theo đơn vị.

2. Report2: Từ bảng **Nhanvien** tính tổng tiền lương của từng đơn vị và vẽ biểu đồ dạng 3D-Column.

Bài 3: Tạo báo biểu in danh sách tất cả các khách hàng của công ty điện lực dựa trên số liệu của cơ sở dữ liệu **QUANLYDIEN** trong phần bài tập chương 2.

CHƯƠNG 6

MACRO VÀ MENU

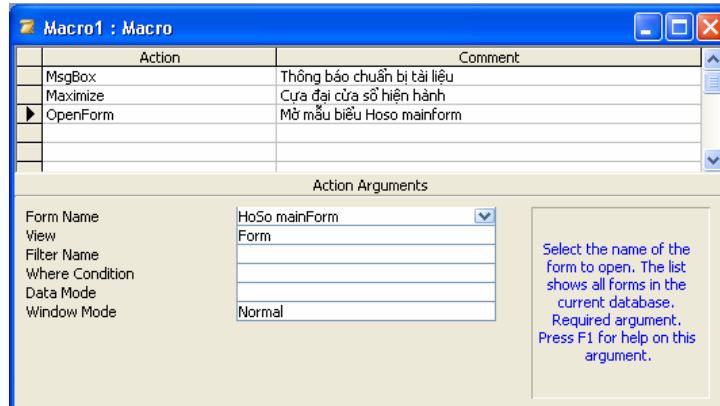
1. MACRO

Macro là tập hợp cá hành động (action) để thực hiện nhiều thao tác cùng một lúc giúp ta thực hiện tự động các công việc xử lý dữ liệu. Ta có thể tạo một macro để mở một Table, Query, Form, Report. Có thể tạo một macro thông báo kèm theo tiếng bíp bíp để gây sự chú ý. Có thể tạo một macro để di chuyển hay đặt lại kích thước cho một cửa sổ, hoặc làm cực đại, cực tiểu hoặc khôi phục một cửa sổ. Macro cũng thường dùng kết hợp với các sự kiện của Form và cá điều khiển trên Form, như sự kiện Current của một Form, sự kiện Click của một nút lệnh hoặc sự kiện DbClick của một hộp văn bản.

1.1 Tạo và sử dụng Macro

Xét CSDL **SINHVIEN** trong bài tập của chương 4, giả sử dùng Form Wizard đã tạo được biểu mẫu “HoSo mainform” để nhập dữ liệu cho hai bảng HoSo và GiaDinh (trong biểu mẫu này có một biểu mẫu con là “GiaDinh subform”). Ta cần tạo một macro thực hiện ba hành động: **Hiện một thông báo với nội dung “Hãy chuẩn bị để nhập hồ sơ sinh viên mới”, làm cực đại cửa sổ hiện hành, mở mẫu biểu “Hoso mainform” để nhập dữ liệu.** Quá trình thực hiện như sau:

Bước 1: Trong cửa sổ Database chọn đối tượng Macros, nhấn nút New xuất hiện cửa sổ thiết kế như hình 6.1



Hình 6.1

Trong cửa sổ này có 3 thành phần: cột Action dùng để nhập một hành động trong macro, cột Comment ghi chú thích cho hành động vừa chọn, khung Action Arguments dùng để nhập các đối số hay thông tin của hành động.

Bước 2: Đưa các hành động vào Macro

- Chọn hành động **MsgBox** trong cột Action, xác định các thông số trong khung Action Argument tương ứng như sau:

Message: “Hãy chuẩn bị để nhập hồ sơ sinh viên mới”

Beep: Yes

Type: Information

Title: Thông báo

- Chọn hành động **Maximize**

- Chọn hành động **OpenForm**, xác định các đối số như sau:

Form name: HoSo mainForm

View: Form (mở mẫu biểu ở chế độ Form view)

Bước 3: Đóng cửa sổ thiết kế và đặt tên cho Macro là M_moFormHoSo

Chú ý: Ta chạy Macro bằng cách nháy đúp chuột lên tên của Macro hoặc nháy nút Run trên thanh công cụ.

1.2 Các hành động Macro thường dùng

OpenTable: mở một bảng theo chế độ Datasheet, Design hay Print Preview.

OpenQuery: mở một truy vấn theo chế độ Datasheet, Design hay Print Preview.

OpenForm: mở một mẫu biểu theo chế độ Form, Datasheet, Design hay Print Preview.

OpenReport: mở một báo biểu theo chế độ Print, Print Preview hay Design.

RunMacro: chạy một macro, đối số Repeat Count: số lần lặp lại.

RunApp: Chạy một ứng dụng trong Windows.

ReQuery: Cập nhật dữ liệu cho đối tượng.

Close: Đóng các đối tượng đang mở như Table, Query, Form, Report.

Quit: Thoát khỏi Access.

MsgBox: hiển thị hộp thông báo.

GotoRecord: di chuyển bản ghi (trước, sau, đầu, cuối, số...) trên Table, Query và Form.

GotoControl: di chuyển con trỏ đến điều khiển xác định trên Form.

Maximize: Cựa đại cửa sổ hiện hành.

Minimize: Cựa tiêu cửa sổ hiện hành.

MoveSize: di chuyển và định lại kích thước của cửa sổ.

Restore: khôi phục lại kích thước cửa sổ.

Beep: Phát âm thanh ra loa.

1.3 Gắn Macro với các sự kiện trên Form/Report

Macro có thể gắn với các sự kiện trên Form hay Report như nháy chuột, ấn phím, di chuyển chuột thực hiện trên bề mặt Form. Nhờ đó ta có thể dùng Form tạo một giao diện để sử dụng cho chương trình.

Giả sử trong CSDL **Sinhvien** trong bài tập của chương 4 đã mô tả hai bảng **HOSO** và **GIADINH**. Ta tạo các Query, Form, Report, Macro như sau:

- Query “**Truy vấn điểm trung bình**” gồm các cột MaSV, Ho, Ten, Khoa, GiaiTich, Daiso, Tinhoc, Triethoc, Anhvan, DiemTB(bảng điểm trung bình của cả 5 môn) lấy từ bảng HOSO.

- Form “**HoSo mainform**” để nhập dữ liệu cho hai bảng HOSO và GIADINH(trong mẫu biểu này có một mẫu biểu con là “GiaDinh Subform”)

- Report “**báo cáo điểm trung bình**”: lấy dữ liệu từ “truy vấn điểm trung bình” nhưng nhóm dữ liệu theo từng khoa, cuối mỗi nhóm có tính trung bình từng cột.

- Macro “**Mở truy vấn điểm trung bình**”: gồm hai hành động: cựa đại cửa sổ hiện hành, mở truy vấn “truy vấn điểm trung bình” ở dạng Datasheet.

- Macro “**Mở mẫu biểu HoSo mainform**”: gồm hai hành động: cựa đại cửa sổ hiện hành, mở mẫu biểu “Hoso mainform” ở dạng Datasheet.

- Macro “**Mở báo cáo điểm trung bình**”: gồm hai hành động: cựa đại cửa sổ hiện hành, mở báo biểu “báo cáo điểm trung bình”.

- Macro “**Kết thúc chương trình**” gồm hai hành động: hành động MsgBox để thông báo “bạn cần lưu dữ liệu trước khi kết thúc chương trình” có kiểu là “Warning!”, hành động Close dùng để đóng Form có tên là “**Bảng điều khiển chính**” (đây là Form giao diện chính của chương trình).

2. MENU BAR VÀ TOOLBAR

Ngoài việc tạo giao diện chương trình dùng các nút lệnh như phần trên đã trình bày. Ta còn một cách khoa học hơn đó là tổ chức giao diện thành các Menu. Phần này trình bày cách thức tạo các Menu cho chương trình ứng dụng như thế nào.

Ta tìm hiểu thông qua ví dụ sau:

Giả sử: có CSDL **QLTAISAN** gồm 3 bảng sau:

- **DMPhong** (**Maphong**, Tenphong)
- **DMTaisan**(**MaTS**, TenTS, Gia, Quicach, Donvi)
- **PhongPhoi**(**MaTS**, **Maphong**, Soluong, NgaySD)

Giả sử đã tạo được các Form và Report sau:

- **Form1**: Nhập dữ liệu cho hai bảng DMPhong, Phanphoi.
- **Form2**: Nhập dữ liệu cho hai bảng DMTaisan, Phanphoi.
- **Form3**: giới thiệu về chương trình, hình ảnh nhà cơ quan.
- **Report1**: Báo cáo từng phòng có các loại tài sản nào.
- **Report2**: Báo cáo từng tài sản hiện có mặt tại những phòng nào.

Vấn đề đặt ra là cần tạo một menu chính ngang gồm 3 mục, khi chọn một mục lại hiện ra một menu dọc tương ứng, chọn một mục của menu dọc sẽ chạy một Form hay Report tương ứng. Hệ thống menu như sau:

Nhập dữ liệu	In báo cáo	Hệ thống
Nhập DMPhong và PhanPhoi	Báo cáo tài sản trong phòng	Giới thiệu chương trình
Nhập DMTaisan và PhanPhoi	Tài sản có ở đâu	Về Access
		Về Windows

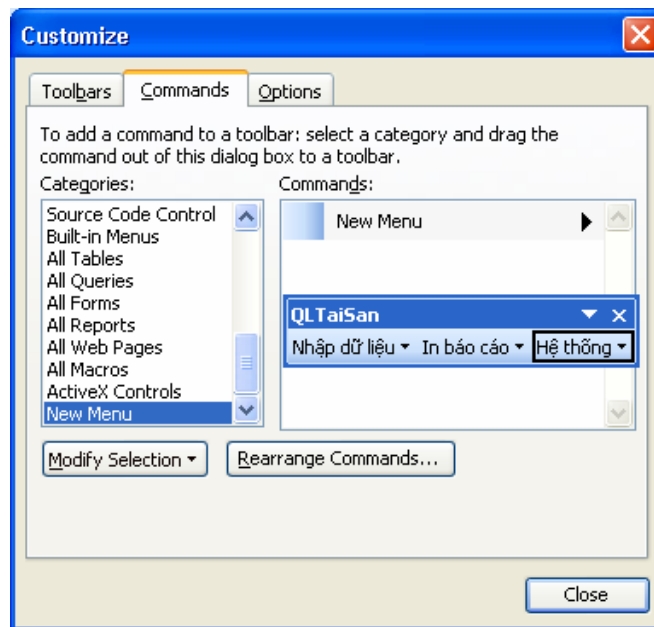
Quá trình thực hiện như sau:

Bước 1: Tạo một thanh Toolbar rỗng mới. Dùng lệnh **Tools/Customize**, xuất hiện hộp thoại Customize, chọn lớp Toolbars, nháy nút New.

Xuất hiện hộp thoại **New Toolbar**, vào tên thanh công cụ mới QLTaiSan, nháy OK, trên màn hình bắt đầu xuất hiện thanh công cụ rỗng và ta phải tiếp tục xây dựng các mục cho nó.

Bước 2: Xây dựng các mục cho menu ngang. Chọn lớp Commands của hộp thoại Customize, chọn mục New Menu trong khung Categories, ba lần kéo mục New Menu từ khung Commands và thả vào thanh công cụ mới đang thiết kế.

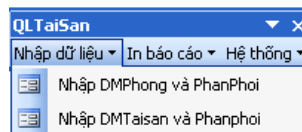
Thay đổi tiêu đề của ba mục trên menu chính: nháy phải lên mục New Menu đầu tiên, xuất hiện menu dọc, gõ tiêu đề mới vào hộp Name là **“Nhập dữ liệu”**. Tương tự mục thứ hai là **“In báo cáo”**, mục ba là **“Hệ thống”**. Như hình 6.2.



Hình 6.2

Bước 3: Xây dựng các menu dọc. Mỗi mục con của các menu dọc sẽ là một lời gọi tới một đối tượng như Table, Query, Form, Report hay Macro.

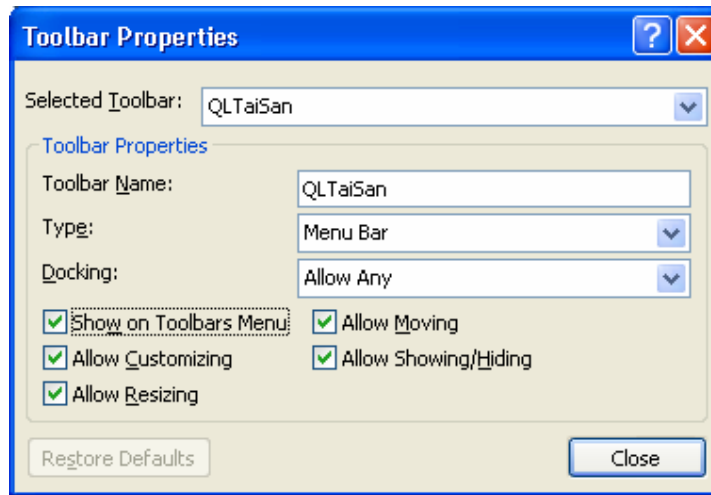
- Xây dựng menu dọc ứng với mục **“Nhập dữ liệu”**: nhấn nút mũi tên ở cuối mục để hiện mục trống đầu tiên của menu dọc. Trong cửa sổ Customize chọn mục All Forms trong khung Categories để chọn kiểu đối tượng, khung Commands hiện danh sách các Form hiện có trong CSDL, kéo Form1 từ khung này và thả vào mục trống của menu dọc cần xây dựng, thả tiếp Form2 vào dưới Form1 để tạo mục thứ 2. Đổi tên mục chọn thứ nhất thành “Nhập DMPHong và Phanphoi”, mục chọn thứ 2 là “Nhập DMTaisan và Phanphoi”. Ta được như hình 6.3



Hình 6.3

- Tương tự ta kéo Report1 và Report2 vào menu dọc ứng với các mục **“In báo cáo”**, kéo Form3 vào menu dọc ứng với mục **“Hệ thống”**. Mục về Access lấy ở File/Close, mục “Về Windows” lấy ở mục File/Exit.

Bước 4: Chọn lớp Toolbars của hộp thoại Customize, chọn thanh công cụ QLTaisan, nhấn nút Properties. Xuất hiện hộp thoại Toolbar Properties, mục Type đổi thành Menu Bar(để thanh công cụ vừa xây dựng trở thành menu chính của ứng dụng, ngầm định là Toolbar), chọn mục Show on Toolbars Menu. Như hình 6.4, đóng hộp thoại Toolbar Properties. Đóng tiếp hộp thoại Customize.



Hình 6.4

Bước 5: Tạo một mẫu biểu FormMain và gắn Menu Bar vừa xây dựng với mẫu biểu này.

