



# GIÁO TRÌNH MICROSOFT ACCESS 2000



# GIỚI THIỆU

Giáo trình này là sự đúc kết kinh nghiệm của tác giả trong giảng dạy các khoá đào tạo về Access cũng như kinh nghiệm sử dụng Access làm hệ quản trị cơ sở dữ liệu và công cụ phát triển phần mềm. Hầu hết nội dung trình bày trong giáo trình đều bám theo những bài tập thực hành cụ thể. Do vậy, sau khi đọc xong và thực hành theo trình tự cuốn sách này, bạn đọc gần như có thể thiết kế và xây dựng được những ứng dụng về quản lý thông tin. Cụ thể ở đây là: Quản lý lương cán bộ một cơ quan và Quản lý bán hàng tại một cửa hàng.

Đối tượng chính của giáo trình là sinh viên, học sinh các trường THCN hệ chuyên CNTT cũng như không chuyên CNTT có học môn Access. Tất nhiên cũng hoàn toàn phù hợp cho những ai quan tâm học hỏi môn học này bởi lẽ các nội dung đều được trình bày rất trực quan, có thứ tự và kết quả rõ ràng.

Thời lượng thiết kế 75 tiết. Trong đó 24 tiết lý thuyết, 46 tiết thực hành và 5 tiết dành cho kiểm tra. Tùy theo mức độ ứng dụng của từng trường, ngành vào môn học này mà mỗi đơn vị sẽ chọn cho mình một quỹ thời gian cũng như các nội dung giảng dạy phù hợp từ giáo trình này.

Nội dung giáo trình gồm một bài mở đầu và 7 chương:

## **Bài mở đầu**

Bài này sẽ trình bày tổng quát về môn học; giới thiệu về Access làm sao hiệu được Access dùng để làm gì? Phân biệt được với một số phần mềm đã được học như Pascal, Foxpro, Word, Excel,... Cuối cùng bạn đọc sẽ hiểu được môi trường làm việc của Access và thực hiện vào ra và quản lý các tệp dữ liệu.

## **Chương 1: Xây dựng cơ sở dữ liệu (CSDL)**

Chương 1 tập trung vào những kiến thức cơ bản giúp có thể xây dựng CSDL Access một cách chắc chắn; đặc biệt thể hiện qua qui trình xây dựng một CSDL tổng kết được nội dung toàn bộ chương học.

## **Chương 2: Truy vấn dữ liệu**

Tập trung vào các kiến thức và kỹ năng giúp xử lý dữ liệu một cách đa dạng thông qua các loại Query.

## **Chương 3: Thiết kế giao diện**

Chương này trình bày các khái niệm cũng như những kỹ thuật, kỹ năng có thể thiết kế được những giao diện phần mềm thông qua việc sử dụng Form và các công cụ đi kèm.

## **Chương 4: In ấn**

Bao gồm các kiến thức cũng như kỹ năng cần thiết để thiết kế các mẫu biểu in ấn đa dạng sử dụng công cụ Report của Access. Hơn nữa kết hợp với Form, sẽ giải quyết được các bài tập mang tính tổng hợp, hoàn thiện.

Nội dung từ chương 1 đến chương 4 mang tính căn bản. Những chương tiếp theo mang tính chuyên ngành và nâng cao hơn.

## **Chương 5: Lập trình VBA căn bản**

Chương này cung cấp những khái niệm căn bản về môi trường, cách làm việc và làm việc trên những bài toán đơn giản sử dụng ngôn ngữ lập trình VBA. Đây là cơ sở để tiếp cận tốt hơn trong chương tiếp theo- lập trình CSDL.

## **Chương 6: Lập trình CSDL**

Trình bày những khái niệm, nguyên tắc, công cụ và những kỹ năng rất căn bản về lập trình CSDL nói chung cũng như lập trình CSDL Access bằng VBA nói riêng. Kết thúc chương này có thể xử lý được CSDL rất đa dạng, thực tế và hoàn toàn có thể dùng Access để phát triển các CSDL thành những sản phẩm phần mềm hoàn chỉnh.

## **Chương 7: Menu & Toolbar**

Cung cấp những kiến thức, những kỹ năng cuối cùng để có thể hoàn thiện một CSDL Access thành phần mềm đóng gói, hoàn chỉnh.

Cuối cùng, chúng tôi xin chân thành cảm ơn sự quan tâm và tạo điều kiện của Sở GD&ĐT Hà Nội; các đồng nghiệp đã tận tình giúp đỡ, đóng góp ý kiến xây dựng để giáo trình được hoàn thiện như bây giờ. Tuy nhiên rất khó tránh khỏi những thiếu sót từ khách quan đến chủ quan. Chúng tôi rất mong nhận được sự góp ý, phê bình để giáo trình dần được hoàn thiện hơn, phục vụ tốt cho việc dạy-học trong nhà trường và các bạn đọc.

*Hà nội, ngày 3 tháng 2 năm 2005*

**Tác giả**

Nguyễn Sơn Hải

Trung tâm Tin học - Bộ Giáo dục và Đào tạo

ĐT: 0913.382.686.

Email: [nshai@moet.edu.vn](mailto:nshai@moet.edu.vn)

*\* Đề nghị ghi rõ xuất xứ khi sử dụng bất kỳ nội dung nào của giáo trình này!*

# BÀI MỞ ĐẦU

Bài mở đầu sẽ trình bày về môi trường làm việc, cũng như cách vào ra phần mềm Access, đó là:

- Giới thiệu Access2000;
- Cách khởi động;
- Tạo tệp Access mới;
- Môi trường làm việc;
- Mở tệp Access đã tồn tại;
- Thoát khỏi Access.

## 1. Giới thiệu Access 2000

Từ cuối những năm 80, hãng Microsoft đã cho ra đời hệ điều hành Windows, đánh dấu một bước ngoặt trong phát triển các ứng dụng phần mềm trên nền Windows (giao diện GUI- Graphical User Interface). Một trong những ứng dụng nổi bật nhất đi kèm lúc đó là bộ phần mềm tin học văn phòng Microsoft Office. Từ đó đến nay, bộ phần mềm này vẫn chiếm thị phần số 1 trên thế giới trong lĩnh vực tin học văn phòng.

Ngoài những ứng dụng về văn phòng quen thuộc phải kể đến như: MS Word - để soạn thảo tài liệu; MS Excel - bảng tính điện tử; MS Powerpoint - để trình chiếu báo cáo; .. còn phải kể đến phần mềm quản trị cơ sở dữ liệu rất nổi tiếng đi kèm: MS Access. Đến nay phiên bản mới nhất là AccessXP. Toàn bộ nội dung giáo trình này, chúng tôi giới thiệu trên Access2000. Về cơ bản, các phiên bản từ Access97 trở lại đây cách sử dụng gần giống nhau. Mỗi phiên bản chỉ khác một số tính năng đặc biệt và một chút về giao diện. Do đó, khi học Access2000, bạn đọc luôn có được những kiến thức cần thiết nhất để tiếp thu những phiên bản Access mới sau này cũng như để nhìn nhận và sử dụng tốt các phiên bản cũ hơn. Chúng tôi khuyên bạn đọc nên sử dụng Access97 trở lên.

Nếu là người chưa từng biết gì về Access, bạn sẽ có một câu hỏi:

### **Access làm được gì, và những ứng dụng của nó trong thực tế?**

- Access là một Hệ quản trị cơ sở dữ liệu quan hệ (RDMS- Relational Database Management System), rất phù hợp cho các bài toán quản lý vừa và nhỏ. Hiệu năng cao và đặc biệt dễ sử dụng- bởi lẽ giao diện sử dụng phần mềm này gần giống hết một số phần mềm khác trong bộ MS Office quen thuộc như : MS Word, MS Excel;
- Hơn nữa, Access còn cung cấp hệ thống công cụ phát triển khá mạnh đi kèm (Development Tools). Công cụ này sẽ giúp các nhà phát triển phần mềm đơn giản trong việc xây dựng trọn gói các dự án phần mềm quản lý qui mô vừa

và nhỏ. Đặc biệt những ai muốn học phát triển phần mềm thì đây là cách dễ học nhất, nhanh nhất giải quyết bài toán này.

- Đến đây có thể khẳng định được 2 ứng dụng chính của Access là :
  1. Dùng để xây dựng hệ cơ sở dữ liệu (chỉ là phần cơ sở dữ liệu, còn phần phát triển thành phần mềm có thể dùng các công cụ khác để làm như : Visual Basic, Visual C, Delphi, .NET,..)
  2. Có thể dùng để xây dựng trọn gói những phần mềm quản lý qui mô vừa và nhỏ.

### **Access2000 có gì mới so với các phiên bản cũ (Access97)?**


- Phải thừa nhận giao diện sử dụng đã có những tiến bộ vượt bậc. Các thao tác sử dụng ít đi, đơn giản hơn và giao diện rất thân thiện;
- Công nghệ truy cập dữ liệu ADO – ActiveX Data Objects hoàn toàn có thể thay thế công nghệ trước đây sử dụng trong Access là DAO – Data Access Objects bởi nhiều lý do cả về cách thức sử dụng lẫn những tối ưu về kỹ thuật. Chúng tôi sẽ giới thiệu kỹ công nghệ này trong Phần 2 cuốn giáo trình;
- Ngôn ngữ lập trình VBA được cải tiến, đặc biệt bản MS Access 2000 Developer còn cung cấp công cụ để đóng gói dự án Access (tức là tạo bộ cài đặt mà khi sử dụng không cần phải cài đặt Access lên máy tính)
- Có khả năng tạo các ứng dụng truy cập cơ sở dữ liệu thông qua giao diện web (web-base). Điều này chưa hề có trong các phiên bản trước đây. Tuy nhiên, khả năng này vẫn còn một số giới hạn, chưa thể thực sự mạnh như các công cụ chuyên nghiệp khác như : ASP, PHP, .NET, ...

Để có thể sử dụng Access, máy tính phải được cài đặt phần mềm này thông qua bộ Microsoft Office 2000. Các bước cài đặt xin tham khảo tài liệu hướng dẫn sử dụng đi kèm đĩa CD phần mềm.

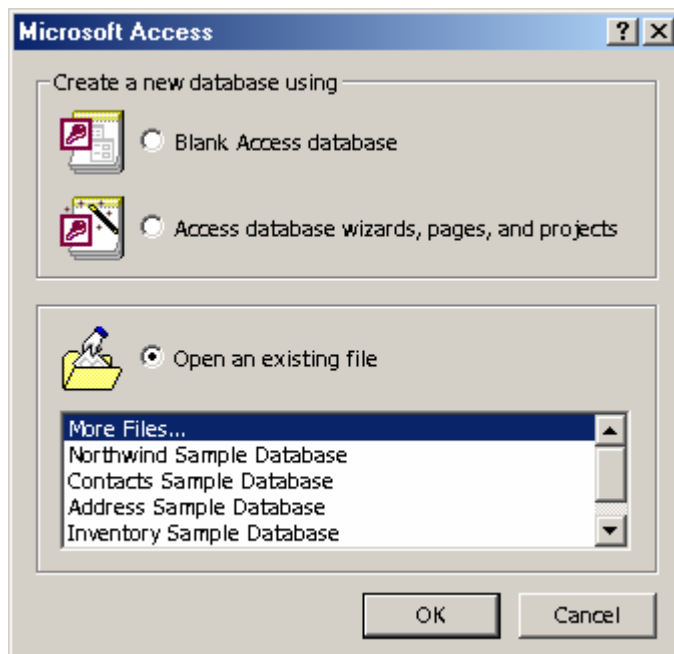
## **2. Khởi động**

Có khởi động Access theo nhiều cách:

- Mở lệnh **Start | Programs | Microsoft Access** của Window;

- Hoặc có thể nhấn kép chuột lên tên tệp có phần mở rộng .MDB với biểu tượng của Microsoft Access 

Sau khi ra lệnh chạy chương trình Access, thông thường màn hình tiếp theo xuất hiện cho phép chọn cách làm việc:



- Chọn **Blank Access database** để bắt đầu tạo một tệp Access mới (tạo mới tệp);
- Hoặc **Access database wizard, page, and project** để tạo một CSDL theo mẫu có sẵn (không trình bày trong giáo trình này);
- Hoặc **Open an existing file** để mở tệp Access đã tồn tại để làm việc tiếp.

### 3. Tạo mới tệp Access

Một dự án Access (Access project) là một hoặc nhiều tệp Access nhằm giải quyết một công việc lớn nào đó. Khuôn khổ giáo trình này chỉ trình bày những dự án có 1 tệp.

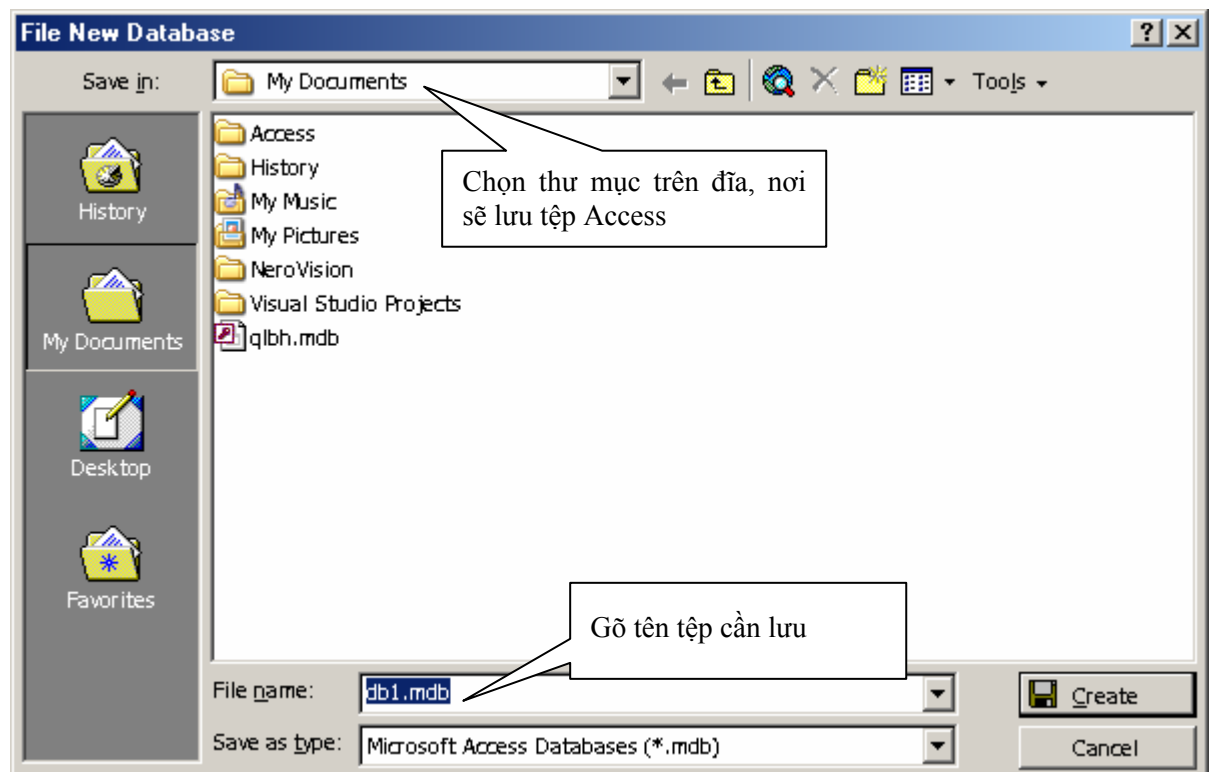
Tệp Access có phần mở rộng \*.MDB (ngoài ra những tệp có phần mở rộng \*.DBE cũng mở được nhưng chỉ có thể thực thi chứ không chỉnh sửa được cấu trúc). Có 5 thành phần chính trên một tệp:


- **Tables** – nơi chứa toàn bộ các bảng dữ liệu;

- **Queries** – nơi chứa toàn bộ các truy vấn dữ liệu đã được thiết kế;
- **Forms** – nơi chứa các mẫu giao diện phần mềm;
- **Reports** – nơi chứa các mẫu báo cáo đã được thiết kế;
- **Macro** – nơi chứa các Macro lệnh phục vụ dự án;
- **Modules** – nơi chứa các khai báo, các thư viện chương trình con phục vụ dự án.

Thông thường mỗi tệp Access cần phải làm việc trên tất cả các thành phần trên.

Để bắt đầu tạo một tệp mới, chọn mục **Blank Access database** ở cửa sổ mục 2, tiếp theo nhấn **OK**, hộp thoại sau xuất hiện yêu cầu chọn nơi (thư mục) lưu trữ và đặt tên tệp Access:

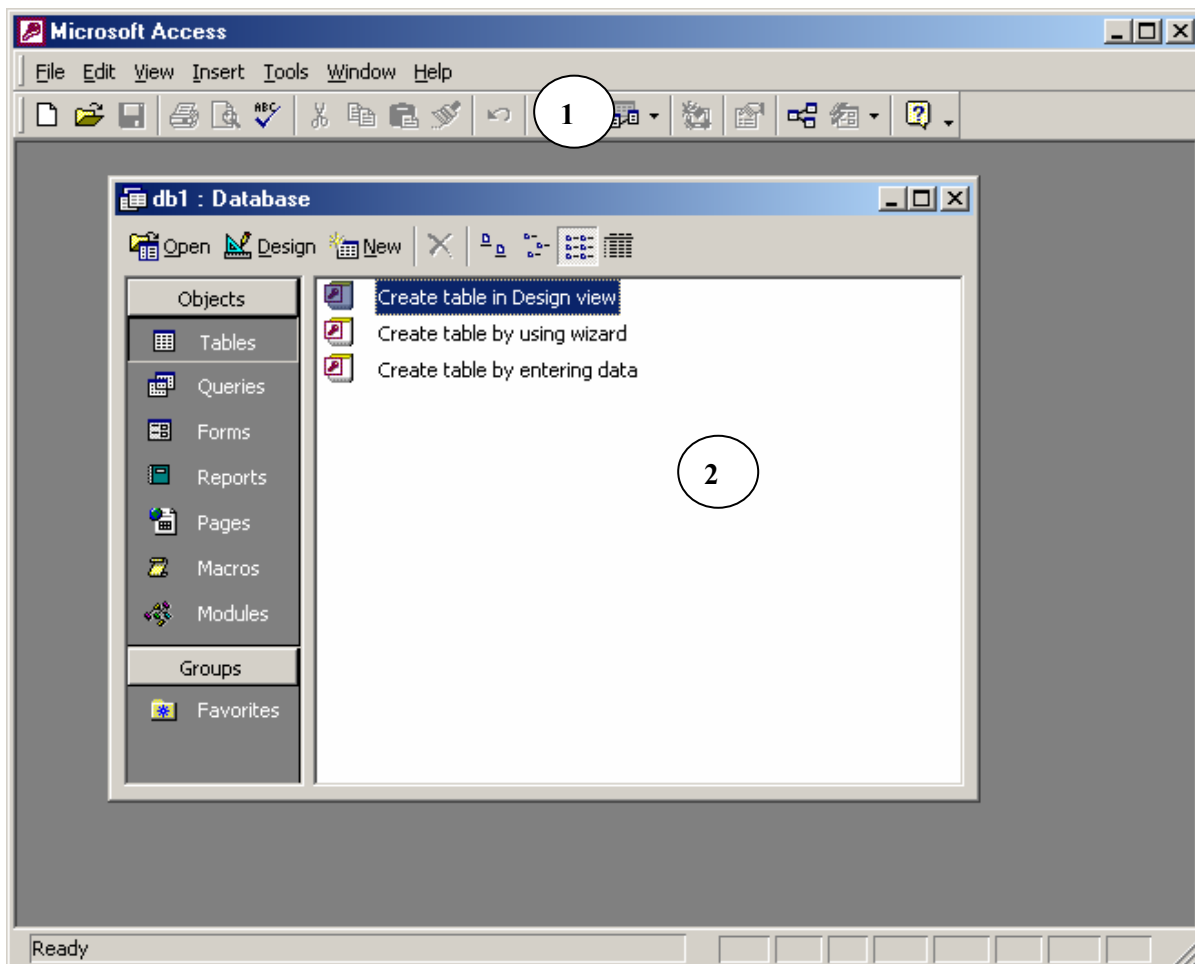


Hãy chọn nơi lưu trữ ở hộp **Save in**; gõ vào tên tệp ở mục **File Name**; nhấn  **Create** để ghi lại thiết lập. Đến đây đã sẵn sàng sử dụng tệp Access vừa khai báo.



## 4. Môi trường làm việc


Sau khi một tệp Access được mở, môi trường làm việc trên Access xuất hiện với những thành phần như sau:

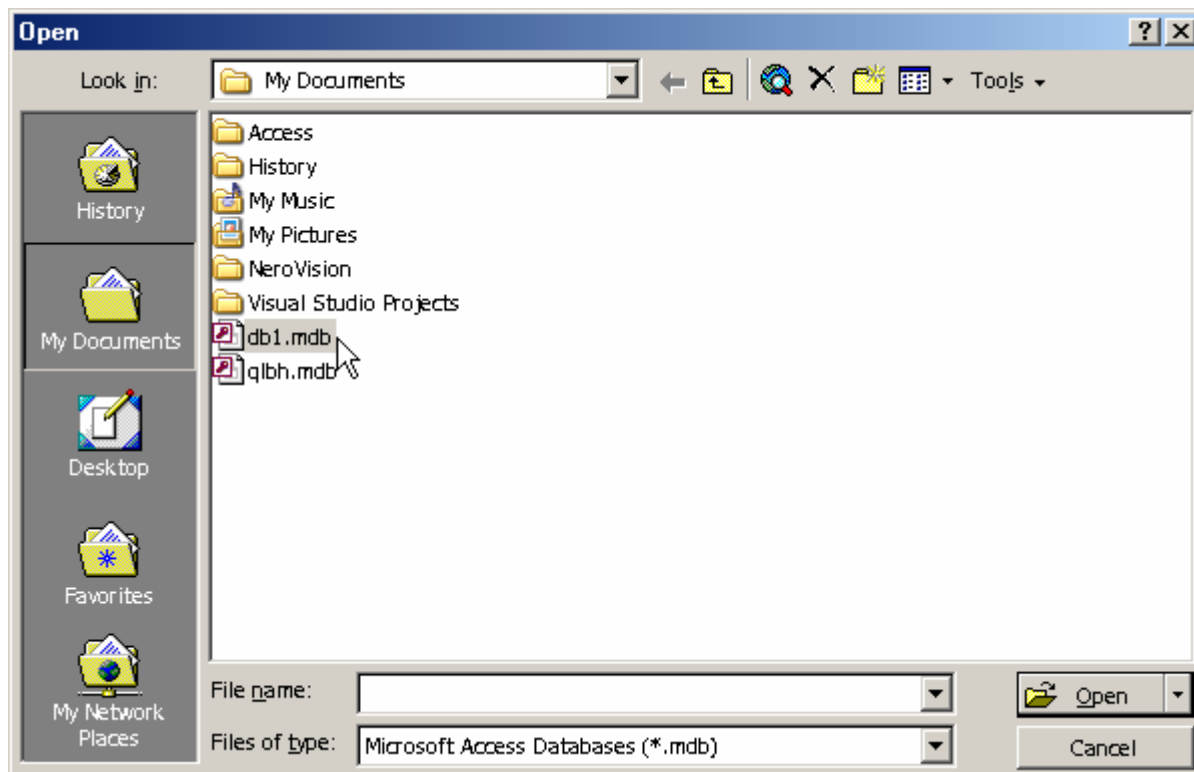


- (1) - hệ thống thực đơn (menu) và các thanh công cụ (Toolbar)- nơi thực hiện các lệnh khi cần;
- (2) - Cửa sổ tệp Access đang làm việc bao gồm 7 phần chính: Tables, Queries, Forms, Reports, Pages, Macros và Modules. Chúng ta sẽ lần lượt học cách làm việc trên từng phần này trong các chương tiếp theo.

## 5. Mở tệp đã tồn tại

Mỗi tệp Access phải tạo mới duy nhất một lần, được mở ra làm việc và ghi lại trong những lần tiếp theo. Để mở một tệp Access đã tồn tại để làm việc, làm như sau:

**Bước 1:** Từ môi trường Access gọi thực đơn: **File | Open** (hoặc nhấn nút **Open**  trên thanh công cụ), hộp thoại **Open** xuất hiện:




**Bước 2:** Tìm đến tệp Access cần mở trên hộp thoại Open bằng cách:

- Tìm đến thư mục – nơi chứa tệp Access cần mở ở hộp **Look in**;
- Tiếp theo chọn tệp Access cần mở trên danh sách và nhấn nút **Open** hoặc Enter.

Đến đây cửa màn hình làm việc Access với tệp vừa mở xuất hiện để tiếp tục làm việc.

## 6. Thoát khỏi Access

Khi không làm việc với Access, hãy ra lệnh thoát khỏi Access bằng một trong các cách:

- Mở thực đơn **File | Exit**;
- Nhấn tổ hợp phím nóng **Alt + F4**;
- Hoặc sử dụng nút Close  trên cửa sổ Access đang mở.

# CHƯƠNG 1

## XÂY DỰNG CƠ SỞ DỮ LIỆU

Xây dựng CSDL là công việc quan trọng đầu tiên trong toàn bộ qui trình phát triển một ứng dụng trên Access. Một CSDL được thiết kế và xây dựng tốt sẽ là những thuận lợi, những tự tin đầu tiên để bước vào một qui trình phát triển ứng dụng; nhưng trái lại sẽ là một thảm họa cho dự án đang phát triển: sẽ thường xuyên gặp phải những khó khăn để phải chỉnh sửa lại CSDL và tồi tệ hơn, dự án có thể phải thực hiện lại từ đầu do việc thiết kế CSDL quá kém.

Trong chương này sẽ trình bày các khái niệm cũng như các kỹ năng để học viên có thể tiếp cận và xây dựng được các hệ CSDL trên Access, cụ thể là:

- Tạo cấu trúc các bảng dữ liệu;
- Thiết lập thuộc tính LookUp;
- Thiết lập quan hệ và các thuộc tính đảm bảo toàn vẹn dữ liệu;
- Nhập dữ liệu cho CSDL;
- Đặc biệt là qui trình xây dựng một CSDL Access.

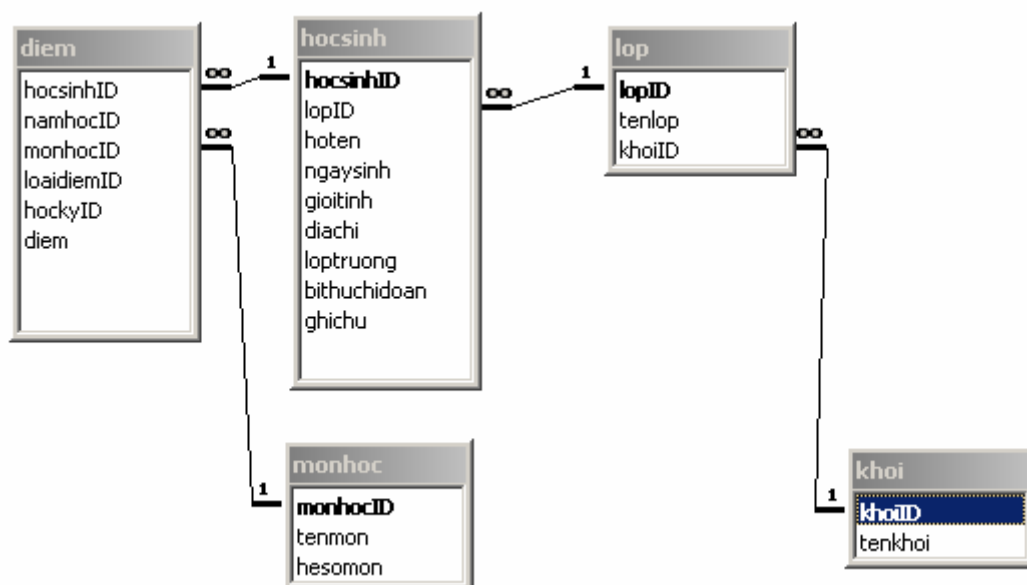
## 1. Các khái niệm về CSDL Access

### 1.1 CSDL Access

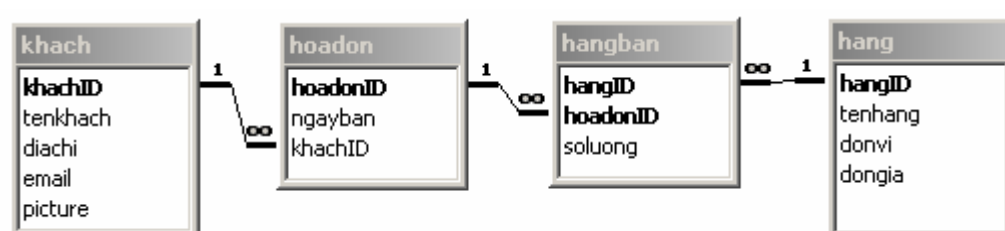
CSDL Access là một đối tượng bao gồm tập hợp các bảng dữ liệu, các kết nối giữa các bảng được thiết kế một cách phù hợp để phục vụ lưu trữ dữ liệu cho một ứng dụng quản lý dữ liệu nào đó.

#### Ví dụ:

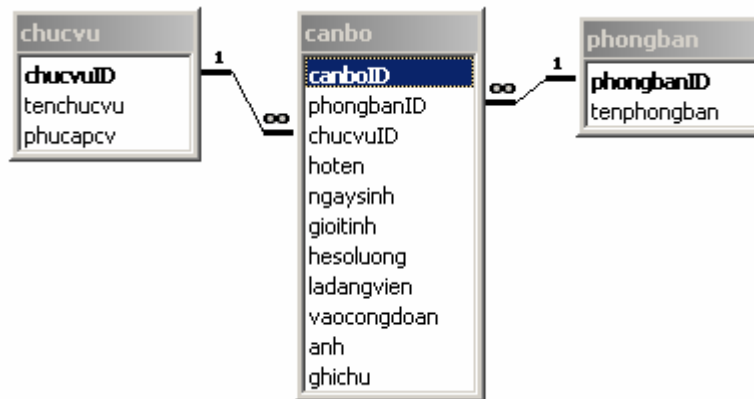
CSDL Quản lý học sinh bao gồm tập hợp các bảng dữ liệu: *HOCSINH*, *LOP*, *KHOI*, *MONHOC*, *DIEM* được kết nối với nhau một cách phù hợp phục vụ lưu trữ dữ liệu cho ứng dụng quản lý học sinh một trường học. Toàn bộ cấu trúc CSDL quản lý học sinh trên Access được mô tả như sau:



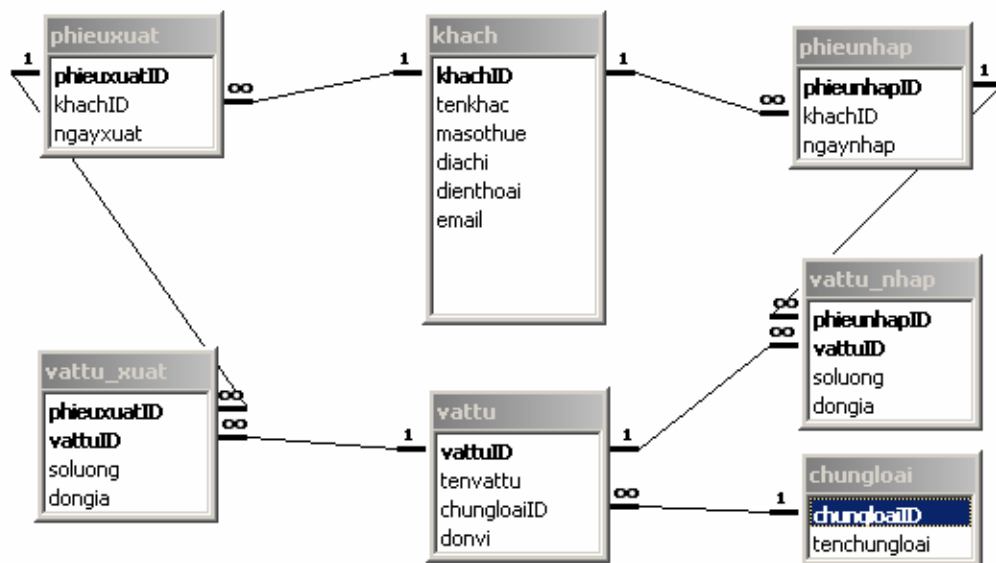
CSDL Quản lý bán hàng bao gồm tập hợp các bảng dữ liệu: *HANG*, *KHACH*, *HOADON*, *HANGBAN* được kết nối với nhau một cách phù hợp, phục vụ ứng dụng quản lý việc bán hàng tại một cửa hàng. Sơ đồ cấu trúc CSDL này như sau:



CSDL Quản lý lương cán bộ, bao gồm các bảng: *PHONGBAN*, *CHUCVU* và *CANBO* được kết nối với nhau phục vụ lưu trữ dữ liệu cho ứng dụng quản lý lương cán bộ một cơ quan. Cấu trúc CSDL này được biểu diễn như sau:



CSDL Quản lý việc nhập-xuất vật tư một cửa hàng. Bao gồm các bảng: *VATTU*, *KHACH*, *PHIEUNHAP*, *PHIEUXUAT*, *VATTU\_NHAP*, *VATTU\_XUAT* được biểu diễn như sau:



## 1.2 Bảng dữ liệu

Bảng dữ liệu (Tables) là một phần quan trọng nhất của CSDL; Là nơi lưu trữ những dữ liệu tác nghiệp cho ứng dụng. Một CSDL có thể có rất nhiều bảng, các bảng phải được thiết kế sao cho có thể lưu trữ được đầy đủ dữ liệu cần thiết, đảm bảo giảm tối đa tình trạng gây dư thừa dữ liệu (dư thừa dữ liệu được hiểu đơn giản là tình trạng lưu trữ những dữ liệu không cần thiết trên một số bảng. Tác hại của hiện tượng này sẽ gây: sai lệch dữ liệu tác nghiệp và làm tăng dung lượng dữ

liệu không cần thiết); giảm tối đa dung lượng CSDL có thể, đồng thời tạo môi trường làm việc thuận lợi cho việc phát triển ứng dụng trong các bước tiếp theo.

Một bảng dữ liệu trên Access bao gồm các thành phần: Tên bảng, các trường dữ liệu, trường khoá, tập hợp các thuộc tính cần thiết cho mỗi trường dữ liệu và tập hợp các bản ghi.

Mô tả một bảng dữ liệu trong trạng thái Datasheet (nhập, xem, sử dữ liệu):

	Mã hàng hoá	tenhang	donvi	dongia
+	h001	Pentium IV	chiếc	120
+	h002	Mainboard	chiếc	
+	h003	RAM	thanh	
▶	h004	HDD	ổ	85
+	h005	FDD	ổ	7
*				

Record: 4 of 5

Mô tả một bảng dữ liệu trong trạng thái Design view (đang thiết kế cấu trúc):

Field Name	Data Type	Description
hangID	Text	
tenhang	Text	
donvi	Text	
dongia	Number	

Field Name: hangID, Data Type: Text, Description: (empty)

Field Size: 10

Format: (empty)

Input Mask: (empty)

Caption: Mã hàng hoá

Default Value: (empty)

Validation Rule: (empty)

Validation Text: (empty)

Required: No

Allow Zero Length: Yes

Indexed: Yes (No Duplicates)

Unicode Compression: Yes

## Tên bảng

Mỗi bảng có một tên gọi. Tên bảng thường được đặt sau khi tạo xong cấu trúc của bảng, tuy nhiên cũng có thể đổi lại tên bảng trên cửa sổ Database như đổi tên tệp dữ liệu trên cửa sổ Windows Explorer.

Không nên sử dụng dấu cách (Space), các ký tự đặc biệt hoặc chữ tiếng Việt có dấu trong tên bảng.

## Trường dữ liệu (Field)

Mỗi cột dữ liệu của bảng sẽ tương ứng với một trường dữ liệu. Mỗi trường dữ liệu sẽ có một tên gọi và tập hợp các thuộc tính miêu tả trường dữ liệu đó ví dụ như: kiểu dữ liệu, trường khoá, độ lớn, định dạng, ..

Mỗi trường dữ liệu phải được định kiểu dữ liệu. Trong Access, trường dữ liệu có thể nhận một trong các kiểu dữ liệu sau:

TT	Kiểu dữ liệu	Độ lớn	Lưu trữ..
1	Number	Tùy thuộc kiểu cụ thể	Số: số thực, số nguyên theo nhiều kiểu
2	Autonumber	4 bytes	Số nguyên tự động được đánh số.
3	Text	tùy thuộc độ dài xâu	Xâu ký tự
4	Yes/No	1 bytes	Kiểu logic
5	Date/Time	8 bytes	Lưu trữ ngày, giờ
6	Currency	Sing	Lưu trữ dữ liệu kèm ký hiệu tiền tệ
7	Memo	tùy thuộc giá trị	kiểu ghi nhớ
8	Hyperlink	tùy thuộc độ dài xâu	Lưu trữ các siêu liên kết (hyperlink)
9	OLE	tùy thuộc dữ liệu	Âm thanh, hình ảnh, đồ hoạ, ... (Objects)

Không nên sử dụng dấu cách (Space), các ký tự đặc biệt hoặc chữ tiếng Việt có dấu trong tên trường.

### **Bản ghi (Record)**

Mỗi dòng dữ liệu của bảng được gọi một bản ghi. Mỗi bảng có một con trỏ bản ghi. Con trỏ bản ghi đang nằm ở bản ghi nào, người dùng có thể sửa được dữ liệu bản ghi đó. Đặc biệt, bản ghi trắng cuối cùng của mỗi bảng được gọi EOF.

### **Trường khoá (Primary key)**

Trường khoá có tác dụng phân biệt giá trị các bản ghi trong cùng một bảng với nhau. Trường khoá có thể chỉ 01 trường, cũng có thể được tạo từ tập hợp nhiều trường (gọi bộ trường khoá).

### **Ví dụ:**

(1)- bảng *THISINH* của CSDL thi tuyển sinh, trường khoá là *SoBaoDanh*. Vì mỗi thí sinh có thể nhiều trường có giá trị hết nhau, nhưng *SoBaoDanh* thì duy nhất.

(2)- bảng *CANBO* trường *MaCanBo* sẽ là trường khóa vì không thể tồn tại 2 cán bộ nào trong bảng này trùng *MaCanBo*

(3)- bảng *HANGBAN* của CSDL Quản lý bán hàng, 2 trường *hangID* và *hoadonID* là một bộ trường khoá. Vì không thể trên một hoá đơn bán hàng nào có bán một mặt hàng nào đó ghi lặp lại 2 lần.

## **1.3 Liên kết các bảng dữ liệu**

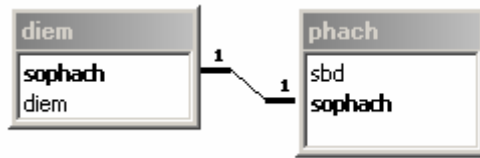
Liên kết các bảng dữ liệu là một kỹ thuật trong thiết kế CSDL quan hệ. Chúng là mối liên kết giữa 2 bảng với nhau theo thiết kế cho trước để đảm bảo được mục đích lưu trữ dữ liệu cho ứng dụng.

Trong Access tồn tại 2 kiểu liên kết: liên kết 1-1 và liên kết 1-n (một-nhiều)

**Liên kết 1-1** là: mỗi bản ghi của bảng này sẽ liên kết với duy nhất tới một bản ghi của bảng kia và ngược lại;

Ví dụ liên kết 1-1:





Mô tả dữ liệu 2 bảng này như sau:

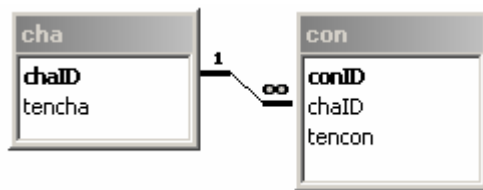
	sbd	sophach
	01	12
	02	02
	02	23
	03	14
	04	15
	05	32
	06	43
	07	08
▶	08	06
	09	10
*		

	sophach	diem
	12	5
	02	44
	23	0
	14	7
	15	7
	32	6
	43	9
	08	4
	06	7
	10	6
▶		0

**Liên kết 1-n** là: mỗi trường của bảng 1 sẽ có thể liên kết với một hoặc nhiều bản ghi của bảng nhiều (n). Ngược lại, mỗi bản ghi của bảng nhiều sẽ liên kết tới duy nhất 1 trường của bảng 1.

Ví dụ liên kết 1-n:



Có thể tham khảo mỗi cha có thể có nhiều con qua 2 bảng sau:

	chaID	tencha
▶	+ 01	Nguyễn Đức Sơn
	+ 02	Phan Anh Quân
	+ 03	Lê Hoàng
*		

	chaID	conID	tencon
▶	01	001	Nguyễn Đức Thìn
	01	002	Nguyễn Thị Thanh
	02	003	Phan Quang
	02	004	Phan Huy Hoàng
	02	005	Phan Thùy Anh
	03	006	Lê Văn Tám
*			

Hoặc xem theo một kiểu khác:

chaID	tencha
01	Nguyễn Đức Sơn
02	Phan Anh Quân
03	Lê Hoàng

conID	tencon
001	Nguyễn Đức Thìn
002	Nguyễn Thị Thanh
003	Phan Quang
004	Phan Huy Hoàng
005	Phan Thủy Anh
006	Lê Văn Tám

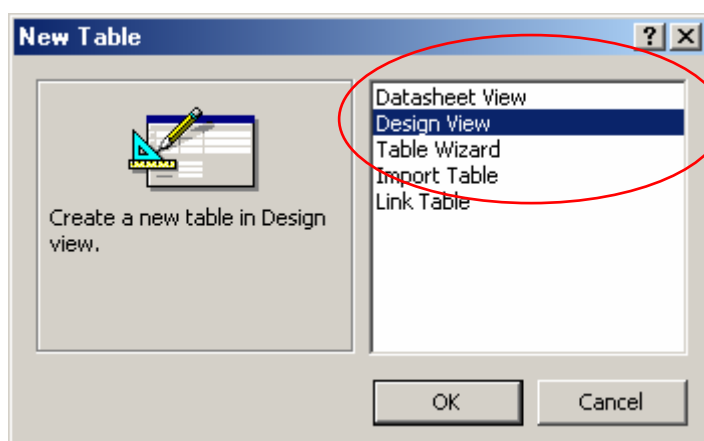
## 2. Xây dựng cấu trúc bảng

Mục này hướng dẫn cách thiết kế cấu trúc một bảng dữ liệu trên CSDL Access. Để có thể làm tốt được công việc này, đề nghị học viên nên tuân thủ theo thứ tự các bước sẽ liệt kê dưới đây.

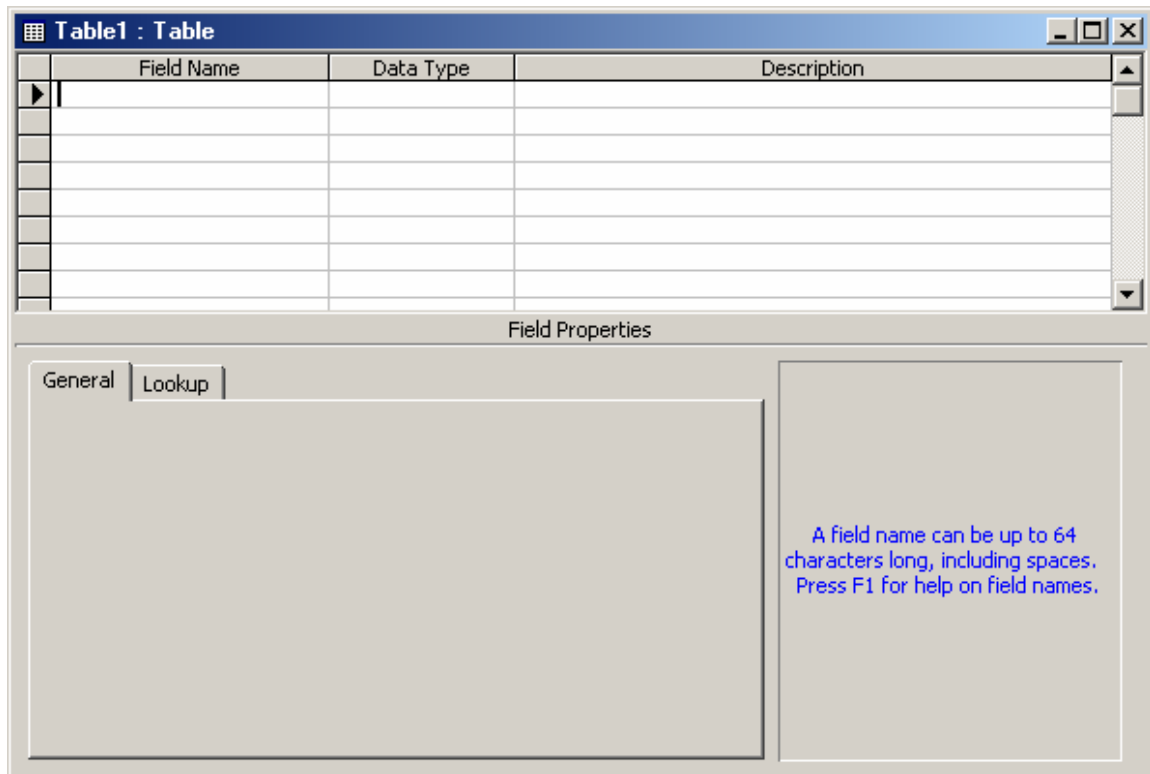
Minh họa này hướng dẫn cách tạo cấu trúc bảng HANG bao gồm các trường *hangID*, *tenhang*, *donvi*, *dongia*.

**Bước 1:** Khởi động trình thiết kế cấu trúc bảng ở chế độ Design View

Ở thẻ Tables, nhấn nút **New**, chọn **Design View**, nhấn **OK**



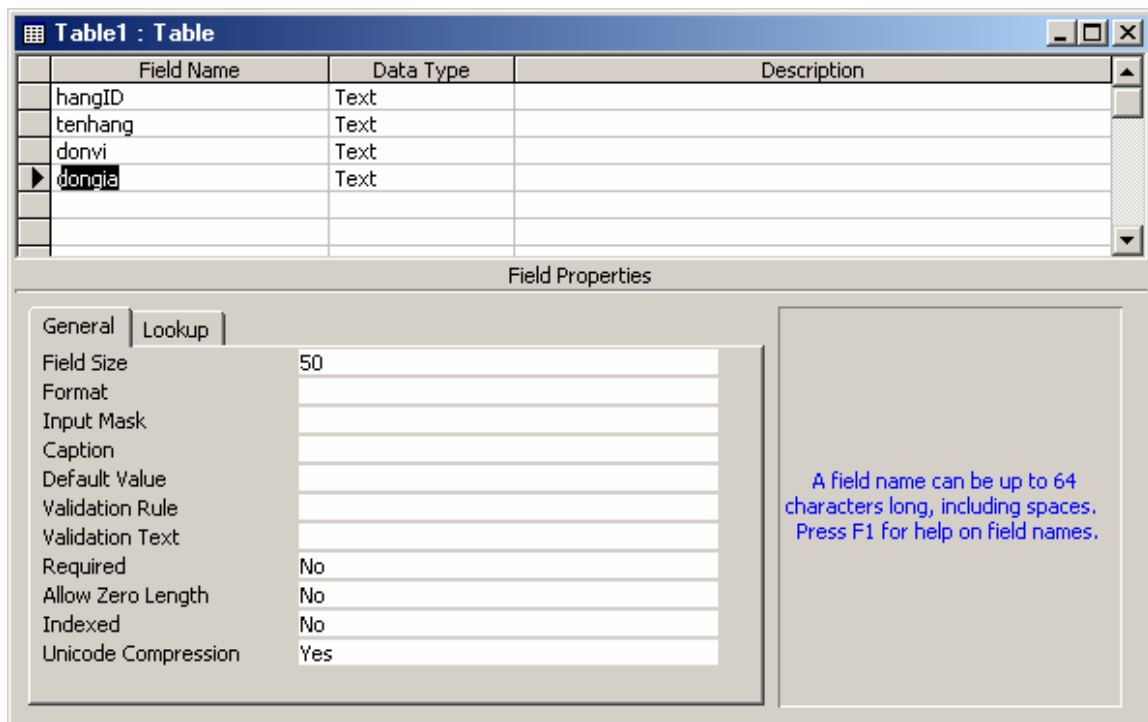
Hoặc nhấn  **Create table in Design view** trên thẻ Tables. Hộp thoại thiết kế cấu trúc một bảng xuất hiện:



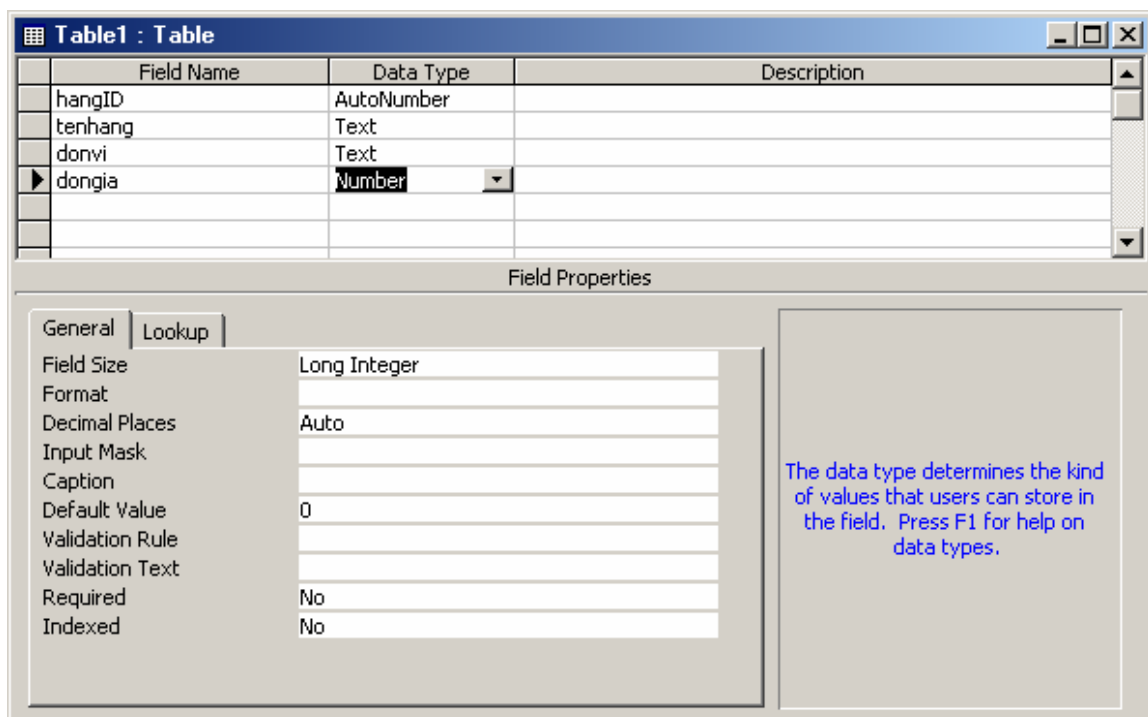
**Bước 2:** Khai báo danh sách tên các trường của bảng: bằng cách gõ danh sách tên các trường lên cột **Field Name** của cửa sổ thiết kế.

**Chú ý:** tên trường không nên chứa dấu cách (space), chữ tiếng Việt có dấu.

Sau khi gõ vào danh sách tên các trường của bảng HANG, hộp thoại thiết kế sẽ có dạng:



**Bước 3:** Khai báo kiểu dữ liệu cho các trường của bảng, bằng cách chọn kiểu dữ liệu cho từng trường ở cột **Data Type** tương ứng. Sau khi chọn xong kiểu dữ liệu cho các trường, hộp thoại thiết kế sẽ có dạng:




Để chọn kiểu dữ liệu, có thể dùng chuột chọn kiểu dữ liệu từ hộp thả; mặt khác cũng có thể nhấn ký tự đầu tiên của kiểu dữ liệu cần chọn mỗi khi định vị đến ô Data Type cần làm việc. Ví dụ:


Nhấn	Sẽ được kiểu
A	<u>A</u> utonomous
N	<u>N</u> umber
T	<u>T</u> ext
Y	<u>Y</u> es/No
D	<u>D</u> ate/Time

Nhấn	Sẽ được kiểu
M	<u>M</u> emo
C	<u>C</u> urrency
H	<u>H</u> yperlink
O	<u>O</u> LE


**Bước 4:** Thiết lập trường khoá cho bảng (những bảng không có trường khoá có thể bỏ qua bước này).

- Chọn các trường muốn thiết lập khoá bằng cách: dùng chuột kết hợp giữ phím **Shift** đánh dấu đầu dòng các trường muốn thiết lập khoá;
- Mở thực đơn **Edit | Primary key** để thiết lập thuộc tính khoá cho các trường vừa chọn. Cũng có thể ra lệnh này bằng cách nhấn nút *Primary key*  trên thanh công cụ.

Sau khi thiết lập khoá, những trường khoá sẽ có biểu tượng như sau:

	Field Name	Data Type
	hangID	Text
	tenhang	Text
	donvi	Text
	dongia	Number

Biểu tượng của trường khoá

**Bước 5:** Lưu lại cấu trúc bảng. Nhấn tổ hợp phím **Alt + S** hoặc nhấn nút  Save trên thanh công cụ, hộp thoại yêu cầu ghi tên cho bảng xuất hiện:

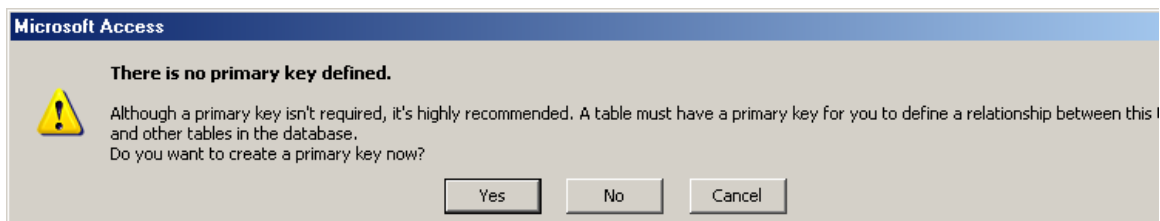


The image shows a 'Save As' dialog box with a text input field containing 'HANG'. The text 'HANG' is circled in red. There are 'OK' and 'Cancel' buttons to the right of the input field.

Hãy gõ tên bảng và nhấn **OK**.

#### Đặc biệt:

- Với những bảng không thiết lập trường khoá, trong quá trình ghi lại cấu trúc bảng, máy tính sẽ hỏi:



Nhấn **Yes**- máy tính sẽ tạo thêm một trường mới có tên ID và thiết lập trường này làm khoá. Nếu không muốn như vậy hãy nhấn **No**; nhấn **Cancel** để huỷ lệnh cất.

- Tên bảng không nên chứa dấu cách, các ký tự đặc biệt khác hoặc chữ Việt có dấu.

Mỗi trường dữ liệu được khai báo trong cửa sổ trên đều có thể thiết lập được rất nhiều các thuộc tính tùy thuộc kiểu dữ liệu trường đó đã nhận. Các thuộc tính này có thể thiết lập tại phần *Tập hợp các thuộc tính của các trường* như đã trình bày ở trên. Dưới đây là danh sách một số các thuộc tính hay được sử dụng.

#### Thuộc tính **Field size**

Để thiết lập kích thước dữ liệu. Chỉ áp dụng cho các trường có kiểu dữ liệu Number và Text.

Đối với các trường kiểu số, **Field size** cho biết trường đó nhận giá trị loại số nào.

TT	Loại số	Độ lớn
1	Byte	số nguyên 2 byte
2	Integer	số nguyên 4 byte
3	Long Integer	số nguyên 6 byte
4	Single	số thực 8 byte
5	Double	số thực 12 byte
6	Decimal	số thực 16 byte

Đối với các trường kiểu Text, thuộc tính này cho biết chiều dài tối đa của xâu ký tự. Ví dụ: Với trường *Hoten* thì Field size khoảng 30.

#### Thuộc tính **Format**

Để thiết lập định dạng dữ liệu khi hiển thị. Trường này áp dụng cho hầu hết các kiểu dữ liệu trừ ra kiểu: Memo, OLE, Yes/No

#### Thuộc tính **Input Mark**

Thiết lập mặt nạ nhập dữ liệu cho các trường. Kiểu này có thể áp dụng cho các loại trường kiểu Text, Number, Datetime, Currency.

#### Thuộc tính **Default Value**

Để thiết lập giá trị ngầm định cho trường mỗi khi ra lệnh thêm mới một bản ghi. Ví dụ: Trường số lượng mỗi khi thêm một bản ghi mới, giá trị trường này tự động là 1. Khi đó phải thiết lập thuộc tính **Default Value** của trường này là **1**.

#### Thuộc tính **Caption**

Thiết lập tiêu đề cột mà trường đó hiển thị. Tên trường không nên chứa dấu cách và chữ Việt có dấu, nhưng *Caption* của các trường thì nên gõ bằng tiếng Việt có dấu sao cho dễ đọc và nhận biết. Đặc biệt giá trị thuộc tính Caption nếu có sẽ được sử dụng làm tiêu đề cho các trường tương ứng mỗi khi sử dụng công cụ Form Wizard hay Report Wizard sau này- sẽ rất tiện lợi.

#### Thuộc tính **Validation Rule**

Thiết lập điều kiện kiểm tra tính đúng đắn của dữ liệu khi được nhập vào.

Ví dụ: trường *NgaySinh* của học sinh phải nhập vào những ngày  $\geq 1/1/1980$  chẳng hạn. Khi đó ở thuộc tính *Validation Rule* của trường *Ngaysinh* hãy gõ vào  **$\geq \#1/1/1980\#$**

#### Thuộc tính **Required**

Để yêu cầu phải nhập dữ liệu cho trường này (nếu thiết lập **Yes**) khi bắt đầu một bản ghi mới hoặc không nếu thiết lập **No**.

Còn một số các thuộc tính khác xin mời tham khảo qua phần Help của Access.

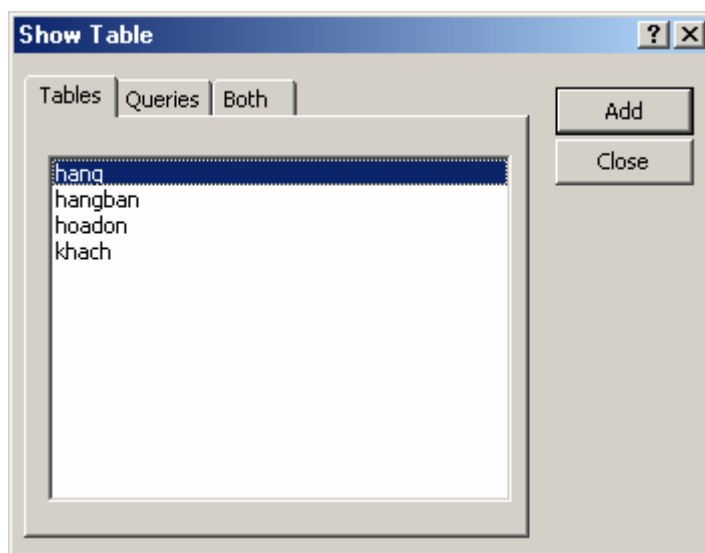
### 3. Thiết lập quan hệ

Một bước quan trọng trong xây dựng CSDL Access là thiết lập quan hệ các bảng trong CSDL. Làm được điều này bạn sẽ gặp được rất nhiều thuận lợi trong quá trình sử dụng các trình Wizard và Design View trong Access sau này.

Dưới đây sẽ là hướng dẫn cách thiết lập quan hệ cho một cặp bảng. Tương tự bạn phải thiết lập toàn bộ các quan hệ có thể trên CSDL.

**Bước 1:** Mở cửa sổ thiết lập quan hệ bởi thực đơn: **Tools | Relationship..**

**Bước 2:** Đưa các bảng (Tables) tham gia thiết lập quan hệ thông qua hộp thoại Show Tables (nếu chưa thấy hộp thoại này dùng thực đơn **Relationship | Show table**):



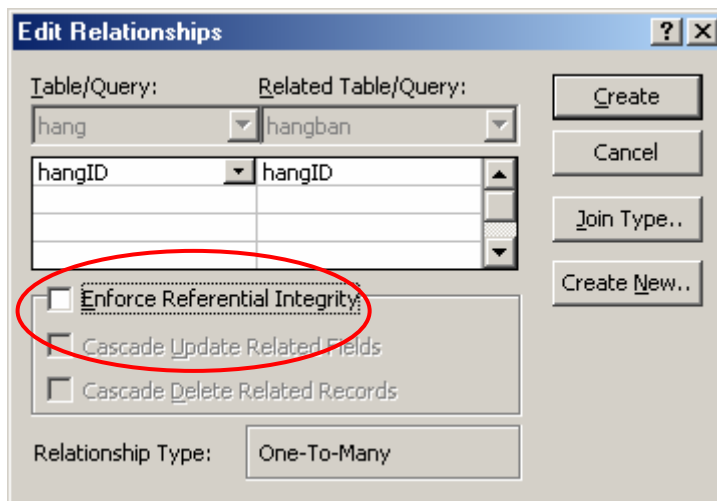
Cách đưa các bảng lên cửa sổ thiết lập quan hệ (Database) như sau:

- Chọn bảng cần tham gia thiết lập quan hệ (thường thì chọn tất cả) ;
- Nhấn nút **Add**;
- Chọn xong toàn bộ nhấn **Close** để đóng cửa sổ.