Tạo một Form không có dữ liệu nguồn ở chế độ Design View. Xác định các thuộc tính của Form như sau:

Menu bar: **QLTaisan** (gắn kết Menu bar với mẫu biểu).

Caption: **Quan ly tai san** Default View: **Single Form**

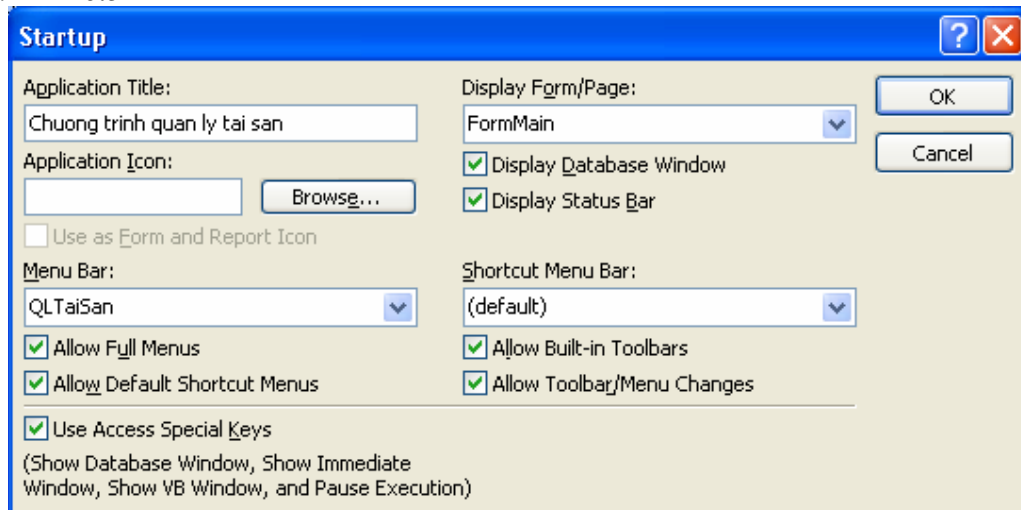
Record Selectors: **No** Navigation Buttons: **No**

Dividing Lines: **No**

On Open: nháy vào nút Build(có dấu ba chấm) soạn thảo đoạn lệnh VBA làm cực đại Form khi mở:

```
Private Sub Form_Open(Cancel As Integer)
    DoCmd.Maximize
End Sub
```

Bước 6: Qui định FormMain chạy ngay khi CSDL mở. Dùng lệnh Tools/Startup, xuất hiện hộp thoại Startup, mục Menu Bar chọn là QLTaisan, đặt tên cho chương trình ứng dụng là “Chương trình quan ly tai san”, hộp Display Form/Page chọn là FormMain. Như hình 6.5



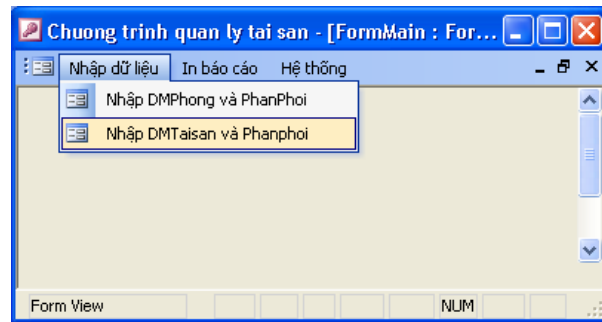
Hình 6.5

Khi CSDL Taisan mở ta được màn hình làm việc như hình 6.6.

Chú ý:

- Khi mở CSDL QLTaisan nếu vào cửa sổ Database thì phải ấn phím Shift trong khi mở CSDL.

- Muốn hủy bỏ Menu Bar “QLTaisan” ta làm như sau: nháy phải chuột lên thanh công cụ này, hiện menu tắt, chọn Customize hiện hộp thoại Customize chọn lớp Toolbars, chọn QLTaisan, nháy Delete.



Hình 6.6

BÀI TẬP CHƯƠNG 6

Xây dựng giao diện cho tất cả các ví dụ và bài tập của các chương trước.

CHƯƠNG 7

LẬP TRÌNH CƠ SỞ DỮ LIỆU TRONG MS ACCESS

1. GIỚI THIỆU LẬP TRÌNH VISUAL BASIC APPLICATION

MS Access không chỉ đơn thuần là một hệ quản trị cơ sở dữ liệu (CSDL) quan hệ mà nó còn cung cấp một môi trường lập trình với các công cụ khá đầy đủ, dễ sử dụng để phát triển các ứng dụng quản lý vừa và nhỏ.

Ngôn ngữ lập trình được phát triển trong MS Access là Access Basic. Tuy nhiên từ phiên bản MS Access for Windows 95, Access Basic được thay thế bởi Visual Basic (VB). Hai ngôn ngữ này khá giống nhau và đều được phát triển từ một thành phần thiết kế chung nhưng ngày nay, VB trở thành ngôn ngữ lập trình chung của chương trình ứng dụng MS Office bao gồm: Access, Excel, Word, PowerPoint. Việc có được một ngôn ngữ lập trình chung xuyên suốt mọi chương trình ứng dụng mang lại một số lợi điểm quan trọng là:

- Người lập trình chỉ cần biết một ngôn ngữ lập trình để tùy biến, phát triển ứng dụng.
- Dễ dàng hợp nhất các đối tượng trong các chương trình ứng dụng.

VB là ngôn ngữ có một số đặc điểm:

- Không phân biệt chữ hoa, thường
- Hướng sự kiện và hướng đối tượng
- Không hỗ trợ các công cụ thao tác với hệ thống như C, C⁺⁺

Phần trình bày này sẽ đề cập đến các nội dung sau:

- Module
- Các kiểu dữ liệu, hằng và biến
- Các cấu trúc điều khiển
- Hàm và thủ tục
- Các mô hình truy cập CSDL

2. MODULE

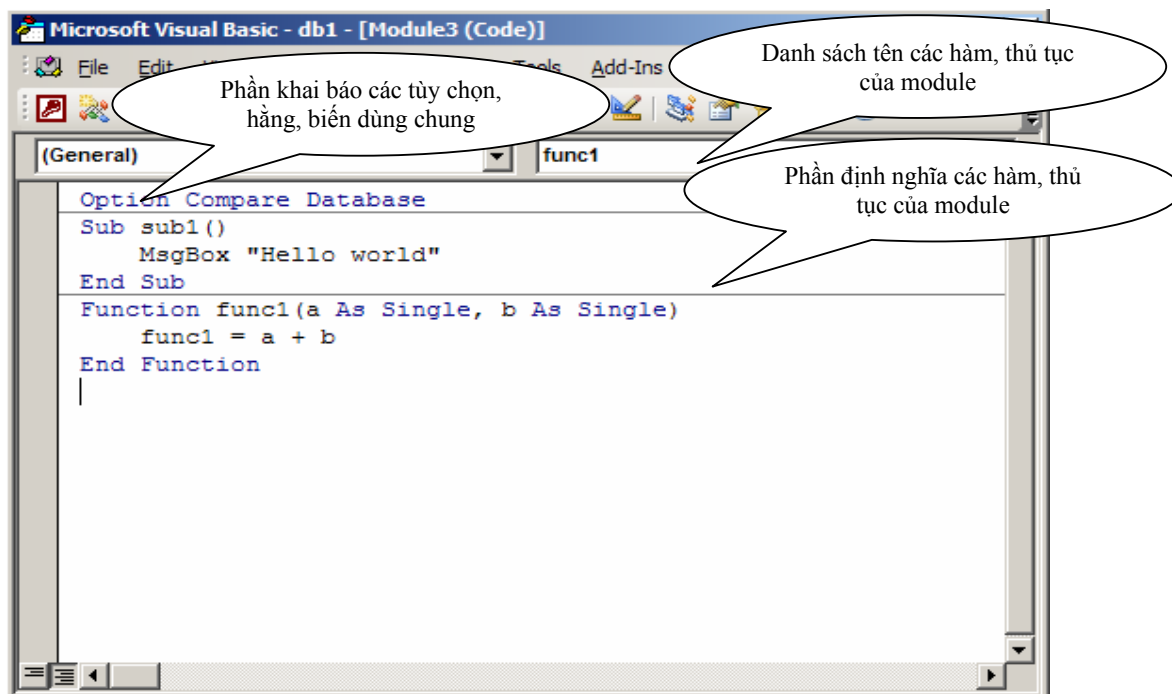
Module là một đối tượng nguyên thủy của môi trường lập trình VB. Toàn bộ mã lệnh VB trong CSDL được lưu trong module dưới dạng các thủ tục (gồm hàm và thủ tục con). Các thủ tục này có thể độc lập hoặc liên quan đến form/report.

MS Access cung cấp ba loại module

2.1 Module chuẩn (Standard Module)

Module này chứa các biến, thủ tục con có thể được gọi từ query, form, report, macro, biểu thức, thủ tục khác hoặc từ bất cứ đâu trong chương trình ứng dụng.

Để tạo module mới, trong cửa sổ thiết kế CSDL, chọn đối tượng *module*, sau đó chọn nút lệnh *new*. Kết quả nhận được là một cửa sổ có dạng như sau:



Hình 7.1: Cửa sổ code của module chuẩn

Trong cửa sổ này, dưới thanh tiêu đề là hai hộp danh sách thả xuống.

Hộp danh sách bên trái luôn có một mục là General, hộp danh sách bên phải là danh sách các hàm, thủ tục trong module.

Cửa sổ soạn thảo mã lệnh gồm hai phần cơ bản, phần khai báo các tùy chọn và các hằng, biến dùng chung cho các hàm, thủ tục trong module và phần định nghĩa các hàm, thủ tục trong module.

Phần khai báo các hằng, biến dùng chung cho các hàm, thủ tục trong module và phần định nghĩa các hàm, thủ tục trong module được trình bày chi tiết ở các phần sau.

Phần khai báo các tùy chọn ở đây có thể có các tùy chọn sau được khai báo:

- **Option Base Statement**

Khai báo chỉ số thấp nhất cho mảng trong toàn module, mặc định là 0.

Cú pháp khai báo:

```
Option Base {0 | 1}
```

Ví dụ: Khi định nghĩa một mảng theo cú pháp

```
Dim a(100) as Integer
```

Mặc định ta sẽ được một mảng tên là a, các chỉ số chạy từ 0 đến 99.

Nếu có tùy chọn Option Base 1 thì mảng a sẽ có 100 phần tử, chỉ số chạy từ 1 đến 100.

Tùy chọn này (nếu có) phải được khai báo trước bất kỳ hàm, thủ tục nào và nó chỉ có tác dụng trong module chứa nó. Option Base chỉ được khai báo một lần trong một module và phải trước các khai báo mảng.

- **Option Compare Statement**

Khai báo phương thức so sánh cho các biểu thức thuộc kiểu chuỗi (String).

Cú pháp khai báo:

```
Option Compare {Binary | Text | Database}
```

Option Compare Binary: so sánh chuỗi theo kiểu nhị phân, nghĩa là theo thứ tự sắp xếp của các ký tự trong bảng mã ASCII. Đây là kiểu mặc định

Ví dụ: A < B < C ... < a < b < c...

Option Compare Text: so sánh theo kiểu trật tự của các ký tự không phân biệt chữ hoa, thường.

Ví dụ: A = a, B = b, ..., Â = â, Ê = ê, ...

Option Compare Database: so sánh chuỗi dựa trên trật tự được xác định cục bộ trong Database chứa module đó.

- **Option Explicit Statement**

Khai báo để yêu cầu các biến phải được khai báo tường minh trước khi sử dụng. Tùy chọn này (nếu có) phải được đặt trước mọi khai báo biến và định nghĩa các hàm, thủ tục con.

Cú pháp khai báo:

```
Option Explicit
```

Ví dụ 1:

```
[1] Option Explicit
[2] Dim a
[3] a = 100
```

Trong đoạn mã này:

- + dòng [1]: yêu cầu các biến phải được khai báo tường minh trước khi sử dụng
- + dòng [2] khai báo một biến a
- + dòng [3] gán cho a giá trị 100

Như vậy, cuối cùng biến a được nhận giá trị là 100

Ví dụ 2:

```
[1] Option Explicit
[2] a = 100
```

Trong đoạn mã này:

- + dòng [1]: yêu cầu các biến phải được khai báo tường minh trước khi sử dụng
- + dòng [2]: gán cho biến a giá trị 100 mà không cần khai báo trước

Đoạn mã này khi dịch trình biên dịch sẽ thông báo lỗi “Variable not defined”. Và do đó cần khai báo biến a trước khi gán giá trị cho nó hoặc ta bỏ khai báo Option Explicit đi.

- **Option Private Statement**

Khai báo tùy chọn để cấm các truy xuất từ bên ngoài (các ứng dụng, dự án khác) vào các thành phần của module.

Cú pháp khai báo:

```
Option Private Module
```

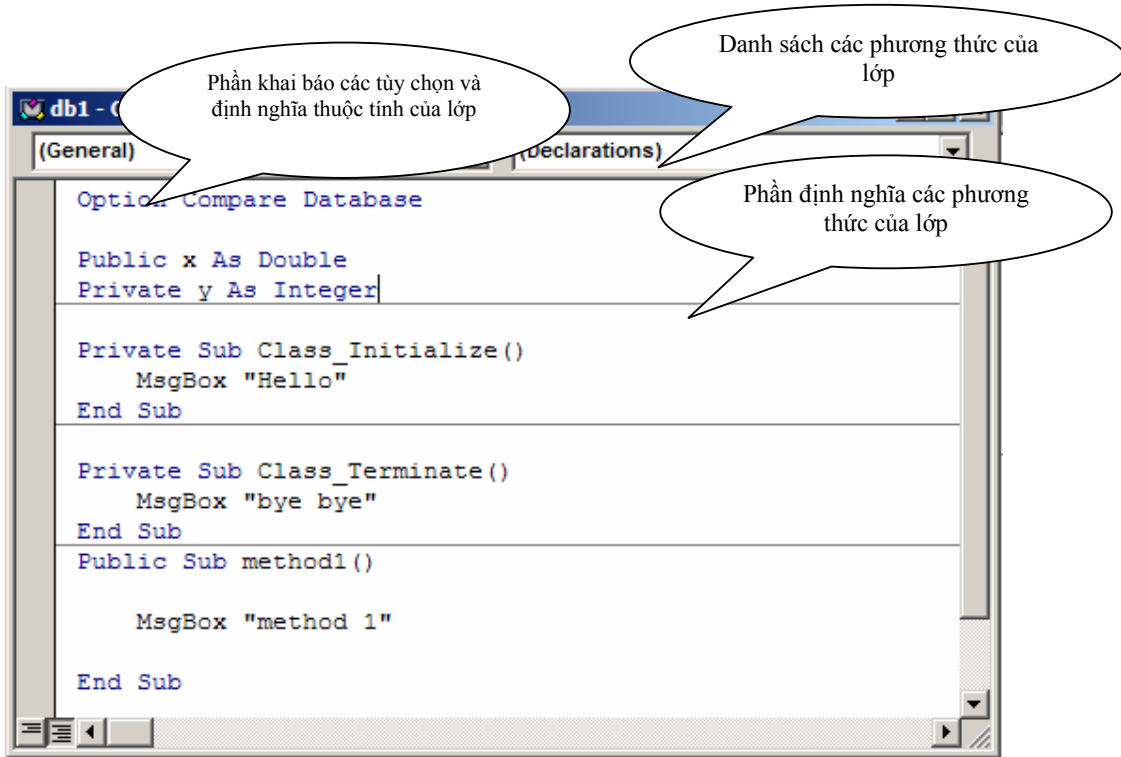
Chú ý: tùy chọn này chỉ cấm các truy cập từ các dự án (có thể trong cùng ứng dụng), ứng dụng khác tới các thành phần (hằng, biến, hàm, thủ tục, kiểu người dùng định nghĩa) của module. Mọi truy xuất từ các module, query, form, ... khác trong cùng cơ sở dữ liệu là được.

2.2 Module lớp (Class Module)

Có thể xem module lớp là loại module để định nghĩa lớp của người dùng. Lớp của người dùng cũng được đối xử tương tự như các lớp sẵn có. Mỗi module lớp có hai hàm mặc định là Class_Initialize() và Class_Terminate(). Khi tạo đối tượng thuộc module lớp, các hàm này sẽ được tự động gọi tương ứng khi đối tượng được thiết lập (bằng lệnh set) và hủy bỏ (tường minh hoặc không tường minh).

Để tạo module lớp, từ menu trong cửa sổ Database, chọn Insert/class module. Kết quả nhận được là một cửa sổ code của module lớp (hình 7.2). Cửa sổ này có cấu trúc giống cửa sổ code của module chuẩn.

Hộp danh sách bên phải là tên các phương thức của lớp và mục (Declarations).
 Phần bên dưới là cửa sổ khai báo các tùy chọn, định nghĩa các thuộc tính và phương thức của lớp.
 Phần khai báo gồm các khai báo như trong module chuẩn.
Chú ý: mỗi module lớp có một cửa sổ code riêng, tên đặt cho module lớp chính là tên của lớp do người dùng tạo ra.



Hình 7.2: Cửa sổ code của module lớp

2.3 Module của form/report (Form/report Module)

Mặc định tất cả các form/report đều có module ẩn sau nó (thuộc tính HasModule được thiết lập mặc định là True). Có thể dùng từ khóa Me khi tham chiếu đến các module ẩn sau form/report.

Trong form/report module chúng ta có thể tạo các thủ tục theo sự kiện (thủ tục được tự động gọi khi có sự kiện tương ứng với nó xảy ra), các thủ tục độc lập có thể được gọi từ các thủ tục khác trong cùng hoặc khác module.

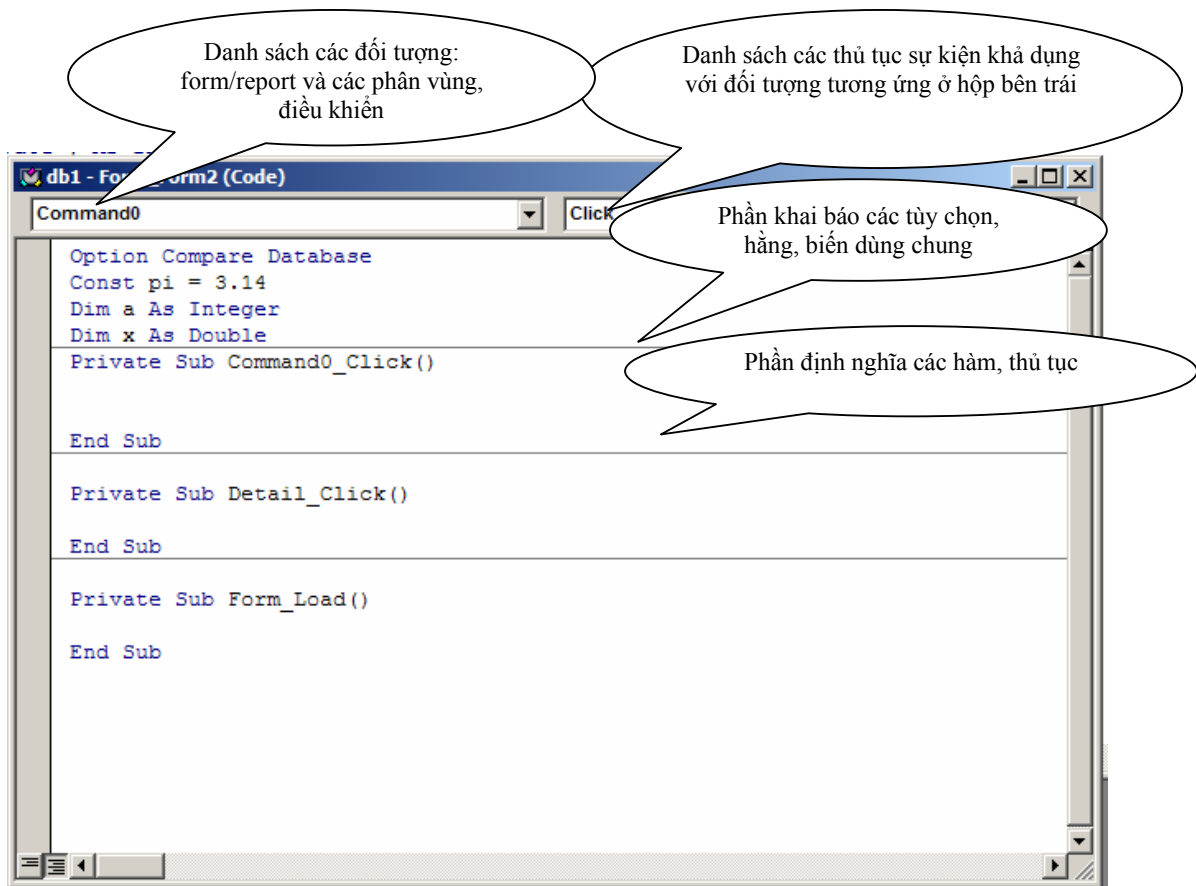
Để hiệu chỉnh form/report module trong cửa sổ Design của form/report chọn lệnh *view/code* trên thanh công cụ, kết quả ta nhận được màn hình có dạng như hình 7.3.

Cửa sổ này là nơi chúng ta viết mã cho các thủ tục của form/report module.

Tại cửa sổ code, ngay dưới thanh tiêu đề và thanh công cụ là hai hộp danh sách thả xuống:

- **Hộp danh sách các đối tượng (bên trái)**

Liệt kê form/report, phân mục và các điều khiển trên form/report có thể phát sinh sự kiện. Mục (General) là phần thiết lập các tùy chọn, khai báo hằng, biến dùng chung cho các thủ tục thuộc form/report.



Hình 7.3: Cửa sổ code của module cho form/report

Chú ý: Với mỗi form/report MS Access cung cấp một cửa sổ code riêng. Trong hộp danh sách các đối tượng: Đối tượng form/report và các vùng trên nó được chỉ định bởi mục form, report, phân vùng tương ứng; các điều khiển được chỉ định bởi tên của nó.

• **Hộp danh sách các thủ tục (bên phải)**

Liệt kê các thủ tục sự kiện khả dụng cho đối tượng tương ứng được chọn ở hộp danh sách các đối tượng. Các mục in đậm là thủ tục ta đã viết mã cho nó.

Phần khai báo các tùy chọn bao gồm các tùy chọn như trong module chuẩn. Để tạo các thủ tục độc lập trong form/report module ta có thể viết mã theo cú pháp tạo sub, function (chi tiết được trình bày sau) tại bất kỳ vị trí nào trong vùng soạn thảo của cửa sổ code, hoặc chọn lệnh *Insert/Procedure ...* trên thanh công cụ và nhập các tham số yêu cầu của hộp thoại nhận được.

Việc sử dụng form/report module mang lại các ưu điểm cơ bản sau so với module chuẩn:

+ Toàn bộ mã cần để tự động hóa form/report thường trú ngay trong form/report đó, không cần phải nhớ tên của đối tượng module riêng biệt liên quan đến form/report.

+ Các thủ tục hướng sự kiện trong module được tự động thi hành dựa trên các sự kiện xảy ra với các yếu tố, thành phần trên form/report, không cần phải xác định lời gọi tường minh cho chúng.

+ MS Access tải module chuẩn (được tham chiếu bởi đối tượng module) vào bộ nhớ khi chỉ cần có một tham chiếu đến thủ tục hoặc biến trong module và cho phép nó thường trú trong bộ nhớ chừng nào CSDL còn mở. Trong khi đó, form/report module chỉ được MS Access tải vào bộ nhớ khi form/report được mở. Khi form/report bị đóng lại thì form/report module tương ứng với nó cũng bị giải phóng khỏi bộ nhớ.

+ Khi xuất đi một form/report, toàn bộ mã lệnh trong form/report module tương ứng với nó cũng được kết xuất theo.

Tuy nhiên, form/report module cũng có nhược điểm là form/report có module lớn đi kèm sẽ được mở ra, đóng lại chậm hơn so với form/report có module nhỏ hoặc không có module đi kèm.

3. KIỂU DỮ LIỆU, HẰNG VÀ BIẾN

3.1 Kiểu dữ liệu

Một kiểu dữ liệu là tập hợp các giá trị mà một biến thuộc về kiểu đó có thể nhận được. Kiểu dữ liệu được đặc trưng bởi hai yếu tố:

- Tập các giá trị thuộc về nó
- Tập hợp các phép toán (toán tử) có thể được thực hiện trên nó

VB cung cấp các kiểu dữ liệu sau đây:

Kiểu dữ liệu	Kích thước (byte)	Ký tự phân loại	Có thể chứa
Byte	1	Không có	Số nguyên từ 0 đến 255
Integer	2	%	Số nguyên từ -2^{15} đến $2^{15} - 1$
Long	4	&	Số nguyên từ 2^{31} đến $2^{31} - 1$
Single	4	!	Số dấu chấm động từ $-3.4 \cdot 10^{38}$ đến $3.4 \cdot 10^{38}$
Double	8	#	Số dấu chấm động từ $-1.79 \cdot 10^{308}$ đến $1.79 \cdot 10^{308}$
Currency	8	@	Số nguyên được chia tỷ lệ với 4 số lẻ phân thập phân từ -922,337,203,685,477.5808 đến 922,337,203,685,477.5807
String	10 + 2 byte cho mỗi ký tự	\$	Chuỗi ký tự dài tối đa 2 tỷ byte, chuỗi có độ dài cố định có thể dài tối đa 65400 ký tự
Boolean	2	Không có	True hoặc False
Date	8	Không có	Giá trị ngày giờ từ 1/1/100 đến 31/12/9999
Object	4	Không có	Bất kỳ tham chiếu đối tượng nào
Variant	16 đến khoảng 2 tỷ byte	Không có	Bất kỳ loại dữ liệu nào

Có thể ngầm định định nghĩa kiểu dữ liệu cho biến bằng cách nối thêm một ký tự phân loại dữ liệu vào sau tên biến trong lần đầu tiên sử dụng biến.

Ngoài những kiểu dữ liệu VB đã xây dựng sẵn, nó còn cung cấp cho người dùng cú pháp để họ tự xây dựng kiểu dữ liệu của riêng mình, phù hợp với yêu cầu của ứng dụng. Đó là kiểu dữ liệu do người dùng định nghĩa (Kiểu cấu trúc).

Cú pháp khai báo:


```
[Private | Public]
Type varname
    elementname [(subscripts)] As type
    [elementname [(subscripts)] As type]
    .
    .
    .
End Type
```

Trong đó:

Public | Private: từ khóa xác định phạm vi truy xuất của kiểu dữ liệu tương ứng cho tất cả các thủ tục trong tất cả các module thuộc tất cả các dự án | chỉ trong phạm vi module chứa nó.

varname: tên kiểu dữ liệu được đặt theo quy tắc đặt tên của VB

elementname: tên các thành phần của kiểu dữ liệu

subscript: đặt chỉ số cho mảng nếu thành phần dữ liệu tương ứng là mảng

type: là một trong những kiểu của VB hoặc kiểu đã được định nghĩa trước.

Ví dụ:

```
Type StateData
    CityCode (1 To 100) As Integer      'Thành phần dữ liệu là một mảng
    100 phần tử có chỉ số từ 1 đến 100.
    County As String * 30              'Thành phần dữ liệu là một xâu
    tối đa là 30 ký tự
End Type
    'Khai báo một biến mảng 100 phần tử có chỉ số từ 1 đến 100, mỗi
    phần tử là một cấu trúc kiểu StateData.
Dim Washington(1 To 100) As StateData
```

Kiểu cấu trúc chỉ được định nghĩa trong phần khai báo của các loại module, không được định nghĩa trong các hàm, thủ tục con. Trong module chuẩn và module lớp, phạm vi truy xuất của kiểu cấu trúc mặc định là *public*.

Cú pháp truy xuất các thành phần của kiểu cấu trúc:

```
varname.elementname
```

Trong đó:

varname: là tên biến kiểu cấu trúc

elementname: tên thành phần của kiểu cấu trúc.

Ví dụ:

Với cấu trúc và khai báo biến của ví dụ trước, ta có các truy xuất hợp lệ sau:

```
Washington(1).CityCode(1) = 1
Washington(1).County = "100"
```

3.2 Hằng

Hằng là đại lượng có giá trị không đổi trong suốt thời gian chương trình thi hành.

Cú pháp khai báo:

```
[Public | Private] Const constname [As type] = expression
```

Trong đó:

Public | Private: từ khóa xác định phạm vi truy xuất của hằng tương ứng cho tất cả các thủ tục trong tất cả các module thuộc tất cả các dự án | chỉ trong phạm vi module chứa nó. Các từ khóa này chỉ được viết trong phần khai báo của module, không được viết trong phạm vi thủ tục, hàm (mặc dù hằng có thể khai báo trong thủ tục, hàm). Private là phạm vi truy xuất mặc định

constname: tên hằng

type: là một trong các kiểu dữ liệu [Byte](#), [Boolean](#), [Integer](#), [Long](#), [Currency](#), [Single](#), [Double](#), [Decimal](#), [Date](#), [String](#), hoặc [Variant](#). Nếu không có kiểu dữ liệu VB tự xác định kiểu phù hợp nhất cho hằng dựa vào giá trị của biểu thức.