**Bước 3:** Thực hiện tạo kết nối giữa từng cặp bảng theo thiết kế, cách làm như sau:

Dùng chuột kéo (Drag) trường cần liên kết của bảng này (ví dụ trường hangID của bảng HANG) thả (Drop) lên trường cần liên kết đến của bảng kia (ví dụ trường hangID của bảng HANGBAN). Khi đó hộp thoại **Edit Relationships** xuất hiện:





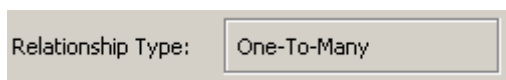
Trong trường hợp muốn thiết lập các thuộc tính đảm bảo toàn vẹn dữ liệu (Enforce Referential Integrity) cho quan hệ hãy thực hiện chọn (checked) 3 mục chọn sau:

**Enforce Referential Integrity** để đồng ý thiết lập các thuộc tính đảm bảo toàn vẹn dữ liệu;

**Cascade Update Related Fields** đảm bảo toàn vẹn dữ liệu khi xoá dữ liệu giữa 2 bảng liên quan. Khi đó, nếu một bản ghi ở bảng có quan hệ 1 bị xoá, toàn bộ các bản ghi có quan hệ với bản ghi hiện tại sẽ được tự động xoá ở bảng có quan hệ nhiều (nếu xoá 1 CHA, toàn bộ các con của cha đó sẽ tự động bị xoá khỏi bảng CON);

**Cascade Delete Related Records** đảm bảo toàn vẹn dữ liệu khi cập nhật dữ liệu giữa 2 bảng liên quan. Khi đó, nếu giá trị trường khoá liên kết ở bảng 1 bị thay đổi, toàn bộ giá trị trường khoá liên kết ở bảng nhiều cũng bị thay đổi theo.

Hộp **Relationship Type**: cho biết kiểu quan hệ giữa 2 bảng đang thiết lập:



- **One – To – One** Kiểu 1-1
- **One – To – Many** Kiểu 1-∞
- **Indeterminate** Không xác định được kiểu liên kết

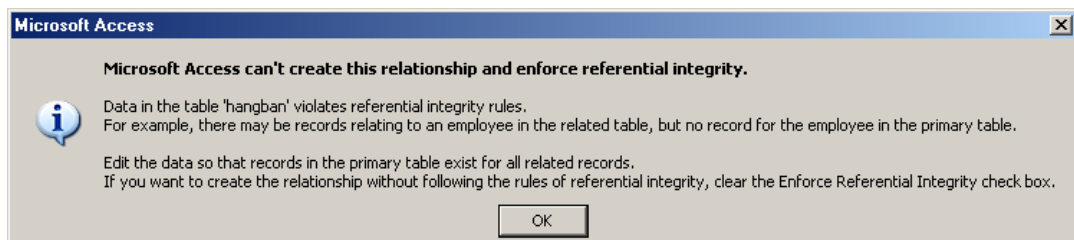
Tùy thuộc vào kiểu khoá của các trường tham gia liên kết mà Access tự xác định ra được kiểu liên kết giữa 2 bảng. Dưới đây là một số kiểu liên kết được Access tự động xác định:

TT	Bảng A	Bảng B	Kiểu liên kết
1	Khoá chính	Khoá chính	1-1
2	Khoá chính	Khoá phụ (hoặc không khoá)	1-n
3	Khoá phụ	Khoá phụ (hoặc không khoá)	Không xác định được kiểu liên kết
4	Không khoá	Không khoá	Không xác định được kiểu liên kết



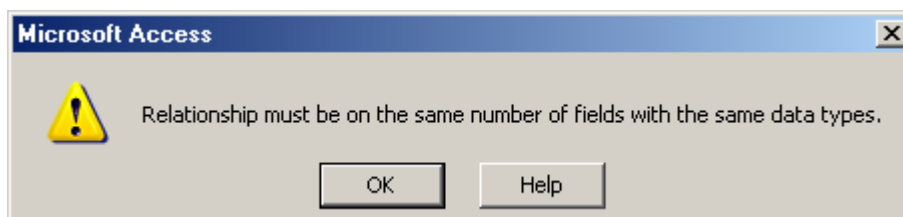
### Chú ý

- Khi hộp Relationships Type chỉ **Indeterminate** có nghĩa là quan hệ đang thiết lập không đúng về cấu trúc khoá của 2 bảng (quan hệ sai);
- Trong trường hợp thiết lập các thuộc tính đảm bảo toàn vẹn dữ liệu nếu gặp phải hộp thoại thông báo lỗi:



Lỗi này do một số dữ liệu đang tồn tại trong bảng quan hệ  $\infty$  không thoả mãn với bảng quan hệ 1. Hiểu đơn giản là: có một số mặt hàng khai báo trong bảng HANGBAN chưa có trong danh mục hàng (bảng HANG) → như vậy là không hợp lý, không thoả mãn các điều kiện về toàn vẹn dữ liệu giữa 2 bảng này.

- Khi gặp phải lỗi sau đây khi kết nối giữa 2 bảng:



Tức là 2 trường tham gia kết nối không cùng kiểu dữ liệu. Ví dụ: một trường kiểu Number, trường kia kiểu Text; hoặc một trường kiểu Text, trường kia kiểu Date/Time - đều bị sai vì không cùng kiểu dữ liệu.

## 4. Nhập dữ liệu

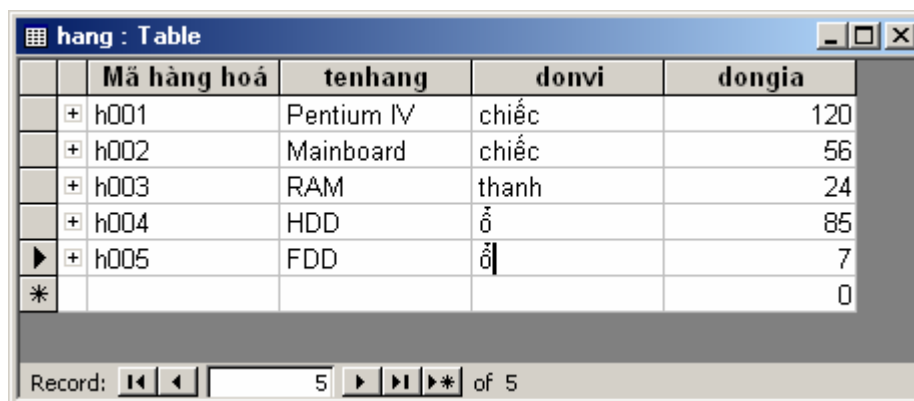
Nhập dữ liệu là yêu cầu bắt buộc đối với bất kỳ một ứng dụng nào. Dữ liệu có thể được nhập vào CSDL bằng nhiều con đường khác nhau. Trong phần này sẽ trình bày cách nhập dữ liệu từ bàn phím vào trực tiếp các bảng dữ liệu trong Access.

### 4.1 Cách nhập dữ liệu

Có 2 bước để có thể nhập trực tiếp dữ liệu vào bảng:

**BƯỚC 1:** Mở bảng để nhập dữ liệu bằng cách: nhấn đúp chuột lên tên bảng cần nhập dữ liệu; hoặc chọn bảng cần nhập dữ liệu rồi nhấn nút Open;

**BƯỚC 2:** Thực hiện nhập dữ liệu vào bảng đang mở bằng bàn phím



	Mã hàng hoá	tenhang	donvi	dongia
+	h001	Pentium IV	chiếc	120
+	h002	Mainboard	chiếc	56
+	h003	RAM	thanh	24
+	h004	HDD	ổ	85
▶	h005	FDD	ổ	7
*				0

Record: 5 of 5



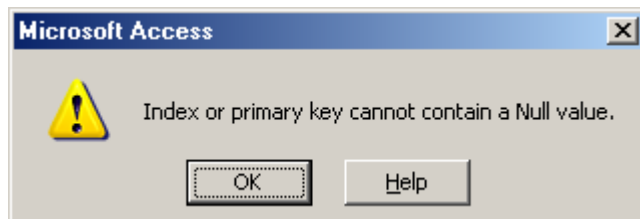
**Chú ý :** Một số lỗi có thể xảy ra khi nhập dữ liệu

#### Lỗi thứ nhất:



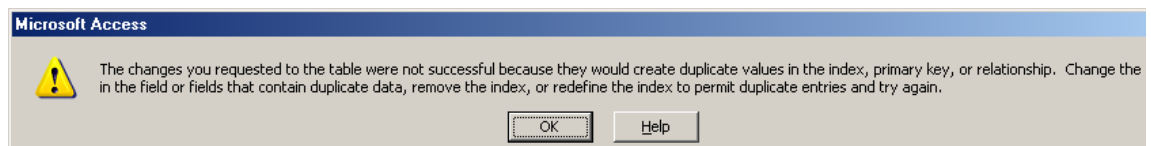
**Lỗi do:** Bạn đã nhập vào giá trị không tương thích với kiểu dữ liệu của trường đã chỉ định. Ví dụ: trường kiểu Numeric mà gõ vào chữ cái; hoặc không gõ đầy đủ các giá trị ngày, tháng, năm cho trường kiểu Date/Time,.. lỗi này sẽ xuất hiện.

**Khắc phục:** hãy nhập lại cho đúng, đủ giá trị các trường đã yêu cầu đến khi không xuất hiện thông báo lỗi.

**Lỗi thứ 2:**

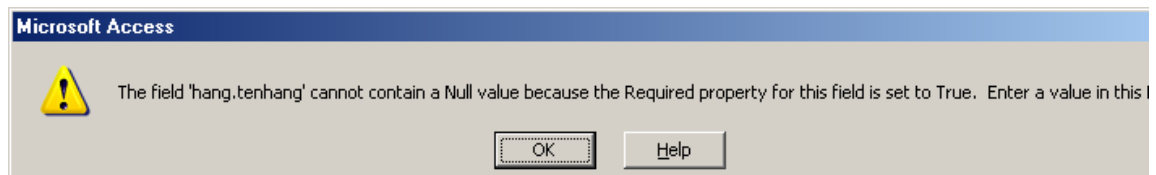
**Lỗi do:** Bạn không nhập giá trị hoặc để trống giá trị trường khoá. Đã là trường khoá luôn yêu cầu phải nhập dữ liệu cho mỗi bản ghi.

**Khắc phục:** phải nhập đầy đủ giá trị cho trường khoá.

**Lỗi thứ 3:**

**Lỗi do:** Giá trị trường khoá trùng nhau. Giá trị trường khoá vừa nhập vào đã trùng với giá trị của một bản ghi nào đó trên bảng dữ liệu.

**Khắc phục:** nhập lại giá trị trường khoá khác sao cho vừa đúng, đủ và không bị trùng khoá.

**Lỗi thứ 4:**

**Lỗi do:** Bản ghi vừa nhập dữ liệu đã bỏ trống trường bắt buộc nhập dữ liệu (những trường được thiết lập thuộc tính *Required=Yes*)

**Khắc phục:** Phải nhập đủ dữ liệu cho các trường bắt buộc phải nhập dữ liệu.

**Lỗi thứ 5:**

**Lý do:** Lỗi do thực hiện một thao tác vi phạm các nguyên tắc đảm bảo toàn vẹn dữ liệu. Ví dụ: Bạn đã nhập dữ liệu trên một bảng có quan hệ mà bản ghi đang nhập không thể liên

kết được tới được một bản ghi nào của bảng có quan hệ 1 với nó (nhập một hàng bán mà mã hàng đó chưa có trong bảng danh mục hàng hoá).

**Khắc phục:** Tìm và nhập cho đúng giá trị theo bảng quan hệ 1 tương ứng. Tham khảo cách khắc phục lỗi này ở mục 5: thuộc tính Lookup.


## 4.2 Một số thao tác xử lý dữ liệu trên bảng

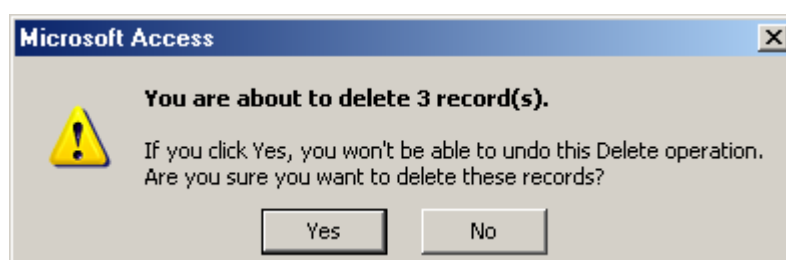
Đứng trước một bảng dữ liệu, có rất nhiều thao tác cần xử lý, ví dụ: sắp xếp bảng, tìm kiếm các bản ghi, xoá bản ghi nào đó,... Nội dung phần này trình bày cách sử dụng một số thao tác đó.

### a. Xoá bản ghi

Xoá bản ghi là thao tác xoá bỏ một số bản ghi ra khỏi bảng. Với bảng dữ liệu đang mở có thể thực hiện 2 bước sau để xoá các bản ghi:

**Bước 1:** Chọn những bản ghi cần xoá. Có thể chọn một hoặc nhiều bản ghi bằng cách dùng chuột đánh dấu đầu dòng những bản ghi cần chọn;

**Bước 2:** Ra lệnh xoá bằng cách: mở thực đơn **Edit | Delete Record** hoặc nhấn nút **Delete Record** trên thanh công cụ  hoặc nhấn phải chuột lên vùng đã chọn, tiếp theo nhấn **Delete Record**. Một hộp thoại xuất hiện để bạn khẳng định một lần nữa việc xoá dữ liệu:



- Chọn **Yes** để đồng ý xoá;
- Nhấn **No** để huỷ lệnh xoá.



### **Chú ý**

Dữ liệu đã đồng ý xoá sẽ không thể phục hồi lại được. Nên phải cân nhắc trước quyết định này cũng như cân nhắc trước các câu hỏi kiểu Yes/No trước khi quyết định.

## b. Sắp xếp dữ liệu

Sắp xếp là việc thay đổi thứ tự hiển thị một bảng dữ liệu theo một trật tự nào đó. Kết quả của việc sắp xếp giúp người dùng có thể quan sát được tốt hơn dữ liệu trên bảng, tất nhiên muốn quan sát bảng dữ liệu theo trường nào phải thực hiện sắp xếp bảng theo dữ liệu trường ấy. Cách sắp xếp dữ liệu trên bảng đang mở như sau:

**Bước 1:** Đặt con trỏ lên trường (cột) muốn sắp xếp;

**Bước 2:** Nhấn nút lệnh sắp xếp trên thanh công cụ:  - sắp xếp tăng dần hoặc  - sắp xếp giảm dần.

Bạn sẽ thu được kết quả như mong muốn.

## c. Lọc dữ liệu

Lọc dữ liệu là việc lọc ra những bản ghi trên bảng có cùng một số giá trị. Kết quả việc lọc dữ liệu sẽ giúp người dùng làm việc một cách hiệu quả trên tập hợp các bản ghi họ mong muốn.

Một ví dụ về tính hiệu quả của việc lọc dữ liệu:

Sắp đến ngày Quốc tế Phụ nữ 8-3. Cơ quan quyết định thưởng mỗi chị em 100,000đ. Đã có cột thưởng trên bảng cùng danh sách tất cả cán bộ cơ quan, làm sao để có thể nhập vào giá trị cột thưởng cho chị em (không nhập cho nam giới) là nhanh nhất.

Nếu cứ lần mò xem ai là nữ rồi nhập 100,000 cho cột thưởng sẽ là rất lâu. Nếu làm theo cách sau:


**Bước 1:** Lọc ra danh sách là các cán bộ nữ;


**Bước 2:** Nhập 100,000 cho tất cả các bản ghi đã lọc. Như vậy chắc chắn những cán bộ đang hiển thị (sau khi đã lọc) đều là chị em phụ nữ, không bỏ sót ai mà cũng không nhầm một người nam nào được lọt vào danh sách thưởng!

Các bước để lọc dữ liệu trên một bảng đang mở như sau:

**BƯỚC 1:** Nhấn phải chuột lên trường cần lọc dữ liệu. Một menu xuất hiện:

**BƯỚC 2:** Thiết lập điều kiện lọc trên trường đang chọn. Có rất nhiều cách để xác định điều kiện lọc:

- Nếu muốn lọc những bản ghi có cùng giá trị của bản ghi đang chọn hãy chọn mục 

- Muốn lọc những bản ghi thoả mãn điều kiện nào đó, hãy gõ điều kiện lên mục:  Ví dụ:

+ Gõ **>=10** - để lọc ra những bản ghi có giá trị trường đang thiết lập lọc lớn hơn hoặc bằng 10;


+ Gõ **<>3** - lọc ra những bản ghi có giá trị trường đang lọc khác 3,...

+ Đặc biệt: giá trị trống, rỗng được miêu tả là **Null**;

+ Lọc ra những người tên Nam gõ như sau **Like '\*Nam'**;

+ Lọc ra giá trị trong khoảng 10 đến 15 gõ như sau: **Between 10 And 15** (tham khảo toán tử Like và Between ở chương sau- Queries).

Muốn huỷ chế độ đặt lọc, nhấn phải chuột lên bảng dữ liệu và chọn mục:



## 5. Thuộc tính LOOKUP

Qua cách nhập dữ liệu cho bảng có quan hệ nhiều trên CSDL ta thấy việc nhập dữ liệu cho trường tham gia liên kết của bảng nhiều đòi hỏi phải có độ chính xác với dữ liệu trên bảng quan hệ 1 (phải nhớ mã để nhập). Trong thực tế với những danh mục lên đến hàng trăm, thậm chí nhiều hơn nữa thì việc nhớ mã để nhập dữ liệu quả là khó khăn: hoặc gõ sai mã, nguy hiểm hơn gõ đúng nhưng nhầm mã. Thuộc tính LOOKUP sẽ giúp giải quyết phần nào việc khó khăn trong nhập dữ liệu trên các bảng quan hệ nhiều như vậy.

Thuộc tính LOOKUP được thiết lập tại trường tham gia liên kết trên bảng có quan hệ nhiều sang trường tham gia liên kết của bảng có quan hệ 1.

Ví dụ: Với CSDL Quản lý bán hàng thì:

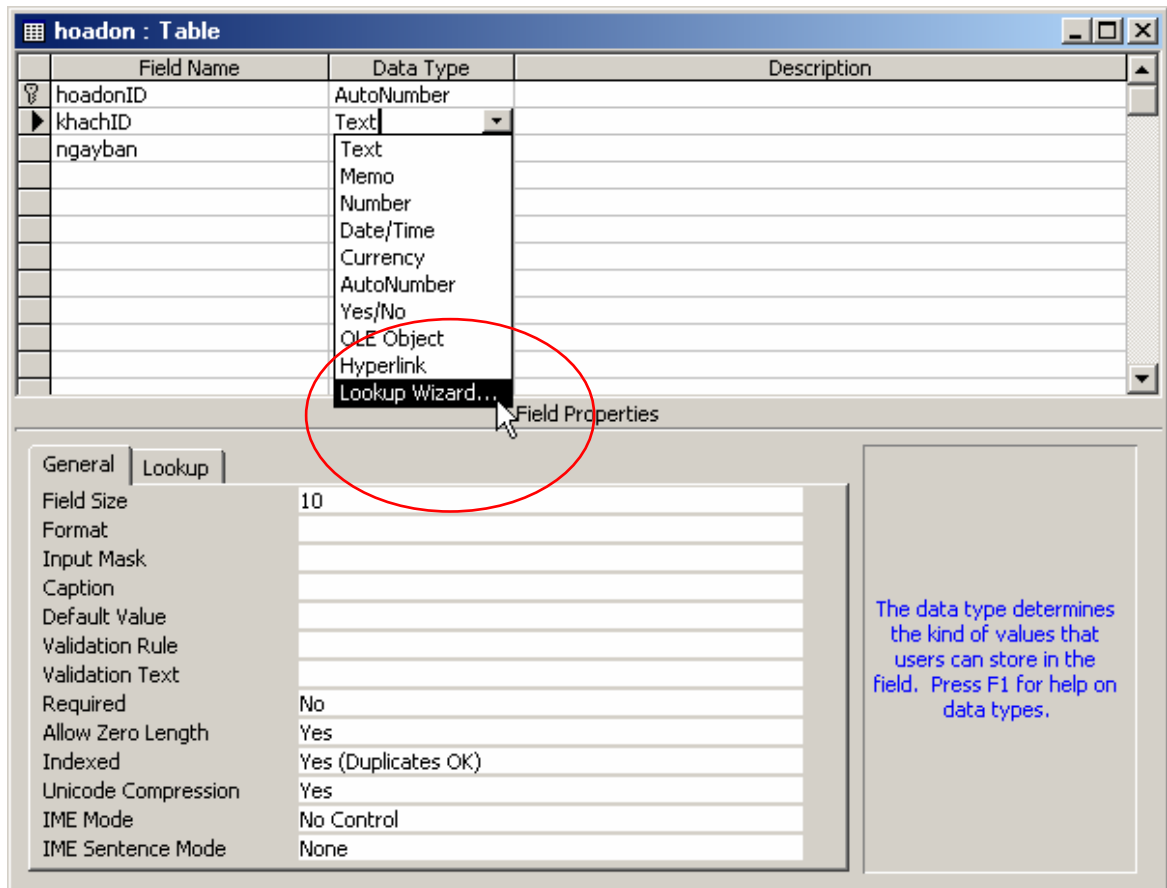
- trường *khachID* của bảng *HOADON* phải thiết lập thuộc tính LOOKUP sang trường *khachID* của bảng *KHACH*;
- trường *hangID* của bảng *HANGBAN* phải thiết lập thuộc tính LOOKUP sang trường *hangID* của bảng *HANG*;
- trường *hoadonID* của bảng *HANGBAN* phải thiết lập thuộc tính LOOKUP sang trường *hoadonID* của bảng *HOADON*.

Thông thường, ứng với mỗi quan hệ 1-∞ đã được thiết kế cần phải thiết lập thuộc tính LOOKUP cho trường tham gia liên kết từ bảng quan hệ 1 sang trường tham gia liên kết của bảng quan hệ nhiều. Một trong các cách thiết lập thuộc tính này đơn giản nhất là trình LookUp Wizard của Access. Dưới đây là ví dụ về thiết lập thuộc tính LOOKUP cho trường *khachID* của bảng *HOADON* sang trường *khachID* của bảng *KHACH* trong CSDL quản lý bán hàng:

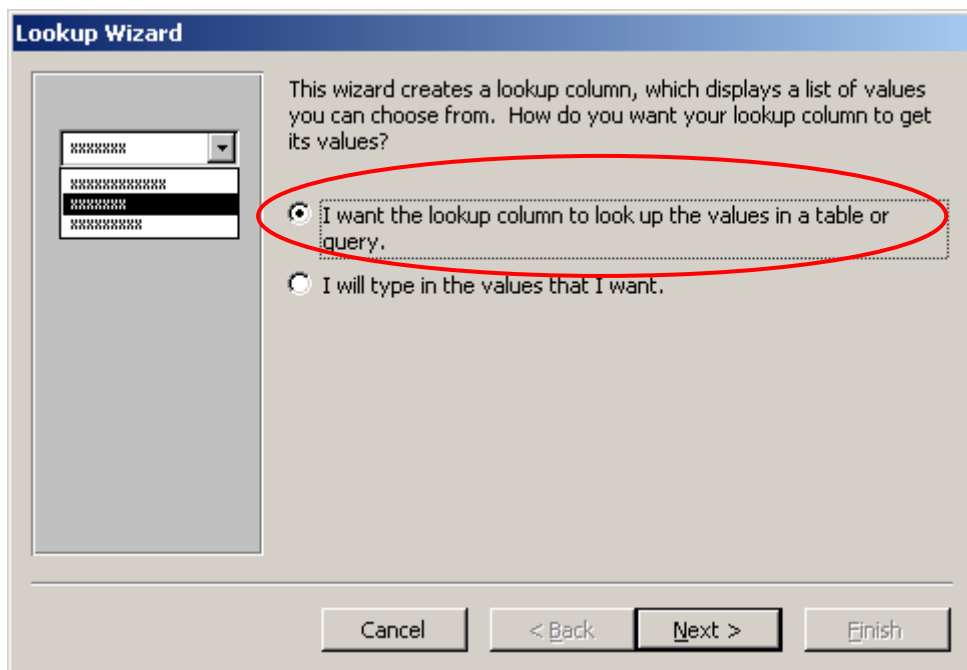
**Bước 1:** Mở bảng có trường cần thiết lập LOOKUP ra (bảng *HOADON*) ở chế độ Design View bằng cách: chọn tên bảng, nhấn nút **Design**;

**Bước 2:** Kích hoạt trình LookUp Wizard bằng cách: Tại cột Data Type của trường cần thiết lập thuộc tính LOOKUP (trường *khachID*), chọn mục **Lookup Wizard..** từ danh sách thả xuống:





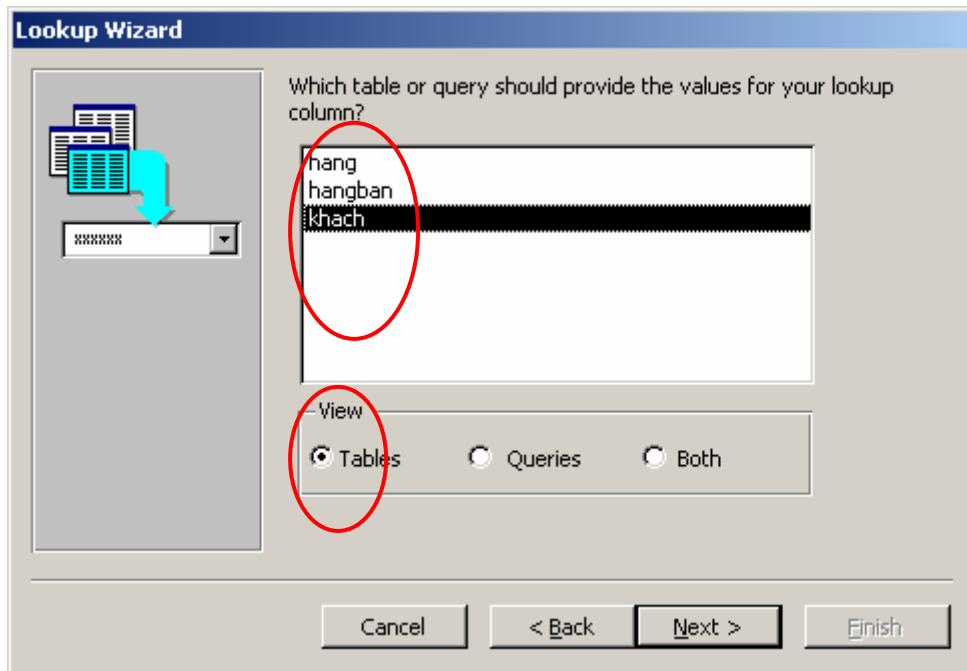
Hộp thoại Lookup Wizard xuất hiện:



Hộp thoại này hỏi *Dữ liệu để đưa vào danh sách chọn lấy từ đâu?* Trong trường hợp này danh sách dữ liệu để chọn lấy từ bảng KHACH, nên chọn mục *I want the lookup column to look up the values in a table or query.*

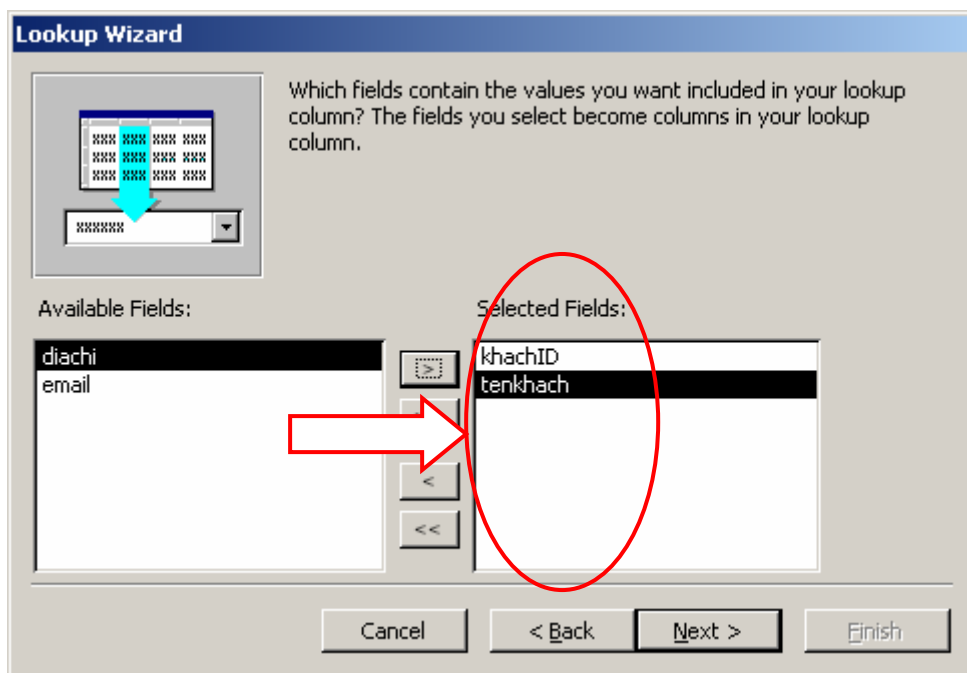
Nhấn **Next** để tiếp tục:

**Bước 3:** Chọn dữ liệu cho danh sách:



Hộp thoại trên trả lời câu hỏi: *Bảng (query) nào chứa dữ liệu cần đưa vào danh sách?*. Trong trường hợp này lấy dữ liệu từ bảng KHACH nên mục **View** chọn *Tables*; và chọn bảng *KHACH*.

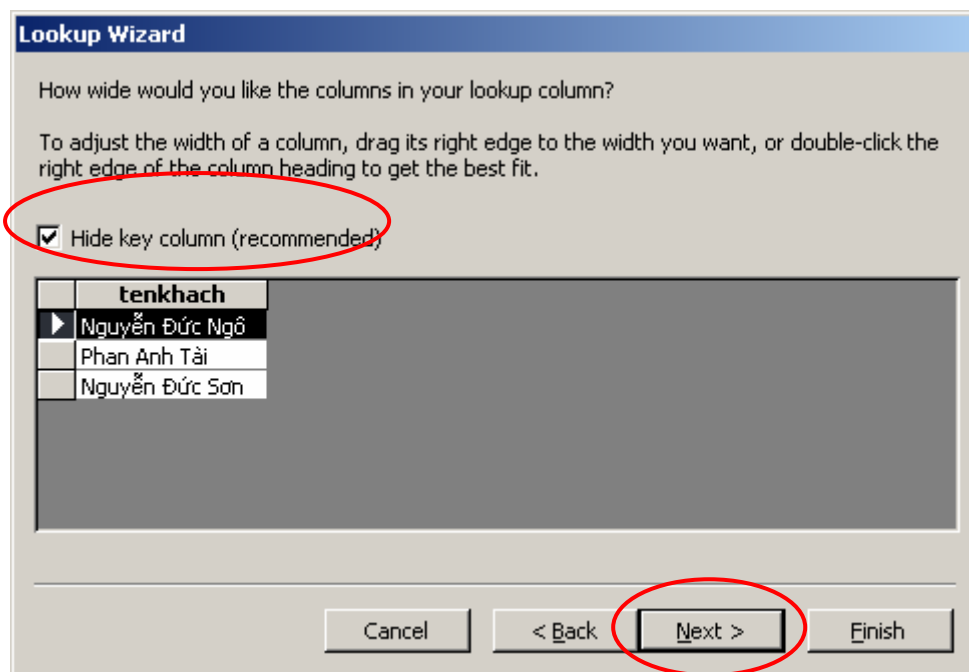
Chọn xong nhấn **Next**, hộp thoại sau xuất hiện:



Hộp thoại này trả lời câu hỏi: *Giá trị những trường nào của bảng (query) đã chọn sẽ được hiển thị trên danh sách?* Hãy dùng các nút >, >>, <, << để đưa 2 trường *khachID* và *tenkhach* từ danh sách **Available Fields:** (danh sách các trường có thể chọn) sang danh sách **Selected Fields:** (danh sách các trường đã chọn).

Lưu ý: Về nguyên tắc chỉ cần đưa trường *khachID* của bảng khác vào danh sách Selected Fields: là đủ, tuy nhiên nên đưa thêm trường *tenkhach* để thuận tiện hơn khi chọn lựa dữ liệu khi nhập sau này.

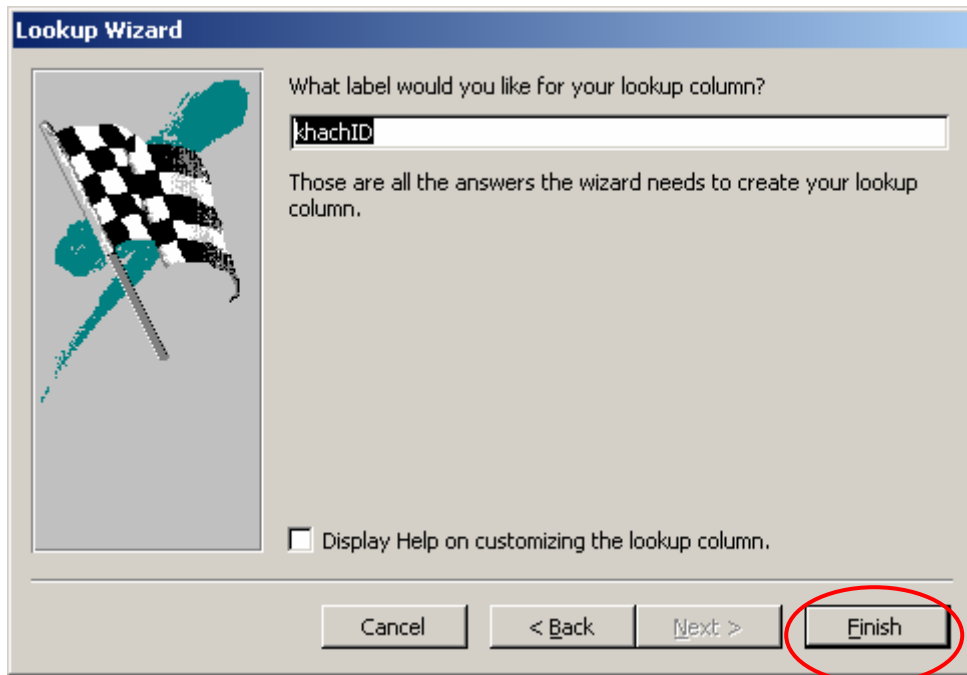
Chọn xong nhấn **Next** để tiếp tục:



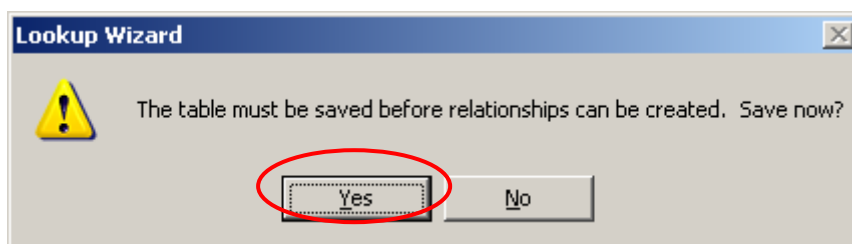
Nếu muốn ẩn mã khách khi chọn dữ liệu hãy chọn (checked) hộp

Hide key column (recommended) nếu không thì bỏ qua;

Nhấn **Next** để tiếp tục



Cuối cùng nhấn **Finish** để kết thúc tiến trình Lookup Wizard. Khi đó hộp thoại yêu cầu ghi lại cấu trúc bảng xuất hiện:



Hãy nhấn **Yes** để đồng ý.

## 6. Quy trình xây dựng CSDL Access


Đến đây chúng tôi có thể khuyến cáo một quy trình tốt để xây dựng một CSDL Access theo thiết kế sẵn có :

**BƯỚC 1** : Lần lượt xây dựng cấu trúc từng bảng dữ liệu trong CSDL. Với mỗi bảng dữ liệu khi khai báo cấu trúc cần giải quyết các công việc sau :

- Khai báo danh sách các trường của bảng ở cột *Field Name*;
- Chọn kiểu dữ liệu phù hợp cho các trường ở cột *Data Type*;
- Thiết lập trường khoá cho bảng;
- Thiết lập một số khác cần thiết cho các trường như : *Field Size*, *Format*, *Input Mark*, *Required*, *Validate Rule*, ...

- Ghi tên bảng

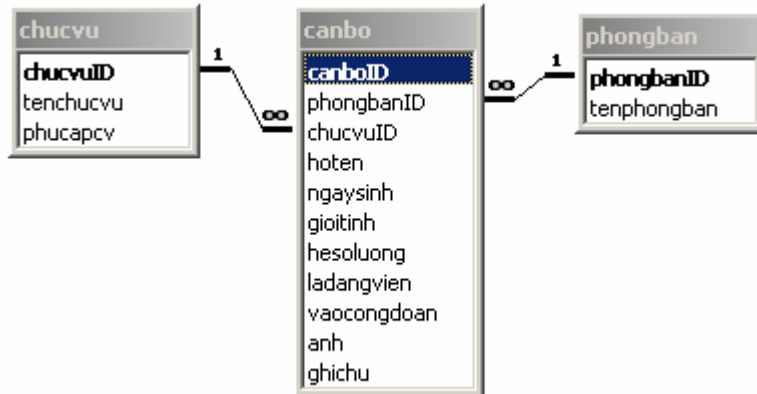
**BƯỚC 2 :** Lần lượt thiết lập thuộc tính LOOKUP cho các trường một cách phù hợp. Mỗi quan hệ trên bảng thiết kế sẽ cần một thao tác thiết lập thuộc tính LOOKUP (sử dụng trình **LookUp Wizard**) từ trường trên bảng quan hệ nhiều sang trường bảng quan hệ một;

**BƯỚC 3 :** Thiết lập các thuộc tính đảm bảo toàn vẹn dữ liệu cần thiết cho các quan hệ tại cửa sổ Relationships (menu **Tool | Relationships..** hoặc nhấn nút  trên thanh công cụ);

**BƯỚC 4 :** Thực hiện nhập dữ liệu cho các bảng nếu cần. Chú ý : bảng có quan hệ 1 phải được nhập dữ liệu trước bảng có quan hệ nhiều.

## Bài tập

1. Xây dựng CSDL Quản lý lương cán bộ một cơ quan có cấu trúc như sau :

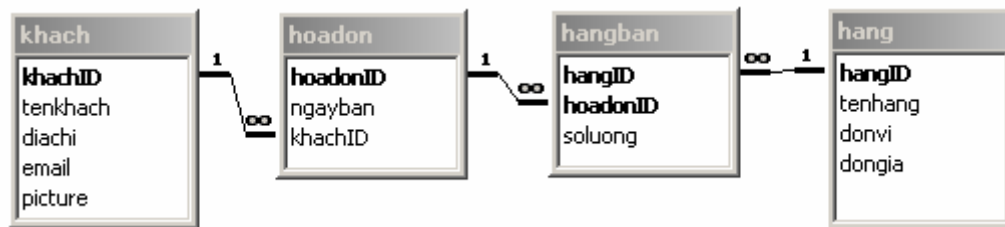


### Yêu cầu :

- Thiết kế cấu trúc các bảng một cách phù hợp : kiểu dữ liệu các trường; trường khoá; thuộc tính Lookup và các thuộc tính khác;
- Thiết lập quan hệ cùng các thuộc tính đảm bảo toàn vẹn dữ liệu cho các quan hệ;
- Nhập dữ liệu như sau :
  - + 4 phòng ban;

- + 5 loại chức vụ;
- + 20 hồ sơ cán bộ.

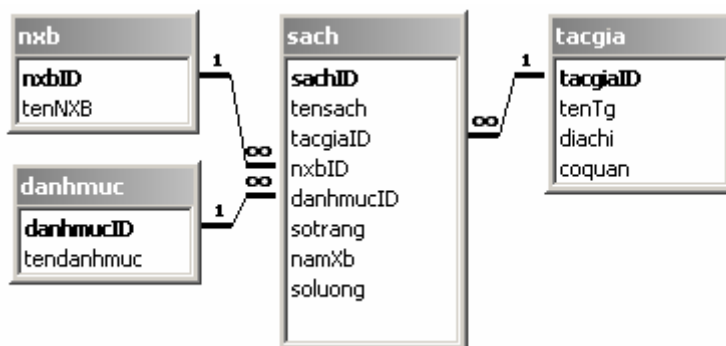
2. Xây dựng CSDL Quản lý việc bán hàng cho một cửa hàng có cấu trúc như sau :



### Yêu cầu :

- Thiết kế cấu trúc các bảng một cách phù hợp : kiểu dữ liệu các trường; trường khoá; thuộc tính Lookup và các thuộc tính khác;
- Thiết lập quan hệ cùng các thuộc tính đảm bảo toàn vẹn dữ liệu cho các quan hệ;
- Nhập dữ liệu như sau :
  - + 5 khách hàng;
  - + 15 danh mục hàng hoá có bán;
  - + lập 10 hoá đơn bán hàng;
  - + với 25 lượt hàng hoá được bán ra.

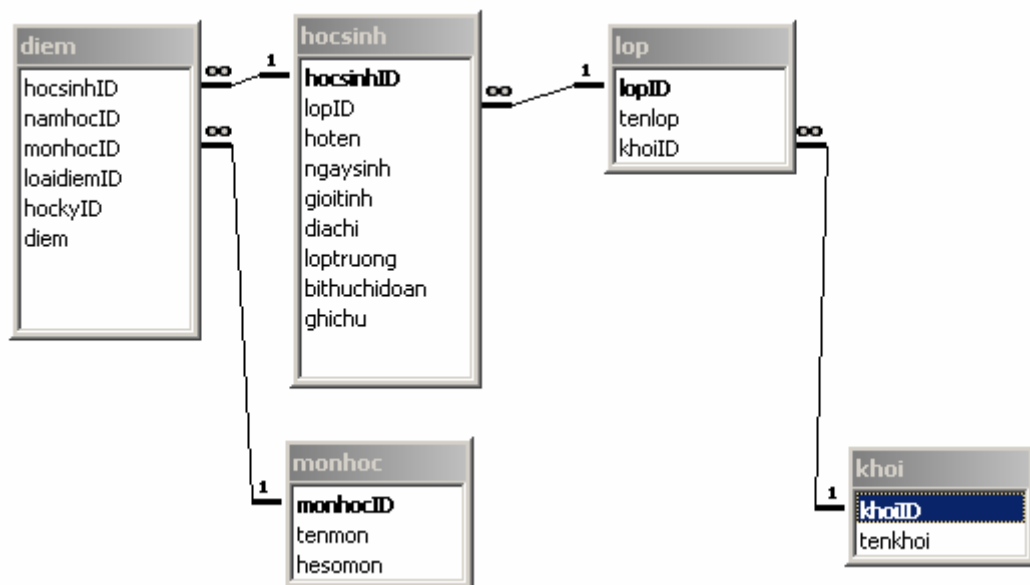
3. Xây dựng CSDL Quản lý sách một thư viện có cấu trúc như sau :



### Yêu cầu :

- Thiết kế cấu trúc các bảng một cách phù hợp : kiểu dữ liệu các trường; trường khoá; thuộc tính Lookup và các thuộc tính khác;
- Thiết lập quan hệ cùng các thuộc tính đảm bảo toàn vẹn dữ liệu cho các quan hệ;
- Nhập dữ liệu như sau :
  - + 5 tác giả;
  - + 3 nhà xuất bản;
  - + 3 danh mục sách;
  - + 20 đầu sách.

4. Xây dựng CSDL Quản lý điểm học sinh trường phổ thông có cấu trúc như sau :



#### Yêu cầu :

- Thiết kế cấu trúc các bảng một cách phù hợp: kiểu dữ liệu các trường; trường khoá; thuộc tính Lookup và các thuộc tính khác;
- Thiết lập quan hệ cùng các thuộc tính đảm bảo toàn vẹn dữ liệu cho các quan hệ;
- Nhập dữ liệu như sau:

- + 10 môn học phổ thông;
- + 3 khối học;
- + 15 lớp chia đều cho 3 khối;
- + 30 học sinh;
- + Và nhập điểm 2 học kỳ cho học sinh của một lớp nào đó.



# CHƯƠNG 2

## TRUY VẤN DỮ LIỆU

Trong chương trước đã giới thiệu những khái niệm cũng như các kỹ năng, trình tự cần thiết để có thể xây dựng tốt một CSDL trên Access. Chương này sẽ cung cấp những khái niệm cũng như những kỹ năng cần thiết để xử lý dữ liệu khi cần. Một trong những công cụ xử lý dữ liệu trực quan, hữu hiệu trên Access là **Query**.

Có rất nhiều dạng yêu cầu xử lý dữ liệu như: trích - lọc - hiển thị dữ liệu; tổng hợp - thống kê; thêm - bớt - cập nhật dữ liệu; ... Vì vậy sẽ tồn tại một số loại Query tương ứng để giải quyết các yêu cầu xử lý dữ liệu trên. Có 7 loại query trong Access :

- Select Query;
- Total Query;
- Crosstab Query;
- Append Query;
- Delete Query;
- Update Query;
- Make table Query.

Bản chất của Query là các câu lệnh SQL (Structured Queries Language- ngôn ngữ truy vấn dữ liệu có cấu trúc)- một ngôn ngữ truy vấn dữ liệu được dùng khá phổ biến trên hầu hết các hệ quản trị CSDL hiện nay. Việc thiết kế một query là gián tiếp tạo ra một câu lệnh xử lý dữ liệu SQL. Việc thi hành query chính là việc thi hành câu lệnh SQL đã tạo ra.

## 1. SELECT queries

### 1.1 Cách tạo

Select query là loại truy vấn dùng trích - lọc - kết xuất dữ liệu từ nhiều nguồn khác nhau từ CSDL ra một bảng kết quả. Ví dụ:

- Đưa ra thông tin chi tiết bảng lương tháng 8;
- Đưa ra danh sách cán bộ là Đảng viên;
- Đưa ra thông tin chi tiết về các hoá đơn bán ra trong ngày hôm nay;
- Đưa ra doanh thu bán hàng của một tháng nào đó;
- ...

Tất cả những yêu cầu dạng như vậy (đưa ra một danh sách kết quả) đều có thể sử dụng **SELECT query** để đáp ứng. Mỗi yêu cầu xử lý dữ liệu cần phải tạo ra một Select query đáp ứng; mỗi query sẽ có một tên gọi (như cách đặt tên bảng dữ liệu); query sau khi đã tạo ra đều có thể chỉnh sửa lại được cấu trúc cũng như nội dung; qui trình để đáp ứng mỗi yêu cầu trên như sau: tạo một query đáp ứng được các yêu cầu → thi hành query đã tạo để thu nhận kết quả.

Tiếp theo minh hoạ qui trình các bước để tạo một select query đáp ứng yêu cầu đơn giản nhất:

Từ CSDL Quản lý lương cán bộ, hãy đưa ra bảng lương cán bộ với những thông tin sau: *canboID*, *hoten*, *ngaysinh*, *tencv*, *luongchinh*, *phucapcv*, *thuclinh*.

Trong đó: **luongchinh = hesoluong \* 290000**

**Thuclinh = luongchinh + phucapcv**

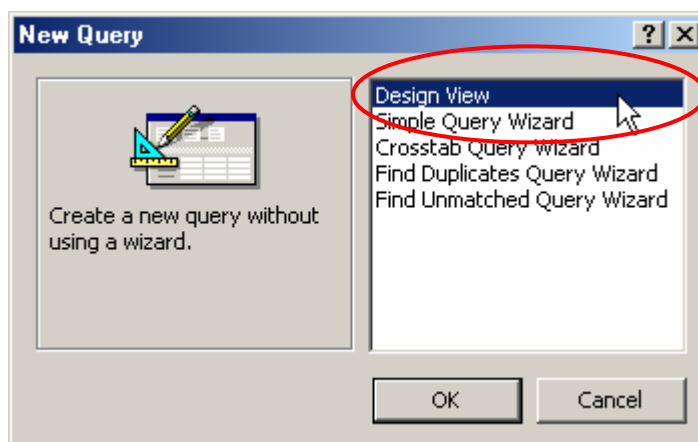
**Phân tích yêu cầu:**

(1). Thông tin yêu cầu đòi hỏi từ nhiều bảng khác nhau (bảng CANBO với các cột: canboID, hoten, ngaysinh; bảng CHUCVU với các cột: ten cv, phucapcv);

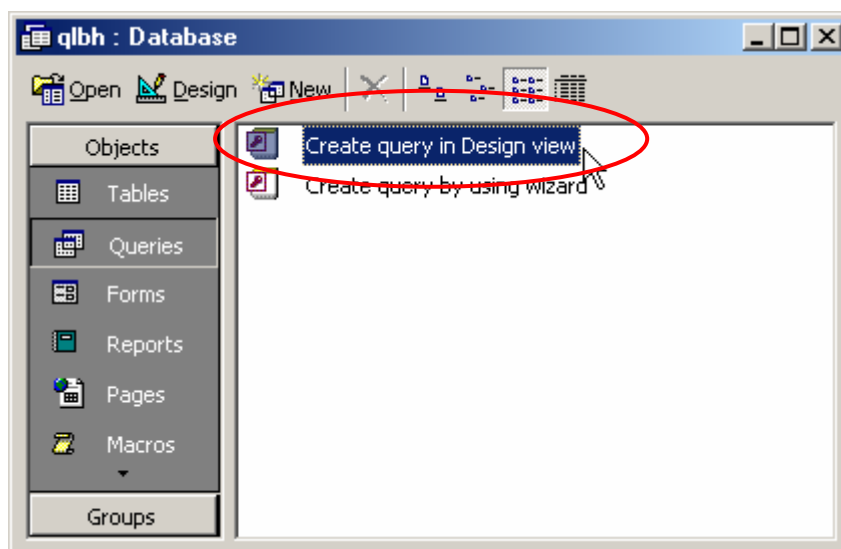
(2). Có những cột thông tin đã có sẵn trên CSDL (5 cột kể trên) nhưng một số cột yêu cầu phải được tính bởi biểu thức: cột luongchinh và cột thuclinh.


### Hướng dẫn cách làm:

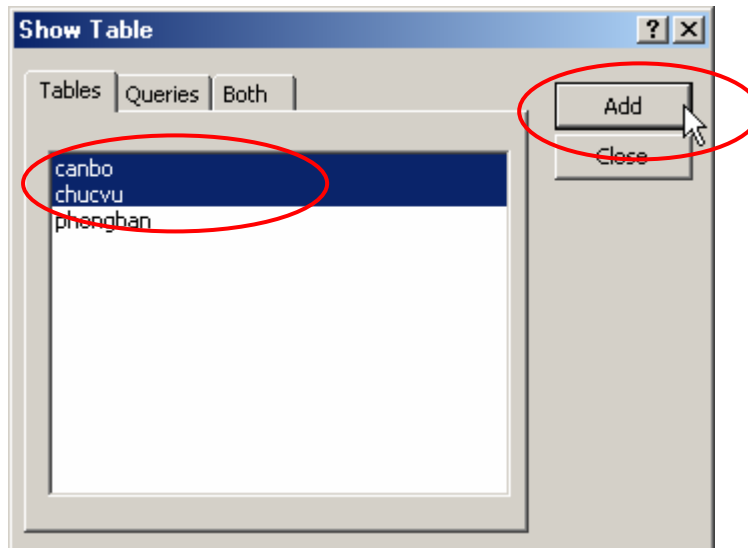
**Bước 1:** Ra lệnh tạo một Select query mới bằng cách: kích hoạt thẻ Queries, nhấn nút **New**, chọn **Design View**, nhấn **Ok** (hình dưới):



Hoặc nhấn đúp chuột lên biểu tượng **Create query in Design view** trên màn hình:

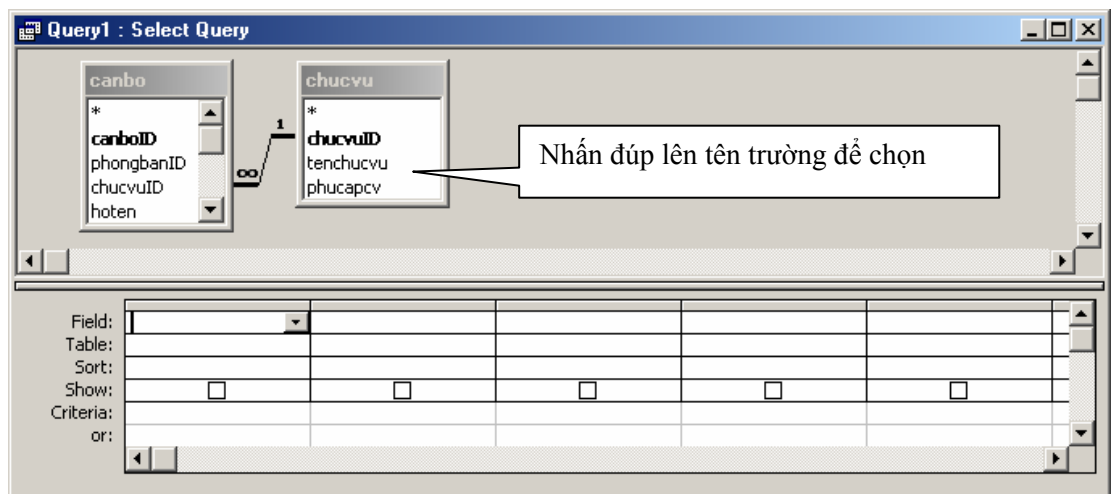


**Bước 2:** Chọn những bảng có chứa dữ liệu liên quan lên màn hình thiết kế query từ cửa sổ **Show Table** (không thấy cửa sổ này nhấn nút Show table  trên thanh công cụ):



Hãy chọn các bảng có liên quan (bảng CANBO và bảng CHUCVU); chọn xong nhấn **Add** và nhấn **Close**. Khi đó màn hình thiết kế query xuất hiện:

**Bước 3:** Khai báo những thông tin cần thiết cho query:

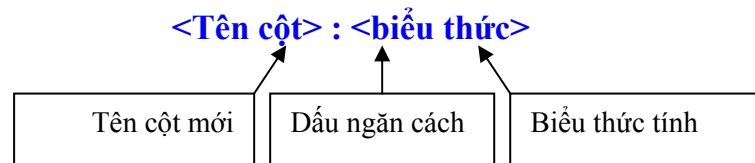


Dòng **Field:** là nơi khai báo danh sách các thông tin (cột dữ liệu) của bảng kết quả.

Có 2 loại thông tin bài toán yêu cầu: thông tin có sẵn từ các trường trên CSDL như *canboID*, *hoten*, *tencv*, *phucapcv* và thông tin phải được tính theo một biểu thức nào đó như: *luongchinh* và *thuclinh*.

Muốn hiển thị trường (field) nào lên query, chỉ việc nhấn đúp chuột lên tên trường đó hoặc dùng chuột kéo tên chúng từ các bảng lên dòng Field. Hãy dùng phương pháp này để đưa 4 trường *canboID*, *hoten*, *ngaysinh* và *phucapcv* lên dòng *Field*.

Với 2 cột *luongchinh* và *thuclinh* phải đưa vào query bằng cách tạo một cột mới với một tên gọi <tên cột> được tính bằng một <biểu thức> trên dòng **Field** theo cú pháp như sau:

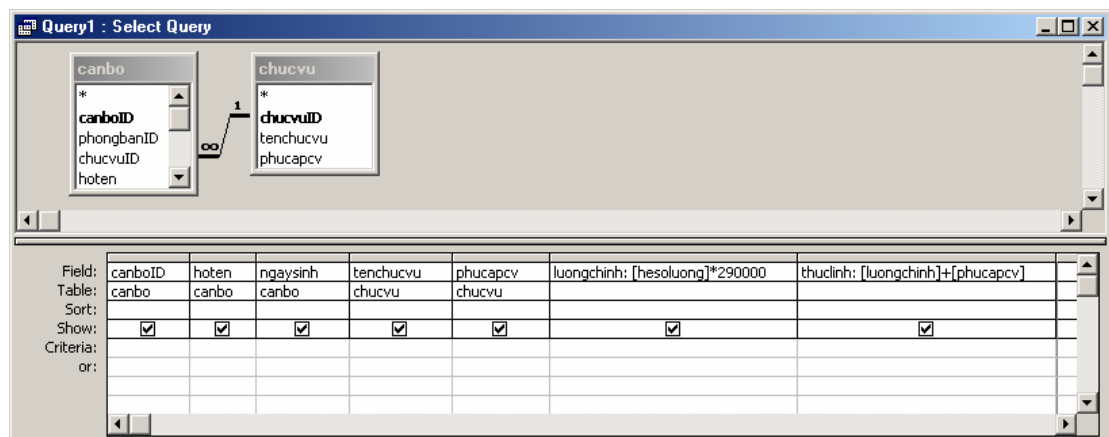


Ví dụ:

**Luongchinh : hesoluong \* 290000** để tạo thêm cột *luongchinh* mới



**Thuclinh : luongchinh + phucapcv** để tạo thêm cột *thuclinh* mới.

Cuối cùng, màn hình thiết kế query như sau:



Có thể ghi query lại với một tên gọi khi ra lệnh cắt **Alt + S**

**Bước 4:** Sử dụng query vừa tạo. Một query sau khi đã tạo xong có thể:

- (1) Kích hoạt chúng để lấy kết quả bằng cách: nhấn đúp chuột lên tên query. Nếu đang trong chế độ thiết kế (Design view), nhấn nút View  trên thanh công cụ hoặc nhấn phải chuột lên Query đang thiết kế chọn  **Datasheet View**. Bạn sẽ xem được kết quả trả về của query và có thể thao tác bảng dữ liệu này như trên một Table (xem mục 4.2, Phần 1- Tạo CSDL):

canboID	hoten	ngaysinh	tenchucvu	phucapcv	luongchinh	thuclinh
1	Nguyễn Đức Ngô	12/04/1976	Giám đốc	500000	904800	1404800
4	Nguyễn Thu Loan	05/02/1978	Nhân viên	200000	516200	716200
5	Hà Anh Tú	31/12/1975	Nhân viên	200000	516200	716200
6	Phan Quang Sáng	03/09/1976	Nhân viên	200000	696000	896000
3	Nguyễn Hải Hà	13/09/1979	Phó giám đốc	400000	611900	1011900
2	Phan Anh Quân	23/07/1978	Trưởng phòng	300000	611900	911900
7	Nguyễn Ánh Tuyết	02/07/1980	Trưởng phòng	300000	696000	996000

(2) Sửa lại được cấu trúc thiết kế query khi chọn chúng và nhấn nút **Design**.



Bản chất của một SELECT query là câu lệnh SQL có dạng:-----

**SELECT ... FROM ...**

### Một số thiết lập khác cho Query

Dòng **Sort**: để thiết lập thứ tự sắp xếp dữ liệu trên Query. Muốn sắp xếp dữ liệu cho trường nào, thiết lập thuộc tính Sort cho trường ấy. Có 2 giá trị cho thuộc tính Sort: *Ascending* - sắp xếp tăng dần và *Descending* - sắp xếp giảm dần. Trường nào đứng trước sẽ được thứ tự sắp xếp trước.

Field:	canboID	hoten	ngaysinh
Table:	canbo	canbo	canbo
Sort:	Ascending		Descending
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			



Bản chất của yêu cầu sắp xếp dữ liệu thể hiện ở mệnh đề:-----

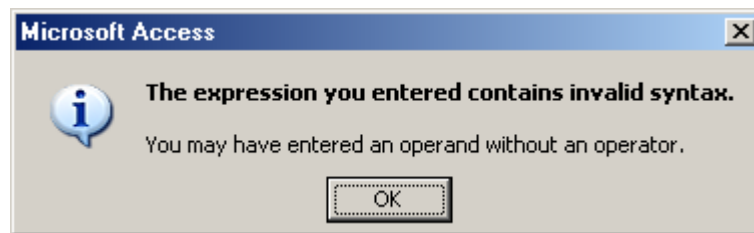
**... ORDER BY ... [ASC] [DESC]** trong câu lệnh SQL

Dòng **Show**: để chỉ định hiển thị hay không hiển thị dữ liệu trường đó ra bảng kết quả. Nếu chọn (checked)- dữ liệu sẽ được hiển thị ra bảng kết quả.