Expression: biểu thức.

Ví dụ:

```
Public Const pi As Double = 3.14159
```

Hằng có thể được khai báo ở trong phạm vi toàn module (khai báo trong phần khai báo của module) hoặc trong thủ tục, hàm con của module. Hằng được khai báo ở đâu thì có tầm vực hoạt động ở đó.

3.3 Biến

Biến là đại lượng có thể thay đổi giá trị, biến thực chất là một vùng nhớ gồm các ô liên tiếp (số lượng ô nhớ tùy thuộc vào kiểu dữ liệu của biến).

VB cung cấp một số loại biến: biến thông thường khai báo bởi từ khóa *Dim*, biến công cộng khai báo bởi từ khóa *public*, biến tĩnh khai báo bởi từ khóa *static*

- **Khai báo biến thông thường dùng từ khóa Dim**

Cú pháp khai báo:

```
Dim [ WithEvents ] varname [ ( [ subscripts ] ) ] [ As [ New ] type ] [ ,  
[ WithEvents ] varname [ ( [ subscripts ] ) ] [ As [ New ] type ] ] ...
```

Trong đó:

WithEvents: biểu thị một biến đối tượng trong phạm vi một module lớp (class module) hồi đáp các sự kiện do đối tượng ActiveX kích hoạt. WithEvents chỉ hợp lệ trong module lớp. Không tạo biến mảng với WithEvents, không sử dụng từ khóa New cùng với WithEvents.

varname: tên biến được đặt theo quy tắc đặt tên của VB

subscripts: khai báo chiều của biến mảng nếu biến là một biến mảng. Tối đa VB cho phép mảng có tới 60 chiều.

New: tạo đối tượng một cách không tường minh, về sau không cần dùng lệnh set để gán tham chiếu tới đối tượng nữa.

type: là một trong các kiểu dữ liệu của VB hoặc kiểu cấu trúc người dùng đã định nghĩa. Nếu kiểu dữ liệu không được chỉ ra khi khai báo, VB sẽ tự gán kiểu Variant cho các biến đó. Khi biến được gán giá trị cụ thể, thì kiểu tương ứng với giá trị đó sẽ được xác định.

Biến có thể được khai báo ở cấp module (khai báo trong phần khai báo biến dùng chung của module), khi đó biến là công cộng có phạm vi hoạt động trong tất cả các thủ tục, hàm trong module. Biến có thể được khai báo trong thủ tục, hàm để có phạm vi cục bộ, khi đó biến có phạm vi hoạt động là cục bộ trong thủ tục, hàm chứa nó.

Chú ý: có thể khai báo các biến có cùng tên ở các mức độ khác nhau, khi đó việc truy xuất đến các biến cùng tên trong các thủ tục, hàm sẽ truy xuất đến biến cục bộ.

Ví dụ:

```
Dim i as Integer      `Khai báo biến i kiểu Integer  
Dim d(1 To 10, 1 To 100) `Khai báo mảng 2 chiều d, mỗi  
phần tử của mảng có kiểu là Variant  
Dim rec as New ADODB.Recordset `Khai báo và tạo một đối  
tượng Recordset.
```

Có thể khai báo mảng mà không xác định số chiều, kích thước ngay, khi đó số chiều, kích thước của mảng cần xác định lại bằng câu lệnh ReDim tại thời điểm chạy:

Cú pháp của câu lệnh như sau:

```
ReDim [Preserve] varname(subscripts) [As type] [,
varname(subscripts) [As type]]...
```

Trong đó:

Preserve: yêu cầu VB không khởi tạo lại các giá trị hiện có trong mảng

varname: tên biến

subscripts: xác định số chiều của mảng, tối đa là 60.

type: kiểu dữ liệu của VB hoặc kiểu cấu trúc đã được người dùng định nghĩa.

Ví dụ:

```
Dim b() As Integer
ReDim b(1 To 100) As Integer
b(1) = 6
```

Trong đoạn mã trên, trước khi gán giá trị cho b(1) ta cần phải định nghĩa lại số chiều và kích thước cho mảng, nếu không một lỗi sẽ xảy ra.

Không dùng ReDim để định nghĩa lại kiểu của các phần tử mảng ngay cả khi kiểu của chúng không được chỉ rõ trong lệnh Dim.

- **Khai báo biến công cộng dùng từ khóa public**

Có thể khai báo biến dùng chung cho tất cả các module thuộc tất cả các dự án theo cú pháp:

```
Public [WithEvents] varname([subscripts]) [As [New]
type] [, [WithEvents] varname([subscripts]) [As [New]
type]]...
```

Các thành phần trong cú pháp trên tương tự các thành phần trong cú pháp khai báo biến với *Dim*.

Biến được khai báo theo cú pháp trên sẽ được truy xuất ở khắp nơi trong cùng ứng dụng với điều kiện không có tùy chọn Option Private Module.

Không sử dụng được từ khóa *public* để khai báo các biến string có độ dài cố định trong module lớp hoặc khai báo các biến cục bộ trong thủ tục, hàm của module.

Để truy xuất đến các biến public trong module khác phải sử dụng cú pháp:

```
modulename.varname
```

Trong đó:

Modulename: tên module chứa biến public

Varname: tên biến được khai báo là public trong module

Ví dụ:

Trong phần khai báo của module3 ta có khai báo sau:

```
Public abc As Double
```

Trong thủ tục sub1() của module1 muốn truy xuất đến biến abc trên ta sử dụng cú pháp:

```
Sub sub1()
    Module3.abc = 100
    MsgBox Module3.abc
End Sub
```

- **Khai báo biến tĩnh**

Biến tĩnh là biến không bị khởi động lại khi thủ tục, hàm chứa nó được gọi ở những lần sau. Biến tĩnh chỉ được phép khai báo ở phạm vi trong thủ tục, hàm

Cú pháp:

```
Static varname([subscripts]) [As [New] type] [,
varname([subscripts]) [As [New] type]] . . .
```

Trong cú pháp trên các tham số của lệnh tương tự các tham số trong cú pháp Dim.

Ví dụ:

```
Sub sub1()  
    Static a As Integer  
    a = a + 1  
    MsgBox "a = " & a  
End Sub
```

Trong ví dụ trên, nếu bạn gọi sub1() lần đầu tiên, hộp MsgBox sẽ thông báo a = 1 (VB khởi tạo các giá trị cho các biến kiểu số khi khai báo là 0). Ta gọi sub1() lần thứ 2, hộp MsgBox sẽ thông báo a = 2 (giá trị của biến tĩnh a không bị khởi tạo lại khi ta gọi ở những lần tiếp theo). Tiếp tục ta gọi sub1() lần thứ 3, hộp MsgBox sẽ thông báo a = 3,

Giá trị của biến tĩnh chỉ bị khởi tạo khi module bị reset hoặc restart. Biến tĩnh trong module lớp được bảo toàn giá trị đối với mỗi thể hiện của lớp và chỉ bị khởi tạo lại giá trị khi thể hiện của lớp bị hủy bỏ. Biến tĩnh trong form/report module được bảo toàn giá trị cho đến khi form/report đóng.

- **Gán giá trị cho biến**

Cú pháp gán giá trị cho biến:

```
varname = expression
```

Trong đó:

varname: là tên biến

expression: biểu thức, nếu giá trị của biểu thức không phù hợp với kiểu của biến VB sẽ tự ép kiểu cho phù hợp (đối với những trường hợp có thể ép kiểu không tương minh).

Ví dụ:

```
Dim i as Integer  
i = 9.76 'Khi đó i thực sự nhận giá trị là 10
```

Đối với các biến đối tượng (biến đối tượng là biến dùng để tham chiếu đến đối tượng không phải là bản sao của đối tượng), để tham chiếu tới một đối tượng thông qua biến thì biến đó phải được thiết lập tham chiếu đến đối tượng theo cú pháp:

```
set varname = object
```

Ví dụ:

```
Dim db As Database  
Dim rec As Recordset  
Set db = CurrentDb()  
Set rec = db.OpenRecordset("SELECT * FROM table1")
```

Trong ví dụ trên db và rec là hai biến đối tượng.

4. CÁC CẤU TRÚC ĐIỀU KHIỂN

4.1 Cấu trúc tuần tự

Cấu trúc tuần tự quy định trình tự thực hiện của các câu lệnh, lệnh nào được thực hiện trước, lệnh nào được thực hiện sau. Trong VB, cấu trúc tuần tự được thể hiện bằng mã xuống dòng (Hex: 0D0A) được sinh ra khi ta nhấn phím enter trong khi soạn thảo mã lệnh và dấu hai chấm (:).

Trong VB một dòng văn bản có thể có nhiều câu lệnh, khi đó các câu lệnh được phân cách nhau bởi dấu hai chấm (:). Một câu lệnh trong VB cũng có thể được viết trên

nhiều dòng, khi đó để nối câu lệnh được viết trên các dòng khác nhau ta cần dùng ký tự gạch dưới (_).

Ví dụ:

```
Sub tinhong(a, b, c)
  Dim tong As Double
  tong = a + b + _
  c
  MsgBox "Tong = " & tong
End Sub
```

4.2 Cấu trúc rẽ nhánh

Cú pháp:

```
If condition Then [statements] [Else statements]
```

Hoặc:

```
If condition Then
  [statements]
[ElseIf condition Then
  [statements]]...
[Else
  [statements]]
End If
```

Trong đó:

condition: là biểu thức chuỗi hoặc số mà VB có thể đánh giá là True (khác 0) hoặc False (0 hoặc Null). Condition cũng có thể là phép thử *TypeOf* objectname *Is* objecttype.

statements: một hoặc nhiều câu lệnh.

Nếu condition = True thì khối lệnh sau Then được thực hiện.

Nếu condition = False thì khối lệnh sau Else (nếu có) được thực hiện.

Sau đó VB thi hành câu lệnh tiếp theo sau if.

Ví dụ:

```
Sub ptb1(a, b)
  If (a = 0) Then
    If (b = 0) Then
      MsgBox "Phuong trinh vo so nghiem!"
    Else
      MsgBox "Phuong trinh vo nghiem!"
    End If
  Else
    MsgBox "Phuong trinh co 1 nghiem la: " & Format_
(-b / a, "0.00")
  End If
End Sub
```

Thủ tục ptb1(a, b) là để giải phương trình bậc 1. Có thể chạy thủ tục này tại cửa sổ immediate (View/Immediate window) bằng lệnh call ptb1(3, -1) để kiểm tra lời giải.

Chú ý: có thể đặt nhiều hơn một câu lệnh sau then trên cùng một dòng với If ... Then. Khi đó các câu lệnh được phân cách nhau bởi dấu hai chấm (:).

Ví dụ: đổi giá trị của hai biến a và b.

```
Dim a, b, tg
a = 5
b = 1
```

```
If a > b Then tg = a: a = b: b = tg
```

Sau khi các lệnh trên được thực hiện, ta có $a = 1$ và $b = 5$

Cú pháp `if ... then [else ...] end if` luôn cho ta rẽ vào một trong hai nhánh dựa vào biểu thức điều kiện. Trong trường hợp muốn rẽ vào một trong nhiều nhánh ta cần sử dụng cấu trúc `if ... then [else ...] end if` lồng nhau. Tuy nhiên, việc lồng nhau của cấu trúc này làm cho chương trình không minh bạch, dễ bị nhầm lẫn. VB cung cấp một cú pháp rẽ nhánh khác khắc phục nhược điểm này, đó là cú pháp `select case ... end select`

Cú pháp:

```
Select Case testexpression
[Case expressionlist
  [statements]]
[Case expressionlist
  [statements]] ...
[Case Else
  [statements]]
End Select
```

Trong đó:

testexpression: là biểu thức chuỗi hoặc số mà VB có thể đánh giá là True (khác 0) hoặc False (0 hoặc Null)

expressionlist: danh sách các biểu thức, phải có nếu có mệnh đề *case*. Danh sách các biểu thức có thể là:

+ Một biểu thức đơn lẻ

+ *Biểu thức To biểu thức* (To là từ khóa xác định một miền các giá trị, khi đó các giá trị phải được sắp theo thứ tự tăng).

Ví dụ: 1 To 10: là các giá trị trong đoạn [1, 10]

+ *Is toán tử so sánh biểu thức* (Is là từ khóa, kết hợp với toán tử so sánh (trừ Is và Like) để xác định một miền các giá trị).

Ví dụ: Is > 100: là các giá trị số phải lớn hơn 100

Mỗi biểu thức trong danh sách các biểu thức được phân cách nhau bởi dấu phẩy (,).

statements: 1 hoặc nhiều lệnh

Nếu **testexpression** bằng (hoặc thuộc) bất kỳ **expressionlist** trong mệnh đề *case* nào thì khối lệnh sau mệnh đề *case* đó được thực hiện. Trường hợp ngược lại khối lệnh trong mệnh đề *else* (nếu có) được thực hiện, sau đó VB sẽ thi hành câu lệnh tiếp theo sau `select case`.

Nếu có nhiều **expressionlist** trong các mệnh đề *case* thỏa mãn **testexpression** thì chỉ có khối lệnh trong mệnh đề *case* đầu tiên được thực hiện.

Ví dụ: hàm tính số ngày của một tháng trong năm

```
Function songay(th, nam)
  Dim sn
  Select Case th
    Case 1, 3, 5, 7, 8, 10, 12
      sn = 31
    Case 4, 6, 9, 11
      sn = 30
    Case Else
```

```

        If (nam Mod 4 = 0) Then sn = 29 Else sn = 28
    End Select
    songay = sn
End Function

```

Có thể chạy hàm này ngay trong cửa sổ immediate bằng lệnh: ?songay(2, 2000). Kết quả bạn nhận được sẽ là 29.

4.3 Cấu trúc lặp

Có hai cấu trúc lặp: lặp với số lần xác định và lặp dựa trên biểu thức điều kiện

- **Lặp với số lần xác định**

Cú pháp 1:

```

For counter = start To end [Step step]
    [statements]
[Exit For]
    [statements]
Next [counter]

```

Trong đó:

counter: biến đếm, kiểu số

start: giá trị khởi tạo của biến đếm

end: giá trị cuối cùng mà biến đếm có thể nhận

step: sau mỗi lần lặp biến đếm được cộng thêm vào giá trị trong step. Nếu không được thiết lập, mặc định step = 1.

statements: 1 hoặc nhiều lệnh

Khi counter còn thuộc phạm vi giữa start và end thì khối lệnh statements còn được thực hiện. Sau mỗi bước của vòng lặp, biến đếm sẽ được tăng (giảm) step giá trị.

Chú ý: Nếu bỏ qua counter trong mệnh đề Next, vòng lặp vẫn được thực thi bình thường. Tuy nhiên, nếu có nhiều for lồng nhau, counter trong mệnh đề Next không tương ứng với counter trong for thì một lỗi sẽ xảy ra khi biên dịch.

Ví dụ: hàm tính n!

```

Function gt(n)
    Dim t As Double
    t = 1
    Dim i As Long
    For i = 1 To n Step 1
        t = t * i
    Next
    gt = t
End Function

```

Cú pháp 2:

```

For Each element In group
    [statements]
[Exit For]
    [statements]
Next [element]

```

Trong đó:

element: biến được sử dụng để lặp thông qua các phần tử của mảng hay tập hợp. Đối với tập hợp, biến có thể là biến Variant hoặc biến đối tượng. Đối với mảng, biến chỉ có thể là biến Variant.

group: tên tập hợp hoặc mảng

statements: một hoặc nhiều lệnh

Khối lệnh **statements** trong **for** được thực hiện khi có ít nhất 1 element thuộc **group**. Khối lệnh này được lặp lại cho đến khi không còn element nào thuộc **group**.

Lần đầu vào vòng lặp, element được gán bằng phần tử đầu tiên của tập hợp, mảng. Sau mỗi lần lặp, element lại được gán bằng phần tử tiếp theo của tập hợp, mảng.

Cú pháp này rất hữu ích khi ta không biết số lượng các phần tử của mảng, tập hợp.

Ví dụ:

```
Dim a(10) As Integer
Randomize (100)
Dim i As Integer
For i = 0 To 9
    a(i) = Rnd * 100
Next
For Each e In a
    MsgBox e
Next
```

Đoạn mã trên tạo một mảng **a** gồm 10 phần tử và điền các giá trị ngẫu nhiên nhỏ hơn 100 vào mảng, sau đó in ra mỗi phần tử của mảng

- **Lặp dựa trên biểu thức điều kiện**

Cú pháp 1:

```
Do [{While | Until} condition]
    [statements]
[Exit Do]
    [statements]
Loop
```

Hoặc:

```
Do
    [statements]
[Exit Do]
    [statements]
Loop [{While | Until} condition]
```

Trong đó:

Condition: biểu thức số hoặc chuỗi mà VB có thể đánh giá được là True (khác 0) hoặc False (0 hoặc Null)

Statements: một hoặc nhiều lệnh

Khối lệnh trong **Do** được thực hiện trong khi điều kiện (**condition**) còn đúng (True) hoặc cho đến khi đúng.

Ví dụ:

```
Function ngT(n As Long)
    if (n = 1) Or (n = 2) Then ngT = True
    Dim i, t As Long
    Dim found As Boolean
    i = 2: t = Sqr(n): found = False
    Do While (i <= t) And (Not found)
        If n Mod i = 0 Then found = True
        i = i + 1
    Loop
    If found Then ngT = False Else ngT = True
End Function
```


Hàm kiểm tra ngT(), kiểm tra một số nguyên n có phải là nguyên tố hay không. Nếu n là nguyên tố, hàm trả về giá trị True, ngược lại, hàm trả về giá trị False.

Cú pháp 2:

```
While condition
    [statements]
Wend
```

Trong đó 2 tham số condition và statements như trong cú pháp Do.

Khối lệnh trong while còn được thực hiện khi điều kiện còn đúng.

Ví dụ: viết lại hàm kiểm tra số nguyên tố ở ví dụ trước

```
Function ngT(n As Long)
    If (n = 1) Or (n = 2) Then ngT = True
    Dim i, t As Long
    Dim found As Boolean
    i = 2: t = Sqr(n): found = False
    While (i <= t) And (Not found)
        If n Mod i = 0 Then found = True
        i = i + 1
    Wend
    If found Then ngT = False Else ngT = True
End Function
```

5. HÀM VÀ THỦ TỤC CON

VB cung cấp hai loại thủ tục: hàm (function) và thủ tục con (subroutine). Mỗi dạng thủ tục đều có thể có tham số. Sự khác nhau giữa hàm và thủ tục con là:

- Hàm có thể trả về một giá trị còn thủ tục con thì không. Vì vậy, trước khi kết thúc định nghĩa hàm thường có câu lệnh: tên hàm = biểu thức để trả giá trị về cho lời gọi hàm.
- Hàm có thể được thi hành từ mọi nơi. Thủ tục con chỉ có thể gọi thi hành từ một hàm, thủ tục con khác hoặc ở dạng thủ tục đáp ứng sự kiện trong form/report.

5.1 Hàm

Khai báo:

```
[Public | Private | Friend] [Static] Function name
[(arglist)] [As type]
    [statements]
    [name = expression]
[Exit Function]
    [statements]
    [name = expression]
End Function
```

Trong đó:

Public: quy định hàm có thể được truy xuất từ khắp nơi trong tất cả các module thuộc tất cả các dự án. Nếu hàm được viết trong module có khai báo tùy chọn On Private Module thì không thể được truy xuất bên ngoài dự án chứa nó.

Private: quy định phạm vi truy xuất của hàm là chỉ trong module chứa nó

Friend: chỉ được sử dụng trong module lớp. Quy định hàm được truy xuất khắp nơi trong dự án.

Static: để bảo toàn tất cả giá trị của các biến được khai báo trong phạm vi hàm tường minh hoặc không tường minh, với điều kiện module chứa hàm còn mở. Điều này tương đương với việc dùng từ khóa static khai báo toàn bộ các biến có trong hàm.

name: tên hàm

arglist: danh sách các tham số, được phân cách nhau bởi dấu phẩy (.). Arglist có dạng:

```
[Optional] [ByVal | ByRef] [ParamArray] varname[( )] [As type] [= defaultvalue]
```

Trong đó:

Optional: chỉ định rằng tham số là tùy chọn. Nếu khai báo một tham số là Optional thì tất cả các tham số khác thêm sau tham số đó cũng phải được khai báo là Optional.

ByVal: chỉ định tham số được tham chiếu theo kiểu tham trị (VB sẽ chuyển bản sao của giá trị của đối số đến hàm khi gọi hàm).

ByRef: chỉ định tham số được tham chiếu theo kiểu tham biến (VB sẽ chuyển địa chỉ của đối số đến hàm khi gọi hàm). ByRef là tham chiếu mặc định. Tham số là mảng luôn được tham chiếu là ByRef.

ParamArray: chỉ được khai báo cho tham số cuối cùng trong danh sách các tham số của hàm. Chỉ định tham số cuối cùng là một mảng *Optional* các phần tử kiểu Variant.

defaultvalue: giá trị mặc định của tham số. Giá trị ở đây là một hằng hoặc biểu thức hằng. Tùy chọn này chỉ dành cho các tham số Optional, nếu type là Object thì defaultvalue chỉ có thể là nothing.

type: là một trong các kiểu của VB hoặc kiểu cấu trúc do người dùng định nghĩa. Nên khai báo type trước mỗi tham số của hàm.

statements: một hoặc một số lệnh

expression: biểu thức, là giá trị trả về của hàm.

Hàm có thể được gọi ngay trong khi định nghĩa nó, đó là định nghĩa hàm kiểu đệ quy.

Có thể gọi hàm tại bất kỳ đâu mà phạm vi truy xuất của nó là được phép. Hàm tham gia vào một thành phần của biểu thức. Khi gọi hàm tại các vị trí trong cùng module, ta chỉ cần chỉ ra tên hàm và danh sách các giá trị, biến tương ứng với các tham số của nó.

Khi gọi hàm bên ngoài module, tên hàm được xác định:

- **Tên module.tên hàm** đối với hàm trong module chuẩn
- **Form_tên form.tên hàm** đối với hàm trong form module
- **Report_tên report.tên hàm** đối với hàm trong report module
- **Tên biến tượng.tên hàm** đối với hàm trong module lớp.

5.2 Thủ tục con

Khai báo:

```
[Private | Public | Friend] [Static] Sub name [(arglist)]  
    [statements]  
    [Exit Sub]  
    [statements]  
End Sub
```

Trong đó các tham số của lệnh tương tự các tham số của hàm.

Thủ tục con cũng có thể được gọi đệ quy ngay trong khi định nghĩa.

Cách gọi thủ tục con cũng tương tự hàm. Tuy nhiên thủ tục con được gọi như một lệnh.

6. CÁC MÔ HÌNH TRUY CẬP CƠ SỞ DỮ LIỆU

6.1 Kiến trúc chương trình ứng dụng MS Access

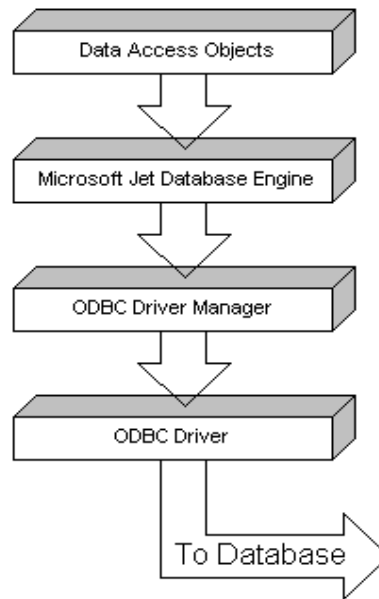
MS Access có hai thành phần: động cơ chương trình ứng dụng (application engine), kiểm soát việc lập trình và giao diện người dùng; Jet DBEngine, kiểm soát việc lưu trữ dữ liệu và thủ tục định nghĩa tất cả các đối tượng trong cơ sở dữ liệu. VB hỗ trợ hai mô hình này tương đối độc lập để thao tác những đối tượng được lưu bởi động cơ cơ sở dữ liệu.

Khi mở cơ sở dữ liệu, động cơ chương trình ứng dụng nạp tập hợp đối tượng thích hợp từ CSDL và file chương trình ứng dụng để liệt kê toàn bộ bảng, truy vấn, mối quan hệ (relationship), mẫu biểu (form), báo biểu (report), macro, module, trang truy cập dữ liệu để hiển thị trong cửa sổ Database. Động cơ chương trình ứng dụng thiết lập đối tượng Application ở cấp cao nhất chứa một tập hợp Forms (tập các mẫu biểu đang mở), tập hợp Reports (tập các báo biểu đang mở), tập hợp References (tất cả các phép tham chiếu đến thư viện VB) và tập hợp Modules (tập các module đang mở, bao gồm cả form/report module). Mỗi mẫu, báo biểu lại chứa tập hợp Controls gồm tất cả các bộ phận điều khiển trên mẫu, báo biểu.

6.2 Kiến trúc DAO (Data Access Objects)

Mô hình đầu tiên và cũ nhất trong hai mô hình có thể dùng để tìm kiếm dữ liệu và kiểm tra hoặc tạo đối tượng dữ liệu mới là mô hình DAO. Mô hình này được sử dụng thích hợp nhất trong phạm vi chương trình ứng dụng CSDL MS Access vì nó cung cấp các đối tượng, phương thức, thuộc tính phù hợp với cách thức mà MS Access và Jet DBEngine làm việc với nhau.

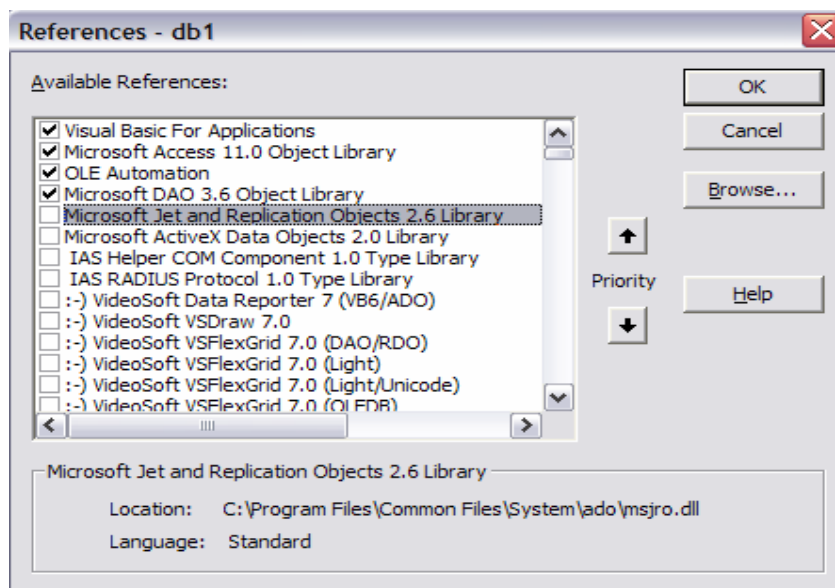
Sơ đồ sau minh họa mối quan hệ giữa DAO và các thành phần dữ liệu khác



Hình 7.4: Mối quan hệ giữa DAO và các thành phần khác

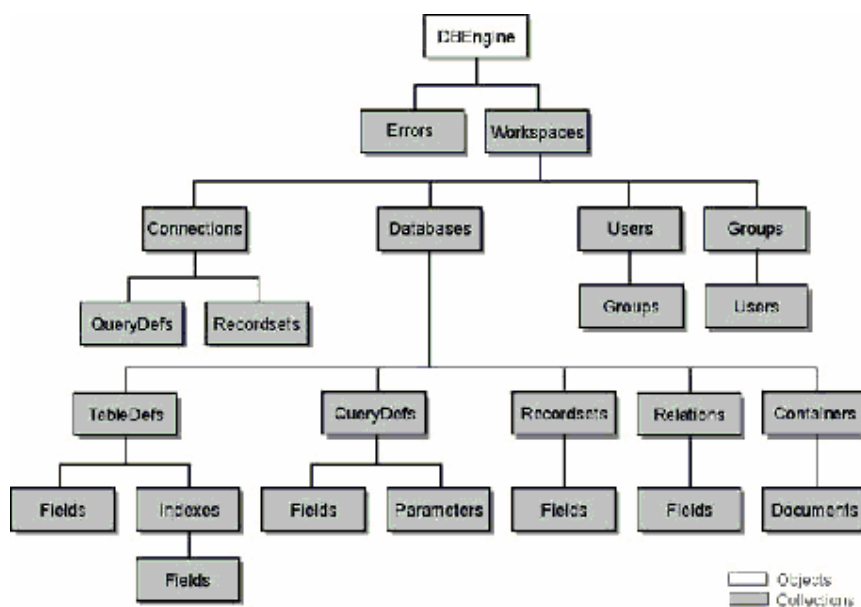
DAO sử dụng động cơ CSDL Microsoft Jet Database Engine để kết nối với server CSDL thông qua Open Database Connectivity (ODBC) driver hoặc driver riêng của server CSDL.