Field:	canboID	hoten	ngaysinh
Table:	canbo	canbo	canbo
Sort:	Descending		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:			

**Chú ý:** Có 2 lỗi hay mắc phải:

- Lỗi gõ sai biểu thức:

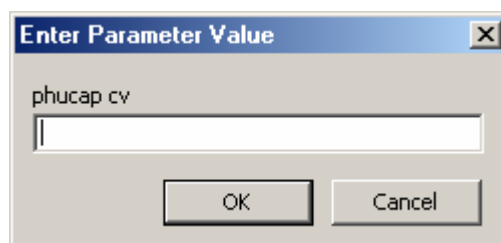


Nguyên nhân gây ra lỗi này là rất nhiều, có thể liệt kê ra đây một số tình huống:

**Tình huống 1:** Có thể một trong số tên các trường bạn gõ trong biểu thức có chứa dấu cách hoặc các ký tự đặc biệt. Ví dụ: trường *hesoluong* của bạn là he so luong thì phải gõ như sau: *[he so luong]* – thêm dấu móc vuông vào 2 đầu của tên trường;

**Tình huống 2:** Sai ký pháp lô gíc của biểu thức: do viết thừa hoặc thiếu các toán tử, toán hạng, hoặc các dấu mở đóng ngoặc không khớp..;

- Gõ không đúng tên trường trong biểu thức:

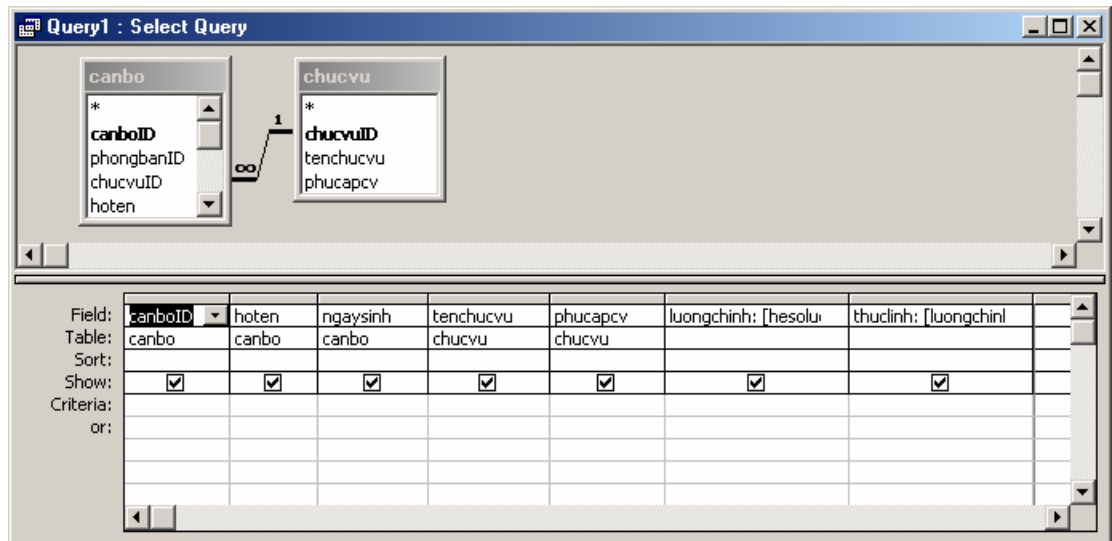


Lỗi này xảy ra khi bạn đã gõ sai tên trường. Tên hiển thị trên hộp thoại (*phucap cv*) máy tính không hiểu, có thể tên đúng của trường này là *phucapcv*. Bạn phải lần tới biểu thức có chứa tên trên và kiểm tra sửa cho đúng với tên trường có trong CSDL. Lỗi này hay xảy ra khi thiết kế bảng đặt tên các trường có chứa dấu cách!

## 1.2 Lọc dữ liệu

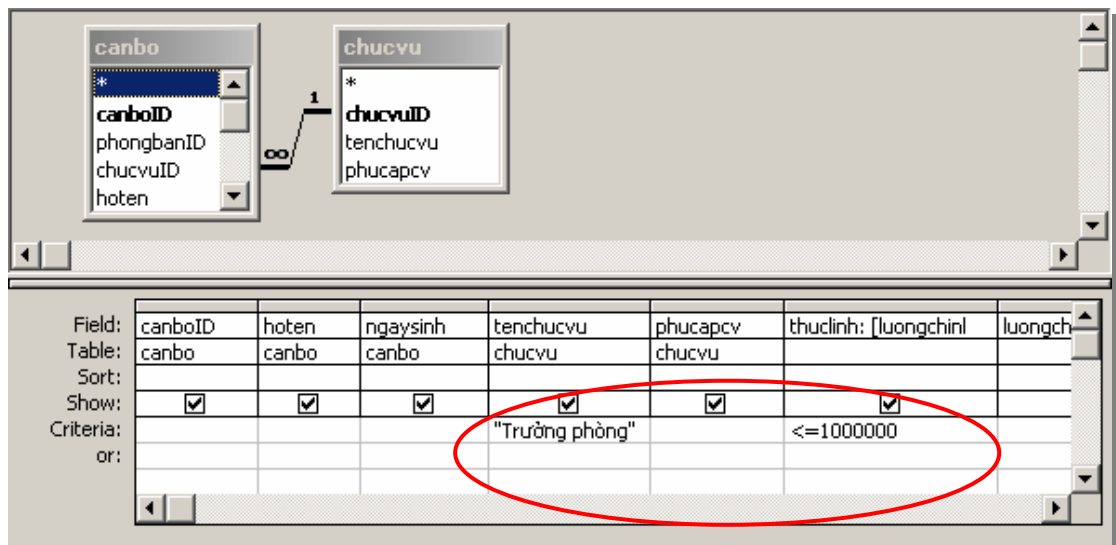
Khác với bảng (Tables), Queries cung cấp một khả năng lọc dữ liệu khá hoàn chỉnh; Có thể lọc ra những dữ liệu theo những điều kiện phức tạp hơn, đặc biệt có thể chấp nhận những giá trị lọc là các tham biến.

Để lọc dữ liệu, bạn phải thiết lập điều kiện đặt lọc lên vùng **Criteria** của queries (trong chế độ đang thiết kế).



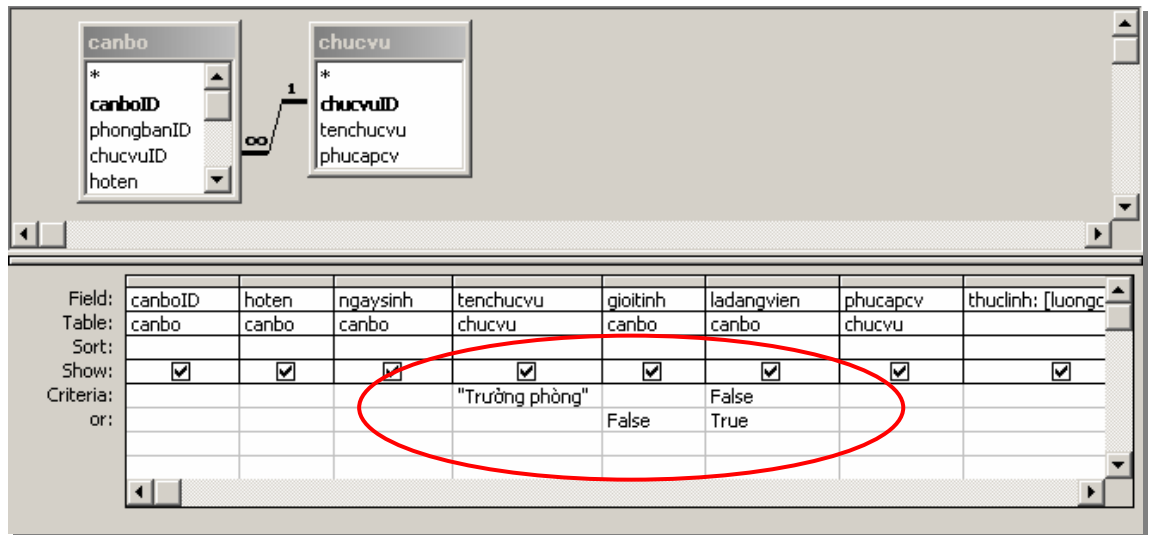
Các điều kiện nằm trên cùng một dòng Criteria sẽ được nối với nhau bởi toán tử **AND** (và); mỗi dòng Criteria sẽ được nối với nhau bởi toán tử **OR** (hoặc). Xét các ví dụ sau thực hiện lọc trên query bảng lương vừa được tạo ra:

**Ví dụ 1:** Lọc ra những cán bộ là *trưởng phòng* có *thực lĩnh*  $\leq 1,000,000$  đ



**Ví dụ 2:** Lọc ra những cán bộ là *Trưởng phòng* mà không phải là *Đảng viên* hoặc những cán bộ *Nữ* là *Đảng viên*:





**Ví dụ 3:** Lọc ra những cán bộ có tên là *Nam*. Chú ý: Tên chỉ là một phần của trường *Hoten*:

canboID	hoten	ngaysinh	tenchucvu
canbo	canbo	canbo	chucvu
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Like '*Nam'		

Toán tử **LIKE** để biểu diễn những giá trị mang tính tương đối (có thể chỉ giống một phần giá trị thực). Cú pháp biểu diễn toán tử này như sau:

**Like '<giá trị tương đối>'**

Trong <giá trị tương đối> có thể chứa hằng (*Nam*) và các ký tự đại diện. Có 2 ký tự đại diện là:

- Ký tự \* để biểu diễn bất kỳ giá trị nào;
- Ký tự ? để biểu diễn một ký tự bất kỳ.

Một số ví dụ minh họa toán tử LIKE:

- **Like 'Nguyễn\*'** - lọc ra những người họ Nguyễn. 6 ký tự đầu là Nguyễn, các ký tự còn lại là thoải mái;
- **Like '\*Đức\*'** - lọc ra những người có Họ hoặc Đệm hoặc Tên là Đức;
- **Like '\*/\*/1980'** - lọc ra những người sinh năm 1980;
- **Like '\*/11/\*'** - lọc ra những người sinh tháng 11;

- **Like '\*/\*/198?'** - lọc ra những người sinh từ năm 1980 đến năm 1989;
- ...

**Ví dụ 4:** Lọc ra những cán bộ có  $2,000,000 \geq Thuclinh \geq 1,000,000$ :

luongchinh: [hes:	thuclinh: [luongchinh]+[phucapcv]
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Between 1000000 And 2000000	

Toán tử **BETWEEN** để lọc ra các giá trị nằm trong một khoảng nào đó. Thường giải quyết các yêu cầu *lớn hơn hoặc bằng bao nhiêu và nhỏ hơn hoặc bằng bao nhiêu?*

Cú pháp toán tử Between như sau:

**Between <giá trị nhỏ> And <giá trị lớn>**

Cả 4 ví dụ trên đều đặt lọc theo các tham trị (lọc cứng). Trong trường hợp sử dụng các tham biến vào điều kiện lọc, xin tiếp tục theo dõi hướng dẫn dưới đây:

Muốn thiết lập giá trị đặt lọc nào là tham biến bạn làm như sau:

*[Gõ một lời nhắc trong ngoặc vuông]* tại đúng vị trí tham số cần thiết lập.

Ví dụ: Lọc ra danh sách các cán bộ phòng ban nào đó, làm như sau:

hoten	tenphongban	ngaysinh
canbo	phongban	canbo
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Gõ tên phòng ban cần xem]		

Khi thi hành query, một hộp thoại yêu cầu gõ vào tham số cần lọc:

Hãy gõ vào Tên phòng ban cần xem. Việc thiết lập tham số kiểu này không giới hạn trên một query.



Bản chất của yêu cầu lọc dữ liệu thể hiện ở mệnh đề:-----

... **WHERE** ... trong câu lệnh SQL

## 2. TOTAL queries

Tổng hợp – là phép xử lý dữ liệu khá phổ biến. Trong Access, query là một trong những công cụ xử lý khá tốt việc này. **Total query** là một điển hình.

Hãy xét một số yêu cầu tổng hợp dữ liệu như sau:

tenphongban	TSCB	Tong LC	tong PCCV	Tong Thuclinh
Kinh doanh	3	2212700	1000000	3212700
Tổng hợp	4	2340300	1100000	3440300

Record: 1 of 2

tenchucvu	tscb
Giám đốc	1
Nhân viên	3
Phó giám đốc	1
Trưởng phòng	2

Record: 1 of 4

hoadonID	ngayban	tenkhach	tongtien
1	12/10/2004	Nguyễn Đức Ngô	893
2	24/10/2004	Nguyễn Đức Ngô	114
3	10/10/2004	Phan Anh Tài	855
4	02/11/2004	Nguyễn Đức Sơn	1394

Record: 3 of 4

	tenhang	ngayban	Tong so
	RAM 256	10/10/2004	12
	CPU P4 2.4	10/10/2004	3
	Hard disk	12/10/2004	9
	Floppy disk	12/10/2004	3
	CPU P4 2.4	12/10/2004	2
	RAM 256	24/10/2004	3
	Floppy disk	02/11/2004	8
	CPU P4 2.4	02/11/2004	10

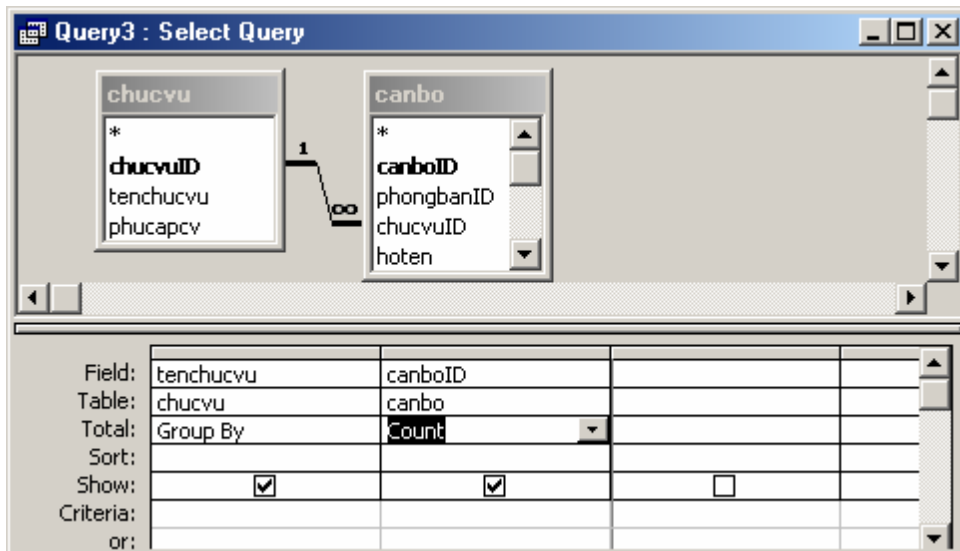
Tất cả những yêu cầu trên đều có thể được đáp ứng bằng các **TOTAL query**. Dưới đây sẽ hướng dẫn cách tạo **Total query**. Ví dụ: tạo query đưa ra bảng tổng hợp *Tenchucvu*, *Tổng số CB* (query thứ 2);

**BƯỚC 1:** Tạo một **select query** với đầy đủ các thông tin liên quan bảng tổng hợp:

Field:	tenchucvu	canboID		
Table:	chucvu	canbo		
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Criteria:				
or:				

**BƯỚC 2:** Tùy chọn **Total query** bằng cách mở thực đơn **View | Total** hoặc nhấn nút **Total**  $\Sigma$  trên thanh công cụ;

**BƯỚC 3:** Thiết lập các tùy chọn **Total** cho các trường một cách phù hợp như sau:



- Vì bài toán yêu cầu tổng hợp thông tin theo từng loại chức vụ, nên **Total** của trường *TENCHUCVU* là **Group By**;
- Trường *CANBOID* dùng để đếm số cán bộ từng chức vụ, nên chọn phép tổng hợp là **Count**.

Đến đây đã tạo xong query và có thể thi hành để thu nhận kết quả.

Ngoài phép Count để đếm, Total query còn cung cấp một số phép tổng hợp khác như:

- Sum** - tính tổng;
- AVG** - tính trung bình cộng;
- Max** - xác định giá trị lớn nhất;
- Min** - xác định giá trị nhỏ nhất



Bản chất của Total query thể hiện ở câu lệnh SQL:-----

**SELECT .. FROM .. GROUP BY ...**

-----

### 3. CROSSTAB queries

Ở mục 2 chúng ta đã được tìm hiểu về **Total query** trong việc đáp ứng các yêu cầu tổng hợp dữ liệu. Trên thực tế còn rất nhiều các yêu cầu tổng hợp khác mà **Total query** không thể đáp ứng được. Nhiều trong số đó như là:

	tenphongban	Giám đốc	Trưởng phòng	Phó giám đốc	Nhân viên
▶	Kinh doanh	1	1		3
	Tổng hợp		1	1	4

Record: 1 of 2

	tenphongban	Ts Nam	Ts Nu
▶	Kinh doanh	1	4
	Tổng hợp	5	1

Record: 1 of 2

	tenhang	Tongsoban	10/10/2004	12/10/2004	24/10/2004
▶	CPU P4 2.4	3	1	2	
	Floppy disk				
	Hard disk				
	RAM 256				1

Record: 1 of 4

Yêu cầu tổng hợp dữ liệu là các tiêu đề dòng và cột!

Tất cả những yêu cầu dạng trên có thể sử dụng **CROSSTAB query** để đáp ứng. Trước khi tìm hiểu cách tạo **Crosstab query**, chúng ta hãy tìm hiểu cấu trúc một **Crosstab**, được minh họa bởi hình sau:

Row heading	Column heading
	Value

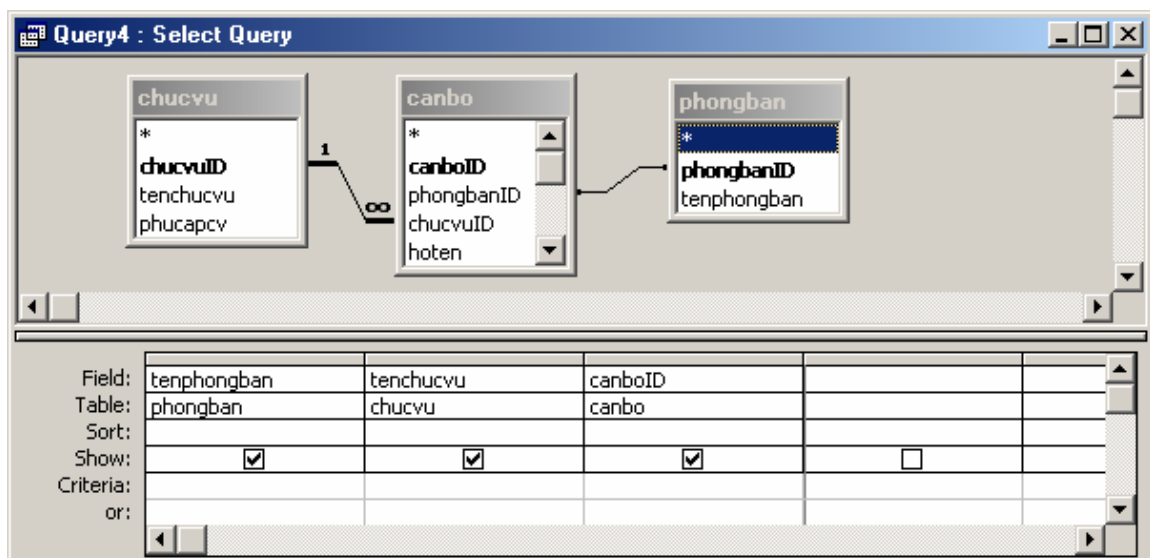
- **Row heading** là tiêu đề các dòng, có chứa các giá trị của các trường nào đó làm tiêu chí thống kê. Mỗi **Crosstab** phải có tối thiểu 1 trường làm Row heading;
- **Column heading** là tiêu đề các cột, có chứa các giá trị của một trường nào đó làm tiêu chí thống kê. Mỗi **Crosstab** chỉ có duy nhất 01 trường làm Column heading;
- **Value** là vùng dữ liệu tổng hợp (là các con số). Chỉ có duy nhất một trường làm **Value**, tương ứng với nó là một phép tổng hợp hoặc: đếm, tính tổng, tính trung bình cộng, max, min,...

Các bước để tạo một **Crosstab query**. Ví dụ tạo query đưa ra được bảng tổng hợp sau:

	tenphongban	Giám đốc	Trưởng phòng	Phó giám đốc	Nhân viên
Kinh doanh		1	1		3
Tổng hợp			1	1	4

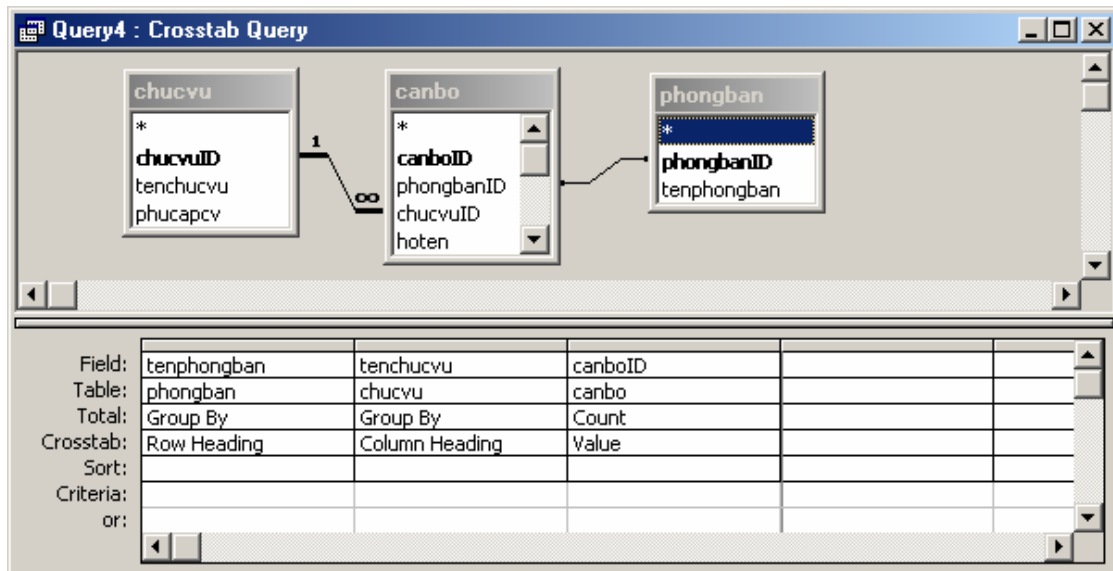
Record: 1 of 2

**Bước 1:** Tạo một **select query** với đầy đủ các trường có liên quan đến phép tổng hợp (không thừa và cũng không thể thiếu một trường nào) như sau:



**Bước 2:** Ra lệnh chọn kiểu **Crosstab query** bằng cách mở thực đơn **Queries | Crosstab Query**;

**Bước 3:** Thiết lập các thuộc tính **Total** cũng như **Crosstab** cho các trường phù hợp như sau:



Cụ thể:

Tên trường	Thuộc tính Total	Thuộc tính Crosstab
Tenphongban	<i>Group By</i>	<i>Row Heading</i>
Tenchucvu	<i>Group By</i>	<i>Column Heading</i>
canboID	<i>Count</i>	<i>Value</i>



Bản chất của Crosstab query thể hiện ở câu lệnh SQL:-----

**TRANSFORM .. SELECT .. FROM .. GROUP BY ... PIVOT ...**

#### 4. MAKE TABLE queries

Select và Total query luôn đưa ra một bảng kết quả- đó là giá trị tức thời, mới nhất được lấy ra từ CSDL tại thời điểm đó. Tại thời điểm khác khi thi hành query đó, rất có thể chúng ta không thu lại được bảng kết quả như thời điểm trước đó. Muốn lưu lại bảng kết quả của một query tại một thời điểm nào đó, trong Access có một cách là đưa dữ liệu kết quả query ra một bảng (Table) để lưu trữ lâu dài bởi một MAKE TABLE query.



Các bước tạo một **Make table** query để lưu trữ kết quả một query ra một bảng dữ liệu được tiến hành theo 2 bước chính:

**Bước 1:** Tạo một **Select query** hoặc một **Total query** đưa ra được bảng kết quả cần lưu trữ;

**Bước 2:** Ra lệnh tạo **Make table** query từ query đang thiết kế và thi hành để thu được bảng kết quả như mong muốn.

Giả sử đã tạo được một Select query đưa ra được bảng kết quả như sau:

hoten	ngaysinh	tenchucvu	tenphongban	luongchin	congdoanph	dangphi	thuclinh
Trần Văn Ngô		Nhân viên	Tổng hợp	1740000	0	20000	1920000
Nguyễn Đức Ngô	12/04/1976	Giám đốc	Kinh doanh	904800	45240	20000	1339560
Phan Anh Quân	23/07/1978	Trưởng phòng	Kinh doanh	611900	30595	20000	861305
Nguyễn Hải Hà	13/09/1979	Phó giám đốc	Tổng hợp	611900	30595	0	981305
Nguyễn Thu Loan	05/02/1978	Nhân viên	Tổng hợp	516200	25810	20000	670390
Hà Anh Tú	31/12/1975	Nhân viên	Tổng hợp	516200	0	0	716200
Phan Quang Sáng	03/09/1976	Nhân viên	Kinh doanh	696000	34800	0	861200
Nguyễn Ánh Tuyết	02/07/1980	Trưởng phòng	Tổng hợp	696000	34800	20000	941200
Nguyễn Văn Hoà		Nhân viên	Kinh doanh	580000	29000	0	751000
Phan Hoài Linh		Nhân viên	Kinh doanh	580000	29000	0	751000
Tạ Quang Luận		Nhân viên	Tổng hợp	1160000	58000	20000	1282000
Mai Vũ Hằng		Nhân viên	Tổng hợp	870000	0	0	1070000

Bài toán đặt ra là: đưa toàn bộ kết quả của query đang hiển thị ra một bảng mới có tên **bangluong**. Cách làm như sau:


**Bước 1:** mở select query đã tạo được ở chế độ **Design View**;

**Bước 2:** ra lệnh tạo Make table query bằng cách mở thực đơn **Queries | Make Table query**, hộp thoại Make table xuất hiện:

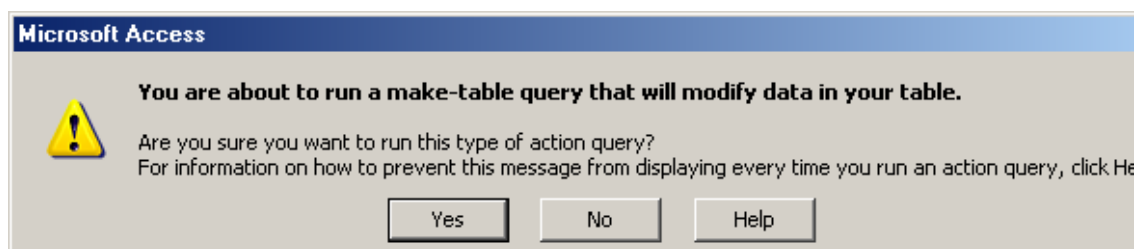
Hãy nhập vào tên bảng dữ liệu cần lưu vào ô **Table Name**: Chú ý:

- Nếu tên bảng nhập vào là mới, **Access** sẽ tự động tạo một bảng mới và sao chép toàn bộ dữ liệu mà query kết xuất được ra bảng này;
- Nếu tên nhập vào trùng một tên bảng đã có sẵn, khi thi hành **Access** sẽ xoá bảng cũ và điền vào dữ liệu mới (cần cân nhắc khi đặt tên bảng trùng tên bảng đã tồn tại).

**Bước 3:** thi hành query để nhận kết quả bằng cách:

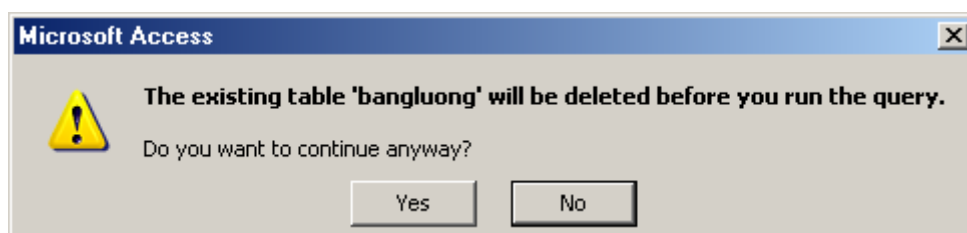
- Nếu query đang ở chế độ thiết kế, nhấn nút Run  trên thanh công cụ;
- Hoặc nháy đúp chuột lên query cần thực hiện.

Khi đó một hộp thoại cảnh báo việc bạn đang ra lệnh thi hành một query có thể làm thay đổi đến dữ liệu trên CSDL:



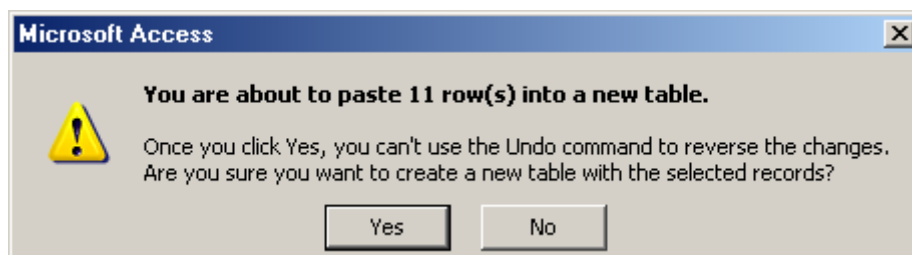
Nhấn **Yes** để tiếp tục hoặc nhấn **No** để huỷ bỏ lệnh.

Nếu trên CSDL đã tồn tại một bảng có cùng tên bảng bạn đã chỉ định cho query này lưu dữ liệu, Access sẽ hỏi bạn: *Đã tồn tại bảng xxx trên CSDL rồi, nó sẽ bị xoá sạch khi query này thi hành, bạn có muốn tiếp tục không?*



Nhấn **Yes** để tiếp tục (tất nhiên bảng dữ liệu đó sẽ bị xoá và thay vào nội dung mới); nhấn **No** để huỷ bỏ.

Cuối cùng một hộp thoại hỏi một lần cuối xem *bạn có đồng ý dán xxx bản ghi vào bảng đã chỉ định hay không?*



Nhấn **Yes** để đồng ý, **No** để huỷ bỏ.

Thi hành xong hãy mở bảng vừa tạo được để kiểm tra kết quả.



Bản chất của Crosstab query thể hiện ở câu lệnh SQL:-----

**SELECT .. INTO .. FROM..**

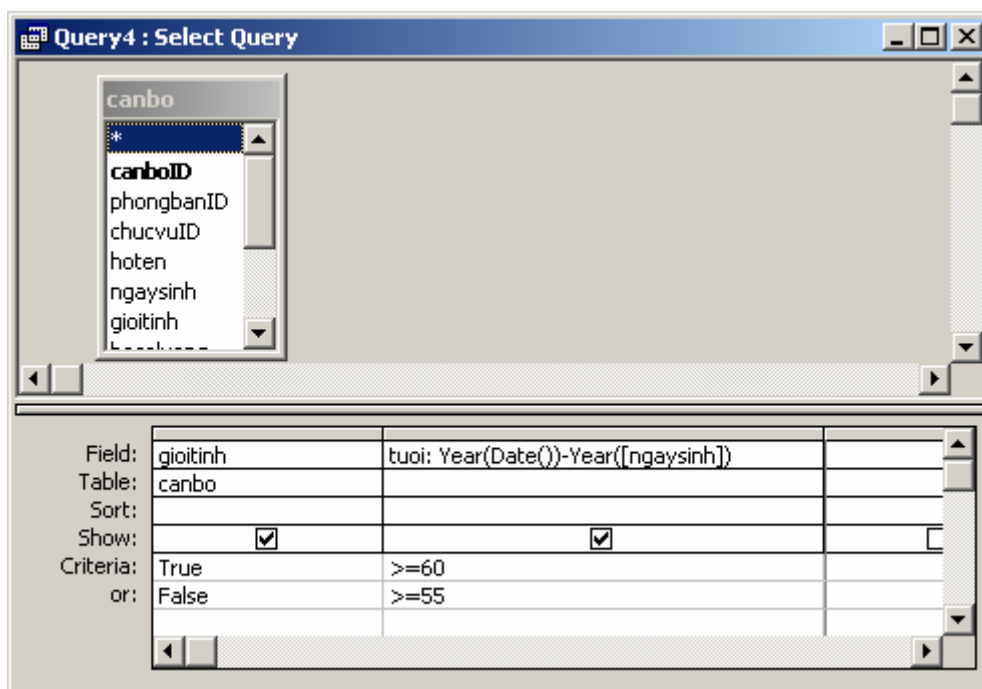
-----

## 5. DELETE queries

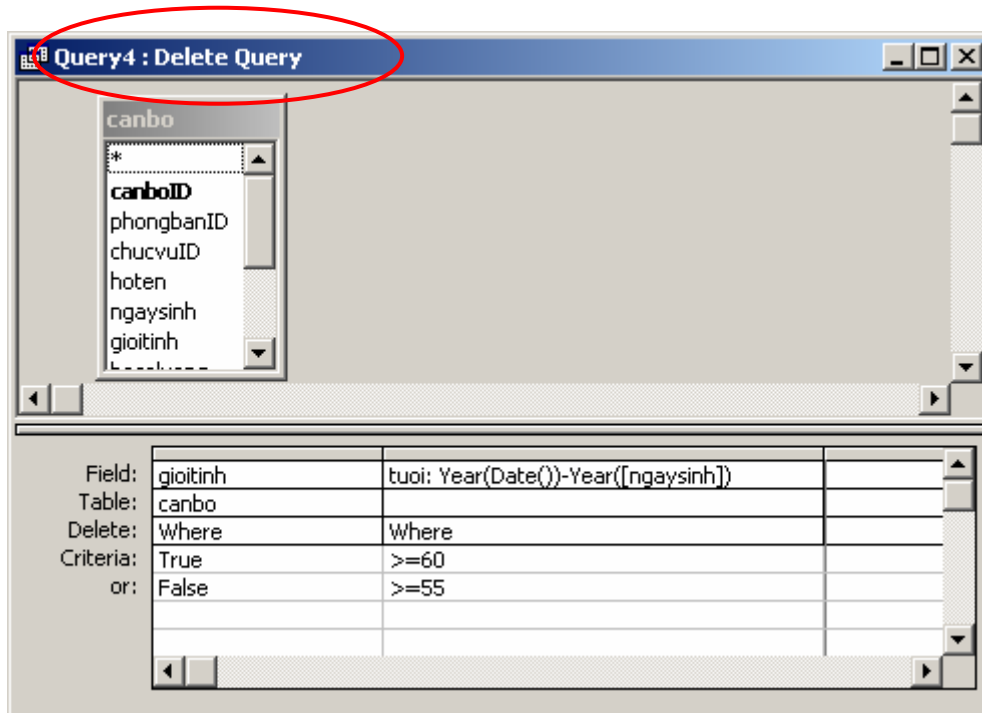
**Delete Query** là một loại **Action Query** (query hành động). Nó có thể gây thay đổi dữ liệu trong CSDL. Trong trường hợp này, Delete query dùng để xoá các bản ghi từ CSDL thoả mãn những điều kiện nào đó.


Dưới đây là hướng dẫn tạo một Delete query xoá đi những cán bộ đến tuổi nghỉ hưu ra khỏi CSDL Quản lý lương cán bộ. Khi query này thi hành, danh sách các cán bộ đến tuổi hưu sẽ bị xoá khỏi bảng *canbo*. Cách tạo query này như sau:

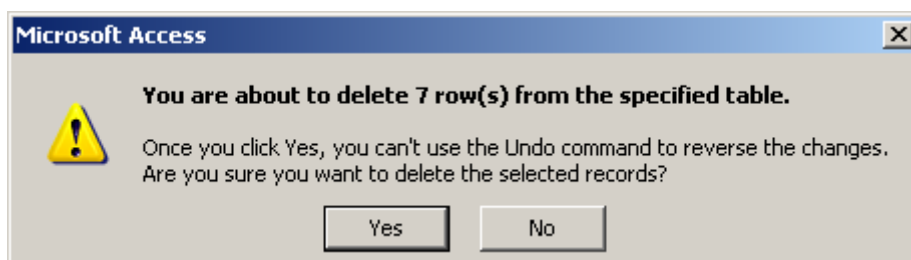
**Bước 1:** Tạo một Select query như sau:



**Bước 2:** đổi query hiện hành thành Delete query bằng cách mở thực đơn **Queries | Delete Query**



Để thi hành việc xoá cán bộ, nhấn nút **Run**  trên thanh công cụ. Khi đó một hộp thoại cảnh báo xuất hiện:



Nhấn **Yes** để đồng ý xoá đi các bản ghi (bản ghi đã xoá không thể phục hồi lại được); nhấn **No** để huỷ bỏ lệnh.

Phải cẩn trọng trước khi quyết định lệnh xoá dữ liệu.



Bản chất của Delete query thể hiện ở câu lệnh SQL:-----

**DELETE ... FROM ... [WHERE]**

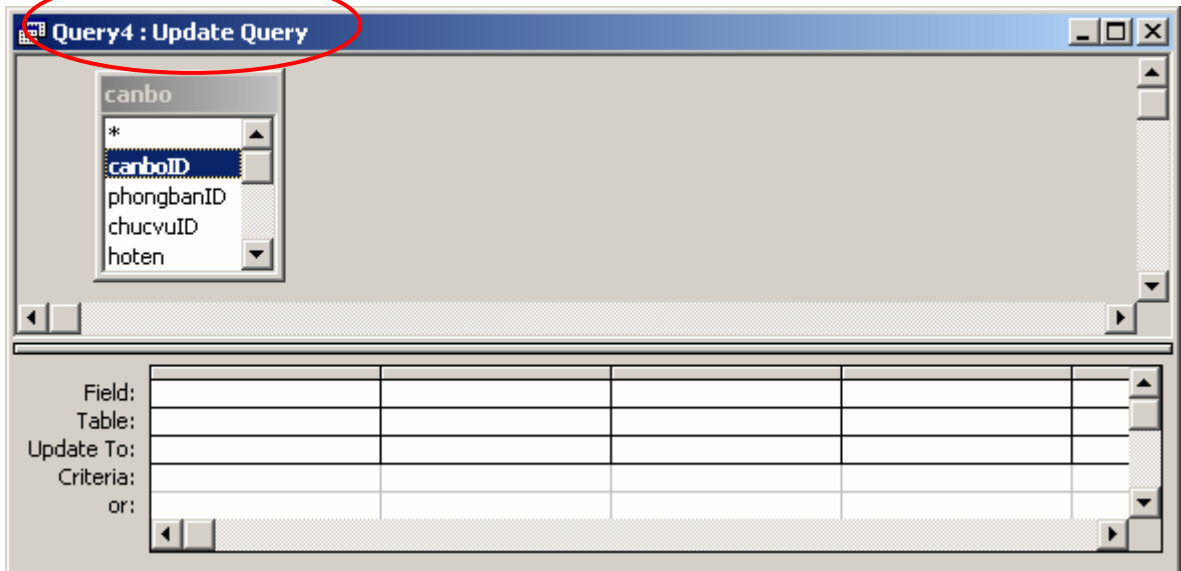
-----

## 6. UPDATE queries

Update query dùng cập nhật dữ liệu một số trường nào đó trong CSDL. Giống với Delete query, Update query là loại query hành động, làm thay đổi nội dung dữ liệu trên CSDL.

Dưới đây là một ví dụ sử dụng **Update query** để tính giá trị cho cột *luongchinh* (lương chính) là một trường mới được thêm vào bảng *canbo*.

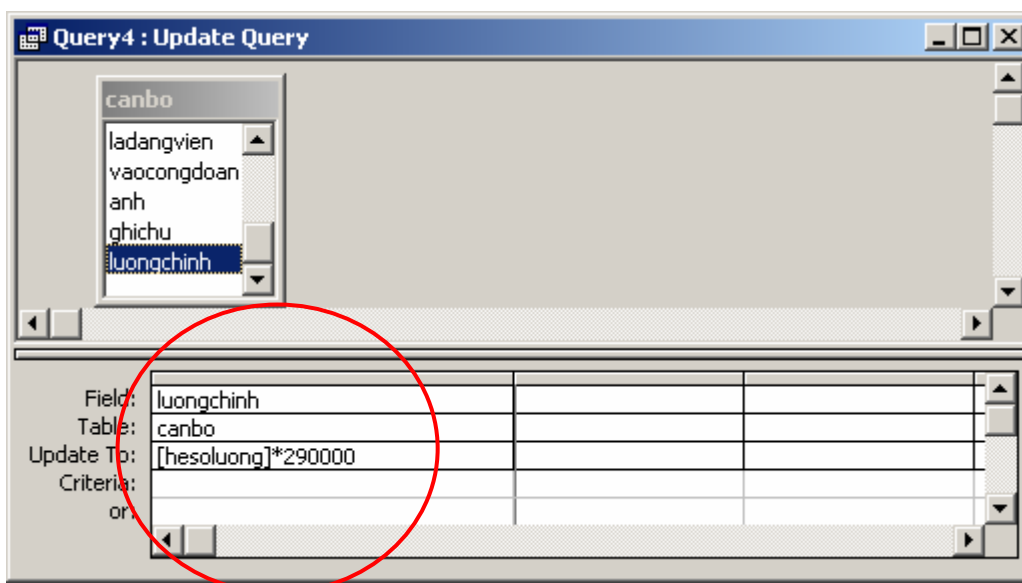
**Bước 1:** Tạo một query và có chứa bảng *canbo* và chuyển thành Update query bằng cách mở thực đơn **Queries | Update query**; hộp thoại thiết kế query như sau:




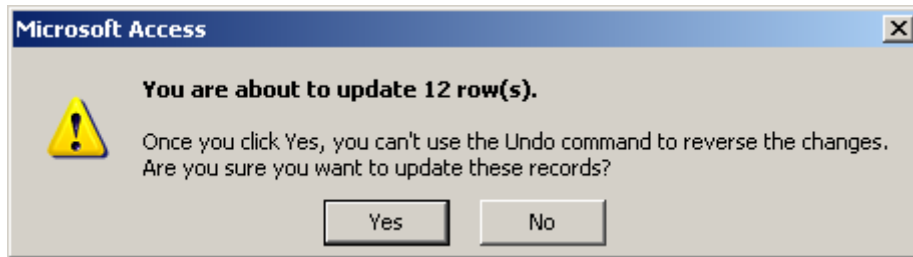
**Bước 2:** Thiết lập các trường cần cập nhật dữ liệu bằng cách:

- Chọn tên trường cần cập nhật dữ liệu ở dòng **Field**;
- Gõ vào biểu thức tính giá trị cho trường đó ở dòng **Update To**;
- Gõ vào điều kiện để tính toán nếu cần ở dòng **Criteria**.

Với yêu cầu trên, **luongchinh = hesoluong \* 290000** của sổ thiết kế query sẽ như sau:



Để thi hành việc xoá cán bộ, nhấn nút *Run*  trên thanh công cụ. Khi đó một hộp thoại cảnh báo xuất hiện:



Nhấn **Yes** để đồng ý cập nhật dữ liệu (dữ liệu sau khi đã cập nhật không thể phục hồi lại được); nhấn **No** để huỷ bỏ lệnh.

Phải cẩn trọng trước khi quyết định lệnh này.



Bản chất của Update query thể hiện ở câu lệnh SQL:-----

**UPDATE ... SET ... [WHERE]**

-----

## Bài tập

### Thực hiện trên CSDL Quản lý lương cán bộ các yêu cầu:

**Bài số 1:** Hãy tính và in ra bảng lương của cơ quan bao gồm các thông tin:

*Hoten, ngaysinh, tenchucvu, tenphongban, luongchinh, phucapcv, dangphi, congdoanphi, thuclinh.*

Trong đó:

**Luongchinh = hesoluong \* 290000**

**Dangphi = 20,000** với những ai là Đảng viên

**Congdoanphi = 5%Luongchinh** với những ai đã vào công đoàn

**Bài số 2:** Hãy đưa ra danh sách các cán bộ là Đảng viên mà chưa vào công đoàn;

**Bài số 3:** Hãy lọc ra danh sách cán bộ của một phòng ban nào đó, tên phòng ban được nhập từ bàn phím khi query được kích hoạt;

**Bài số 4:** Hãy lọc ra danh sách cán bộ là Đảng viên và có thực lĩnh lớn hơn một số nào đó- được nhập từ bàn phím khi query được kích hoạt;

**Bài số 5:** Hãy lọc ra danh sách cán bộ mà năm sinh rơi vào trong một khoảng năm nào đó được nhập vào từ bàn phím. Ví dụ: khi kích hoạt query, Access yêu cầu nhập vào khoảng năm sinh từ năm nào, đến năm nào. Khi đó query sẽ lọc ra danh sách các cán bộ thoả mãn điều kiện đó;

**Bài số 6:** Lọc ra danh sách cán bộ có tên được nhập vào từ bàn phím khi kích hoạt query. Ví dụ: khi query kích hoạt sẽ hỏi: **Gõ vào tên cần xem?** Sau khi gõ vào

một tên (có thể cả họ hoặc cả họ và tên), máy tính sẽ lọc ra danh sách cán bộ thỏa mãn điều kiện;

**Bài số 7:** Lọc ra danh sách cán bộ đến tuổi nghỉ hưu, biết rằng điều kiện nghỉ hưu xác định như sau:

- Là Nam giới, tuổi  $\geq 60$
- Là Nữ giới, tuổi  $\geq 55$

**Bài số 8:** Hãy đưa ra bảng tổng hợp số cán bộ đã vào Đảng hay chưa vào Đảng theo giới tính như sau:

	Giới tính	Chưa vào Đảng	Là Đảng viên
▶	Nam	3	4
	Nữ	3	1

**Bài số 9:** Hãy đưa ra bảng tổng hợp tổng số cán bộ theo từng loại chức vụ của các phòng ban như sau:

	tenphongban	Tổng số CB	Giám đốc	Nhân viên	Phó giám đốc	Trưởng phòng
▶	Kinh doanh	5	1	3		1
	Tổng hợp	6		4	1	1

**Bài số 10:** Đưa ra bảng tổng hợp về chức vụ theo giới tính như sau:

	Chức vụ	Tổng số CB	Nam	Nữ
▶	Giám đốc	1	1	
	Nhân viên	7	4	3
	Phó giám đốc	1	1	
	Trưởng phòng	2	1	1



**Bài số 11:** Đưa ra bảng tổng hợp lương theo từng phòng ban như sau:

tenphongban	TSCB	Tong LC	Tổng CĐP	Tổng Đảng phí	Tổng TL
Kinh doanh	5	3372700	168635	40000	4564065
Tổng hợp	6	4370300	149205	60000	5661095

**Thực hiện trên CSDL Quản lý bán hàng các yêu cầu:**

**Bài số 12:** Tạo query hiển thị chi tiết thông tin từng hàng bán:

ngayban	hoadonID	tenhang	soluong	dongia	thanhtien
12/10/2004	1	Floppy disk	3	8	24
12/10/2004	1	Hard disk	9	67	603
12/10/2004	1	CPU P4 2.4	2	133	266
24/10/2004	2	RAM 256	3	38	114
10/10/2004	3	RAM 256	12	38	456
10/10/2004	3	CPU P4 2.4	3	133	399
12/10/2004	4	Floppy disk	8	8	64
12/10/2004	4	CPU P4 2.4	10	133	1330
*	utoNumber)				

**Bài số 13:** Tạo query đưa ra bảng tổng hợp tổng tiền bán được từng ngày của cửa hàng như sau:

ngayban	Tongtien
10/10/2004	855
12/10/2004	2287
24/10/2004	114

**Bài số 14:** Tạo query đưa ra bảng tổng hợp tổng số bán và tổng tiền đã bán ra của từng loại mặt hàng như sau:

tenhang	Tong so ban	Tongtien
CPU P4 2.4	15	1995
Floppy disk	11	88
Hard disk	9	603
RAM 256	15	570

**Bài số 15:** Tạo bảng tính tổng tiền từng hoá đơn bán hàng như sau:

	hoadonID	ngayban	tenkhach	Tongtien
▶	1	12/10/2004	Nguyễn Đức Ngô	893
	2	24/10/2004	Nguyễn Đức Ngô	114
	3	10/10/2004	Phan Anh Tài	855
	4	12/10/2004	Nguyễn Đức Sơn	1394

**Bài số 16:** Lập bảng tổng hợp số lượng bán ra từng ngày của từng mặt hàng như sau:

tenhang	10/10/2004	12/10/2004	24/10/2004
▶ CPU P4 2.4	3	12	
Floppy disk		11	
Hard disk		9	
RAM 256	12		3

**Bài số 17:** Lập bảng tổng hợp số lượng mua từng mặt hàng của các khách hàng như sau:

tenkhach	10/10/2004	12/10/2004	24/10/2004
▶ Nguyễn Đức Ngô		893	114
Nguyễn Đức Sơn		1394	
Phan Anh Tài	855		

# CHƯƠNG 3

## THIẾT KẾ GIAO DIỆN

Chương này trình bày về FORMs - một công cụ thiết kế giao diện người sử dụng phần mềm- bộ phận giao tiếp giữa người dùng với ứng dụng.

Toàn chương sẽ đề cập đến các mảng kiến thức:

- Các khái niệm về Forms;
- Sử dụng Form wizard;
- Sử dụng Form design view;
- Kỹ thuật Sub-form

Sau khi học xong chương này, học viên sẽ:

- Nắm được các khái niệm về Form, về thiết kế giao diện, về lập trình trực quan, hướng đối tượng;
- Qua đó, học viên hoàn toàn có những kỹ năng tốt để có thể tạo ra được các dạng Form nhập dữ liệu; form điều khiển, xử lý dữ liệu theo những yêu cầu của bài toán thực tế đặt ra. Tuy nhiên, để có một khả năng phát triển chuyên nghiệp hơn, học viên cần chuyên sâu đến phần 2 của giáo trình này nói về Lập trình CSDL.

## 1. Khái niệm Forms

Khi sử dụng một ứng dụng, đa phần công việc của người dùng làm trên các hộp thoại (Dialogue), cửa sổ (Windows). Cả 2 thành phần này trong lập trình đều được gọi là **Form**. Với người dùng, **Form** là giao diện để sử dụng phần mềm; còn với những người phát triển phần mềm, **Form** là những cái mà họ phải nghĩ, phải thiết kế và tạo ra sao cho người dùng họ cảm thấy rất thoải mái, phù hợp và dễ dùng.

Có 2 môi trường dùng tạo Form trong Access:

- Sử dụng trình **Form Wizard**. Đây là cách rất đơn giản, nhanh chóng, dễ dùng giúp tạo nhanh một Form. Tuy nhiên Form được tạo ra có nhiều hạn chế, không gần gũi với yêu cầu của người dùng;
- Sử dụng trình **Form Design View** - một công cụ tương đối hoàn chỉnh để tạo ra các form đáp ứng nhiều yêu cầu khác nhau của người sử dụng.

Tiếp theo sẽ giới thiệu cả 2 môi trường này.

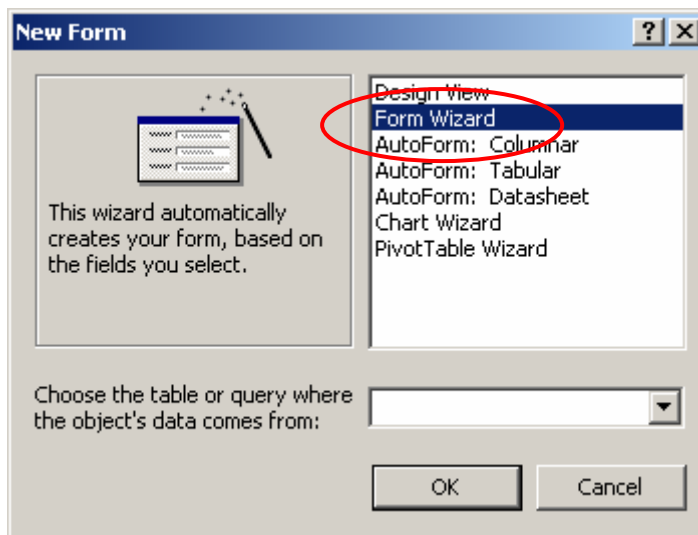
## 2. Sử dụng FORM WIZARD

Form wizard là công cụ trên Access giúp người dùng nhanh chóng và dễ dàng tạo ra một Form dùng làm giao diện nhập dữ liệu cho các bảng trên CSDL.

Dưới đây trình bày các bước để tạo một form làm giao diện nhập dữ liệu cho bảng *CANBO* trong CSDL Quản lý lương cán bộ. (bạn phải chắc chắn đã tạo hoàn chỉnh cấu trúc CSDL trên):

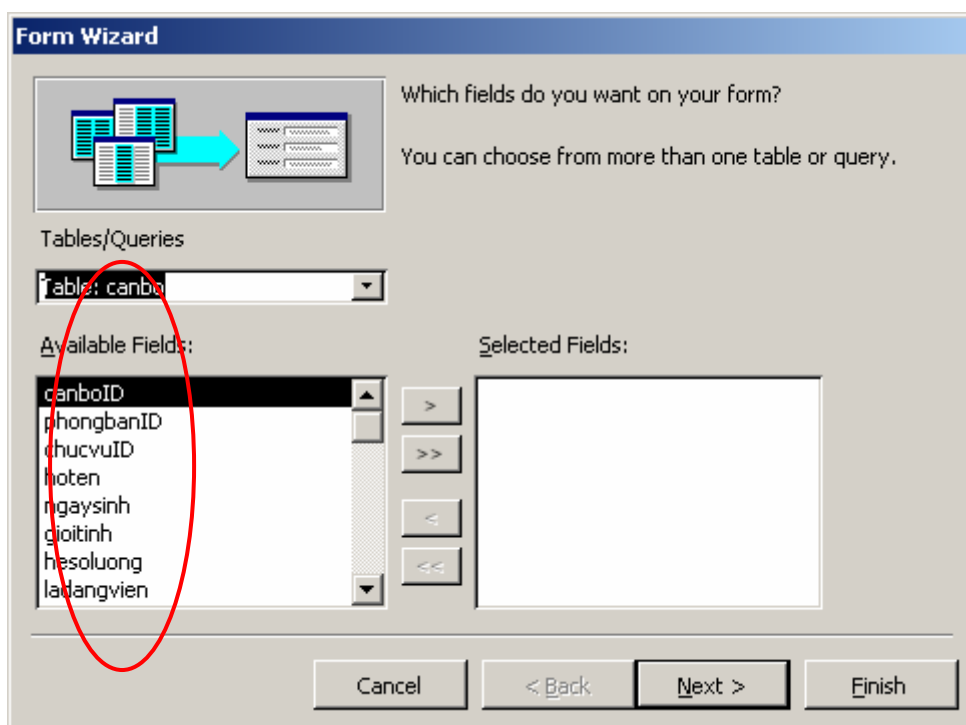
**Bước 1:** Kích hoạt trình Form Wizard

Ở thẻ Forms, nhấn nút **New**, chọn **Form Wizard**, nhấn **OK**;



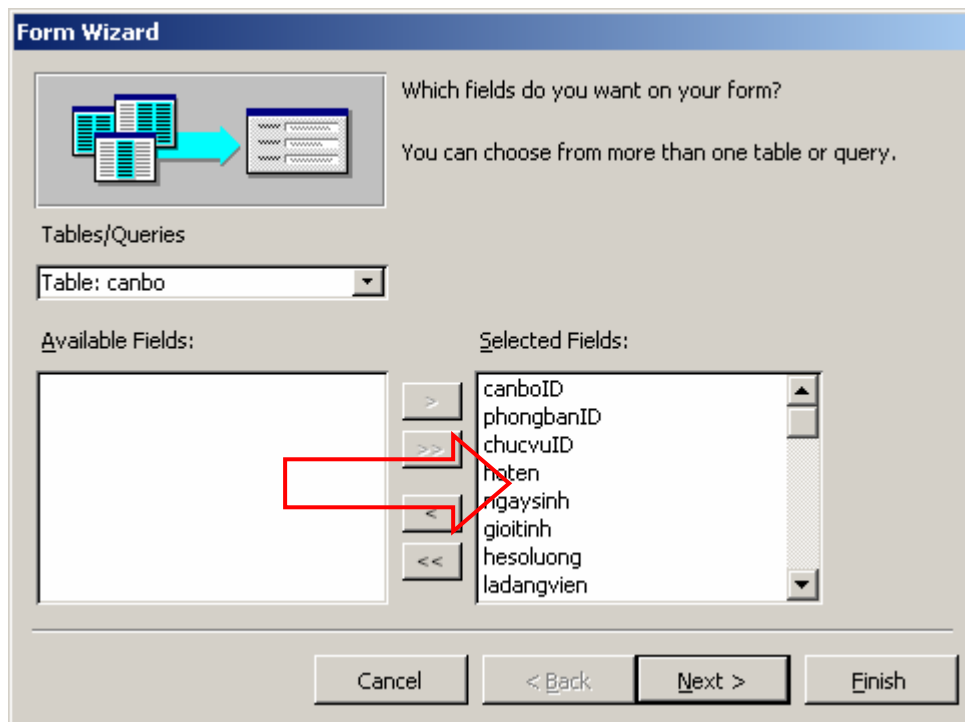
Hoặc có thể nhấn đúp chuột lên Shortcut **Creat Form by using Form wizard**

**Bước 2:** Chọn các thông tin cần đưa lên form. Đây là bước quan trọng nhất để khai báo những trường dữ liệu nào của form sẽ có thể được nhập dữ liệu thông qua form đang tạo. Học viên phải cực kỳ cẩn nhắc các thông tin ở đây: chọn bảng nào và chọn những trường nào cho đúng?



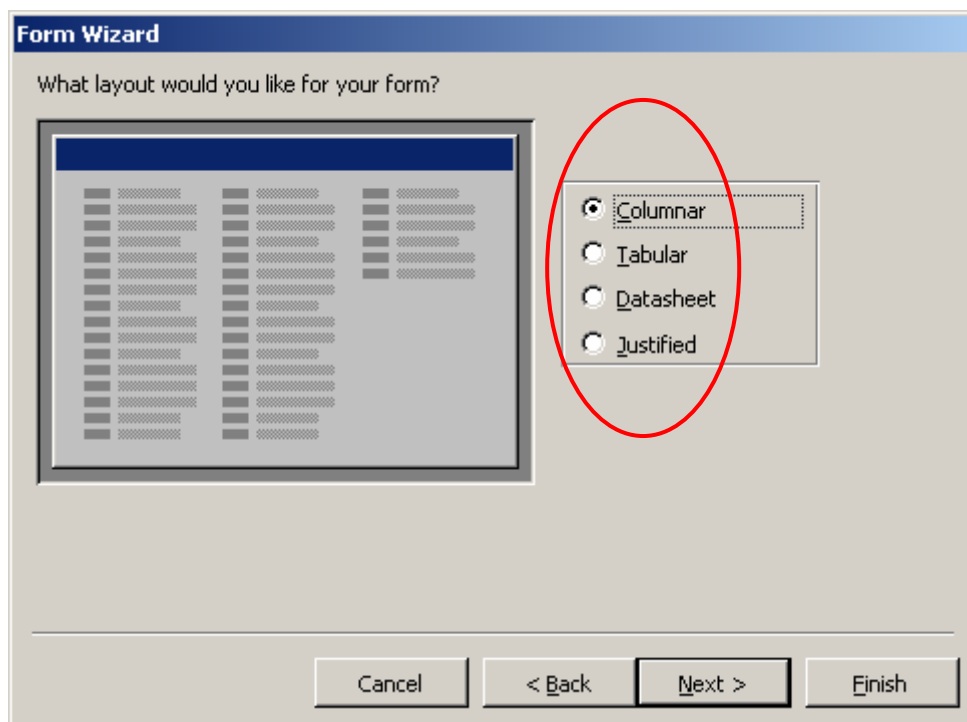
- Chọn bảng - nơi có trường cần đưa lên form để nhập dữ liệu ở hộp **Tables/Queries**;
- Tiếp theo đưa các trường cần nhập dữ liệu lên form từ danh sách **Available Fields**: sang danh sách **Selected Fields**: bằng các nút lệnh: >, >>, <, <<;

- Hãy làm lần lượt đến khi chọn được đủ các trường cần nhập dữ liệu lên form.