Để tham chiếu đến mô hình DAO, ta phải yêu cầu VB nạp một tham chiếu đến Microsoft DAO 3.6 Object Library bằng cách mở module bất kỳ, trên thanh công cụ chọn lệnh: *Tool/Reference*, kết quả nhận được là hộp thoại có dạng như hình dưới, sau đó chọn vào mục *Microsoft DAO 3.6 Object Library*.



Hình 7.5: Tham chiếu đến Microsoft DAO 3.6 Object Library

Sơ đồ sau minh họa kiến trúc phân tầng gồm nhiều thành phần, đối tượng trong mô hình DAO.



Hình 7.6: Các đối tượng và tập hợp trong kiến trúc DAO

1. DBEngine

Thuộc tính DBEngine của đối tượng Application được dùng như một cầu nối giữa động cơ chương trình ứng dụng và Jet DBEngine. Thuộc tính DBEngine đại diện cho đối tượng DBEngine – vốn là đối tượng cao cấp nhất trong hệ thống DAO.

Đối tượng DBEngine kiểm soát tất cả các đối tượng CSDL trong CSDL thông qua một hệ thống phân cấp tập hợp, đối tượng và thuộc tính. Khi mở một CSDL MS Access, trước tiên đối tượng DBEngine thiết lập tập hợp Workspace và một đối tượng Workspace mặc định. Nếu nhóm làm việc của bạn được bảo mật, MS Access nhắc bạn nhập UID và Password để đối tượng DBEngine có thể tạo một đối tượng người dùng (user) và một đối tượng nhóm (group) ở vùng làm việc mặc định. Nếu nhóm làm việc không được bảo mật, DBEngine tạo một tài khoản người dùng mặc định là Admin trong nhóm mặc định Admins.

Cuối cùng, DBEngine tạo đối tượng Database trong phạm vi đối tượng Workspace mặc định. DBEngine sử dụng thông tin đối tượng User và/hoặc Group hiện hành để quyết định bạn có thẩm quyền truy cập các đối tượng trong CSDL hay không.

Sau khi DBEngine tạo đối tượng Database, động cơ chương trình ứng dụng kiểm tra các tùy chọn khởi động CSDL để hiển thị mẫu biểu khởi động, thanh menu, thanh tiêu đề hay sử dụng một hoặc nhiều tùy chọn khởi động khác.

2. Errors

Chứa tất cả các đối tượng Error trong DAO. Khi thi hành bất kỳ một đối tượng nào trong mô hình DAO đều có thể sinh ra một hoặc nhiều lỗi. Mỗi khi một lỗi xảy ra, VB tạo một hoặc nhiều đối tượng Error và đưa vào trong tập Errors. Khi một đối tượng trong DAO được thi hành và sinh ra lỗi thì tập hợp Errors hiện tại bị xóa và các đối tượng Error mới được sinh ra tương ứng với lỗi đó được đưa vào trong tập Errors.

3. Workspaces

Định nghĩa một tập hợp các vùng làm việc (đang hoạt động hoặc ẩn) của DBEngine. Có thể khai báo nhiều vùng làm việc một lúc. Khi lần đầu tham chiếu hoặc sử dụng đối tượng Workspace, VB sẽ tự động tạo Workspace mặc định

DBEngine.Workspaces(0). Để tham chiếu đến đối tượng Workspace, bạn sử dụng một trong các cú pháp sau:

```
DBEngine.Workspaces(0)
DBEngine.Workspaces("name")
DBEngine.Workspaces![name]
```

4. Connections

Chứa các đối tượng Connection hiện tại của đối tượng Workspace. Khi mở một đối tượng Connection mới, nó sẽ tự động được thêm vào tập các Connections của đối tượng Workspace. Khi một đối tượng Connection bị đóng, nó sẽ được loại bỏ khỏi tập các Connections.

Tại thời điểm mở một Connection thì một đối tượng Database tương ứng được tạo ra và thêm vào tập Databases trong cùng một Workspace và ngược lại. Khi đóng một Connection thì đối tượng Database tương ứng với nó bị xóa khỏi tập Databases và ngược lại.

Để tham chiếu đến đối tượng Connection trong tập Connections chứa nó bạn có thể sử dụng một trong các cú pháp sau:

```
Connections(0)
Connections("name")
Connections![name]
```

name: là một chuỗi ký tự chứa đường dẫn đến file CSDL.

5. Database

Chứa tập các đối tượng Database được mở hoặc tạo ra thuộc 1 đối tượng Workspace. Khi tạo mới hoặc mở một đối tượng Database đã có, VB tự động nó thêm vào tập Databases. Khi xóa một đối tượng Database, nó cũng bị xóa bỏ khỏi tập Databases. Có thể tham chiếu đến đối tượng Database trong tập Databases chứa nó bởi một trong các cú pháp sau:

```
Databases(0)
Databases("name")
Databases![name]
```

name: là chuỗi ký tự chứa đường dẫn đến file CSDL.

Ví dụ: Đoạn mã sau để mở 1 CSDL có tên là DB1.MDB trong thư mục D:\Data\Mdb

```
Dim db As DAO.Database
Set db = DBEngine.OpenDatabase("D:\Data\Mdb\DB1.MDB")
```

Để mở CSDL hiện tại, có thể dùng lệnh:

```
Set db = CurrentDB
```

Đóng CSDL vừa mở bạn dùng lệnh:

```
db.Close
```

6. Users

Chứa các đối tượng User trong Workspace hoặc Group. Có thể tham chiếu đến đối tượng User bởi một trong các cú pháp sau:

```
[workspace | group].Users(0)
[workspace | group].Users("name")
[workspace | group].Users![name]
```

name: là tên của user

7. Groups

Chứa tập các đối tượng Group của Workspace. Có thể tham chiếu đến đối tượng Group bởi một trong các cú pháp sau:

```
Groups(0)
```

```
Groups ("name")
Groups! [name]
```

name: là tên của Group

8. Recordsets

Recordsets chứa tập các đối tượng Recordset. Mỗi đối tượng Recordset chứa tập các bản ghi của một bảng, truy vấn hoặc tập các bản ghi là kết quả của câu lệnh SQL.

Một số thuộc tính, phương thức của đối tượng Recordset

+ **AbsolutePosition**

Thuộc tính này cho biết vị trí tuyệt đối của bản ghi hiện tại tính từ bản ghi đầu tiên có thứ tự là 0.

+ **EOF**

Thuộc tính này cho biết con trỏ bản ghi có ở vị trí cuối file hay không. Nếu ở cuối file, giá trị trả về là True, ngược lại là False

+ **BOF**

Thuộc tính này cho biết con trỏ bản ghi có ở vị trí đầu file hay không. Nếu ở đầu file, giá trị trả về là True, ngược lại là False.

+ **RecordCount**

Trả về tổng số bản ghi của Recordset

+ **Fields**

Dùng để tham chiếu đến các field trong Recordset

Ví dụ: Tham chiếu đến field Hoten

```
Rec.Fields("Hoten").Value
```

+ **OpenRecordset**

Tạo một đối tượng Recordset mới và thêm nó vào tập hợp Recordsets

+ **Move n** (n là số nguyên)

Di chuyển con trỏ bản ghi đến bản ghi thứ n trong tập các bản ghi của Recordset

+ **MoveFirst, MoveLast, MovePrevious, MoveNext**

Theo thứ tự là các phương thức để di chuyển con trỏ bản ghi đến bản ghi đầu tiên, cuối cùng, bản ghi liền trước bản ghi hiện tại, bản ghi liền sau bản ghi hiện tại.

+ **AddNew, Update**

Để thêm mới một bản ghi vào Recordset

Quá trình thêm mới một bản ghi gồm các thao tác:

- Ra lệnh AddNew
- Gán giá trị cho các field theo cú pháp: rec.Fields("Tên field") = giá trị
- Ra lệnh Update

Ví dụ: đoạn mã sau sẽ thêm vào trong bảng Table2(ID, Name) của CSDL hiện tại một bản ghi

```
Dim db As DAO.Database
Set db = CurrentDb
Dim rec As Recordset
Set rec = db.OpenRecordset("SELECT * FROM Table2")
rec.addnew
rec.Fields(0) = "001"
rec.Fields(1) = "Nguyen Hong Hanh"
rec.Update
```

+ **Edit, Update**

Để sửa dữ liệu của một bản ghi đã có trong Recordset

Quá trình sửa một bản ghi gồm các thao tác:

- Định vị con trỏ bản ghi tới bản ghi cần sửa bằng các phương thức Move, MoveNext, MoveFirst, MoveLast, MovePrevious
- Ra lệnh Edit
- Gán giá trị cho các field theo cú pháp: rec.Fields("Tên field") = giá trị
- Ra lệnh Update

Ví dụ: đoạn mã sau sẽ sửa bản ghi có ID = "001" trong bảng Table2(ID, Name) của CSDL hiện tại

```
Dim db As DAO.Database
Set db = CurrentDb
Dim rec As Recordset
Set rec = db.OpenRecordset("SELECT * FROM Table2")
rec.MoveLast
rec.edit
rec.Fields("ID") = "100"
rec.Fields("Name") = "Nguyen Mai Trang"
rec.Update
```

+ Delete

Xóa bản ghi hiện tại ra khỏi Recordset. Quá trình xóa một bản ghi trong Recordset gồm các thao tác:

- Định vị con trỏ bản ghi đến bản ghi cần xóa
- Ra lệnh Delete

Ví dụ: Đoạn mã sau sẽ xóa bản ghi cuối cùng trong bảng Table2 của CSDL hiện tại:

```
Dim db As DAO.Database
Set db = CurrentDb
Dim rec As Recordset
Set rec = db.OpenRecordset("SELECT * FROM Table2")
rec.MoveLast
rec.delete
```

+ Close

Đóng đối tượng Recordset

Để tham chiếu đến đối tượng Recordset trong tập Recordsets, ta có thể sử dụng một trong 3 cú pháp sau:

```
Recordsets (0)
Recordsets ("name")
Recordsets! [name]
```

9. QueryDefs

QueryDefs chứa tập các đối tượng QueryDef. Mỗi đối tượng QueryDef dùng để tham chiếu tới các query có sẵn trong CSDL MS Access, hoặc có thể là các query lập trình bằng câu lệnh SQL tạo ra. Sử dụng phương thức CreateQueryDef của đối tượng Database để tạo một query mới.

Quá trình tạo và thi hành một query trong VB sử dụng đối tượng QueryDef gồm các bước:

- Tạo một query mới (nếu không đặt tên thì nó chỉ tồn tại trong bộ nhớ) dùng phương thức CreateQueryDef của đối tượng Database.
- Gán chuỗi lệnh SQL cho query
- Ra lệnh thi hành query
- Đóng query

Ví dụ: Đoạn mã sau sẽ cho hiển thị tất cả các bản ghi của Table1 trong CSDL hiện tại ra cửa sổ Debug


```

Dim db As DAO.Database
Set db = CurrentDb
Dim qr As QueryDef
Set qr = db.CreateQueryDef("showrec")
qr.SQL = "SELECT * FROM Table2"
Dim rec As DAO.Recordset
Set rec = qr.OpenRecordset()
While Not rec.EOF
    Debug.Print rec(0) & Chr(32) & rec(1)
    rec.MoveNext
Wend
rec.Close
qr.Close

```

Ví dụ: đoạn mã sau sẽ thêm một bản ghi mới vào bảng Table2(ID, Name) trong CSDL hiện tại:

```

Dim db As DAO.Database
Set db = CurrentDb
Dim qr As QueryDef
Set qr = db.CreateQueryDef("insertRec")
qr.SQL = "INSERT INTO Table2 Values(123, 'Le Thu Ha')"
qr.Execute
qr.Close

```

Để tham chiếu đến đối tượng QueryDef trong tập QueryDefs, ta có thể sử dụng một trong 3 cú pháp sau:

```

QueryDefs(0)
QueryDefs("name")
QueryDefs![name]

```

10. TableDefs

TableDefs chứa tập các đối tượng TableDef. Mỗi đối tượng TableDef dùng để tham chiếu đến các bảng dữ liệu trong CSDL. Có thể chỉnh sửa, thiết kế cấu trúc của bảng trong chế độ Run time như trong chế độ thiết kế Design View bằng đối tượng này.

Một số thuộc tính, phương thức cơ bản:

- + **Name**: tên bảng được gán vào biến kiểu TableDef
- + **RecordCount**: tổng số bản ghi hiện có trên bảng được gán vào biến TableDef
- + **DateCreated**: thời gian tạo bảng được gán vào biến TableDef
- + **Fields**

Để tham chiếu đến các field của bảng, thuộc tính này là một đối tượng có các thuộc tính và phương thức khác.

+ CreateField

Dùng để tạo field mới cho bảng kiểu TableDef.

Để tạo một đối tượng TableDef mới, ta sử dụng phương thức CreateTableDef của đối tượng Database.

Ví dụ: Đoạn mã sau sẽ tạo ra thêm trong CSDL hiện tại một bảng có tên là table2, gồm các field là id (long integer) và name (text)

```

Dim db As DAO.Database
Dim tbl As DAO.TableDef
Set db = CurrentDb

```

```

Set tbl = db.CreateTableDef("table2")
tbl.Fields.Append tbl.CreateField("ID", dbLong)
tbl.Fields.Append tbl.CreateField("Name", dbText, 200)
db.TableDefs.Append tbl

```

Phương thức CreateTableDef yêu cầu đưa vào tham số là tên của bảng, nếu để là rỗng thì bảng đó chỉ tồn tại trong bộ nhớ, không lưu lại được trong CSDL.

Để tham chiếu đến đối tượng TableDef trong tập TableDefs, ta có thể sử dụng một trong 3 cú pháp:

```

TableDefs(0)
TableDefs("name")
TableDefs![name]

```

11. Relations

Relations chứa một tập các đối tượng Relation. Mỗi đối tượng Relation dùng để tạo kết nối (Relationship) giữa 2 bảng, truy vấn trong CSDL MS Access. Để tạo một đối tượng Relation, ta sử dụng phương thức CreateRelation của đối tượng Database.

Ví dụ: đoạn mã sau sẽ tạo Relationship giữa 2 bảng Tinh (Matinh, Tentinh) và Sinhvien (Masv, Hoten, Nsinh, Matinh) trong CSDL hiện tại.

```

Dim db As DAO.Database
Set db = CurrentDb
Dim rls As DAO.Relation
Set rls = db.CreateRelation("sv_tinh", "tinh", "sinhvien",
dbRelationUpdateCascade)
rls.Fields.Append rls.CreateField("Matinh", dbLong)
rls.Fields("Matinh").ForeignName = "Matinh"
db.Relations.Append rls

```

Phương thức CreateRelation yêu cầu đưa vào các tham số lần lượt là: tên, bảng chính, bảng phụ và kiểu của mối quan hệ. Để thiết lập khóa chính của liên kết ta sử dụng phương thức CreateField của đối tượng Relation và đưa vào tham số yêu cầu là tên của field làm khóa chính của mối quan hệ thuộc bảng chính của liên kết (trong ví dụ trên là field Matinh của bảng Tinh). Sau đó cần chỉ ra khóa ngoại của liên kết (trong ví dụ là field Matinh của bảng Sinhvien).

Để tham chiếu đến đối tượng Relation trong tập Relations ta có thể sử dụng một trong 3 cú pháp:

```

Relations(0)
Relations("name")
Relations![name]

```

12. Containers

Containers chứa tập các đối tượng Container. Mỗi đối tượng Database chứa một tập các đối tượng Container. Đối tượng Container có thể được định nghĩa bởi Microsoft Jet Database Engine hoặc bởi người dùng (khi đó DAO không hỗ trợ bất kỳ phương thức hay thuộc tính nào). Sau đây liệt kê một số đối tượng Container và kiểu thông tin mà nó chứa được Microsoft Jet Database Engine hỗ trợ:

Tên Container	Chứa thông tin về
Databases	Tất cả các đối tượng database được lưu trong CSDL

Tables	Tất cả các đối tượng table và query được lưu trong CSDL
Relations	Tất cả các đối tượng relationship được lưu trong CSDL

Chú ý:

+ Đối tượng Databases Container tham chiếu đến tất cả các đối tượng Database được lưu trong CSDL. Còn tập Databases chỉ tham chiếu đến các đối tượng Database đang được mở trong đối tượng Workspace cụ thể. Tương tự với các loại đối tượng Container khác.

+ Container là các đối tượng có sẵn (built - in), không thể tạo ra hay hủy bỏ chúng. Để tham chiếu đến đối tượng Container trong tập các đối tượng Containers, ta có thể sử dụng 1 trong 3 cú pháp sau:

```
Containers (0)
Containers ("name")
Containers! [name]
```

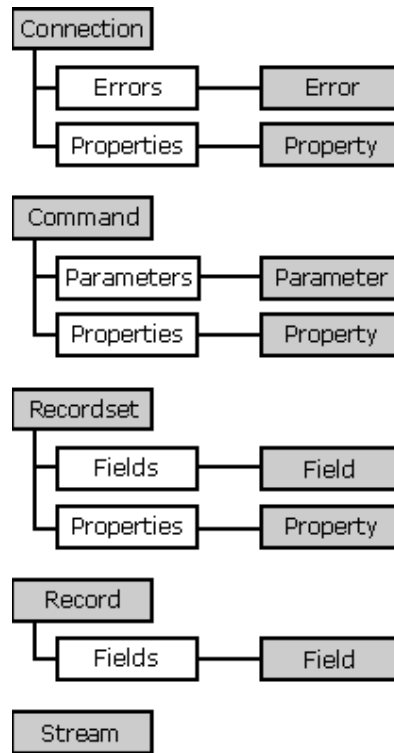
6.3 Kiến trúc ADO (ActiveX Data Objects)

Microsoft giới thiệu một tập hợp mô hình đối tượng dữ liệu mang tính khái quát hơn nhằm cung cấp phép tham chiếu không chỉ cho đối tượng lưu trong JET DBEngine mà còn cho dữ liệu được lưu trữ trong hệ quản trị CSDL khác như Microsoft SQL Server. Những mô hình này gọi là kiến trúc ADO (ActiveX Data Object). MS Access 2000 có khả năng hỗ trợ trực tiếp cho ADO thông qua các thư viện cài đặt sẵn và phép tham chiếu trực tiếp đến đối tượng chính trong mô hình từ đối tượng Application của MS Access.

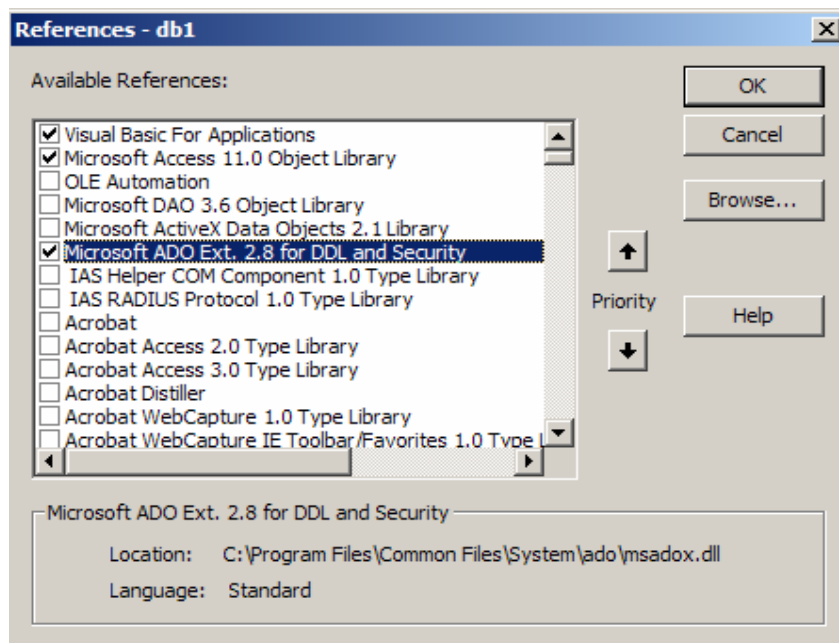
Vì kiến trúc ADO được thiết kế là để cung cấp tập hợp các đối tượng chung cho bất kỳ hệ thống dữ liệu nào hỗ trợ ActiveX Data Options nên chúng không nhất thiết phải hỗ trợ tất cả các đặc tính có trong DAO vốn được thiết kế chuyên dụng cho Microsoft Jet DBEngine.

Kiến trúc ADO gồm 2 thành phần là ADOX (Microsoft ActiveX Data Objects Extensions for Data Definition Language and Security) và ADODB.

Mô hình ADO cơ bản cho phép bạn mở và thao tác với tập các bản ghi thông qua đối tượng Recordset, và thi hành các truy vấn thông qua đối tượng Command. Hình sau minh họa các đối tượng, tập hợp và mối quan hệ giữa chúng trong kiến trúc ADO.

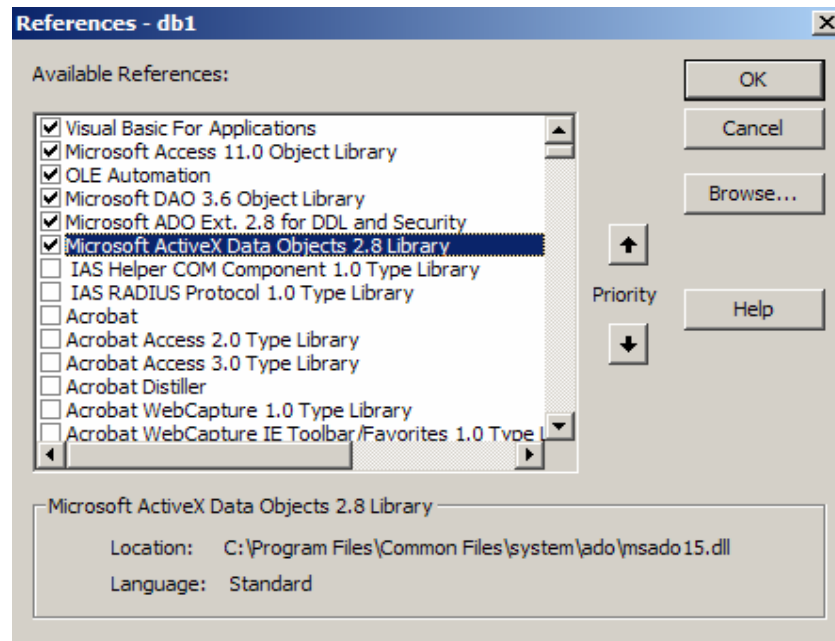


Hình 7.7: Các đối tượng và mối quan hệ giữa chúng trong kiến trúc ADO
 Trong kiến trúc này, các đối tượng Connection, Command, Recordset là thuộc thành phần ADODB. Các đối tượng còn lại thuộc thành phần ADOX.
 Để tham chiếu đến thành phần ADOX, ta phải yêu cầu VB nạp một tham chiếu đến Microsoft ADO Ext. 2.8 for DDL and Security bằng cách mở module bất kỳ, trên thanh công cụ chọn lệnh: *Tool/Reference*, kết quả nhận được là hộp thoại có dạng như hình dưới, sau đó chọn vào mục *Microsoft ADO Ext. 2.8 for DDL and Security*



Hình 7.8: Tham chiếu đến thành phần ADOX trong kiến trúc ADO
 Để tham chiếu đến thành phần ADODB, ta phải yêu cầu VB nạp một tham chiếu đến Microsoft ActiveX Data Objects Library bằng cách mở module bất kỳ, trên thanh công

cụ chọn lệnh: *Tool/Reference*, kết quả nhận được là hộp thoại có dạng như hình dưới, sau đó chọn vào mục *Microsoft ActiveX Data Objects 2.8 Library*



Hình 7.9: Tham chiếu đến thành phần ADODB trong kiến trúc ADO
 Kiến trúc ADO cung cấp 9 đối tượng (object) và 4 tập hợp (collection) bao gồm:

1. Đối tượng Connection

Dùng để thiết lập một phiên làm việc với nguồn dữ liệu. Trong các hệ quản trị CSDL theo mô hình Client/Server, Connection tương đương với một kết nối thực sự qua mạng đến server CSDL. Tùy thuộc vào Provider mà một số các tập hợp, phương thức, thuộc tính của đối tượng Connection có thể không được hỗ trợ.

Sau đây là một số thuộc tính và phương thức của đối tượng Connection

- Phương thức **Open**

Phương thức này dùng để mở một kết nối của đối tượng Connection đến nguồn dữ liệu.

Cú pháp:

```
Connection.Open ConnectionString, UserID, Password, Options
```

Trong đó:

ConnectionString: là một chuỗi ký tự để thiết lập kết nối, gồm các thuộc tính sau (mỗi thuộc tính được phân cách nhau bởi dấu chấm phẩy ;)

+ **Provider** = : tên của provider được sử dụng cho Connection.

Ví dụ:

Provider = Microsoft.Jet.OLEDB.4.0 là provider cho nguồn dữ liệu của hệ quản trị CSDL MS Access.

Provider = SQLOLEDB là provider cho nguồn dữ liệu của hệ quản trị CSDL SQL Server.

+ **File Name** =: Tên của file CSDL

+ **Remote Provider**: Tên của provider khi mở một connection ở phía client

+ **Remote Server**: Tên của server khi mở connection ở phía client.

UserID: Chuỗi ký tự chỉ định tên của người dùng để thiết lập kết nối.

Password: Chuỗi ký tự là password của người dùng trong UserID.

Options: là một giá trị thuộc `ConnectOptionEnum` xác định phương thức được trả về trước hoặc sau khi kết nối được thiết lập.

Ví dụ: Đoạn mã sau sẽ mở một kết nối của đối tượng `cn` đến file CSDL `e:\db1.MDB`.

```
Dim cn As New ADODB.Connection
Dim sql
sql = "Provider = Microsoft.Jet.OLEDB.4.0; Data Source =
E:\db1.mdb"
cn.Open sql
```

- Phương thức **Excute**

Dùng để thi hành một câu lệnh SQL hoặc một thủ tục nội (stored procedure)

Cú pháp:

```
Connection.Excute CommandText, RecordsAffected, Options
```

Trong đó:

CommandText: Câu lệnh SQL, tên thủ tục nội, tên bảng

RecordsAffected: Là một biến kiểu long trả về số các bản ghi nhận được tác động của việc thi hành lệnh.

Options: Là một giá trị kiểu long chỉ định cách thức mà provider định giá tham số `CommandText` thế nào cho tối ưu.

Ví dụ: Đoạn mã sau sẽ chèn thêm một bản ghi vào bảng `Table1(ID, Name)` trong CSDL `E:\DB1.MDB`

```
Dim cn As New ADODB.Connection
Dim cnstr, sql
cnstr = "Provider = Microsoft.Jet.OLEDB.4.0; Data Source =
E:\db1.mdb"
cn.Open cnstr
sql = "INSERT INTO Table1(ID, Name) VALUES (1001, 'Nguyen
Minh Hang') "
cn.Execute sql
```

- Phương thức **Close**

Dùng để đóng một connection đang mở

Cú pháp: `Connection.Close`

Ví dụ: `cn.Close`

- Phương thức **BeginTrans**

Bắt đầu một giao dịch mới giữa đối tượng `Connection` và nguồn dữ liệu.

Cú pháp:

```
Tên biến = Connection.BeginTrans
```

Giá trị trả về khi gọi phương thức này là một số kiểu long chỉ ra mức độ của giao dịch.

- Phương thức **CommitTrans**

Lưu lại các thay đổi do giao dịch hiện tại sinh ra và kết thúc nó.

Cú pháp:

```
Connection.CommitTrans
```

- Phương thức **RollbackTrans**

Hủy bỏ các thay đổi do giao dịch hiện tại sinh ra và kết thúc nó.

Cú pháp:

Connection.RollbackTrans

- Thuộc tính **Provider**

Là một chuỗi ký tự chỉ ra tên của provider như trong phương thức Open

- Thuộc tính **ConnectionString**

Là một chuỗi ký tự chỉ ra chuỗi kết nối như trong phương thức Open

2. Đối tượng Command

Được sử dụng để định nghĩa một câu lệnh SQL.

Một số thuộc tính và phương thức của đối tượng Command

- Thuộc tính **ActiveConnection**

Để thiết lập đối tượng Connection cho Command.

- Thuộc tính **CommandText**

Là một câu lệnh SQL

- Phương thức **Execute**

Để thi hành câu lệnh trong CommandText

Ví dụ: Đoạn mã sau sẽ thêm một bản ghi mới vào bảng Table1(ID, Name) trong CSDL E:\db1.MDB

```
Dim cn As adodb.Connection
Dim cnstr, sql
Dim cmd As adodb.Command
cnstr = "Provider = Microsoft.Jet.OLEDB.4.0; Data Source = e:\db1.mdb"
Set cn = New adodb.Connection
cn.Open cnstr
Set cmd = New adodb.Command
cmd.ActiveConnection = cn
cmd.CommandText = sql
cmd.Execute
cn.Close
```

3. Đối tượng Recordset

Chứa một tập các bản ghi của một bảng hoặc là kết quả của câu lệnh SQL được thi hành. Tại bất kỳ thời điểm nào, đối tượng Recordset cũng chỉ tham chiếu đến 1 bản ghi trong tập các bản ghi đó và được gọi là bản ghi hiện thời.