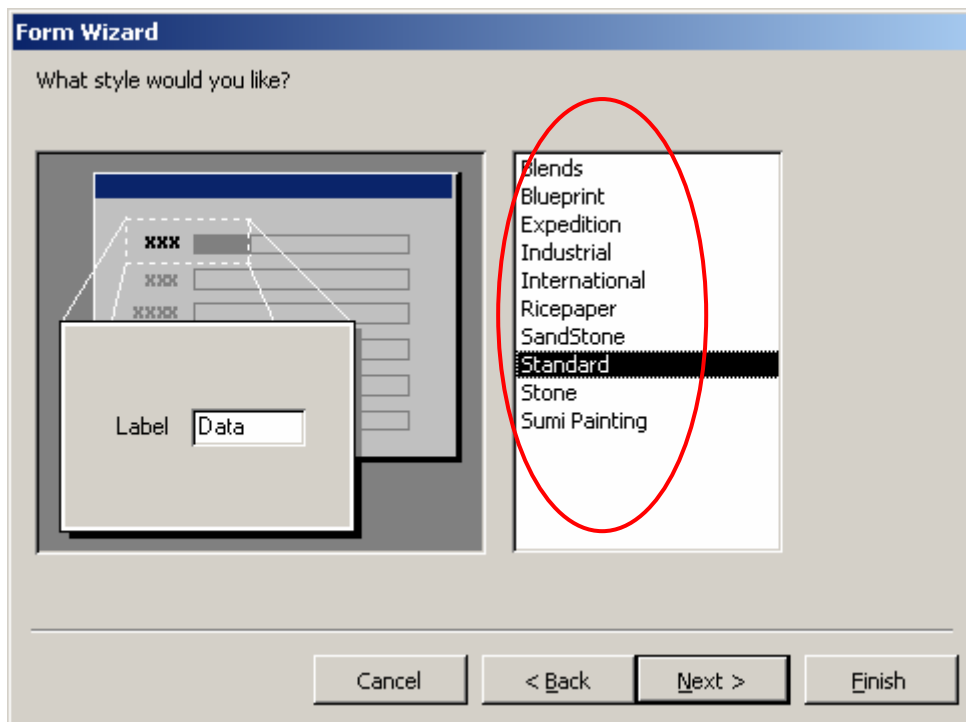
Để tiếp tục nhấn **Next**;

**Bước 3:** Chọn bố cục (Layout) cho form. Có 4 kiểu bố cục có thể thiết lập được cho form là: *Columnar*; *Tabular*; *Datasheet* và *Justified*. Hãy lần lượt nhấn lên từng kiểu bố cục (hình dưới) và xem kết quả minh họa (Preview) ở hộp chữ nhật bên trái hộp thoại để quyết định nên chọn loại nào:



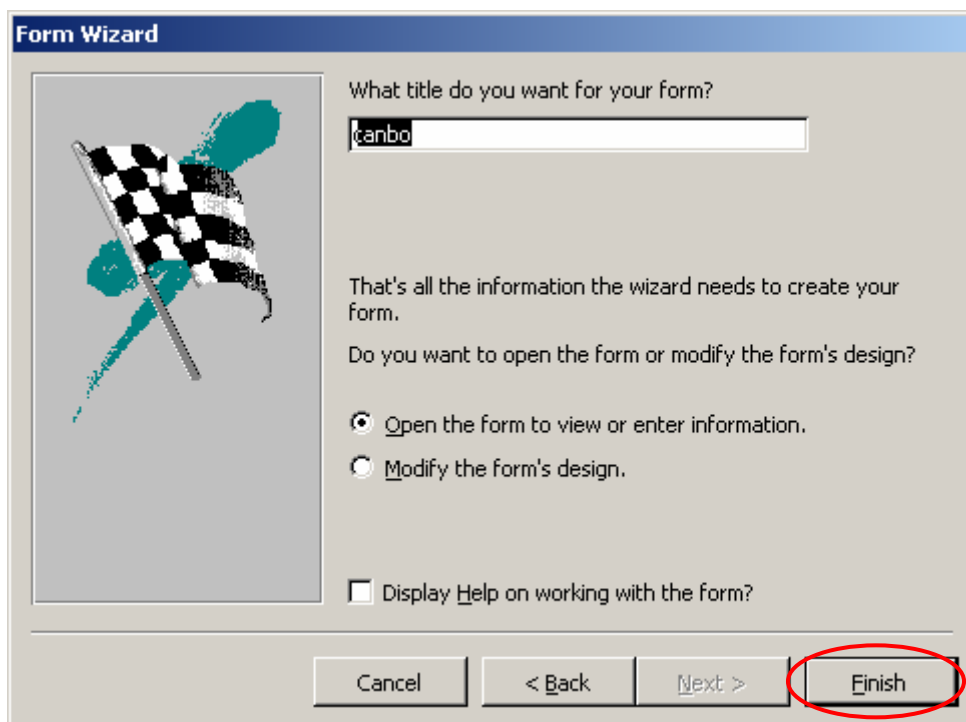
Chọn xong nhấn **Next** để tiếp tục;

**Bước 4:** Chọn kiểu dáng (Style) cho form. Có một danh sách các kiểu dáng để chọn. Hãy thử từng kiểu dáng và chọn cho form một phong cách phù hợp:



Chọn xong nhấn **Next**:

**Bước 5:** Hoàn thiện công việc:




- Bạn có thể gõ vào tên gọi cũng như tiêu đề cho form ở hộp **What title do you want for your form?**
- Có thể tùy chọn kích hoạt ngay form vừa tạo nếu chọn **Open the form to view or enter information** hoặc mở form ra ở chế độ Design view để sửa cấu trúc nếu chọn mục **Modify the form's design**;
- Nhấn **Finish** để hoàn tất toàn bộ công việc.




### Chú ý

Quan trọng nhất là bước 1, 2. Từ bước 3 trở đi có thể bỏ qua bằng cách nhấn **Finish** ngay từ bước 2 nếu đảm bảo toàn bộ các thiết lập từ bước 3, 4, 5 là phù hợp.





### Sử dụng form đã tạo được vào việc nhập dữ liệu:

Chọn form, nhấn nút **Open**. Nếu form đang ở chế độ thiết kế có thể nhấn nút View  trên thanh công cụ. Lúc này có thể sử dụng form để cập nhật dữ liệu:

Tại mỗi thời điểm, form nhập dữ liệu chỉ hiển thị giá trị của một bản ghi. Bạn có thể nhập, sửa trực tiếp các trường của bản ghi hiện tại trên form này. Thanh định hướng (Navigator bar) sẽ giúp xử lý một số thao tác trên form, cách sử dụng như sau:

- Nút  để chuyển đến bản ghi kế sau;



- Nút  để chuyển đến bản ghi kê trước;
- Nút  để chuyển về bản ghi đầu tiên;
- Nút  để chuyển đến bản ghi cuối cùng;
- Nút  để thêm một bản ghi mới

### 3. Sử dụng FORM DESIGN VIEW


Để tạo ra được form mang tính chuyên nghiệp và đáp ứng được sát yêu cầu thực tế bắt buộc phải sử dụng đến **Form Design View**. Nói như vậy không có nghĩa là không dùng Form Wizard để tạo form, mà trong nhiều tính huống vẫn dùng Form Wizard rồi sử dụng đến Form Design View để tiếp tục hoàn thiện yêu cầu.

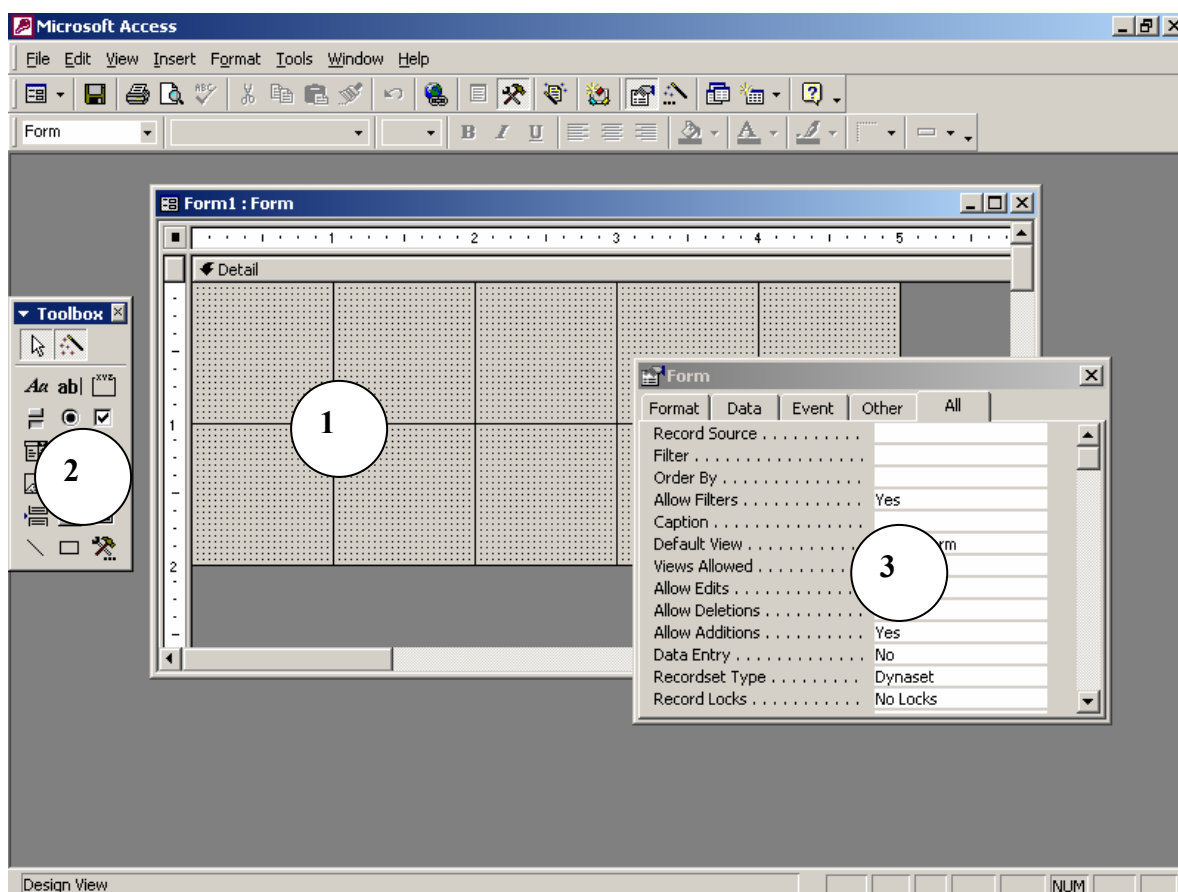
Nội dung mục này trình bày môi trường thiết kế form cũng như cách tạo một số kiểu form từ đơn giản đến phức tạp sử dụng Form Design View.

#### 3.1 Thiết kế Form nhập dữ liệu đơn giản

Chúng ta sẽ tìm hiểu cách thiết kế một form nhập dữ liệu đơn giản thông qua yêu cầu: tạo form dùng làm mẫu nhập dữ liệu cho bảng CANBO.

##### **Bước 1:** Ở thẻ Form, nhấn **New**, chọn **Design View**

Hoặc nhấn biểu tượng  **Create form in Design view** trên cửa sổ QLL Database. Môi trường thiết kế form xuất hiện:



Có 3 thành phần quan trọng để làm việc:

- (1) Cửa sổ **Form** – nơi sẽ thiết kế và xây dựng các thông tin cần thiết theo yêu cầu bài toán. Cấu trúc form gồm 3 phần:

*Form Header* - phần tiêu đề đầu form;

*Form Footer* - phần tiêu đề cuối;

*Detail* - phần thân form..

Toàn bộ các thông tin trên form đều được chứa trong các đối tượng điều khiển (Control), các đối tượng này được lấy từ thanh công cụ Toolbox sẽ trình bày phân tiếp sau.

- (2) Thanh công cụ **Toolbox** – nơi chứa những đối tượng, những công cụ có thể đưa lên form với mục đích thiết kế giao diện và điều khiển dữ liệu theo bài toán. Ví dụ: muốn tạo ô nhập Họ tên trên form có thể dùng đối tượng Textbox, muốn đưa một chú thích (nhãn hiển thị) có thể dùng Label, muốn tạo một nút lệnh có thể dùng CommandButton,...

Mỗi đối tượng sẽ có tập hợp các thuộc tính (Properties) và tập các sự kiện (Events). Thuộc tính để mô tả tính chất cho đối tượng đó, ví dụ như: màu sắc, kích thước, tính chất dữ liệu,.. Sự kiện- nơi có thể gắn các mã lệnh VBA hoặc gắn các Macro lệnh để xử lý những công việc nào đó. Chúng ta sẽ tìm hiểu kỹ việc sử dụng các sự kiện trong phần lập trình VBA.

- (3) Cửa sổ **Properties** – nơi có thể thiết lập các thuộc tính (properties) cho form cũng như các đối tượng trên form;

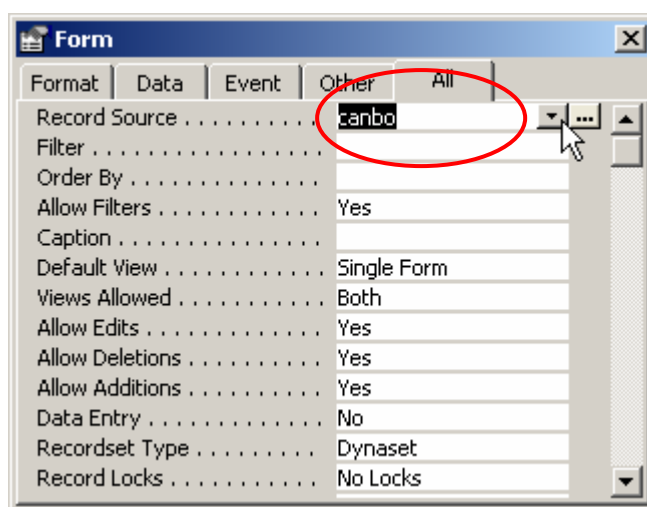
**Bước 2:** Thiết lập nguồn dữ liệu cho form ở thuộc tính Record Source.

Form đang thiết kế là loại để nhập dữ liệu, bước này để xác định nguồn dữ liệu để form làm việc. Vì chỉ để nhập dữ liệu cho bảng CANBO, nên nguồn dữ liệu sẽ là bảng cán bộ. Cách thiết lập thuộc tính này như sau:

- Chọn thuộc tính form bằng cách chọn tên đối tượng *Form* ở hộp chọn **Object** trên thanh công cụ **Formatting**:

Hoặc nhấn chuột lên ô vuông- vị trí giao giữa 2 thước kẻ ngang-dọc của form đang thiết kế. Làm sao khi tiêu đề cửa sổ Properties là *Form* là ok.

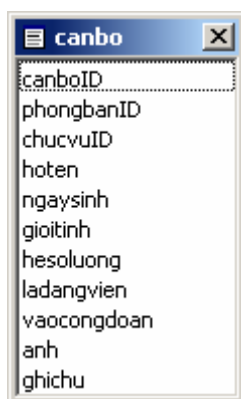
- Thiết lập thuộc tính **Record Source** cho form bằng cách chọn tên bảng *CANBO* ở hộp **Record Source**. Có thể tìm thuộc tính này ở thẻ **Data** - chỉ những thuộc tính liên quan đến dữ liệu; hoặc thẻ **All**- có đầy đủ tất cả các thuộc tính và sự kiện:



**Bước 3:** Mở cửa sổ Field List.

Cửa sổ **Field List** có chứa danh sách các trường trên CSDL có trong nguồn dữ liệu của Form. Nó hỗ trợ việc đưa những trường dữ liệu này lên form để nhập và hiển thị dữ liệu rất tốt. Trong trường hợp này ta dùng để đưa những trường cần nhập dữ liệu từ bảng CANBO lên form.

Nếu chưa thấy cửa sổ này xuất hiện, hãy thực hiện hiển thị nó bằng cách mở thực đơn **View | Field List** hoặc nhấn nút *Field List* trên thanh công cụ chuẩn.



**Bước 4:** Đưa những trường cần nhập dữ liệu từ cửa sổ Field List lên Form đang thiết kế bằng cách: Dùng chuột kéo từng trường muốn thiết kế lên form từ cửa sổ Field List thả lên vị trí hợp lý trên form (với bài này phải kéo toàn bộ các trường lên form). Lúc này cửa sổ thiết kế form có dạng:

Mỗi khi kéo một trường từ Field List lên form, Access sẽ tự động tạo một đối tượng gắn kết tới trường dữ liệu tương ứng, đối tượng này có thể là Textbox, Combobox hay đối tượng khác tùy thuộc vào kiểu dữ liệu của trường tương ứng; và đối tượng Label đi kèm nhằm tạo nhãn chú thích cho trường dữ liệu.

Đến đây đã tạo xong form nhập dữ liệu đơn giản cho một bảng.

### 3.2 Tinh chỉnh cấu trúc Form

Khi thiết kế một form, đòi hỏi lập trình viên không những phải tạo ra được form đáp ứng đầy đủ các yêu cầu về xử lý dữ liệu của bài toán, mà còn phải đáp ứng tính thẩm mỹ. Thế nào là form có thẩm mỹ cao? Câu hỏi này rất khó! Tùy thuộc vào bài toán, vào đối tượng người sử dụng mà thiết kế những kiểu giao diện thân thiện. Thanh công cụ Toolbox và cửa sổ Properties là những công cụ cần thiết để thiết kế giao diện.

#### a. Sửa thuộc tính

Mở form để sửa ở chế độ thiết kế (Design view) bằng cách: chọn form, nhấn nút **Design**; hoặc nhấn nút *Design* trên thanh công cụ. Màn hình thiết kế form xuất hiện:

Tùy từng mục đích, đối tượng làm việc cụ thể mà có các cách làm việc khác nhau. Sau đây là một số ví dụ:

#### Sửa nhãn (Label):

Label là đối tượng tạo ra dòng chữ chú thích trên form. Dòng chữ này (nhãn) phải được nhập trực tiếp từ bàn phím làm sao thật ngắn gọn, xúc tích để người dùng có thể hiểu được bản chất cũng như ý nghĩa của việc bạn đang giải thích.

Giá trị hiển thị trên nhãn chính là giá trị của thuộc tính *Caption*. Sửa nhãn là sửa thuộc tính *Caption* hoặc có thể bấm chuột trực tiếp lên nhãn để sửa giá trị.

### **Thay đổi kích thước đối tượng (Resize):**

Kích thước của đối tượng thường được mô tả ở thuộc tính: *Height* - chiều cao và *With* - chiều rộng. Tuy nhiên bạn hoàn toàn có thể thay đổi kích thước đối tượng một cách trực quan bằng chuột. Cách làm tương tự như việc thay đổi kích thước một đối tượng đồ họa (Graphic) trên Word. Cụ thể như sau:

Khi trỏ chuột (chọn) lên đối tượng cần thay đổi, sẽ xuất hiện 6 điểm trên đối tượng:



Khi đó có thể đặt chuột lên từng điểm (khi nào con trỏ chuột chuyển thành mũi tên 2 đầu) để điều chỉnh độ lớn bằng cách giữ và di chuột.

### **Di chuyển đối tượng:**

Mỗi đối tượng nằm trên form đều được xác định bởi một tọa độ, tọa độ này được thể hiện qua 2 thuộc tính: *Top* - khoảng cách từ tiêu đề form đến đối tượng và *Left* - khoảng cách từ mép form bên trái đến đối tượng. Tuy nhiên bạn nên di chuyển đối tượng bằng cách trực quan sử dụng chuột với thao tác kéo-thả. Chỉ trong những trường hợp đòi hỏi phải di chuyển đối tượng đến các vị trí chính xác mới cần dùng đến 2 thuộc tính *Top* và *Left* để thiết lập.

### **Thay đổi Font chữ:**

Những đối tượng có hiển thị chữ (Text) hoàn toàn có thể thay đổi được phong chữ. Giá trị này được miêu tả ở thuộc tính *Font Name*. Tuy nhiên bạn hoàn toàn sử dụng hộp *Font* trên thanh công cụ *Formatting* để thiết lập nhanh.

### **Thay đổi màu nền:**

Mỗi đối tượng thường có thể thiết lập được màu nền, chúng thể hiện ở thuộc tính *BackColor*. Tuy nhiên hoàn toàn có thể thiết lập màu nền một cách nhanh chóng ở hộp *Fill\Back Color* trên thanh *Formatting*.

### Thay đổi màu chữ:

Mỗi đối tượng thường hiển thị chữ có thể thiết lập được màu chữ, chúng thể hiện ở thuộc tính *ForeColor*. Tuy nhiên hoàn toàn có thể thiết lập màu chữ một cách nhanh chóng ở hộp *Font\Fore Color* trên thanh Formatting

Ngoài ra có thể sử dụng thuộc tính *Format* và *InputMark* để định dạng dữ liệu khi hiển thị cũng như mặt nạ khi nhập dữ liệu. Xin tham khảo cách làm này ở phần các thuộc tính của bảng dữ liệu (Chương 1; mục 2- Xây dựng cấu trúc bảng).

Hãy sử dụng các tính năng định dạng như trên một cách phù hợp, hoàn toàn có thể đưa form ban đầu về dạng dễ nhìn, dễ sử dụng hơn như sau:

The screenshot shows a Microsoft Access form window titled "Form1 : Form". The form is titled "NHẬP HỒ SƠ CÁN BỘ" (Enter Staff Record). It contains the following fields and controls:

- Mã cán bộ: canboID (text box)
- Phòng ban: phongbanID (dropdown menu)
- Họ tên: hoten (text box)
- Chức vụ: chucvuID (dropdown menu)
- Ngày sinh: ngaysinh (text box)
- Hệ số lương: hesoluong (text box)
- Ghi chú: ghichu (text area)
- Three checked checkboxes: "Là nam giới", "Là Đảng viên", and "Vào công đoàn".

### b. Sử dụng Command Button Wizard

Tuy chưa được học một chút gì về lập trình trên Access, nhưng học viên vẫn có thể tạo được một số các nút lệnh đặt trên form điều khiển công việc. Tính năng **Command Button Wizard** giúp làm điều đó.

Nút lệnh (Command Button) thường được dùng để lập trình xử lý các công việc nào đó. Ví dụ: khi nhấn lên nút **In bảng lương** - một bảng lương sẽ được in ra hoặc khi nhấn nút **Đóng Form** - form đang làm việc sẽ được đóng lại. Muốn vậy, người lập trình phải viết các lệnh phía sau nút đó, để làm sao khi nhấn lên nút, các lệnh sẽ

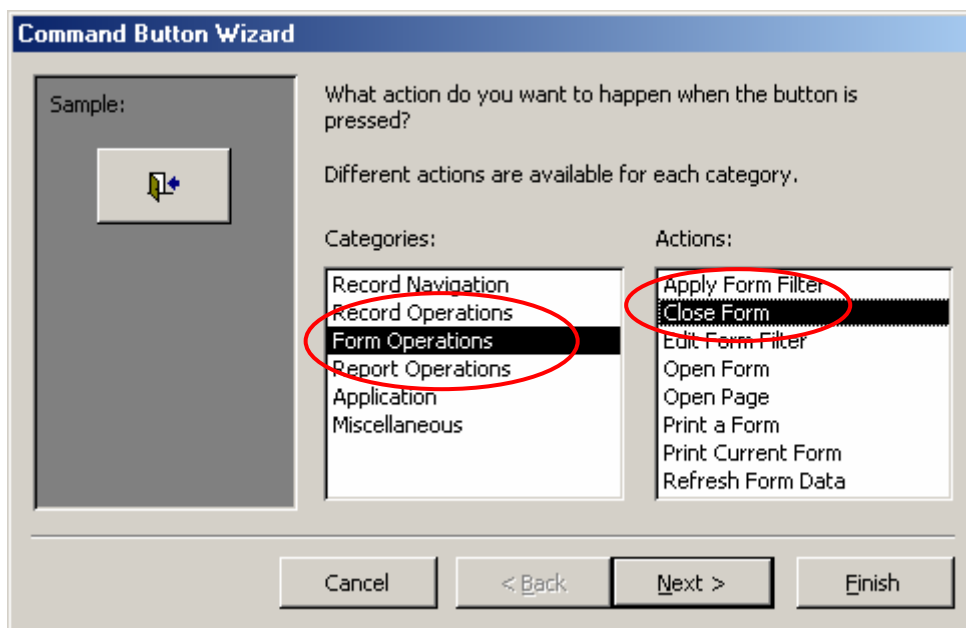
được thi hành để điều khiển công việc như yêu cầu. Control Button Wizard sẽ giúp tạo một số loại nút lệnh mà không cần biết đến lập trình. Dưới đây là các bước sử dụng:

**Ví dụ: Tạo nút** 

Trước khi sử dụng tính năng này, phải đảm bảo nút **Control Wizard** trên thanh công cụ Toolbox đã được nhấn chìm xuống:



**Bước 1:** Dùng chuột đưa đối tượng **Command Button** từ thanh công cụ lên vị trí thích hợp trên Form, hộp thoại sau xuất hiện:



**Bước 2:** Chọn hành động cần làm cho nút lệnh. Quan sát hộp thoại trên có 2 danh sách:

- **Categories:** chứa các nhóm thao tác mà một nút lệnh có thể nhận;
- **Actions:** chứa danh sách các lệnh của mỗi nhóm.

Bảng dưới đây liệt kê danh sách lệnh của từng nhóm:

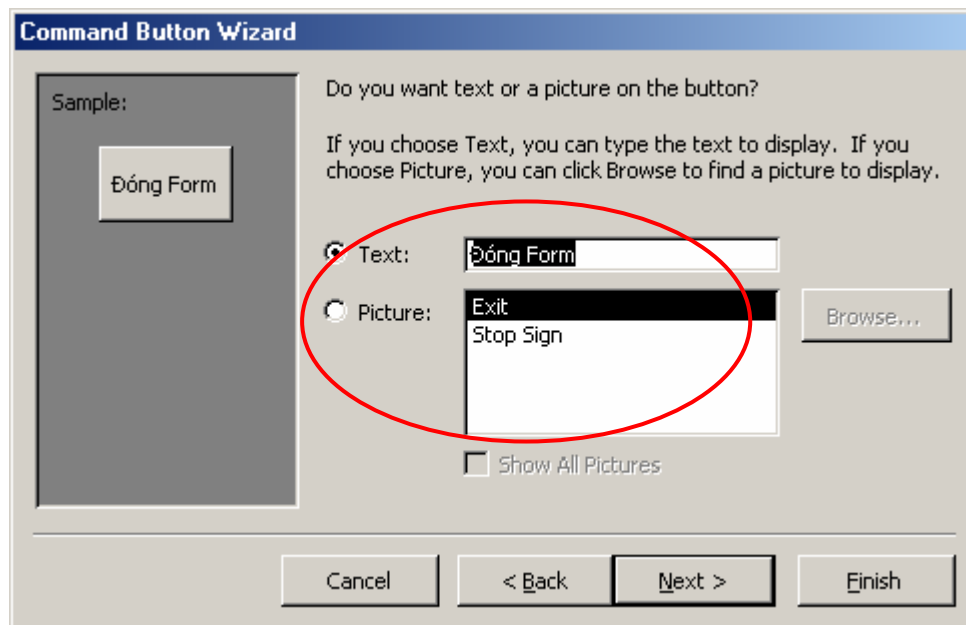
TT	Nhóm \ Lệnh	Ý nghĩa
1	<b>Record Navigation</b>	<b>Nhóm định hướng bản ghi</b>



1.1	- Goto First Record	- Chuyển về bản ghi đầu
1.2	- Goto Last Record	- Chuyển đến bản ghi cuối cùng
1.3	- Goto Next Record	- Chuyển bản ghi kế sau
1.4	- Goto Previous Record	- Chuyển bản ghi kế trước
2	<b>Record Operations</b>	<b>Các xử lý với bản ghi</b>
2.1	- AddNew Record	- Thêm bản ghi mới
2.2	- Delete Record	- Xoá bản ghi hiện tại
2.3	- Update Record	- Cập nhật bản ghi hiện tại
2.4	- Undo Record	- Phục hồi thay đổi dữ liệu bản ghi
3	<b>Form Operations</b>	<b>Các xử lý với Form</b>
3.1	- Close Form	- Đóng form
3.2	- Open Form	- Mở một form khác
4	<b>Report Operations</b>	<b>Các xử lý với Report</b>
4.1	- Preview report	- Xem trước nội dung (Preview) report
4.2	- Print report	- In report
4.3	- Send report to file	- Xuất report ra một tệp tin ngoài
5	<b>Applications</b>	<b>Xử lý ứng dụng</b>
5.1	- Quit Application	- Thoát khỏi Access
5.2	- Run Application	- Chạy một ứng dụng nào đó (tệp .exe)
6	<b>Miscellaneous</b>	<b>Một số thao tác khác</b>
6.1	- Print table	- In nội dung một bảng
6.2	- Run macro	- Thi hành một Macro
6.3	- Run query	- Thi hành một query

Với yêu cầu đóng form, phải chọn: **Categories** = *Form* Actions và **Actions** = *Close Form*.

Chọn xong nhấn **Next**:

**Bước 3:** Chọn hiển thị cho nút lệnh trên hộp thoại dưới:

- Chọn tùy chọn **Text** nếu muốn nút lệnh hiển thị bằng chữ. Khi đó gõ vào chữ hiển thị trên nút. Hình trên gõ *Đóng Form*;
- Chọn tùy chọn **Picture** nếu muốn thể hiện một hình ảnh lên nút lệnh. Khi đó có thể chọn một hình ảnh trên danh sách.

Bạn có thể quan sát trước được thể hiện của nút lệnh ở hộp Sample bên trái hộp thoại.

- Cuối cùng, nhấn **Finish** để kết thúc việc tạo một nút lệnh.

## 4. Kỹ thuật Sub-form

Kỹ thuật thiết kế form đã trình bày ở các phần trên có thể gọi là Single-form. Sub-form là kỹ thuật thiết kế giao diện rất mạnh, đáp ứng được những yêu cầu xử lý dữ liệu phức tạp mà kỹ thuật Single-form chưa thể đáp ứng. Có thể hiểu Sub-form là việc form này lồng trong form kia (có thể lồng trong nhau nhiều lớp). Form chứa gọi là form mẹ (Main form); form được lồng vào gọi là form con (Sub-form). Việc xử lý dữ liệu trên từng form có thể xử lý độc lập hoặc có quan hệ với nhau tùy theo mục đích công việc.

Sub-form có thể được sử dụng trong các loại form nhập dữ liệu, hoặc sử dụng để hiển thị dữ liệu. Dưới đây minh họa 2 ví dụ tạo form có sử dụng kỹ thuật này.

## Sử dụng Sub-form trong form nhập dữ liệu:

**Bài toán:** Thiết kế form nhập thông tin hoá đơn bán hàng như sau:

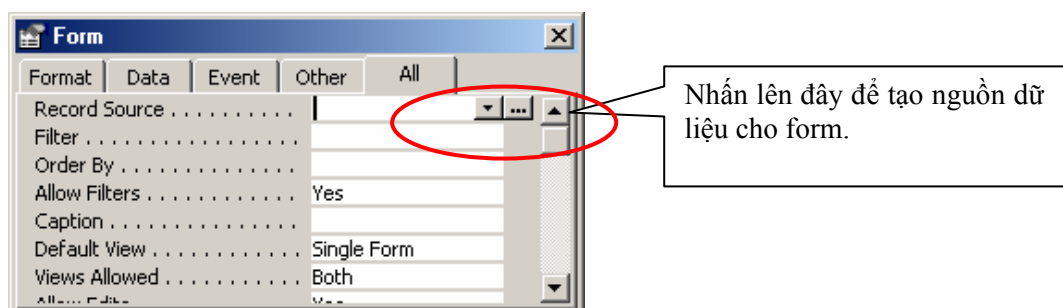
hangID	donvi	dongia	soluong	Thanhtien
Floppy disk	chiếc	8	3	24
Hard disk	ổ	67	9	603
CPU P4 2.4	con	133		266

- Với yêu cầu này, phải tạo một form con cho phép hiển thị và nhập các thông tin về hàng bán (1);
- Phải tạo một form mẹ để nhập thông tin về hoá đơn bán hàng (2);
- Phải kết nối form mẹ với form con thông qua trường hoadonID.

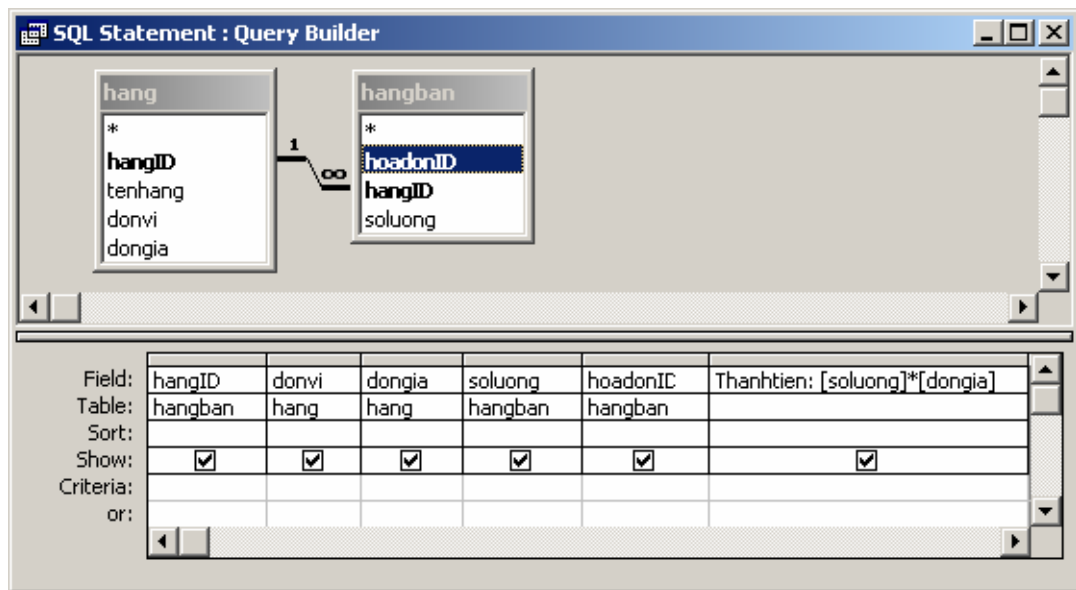
Cách làm như sau (yêu cầu thực hiện theo đúng thứ tự công việc các bước):

### Bước 1: Tạo form con

- Tạo mới form ở chế độ **Design View**;
- Vì form con sẽ hiển thị dữ liệu dạng bảng nên phải thiết lập thuộc tính **Default View** cho form con là *Datasheet*;
- Thiết lập nguồn dữ liệu làm việc cho form con ở thuộc tính **Record Source** bằng cách nhấn nút ... (hình dưới)



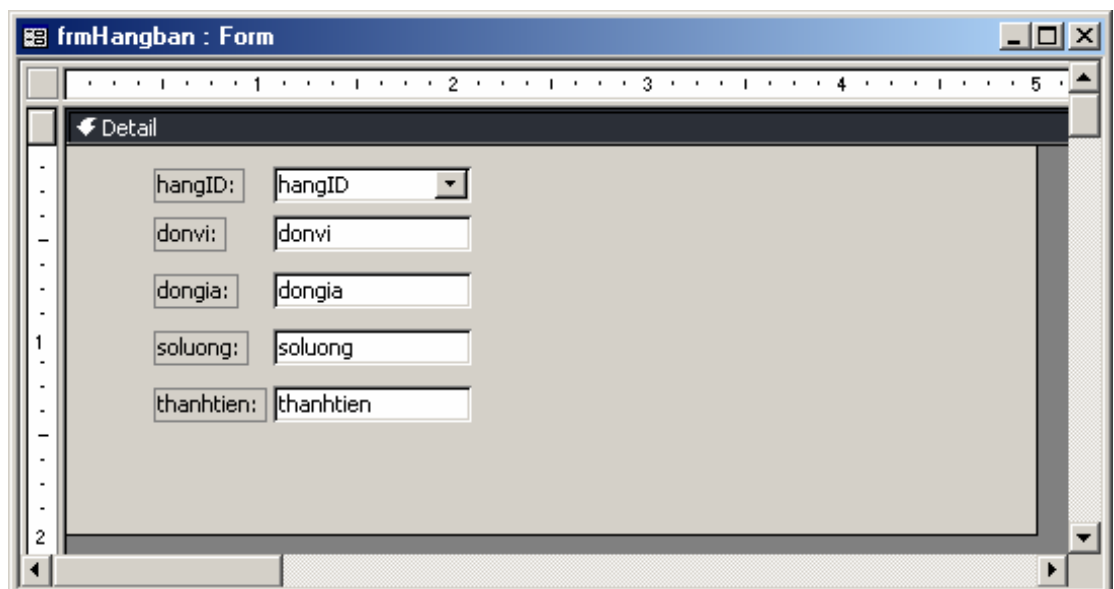
Một màn hình thiết kế query xuất hiện. Hãy thiết lập query này như sau:



(Chú ý phải chọn đúng tên trường của các bảng đã chỉ dẫn)

Tiếp theo đóng màn hình thiết kế query

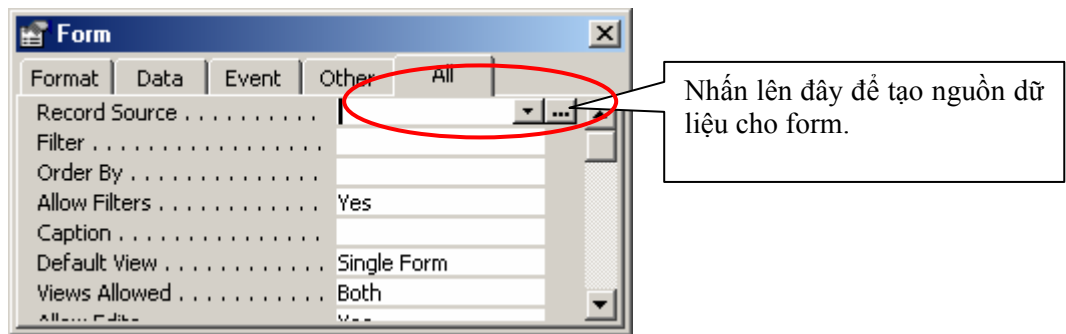
- Sử dụng cửa sổ **Field List** để đưa các trường cần thiết lên vùng Detail của form bao gồm: *hangID*, *donvi*, *dongia*, *soluong*, *thanhtien*;
- Ghi lại form với một tên gọi (ví dụ: *frmHangban*) và đóng lại, chuẩn bị tạo form mẹ.



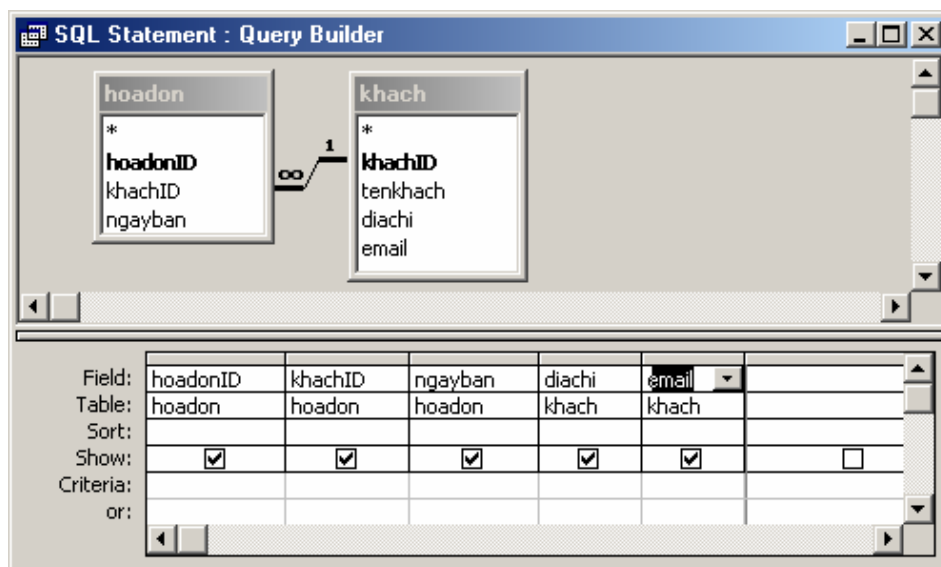
## BƯỚC 2: Tạo form mẹ

- Tạo mới form ở chế độ **Design View**;

- Thiết lập nguồn dữ liệu làm việc cho form mẹ ở thuộc tính **Record Source** bằng cách nhấn nút ... (hình dưới)



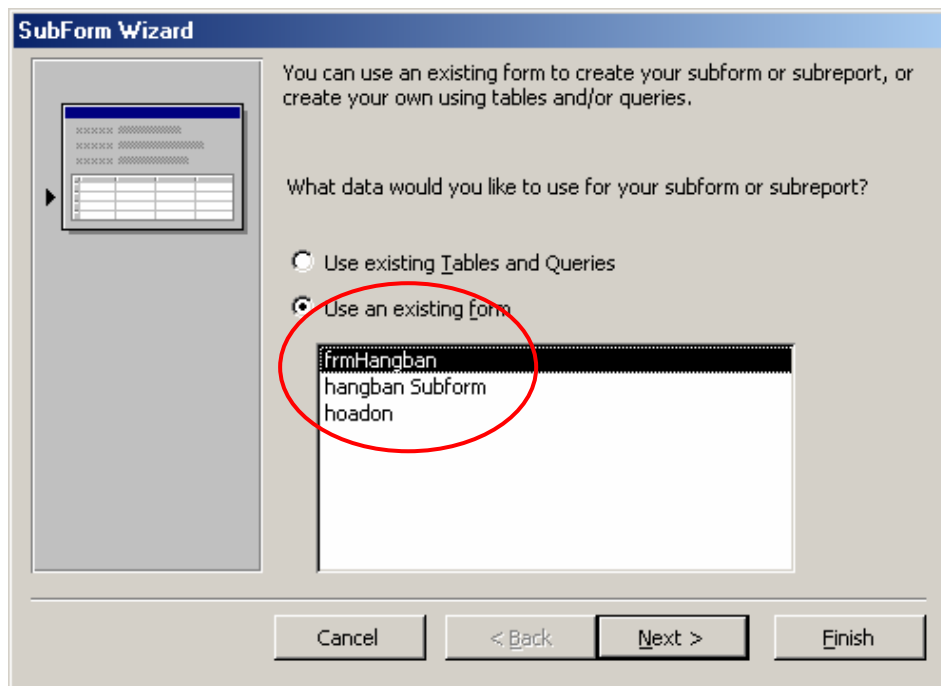
Một màn hình thiết kế query xuất hiện. Hãy thiết lập query này như sau:



*(Chú ý phải chọn đúng tên trường của các bảng đã chỉ dẫn)*

Tiếp theo đóng màn hình thiết kế query

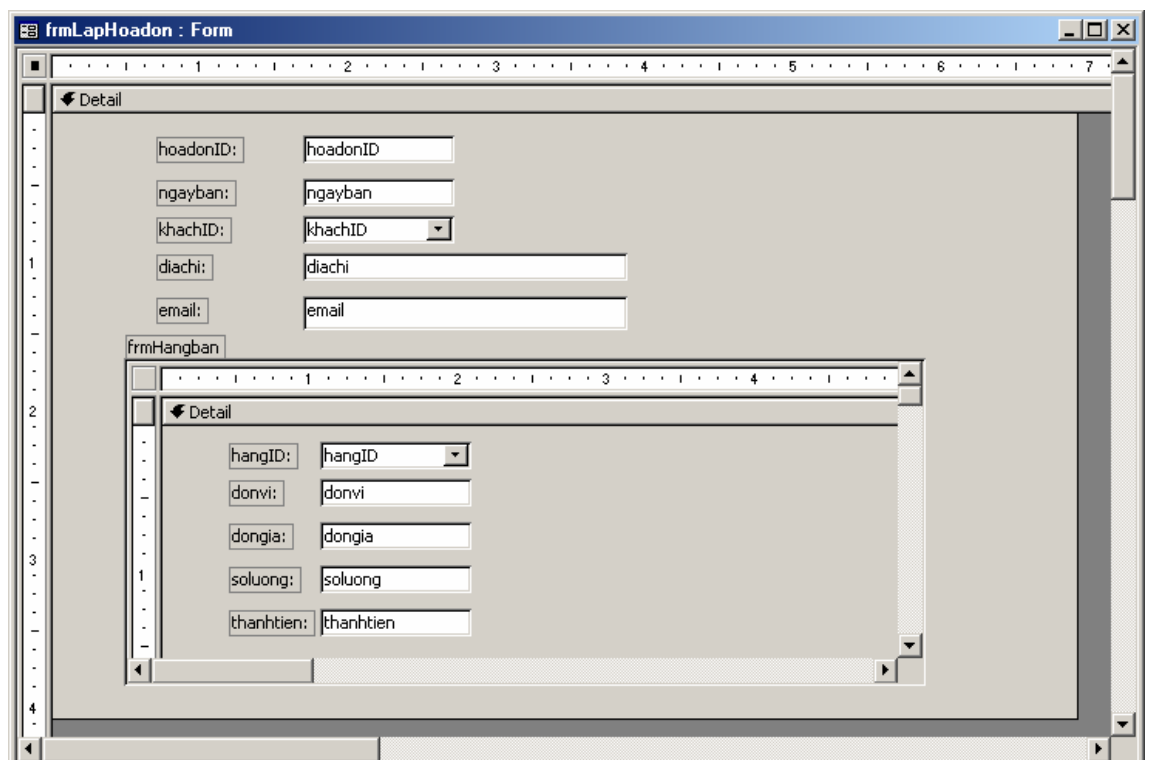
- Sử dụng cửa sổ **Field List** để đưa các trường cần thiết lên vùng Detail của form bao gồm: *hoadonID, khachID, ngayban, diachi, email*;
- Sử dụng đối tượng **Subform\Subreport** trên thanh công cụ Toolbox để đưa form con lên form mẹ. Sau khi đưa một đối tượng Subform\Subreport lên form, một hộp thoại xuất hiện;



Hãy chọn tên form con cần đưa lên ở danh sách **Use an existing form**.  
 Chọn xong nhấn **Finish**.

*(Có thể đưa form con lên form mẹ bằng cách dùng chuột gấp tên form con  
 thả lên form mẹ)*

- Ghi lại form với một tên gọi (ví dụ: *frmLapHoadon*) và đóng lại, kết thúc yêu cầu cơ bản bài toán, thu được kết quả như sau:



Khi sử dụng Form wizard, bạn cũng rất dễ dàng khi tạo form theo yêu cầu trên. Cách làm ở đây là kết hợp giữa form wizard với form Design view. Chúng tôi trình bày thêm cách này để các bạn tham khảo.

**Bước 1:** Tạo form wizard lấy những dữ liệu sau:

Bảng *HOADON* lấy các trường: *hoadonID*, *khachID*, *ngayban*;

Bảng *KHACH* lấy các trường: *diachi*, *email*;

Bảng *HANGBAN* lấy các trường: *hangID*, *soluong*;

Bảng *HANG* lấy các trường: *donvi*, *dongia*.

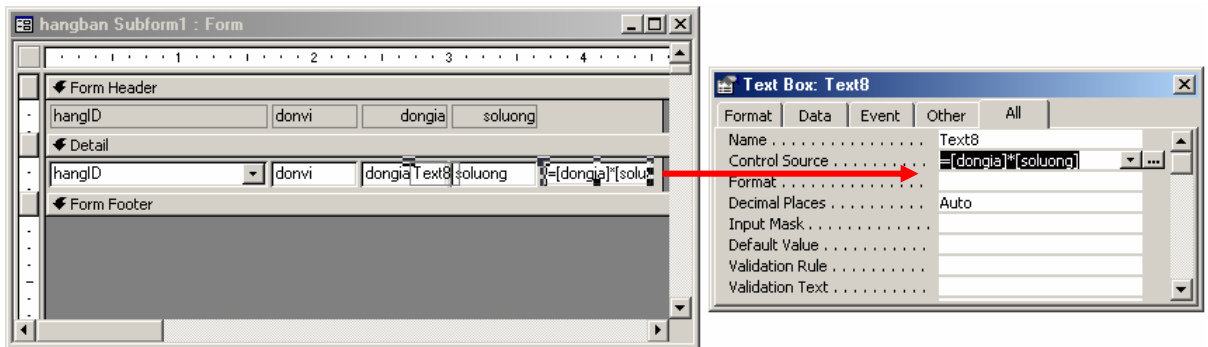
Sau bước này Access sẽ tạo được form theo yêu cầu có dạng:

hangID	donvi	dongia	soluong
Floppy disk	chiếc	8	3
Hard disk	ổ	67	9
CPU P4 2.4	con	133	2

Form này gần đáp ứng yêu cầu đầu bài (chỉ thiếu cột *Thanhtien* trong form con).

**Bước 2:** Sửa form con để đáp ứng bài toán:

- Mở form con ở chế độ Design View (thông thường tên form con này Access sẽ đặt *hangban subform*);
- Thêm một **Textbox** lên vùng Detail của form con và thiết lập thuộc tính **Control Source** cho text box này như sau (= **[dongia]\*[soluong]**):



- Đóng và ghi lại tất cả. Bạn đã tạo được form như yêu cầu.

### Sử dụng Sub-form trong form lọc dữ liệu

**Bài toán:** Thiết kế form lọc ra danh sách cán bộ một phòng ban nào đó như sau:

	Họ tên	Ngày sinh	Giới tính	Chức vụ
▶	Nguyễn Đức Ngô	4/12/1976	Nam	Giám đốc
	Phan Quang Sáng	8/8/1976	Nữ	Nhân viên
	Nguyễn Văn Hoà		Nữ	Nhân viên
	Phan Hoài Linh		Nam	Nhân viên
	Phan Anh Quân	7/23/1978	Nữ	Trưởng phòng
*				

Record: 1 of 5

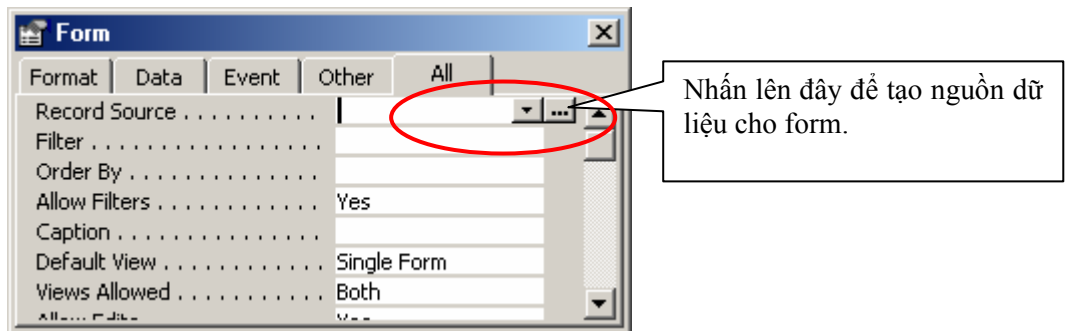
- Với yêu cầu này, phải tạo một form con cho phép hiển thị các thông tin về danh sách cán bộ cần in ra (1);
- Phải tạo một form mẹ cho phép nhập các tham số cần lọc (2);
- Phải thiết lập tham số cho form con sao cho mỗi khi chọn một phòng ban trên form mẹ, danh sách cán bộ ở phòng ban đó sẽ tự động hiển thị ra form con.

Cách làm như sau (yêu cầu thực hiện theo đúng thứ tự công việc các bước):

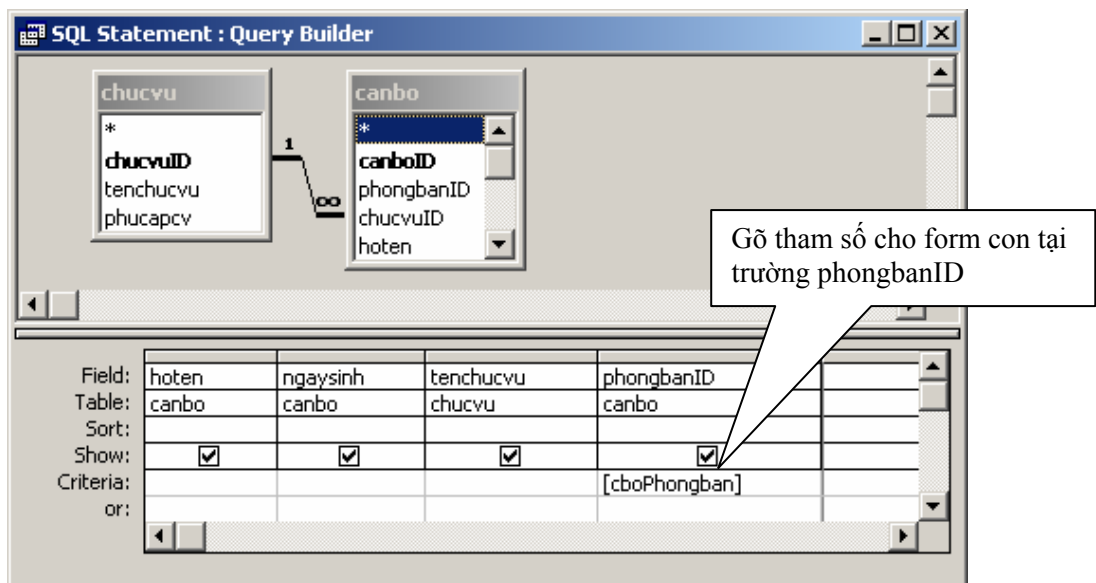


**Bước 1:** Tạo form con

- Tạo mới form ở chế độ Design View;
- Vì form con sẽ hiển thị dữ liệu dạng bảng nên phải thiết lập thuộc tính **Default View** cho form con là *Datasheet*;
- Thiết lập nguồn dữ liệu làm việc cho form con ở thuộc tính **Record Source** bằng cách nhấn nút ... (hình dưới)



Một màn hình thiết kế query xuất hiện. Hãy thiết lập query này như sau:



(Chú ý phải chọn đúng tên trường của các bảng đã chỉ dẫn)

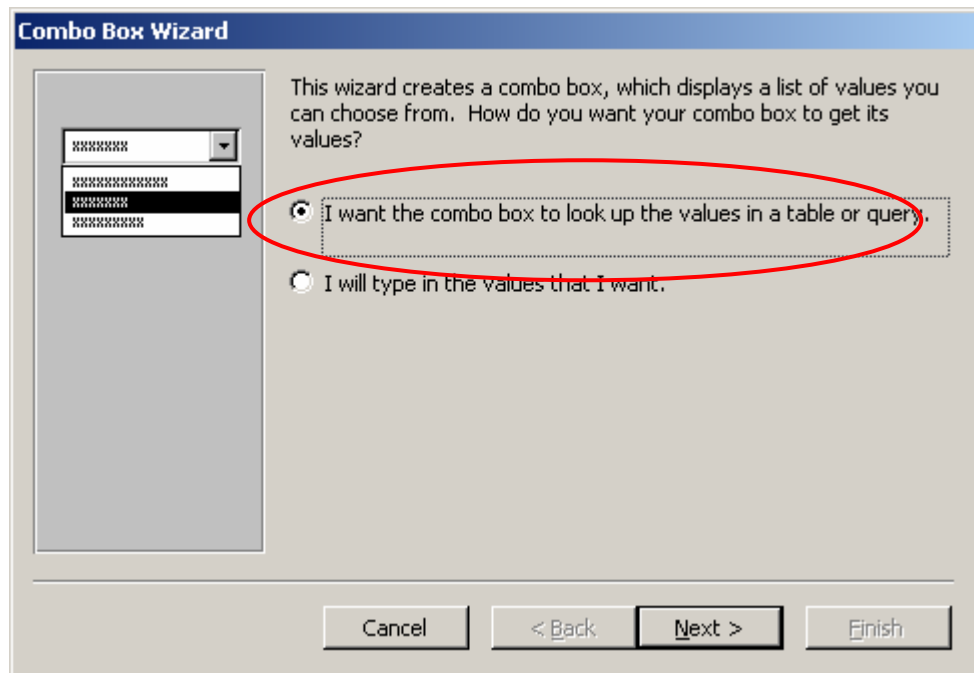
Tiếp theo đóng màn hình thiết kế query

- Sử dụng cửa sổ **Field List** để đưa các trường cần thiết lên vùng Detail của form bao gồm: *hoten*, *gioitinh*, *ngaysinh*, *tenchucvu*;
- Ghi lại form với một tên gọi (ví dụ: *frmDscb\_subform*) và đóng lại, chuẩn bị tạo form mẹ.

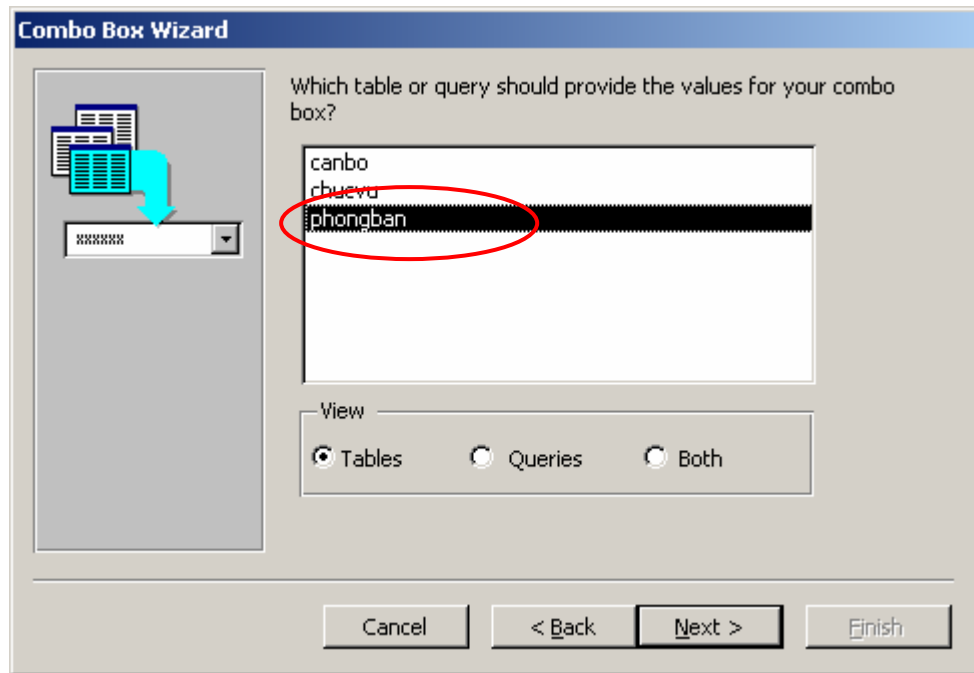
**Bước 2:** Tạo form mẹ

- Tạo mới form ở chế độ **Design View**;
- Tạo hộp **Combo box** chứa danh sách các phòng ban có thể chọn trên form mẹ bằng cách:

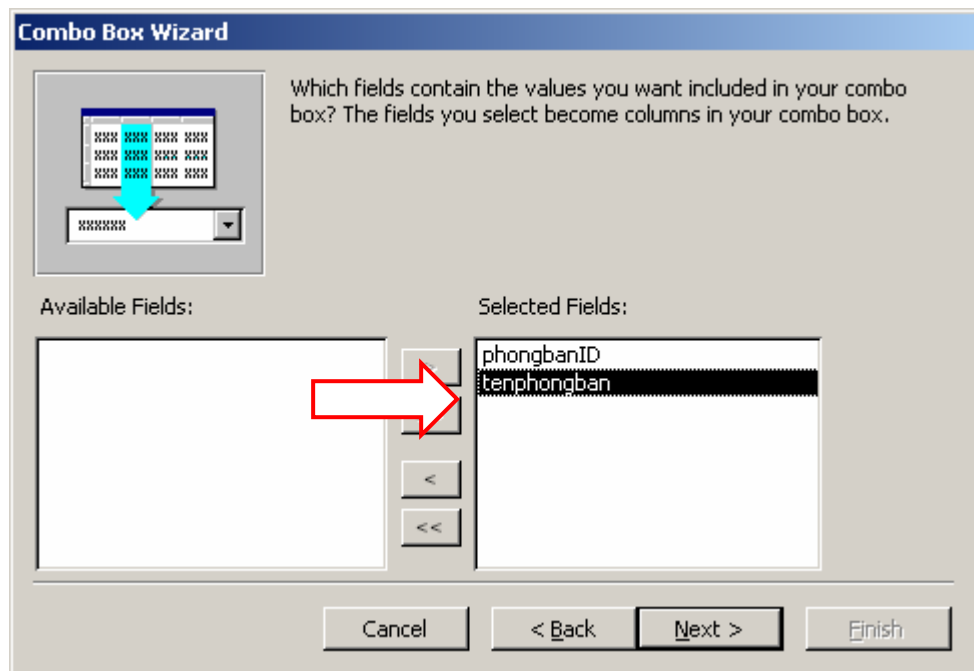
Dùng chuột kéo đối tượng **Combobox** từ thanh công cụ **ToolBox** thả lên form mẹ, hộp thoại sau xuất hiện:



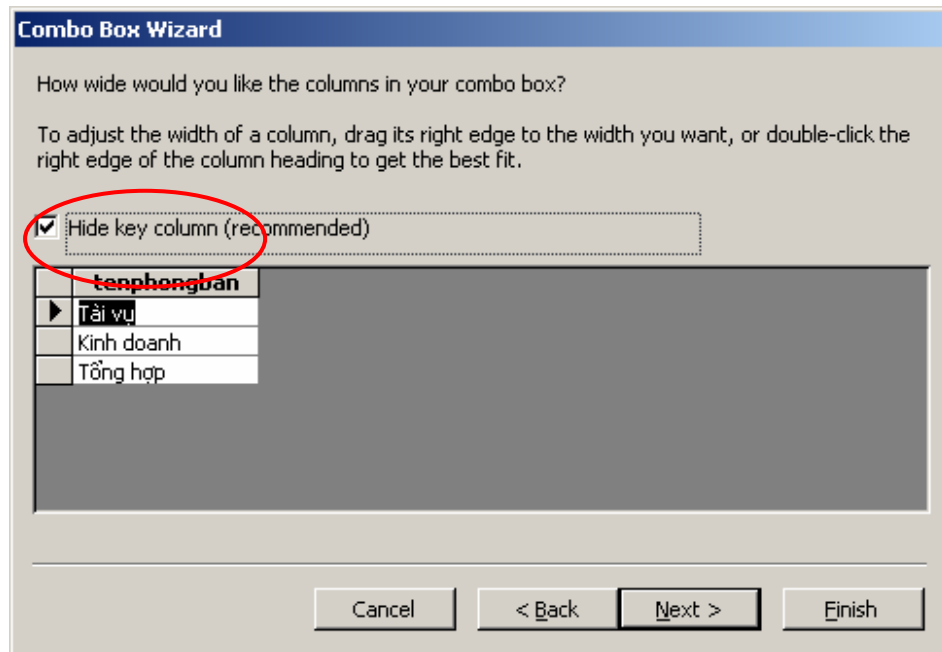
Hãy chọn mục **I want the combo box look up the value in a table or query**. Nhấn **Next** để tiếp tục, hộp thoại sau xuất hiện:



Chọn bảng từ danh sách **Table\Queries** – nơi có chứa danh sách phòng ban cần đưa vào hộp chọn Combobox. Chọn PHONGBAN, nhấn **Next**, hộp thoại tiếp theo xuất hiện:



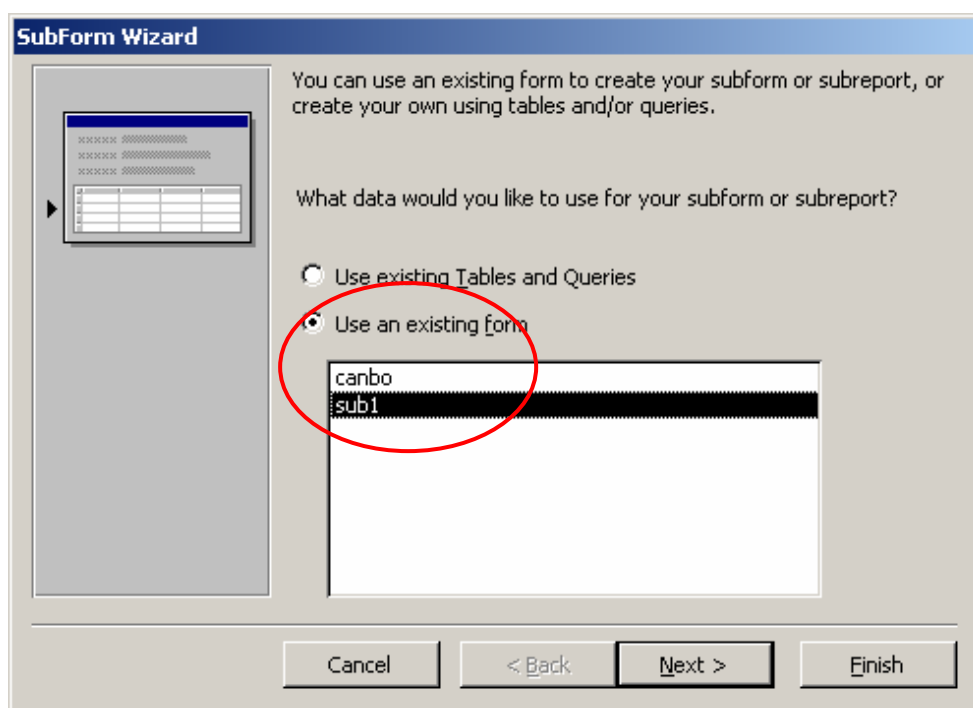
Chọn các trường sẽ hiển thị giá trị lên Combo box từ danh sách **Available Fields:** → **Selected Fields:** (chọn 2 trường *phongbanID* và *Tenphan*). Nhấn **Next** để tiếp tục:



Nên chọn mục **Hide key column (recommended)** để ẩn đi khoá làm cho danh sách thả xuống hộp thả gọn gàng hơn. Cuối cùng nhấn **Finish** để kết thúc việc tạo Combo box.

Thiết lập thuộc tính **Name** cho **Combo box** trùng tên với tham số đã thiết lập ở form con là *cboPhongban*.

- Sử dụng đối tượng **Subform\Subreport** trên thanh công cụ **ToolBox** để đưa form con lên form mẹ. Sau khi đưa một đối tượng **Subform\Subreport** lên form, một hộp thoại xuất hiện;



Hãy chọn tên form con cần đưa lên ở danh sách **Use an existing form**.  
Chọn xong nhấn **Finish**.

- Ghi lại form với một tên gọi (ví dụ: *frmTracuuDanhSachCanBo*) và đóng lại, kết thúc yêu cầu cơ bản bài toán.



### Chú ý

1. Giá trị thuộc tính Name của hộp Combobox trên form mẹ phải trùng với tham số đã đặt ở form con. Nếu chưa trùng nhau phải sửa lại cho khớp; hoặc sửa Name của Combo cho khớp với tham số form con hoặc thao tác ngược lại mở form con sửa tham số cho khớp Name của ô Combobox.

## Bài tập

### Cơ sở Quản lý lương cán bộ

**Bài số 1:** Tạo form cho phép xem danh sách cán bộ từng phòng ban như sau:

Chọn phòng ban:

**Danh sách cán bộ:**

	Họ tên	Ngày sinh	Giới tính	Chức vụ
▶	Nguyễn Đức Ngô	12/04/1976	Nam	Giám đốc
	Phan Quang Sáng	03/09/1976	Nữ	Nhân viên
	Nguyễn Văn Hoà		Nữ	Nhân viên
	Phan Hoài Linh		Nam	Nhân viên
	Phan Anh Quân	23/07/1978	Nữ	Trưởng phòng
*				

Record:  of 5

Mỗi khi chọn một phòng ban từ hộp thả

Chọn phòng ban:

danh sách cán bộ phòng ban đó được hiển thị lên **Subform**.

**Bài số 2:** Tạo form cho phép tìm kiếm cán bộ theo tên như sau:

Nhập tên cán bộ  
 muốn tìm:

**Danh sách cán bộ tìm thấy:**

	hoten	ngaysinh	tenchucvu	tenphongban
▶	Nguyễn Đức Ngô	12/04/1976	Giám đốc	Kinh doanh
	Trần Văn Ngô		Nhân viên	Tổng hợp
*				

Record:  of 2

Sau khi gõ một tên (có thể là đệm + Tên hoặc đầy đủ họ và tên), danh sách các kết quả tìm thấy sẽ được liệt kê lên Subform.

### Trên CSDL Quản lý bán hàng hãy:

**Bài số 3:** Thiết kế form lập hoá đơn bán hàng theo mẫu:

hoadonID:  Ngày bán:

Tên khách:

Địa chỉ:

E-mail:

**Chi tiết hàng bán:**

	hangID	donvi	dongia	soluong	thanhtien:
	Floppy disk	chiếc	8	3	24
	Hard disk	ổ	67	9	603
▶	CPU P4 2.4	con	133	2	266
*					

Record:  of 3

Tổng tiền:

Yêu cầu chi tiết:

- Các nút Tiến, Lùi để định vị hoá đơn cần làm việc;

- Nút Thêm mới để bắt đầu tạo mới một hoá đơn;
- Nút Xoá HĐ để xoá hoá đơn hiện tại. Yêu cầu phải có xác nhận trước khi xoá;
- Nút In HĐ để in chi tiết hoá đơn bán hàng ra report (chức năng này sẽ được hoàn thiện sau khi học xong Chương Report)

**Bài số 4:** Thiết kế form tổng hợp thông tin bán hàng như sau:

**TỔNG HỢP THÔNG TIN BÁN HÀNG**

Từ ngày: 1/1/2004 Đến ngày: 12/1/2004

**Chi tiết hàng đã bán:**

Tên hàng:	Đơn vị:	Tổng số bán:
CPU P4 2.4	con	15
Floppy disk	chiếc	11
Hard disk	ổ	9
RAM 256	thanh	15

Record: 1 of 4

Sau khi nhập xong giá trị từ ngày và đến ngày, thông tin chi tiết về các mặt hàng bán ra trong phạm vi ngày đó được tổng hợp theo như giao diện trên.

**Bài số 5:** Thiết kế form tổng hợp thông tin bán hàng cho một khách hàng nào đó trong một khoảng ngày nào đó như sau:



**TỔNG HỢP THÔNG TIN BÁN HÀNG**

Chọn khách hàng: Nguyễn Đức Ngô

Từ ngày: 1/1/2004 Đến ngày: 12/1/2004

**Chi tiết hàng đã bán:**

Tên hàng:	Đơn vị:	Tổng số bán:
CPU P4 2.4	con	5
Floppy disk	chiếc	3
Hard disk	ổ	9
RAM 256	thanh	15

Record: 1 of 4

Yêu cầu: Sau khi chọn một khách hàng, nhập vào khoảng ngày cần tổng hợp. Thông tin tổng hợp về từng mặt hàng của khách đó mua sẽ được liệt kê.

**Bài số 6:** Tạo form tổng hợp thông tin hàng đã bán chi tiết theo từng ngày như sau:

**THEO DÕI CHI TIẾT HÀNG BÁN**

Từ ngày: 1/10/2004 Đến ngày: 30/10/2004

**Chi tiết hàng bán:**

tenhang:	10/10/2004:	12/10/2004:	24/10/2004:
CPU P4 2.4	3	12	
Floppy disk		11	
Hard disk		9	
RAM 256	12		3

Record: 1 of 4

**Bài số 7:** Tạo form theo dõi thông tin hoá đơn bán hàng của một khách hàng nào đó:

Chọn khách: Nguyễn Đức Ngô

**Danh sách hoá đơn đã mua:**

	hoadonID:	ngayban:	Tongtien:
▶	1	12/10/2004	893
	2	24/10/2004	114
	3	10/10/2004	855
	4	12/10/2004	1394

Record: 1 of 4

Sau khi chọn tên một khách, thông tin về các hoá đơn mua hàng được hiển thị.

# CHƯƠNG 4

## THIẾT KẾ BÁO CÁO

Report là công cụ dùng thiết kế in ấn dữ liệu hữu hiệu trong Access. Hầu như tất cả những gì bạn muốn in và in theo bố cục như thế nào? Report đều có thể đáp ứng!

Chương này sẽ trình bày từ những khái niệm căn bản về thiết kế in ấn, về Report của Access đến tiếp cận những kỹ thuật in ấn phức tạp như: report có tham số, sub-report, ..

Nội dung cụ thể bao gồm:

- Các khái niệm về Report;
- Sử dụng report wizard;
- Sử dụng report design view;
- Kỹ thuật sub-report;
- Tham số cho report.

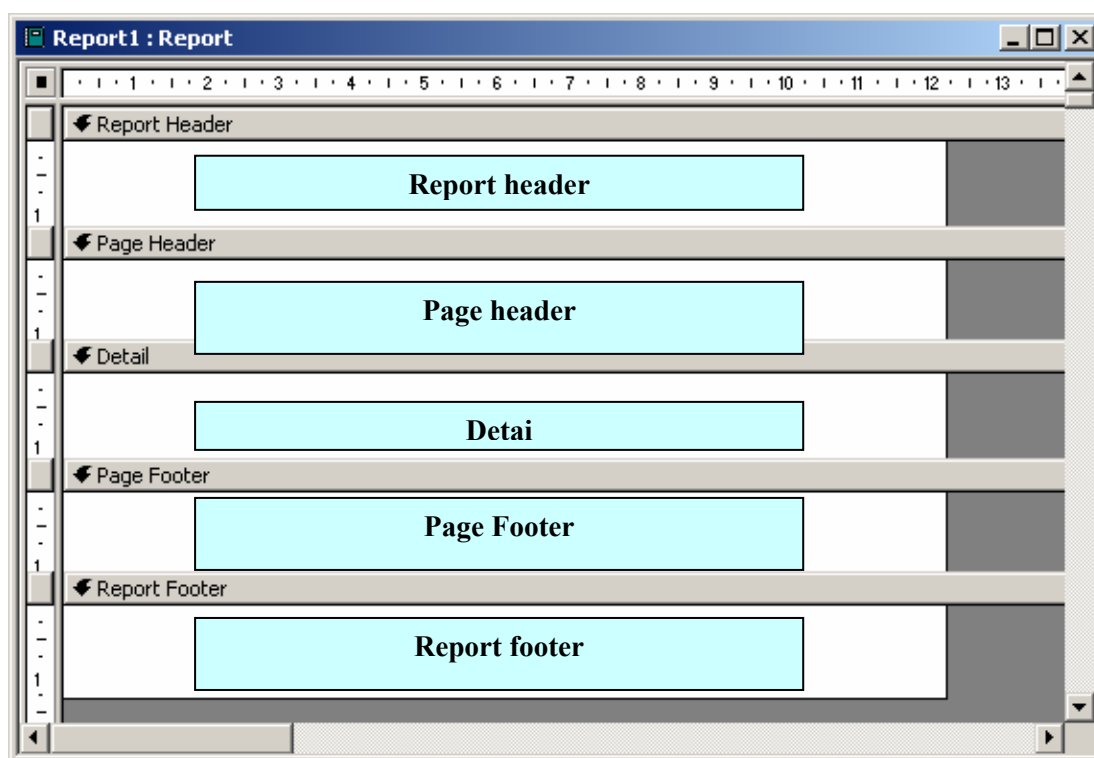
Kết thúc chương, học viên hoàn toàn có thể thiết kế được những mẫu biểu in ấn đơn giản đến những biểu phức tạp phục vụ nhu cầu in ấn đa dạng trong các bài toán thực tế từ CSDL.

## 1. Các khái niệm về Report

Muốn in các báo cáo trong Access bạn có thể sử dụng Report- một công cụ in ấn rất mạnh. Ứng với mỗi mẫu báo cáo có thể thiết kế các thông tin lên một report. Mỗi khi report hiển thị kết quả (preview) là lúc có thể in được nội dung báo cáo ra giấy.

### 1.1 Cấu trúc Report

Cấu trúc một report thông thường gồm 5 phần:



- **Page Header**

Là phần đầu tiên của một trang báo cáo. Giống như khái niệm Page header trong Word và Excel. Phần này có thể có hoặc không có thông tin tùy vào người thiết kế.

- **Page Footer**

Là phần cuối cùng của mỗi trang báo cáo. Giống như khái niệm Page footer trong Word và Excel. Phần này có thể có hoặc không có thông tin tùy vào người thiết kế.

- **Detail**

Là phần thân của report – nơi hiển thị giá trị các bản ghi sẽ in ra. Phần này có thể bị thay đổi, phụ thuộc vào nguồn dữ liệu tại thời điểm sẽ in ra của report. Ví dụ: Danh sách học sinh lớp 10A1 là 45 khi đó Detail report sẽ in ra 45 dòng; nếu chuyển sang in danh sách học sinh lớp 10A2 có 48 học sinh, lúc này Detail report sẽ in ra 48 dòng (tương ứng với số bản ghi của nguồn dữ liệu sẽ in ra).

- **Report Header**

Là phần tiêu đề của trang đầu tiên Report, nằm tiếp theo phần Page header và nằm trên phần Detail. Mỗi Report sẽ chỉ có nhiều nhất 1 Report header. Phần này có thể có hoặc không có thông tin tùy vào người thiết kế.

- **Report Footer**

Là phần tiêu đề cuối của trang cuối cùng Report, nằm tiếp theo phần Detail và phía trước phần Page Footer. Mỗi Report sẽ chỉ có nhiều nhất 1 Report footer. Phần này có thể có hoặc không có thông tin tùy vào người thiết kế.

## **1.2 Môi trường làm việc**

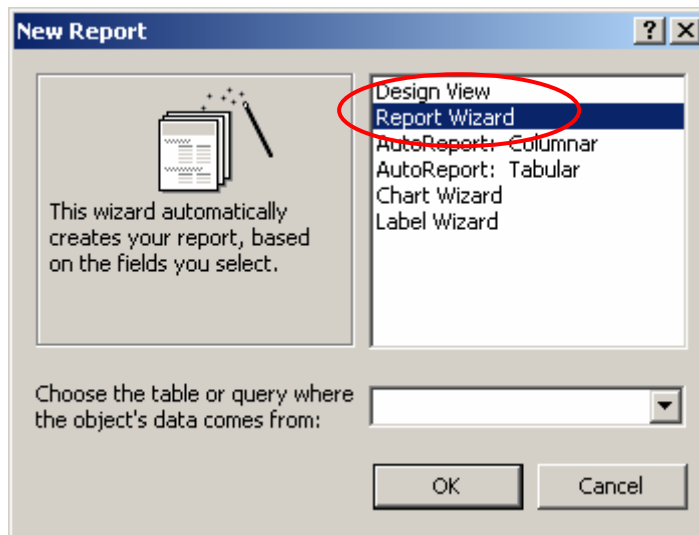
Làm việc với **Report** gần giống làm việc với **Form**. Sự khác nhau cơ bản của Report và Form là: Form có thể hiển thị, tra cứu, cập nhật và điều khiển dữ liệu từ CSDL; còn Report chỉ có thể lập báo cáo và in ra, đặc biệt Report không làm thay đổi giá trị dữ liệu trong CSDL.


Thông thường mỗi report sẽ in dữ liệu của một Table hoặc một Query nào đó. Tức là phải có một nguồn dữ liệu cần in cụ thể (trường hợp đặc biệt Report không có nguồn dữ liệu sẽ nói đến phần cuối)

## 2. Sử dụng Report wizard

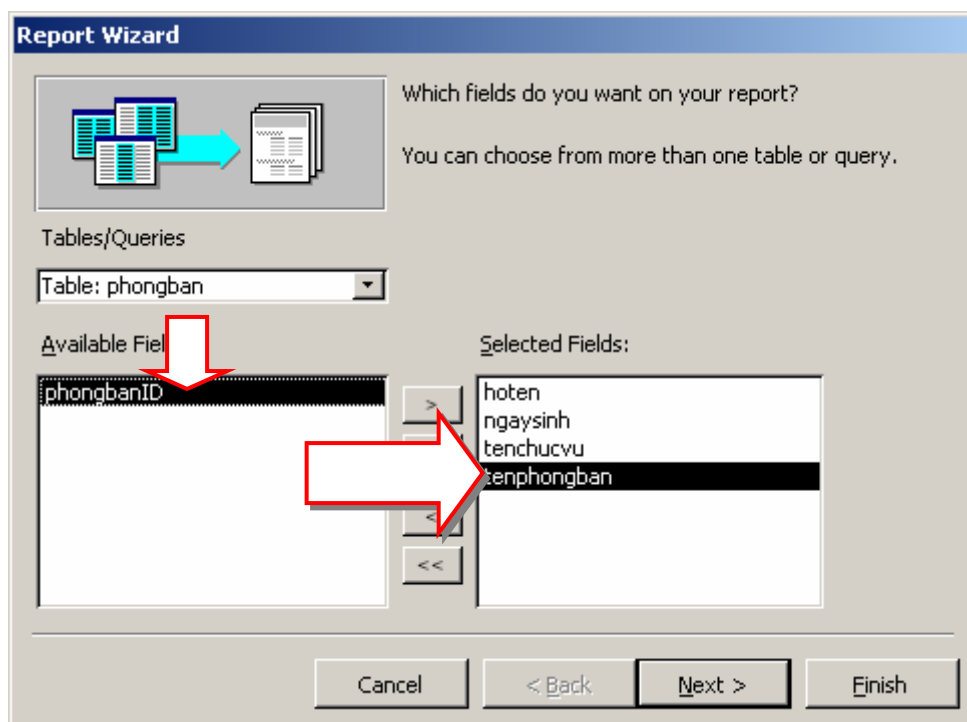
Giống như Form wizard, Report wizard là một công cụ rất đơn giản, dễ dùng để tạo nhanh một Report. Dưới đây hướng dẫn từng bước dùng Report wizard để tạo một report in ra danh sách cán bộ từ CSDL Quản lý lương bao gồm các thông tin: *Hoten, ngaysinh, tenchucvu, Tenphongban*.

**Bước 1:** Ở thẻ Reports, nhấn **New**, chọn *Report wizard*, nhấn **OK**:



Hoặc nhấn đúp chuột lên biểu tượng  **Create report by using wizard**

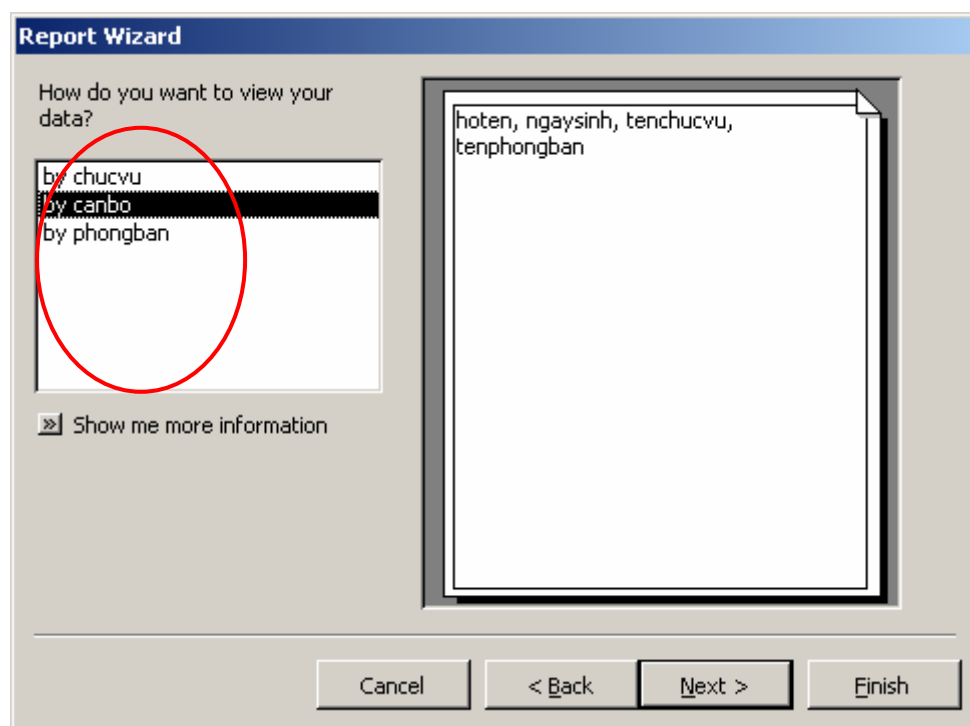
**Bước 2:** Chọn dữ liệu cần in trên hộp thoại sau:



Để chọn dữ liệu cần in lên report, có 2 thao tác phải làm trên hộp thoại này:

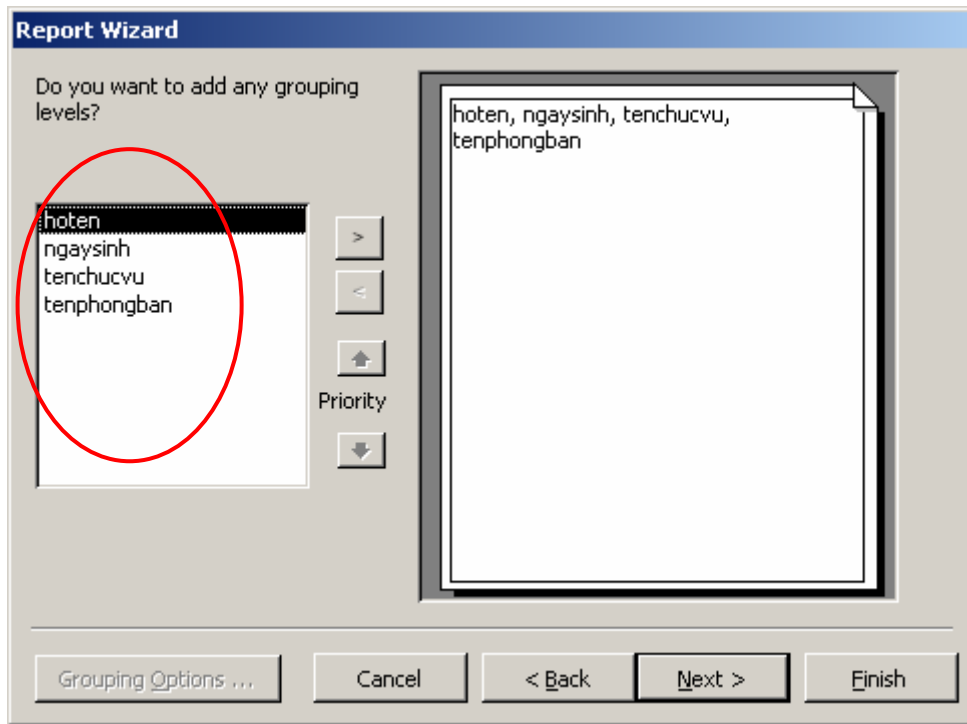
- Chọn bảng hoặc query nơi có chứa trường dữ liệu cần in ra ở hộp **Table/Queries**;
- Sử dụng các nút lệnh >, >>, <, << để đưa các trường cần in từ danh sách **Available Fields**: (danh sách các trường có thể in) sang danh sách **Selected Fields**: (danh sách các trường sẽ được in ra report). Hãy lần lượt thực hiện chọn các trường *hoten*, *ngaysinh* (từ bảng *CANBO*), trường *tenchucvu* (bảng *CHUCVU*) và trường *tenphongban* (bảng *PHONGBAN*). Chọn xong nhấn **Next**:

**Bước 3:** Chọn kiểu cách hiển thị dữ liệu trên report:



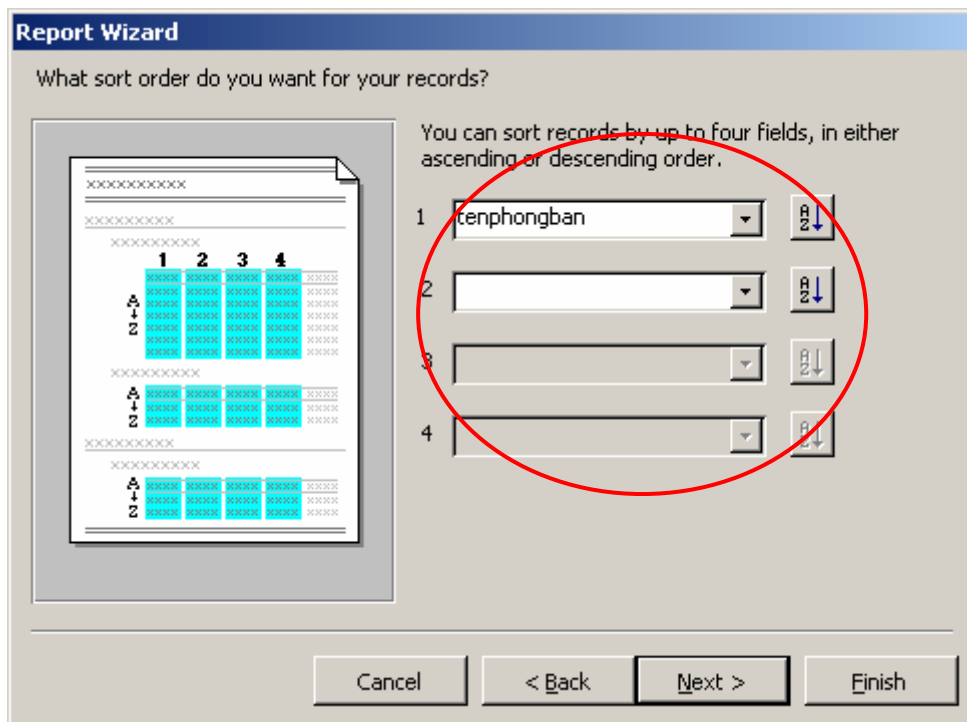
Bạn muốn in dữ liệu theo kiểu nào: Hãy chọn kiểu cần in từ danh sách bên trái hộp thoại. Trong yêu cầu này có thể có 3 kiểu hiển thị dữ liệu trên report (vì dữ liệu được chọn ra từ 3 bảng khác nhau). Vì muốn in một danh sách cán bộ nên chọn kiểu *by CANBO* (có thể xem kiểu hiển thị bên phải hộp thoại). Nhấn **Next** để tiếp tục:

**Bước 4:** Thiết lập các nhóm dữ liệu cần thiết:



Muốn hiển thị dữ liệu theo nhóm giá trị của trường nào, hãy Add trường đó từ danh sách bên trái hộp thoại sang hộp preview bên phải hộp thoại. Trong bài này chỉ cần hiển thị một danh sách chung nên không cần thiết lập nhóm ở bước này. Nhấn **Next** để tiếp tục:

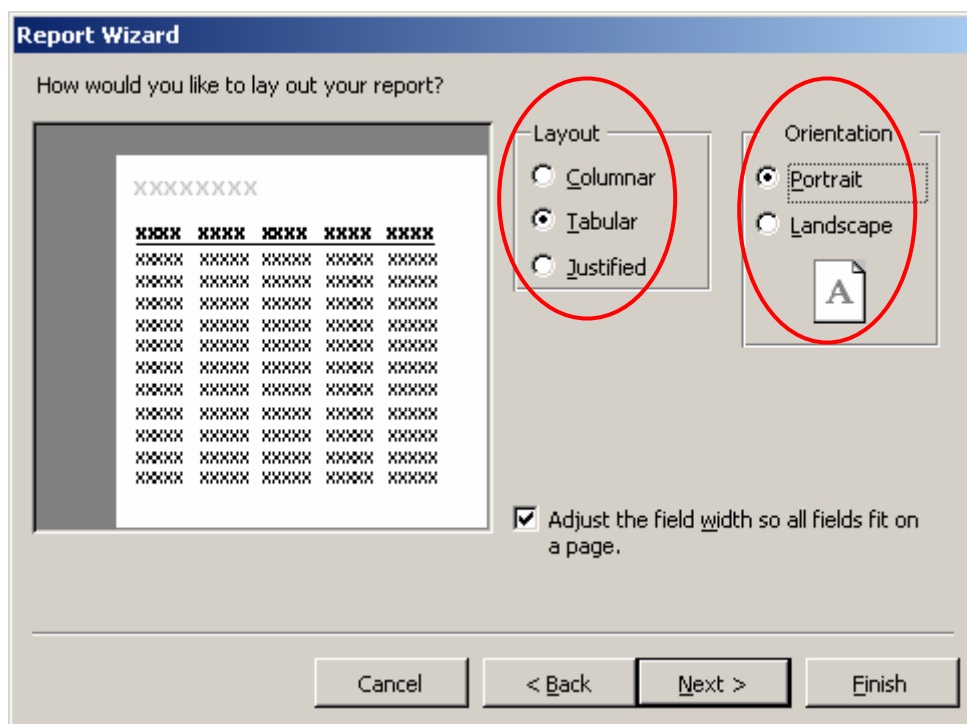
**Bước 5:** Thiết lập sắp xếp dữ liệu trên báo cáo:





Report wizard cho phép tối đa 4 mức ưu tiên sắp xếp dữ liệu được đánh số từ 1 đến 4 (hộp thoại trên). Trường nào thiết lập trước, sẽ được ưu tiên sắp xếp trước. Trong trường hợp giá trị trường đó trùng nhau, Access sẽ chuyển đến các mức tiếp theo để sắp xếp. Kiểu sắp xếp (theo chiều tăng hoặc giảm) có thể thiết lập khi nhấn nút ... bên cạnh. Thiết lập xong nhấn **Next** để tiếp tục:

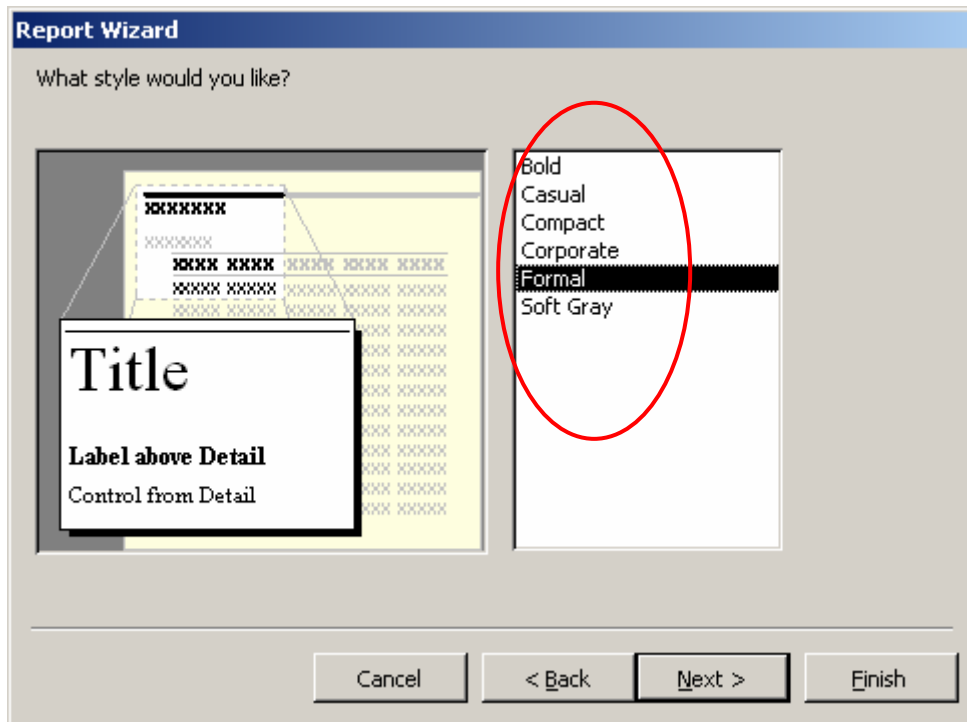
**Bước 6:** Chọn bố cục (Layout) cho Report:



Có 2 thiết lập trong bước này:

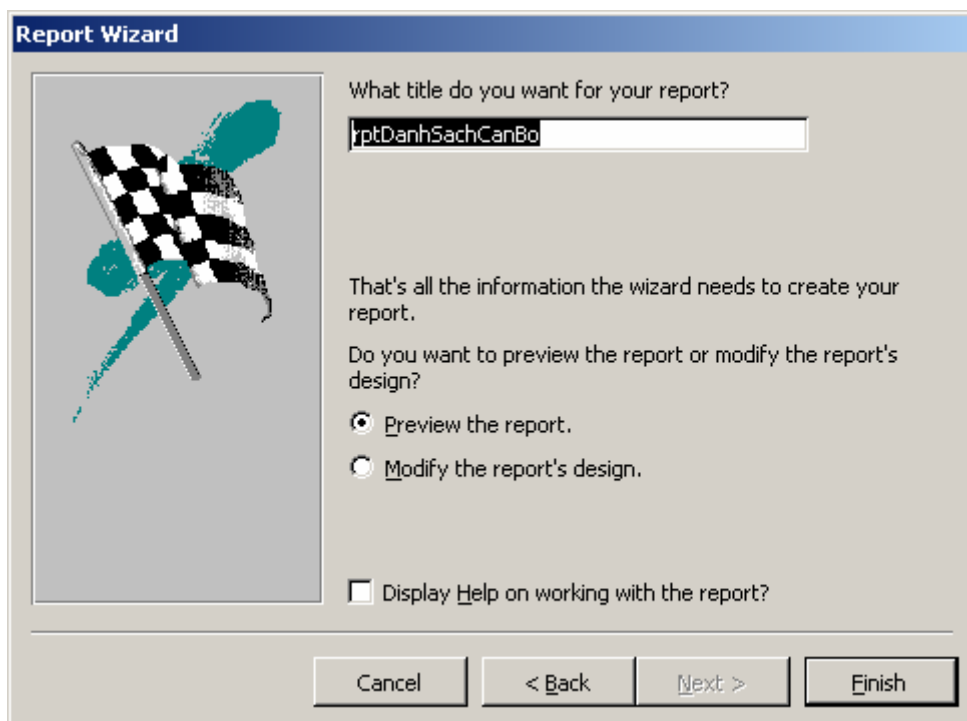
- Chọn bố cục cho report ở hộp **Layout**. Có 3 kiểu Layout: *Columnar*, *Tabular* và *Justified*- hãy chọn một kiểu phù hợp (xem hộp preview bên trái để biết trước kết quả);
- Chọn hướng giấy in ở hộp **Orientation**. Có 2 kiểu hướng in là: *Portrait* – in theo khổ giấy dọc và *Landscape*- in theo khổ giấy ngang;
- Chọn xong nhấn **Next** để tiếp tục:

**Bước 7:** Chọn mẫu định dạng (Style) cho report:



Hãy chọn một mẫu định dạng từ danh sách bên trái hộp thoại. Chọn xong nhấn **Next** để tiếp tục:

**Bước 8:** Hoàn thiện các thông tin cuối cùng:

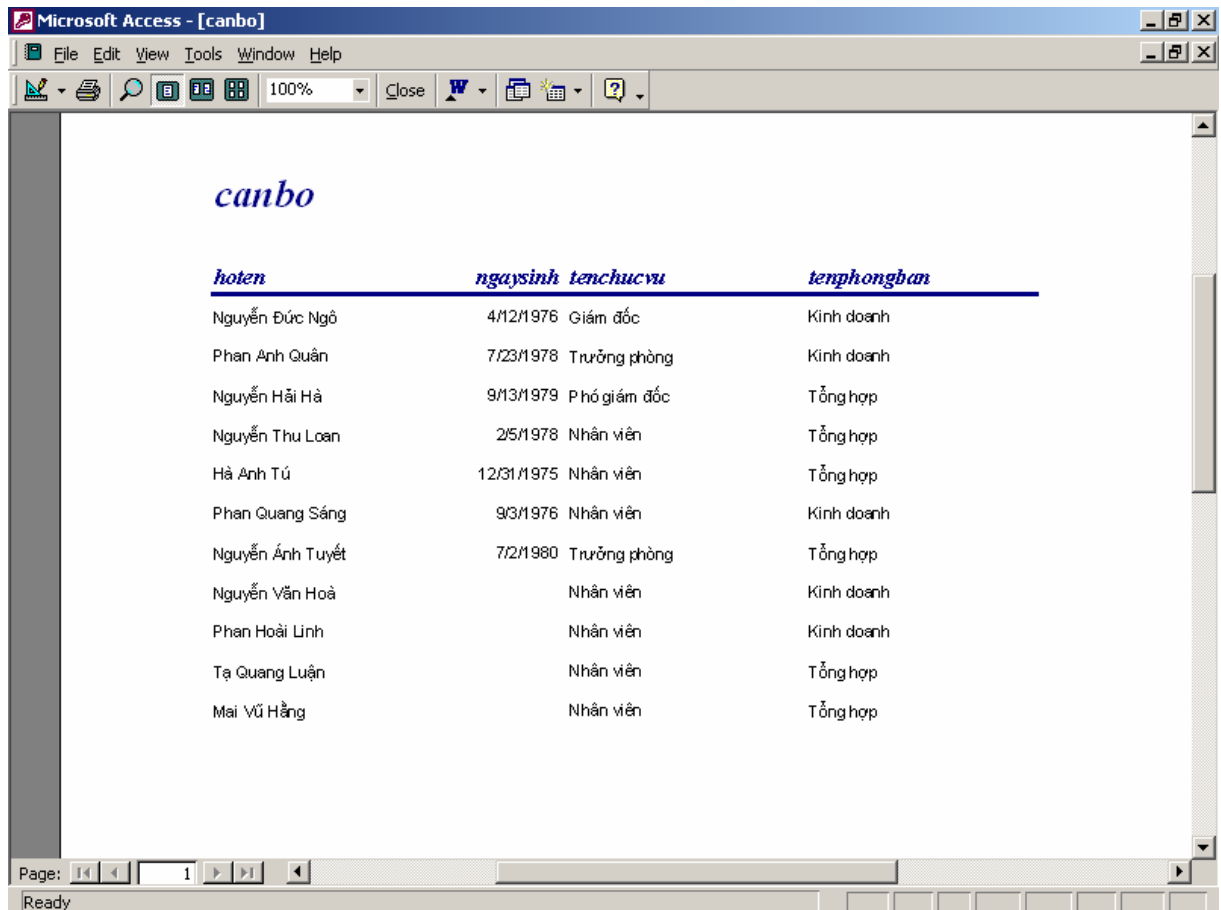


Bao gồm các thông tin sau:

- Gõ vào tiêu đề report cũng như tên report trên hộp **What do you want for your report?**;

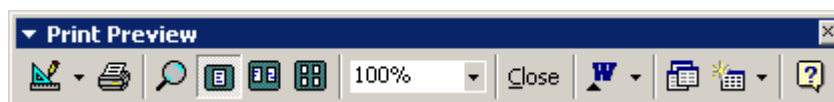
- Chọn **Preview the report** để hiển thị dữ liệu của report ngay sau khi kết thúc; hoặc hiển thị ngay màn hình thiết kế để sửa cấu trúc report khi chọn **Modify the report's design**;
- Cuối cùng nhấn **Finish** để kết thúc các công việc sử dụng report wizard.


Màn hình hiển thị dữ liệu report như sau:




<i>hoten</i>	<i>ngaysinh</i>	<i>tenchucvu</i>	<i>tenphongban</i>
Nguyễn Đức Ngô	4/12/1976	Giám đốc	Kinh doanh
Phan Anh Quân	7/23/1978	Trưởng phòng	Kinh doanh
Nguyễn Hải Hà	9/13/1979	Phó giám đốc	Tổng hợp
Nguyễn Thu Loan	2/5/1978	Nhân viên	Tổng hợp
Hà Anh Tú	12/31/1975	Nhân viên	Tổng hợp
Phan Quang Sáng	9/3/1976	Nhân viên	Kinh doanh
Nguyễn Ánh Tuyết	7/2/1980	Trưởng phòng	Tổng hợp
Nguyễn Văn Hoà		Nhân viên	Kinh doanh
Phan Hoài Linh		Nhân viên	Kinh doanh
Tạ Quang Luận		Nhân viên	Tổng hợp
Mai Vũ Hằng		Nhân viên	Tổng hợp






Từ màn hình này có thể thực hiện rất nhiều các thao tác thông qua thanh công cụ Print Preview như sau:



Nút *Print*  Để in nội dung report ra máy in;

Nút *Design*  Để mở report ra chế độ thiết kế;

Nút *One page*  Để hiển thị report ra màn hình trong từng trang báo cáo;

- Nút *Two page*  Để hiển thị 2 trang báo cáo một trên màn hình;
- Nút *Multi Page*  Để chọn nhiều trang báo cáo có thể hiển thị trên màn hình;
- Nút *Zoom*  Để phóng to, thu nhỏ nội dung report;
- Nút *Office Link*  Để kết xuất (Export) thông tin trên report ra các loại định dạng khác của MS Office như Word, Excel, HTML.
- Nút *Close*  Để đóng màn hình preview report.

Toàn bộ 8 bước sử dụng report wizard liệt kê ở trên, không phải lúc nào cũng cần thực hiện đầy đủ. Với những report đơn giản như trên, chỉ cần thực hiện Bước 1; Bước 2 đã có thể nhấn **Finish** để kết thúc. Tuy nhiên tùy vào yêu cầu in báo cáo mà muốn dừng lại ở bước nào để thiết lập các thông tin cho phù hợp hãy chuyển nhanh đến bước đó.

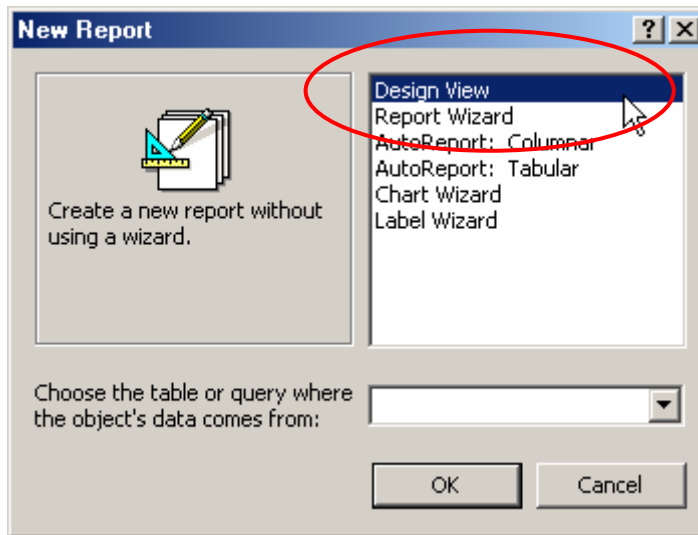
### 3. Thiết kế report

Phần trước đã trình bày các bước dùng Report wizard để có thể thiết kế các mẫu báo cáo in ấn dữ liệu từ CSDL. Đó là cách làm rất dễ dàng và nhanh chóng. Tuy nhiên, các mẫu báo cáo được sinh ra chỉ giới hạn theo một số mẫu mã máy tính đã cung cấp sẵn, không thể in ra được những biểu báo cáo phức tạp theo như những yêu cầu đa dạng của người sử dụng. Report wizard chỉ có thể tạo ra các report in dữ liệu từ các bảng hoặc queries theo cấu trúc đơn giản; trong những trường hợp yêu cầu cấu trúc report phức tạp, hoặc phải in dữ liệu không phải hoàn toàn chỉ từ các bảng và queries hoặc in ấn báo cáo theo các tiêu chí, tham số động nào đó thì không thể đáp ứng được. Trong trường hợp này phải dùng đến Report design view.

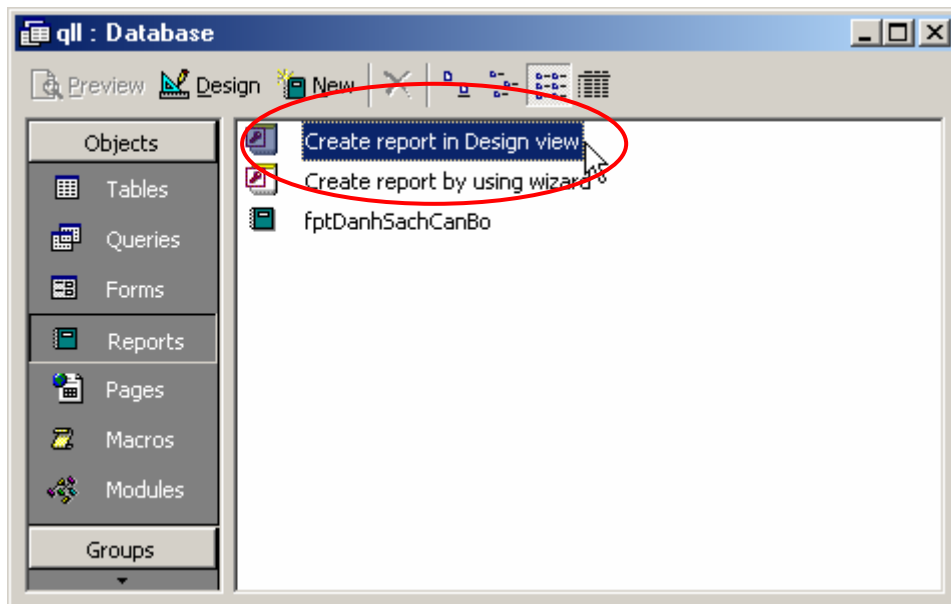
Trong chừng mực nào đó, cách làm việc với Report design view gần giống cách làm việc với Form design view. Sau đây là từng bước hướng dẫn sử dụng Report design view để thiết report theo yêu cầu như trên:

**Bước 1:** Khởi động Report design view:

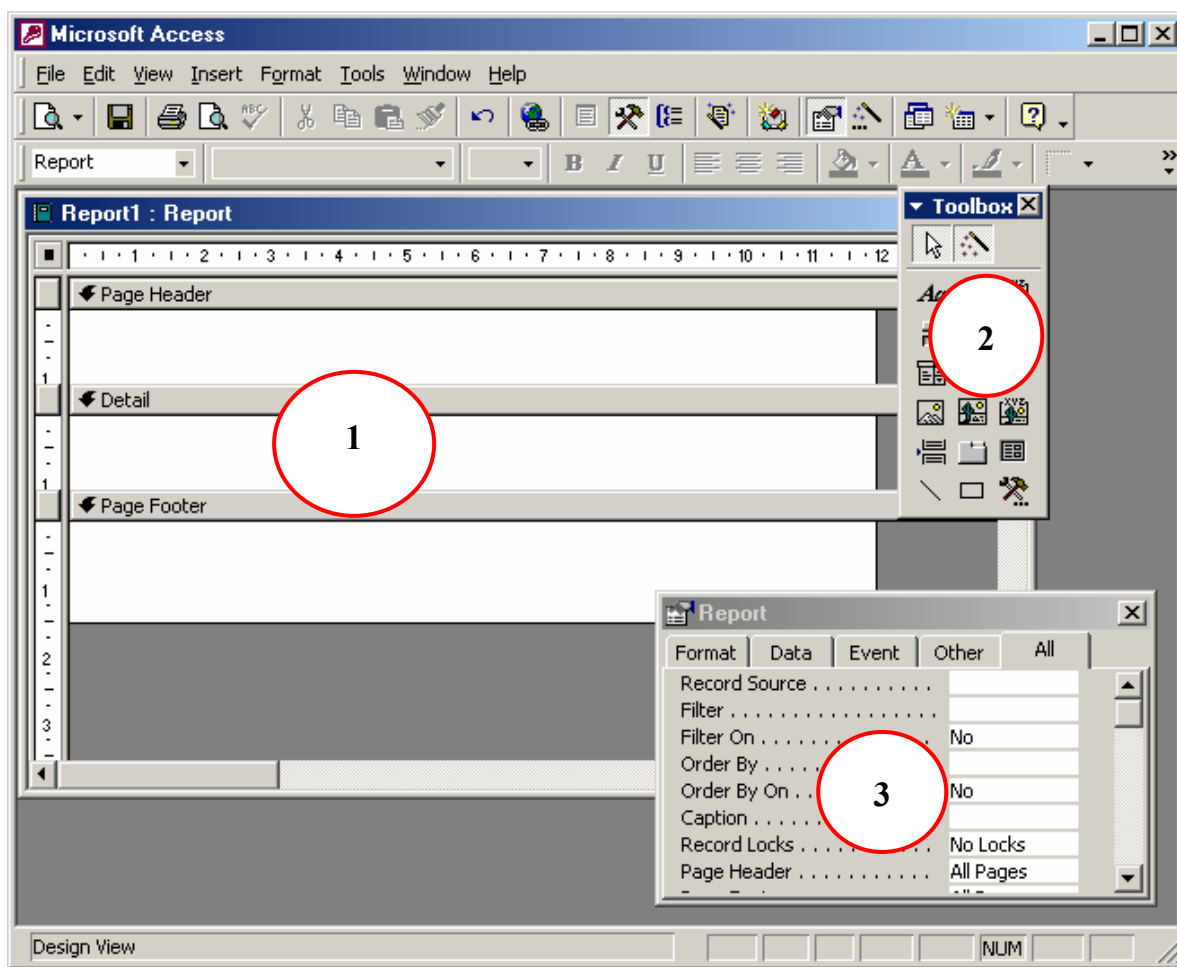
Từ thẻ Report nhấn nút **New**, chọn **Design view**, nhấn **OK**



Hoặc nhấn đúp chuột lên biểu tượng *Create report in Design view* trên cửa sổ dự án:



Môi trường làm việc với Report design view xuất hiện:



Có 3 phần quan trọng trên môi trường làm việc này:

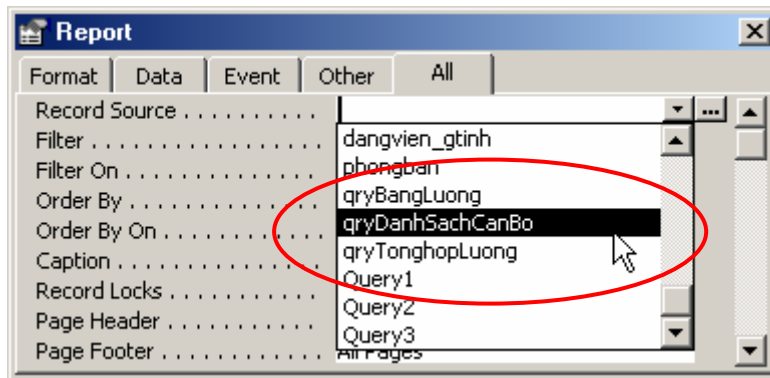
- (1) Cửa sổ thiết kế Report (hình trên tiêu đề cửa sổ này là *Report1: Report*)- nơi để thiết kế nội dung cần in ấn. Nội dung được thiết kế trên cửa sổ này là các đối tượng từ thanh công cụ Toolbox sau khi đã được thiết lập các thuộc tính phù hợp với mục đích;
- (2) Thanh công cụ Toolbox- nơi chứa những đối tượng giúp đưa các thông tin cần thiết lên report. Chức năng và cách làm việc trên thanh công cụ này gần giống với làm việc trên thanh công cụ Toolbox của Form design view;
- (3) Cửa sổ Properties – nơi thiết lập các thuộc tính phù hợp cho các đối tượng trên màn hình thiết kế report.

Chúng ta sẽ được tìm hiểu kỹ hơn các thành phần trên trong các bước tiếp theo khi làm việc cụ thể với chúng.

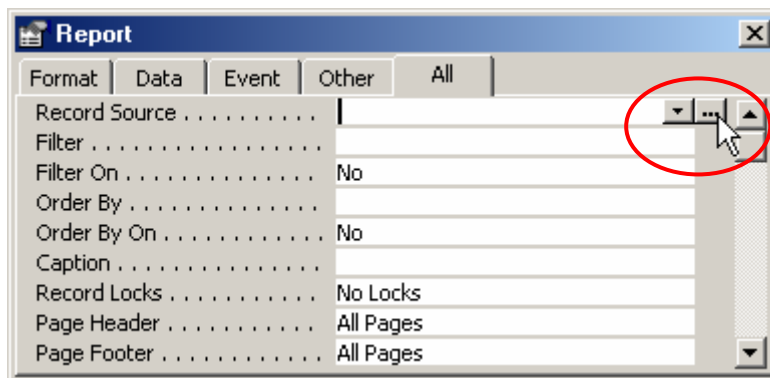
**Bước 2:** Xây dựng nguồn dữ liệu để in ấn cho report.

*Report sẽ in những thông tin gì? Phạm vi như thế nào?* Là các câu hỏi phải được trả lời ở bước này bằng cách thiết lập thuộc tính **Record Source** cho report.

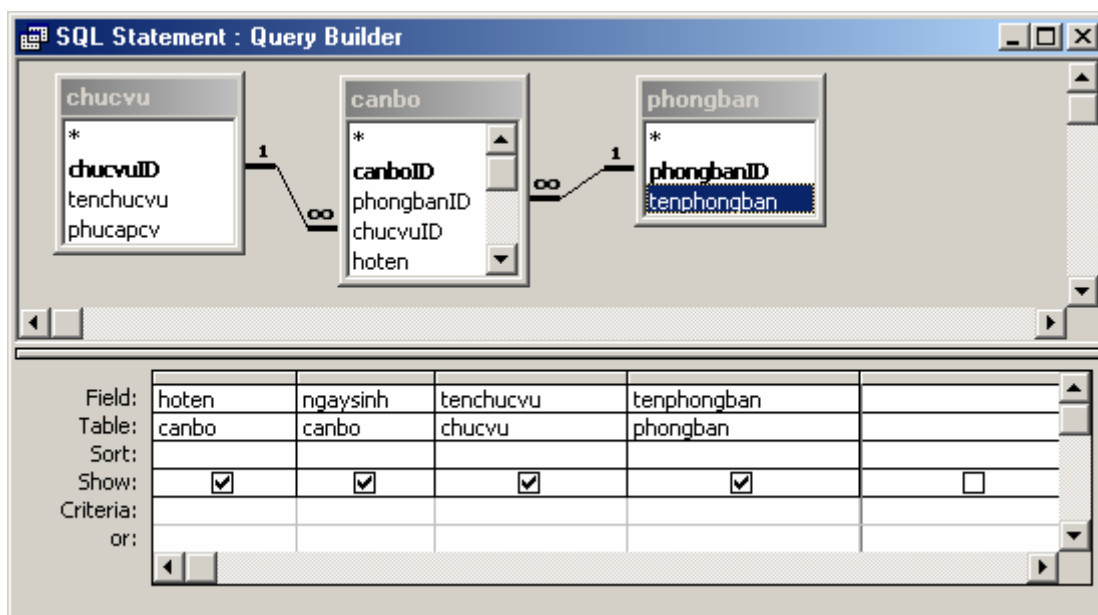
Thông thường, report sẽ in dữ liệu từ một hay nhiều bảng hoặc từ kết quả của một query, tổng quát là từ một query. Query đó có thể được xây dựng sẵn từ danh sách các Queries của dự án (chứa trên thẻ Queries)- khi đó bước này chỉ chọn query cần in tại thuộc tính *Record Source* của report:



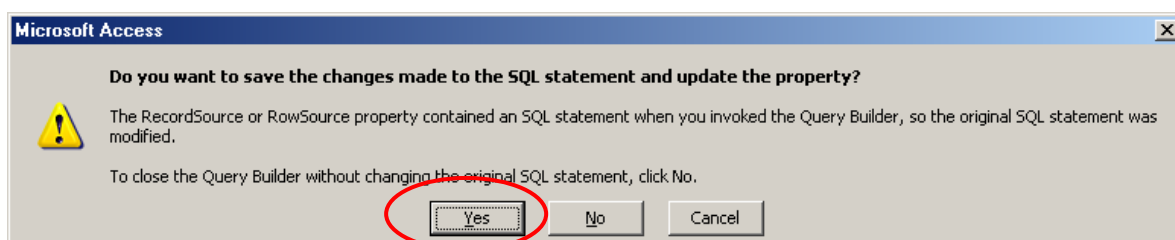
Tuy nhiên query cần in có thể được tạo ra trong chính bản thân report (không hiển thị tên query trên thẻ Queries)- điều này nên làm vì như vậy sẽ đảm bảo sự chắc chắn của report. Khi đó hãy nhấn chuột lên nút ... của thuộc tính *Record Source*:



Một cửa sổ thiết kế query làm nguồn dữ liệu cho report xuất hiện:



Hãy thiết lập các thông tin cần in ra report trên query này. Thiết lập xong nhấn nút đóng cửa sổ thiết kế query (hoặc nhấn phím nóng *Ctrl+W*) và chọn **Yes** trong hộp thoại sau:



**Bước 3:** Đưa các thông tin lên cửa sổ thiết kế report:

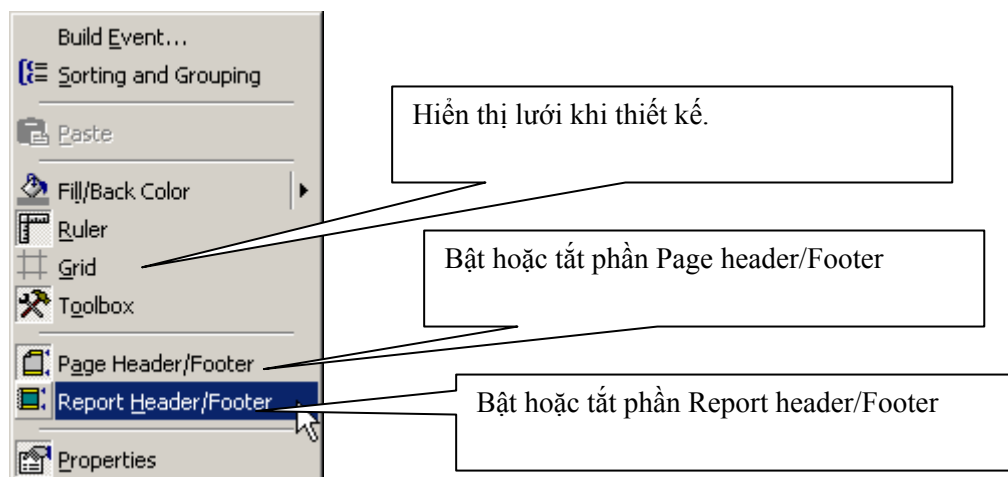
- (1) Có rất nhiều loại thông tin phải đưa lên report. Mỗi khi có ý định đưa một thông tin lên, bạn phải trả lời được 2 câu hỏi:

***Đó là thông tin gì?*** là tiêu đề “*DANH SÁCH CÁN BỘ*”.

***Sẽ đặt thông tin đó lên phần nào của report?*** Đặt lên phần *Page Header*!


- (2) Theo ngầm định, cửa sổ thiết kế report chỉ xuất hiện 3 phần: Page header, Page footer và Detail. Nếu report đang thiết kế yêu cầu có cả Report header và Report footer, hãy hiển thị chúng bằng cách: nhấn phải chuột lên cửa sổ thiết kế report và chọn:






Những phần nào không dùng đến khi thiết kế có thể tắt đi hoặc dùng chuột thu lại diện tích phần đó.

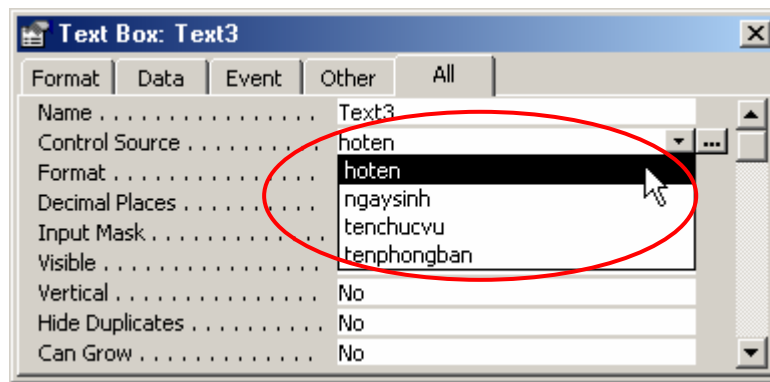
### (3) Sử dụng công cụ **Label**

- Dùng chuột nhấp nút Label  trên thanh công cụ Toolbox thả lên vị trí cần đặt vào trên Report;
- Gõ vào nội dung tiêu đề cần in ra report;
- Có thể sử dụng thanh công cụ Formatting để định dạng hộp dữ liệu này.