Đối tượng này có các thuộc tính và phương thức tương tự các thuộc tính và phương thức của đối tượng Recordset trong mô hình DAO. Sau đây sẽ minh họa một số ví dụ:

Ví dụ 1: Đoạn mã sau sẽ cho hiển thị tất cả các bản ghi có trong Table1(ID, Name) của CSDL e:\DB1.MDB lên cửa sổ Debug.

```
Dim cn As adodb.Connection
Dim cnstr, sql
Dim rec As adodb.Recordset
cnstr = "Provider = Microsoft.Jet.OLEDB.4.0; Data Source = e:\db1.mdb"
Set cn = New adodb.Connection
Set rec = New adodb.Recordset
cn.Open cnstr
sql = "SELECT * FROM Table1"
rec.Open sql, cn
While Not rec.EOF
    Debug.Print rec(0) & Chr(32) & rec(1)
    rec.MoveNext
Wend
```

```
rec.Close
cn.close
```

Hoặc sử dụng đối tượng Command:

```
Dim cn As adodb.Connection
Dim cnstr, sql
Dim rec As adodb.Recordset
Dim cmd As adodb.Command
cnstr = "Provider = Microsoft.Jet.OLEDB.4.0; Data Source =
e:\db1.mdb"
Set cn = New adodb.Connection
Set cmd = New adodb.Command
Set rec = New adodb.Recordset
cn.Open cnstr
sql = "SELECT * FROM Table1"
cmd.ActiveConnection = cn
cmd.CommandText = sql
Set rec = cmd.Execute
While Not rec.EOF
    Debug.Print rec(0) & Chr(32) & rec(1)
    rec.MoveNext
Wend
rec.Close
cn.Close
```

Ví dụ 2: Thêm mới một bản ghi vào bảng Table1(ID, Name) của CSDL e:\DB1.MDB

```
Dim cn As adodb.Connection
Dim cnstr, sql
Dim rec As adodb.Recordset
cnstr = "Provider = Microsoft.Jet.OLEDB.4.0; Data Source =
e:\db1.mdb"
Set cn = New adodb.Connection
Set rec = New adodb.Recordset
cn.Open cnstr
rec.Open "Table1", cn, adOpenDynamic, adLockOptimistic
rec.addnew
rec![ID] = 1002
rec![Name] = "Doan Ngoc Diep"
rec.Update
rec.Close
cn.Close
```

Ví dụ 2: Sửa bản ghi đầu tiên trong bảng Table1(ID, Name) của CSDL e:\DB1.MDB

```
Dim cn As adodb.Connection
Dim cnstr, sql
Dim rec As adodb.Recordset
cnstr = "Provider = Microsoft.Jet.OLEDB.4.0; Data Source =
e:\db1.mdb"
Set cn = New adodb.Connection
Set rec = New adodb.Recordset
cn.Open cnstr
rec.Open "Table1", cn, adOpenDynamic, adLockOptimistic
rec.MoveFirst
rec.Fields![id] = 1005
rec.Fields![Name] = "Nguyen Mai Trang"
```



```
rec.Update  
rec.Close  
cn.Close
```

Ví dụ 3: Đoạn mã sau sẽ xóa bản ghi đầu tiên trong Table1(ID, Name) của CSDL e:\DB1.MDB

```
Dim cn As adodb.Connection  
Dim cnstr, sql  
Dim rec As adodb.Recordset  
cnstr = "Provider = Microsoft.Jet.OLEDB.4.0; Data Source =  
e:\db1.mdb"  
Set cn = New adodb.Connection  
Set rec = New adodb.Recordset  
cn.Open cnstr  
rec.Open "Table1", cn, adOpenDynamic, adLockOptimistic  
rec.MoveFirst  
rec.delete  
rec.Close  
cn.Close
```

4. Đối tượng Record

Là một bản ghi (row) của đối tượng Recordset.

5. Field

Thể hiện một cột (column) dữ liệu của đối tượng Recordset.

6. Property

Chứa các thuộc tính động của đối tượng ADO được định nghĩa bởi provider

7. Parameter

Chứa các tham số của đối tượng Command. Đó là các tham số cho truy vấn tham số và thủ tục nội

8. Đối tượng Error

Chứa chi tiết về các lỗi truy cập dữ liệu.

9. Đối tượng Stream

Là một dòng (stream) dữ liệu dạng nhị phân.

10. Tập hợp Errors

Chứa tất cả các đối tượng Error được sinh ra khi truy cập với dữ liệu.

11. Tập hợp Fields

Chứa tất cả các đối tượng Field của đối tượng Recordset.

12. Tập hợp Parameters

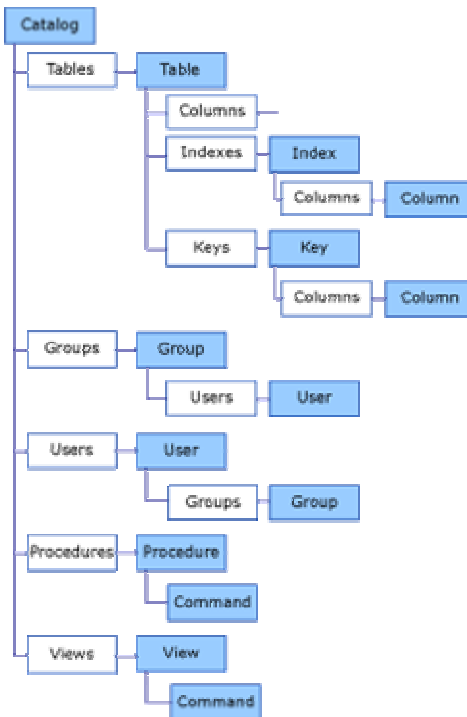
Chứa tất cả các đối tượng Parameter của đối tượng Command

13. Tập hợp Properties

Chứa tất cả các đối tượng Property của một đối tượng cụ thể.

ADOX là một mở rộng của ADO cho phép bạn tạo, mở, thao tác với các Tables, View, Procedure (Trigger và Procedure). Để sử dụng ADOX, bạn phải thiết lập một tham chiếu tới thư viện của ADOX, đó chính là "Microsoft ADO Ext. for DDL and Security". Tập tin thư viện này có tên là Msadox.dll và mã chương trình (ProgID) là "ADOX".

Hình sau minh họa chi tiết các đối tượng và mối quan hệ giữa chúng trong thành phần ADOX



Hình 7.10: Các đối tượng và mối quan hệ giữa chúng trong ADOX
 Mô hình ADOX cung cấp 9 đối tượng và 8 tập hợp bao gồm:

1. Đối tượng Catalog

Chứa một tập các mô tả về schema của nguồn dữ liệu

2. Đối tượng Column

Thể hiện một cột của một bảng (table), index hoặc key

3. Đối tượng Group

Thể hiện một tài khoản nhóm được phép làm việc với CSDL

4. Đối tượng Index

Thể hiện một index của một bảng trong CSDL

5. Đối tượng Key

Thể hiện một field là khóa chính, khóa ngoại của một bảng trong CSDL

6. Đối tượng Procedure

Thể hiện một thủ tục nội trong CSDL

7. Đối tượng Table

Thể hiện một bảng trong CSDL, bao gồm các field, index và key

8. Đối tượng User

Thể hiện một tài khoản người dùng được phép làm việc với CSDL

9. Đối tượng View

Thể hiện một bảng ảo (view) của CSDL.

10. Tập hợp Columns

Chứa tất cả các đối tượng Column của table, index, key

11. Tập hợp Groups

Chứa tất cả các đối tượng Group của một catalog hoặc của người dùng

12. Tập hợp Indexes

Chứa tất cả các đối tượng Index của một table

13. Tập hợp Keys

Chứa tất cả các đối tượng Key của một table

14. Tập hợp Procedures

Chứa tất cả các đối tượng Procedure của một catalog

15. Tập hợp [Tables](#)

Chứa tất cả các đối tượng Table của một catalog

16. Tập hợp [Users](#)

Chứa tất cả các đối tượng User của một catalog hoặc một nhóm người dùng

17. Tập hợp [Views](#)

Chứa tất cả các đối tượng View của một catalog

TÀI LIỆU THAM KHẢO

- [1] Phạm Văn Át, Hướng dẫn sử dụng Microsoft Access97.
- [2] Nguyễn Tiến, Đặng Xuân Hùng, Nguyễn Văn Hoài, Hướng dẫn lập trình Microsoft Access97, NXB Giáo dục, 1998.
- [3] MSDN Library – April 2001
- [4] Bùi Thế Tâm, Giáo trình Microsoft Access, NXB GTVT, 2005.
- [5] VN-Guide, Microsoft Access 2000 cho mọi người, NXB Thống kê, 1999.

MỤC LỤC

LỜI NÓI ĐẦU.....	1
CHƯƠNG 1.....	2
TỔNG QUAN VỀ MICROSOFT ACCESS.....	2
1. GIỚI THIỆU VỀ MICROSOFT ACCESS.....	2
2. LÀM VIỆC VỚI MICROSOFT ACCESS.....	2
2.1 Khởi động Access.....	2
2.2 Tạo một cơ sở dữ liệu mới.....	3
2.3 Làm việc với cơ sở dữ liệu đã tồn tại.....	4
CHƯƠNG 2.....	6
BẢNG VÀ MỐI QUAN HỆ.....	6
1. CẤU TRÚC CỦA MỘT BẢNG.....	6
2. THIẾT KẾ BẢNG.....	6
2.1 Tạo bảng.....	6
2.2 Đặt khoá chính.....	9
2.3 Thuộc tính của trường.....	9
2.4 Thuộc tính của bảng.....	10
2.5 Cập nhật dữ liệu vào bảng.....	10
2.6 Xoá và đổi tên bảng.....	11
3. THIẾT LẬP QUAN HỆ GIỮA CÁC BẢNG.....	11
3.1 Mối quan hệ giữa các bảng.....	11
3.2 Cách tạo quan hệ.....	11
3.3 Chỉnh sửa quan hệ.....	13
3.4 Tạo các toàn vẹn tham chiếu.....	13
BÀI TẬP CHƯƠNG 2.....	15
CHƯƠNG 3.....	17
TRUY VẤN.....	17
1. TỔNG QUAN VỀ TRUY VẤN.....	17
2. THAO TÁC TRÊN TRUY VẤN.....	17
2.1 Các bước tạo một truy vấn mới.....	17
2.2 Đặt điều kiện trong truy vấn.....	20
2.3 Tạo trường tính toán trong truy vấn.....	21
2.4 Thuộc tính của trường và truy vấn.....	21
2.5 Các phép toán và một số hàm trong Access.....	22
3. CÁC LOẠI TRUY VẤN KHÁC.....	26
3.1 Truy vấn tính tổng (Totals).....	26
3.2 Truy vấn thông số (Parameter Query).....	27
3.3 Truy vấn Crosstab.....	29
3.4 Truy vấn tạo bảng (Make Table Query).....	33
3.5 Truy vấn bổ sung (Append Query).....	34
3.6 Truy vấn xoá (Delete Query).....	36
3.7 Truy vấn chỉnh sửa (Update Query).....	37
CHƯƠNG 4.....	43
MẪU BIỂU.....	43
1. TẠO MẪU BIỂU BẰNG AUTOFORM VÀ WIZARD.....	43
1.1 Tạo mẫu biểu bằng AutoForm.....	43

1.2 Tạo mẫu biểu bằng Form Wizard.....	44
2. THIẾT KẾ FORM Ở CHẾ ĐỘ DESIGN	46
2.1 Các thành phần trong một Form.....	46
2.2 Các bước thiết kế Form	46
3. HỘP CÔNG CỤ TOOLBOX.....	49
3.1 Điều khiển Text box	50
3.2 Điều khiển Combo Box	51
3.3 Điều khiển Command Button.....	56
3.4 Thiết kế dạng Main – Sub Form.....	58
BÀI TẬP CHƯƠNG 4	62
CHƯƠNG V	64
BÁO BIỂU	64
1. TẠO BÁO BIỂU BẰNG AUTOREPORT	64
1.1 Tạo báo biểu tự động kiểu Tabular.....	64
1.2 Tạo báo biểu tự động kiểu Columnar.....	65
2. TẠO BÁO BIỂU DÙNG REPORT WIZARD	65
3. TẠO BÁO BIỂU DÙNG DESIGN VIEW	69
4. TẠO CÁC NHÃN DÙNG LABEL WIZARD.....	72
BÀI TẬP CHƯƠNG 5	74
CHƯƠNG 6.....	75
MACRO VÀ MENU	75
1. MACRO.....	75
1.1 Tạo và sử dụng Macro	75
1.2 Các hành động Macro thường dùng	76
1.3 Gắn Macro với các sự kiện trên Form/Report.....	76
1.4 Macro AutoExec.....	77
2. MENU BAR VÀ TOOLBAR	78
BÀI TẬP CHƯƠNG 6	81
CHƯƠNG 7.....	82
LẬP TRÌNH CƠ SỞ DỮ LIỆU TRONG MS ACCESS.....	82
1. GIỚI THIỆU LẬP TRÌNH VISUAL BASIC APPLICATION	82
2. MODULE	82
2.1 Module chuẩn (Standard Module).....	82
2.2 Module lớp (Class Module).....	84
2.3 Module của form/report (Form/report Module)	85
3. KIỂU DỮ LIỆU, HẰNG VÀ BIẾN.....	87
3.1 Kiểu dữ liệu	87
3.2 Hằng.....	88
3.3 Biến.....	89
4. CÁC CẤU TRÚC ĐIỀU KHIỂN.....	91
4.1 Cấu trúc tuần tự	91
4.2 Cấu trúc rẽ nhánh.....	92
4.3 Cấu trúc lặp.....	94
5. HÀM VÀ THỦ TỤC CON	96
5.1 Hàm	96
5.2 Thủ tục con	97
6. CÁC MÔ HÌNH TRUY CẬP CƠ SỞ DỮ LIỆU	98
6.1 Kiến trúc chương trình ứng dụng MS Access	98

6.2 Kiến trúc DAO (Data Access Objects).....	98
6.3 Kiến trúc ADO (ActiveX Data Objects)	106
TÀI LIỆU THAM KHẢO	115
MỤC LỤC	116