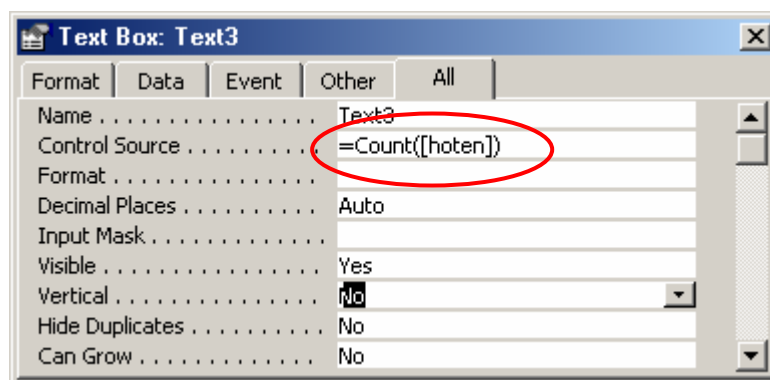
### (4) Sử dụng công cụ **Text box**

Text box là ô dùng hiển thị dữ liệu của một trường dữ liệu (Field) nào đó, hiển thị dữ liệu của một biểu thức (ví dụ tính Tổng tiền chẳng hạn). Cách sử dụng Textbox như sau:

- Dùng chuột nhấp nút Textbox  trên thanh công cụ Toolbox thả lên vị trí cần đặt vào trên Report;
- Gõ vào nguồn dữ liệu sẽ hiển thị lên Textbox ở thuộc tính **Control Source**. Giá trị thuộc tính này có thể:
  - + Hiển thị giá trị một trường: Khi đó hãy chọn trường muốn đưa thông tin vào (hình dưới):



+ Hoặc hiển thị kết quả một biểu thức. Khi đó gõ biểu thức lên thuộc tính này bắt đầu một dấu bằng “=”. Ví dụ:




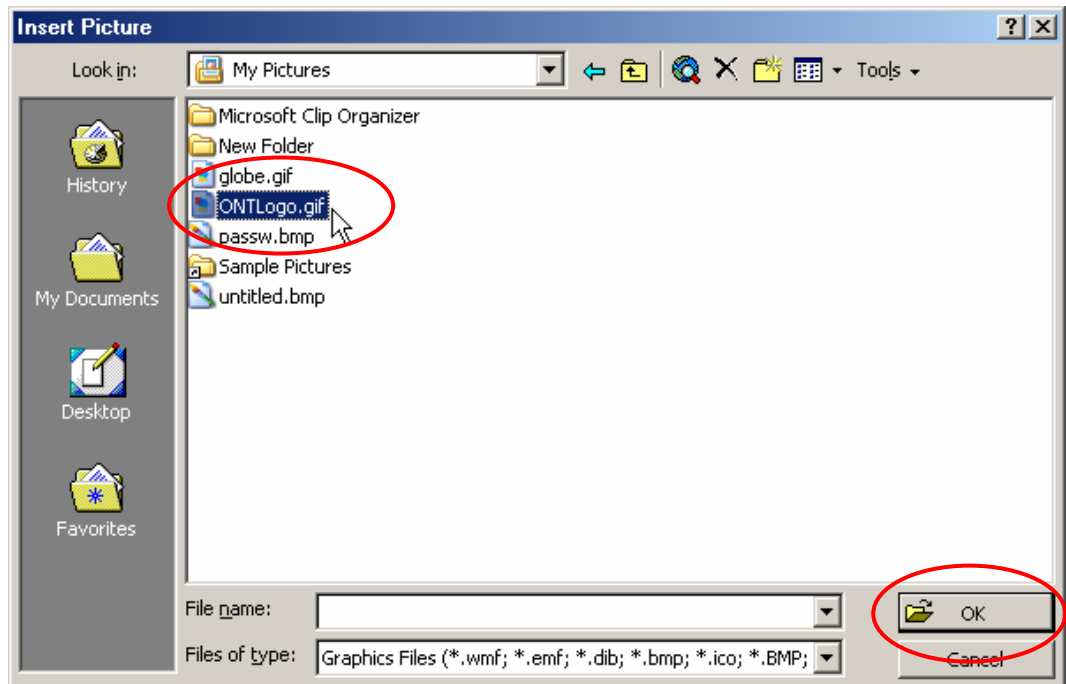
- Có thể sử dụng thanh công cụ Formatting để định dạng hộp dữ liệu này.

#### (5) Sử dụng cửa sổ **Field List**

Danh sách các trường dữ liệu có thể in ra được report thể hiện ở cửa sổ **Field List** (nếu chưa thấy hiển thị mở thực đơn **View | Field List**). Muốn in ra giá trị của trường nào lên vị trí nào của report có thể dùng đối tượng Textbox như giới thiệu ở trên hoặc có thể dùng cửa sổ Field List này bằng cách: dùng chuột kéo trường cần in ra từ cửa sổ Field List thả lên vị trí cần in trên cửa sổ thiết kế report (nên làm theo cách này thay vì dùng Textbox).


#### (6) Sử dụng công cụ **Image**

Công cụ Image  trên thanh công cụ Toolbox giúp đưa ảnh từ các tệp tin ảnh in ra report. Sau khi dùng chuột nhấp nút nút Image từ thanh công cụ đặt lên report, một hộp thoại xuất hiện cho phép tìm đến tệp tin ảnh cần đưa lên report:



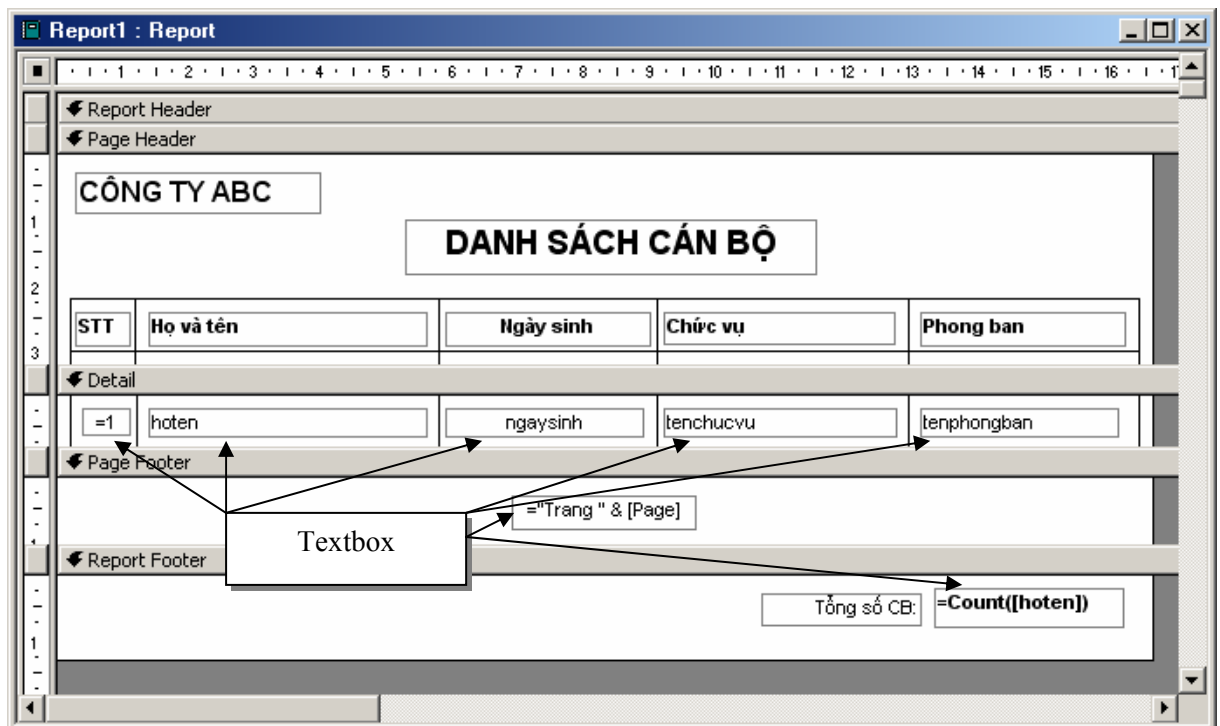
Hãy tìm và chọn tệp ảnh, chọn xong nhấn **OK** để hoàn tất công việc.

#### (7) Sử dụng công cụ **Line**

Công cụ Line  dùng để vẽ các đường thẳng lên Report. Rất hữu hiệu trong việc kẻ bảng biểu.

Sau khi nhấp nút Line trên thanh công cụ, hãy thực hiện kẻ bảng cách di chuột.

Với yêu cầu như trên, bằng cách sử dụng các công cụ như đã hướng dẫn hãy thiết kế một report như sau:



Trong đó:

- Textbox bao gồm các ô như đã đánh mũi tên chỉ dẫn. Trong đó:
  - *hoten, ngaysinh, tenchucvu, tenphongban* được tạo bằng cách kéo từ cửa sổ Field List lên phần Detail;
  - Textbox đếm tổng số cán bộ thiết lập thuộc tính **Control Source** là `=Count([hoten])`, để ở phần Report footer.
  - Textbox để tự động đánh số thứ tự phải thiết lập như sau: **Control Source** là `=1`; thuộc tính **Running sum** là *Over Group*.
- Các hộp chữ còn lại dùng đối tượng Label;
- Toàn bộ bảng biểu sử dụng đối tượng Line. Chú ý các đường kẻ phải được nối khít với nhau. Nếu không khít sẽ tạo ra các khe hở và nét đứt (không liền nét)

#### 4. Report chứa tham số

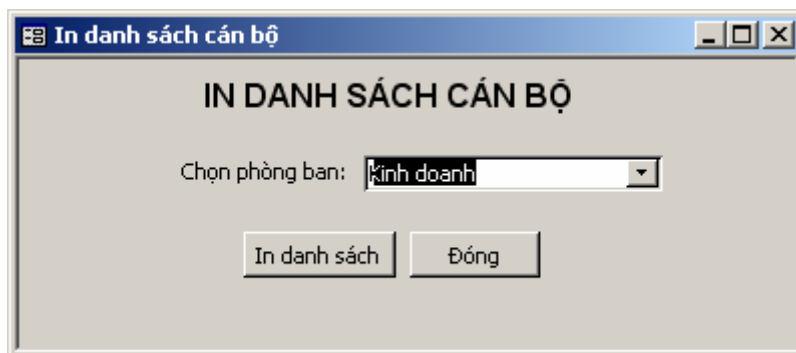
Report có tham số thực chất là loại Report có khả năng lọc dữ liệu khi in. Ví dụ: bình thường Report in ra danh sách cán bộ ở trên sẽ in danh sách toàn bộ cán bộ trong cơ quan. Bây giờ muốn in danh sách cán bộ một phòng ban nào đó? Lúc này phải cần đến report có tham số.

Có nhiều cách để thiết lập và sử dụng report có tham số như:

- Thiết lập tham số trên Record Source của report;
- Thiết lập tham số trên điều kiện lọc (Where Condition) dùng Macro;
- Thiết lập tham số trong câu lệnh VBA DoCmd;
- ...

Trong phần này, chúng tôi hướng dẫn cách thiết lập tham số trên thuộc tính Record Source của report. Cách dễ làm, không cần am hiểu về lập trình VBA. Cách dùng Macro cũng tốt nhưng không nên dùng. Vì phương pháp lập trình này đã trở nên cứng nhắc. Thực tế, phương pháp dùng câu lệnh DoCmd trên VBA là tốt nhất, nó thể hiện tính linh hoạt và chuyên nghiệp của cách giải quyết vấn đề. Riêng cách này, các bạn có thể tham khảo ở Chương Lập trình CSDL.

Sau đây là cách giải quyết bài toán: In danh sách cán bộ một phòng ban nào đó. Phòng ban cần in được chọn từ Combo box một form như sau:



Sau khi chọn tên một phòng ban từ danh sách, nhấn nút **In danh sách**. Danh sách cán bộ phòng đã chọn sẽ được in ra một report.

### **Cách làm:**

#### **Bước 1:** Tạo Report đáp ứng thông tin cần in

Có thể sử dụng report Wizard hoặc Report Design View để tạo ra một Report in danh sách cán bộ với các thông tin như sau:

### Bước 2: Tạo form theo yêu cầu bài toán

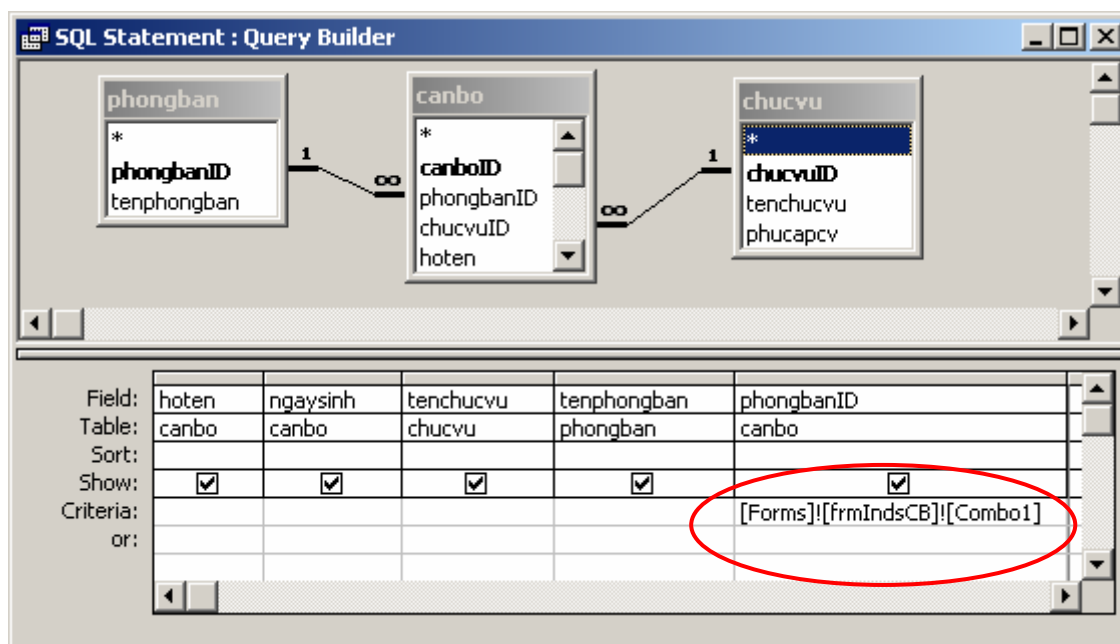
- Có thể sử dụng Combo Wizard để tạo Combobox lấy ra danh sách các phòng ban từ bảng phongban;
- Có thể sử dụng Command button Wizard để tạo các nút lệnh **Đóng** và **In danh sách**;

Cuối cùng được form như sau:

### Bước 3: Thiết lập tham số cho Report

Ở đây trình bày phương pháp thiết lập tham số vào thuộc tính Record Source của Report. Cách làm như sau:

- Mở report đã tạo được ở chế độ Design View (chế độ thiết kế);
- Mở query được thiết lập ở thuộc tính Record Source –nơi tạo nguồn dữ liệu và thiết lập tham số cho trường phongbanID của query như sau:



Trong đó: tham số cho trường phongbanID là tên (Name) của ô Combo box chứa phòng ban được chọn trên form. Cú pháp viết tham chiếu tới một đối tượng trên form như sau:

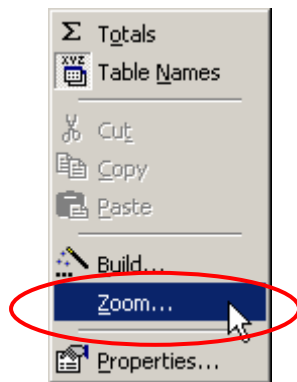
**Forms!<Tên form>!<tên đối tượng>**

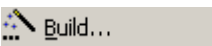

Trong trường hợp này tên ô Combo đó là Combo1 và tên của form (Name của form) là frmIndsCB, cách viết tham số sẽ như sau:

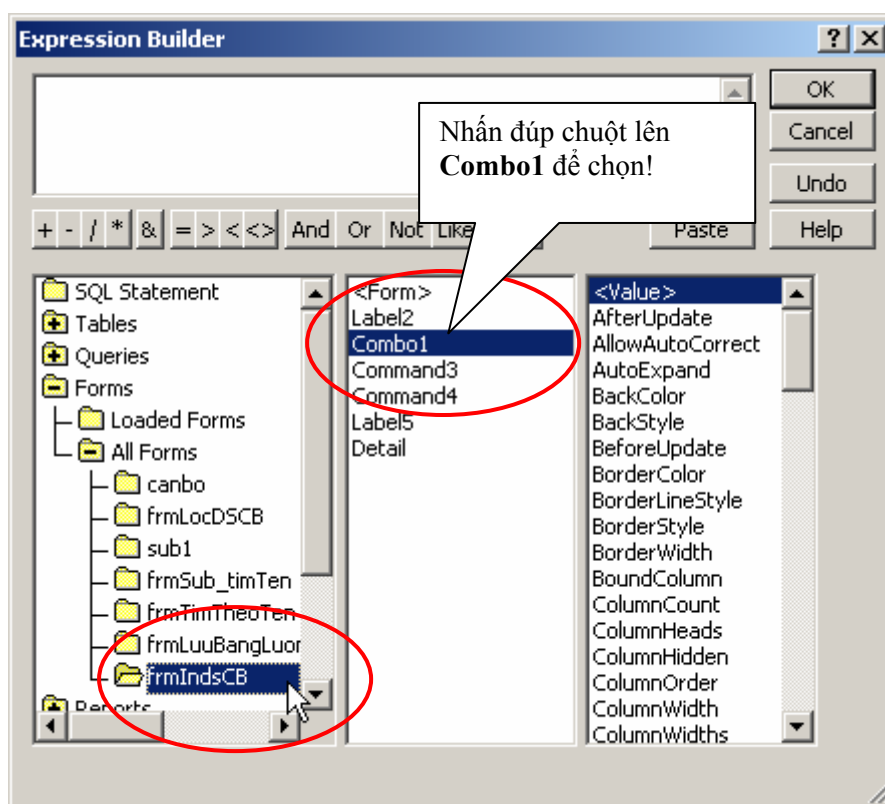
**Forms!frmIndsCB!Combo1**

Nếu cách viết này khó thực hiện, bạn có thể sử dụng tính năng Build Expression có sẵn trên Access để giúp tạo biểu thức này như sau:

Nhấn phải chuột lên ô Criteria của trường *phongbanID*- nơi sẽ gõ vào tham số. Một menu xổ xuống xuất hiện:



Chọn  hoặc có thể nhấn nút *Build*  trên thanh công cụ Standard, hộp thoại Expression Builder xuất hiện:

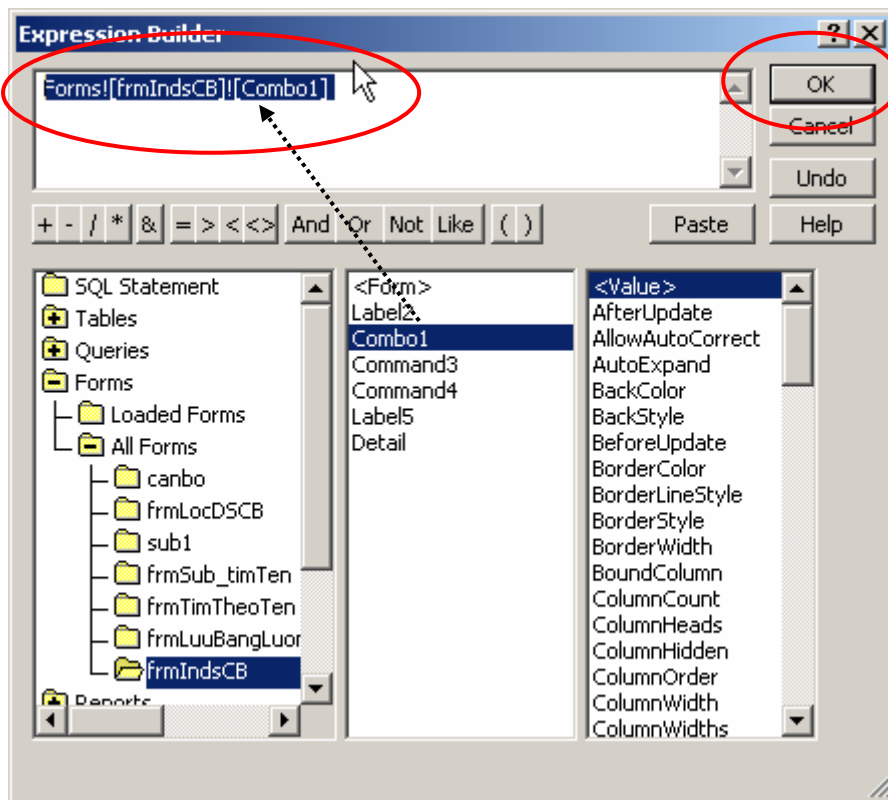


Ở cây các đối tượng bên trái hộp thoại, hãy chọn:

**Forms | All Forms | frmIndsCB** – đây là form chứa đối tượng combo box phòng ban cần lọc. Khi đó một danh sách các đối tượng trên form frmIndsCB xuất hiện ở giữa hộp thoại;

Hãy nhấn đúp chuột lên **Combo1** - đối tượng chứa phòng ban cần lọc, được kết quả như hình sau:





Cuối cùng nhấn **OK** để hoàn tất việc tạo tham số cho query bằng hộp thoại Expression Builder.

## Bài tập

Trên CSDL Quản lý lương cán bộ thực hiện các yêu cầu sau:

**Bài số 1:** Thiết kế report in danh sách cán bộ với các thông tin: *canboID*, *hoten*, *ngaysinh*, *gioitinh* (phải ghi rõ là Nam hay Nữ), *tencv*, *tenphan*.

**Bài số 2:** In danh sách cán bộ một phòng ban nào đó như sau:

Sau khi chọn tên một phòng ban, nhấn nút In danh sách cán bộ. danh sách cán bộ phòng ban đã chọn sẽ được in ra một report.

**Bài số 3:** Thiết kế report in bảng lương cán bộ cơ quan, bao gồm các thông tin: *Hoten*, *tencv*, *ngaysinh*, *luongchinh*, *phucapcv*, *dangphi*, *congdoanphi*, *thuclinh*

**Bài số 4:** Thiết kế query in ra bảng tổng hợp cán bộ như sau:

STT	Tên chức vụ	Tổng số cán bộ

**Bài số 5:** Tạo form và report để xem và in bảng lương của các phòng ban như sau:

Chọn phòng ban: kinh doanh In bảng lương

**Bảng lương cán bộ:**

	Họ tên	Ngày sinh	Giới tính	Chức vụ	Lương chính:
▶	Nguyễn Đức Ngô	12/04/1976	Nam	Giám đốc	655200
	Phan Quang Sáng	03/09/1976	Nữ	Nhân viên	504000
	Nguyễn Văn Hoà		Nữ	Nhân viên	420000
	Phan Hoài Linh		Nam	Nhân viên	420000
	Phan Anh Quân	23/07/1978	Nữ	Trưởng phòng	443100
*					

Record: 1 of 5

Sau khi chọn tên một phòng ban, bảng lương các cán bộ phòng đó hiển thị lên sub-form. Nhấn nút In bảng lương, bảng lương riêng phòng ban đó sẽ được in ra một report.

# CHƯƠNG 5

## LẬP TRÌNH VBA CĂN BẢN

Access không những là hệ quản trị CSDL rất mạnh- cụ thể các bạn đã được tìm hiểu rất kỹ ở Chương 1 và Chương 2. Hơn nữa, với những công cụ có sẵn đi kèm như Forms, Report, Macros và đặc biệt là ngôn ngữ lập trình VBA – Access sẽ còn là một công cụ phát triển phần mềm rất mạnh, dễ sử dụng cho các ứng dụng quản lý vừa và nhỏ.

Nội dung chương này sẽ trình bày căn bản về ngôn ngữ lập trình VBA (Visual Basic for Application)- một ngôn ngữ khá quen thuộc đối với những người sử dụng chuyên sâu sản phẩm Microsoft Office. Đây chính là cơ sở quan trọng để các bạn tiếp cận cụ thể chuyên ngành lập trình CSDL sẽ được giới thiệu trong chương tiếp theo.

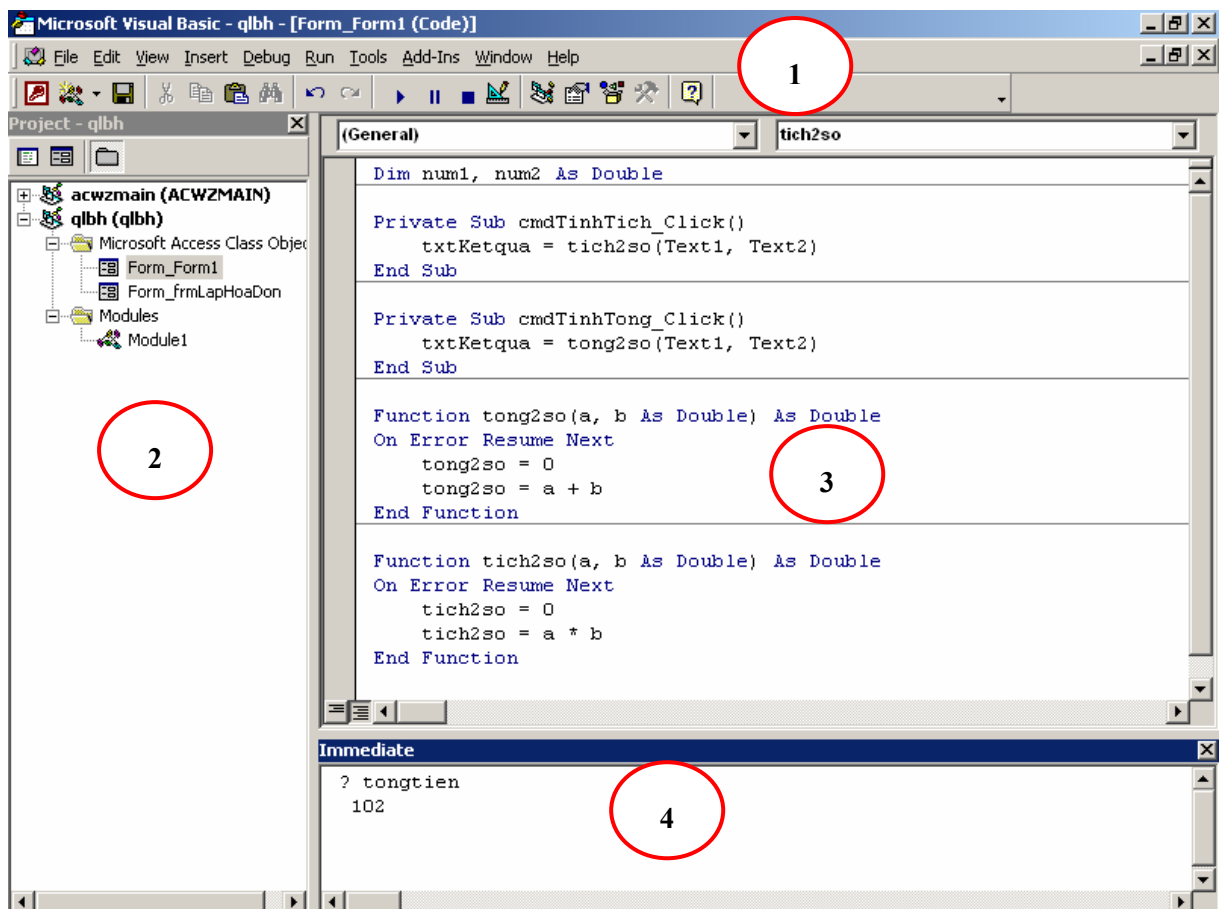
Qua chương này, học viên sẽ hiểu được môi trường làm việc ngôn ngữ VBA; biết cách sử dụng các cấu trúc lệnh; viết và sử dụng tốt chương trình con; đặc biệt dần làm quen việc lập trình trên các đối tượng ActiveX- sẵn sàng tiếp cận các công cụ lập trình hướng đối tượng trực quan hiện đại như Visual Basic và Visual Basic .NET.

## 1. Môi trường lập trình VBA

Bộ phần mềm tin học văn phòng Microsoft Office phải nói là nổi tiếng nhất thế giới trong lĩnh vực tin học văn phòng. Word cung cấp khả năng chế bản điện tử đẹp đẽ và hiện đại; Excel với khả năng bảng tính điện tử mạnh mẽ; FrontPage với khả năng tạo ra các trang web sống động; Access với khả năng quản trị CSDL;... tất cả các phần mềm đó đã tạo nên sự phổ biến của bộ phần mềm này với hầu hết người dùng máy tính trên toàn thế giới.

Không dừng ở mức ứng dụng có sẵn, bộ phần mềm này còn có một ngôn ngữ lập trình đi kèm VBA – Visual Basic for Application để giúp người dùng có thể tạo ra các tùy biến mạnh hơn, thân thiện hơn với trong công việc của mình. Với Word, Excel bạn hoàn toàn có thể tự tạo ra các macro để tăng tốc độ sử dụng ứng dụng; hơn thế nữa VBA trên Access đã thể hiện tính chuyên nghiệp trong lập trình, phần nào biến được một CSDL đơn giản trở thành những sản phẩm đóng gói thương mại.

Màn hình làm việc ngôn ngữ VBA thường có dạng:



**Trong đó:****(1) Hệ thống thực đơn và thanh công cụ**

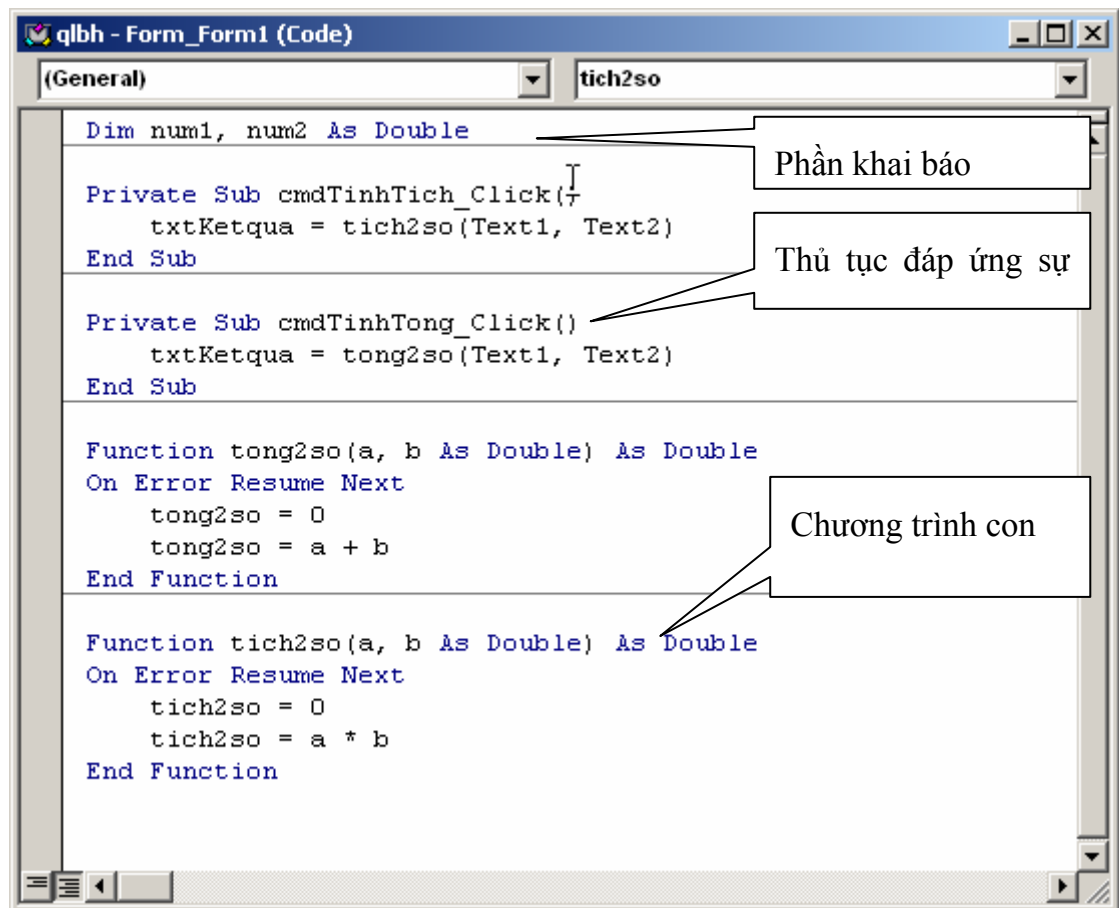
Cũng như bất kỳ môi trường làm việc nào đều có hệ thống thực đơn và thanh công cụ đi kèm. Trên đó có chứa các lệnh để gọi, thi hành hoặc thiết lập các điều khiển cần thiết.

**(2) Cửa sổ Project Explorer;**

Có rất nhiều các thành phần có thể lập trình được bởi VBA như: Forms, Reports, Modules. Cửa sổ Project Explorer là cây phân cấp lớp các đối tượng có chứa mã lệnh VBA, đồng thời giúp lập trình viên dễ dàng trong việc viết (coding) cũng như quản lý các mã lệnh VBA đã viết.

**(3) Cửa sổ viết lệnh;**

Cửa sổ viết lệnh là nơi soạn thảo các dòng lệnh VBA. Mỗi cửa sổ sẽ chứa toàn bộ mã lệnh cho một đối tượng như: Forms, Reports, Modules. Trong mỗi cửa sổ có thể có nhiều phần được viết lệnh, mỗi phần có thể là nội dung một khai báo, một chương trình con, nội dung một thủ tục đáp ứng sự kiện. Ví dụ:



#### (4) Cửa sổ Intermediate

Cửa sổ Intermediate là nơi giúp thi hành trực tiếp một câu lệnh nào đó, rất hữu dụng trong việc gỡ lỗi phần mềm (sẽ quay trở lại vấn đề gỡ rối phần mềm ở cuối chương)

## 2. Các kiểu dữ liệu và khai báo

### 2.1 Các kiểu dữ liệu cơ bản

Cũng như các ngôn ngữ lập trình khác, VBA đều hỗ trợ các kiểu dữ liệu cơ bản. Dưới đây giới thiệu chi tiết về từng kiểu.

#### Boolean

Kiểu lô gíc, tương tự kiểu Boolean trên Pascal. Kiểu này chiếm 2 byte bộ nhớ; chỉ nhận một trong 2 giá trị là: Yes – No hoặc True – False hoặc đôi khi thể hiện

dưới dạng số 0 tương đương với False, True tương ứng với bất kỳ số nào khác 0. Khi lập trình CSDL, kiểu Boolean tương ứng với kiểu Yes/No trong bảng dữ liệu.

### **Byte**

Kiểu số nguyên dương trong phạm vi từ 0..255. Kiểu này chiếm 1 byte bộ nhớ.

### **Integer**

Kiểu nguyên, có giá trị trong khoảng -32768...32767. Kiểu này chiếm 2 bytes bộ nhớ.

### **Long**

Kiểu số nguyên dài, có giá trị trong khoảng 2,147,483,648 .. 2,147,483,647. Kiểu này chiếm 4 bytes bộ nhớ.

### **Single**

Kiểu số thực, có giá trị trong khoảng 1.401298E-45 to 3.402823E38. Chiếm 4 bytes bộ nhớ.

### **Double**

Kiểu số thực có độ lớn hơn kiểu Single, có giá trị trong khoảng 4.94065645841247E-324 to 1.79769313486232E308. Chiếm 8 bytes bộ nhớ.

### **Currency**

Kiểu tiền tệ. Bản chất là kiểu số, độ lớn 8 bytes, có giá trị trong khoảng -922,337,203,685,477.5808 to 922,337,203,685,477.5807. Đặc biệt, kiểu này luôn có ký hiệu tiền tệ đi kèm.



## String

Kiểu chuỗi ký tự. Kiểu này tương ứng với kiểu String trong Pascal, tương ứng với kiểu Text trong các trường CSDL Access. Độ lớn tối đa 255 bytes tương đương với khả năng xử lý chuỗi dài 255 ký tự.

## Variant

Variant là kiểu dữ liệu không tường minh. Biến kiểu này có thể nhận bất kỳ một giá trị nào có thể. Ví dụ :

```
Dim a As Variant  
a = 123  
a = "Nguyễn Văn Ngô"
```

Hoàn toàn không có lỗi.

Người ta thường khai báo biến kiểu Variant trong những trường hợp phải xử lý biến đó mềm dẻo. Khi thì biến nhận giá trị kiểu này, khi thì nhận giá trị và xử lý theo kiểu dữ liệu khác.

## Object

Object là một loại biến kiểu Variant, chiếm dung lượng nhớ 4 bytes, dùng để tham chiếu tới một loại đối tượng (Object) nào đó trong khi lập trình. Tất nhiên muốn khai báo biến Object kiểu nào, phải chắc chắn đối tượng đó đã được đăng ký vào thư viện tham chiếu VBA bởi tính năng **Tool | Reference**. Chúng ta sẽ còn trở lại vấn đề này khi lập trình CSDL.

## 2.2 Biến và cách sử dụng biến

### a. Biến – khai báo biến đơn giản

Biến (Variable) là thành phần của một ngôn ngữ lập trình, giúp xử lý dữ liệu một cách linh hoạt và mềm dẻo.

Thông thường trong các ngôn ngữ lập trình, mỗi biến khi tồn tại phải được định kiểu, tức là phải nhận một kiểu dữ liệu xác định. Tuy nhiên trong VBA thì không,

mỗi biến có thể định kiểu (được khai báo trước khi sử dụng) hoặc không định kiểu (không khai báo vẫn sử dụng được). Trong trường hợp này biến đó sẽ tự nhận kiểu giá trị Variant.

Biến có thể được khai báo bất kỳ ở đâu trong phân viết lệnh của VBA. Tất nhiên, biến có hiệu lực như khai báo chỉ bắt đầu từ sau lời khai báo và đảm bảo phạm vi hoạt động như đã qui định. Vì biến trong VBA hoạt động rất mềm dẻo, nên có nhiều cách khai báo biến như:

**Ví dụ 1:** Khai báo biến i kiểu Integer

```
Dim i As Integer
```

**Ví dụ 2:** Khai báo 2 biến i, j kiểu Integer

```
Dim i, j As Integer
```

**Ví dụ 3:** Khai báo biến i kiểu Integer, st kiểu String độ dài 15 ký tự

```
Dim i As Integer, st As String*15
```

**Ví dụ 4:** Khai báo biến i kiểu Variant

```
Dim i As Variant  
'hoặc  
Dim i
```

**Ví dụ 5:** Khai báo biến txt kiểu Textbox

```
Dim txt As TextBox
```

**Ví dụ 6:** Khai báo mảng kiểu String\*30 gồm 46 phần tử

```
Dim Hoten(45) As String * 45
```

**Ví dụ 7:** Khai báo biến mảng 2 chiều  $A(i, j)$  trong đó:  $i = 0..3$  và  $j = 0..4$

```
Dim A(3, 4) As Integer
```

**Ví dụ 8:** Khai báo mảng 3 chiều  $A(i, j, k)$  trong đó:  $i = 1..5$ ;  $j = 4..9$  và  $k = 3..5$

```
Dim A(1 To 5, 4 To 9, 3 To 5) As Double
```

**Ví dụ 9:** Khai báo một mảng động kiểu Variant. Mảng động là mảng không có định chiều dài.

```
Dim MyArray()
```

## b. Phạm vi biến

Như chúng ta đã biết, mỗi biến sau khi được khai báo nó sẽ nhận một kiểu dữ liệu và có một phạm vi hoạt động, tức là lời khai báo biến chỉ có tác dụng trong những vùng đã được chỉ định; ngoài vùng chỉ định đó biến sẽ không có tác dụng, nếu có tác dụng sẽ theo nghĩa khác (biến cục bộ kiểu Variant chẳng hạn).

### **Biến cục bộ:**

Biến cục bộ được khai báo sau từ khoá **Dim**, nó chỉ có tác dụng trong một chương trình con, cục bộ trong một form hoặc một module nào đó. Dưới đây sẽ chỉ ra 3 trường hợp biến cục bộ này:

- Trong một chương trình con, nếu nó được khai báo trong chương trình con đó;
- Trong cả một Form, nếu nó được khai báo trong phần Declarations của Form đó;

- Trong cả một Reports, nếu nó được khai báo trong phần Decralations của Report đó;
- Trong cả một Modules, nếu nó được khai báo trong phần Decralations của Modules đó;

\* *Biến chỉ có tác dụng sau lệnh khai báo Dim*

### **Biến toàn cục:**

Biến toàn cục được khai báo sau cụm từ khoá **Public**, nó có tác dụng trong toàn bộ chương trình (ở bất kỳ chỗ nào có thể viết lệnh). Loại biến này luôn phải được khai báo tại vùng *Decralations* của một Module nào đó.

### **Ví dụ:**

```
Public Hoten(45) As String * 45
```

Trên một tệp Access, không được phép khai báo trùng tên biến toàn cục. Tuy nhiên tên biến cục bộ vẫn có thể trùng tên biến toàn cục, trong trường hợp đó VBA sẽ ưu tiên sử dụng biến cục bộ trong phạm vi của nó.

## **2.3 Hằng và cách sử dụng hằng**

### **a. Khai báo hằng**

Hằng (Constan) là đại lượng có giá trị xác định và không bị thay đổi trong bất kỳ hoàn cảnh nào. Tương ứng với từng kiểu dữ liệu, sẽ có những hằng tương ứng.

Khai báo hằng số bởi từ khoá **Const**. Sau đây là các ví dụ về khai báo các loại hằng:

**Ví dụ 1:** Hằng a =5 (hằng số)

```
Const a = 5
```

**Ví dụ 2:** Hằng ngày = 24/12/2004 kiểu Date (bao bởi cặp dấu thăng #..#)

```
Const ngay = #24/12/2004#
```

**Ví dụ 3:** Hằng xâu ký tự (bao bởi cặp dấu nháy kép “..”)

```
Const phongban = "Tài vụ"
```

**Ví dụ 4:** Hằng kiểu Logic xác định bởi *True* hoặc *False*

```
Const ok = True
```

## b. Phạm vi hằng

Tương tự như biến, hằng cũng có những phạm vi hoạt động của nó. Hằng được khai báo trong thủ tục nào, hoặc cục bộ trong form, report hoặc module nào sẽ chỉ có tác dụng trong phạm vi đó.

Muốn hằng có phạm vi toàn cục, phải được khai báo sau từ khoá **Public Const**, tại vùng *Declarations* của một module nào đó như sau:

```
Public Const a = 12
```

## 3. Các cấu trúc lệnh VBA

Các cấu trúc lệnh là thành phần cơ bản của mỗi ngôn ngữ lập trình. Thông thường các ngôn ngữ lập trình đều có các cấu trúc lệnh như nhau: lệnh xử lý điều kiện, lệnh lặp biết trước số vòng lặp, lệnh lặp không biết trước số vòng lặp,.. Tuy nhiên cách thể hiện (cú pháp) mỗi cấu trúc lệnh có thể khác nhau tùy thuộc vào mỗi ngôn ngữ lập trình. Hơn nữa, mỗi ngôn ngữ cũng có thể có một số điểm khác biệt, đặc trưng trong mỗi cấu trúc lệnh.

Cũng giống như nhiều ngôn ngữ lập trình hiện đại khác, các cấu trúc lệnh trong VBA đều tuân thủ các nguyên tắc:

- Có cấu trúc: mỗi cấu trúc lệnh đều có từ khoá bắt đầu và một từ khóa báo hiệu kết thúc;
- Thực hiện tuần tự (loại trừ trường hợp đặc biệt thủ tục Goto <Label>);
- Có khả năng lồng nhau;

### 3.1 Cấu trúc IF... END IF

Cấu trúc này thường gọi là *lệnh lựa chọn*. Tức là nếu một điều kiện nào đó xảy ra sẽ là gì, hoặc trái lại có thể làm gì. Trong VBA cú pháp lệnh này như sau:

```
If <điều kiện> Then
    <thủ tục 1>
[ Else
    <thủ tục 2> ]
End If
```

Ý nghĩa lệnh trên là: **nếu** <điều kiện> = True **thì** thực hiện các lệnh trong <thủ tục1>. Trái lại thực hiện các lệnh trong <thủ tục 2>.

Phần trong cặp dấu ngoặc vuông [...] có thể có hoặc không có trong câu lệnh, tùy thuộc vào mục đích xử lý.

**Ví dụ 1:** Kiểm tra và trả lời một số là *chẵn* hay *lẻ*?

```
If so Mod 2 = 0 Then
    MsgBox "Là số chẵn !"
Else
    MsgBox "Là số lẻ !"
End If
```

Cho biết *thang* (số nguyên) rơi vào đầu năm (1..4), giữa năm (5..8) hay cuối năm (9//12)?

```
If thang >=9 Then
    MsgBox "Cuối năm "
Else
    If thang >=5 Then
        MsgBox "Giữa năm "
    Else
        MsgBox "Đầu năm "
    End If
End If
```

### 3.2 Cấu trúc SELECT CASE .. END SELECT

Đây là một loại của cấu trúc lựa chọn. Thông thường hoàn toàn có thể sử dụng If .. End If để thực hiện các xử lý liên quan đến kiểu cấu trúc này, nhưng trong những trường hợp đặc biệt, cấu trúc **Select Case .. End Select** thể hiện được sự tiện dụng vượt trội. Trong VBA cú pháp lệnh này như sau:

```
Select Case <biểu thức>
  Case <giá trị 1>
    <thủ tục 1>
  Case <giá trị 2>
    <thủ tục 2>
  .....
  Case <giá trị n>
    <thủ tục n>
  [Case Else
    <thủ tục n+1>]
End Select
```

**Trong đó:** <Biểu thức> luôn trả về giá trị kiểu vô hướng đếm được như: số nguyên, xâu ký tự, kiểu lô gíc,..

Với cấu trúc này, VBA hoạt động như sau:

(1) Tính giá trị của biểu thức

(2) Kiểm tra <biểu thức> = <giá trị 1> ?

- Nếu đúng thực hiện <thủ tục 1> và kết thúc lệnh, thực hiện lệnh tiếp theo sau từ khoá End Select.

- Nếu sai, thực hiện tiếp việc so sánh <biểu thức> = <giá trị i> tiếp theo và xử lý tương tự qui trình nêu trên.

(3) Trong trường hợp <biểu thức> <> <giá trị i>, i=1..n khi đó có 2 khả năng:

- Nếu có tùy chọn *Case Else* thì VBA sẽ thực hiện <thủ tục n+1>;

- Nếu không có tùy chọn *Case Else*, VBA sẽ không thực hiện bất kỳ thủ tục nào đã liệt kê trong vùng `Select .. End Select` cả mà chuyển tới thực hiện lệnh tiếp theo sau từ khoá `End Select`.

Xét ví dụ sau: Kiểm tra một số nguyên (so) và trả về tên tiếng Anh tháng tương ứng với số nguyên đó (biến thang), ví dụ:

1 - *January*

2 - *February*

...

12 - *December*

>12 - *Không xác định*

Nếu dùng lệnh `If` hoàn toàn có thể đáp ứng được bài toán này, thay vào đó sẽ là một tập hợp 12 lệnh `If .. Else .. End If` như sau:

```
If so = 1 Then
    thang = "January"
Else
    If so = 2 Then
        thang = "Feb"
    Else
        If so = 3 Then
            thang = "Feb"
        Else
            If so = 4 Then
                thang = "Feb"
            Else
                If so = 5 Then
                    thang = "Feb"
                Else
                    If so = 6 Then
                        thang = "Feb"
                    Else
                        If so = 7 Then
                            thang = "Feb"
                        Else
                            If so = 8 Then
                                thang = "Feb"
                            Else
                                If so = 9 Then
                                    thang = "Feb"
                                Else
                                    If so = 10 Then
                                        thang = "Feb"
                                    Else
                                        If so = 11 Then
```





### 3.3 Cấu trúc FOR ... NEXT

**For... Next** là một cấu trúc lặp biết trước số lần lặp trong VBA, tuy nhiên trong những tình huống đặc biệt, vẫn có thể sử dụng cấu trúc này như cấu trúc không biết trước được số lần lặp.

Cú pháp cấu trúc For...Next như sau:

```
For <biến chạy> = <giá trị 1> To <giá trị 2> [Step <n>]
    <thủ tục>
[Exit For]
Next
```

#### Trong đó:

- <biến chạy> là biến kiểu vô hướng đếm được, hay dùng nhất là biến kiểu nguyên;
- <giá trị 1>, <giá trị 2> là các giá trị mà biến chạy sẽ nhận và thực hiện dịch chuyển sau mỗi lần lặp. Có thể dịch chuyển đi 1 đơn vị, có thể dịch chuyển đi nhiều đơn vị một lần, có thể dịch chuyển tiến, cũng có thể dịch chuyển lùi- tất cả điều này tùy thuộc vào việc có hay không có tùy chọn [Step <n>];
- Nếu có tùy chọn [Step <n>] biến chạy sẽ dịch n đơn vị sau mỗi lần lặp. Khi đó, nếu n>0 dễ dịch tiến, ngược lại sẽ dịch lùi;
- Mỗi lần lặp, VBA sẽ thực hiện <thủ tục> một lần;
- Trong trường hợp đặc biệt nếu gặp phải lệnh **Exit For** trong vòng lặp, ngay lập tức thoát khỏi lệnh lặp và thực hiện lệnh tiếp ngay sau từ khoá Next. Chính **Exit For** đã làm mất đi tính lặp biết trước được số lần lặp của loại lệnh này.

Tiếp theo là các ví dụ:

**Ví dụ 1:** Tính tổng các số từ 1 đến 50, giá trị được lưu vào biến *tong*.

```
Dim i As Byte
Dim tong As Integer
```

```
tong = 0
For i = 1 To 50
    tong = tong + i
Next
Msgbox tong
```

**Ví dụ 2:** Tính tổng các số chia hết cho 3 từ 1 đến 50, giá trị được lưu vào biến *tong*.

```
Dim i As Byte
Dim tong As Integer

tong = 0
For i = 3 To 50 Step 3
    tong = tong + i
Next
Msgbox tong
```

Lệnh For trong ví dụ này chỉ khác lệnh For ở ví dụ 1 ở chỗ **Step 3**. Vì <giá trị *i*> = 3 là số chia hết cho 3, nên tất cả các giá trị *i* còn lại sẽ chia hết cho 3 (vì  $i = i + 3$ ).

**Ví dụ 3:** Kiểm tra một số nguyên (>2) có phải là nguyên tố hay không?

```
Dim so As Integer
Dim uoc As Integer
Dim nguyento As Boolean

nguyento = True
For uoc = 2 To Int(so / 2)
    If so Mod uoc = 0 Then
        nguyento = False
        Exit For
    End If
Next

If nguyento Then
    MsgBox "là nguyên tố"
Else
    MsgBox "không là nguyên tố !"
End If
```

Giải thuật đơn giản để xác định một số có phải nguyên tố hay không là: xác định xem tất cả các số (*uoc*) có thể trở thành ước của số (*so*) cần kiểm tra. Nếu tìm thấy

một ước thực sự đầu tiên, kết luận ngay không phải số nguyên tố bởi lệnh *nguyento* = *False* và thoát khỏi vòng lặp bằng lệnh **Exit For**; trong trường hợp xét toàn bộ các ước có thể mà không tìm được một số nào là ước thực sự, kết luận đây là số nguyên tố (biến *nguyento* = *True* như giá trị ban đầu)

### 3.4 Cấu trúc WHILE ... WEND

**While ... Wend** là một cấu trúc lặp không biết trước số lần lặp trong VBA. Cấu pháp cấu trúc While...Wend như sau (Wend - viết tắt của cụm từ **While End**):

```
While <điều kiện>
    <thủ tục>
Wend
```

#### Trong đó:

- **While, Wend** là các từ khoá của lệnh lặp;
- Nếu <điều kiện> = *True*, các lệnh trong <thủ tục> sẽ được thực hiện. Thực hiện xong lại quay lên dòng lệnh While để kiểm tra tiếp <điều kiện>;
- Nếu <điều kiện> = *False*, sẽ thoát khỏi vòng lặp và thực hiện lệnh tiếp theo từ khoá **Wend**.

**Chú ý:** Luôn phải chứng minh được rằng, sau một số hữu hạn lần thực hiện <thủ tục>, giá trị của <biểu thức> phải là *False* để thoát khỏi vòng lặp. Trong trường hợp không thể thoát khỏi vòng lặp, có nghĩa người lập trình đã mắc phải *lỗi lặp vô hạn*. Có thể dẫn đến chương trình bị treo.

Các ví dụ:

**Ví dụ 1:** Tính tổng các số chia hết cho 3 trong khoảng từ 1 đến 50

```
Dim i As Byte
Dim tong As Integer

tong = 0
i = 3
While i <= 50
    tong = tong + i
```

```

    i = i + 3
Wend
Msgbox tong

```

**Ví dụ 2:** Ví dụ này thể hiện vòng lặp vô hạn. Lý do có thể là chủ quan, rất đơn giản vì gõ nhầm! Hãy chỉ ra dòng lệnh gõ nhầm và thực hiện sửa cho đúng.

```

Dim i As Byte
Dim tong As Integer

tong = 0
i = 1
While i <= 50
    If i Mod 3 = 0 Then
        tong = tong + i
    End If
    j = i + 1
Wend
Msgbox tong

```

### 3.5 Lệnh DoCmd

Bạn có thể dùng lệnh **DoCmd** để thi hành các công việc thông thường trên Access thông qua môi trường VBA. Ví dụ như: dùng DoCmd để có thể mở form, mở report, query, lọc dữ liệu, thi hành macro xử lý bản ghi, ứng dụng,... Hầu hết các thao tác xử lý trên các đối tượng của Access đều có thể dùng lệnh doCmd để gọi ra thực hiện trong môi trường VBA.

Dưới đây liệt kê một số các phép xử lý của lệnh DoCmd thông dụng:

#### Lệnh đóng một đối tượng

Lệnh này để đóng (Close) hoặc giải phóng đối tượng nào đó ra khỏi bộ nhớ. Hay dùng lệnh này để đóng form đang hoạt động hoặc đóng một report đang preview.

Cú pháp như sau:

**DoCmd.Close** [*ObjectType*], [*ObjectName*], [*SaveOption*]

Trong đó:

**ObjectType** chỉ kiểu đối tượng cần đóng. Cụ thể như sau:

*acForm* Đóng form  
*acReport* Đóng report  
*acQuery* Đóng query  
*acTable* Đóng bảng

**ObjectName** - chỉ tên đối tượng cần đóng;

**SaveOption** - chỉ định tùy chọn ghi lại cấu trúc (nếu có sự thay đổi). Cụ thể:

*SaveNo* Không khi lại  
*SaveYes* Luôn ghi lại  
*SavePromt* Hiện thị hộp thoại nhắc để ghi nếu có sự thay đổi

Ví dụ sau để đóng form frmHoadon, không cần ghi lại cấu trúc nếu có sự thay đổi.

```
DoCmd.Close acForm, "frmHoadon", acSaveNo
```

Đặc biệt, để ra lệnh đóng đối tượng chủ đang mở chỉ cần ra lệnh sau:

```
DoCmd.Close
```

## Lệnh mở form

Là một lệnh hoàn chỉnh để mở và thiết lập môi trường làm việc cho một form. Cú pháp như sau:

```
DoCmd.OpenForm [ObjectName], [ViewMode], [FilterName], [WhereCondition],  
[DataMode], [WindowsMode]
```

Trong đó:

**ObjectName** – tên form muốn mở;

**ViewMode** - chế độ mở. Cụ thể:

*acDesign* Mở form ra chế độ thiết kế  
*acNormal* Mở form ra để thi hành

**FilterName** - Đặt lọc

**WhereCondition** - Giới hạn các bản ghi trong nguồn dữ liệu

**DataMode** - thiết lập chế độ dữ liệu trên form, cụ thể:

**WindowsMode** - thiết lập kiểu cửa sổ form là:

*acDialog* Kiểu hộp thoại  
*acWindowsNormal* Kiểu cửa sổ bình thường

Ví dụ: Dưới đây là lệnh mở form lập hoá đơn bán hàng (frmLapHoaDon), trong đó chỉ hiển thị nội dung của hoá đơn có mã "HD0035"

```
DoCmd.OpenForm "frmLapHoaDon", , , "hoadonID = 'HD0035'"
```

## Lệnh mở report

Là một lệnh hoàn chỉnh để mở và thiết lập môi trường làm việc cho một report. Cú pháp như sau:

**DoCmd.OpenReport** [*ObjectName*], [*ViewMode*], [*FilterName*], [*WhereCondition*], [*DataMode*], [*WindowsMode*]

Trong đó:

**ObjectName** – tên Report muốn mở;

**ViewMode** - chế độ mở. Cụ thể:

*acDesign* Mở Report ra chế độ thiết kế  
*acNormal* Mở Report ra để thi hành

**FilterName** - Đặt lọc

**WhereCondition** - Giới hạn các bản ghi trong nguồn dữ liệu

**DataMode** - thiết lập chế độ dữ liệu trên Report , cụ thể:

**WindowsMode** - thiết lập kiểu cửa sổ Report là:

*acDialog* Kiểu hộp thoại  
*acWindowsNormal* Kiểu cửa sổ bình thường

Ví dụ: Dưới đây là lệnh **Preview report** để in ra hoá đơn bán hàng (rptHoaDon), trong đó chỉ hiển thị nội dung của hoá đơn hiện tại trên một form (ô chứa mã hoá đơn là *txtHoadonID*)

```
DoCmd.OpenReport "rptHoadon", , , "hoadonID = '" + txtHoadonID + ""
```

## Lệnh thi hành câu lệnh SQL

Dùng để thi hành một lệnh SQL. Cú pháp như sau:

### DoCmd.RunSQL

Giả sử trên bảng canbo có thêm trường *luongchinh*. Lệnh sau đây sẽ cập nhật giá trị cho trường này thông qua lệnh SQL cập nhật dữ liệu:

```
DoCmd.RunSQL "UPDATE canbo SET luongchinh = hessoluong*290000"
```

Hoặc dưới đây là lệnh xoá bỏ những cán bộ đến tuổi nghỉ hưu ra khỏi CSDL:

```
DoCmd.RunSQL "DELETE * FROM canbo " _
+ " WHERE (Year(Date())-Year([ngaysinh])>=60 AND gioitinh=Yes) "
+ " OR (Year(Date())-Year([ngaysinh])>=55 AND gioitinh=No) "
```

## 4. Chương trình con

Chương trình con là một đơn vị mã lệnh VBA, nó có thể chứa tập hợp các câu lệnh nhằm thao tác, tính toán hoặc điều khiển mục đích hoặc dữ liệu nào đó. Trong VBA có 2 loại chương trình con:

- Chương trình con dạng thủ tục, được khai báo bởi từ khoá Sub;
- Chương trình con dạng hàm, được khai báo bởi từ khoá Function.

Về bản chất, 2 loại chương trình con trên đều như nhau: khai báo, tham số và truyền tham số. Tuy nhiên, điểm khác nhau cơ bản là:

- Function luôn trả về một giá trị kiểu vô hướng chuẩn, ví dụ: hàm Date() - trả về giá trị ngày hiện tại kiểu Date. Trong Access đã sẵn có rất nhiều các hàm tính toán (tham khảo ở trang ...), chúng được gọi là các build-in function. Hơn nữa, người dùng hoàn toàn có thể tạo ra các hàm để sử dụng cho các mục đích riêng loại hàm này gọi là user-define function;
- Còn Sub thì không, nó chỉ thực hiện một số các công việc. Tất nhiên những công việc này hoàn toàn có thể làm thay đổi dữ liệu theo mong muốn trong chương trình. Cũng như Function, Access và VBA sẵn có một thư viện các thủ



tục; hơn nữa người dùng cũng có thể tự tạo thêm những thủ tục mới phục vụ việc xử lý dữ liệu theo mục đích riêng. Đặc biệt, Access còn định nghĩa thủ tục đáp ứng sự kiện. Thủ tục này sẽ được tự động gọi ra khi sự kiện đáp ứng bị ảnh hưởng. Chúng ta sẽ trở lại nội dung này qua các ví dụ lập trình VBA.

Tùy từng tính huống cụ thể sẽ lựa chọn sử dụng Function hoặc Sub.

## 4.1 Chương trình con dạng hàm

Cú pháp

```
Function <tên hàm>([<danh sách các tham số>]) As <kiểu DL hàm>
    <thủ tục>
End Function
```

**Trong đó:**

- **Function, End Function** là các từ khoá bắt buộc khai báo cấu trúc một chương trình con dạng hàm;
- <tên hàm> là tên gọi hàm định khai báo. Tên không được chứa dấu cách (space) và các ký tự đặc biệt;
- <danh sách các tham số> - danh sách các tham số cần thiết cho hàm. Có hay không có danh sách này tùy thuộc vào hàm cần định nghĩa;
- <kiểu DL hàm> - kiểu dữ liệu mà hàm sẽ trả lại. Phần này bắt buộc phải được khai báo với mỗi hàm;
- <thủ tục> - thân chương trình con. Trong đó câu lệnh <tên hàm> = <biểu thức> phải xuất hiện ít nhất một lần trong thủ tục. Câu lệnh này có tác dụng gán giá trị cho hàm.

Nếu không có từ khoá Public trước Function, hàm đó chỉ có tác dụng cục bộ: trong một module, trong một report hoặc trong một form. Khi có từ khoá Public trước Function, hàm sẽ có tác dụng toàn cục. Tức là có thể sử dụng bất kỳ nơi nào trên tệp Access đó. Tất nhiên, tất cả những gì khai báo là Public phải được khai báo trong phần **Declarations** của một Module nào đó.

Các ví dụ:

**Ví dụ 1:** hàm tính tổng 2 số

```

Function Tong2So(a, b As Double) As Double
    Tong2So = a + b
End Function

```

**Ví dụ 2:** hàm kiểm tra một số có phải là nguyên tố hay không?

```

Function laNguyenTo(so As Integer) As Boolean
    Dim uoc As Integer

    laNguyenTo = True
    If so > 2 Then
        For uoc = 2 To Int(Sqr(so))
            If so Mod uoc = 0 Then
                laNguyenTo = False
                Exit For
            End If
        Next
    End If
End Function

```

Ví dụ trên có sử dụng đến:

- hàm *Int(number)* – hàm lấy phần nguyên của một số;
- hàm *Sqr(number)* – hàm lấy căn bậc hai một số

**Ví dụ 3:** hàm tách tên trong chuỗi họ và tên.

Đây là một bài toán gặp phải rất nhiều trong thực tế. Cụ thể bài toán giải quyết vấn đề sau:

Nếu biết họ tên là *Nguyễn Sơn Hải*, hàm sẽ tách ra được tên là *Hải*. Toàn bộ mã lệnh hàm như sau:

```

Function GetTen(hoten As String) As String
    Dim pos As Integer

    pos = 1
    If InStr(pos, Trim(hoten), " ") = 0 Then
        GetTen = hoten
        Exit Function
    End If
    While InStr(pos + 1, Trim(hoten), " ") > 0

```

```

        pos = InStr(pos + 1, Trim(ten), " ")
    Wend

    GetTen = Mid(hoten, pos)
End Function

```

**Ví dụ 4:** Hàm dùng so sánh 2 chuỗi kiểu chữ TCVN3 chúng tôi đưa ra dưới đây là một tham khảo rất tốt. Trong Word, Access cũng như các bảng dữ liệu tiếng Việt có dấu trên máy tính, việc sắp xếp chuỗi ký tự là một bài toán mà người Việt phải giải quyết. Ví dụ, dưới đây là một danh sách trên Word:

STT	Tên
1	Quang
2	Đức
3	Đoàn
4	Băng
5	Bang
6	An
7	Ân

Sau khi sử dụng tính năng sắp xếp (Sort) của Word theo cột Tên theo thứ tự tăng dần, được danh sách kết quả như sau:

STT	Tên
7	Ân
3	Đoàn
2	Đức
6	An
4	Băng



```

kq = ""
xd = ""

For i = 1 To Len(Ckt)
    kti = Mid(Ckt, i, 1)
    vt1 = InStr(Cgoc1, kti)
    If vt1 <> 0 Then
        kq = kq & Mid(Cma1, 1 + ((vt1 - 1) \ 6) * 2, 2)
        xd = xd & Mid(Cma1, 25 + ((4 + vt1) Mod 6) * 2, 2)
    Else
        vt2 = InStr(Cgoc2, kti)
        If vt2 <> 0 Then
            kq = kq & Mid(Cma2, (vt2) * 2 - 1, 2)
        Else
            kq = kq + kti
        End If
    End If
Next i
Mahaotcvn3 = kq & xd
End Function

Function Mahaotcvn3(Ckt As String) As String
    Dim vt1 As Integer
    Dim kq, Ctam As String
    Ckt = Ckt & " "
    kq = ""
    vt1 = InStr(Ckt, " ")

    Do While vt1 <> 0
        Ctam = Trim(Left(Ckt, vt1 - 1))
        Ckt = Right(Ckt, Len(Ckt) - vt1)
        kq = Mahaotcvn3(Ctam) & " " & kq
        vt1 = InStr(Ckt, " ")
    Loop
    Mahaotcvn3 = kq
End Function

```

## 4.2 Chương trình con dạng thủ tục

Cú pháp

```

[Public] [Private] Sub <tên CTC>([<danh sách các tham số>])
    <thủ tục>
End Sub

```

**Trong đó:**

- **Sub, End Sub** là các từ khoá bắt buộc khai báo cấu trúc một chương trình con dạng thủ tục;
- *<tên CTC>* là tên gọi thủ tục định khai báo. Tên không được chứa dấu cách (space) và các ký tự đặc biệt;
- *<danh sách các tham số>* - danh sách các tham số cần thiết cho thủ tục. Có hay không có danh sách này tùy thuộc vào thủ tục cần tạo
- *<thủ tục>* - thân chương trình con.

Nếu không có từ khoá Public trước Sub, thủ tục đó chỉ có tác dụng cục bộ: trong một module, trong một report hoặc trong một form. Khi có từ khoá Public trước Sub, thủ tục sẽ có tác dụng toàn cục. Tức là có thể sử dụng bất kỳ nơi nào trên tệp Access đó. Tất nhiên, tất cả những gì khai báo là Public phải được khai báo trong phần Declarations của một Module nào đó.

Các ví dụ:

#### **Ví dụ 1:** Thủ tục tính tổng hai số

```
Sub tong2so(a, b As Double)
    tong = a + b
```

*'chú ý: **tong**- là biến được khai báo toàn cục*  
End Sub

#### **Ví dụ 2:** Cũng là tính tổng, nhưng thủ tục sau đây không có ý nghĩa gì!

```
Sub tong2so(a, b As Double)
    Dim tong As Double
```

```
tong = a + b
```

*'chú ý: **tong**- là biến được khai báo toàn cục*  
End Sub

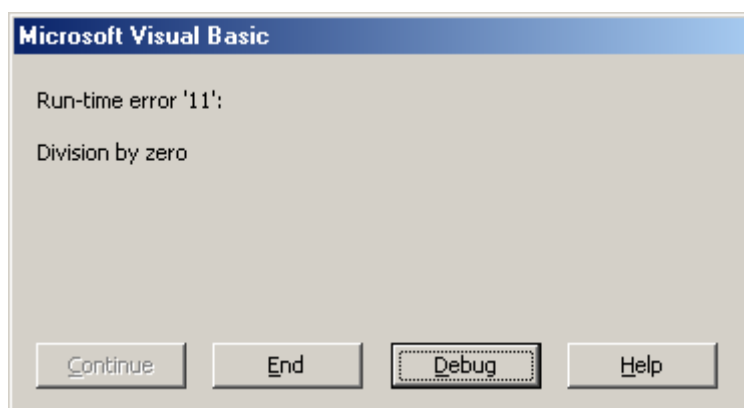
Vì sao? Vì biến *tong* được khai báo cục bộ trong CTC *tong2so*, nên khi CTC này kết thúc, biến *tong* cũng bị giải thoát khỏi bộ nhớ luôn. Không gây ảnh hưởng gì đến dữ liệu cũng như thể hiện của chương trình.

## 5. Kỹ thuật xử lý lỗi

Xử lý lỗi là kỹ thuật rất quan trọng trong lập trình. Đã lập trình thì khó tránh khỏi lỗi (Errors). Có rất nhiều nguyên nhân gây ra lỗi; các nguyên nhân này có thể được lường trước hoặc không được lường trước. Kỹ thuật xử lý lỗi bao gồm các kỹ năng phát hiện và xử lý các tình huống khi chương trình gây lỗi.

### 5.1 Xử lý lỗi

Là việc xử lý khi đang lập trình gặp phải lỗi. Thông thường khi chạy thử chương trình trong lúc đang xây dựng phần mềm nếu gặp phải lỗi, sẽ xuất hiện hộp thoại thông báo lỗi có dạng:




Thông thường một hộp thoại thông báo lỗi gồm 2 thành phần:


- Thành phần báo lỗi bao gồm:

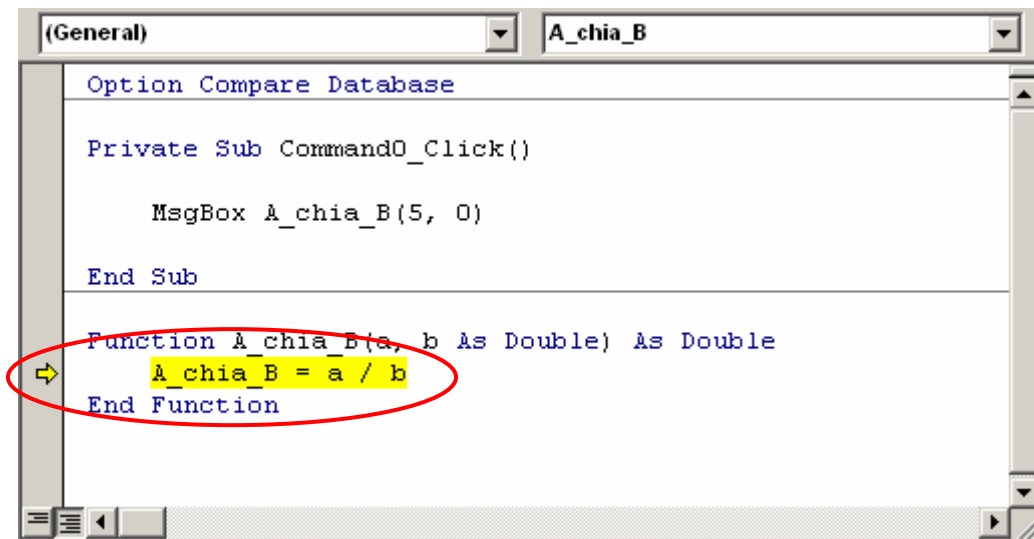
+ **Mã số lỗi** - Mỗi lỗi mà VBA có thể kiểm tra được đều có một mã số, được hiển thị ở dòng thông báo: *Run-time error 'mã số lỗi'*: Ví dụ trên là : *Run-time error '11'*:

+ **Tên lỗi**. Ở ví dụ trên tên lỗi là: *Division by zero* - lỗi sai kiểu dữ liệu.

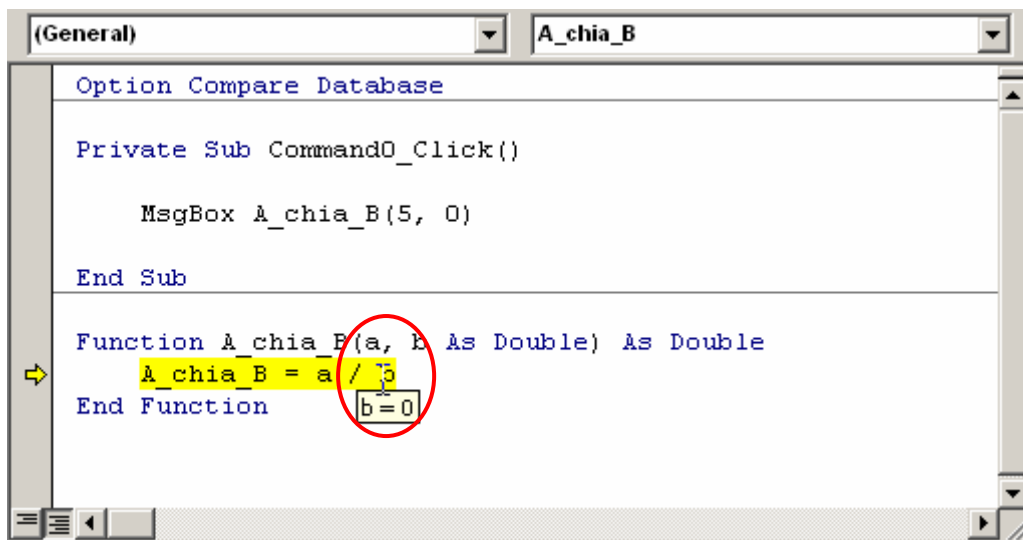
- Thành phần xử lý lỗi gồm 2 nút lệnh:

+ Nút  - để dừng ngay chương trình, chuyển về chế độ **Design** - thiết kế bình thường;

+ Nút  - để dừng chương trình chuyển về chế độ **Break** - sửa lỗi trực tiếp. Khi đó câu lệnh lỗi sẽ được tô bởi màu nền vàng cho phép người lập trình có thể sử dụng mã chương trình:

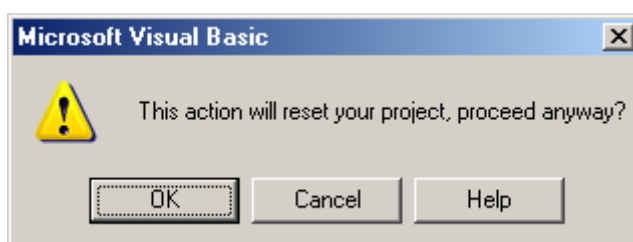


Khi dịch chuột lên một biến nào đó, giá trị biến sẽ được hiển thị dưới dạng **Tool tip**.



Hình trên khi di chuột lên biến b, giá trị biến b xuất hiện dưới dạng Tool tip (giá trị b = 0).

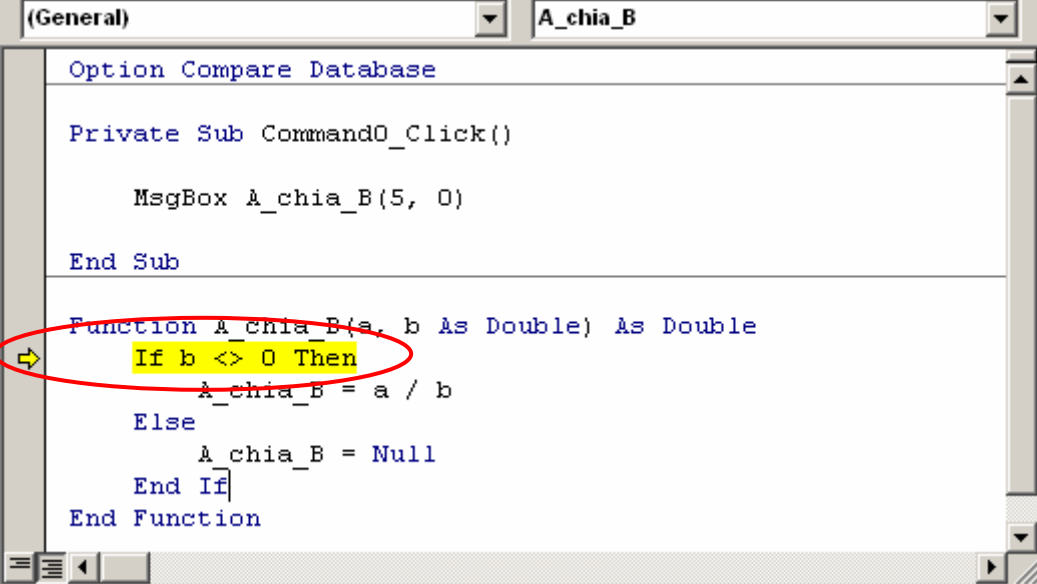
Sau khi chọn nút **Debug**, bạn hoàn toàn có thể thực hiện sửa mã lệnh trong chương trình. Tuy nhiên, trong một số trường hợp khi sửa mã lệnh VBA sẽ hỏi:





Điều này có nghĩa: việc sửa đổi mã lệnh vừa rồi, VBA yêu cầu phải trở về chế độ thiết kế bình thường nếu nhấn **Ok**; trái lại nhấn **Cancel**- việc thay đổi mã lệnh sẽ không được chấp nhận.

Sau khi thực hiện sửa mã lệnh, bạn có thể yêu cầu VBA thực thi tiếp chương trình.





```
(General) A_chia_B
Option Compare Database

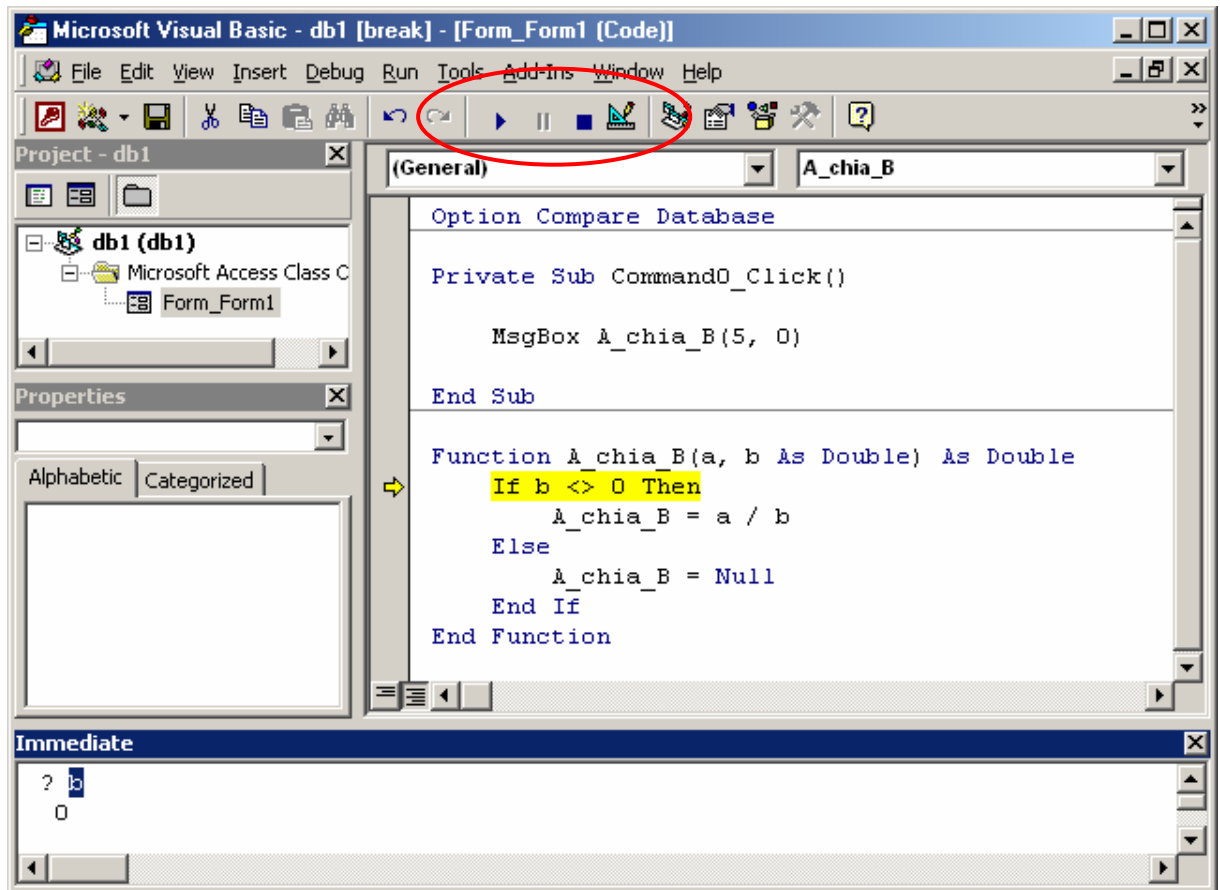
Private Sub Command0_Click()

    MsgBox A_chia_B(5, 0)

End Sub

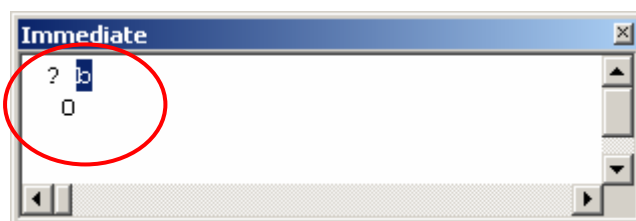
Function A_chia_B(a, b As Double) As Double
    If b <> 0 Then
        A_chia_B = a / b
    Else
        A_chia_B = Null
    End If
End Function
```

Việc thực thi sẽ được tiến hành tiếp tục tại vị trí vệt sáng đang trỏ. Bạn có thể dùng chuột để dịch chuyển vệt sáng về lệnh cần thực thi (chỉ trong cùng một chương trình con). Để thực thi tiếp nhấn phím F5 hoặc nút **Continue**  trên thanh công cụ; hoặc nhấn nút **Stop**  nếu muốn dừng việc sửa mã lệnh trong chế độ *Break*, chuyển về chế độ *Design*.



## Cửa sổ Immediate

Là công cụ hữu hiệu hỗ trợ việc dò lỗi bởi: hộp thoại này cho phép thực thi từng câu lệnh trên chế độ hội thoại.



Giả sử ví dụ trên sau khi gõ lệnh:

```
? b
```

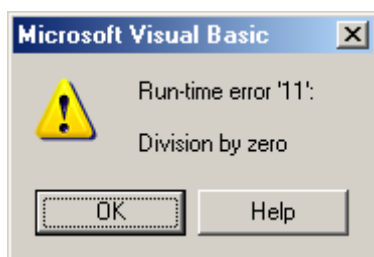
Xem giá trị của biến b. Sau khi nhấn Enter sẽ nhận được kết quả

```
0
```

Hoặc nếu gõ:

? a / b

Sẽ nhận được thông báo lỗi:



Vì  $b = 0$ .

## 5.2 Bẫy lỗi

Mục 5.1 đã trình bày những kỹ năng để xử lý lỗi khi đang soạn thảo chương trình. Các thao tác đó chỉ được thực hiện trong lúc đang xây dựng phần mềm (VBA IDE), do người lập trình xử lý. Khi phần mềm đã được đóng gói để chuyển đến người dùng nếu gặp lỗi, nó sẽ hiển thị một hộp thoại thông báo lỗi (*Error Dialog*) cho biết lý do vắn tắt về lỗi. Sau khi bạn nhấn OK, chương trình sẽ ngừng hoạt động, bị thoát.

Để xử lý lỗi trong tình huống này, có 2 phương pháp bẫy lỗi mà chúng tôi đưa ra dưới đây để tham khảo; hy vọng bạn sẽ chọn lựa được tình huống phù hợp để sử dụng một trong các phương pháp này đảm bảo chương trình viết ra chạy được đúng theo mục đích.

### Sử dụng lệnh **On Error Resume Next**

Khi đó từ chỗ đó trở đi, nếu chương trình gặp lỗi, nó sẽ bỏ qua (ignore) hoàn toàn. Điểm này tiện ở chỗ giúp chương trình EXE của ta tránh gặp lỗi thoát khỏi đột ngột như phân tích ở trên. Nhưng nó cũng bất lợi là khi khách hàng cho hay họ gặp những trường hợp lạ, không giải thích được (vì lỗi đã bị bỏ qua mà không ai để ý), thì ta cũng bí luôn, có thể không biết bắt đầu từ đâu để gỡ lỗi. Do đó, trong lúc gỡ lỗi ta không nên dùng nó, nhưng trước khi giao cho khách hàng bạn nên cân nhắc kỹ có nên sử dụng trong các đoạn mã lệnh hay không.

Ví dụ sử dụng **On Error Resume Next** để bỏ qua lỗi:

```
Function A_chia_B(a, b As Double) As Double
    On Error Resume Next

    A_chia_B = Null

    A_chia_B = a / b

End Function
```

Trong chương trình con trên, nếu  $b = 0$ , lệnh  $A\_chia\_B = a / b$  sẽ gặp phải lỗi. Do có lời khai báo On Error Resume Next nên lệnh lỗi này được bỏ qua (không thực hiện). Tức là giá trị hàm là Null.

### Sử dụng câu lệnh **On Error Goto <nhãn>**

Khi một thủ tục được đặt câu lệnh này, nếu gặp phải một lỗi nào đó, VBA sẽ chuyển thẳng việc thực hiện đến <nhãn> đã chỉ định. Thông thường các lệnh tiếp theo của <nhãn> là xử lý các tính huống lỗi.

Sau đây là ví dụ sử dụng phương pháp **On Error Goto <nhãn>** để bẫy lỗi:

```
Function A_chia_B(a, b As Double) As Double
    On Error GoTo Loi

    A_chia_B = a / b
    MsgBox " Ok! "

    Loi:

    If Err.Number = 11 Then
        MsgBox "Lỗi chia cho 0 !"
    End If

End Function
```

Trong chương trình con trên, trong trường hợp  $b = 0$  câu lệnh  $A\_chia\_B = a / b$  sẽ gây ra lỗi. Theo như khai báo **On Error Goto Loi** ban đầu, VBA sẽ bỏ qua tất cả các lệnh sau lệnh lỗi và chuyển thẳng tới các lệnh sau nhãn **Loi**: Ở đây là lệnh kiểm tra lỗi. Nếu Mã lỗi = 11 → kết luận ngay một thông báo lỗi tiếng Việt. *Lỗi chia cho 0!*

Phương pháp này cũng được dùng phổ biến cả trong quá trình xây dựng để phát hiện lỗi, cũng như trong phần mềm đã đóng gói gửi đến khách hàng. Mỗi khi gặp lỗi sẽ được thông báo nguyên nhân gây ra lỗi bằng tiếng Việt (chẳng hạn) mà vẫn không ảnh hưởng đến hoạt động khác của phần mềm.

Trong phương pháp này, người lập trình nên khai thác tối đa đối tượng **Err** - đối tượng mang những thông tin về lỗi đang xảy ra, cụ thể:

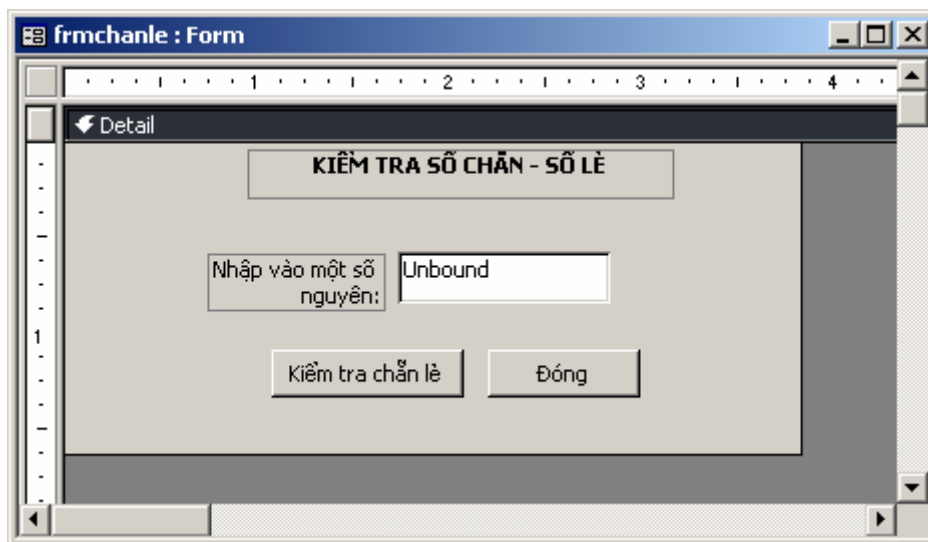
Hành động	Kết quả
Err.Description	Mô tả tên lỗi
Err.Number	Đưa ra mã lỗi
Err.Number	Xoá bỏ các giá trị của đối tượng Err

## 6. Một số ví dụ

Phần này trình bày một số ví dụ sử dụng Form, một số đối tượng điều khiển (Control), các khai báo, các cấu trúc lệnh và những kỹ thuật liên quan để giải quyết một số bài toán thực tế đơn giản.

**Bài toán 1:** Nhập vào một số nguyên và kiểm tra số đó là chẵn hay số lẻ?

Thiết kế form như sau:



Thiết lập thuộc tính cho các đối tượng như sau:

### Form

Caption: *Kiểm tra số chẵn - lẻ*  
 Default view: *Single Form*  
 Scroll bar: *Neither*  
 Record selector: *No*

Navigation Buttons: *No*

Diving line: *No*

### **Ô nhập số cần kiểm tra**

Name: *Text0*

### **Nút Kiểm tra chẵn lẻ**

Name: *cmdChanLe*

Caption: *Kiểm tra chẵn lẻ*

### **Nút Đóng**

Name: *cmdClose*

Caption: *Đóng*

Và mã lệnh cho form như sau:

```
'-----  
'Lệnh cho nút Kiểm tra chẵn lẻ  
'  
Private Sub cmdChanLe_Click()  
    If Text0 Mod 2 = 0 Then  
        MsgBox Text0 + " Là số chẵn !"  
    Else  
        MsgBox Text0 + " Là số lẻ !"  
    End If  
End Sub  
  
'-----  
'Lệnh cho nút Đóng  
'  
Private Sub cmdClose_Click()  
    DoCmd.Close  
End Sub
```

**Bài toán 2:** Nhập vào 2 số nguyên và tính USC và BCS của 2 số đó

Thiết kế form như sau:

Thiết lập thuộc tính cho các đối tượng như sau:

### Form

Caption: *Tim USC và BSC*  
 Default view: *Single Form*  
 Scroll bar: *Neither*  
 Record selector: *No*  
 Navigation Buttons: *No*  
 Diving line: *No*

### Ô nhập số cần kiểm tra

Ô	Thuộc tính Name
A:	txtA
B:	txtB
USC:	txtUSC
BSC:	txtBSC

### Nút Tính toán

Name: *cmdTinhToan*  
 Caption: *Tính toán*

### Nút Đóng

Name: *cmdClose*  
 Caption: *Đóng*

Và mã lệnh cho form như sau:

```

'-----
'Hàm xác định USCNN của 2 số nguyên (thuật toán O-cơ-lít)
'
Function usc(a, b As Integer) As Integer
    Dim a1, b1 As Integer

    a1 = a
    b1 = b
    While a1 <> b1
        If a1 > b1 Then
            a1 = a1 - b1
        Else
            b1 = b1 - a1
        End If
    Wend
    usc = a1

End Function

'-----
'Mã lệnh cho nút Tính toán
'
Private Sub cmdTinhToan_Click()
    txtUSC = usc(txtA, txtB)
    txtBSC = txtA * txtB / usc(txtA, txtB)
End Sub

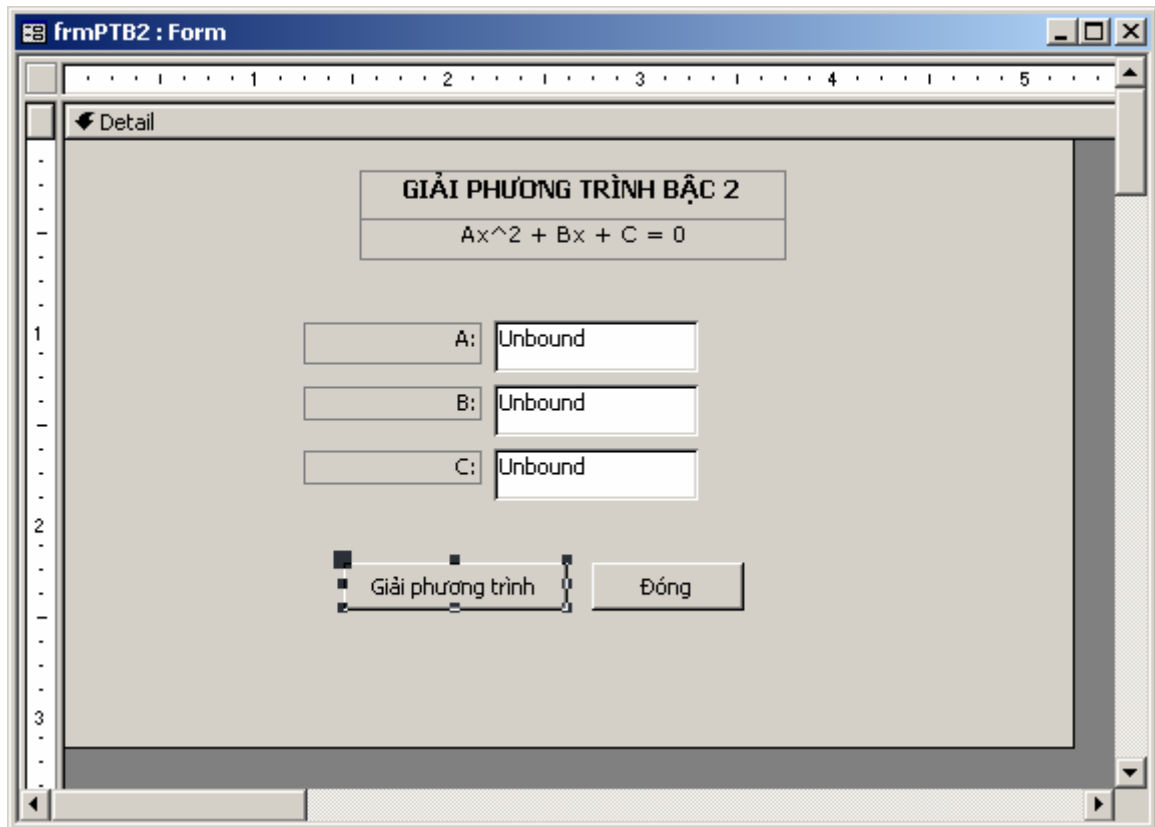
'-----
'Mã lệnh cho nút Đóng
'
Private Sub cmdClose_Click()
    DoCmd.Close
End Sub

```

**Bài toán 3:** Nhập vào 3 hệ số A, B, C của phương trình bậc hai  $Ax^2 + Bx + C = 0$  và cho biết nghiệm phương trình đó:

Thiết kế form như sau:





Thiết lập thuộc tính cho các đối tượng như sau:

### Form

Caption: *Giải phương trình bậc 2*  
 Default view: *Single Form*  
 Scroll bar: *Neither*  
 Record selector: *No*  
 Navigation Buttons: *No*  
 Diving line: *No*

### Ô nhập số cần kiểm tra

Ô	Thuộc tính Name
A:	txtA
B:	txtB
C:	txtC

### Nút Tính toán

Name: *cmdGPTB2*  
 Caption: *Giải phương trình*

### Nút Đóng

Name: *cmdClose*  
 Caption: *Đóng*

Và mã lệnh cho form như sau:

```
'-----  
'Mã lệnh cho nút Giải phương trình  
'  
Private Sub cmdChanLe_Click()  
    Dim delta, x, x1, x2 As Double  
    Dim kqua As String  
  
    delta = txtB * txtB - 4 * txtA * txtC  
    If delta = 0 Then  
        x = -txtB / (2 * txtA)  
        kq = "Nghiem kép: x1 = x2 = " + Trim(Str(x))  
    Else  
        If delta > 0 Then  
            x1 = (-txtB + Sqr(delta)) / (2 * txtA)  
            x2 = (-txtB - Sqr(delta)) / (2 * txtA)  
            kqua = "Có 2 nghiệm phân biệt:" + Chr(13) +  
                + " X1 = " + Trim(Str(x1)) + Chr(13) +  
                + " X2 = " + Trim(Str(x2))  
        Else  
            kqua = "Phương trình vô nghiệm"  
        End If  
    End If  
  
    MsgBox kqua  
  
End Sub  
  
'-----  
'Lệnh cho nút Đóng  
'  
Private Sub cmdClose_Click()  
    DoCmd.Close  
End Sub
```

# CHƯƠNG 7

## LẬP TRÌNH CƠ SỞ DỮ LIỆU

Access không những là một hệ quản trị CSDL mạnh mà còn cung cấp những công cụ mạnh cho phép phát triển một CSDL đơn thuần thành một sản phẩm đóng gói thương mại. Chúng ta đã được học Queries, Forms, Report – đó là những công cụ khá mạnh và dễ học dùng để xây dựng các truy vấn, biểu mẫu và báo cáo in ấn trong Access. Chương 7 đã được làm quen với ngôn ngữ VBA- một ngôn ngữ lập trình trực quan, hướng đối tượng trong các ứng dụng Microsoft Office nói chung cũng như trên Access. Đó là cơ sở quan trọng để trong chương này chúng ta tìm hiểu những kỹ thuật lập trình CSDL bằng VBA- có thể nói là một mức chuyên sâu tiếp theo những gì đã học VBA căn bản.

Các chủ đề sẽ được bàn đến trong chương này bao gồm:

- Lớp đối tượng truy cập dữ liệu (DAO- Data Access Objects);
- Bài toán tìm kiếm;
- Bài toán đặt lọc;
- Một số bài toán khác.

## 1. Kỹ thuật DAO

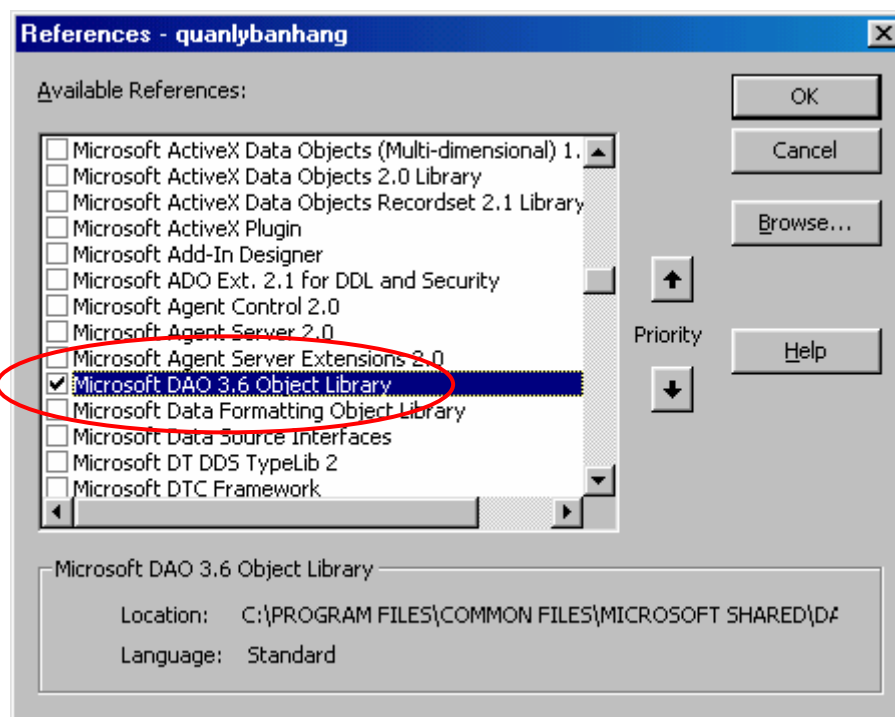
DAO (Data Access Objects – Các đối tượng truy xuất dữ liệu) là tập hợp bao gồm lớp các đối tượng có thể dùng để lập trình truy cập và xử lý dữ liệu trong các hệ CSDL. Ở đây CSDL Access, ngôn ngữ lập trình VBA.

DAO được phát triển khá sớm, gần đây nhất là phiên bản DAO 3.5 và 3.51- nó có thể thực hiện tốt được trên các phiên bản Access từ 97 trở về trước. Với Access 2000, XP phải dùng phiên bản DAO 3.6. Với phiên bản mới này, DAO 3.6 sử dụng nền Microsoft Jet 4.0. Vì vậy, có thể làm việc được trên nền Unicode dễ dàng.

Để nạp thư viện DAO3.6 vào làm việc, hãy thực hiện như sau:

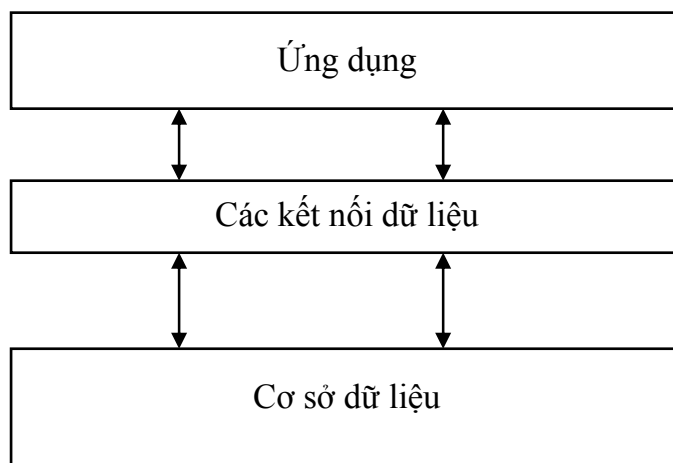
**Bước 1:** Mở cửa sổ lập trình VBA;

**Bước 2:** Chọn thực đơn **Tools | References ..** Hộp thoại sau xuất hiện:



Hãy chọn (tích) mục **Microsoft DAO 3.6 Object Library** trên danh sách *Available References*; chọn xong, nhấn **OK** để đóng lại.

Trước khi bước vào học lập trình CSDL, các bạn hãy xem cách thức làm việc như thế nào?



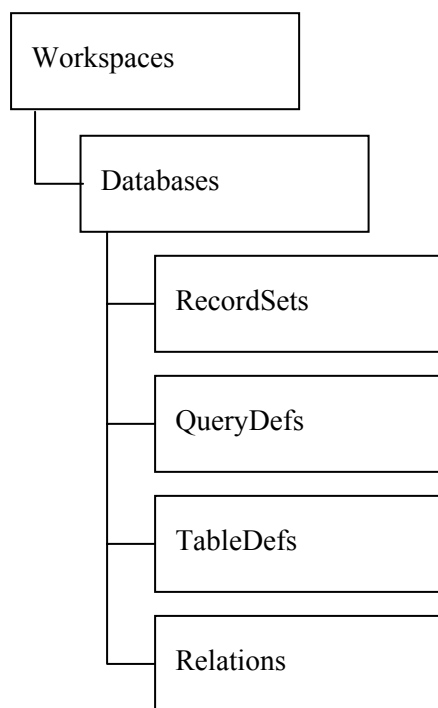
Trong đó:

- Tầng ứng dụng: bao gồm những giao diện người sử dụng cũng như những công cụ đơn giản mà người lập trình có thể dùng để xử lý dữ liệu theo các bài toán;
- Tầng Kết nối dữ liệu: bao gồm tập hợp các công cụ, phương thức để kết nối tới những dữ liệu cần làm việc trong CSDL. Ở đây, tầng kết nối bao gồm các chuẩn Microsoft Jet 4.0 và các lớp đối tượng DAO;
- Tầng Cơ sở dữ liệu: bao gồm các bảng, các query trong cơ sở dữ liệu thực tại.

Như vậy để lập trình trên một CSDL phải sử dụng các đối tượng, các phương thức ở tầng kết nối như là những công cụ để có thể truy cập được vào CSDL tác nghiệp xử lý. Tầng kết nối đó chính là Jet 4.0 và DAO 3.6 mà chúng ta sẽ được tìm hiểu dưới đây.

## 1.1 Lớp đối tượng DAO

Cấu trúc một CSDL bao gồm nhiều thành phần, đòi hỏi lập trình cũng cần có những thành phần tương ứng để làm việc. Lớp các thành phần tương ứng để có thể lập trình được trên toàn bộ cấu trúc CSDL là lớp các đối tượng DAO. Chúng có tên gọi, có những tập thuộc tính, các phương thức làm việc và có quan hệ mật thiết với nhau. Cây phân cấp lớp các đối tượng DAO sau đây thể hiện điều đó:

**Trong đó:**

- **Workspaces** – định nghĩa tập hợp các vùng làm việc. Đây có thể coi là lớp làm việc cao nhất. Về lý thuyết có thể khai báo một vài vùng làm việc (Workspace), nhưng trên thực tế chỉ cần khai báo một vùng làm việc và vùng này luôn được khai báo ngầm định cho CSDL hiện tại. Nên sẽ không cần bàn nhiều đến lớp các WorkSpace này;
- **Databases** - định nghĩa tập hợp các CSDL Access cần làm việc trên một dự án;
- **RecordSets**- định nghĩa các tập hợp bản ghi (Records) cần làm việc;
- **QueryDefs** - định nghĩa tập hợp các Query để làm việc. Querydefs và Recordsets là khả năng truy xuất, xử lý dữ liệu (Data Manipulation) của DAO;
- **TableDefs** - định nghĩa tập hợp các bảng (Table) cần làm việc. Đây là khả năng định nghĩa dữ liệu (Data-Definition Language);
- **Relations** - định nghĩa tập hợp các quan hệ (Relationship) cần làm việc;

Mỗi lớp các đối tượng trên sẽ bao gồm tất cả các đối tượng đối tượng cùng loại trong một đối tượng mẹ đang mở. Ví dụ:

- Databases sẽ bao gồm tất cả các CSDL đang được mở trong vùng làm việc hiện tại;

- RecordSets sẽ bao gồm tập hợp tất cả các Recordset đang được mở trên CSDL hiện tại.

Khi đó, để tham chiếu đến một đối tượng cụ thể cần làm việc, có thể dùng chỉ số (số thứ tự của đối tượng đó trên tập hợp tất cả các đối tượng đó) hoặc dùng tên gọi đối tượng đó để tham chiếu. Ví dụ sau liệt kê tên của tất cả các Recordset đang sử dụng trong CSDL db.

```
Dim db As DAO.Database
'-----
'các câu lệnh tiếp theo ở đây..
'
For i = 0 To db.Recordsets.Count
    MsgBox db.Recordsets(i).Name
Next
```

Để làm việc tới một đối tượng cụ thể, cần phải tham chiếu từ lớp các đối tượng mẹ của nó.

Ví dụ: Để hiển thị giá trị của trường (Field) hoten trên tập hợp các bản ghi (Recordset) rs1 làm như sau:

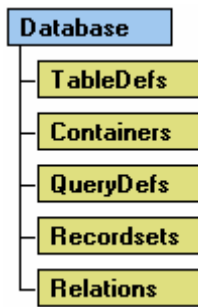
```
MsgBox rs1.Fields("hoten").Value

' hoặc
MsgBox rs1.Fields![hoten].Value
```

## 1.2 Đối tượng Database

Database là đối tượng dùng làm việc với một CSDL (trong trường hợp này có thể hiểu một CSDL như một tệp Access .MDB).

Lớp các đối tượng con của Database được thể hiện qua sơ đồ sau:



## Khai báo

```

Dim db As DAO.Database

' Gán db cho một CSDL cụ thể
Set db = OpenDatabase("C:\Baitap\qlbh.mdb")

'Đặc biệt, lệnh gán db cho CSDL hiện tại như sau:
Set db = CurrentDb
  
```

Khi không làm việc với CSDL nào đó, có thể ra lệnh đóng để giải phóng bộ nhớ bằng cách:

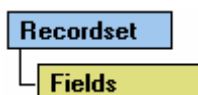
```
db.Close
```

Sau khi lệnh này thực thi, tất cả các đối tượng con của db nếu đang mở sẽ được đóng lại để giải phóng bộ nhớ. Bản thân db cũng được giải phóng bộ nhớ (bằng Nothing), tất nhiên tệp CSDL và dữ liệu vẫn còn nguyên trên đĩa.

## 1.3 Đối tượng RecordSet

Recordset là đối tượng dùng để miêu tả tập hợp các bản ghi của một bảng, của một query hoặc tập các bản ghi kết quả của việc thi hành một câu lệnh SQL nào đó.

Lớp các đối tượng con của Recordset được thể hiện qua sơ đồ sau:



## Khai báo

```
Set rs=db.OpenRecordset (<Name>)
```

## Trong đó:



- **Set rs = db.OpenRecordset** là lệnh để tạo ra tập hợp các bản ghi từ CSDL db gán vào biến kiểu recordset *rs*;
- *<Name>* là một chuỗi ký tự chỉ ra nguồn dữ liệu sẽ trả về cho Recordset. Chuỗi này có thể là tên một bảng, một Query hoặc một câu lệnh SQL;

Mỗi biến Recordset khi làm việc, phải được chỉ ra Database xuất xứ của nó (phải được tham chiếu từ một biến kiểu Database đã được khai báo). Sau đây là các ví dụ:

### Ví dụ 1:

Gán tập hợp các bản ghi từ một bảng vào biến Recordset (ở đây là bảng *canbo*).

```
Dim rs As DAO.Recordset  
  
Set rs = db.OpenRecordset ("canbo")
```

### Ví dụ 2:

Gán tập hợp các bản ghi từ một câu lệnh chọn dữ liệu SQL vào biến Recordset (ở đây là các thông tin *hoten, ngaysinh* của tất cả các cán bộ *nữ* từ bảng *canbo*).

```
Dim rs As DAO.Recordset  
  
Set rs = db.OpenRecordset ("SELECT hoten, ngaysinh FROM canbo  
WHERE gioitinh = False")
```

## Một số thuộc tính của Recordset

### Thuộc tính **Name**

Trả về chuỗi ký tự trong tham số *<name>* của lệnh gọi Recordset. Ví dụ: lệnh sau sẽ cho biết chuỗi ký tự tạo nguồn dữ liệu cho Recordset là gì?

```
MsgBox rs.Name
```

### Thuộc tính **AbsolutePosition**

Cho biết vị trí bản ghi hiện tại (được tính từ 0). Trong trường hợp không có bản ghi nào trên recordset hoặc con trỏ bản ghi đang nằm ở EOF- sẽ không thể lấy được giá trị thuộc tính này. Do vậy để sử dụng thuộc tính này thường phải đi kèm thuộc tính kiểm tra có tồn tại bản ghi nào hay không (`RecordCount > 0`) và con trỏ bản ghi có ở cuối tệp chưa (`EOF = False`).

### Thuộc tính **RecordCount**

Cho biết tổng số bản ghi trả về trên Recordset

### Thuộc tính **EOF**

Cho biết con trỏ bản ghi hiện tại có nằm ở EOF hay không? Nếu có giá trị thuộc tính này là True, trái lại là False.

### Thuộc tính **Fields**

Dùng tham chiếu tới các trường (Fields) trên tập hợp các bản ghi mà Recordset trả về. Thực tế Field cũng là một đối tượng và cũng có bộ thuộc tính và các phương thức của nó. Với Field của Recordset thông thường người ta hay sử dụng thuộc tính *Value*. Nếu không chỉ định thuộc tính cụ thể nào cho Field, VBA vẫn hiểu ngầm định đó là Value.

Ví dụ: Hiện thị giá trị trường *hoten* trong Recordset *rs*

```
Msgbox rs.Fields("hoten").Value  
  
'hoặc  
Msgbox rs.Fields("hoten")
```

## Một số phương thức của Recordset

### Phương thức **Close**

Để đóng Recordset, giải phóng bộ nhớ. Chỉ thực hiện hành động này khi không làm việc với Recordset nào đó.

*Các phương thức di chuyển bản ghi của Recordset*

### Phương thức **MoveFirst**

Để chuyển con trỏ về bản ghi đầu tiên

### Phương thức **MoveLast**

Để di chuyển con trỏ về bản ghi cuối cùng

### Phương thức **MoveNext**

Dịch đến bản ghi kế sau

### Phương thức **MovePrevious**

Dịch đến bản ghi kế trước

### Ví dụ 3:

Ví dụ sau duyệt và hiển thị toàn bộ *Hoten* của bảng *canbo*

```
Dim db As DAO.Database
Dim rs As DAO.Recordset

Set db = CurrentDb
Set rs = db.OpenRecordset ("canbo")

If rs.RecordCount > 0 Then
    rs.MoveFirst
    While rs.EOF = False
        MsgBox rs.Fields("hoten").Value
        rs.MoveNext
    Wend
End If
```

### Phương thức **AddNew, Update**

Để thêm mới một bản ghi vào Recordset. Qui trình thêm một bản ghi mới như sau:

1. Ra lệnh Addnew
2. Gán giá trị cho các trường của bản ghi mới
3. Ra lệnh Update

Dưới đây là ví dụ thêm mới một hồ sơ cán bộ mới vào bảng *canbo*.

### Ví dụ 4:

```
Dim db As DAO.Database
Dim rs As DAO.Recordset
```

```

Set db = CurrentDb
Set rs = db.OpenRecordset ("canbo")

'-----
'Ra lệnh thêm mới bản ghi
'
rs.AddNew

'-----
'Gán giá trị cho các trường của bản ghi mới
'
rs.Fields("canboID") = "CB00565"
rs.Fields("hoten") = "Nguyễn Sơn Hải"
rs.Fields("ngaysinh") = #2/11/1975#
rs.Fields("gioitinh") = True
rs.Fields("chucvuID") = "CV002"

'-----
'Ra lệnh ghi lại dữ liệu
'
rs.Update

```

### Phương thức **Edit, Update**

Phương thức Edit để sửa dữ liệu một bản ghi nào đó trên recordset. Qui trình để sửa một bản ghi như sau:

1. Định vị tới bản ghi cần sửa trên recordset
2. Ra lệnh Edit
3. Gán giá trị mới cho các trường cần sửa
4. Ra lệnh Update

Dưới đây là ví dụ về sửa hồ sơ cán bộ có mã *CB000565*

#### **Ví dụ 5:**

```

Dim db As DAO.Database
Dim rs As DAO.Recordset

Set db = CurrentDb

'-----
'Định vị tới bản ghi cần sửa
'
Set rs = db.OpenRecordset("SELECT * FROM canbo WHERE
canboID='CB000565'")

If rs.RecordCount > 0 Then

```

```

rs.MoveFirst

'-----
'Ra lệnh sửa bản ghi
'
rs.Edit

'-----
'Thực hiện sửa các trường dữ liệu cần thiết
'
rs.Fields("hoten") = "Nguyễn Văn Hải"
rs.Fields("ngaysinh") = #22/11/1975#

'-----
'Ra lệnh ghi lại dữ liệu vừa sửa
'
rs.Update
End If

```

### Phương thức **Delete**

Để xoá bản ghi hiện tại ra khỏi Recordset. Khi đó bản ghi hiện tại sẽ bị xoá bỏ khỏi CSDL. Cần thận trọng mỗi khi ra lệnh này. Thông thường các lệnh một nút Xoá bản ghi của một mẫu nhập liệu (nhập vào biến Recordset *rs*) như sau:

#### **Ví dụ 6:**

```

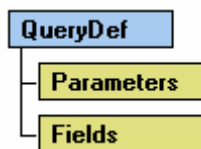
Private Sub cmDelete_Click()
    Dim tbao

    tbao = MsgBox("Đã chắc chắn xoá chưa?", vbYesNo + vbCritical)
    If tbao = vbYes Then
        rs.Delete
        rs.MoveNext
    End If
End Sub

```

## **1.4 Đối tượng QueryDef**

Đối tượng Querydef dùng để tham chiếu tới các Query có sẵn (Buil-in) trên CSDL Access, hoặc cũng có thể lập trình tạo các Query từ các câu lệnh SQL.



Để tạo và kích hoạt một query trên VBA bằng cách thực thi câu lệnh SQL bạn làm theo hướng dẫn sau:

```

'-----
'Khai báo một biến kiểu Database và một biến kiểu QueryDef

Dim db As DAO.Database
Dim qr As DAO.QueryDef

'-----
'Ra lệnh tạo một Query mới, có tên rỗng (chỉ ở trong bộ nhớ)

Set qr = db.CreateQueryDef(<tên query>)

'-----
'Gán chuỗi lệnh SQL vào thuộc tính SQL của query
'
qr.SQL = "Gõ lệnh SQL cần thi hành vào đây"

'-----
'Ra lệnh thi hành query
'
qr.Execute

'-----
'giải phóng bộ nhớ

qr.Close
  
```

#### Trong đó:

- Bắt buộc phải khai báo một biến kiểu QueryDef để làm việc (biến *qr*);
- Phải có một biến Database đã được khai báo sẵn (biến *db*);
- Lệnh **Set qr = db.CreatQueryDef(<tên query>)** để tạo một query mới lên CSDL. <tên query> sẽ được hiển thị trên danh sách trong thẻ Queries trên cửa sổ Database. Nếu <tên query>="", query này sẽ chỉ tồn tại trong bộ nhớ. Tùy thuộc vào mục đích công việc mà có đặt tên query hay không, nếu chỉ đơn thuần tạo một query để xử lý công việc nào đó rồi giải phóng, nên đặt <tên query>="";

- Lệnh **qr.SQL=<câu lệnh SQL>** để gán lệnh SQL cần thực thi vào Query. Tùy thuộc vào câu lệnh SQL này mà query sẽ thực hiện những gì. Ví dụ: **qr.SQL = "DELETE \* FROM canbo"** lệnh này sẽ xoá tất cả các bản ghi trên bảng cán bộ;
- Lệnh **qr.Excute** để thi hành câu lệnh SQL đã được thiết lập. Lệnh này tương đương nhấn nút Run đối với một query trên chế độ thiết kế;
- Lệnh **qr.Close** để đóng query hiện tại và giải phóng bộ nhớ khi không cần làm việc đến nữa.

Sau đây là một số ví dụ về sử dụng biến QueryDef để tạo ra một số loại query xử lý dữ liệu trong môi trường VBA.

**Ví dụ 1:** Tạo DELETE query để xoá danh sách những cán bộ có tuổi lớn hơn 60 ra khỏi bảng *canbo* (cán bộ đã nghỉ hưu)

```
Dim db As DAO.Database
Dim qr As DAO.QueryDef

Set db = CurrentDb

Set qr = db.CreateQueryDef("")

qr.SQL="DELETE * FROM canbo WHERE Year(Date()) - " _
& " Year(Ngaysinh)>=60"

qr.Execute

qr.Close
```

**Ví dụ 2:** Giả sử đã thêm một trường mới trên bảng cán bộ có tên *luongchinh*. Tạo UPDATE query để tính giá trị cho trường này = **hesoluong \* 290000**.

```
Dim db As DAO.Database
Dim qr As DAO.QueryDef

Set db = CurrentDb

Set qr = db.CreateQueryDef("")

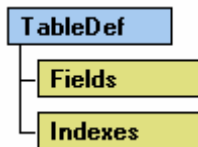
qr.SQL = "UPDATE canbo SET canbo.luongchinh = hesoluong * 290000"

qr.Execute

qr.Close
```

## 1.5 Đối tượng TableDef

Đối tượng TableDef được dùng để tham chiếu tới các bảng dữ liệu (Table) trên CSDL. Thông qua đối tượng này có thể thiết kế, chỉnh sửa được cấu trúc các bảng dữ liệu trong chế độ Run-time của VBA như trên chế độ thiết kế bảng Design View.



### Một số thuộc tính quan trọng của TableDef

Thuộc tính **Name**

Cho biết tên bảng được gán vào biến kiểu TableDef

Thuộc tính **RecordCount**

Cho biết tổng số bản ghi hiện có trên bảng được gán bởi biến TableDef

Thuộc tính **DateCreated**

Cho biết thời gian tạo ra bảng được gán vào biến kiểu TableDef

Thuộc tính **Fields**

Để tham chiếu tới các trường của bảng. Đây là thuộc tính hay được sử dụng nhất đối với TableDef. Thực chất, Field ở đây là một đối tượng, do đó cũng có tập các thuộc tính và phương thức riêng cho thuộc tính này.

Dưới đây là thủ tục hiển thị tên của tất cả các trường trong một bảng nào đó (ngầm định trên một CSDL đã được khai báo và gán biến *db* - kiểu Database).

#### Ví dụ 1:

```
Sub LietKeTenTruong(tenbang As String)
    Dim tbl As DAO.TableDef
```



```

Set tbl = db.TableDefs (tenbang)

For i = 0 To tbl.Fields.Count - 1
    MsgBox tbl.Fields(i).Name
Next

End Sub

```

## Một số phương thức của TableDef

### Phương thức **CreateTableDef**

Để tạo ra một bảng mới bằng VBA. Cú pháp tạo bảng mới như sau:

```

Set tbl = db.CreateTableDef (<Tên bảng mới>)

'-----
'...Các thủ tục tạo trường mới cho bảng
'-----

db.TableDefs.Append tbl

```

#### Trong đó:

- *db* – là biến kiểu Database đã được gán bởi CSDL cần làm việc (bảng mới sẽ được tạo ra trên CSDL này);
- <Tên bảng mới> là tên bảng cần tạo.
- Lệnh **db.TableDefs.Append tbl** là lệnh ghi cấu trúc bảng đang khai báo lên CSDL đã chỉ định.

### Phương thức **CreateField**

Để tạo ra các trường cho một bảng kiểu TableDef nào đó. Để thêm một trường mới lên bảng, sử dụng cú pháp sau:

```
tbl.Fields.Append tbl.CreateField (<tên trường>, <KiểuDL>, <độ lớn>)
```

#### Trong đó:

- <tên trường> - tên trường mới cần tạo;
- <KiểuDL> - là một tùy chọn để khai báo kiểu dữ liệu của trường cần tạo. Kiểu dữ liệu được khai báo theo các hằng số như sau:

**Giá trị: Tương ứng với kiểu**

<i>dbBoolean</i>	Boolean
<i>dbByte</i>	Byte
<i>dbChar</i>	Char
<i>dbCurrency</i>	Currency
<i>dbDate</i>	Date/Time
<i>dbDecimal</i>	Decimal
<i>dbDouble</i>	Double
<i>dbFloat</i>	Float
<i>dbGUID</i>	GUID
<i>dbInteger</i>	Integer
<i>dbLong</i>	Long
<i>dbMemo</i>	Memo
<i>dbNumeric</i>	Numeric
<i>dbSingle</i>	Single
<i>dbText</i>	Text
<i>dbTime</i>	Time

- <Độ lớn> là một tùy chọn để khai báo độ lớn dữ liệu nếu cần.

Tiếp theo là ví dụ minh họa cách tạo cấu trúc một bảng dữ liệu tổng hợp những hướng dẫn đã trình bày trên.

**Ví dụ 2:**

```
Sub TaoBangMoi()  
On Error GoTo Loi  
    Dim tbl As DAO.TableDef  
  
    Set tbl = db.CreateTableDef("NewTable")  
  
    tbl.Fields.Append tbl.CreateField("ID", dbInteger)  
    tbl.Fields.Append tbl.CreateField("Name", dbText)  
    tbl.Fields.Append tbl.CreateField("Age", dbByte)  
    tbl.Fields.Append tbl.CreateField("DateBirth", dbDate)  
    tbl.Fields.Append tbl.CreateField("Comment", dbMemo)  
  
    db.TableDefs.Append tbl  
  
Exit Sub
```

```

Loi:
    If Err.Number = 3010 Then
        MsgBox "Đã tồn tại bảng có tên " + tbl.Name
    End If
End Sub

```

## 1.6 Đối tượng Relation

Đối tượng Relation dùng để tạo kết nối (Relationship) giữa 2 bảng trong CSDL Access. Dưới đây là một ví dụ tạo kết nối giữa 2 bảng hoadon và khách trong CSDL Quản lý bán hàng.

```

Sub CreatRelationship()
On Error GoTo Loi
    Dim db As DAO.Database
    Dim rls As DAO.Relation

    Set db = CurrentDb

    Set rls = db.CreateRelation("TaoQuanHe", "khach", "hoadon",
dbRelationUpdateCascade)

    rls.Fields.Append rls.CreateField("khachID")
rls.Fields("khachID").ForeignName = "khachID"
db.Relations.Append rls

Loi:
    If Err.Number = 3012 Then
        MsgBox "Đã tồn tại quan hệ này !"
    End If

End Sub

```

Trong trường hợp đã tồn tại kết nối này, một thông báo lỗi tiếng Việt "*Đã tồn tại quan hệ này !*" xuất hiện.

## 2. Bài toán đặt lọc dữ liệu

Đặt lọc là lớp bài toán phổ dụng trong thực tế. Với bài toán này phải có những yêu cầu cụ thể về lọc dữ liệu (điều kiện lọc). Kết quả trả về sẽ là một tập hợp các bản ghi, có thể được kết xuất trên form hoặc được in ra máy in dưới dạng report.

Dưới đây là minh hoạ một bài toán đặt lọc sau khi đã được xây dựng và đang hoạt động:

Chọn khách cần xem: Nguyễn Đức Ngộ

**Các hoá đơn đã mua:**

	hoadonID:	ngayban:	tongtien:
▶	1	12/10/2004	893
	2	24/10/2004	114

Record: 1 of 2

Qui trình xây dựng một bài toán đặt lọc có thể thực hiện như sau:

**Bước 1:** Xây dựng form con - form sẽ chứa những kết quả lọc được;

**Bước 2:** Xây dựng form mẹ - form chứa những thiết lập điều kiện để lọc;

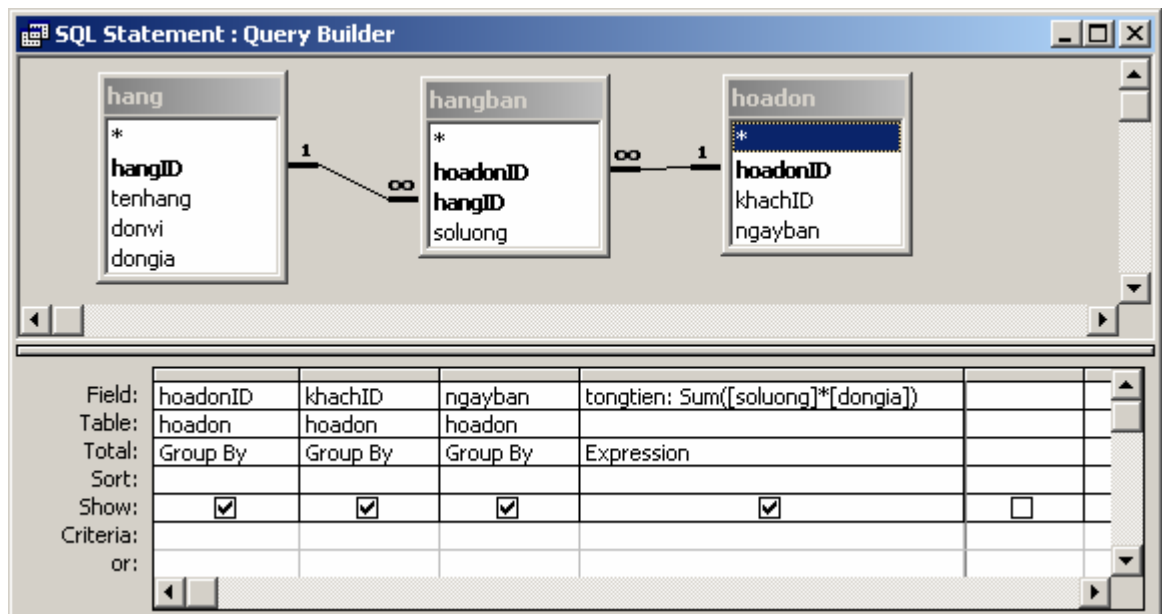
**Bước 3:** Thực hiện lọc ra các bản ghi thoả mãn các điều kiện trên form mẹ và hiển thị kết quả lên form con.

Với bài toán trên, cách giải quyết như sau:

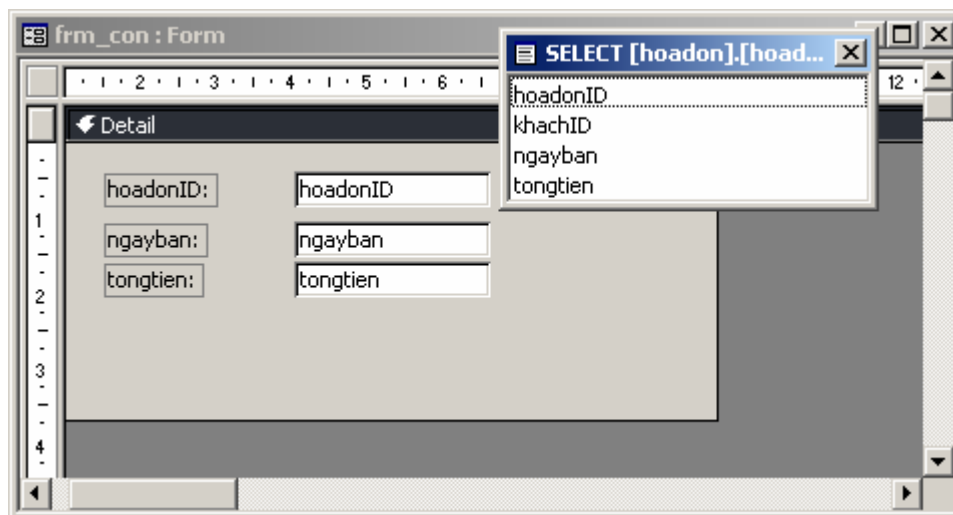
**Bước 1: Xây dựng Form con**

Sử dụng các kỹ năng thông thường để tạo một form con đáp ứng được các kết quả theo như bài toán. Cụ thể từng bước như sau:

- Tạo mới form ở chế độ Design view;
- Thiết lập thuộc tính Record Source cho form là một Total Query như sau:



- Thiết lập thuộc tính Default View cho form con là *Datasheets*;
- Mở cửa sổ FieldList và thiết kế form như sau:



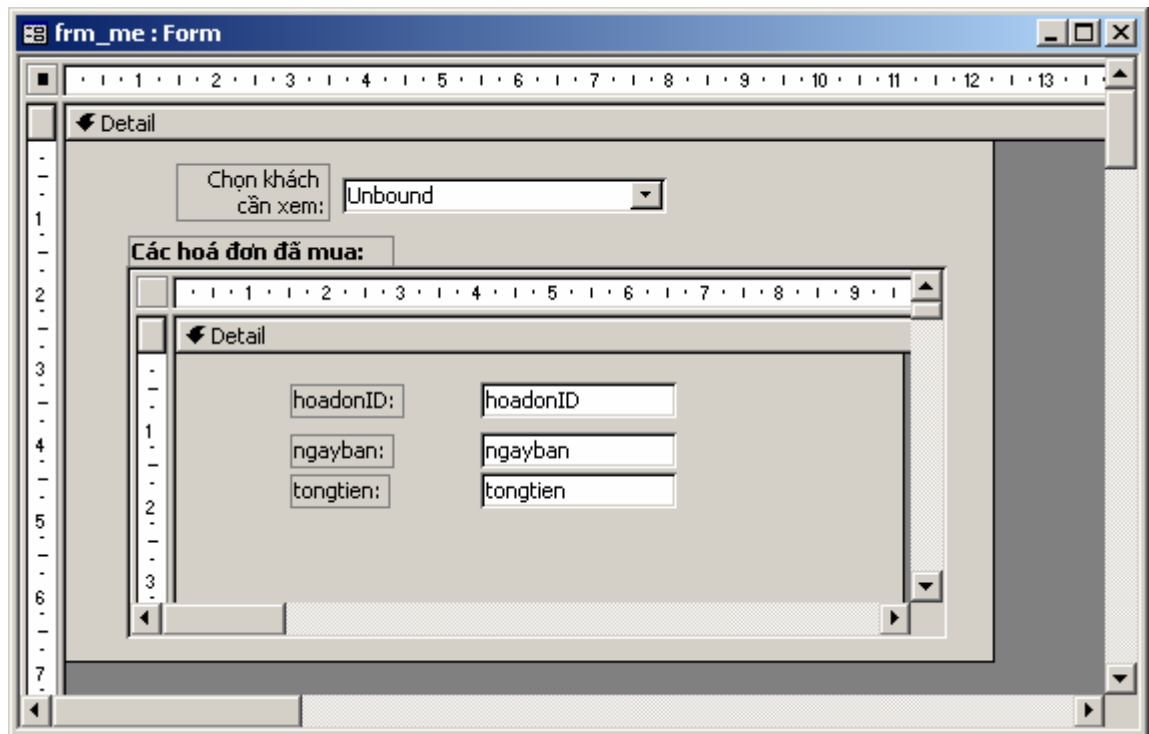
- Đóng thiết kế form con lại và ghi tên là *frm\_formcon*.

## Bước 2: Xây dựng form mẹ

- Tạo mới form ở chế độ Design view;
- Đưa hộp Combobox từ thành công cụ Toolbox lên form mẹ (giả sử tên (Name) của Combo này là *Combo0*). Sử dụng tính năng Combobox Wizard của Access để làm việc này. Mục đích là đưa danh sách các khách hàng từ bảng khách vào hộp Combo để chọn, phục vụ nhu cầu lọc dữ liệu;
- Sử dụng đối tượng Sub-form/Sub-report trên thanh công cụ Toolbox để đưa form con vừa tạo lên form mẹ. Ngầm định tên của sub-form này là

frm\_formcon. Chú ý đến tên gọi này vì phần tiếp sau sẽ sử dụng nó để lập trình.

Cuối cùng, giao diện thiết kế form mẹ như sau:



### Bước 3: Thiết lập lệnh lọc dữ liệu trên form mẹ

Công việc của bước này là làm sao để sau khi chọn tên một khách hàng ở hộp Combobox, danh sách các hoá đơn mua hàng của khách đó sẽ được hiển thị lên form con. Muốn thế, việc lập trình lọc dữ liệu ở đây phải được thực hiện trong thủ tục đáp ứng sự kiện **Combo0\_Click**. Giải thuật sẽ là:

- Tạo một biến Recordset để thi hành câu lệnh SQL đưa ra danh sách kết quả thoả mãn điều kiện đặt lọc. Câu lệnh SQL trong trường hợp này là:

```
"SELECT hoadonID, kháchID, ngayban, Sum([soluong]*[dongia]) " _
+ " AS tongtien FROM " _
+ " hoadon INNER JOIN (hang INNER JOIN hangban ON " _
+ " hang.hangID = hangban.hangID) ON hoadon.hoadonID =" _
+ " hangban.hoadonID WHERE Trim(kháchID)='"+Trim(Combo0)'" _
+ " GROUP BY hoadonID, kháchID, ngayban "
```

- Gán thuộc tính Recordset của form con là biến kiểu recordset vừa tạo ra (chứa kết quả đã lọc);
- Ra lệnh làm tươi dữ liệu cho form con.

Tất nhiên, trước đó phải khai báo một biến kiểu Database toàn cục trong form và định nghĩa nó ở thủ tục Form\_Load()

Toàn bộ mã lệnh cho bài toán trên như sau:

```
Dim db As DAO.Database

Private Sub Form_Load()
    Set db = CurrentDb
End Sub

Private Sub Combo0_Click()
    Dim rs As DAO.Recordset

    Set rs = db.OpenRecordset("SELECT hoadonID, khachID, " _
    + " ngayban, Sum([soluong]*[dongia]) AS tongtien FROM" _
    + " hoadon INNER JOIN (hang INNER JOIN hangban ON " _
    + " hang.hangID = hangban.hangID) ON hoadon.hoadonID =" _
    + " hangban.hoadonID WHERE Trim(khachID)='"&Trim(Combo0)'" _
    + " GROUP BY hoadonID, khachID, ngayban ")

    Set frm_formcon.Form.Recordset = rs

    frm_formcon.Requery

End Sub
```

Tóm lại, với mỗi bài toán đặt lọc áp dụng theo phương pháp kể trên đều giống nhau về cách làm việc, điểm khác nhau duy nhất là câu lệnh SQL. Có thể kết luận mấu chốt của bài toán này là tạo ra được lệnh SQL đúng để đáp ứng mọi điều kiện đặt lọc từ form mẹ.



### Chú ý

Việc viết ra câu lệnh SQL đúng là rất khó, đòi hỏi lập trình viên phải có nhiều kinh nghiệm. Để làm tốt được việc này, nên thiết kế một query ở chế độ design view; rồi sử dụng tính năng Copy, Paste để dán câu lệnh SQL mà query đã tạo lên nơi soạn thảo lệnh VBA và chỉnh sửa cho phù hợp. Thông thường sửa mệnh đề WHERE của câu lệnh.

# CHƯƠNG 7

## MENU & TOOLBAR

Chương này trình bày những kỹ thuật tạo menu và toolbar nhằm liên kết toàn bộ những gì đã làm được thành một hệ thống phần mềm hoàn chỉnh. Bao gồm các mảng kiến thức:

- Tạo Menu;
- Tạo Toolbar;
- Tạo form chính phần mềm



## 1. Tạo Menu

Menu là đối tượng được thiết kế có cấu trúc nhằm gắn kết các chức năng phần mềm lại để người dùng thuận tiện trong việc khai thác các tính năng cũng như CSDL.

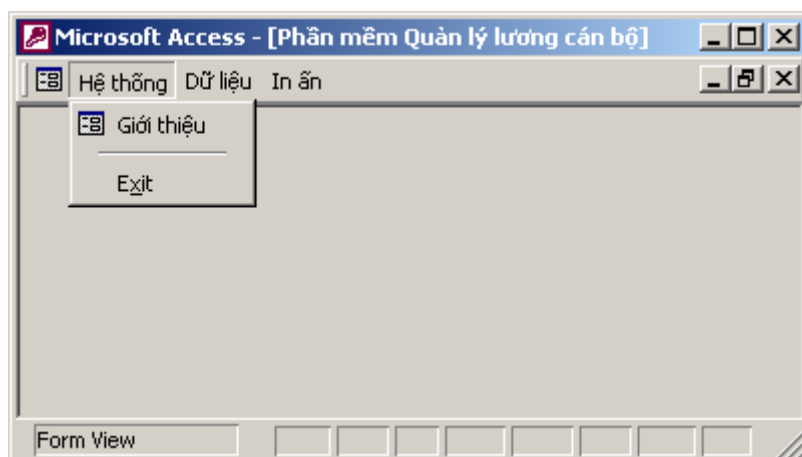
Có nhiều cách tạo menu trong Access, dưới đây chúng tôi trình bày cách đơn giản, dễ tạo và hiệu quả, đó là: sử dụng khả năng Customize của thanh công cụ.

Trước khi tạo menu, bạn phải đảm bảo đã xây dựng xong đầy đủ các tính năng cần thiết của phần mềm; bước này sẽ là cuối cùng, mang tính chất gắn kết các chức năng lại thành phần mềm hoàn chỉnh.

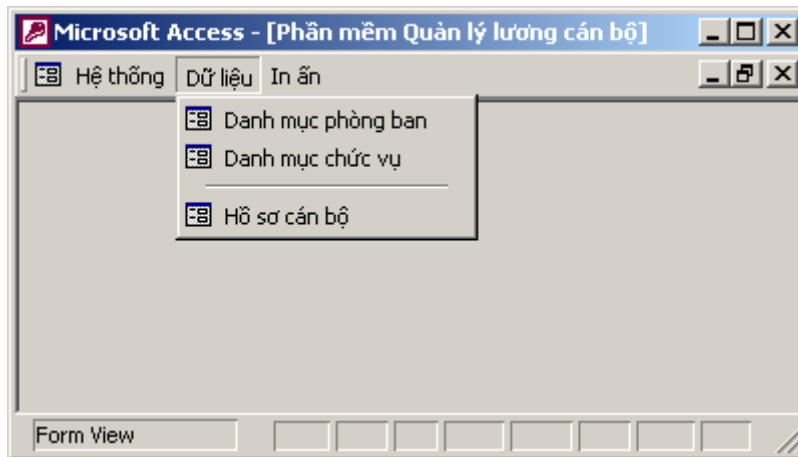
Giả sử trong CSDL Quản lý lương cán bộ đã làm được những việc sau:

1. Đã tạo xong CSDL
2. Đã tạo form nhập dữ liệu cho bảng *chucvu*, form nhập dữ liệu cho bảng *phongban*, form nhập dữ liệu cho bảng *canbo*;
3. Đã tạo xong report để in ra Danh sách chức vụ, Danh sách phòng ban;
4. Đã tạo xong form tìm kiếm cán bộ theo tên, in danh sách cán bộ một phòng ban
5. Đã tạo xong form để giới thiệu phần mềm (frmAbout)

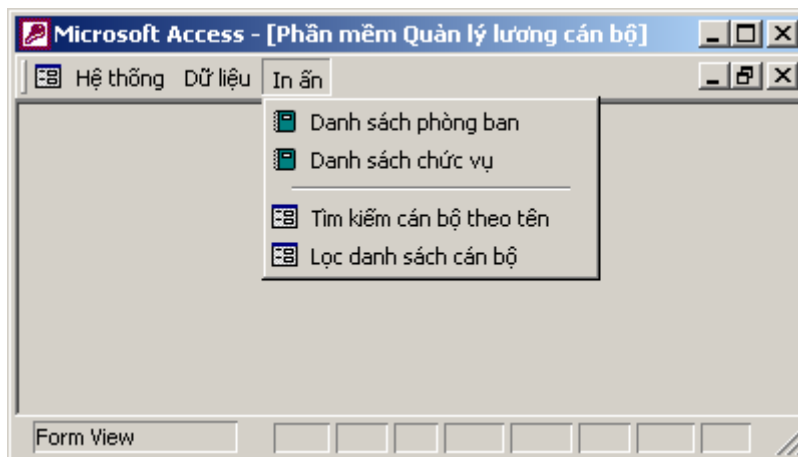
Dưới đây là hướng dẫn để tạo ra hệ thống menu có cấu trúc như sau:



(các mục chọn của menu Hệ thống)



(Các mục chọn của menu Dữ liệu)

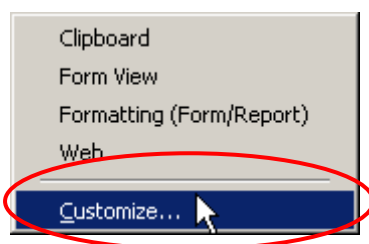


(Các mục chọn của menu In ấn)

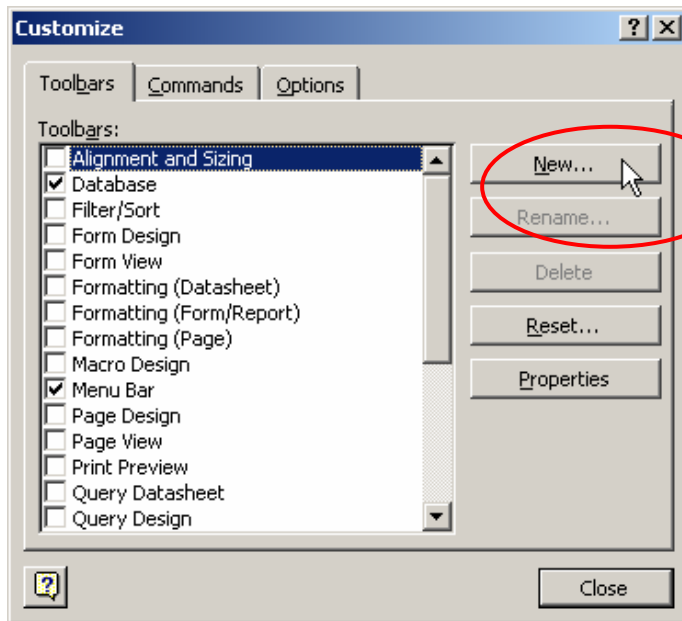
Các bước để tạo hệ thống menu trên như sau:

**Bước 1:** Tạo một Toolbar mới

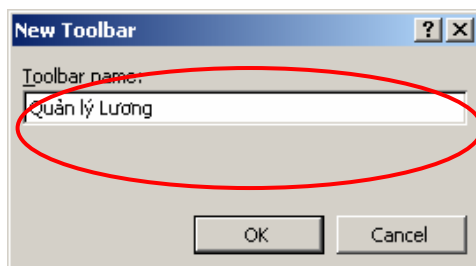
Nhấn chuột phải lên thanh công cụ của Access, chọn Customize



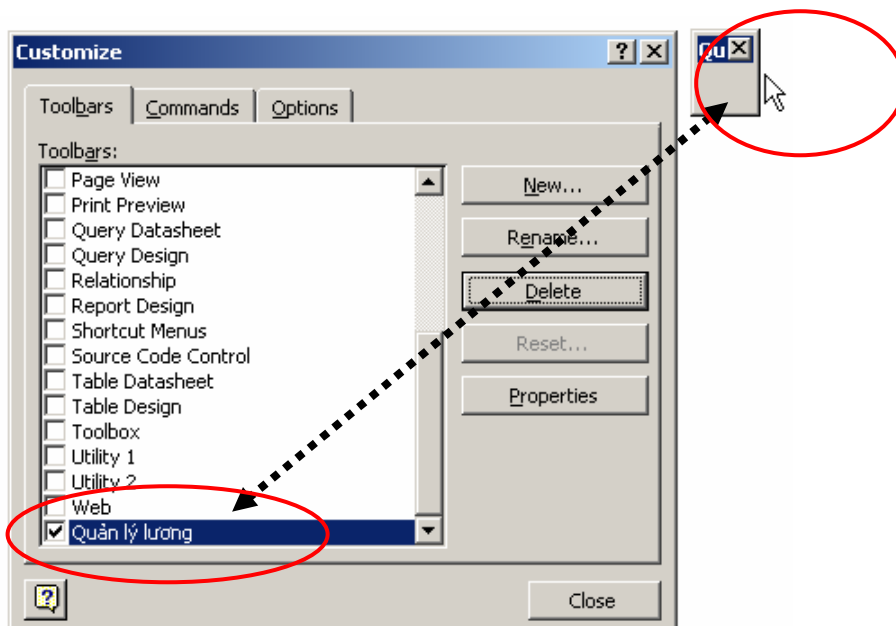
Hộp thoại **Customize** xuất hiện:



Để tạo mới một thanh công cụ, nhấn nút New. Hộp thoại sau xuất hiện, yêu cầu gõ vào tên cho thanh công cụ:

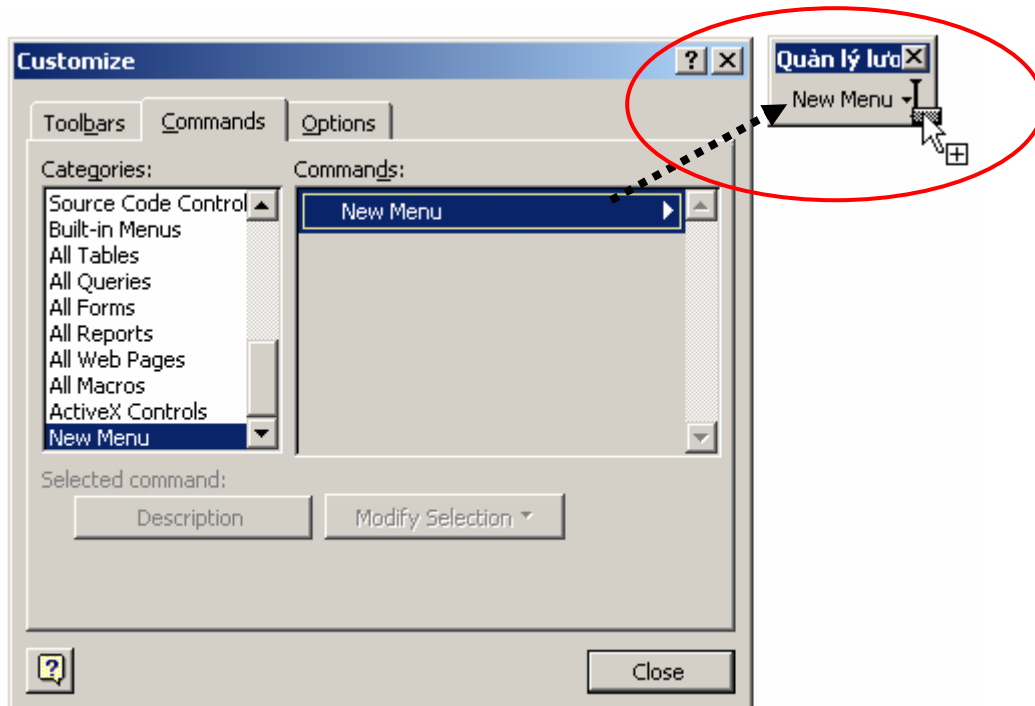


Hãy gõ tên cho thanh công cụ, giả sử *Quản lý lương*. Gõ xong nhấn **OK**. Khi đó mà hình làm việc xuất hiện một thanh công cụ trắng. Công việc tiếp theo là xây dựng hệ thống Menu trên thanh công cụ này.



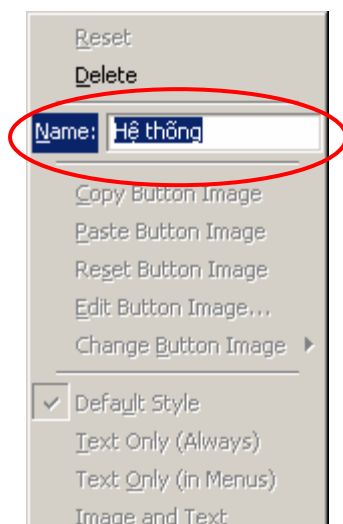
## Bước 2: Xây dựng các mục cấp 1

Các mục cấp 1 là: *Hệ thống*, *Dữ liệu* và *In ấn*. Để xây dựng các mục này, trên hộp thoại Customize mở thẻ **Commands**. Ở danh sách **Categories** chọn **New Menu** và lần lượt kéo mục **New Menu** trên danh sách Commands thả lên Toolbar đang thiết kế:



Hãy lần lượt kéo thả đủ số menu cấp 1 (3 menu cấp 1). Tiếp theo lần lượt thay đổi tiêu đề cho 3 menu này bằng cách:

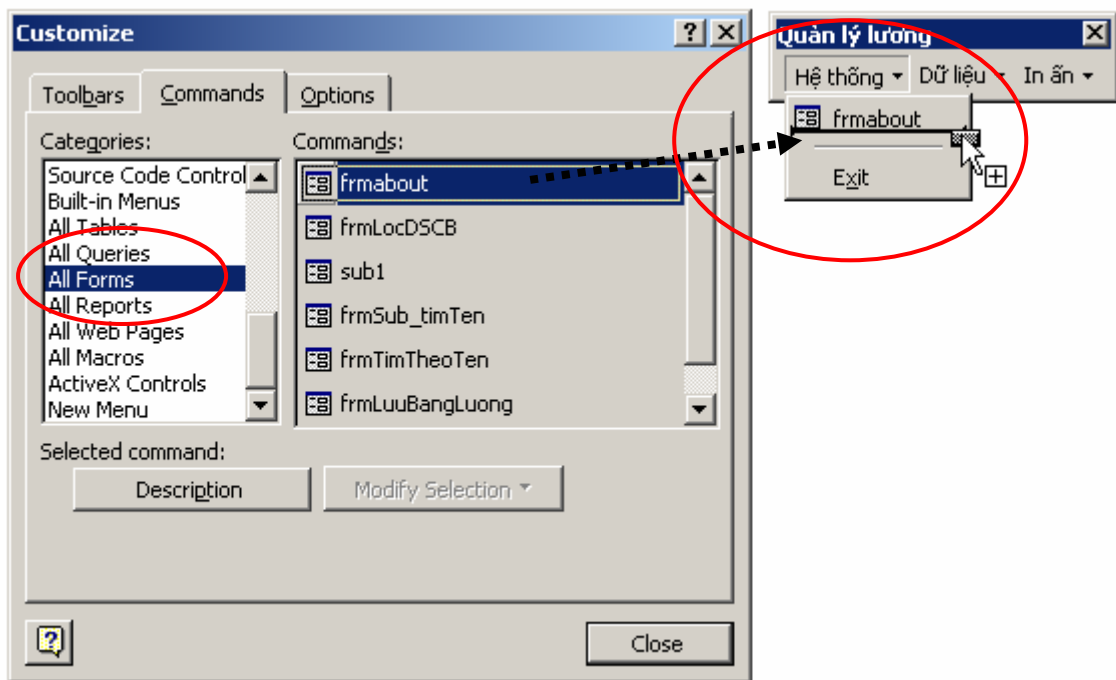
- Nhấn chuột phải lên menu cần thay đổi;
- Gõ tiêu đề mới vào hộp *Name* (hình dưới):



### Bước 3: Xây dựng các mục cấp con

3 mục tạo trên là cấp 1, các mục còn lại nằm trong 3 mục đó đều là mục con, cháu. Bước này sẽ xây dựng toàn bộ các mục con như vậy.

Mỗi mục con có thể là lời gọi một Form, một Report, một bảng,... (gọi tắt là đối tượng) ra để làm việc. Muốn mở đối tượng nào, hãy xác định chúng loại đối tượng đó bên danh sách **Categories**; tiếp theo dùng chuột kéo-thả đối tượng cần đưa lên menu từ danh sách **Commands** lên vị trí xác định trên menu đang thiết kế.



Hình trên minh họa cách đưa lời gọi đến form *frmAbout* lên menu *Hệ thống*.

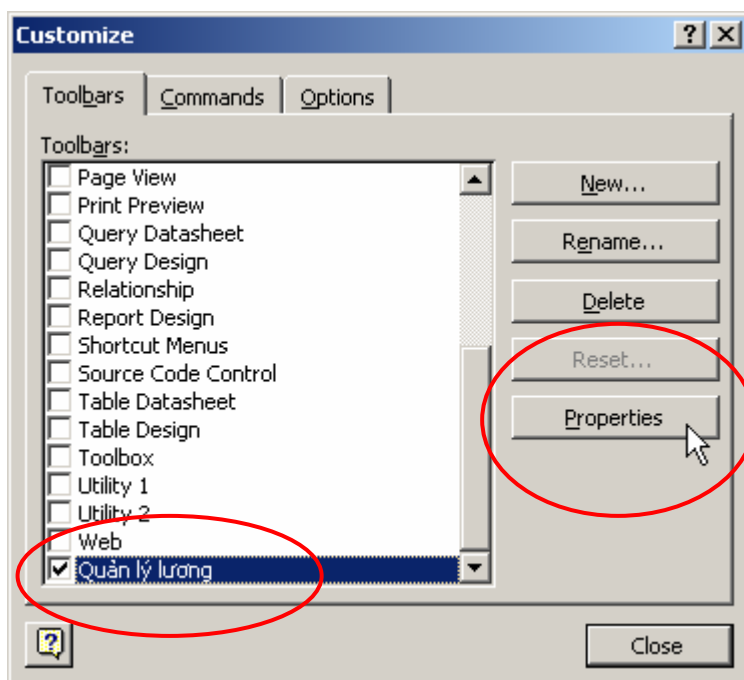
Tương tự hãy đưa các form và report còn lại lên vị trí thích hợp trên menu *Quản lý lương*. Cách đặt tên cho các menu này thực hiện tương tự như cách đặt tên cho 3 menu mẹ đã trình bày.

Đặc biệt mục *Exit* lấy ở **File | Exit**.

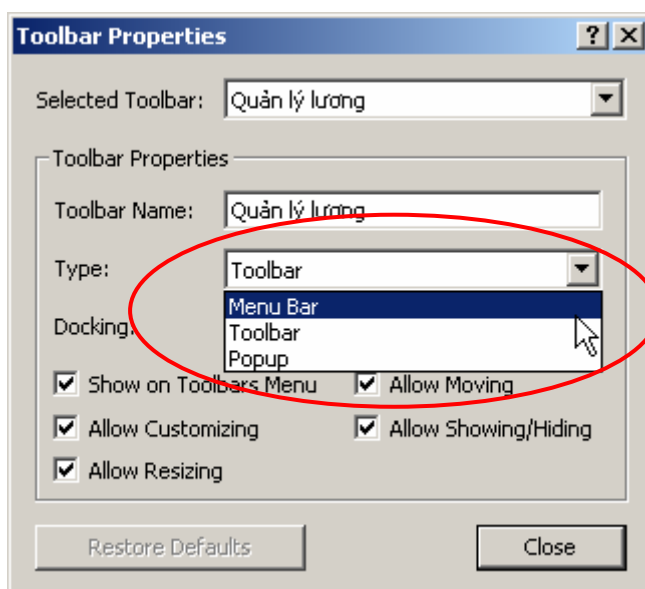
### Bước 4: Quyết định là Menu hay ToolBar

Với các bước làm việc như trên, Access ngầm hiểu là đang tạo một ToolBar. Bước này hướng dẫn cách tùy chọn lại là menu hay toolbar? Cách làm như sau:

Trở về thẻ ToolBars của hộp thoại *Customize*, chọn tên toolbar cần làm việc (*Quản lý lương*) và nhấn nút **Properties**:



Hộp thoại Toolbar Properties xuất hiện cho phép thiết lập các thuộc tính cho Toolbar đang chọn:



- **Toolbar Name** - để thiết lập tên cho Toolbar;
- **Type** - để xác định kiểu là Toolbar hay Menu?
- **Docking** - để tùy chọn các kiểu khoá Menu: không cho phép thay đổi lại (Can't Change); cho phép thay đổi lại Any Allow);
- **Show on Toolbars Menu** – cho phép hiển thị trên hệ thống thanh công cụ và menu;
- **Allow Customizing** – cho phép định nghĩa lại các mục;

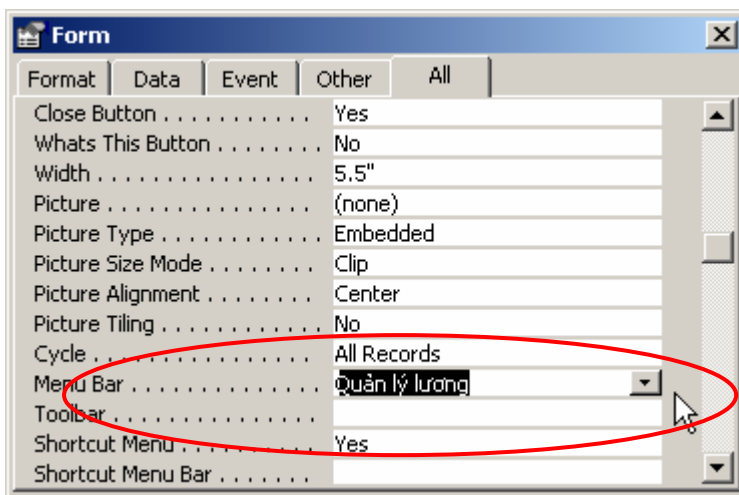
- **Allow Resizing** – cho phép thay đổi kích cỡ hiển thị;
- **Allow Moving** – cho phép di chuyển được;

Thiết lập xong nhấn **Close**.

## 2. Gắn kết Menu, Toolbar

Một Menu hay Toolbar sau khi đã tạo ra, muốn đi kèm với đối tượng nào phải thực hiện gắn kết vào đối tượng đó bằng cách thiết lập thuộc tính Toolbar - nếu muốn gắn Toolbar hoặc Menu Bar – nếu muốn gắn menu bar. Cách làm như sau:

**Bước 1:** Mở đối tượng cần gắn kết Menu hoặc ToolBar (ví dụ một form) ra chế độ Design View:



**Bước 2:** Chọn Menu Bar hoặc ToolBar cần gắn kết bằng cách thiết lập thuộc tính **ToolBar** hoặc **Menu Bar** (hình trên chọn Menu Bar *Quản lý lương*).

## 3. Tạo form chính

Form chính là form chứa hệ thống menu (hoặc toolbar) của phần mềm, nó xuất hiện ngay sau khi khởi động (Start-Up Object). Với mỗi phần mềm đóng gói hầu hết phải tạo form chính. Để từ đây có thể mở đến các chức năng phần mềm cần làm việc.

Trong một tệp Access, form chính là một form được thiết kế ở chế độ Design view; có gắn Menu (hoặc ToolBar) và được thiết lập khởi động đầu tiên mỗi khi tệp này được mở.

Dưới đây là hướng dẫn cách tạo form chính cho CSDL Quản lý lương như thể hiện ở mục 1. Tạo menu

### Bước 1: Tạo frmMain

- Tạo một form mới ở chế độ Design View;
- Đặt tên form này là *frmMain*;
- Thiết lập một số thuộc tính cho form như sau:

Menu Bar *Quản lý lương*

Caption *Quản lý lương*

Record Selector *No*

Navigation Button *No*

Diving Line *No*

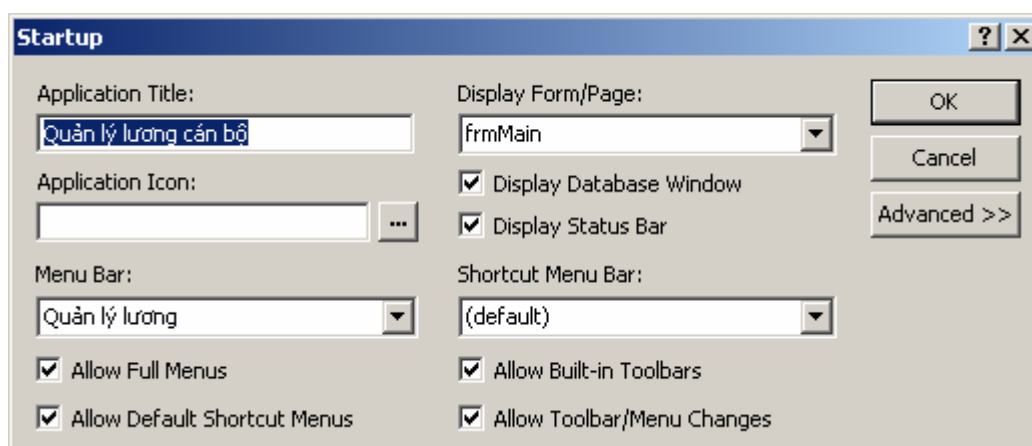
Để mỗi khi form này khởi động sẽ tự động phóng cực đại cửa sổ, hãy mở cửa sổ VBA và viết lệnh cho sự kiện **Form\_Open** như sau:

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

Ý nghĩa lệnh **DoCmd.Maximize** là phóng to (Maximize) form đang làm việc.

### Bước 2: Thiết lập các thông tin về ứng dụng

Mở thực đơn **Tools | Startup...**, hộp thoại **Startup** xuất hiện:

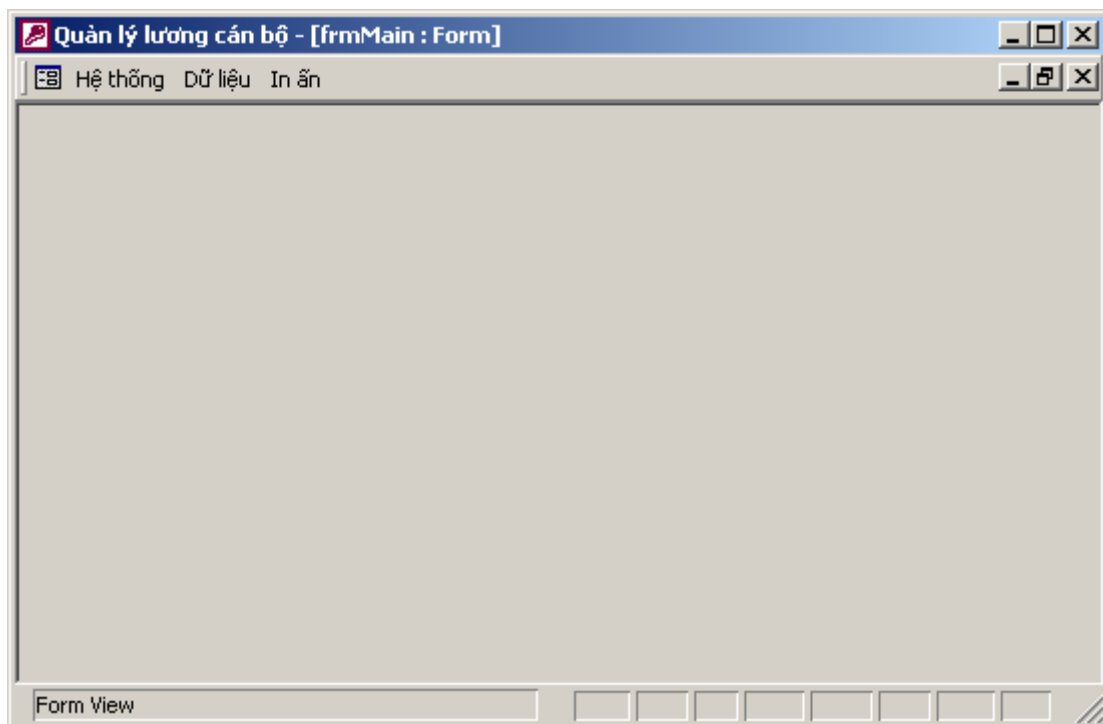


- Gõ tiêu đề phần mềm vào ô **Application Title**;



- Chọn form chính ở hộp **Display Form/Page** (chọn *frmMain*);
- Chọn Menu cho ứng dụng ở hộp **Menu Bar** (chọn menu Quản lý lương);
- Hộp **Application Icon** để chọn một biểu tượng cho ứng dụng (loại tệp \*.ico);
- **Display Database Window** Cho phép hiển thị cửa sổ Database hay không?;
- **Display Status Bar** Cho phép hiển thị thanh trạng thái?;
- **Allow Toolbar/Menu Changes** Cho phép thay đổi menu bar và toolbar?;
- ...

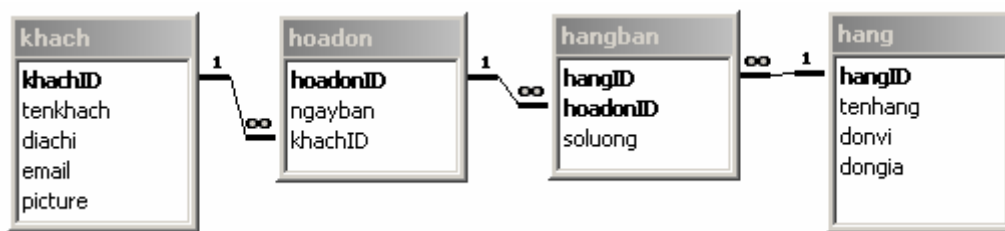
Thiết lập xong nhấn **OK** để đóng lại. Lần sau khi mở tệp CSDL này, màn hình đầu tiên bạn gặp là form *frmMain*:



## Bài tập

### Bài số 1: Xây dựng trên CSDL Quản lý bán hàng

1. Xây dựng CSDL theo cấu trúc như sau :



#### Yêu cầu :

- Thiết kế cấu trúc các bảng một cách phù hợp : kiểu dữ liệu các trường; trường khoá; thuộc tính Lookup và các thuộc tính khác;
- Thiết lập quan hệ cùng các thuộc tính đảm bảo toàn vẹn dữ liệu cho các quan hệ;

2. Tạo các form làm giao diện nhập dữ liệu cho các bảng :

**Lập hoá đơn bán hàng**

hoadonID:  Ngày bán:

Tên khách:

Địa chỉ:

E-mail:

**Chi tiết hàng bán:**

	hangID	donvi	dongia	soluong	thanhtien:
	Floppy disk	chiếc	8	3	24
	Hard disk	ổ	67	9	603
▶	CPU P4 2.4	con	133	2	266
*					

Record:  of 3

Tổng tiền:

Tiến > < Lùi Thêm mới Xoá HĐ In HĐ Thoát

Yêu cầu chi tiết:

- Các nút Tiến, Lùi để định vị hoá đơn cần làm việc;
- Nút Thêm mới để bắt đầu tạo mới một hoá đơn;
- Nút Xoá HĐ để xoá hoá đơn hiện tại. Yêu cầu phải có xác nhận trước khi xoá;
- Nút In HĐ để in chi tiết hoá đơn bán hàng ra report.

3. Thiết kế form tổng hợp thông tin bán hàng theo ngày như sau:

The screenshot shows a form window titled "Tổng hợp thông tin bán hàng..". The main title is "TỔNG HỢP THÔNG TIN BÁN HÀNG". Below the title, there are two date input fields: "Từ ngày: 1/1/2004" and "Đến ngày: 12/1/2004". Underneath, the text "Chi tiết hàng đã bán:" is followed by a table with the following data:

	Tên hàng:	Đơn vị:	Tổng số bán:
▶	CPU P4 2.4	con	15
	Floppy disk	chiếc	11
	Hard disk	ổ	9
	RAM 256	thanh	15

At the bottom of the form, there is a record navigation bar showing "Record: 1 of 4" with navigation buttons for first, previous, next, and last records.

Sau khi nhập xong giá trị từ ngày và đến ngày, thông tin chi tiết về các mặt hàng bán ra trong phạm vi ngày đó được tổng hợp theo như giao diện trên.

4. Thiết kế form tổng hợp thông tin bán hàng cho khách hàng theo ngày như sau:

**TỔNG HỢP THÔNG TIN BÁN HÀNG**

Chọn khách hàng: Nguyễn Đức Ngô

Từ ngày: 1/1/2004 Đến ngày: 12/1/2004

**Chi tiết hàng đã bán:**

	Tên hàng:	Đơn vị:	Tổng số bán:
▶	CPU P4 2.4	con	5
	Floppy disk	chiếc	3
	Hard disk	ổ	9
	RAM 256	thanh	15

Record: 1 of 4

Yêu cầu: Sau khi chọn một khách hàng, nhập vào khoảng ngày cần tổng hợp. Thông tin tổng hợp về từng mặt hàng của khách đó mua sẽ được liệt kê.

5. Tạo form để theo dõi thông tin chi tiết bán hàng theo ngày như sau:

**THEO DÕI CHI TIẾT HÀNG BÁN**

Từ ngày: 1/10/2004 Đến ngày: 30/10/2004

**Chi tiết hàng bán:**

	tenhang:	10/10/2004:	12/10/2004:	24/10/2004:
▶	CPU P4 2.4	3	12	
	Floppy disk		11	
	Hard disk		9	
	RAM 256	12		3

Record: 1 of 4

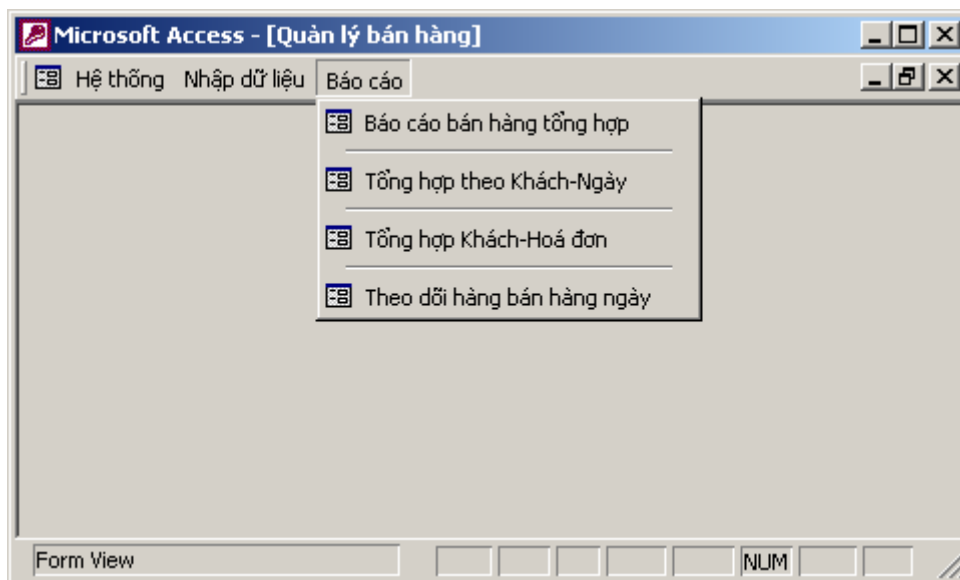
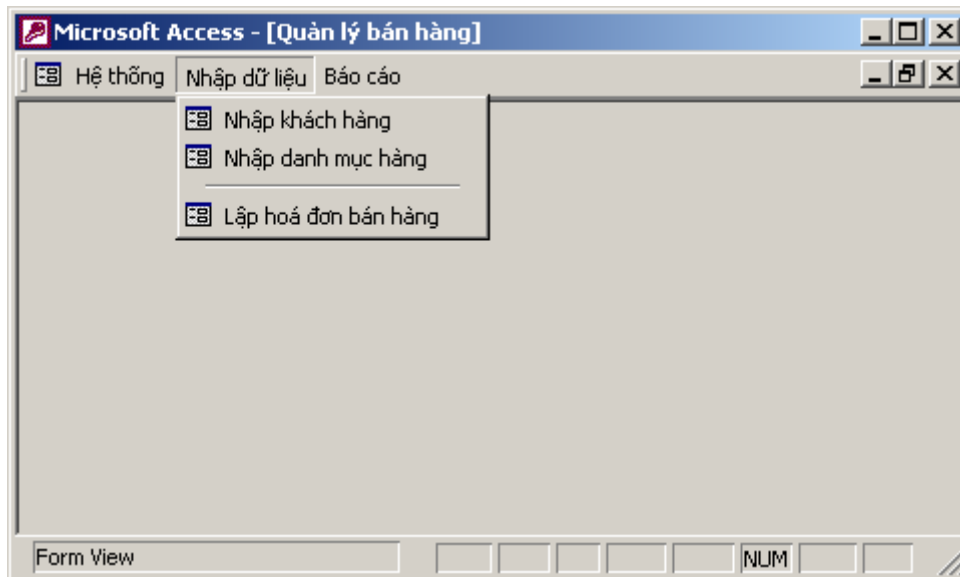
6. Tạo form để theo dõi hoá đơn mua hàng các khách hàng như sau:

	hoadonID:	ngayban:	Tongtien:
▶	1	12/10/2004	893
	2	24/10/2004	114
	3	10/10/2004	855
	4	12/10/2004	1394

Sau khi chọn tên một khách, thông tin về các hoá đơn mua hàng được hiển thị.

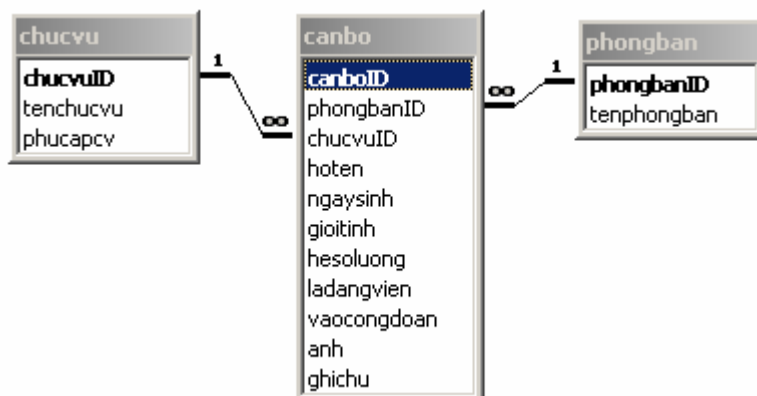
7. Tạo form frmAbout đưa những thông tin giới thiệu về phần mềm này.

8. Thiết kế form chính và menu bar để liên kết toàn bộ các chức năng đã làm được ở trên CSDL Quản lý bán hàng như sau:



**Bài số 2:** Xây dựng trên CSDL Quản lý lương cán bộ

1. Xây dựng CSDL như sau:



**Yêu cầu :**

- Thiết kế cấu trúc các bảng một cách phù hợp : kiểu dữ liệu các trường; trường khoá; thuộc tính Lookup và các thuộc tính khác;
- Thiết lập quan hệ cùng các thuộc tính đảm bảo toàn vẹn dữ liệu cho các quan hệ;

## 2. Thiết kế form làm giao diện nhập dữ liệu cho các bảng:

*frmPhongban* - cập nhật dữ liệu danh sách phòng ban;

*frmChucvu* - cập nhật danh mục chức vụ.

## 3. Thiết kế form để cập nhật hồ sơ cán bộ như sau:

**QUẢN LÝ HỒ SƠ CÁN BỘ**

Mã cán bộ:

Họ tên:

Ngày sinh:

Là nam giới

Phòng ban:

Chức vụ:

Hệ số lương:

Là Đảng viên

Vào Công đoàn

Ghi chú:

Record:      of 12

3. Tạo form cho phép xem và in danh sách cán bộ từng phòng ban như sau:

Chọn phòng ban:

**Danh sách cán bộ:**

	Họ tên	Ngày sinh	Giới tính	Chức vụ
▶	Nguyễn Đức Ngô	12/04/1976	Nam	Giám đốc
	Phan Quang Sáng	03/09/1976	Nữ	Nhân viên
	Nguyễn Văn Hoà		Nữ	Nhân viên
	Phan Hoài Linh		Nam	Nhân viên
	Phan Anh Quân	23/07/1978	Nữ	Trưởng phòng
*				

Record:       of 5

4. Tạo form cho phép tìm kiếm cán bộ theo tên như sau:

Nhập tên cán bộ muốn tìm:

**Danh sách cán bộ tìm thấy:**

	hoten	ngaysinh	tenchucvu	tenphongban
▶	Nguyễn Đức Ngô	12/04/1976	Giám đốc	Kinh doanh
	Trần Văn Ngô		Nhân viên	Tổng hợp
*				

Record:       of 2

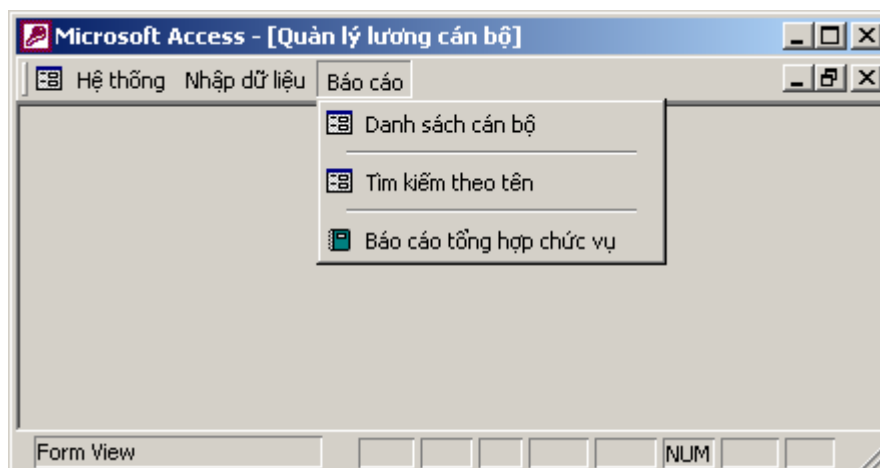
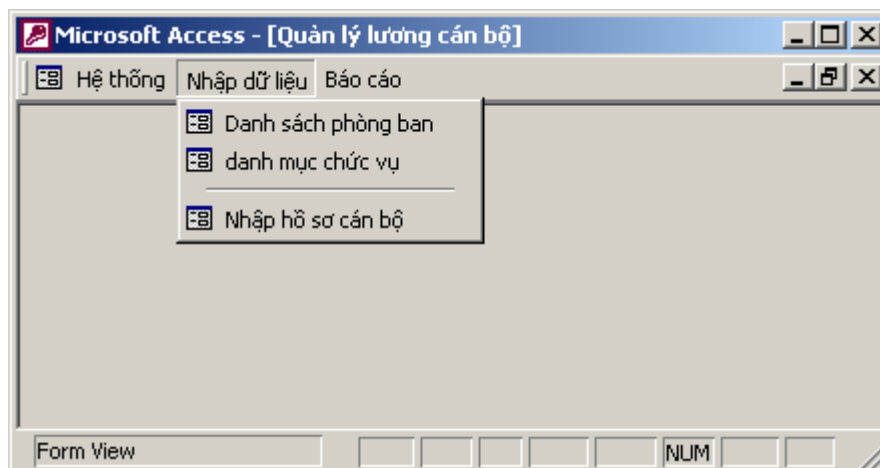
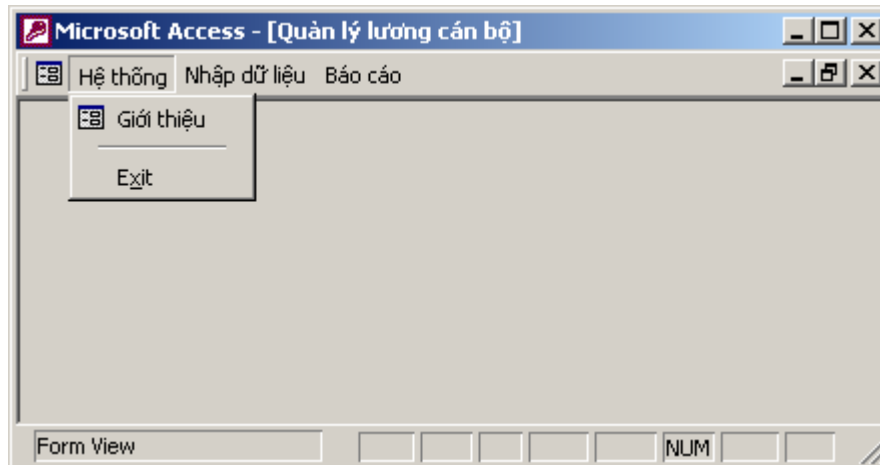
Sau khi gõ một tên (có thể là đệm + Tên hoặc đầy đủ họ và tên), danh sách các kết quả tìm thấy sẽ được liệt kê lên Subform.

5. Tạo report in ra báo cáo tổng hợp tổng số cán bộ theo chức vụ của từng phòng ban.



6. Tạo form frmAbout để giới thiệu các thông tin về phần mềm.

7. Tạo form chính cùng hệ thống menu gọi tới tất cả các chức năng đã làm đwoj như sau:



# THUẬT NGỮ TIN HỌC

Dưới đây là danh sách một số các thuật ngữ tin học hay gặp phải trong khi học và sử dụng Microsoft Access được xếp theo thứ tự từ điển để bạn đọc tham khảo.

<b>Append</b>	Chỉ hành động chèn thêm các bản ghi lên các bảng trong cơ sở dữ liệu
<b>Child</b>	Quan hệ con. Thuật ngữ này thường đi với Master – quan hệ cha
<b>Column</b>	Cột trong bảng dữ liệu
<b>Data</b>	Dữ liệu, dữ kiện, số liệu
<b>Data integrity</b>	Tính toàn vẹn dữ liệu. Một yếu tố rất quan trọng khi thiết kế một cơ sở dữ liệu.
<b>Data source</b>	Nguồn cung cấp dữ liệu, có thể là cung cấp cho một Form, cung cấp dữ liệu cho một Report.
<b>Data Table</b>	Bảng dữ liệu- nơi chứa dữ liệu vật lý của cơ sở dữ liệu
<b>Data type</b>	Kiểu dữ liệu. Dữ liệu trong cơ sở dữ liệu phải được định kiểu: kiểu số, kiểu chữ, kiểu lô gíc, ...- đó là các kiểu dữ liệu
<b>Database</b>	Cơ sở dữ liệu. Nơi chứa toàn bộ dữ liệu cho một mục đích quản lý nào đó, ở đó dữ liệu được thiết kế và lưu trữ theo các cấu trúc tối ưu.
<b>Datasheet</b>	Bảng dữ liệu- nơi dữ liệu hiển thị dưới dạng bảng bao gồm các cột và các dòng dữ liệu.
<b>Design View</b>	Môi trường thiết kế, có thể là thiết kế Form, thiết kế report, query,...
<b>Detete</b>	Chỉ hành động xóa các bản ghi ra khỏi cơ sở dữ liệu
<b>EOF</b>	End Of File – một dấu hiệu cho biết con trỏ bản ghi đã nằm ở vị trí cuối cùng của một recorset.

<b>Field</b>	Trường dữ liệu
<b>Field size</b>	Độ lớn của trường dữ liệu
<b>Form</b>	Đối tượng dùng thiết kế giao diện trong phát triển phần mềm. Kết quả của form khi sử dụng sẽ là các cửa sổ, hộp thoại- nơi mà người dùng có thể sử dụng để thao tác trên phần mềm.
<b>Form wizard</b>	Một trình điều khiển giúp đơn giản tạo một form nhập dữ liệu
<b>Formating</b>	Chỉ việc định khuôn dạng dữ liệu khi hiển thị
<b>Group By</b>	Một mệnh đề trong câu lệnh SQL dùng nhóm các bản ghi phục vụ các việc tổng hợp dữ liệu nào đó.
<b>Input mark</b>	Mặt nạ để nhập dữ liệu hay khuôn nhập dữ liệu
<b>Insert</b>	Chỉ hành động chèn thêm các bản ghi lên các bảng trong cơ sở dữ liệu
<b>Master</b>	Quan hệ cha. Thuật ngữ này thường đi với Child – quan hệ con
<b>Order By</b>	Một mệnh đề trong câu lệnh SQL dùng sắp xếp các bản ghi
<b>Query</b>	Một đối tượng dùng truy xuất, xử lý dữ liệu trên các hệ cơ sở dữ liệu
<b>Record</b>	Bản ghi
<b>Recordset</b>	Tập hợp các bản ghi
<b>Relationship</b>	Chỉ quan hệ giữa các bảng trong cơ sở dữ liệu
<b>Report wizard</b>	Một trình điều khiển giúp đơn giản tạo một report để hiển thị và in ấn dữ liệu
<b>Row</b>	Dòng trong một bảng dữ liệu
<b>Select</b>	Chỉ hành động chọn các bản ghi từ cơ sở dữ liệu
<b>SQL</b>	Structured Query Language – ngôn ngữ truy vấn dữ liệu có cấu trúc. Một ngôn ngữ xử lý dữ liệu khá phổ dụng trên các hệ cơ sở dữ liệu. Với Access, Query Design chính là nơi để thiết kế tạo ra các câu lệnh SQL thi hành các phép xử lý dữ liệu.

<b>Sub-form</b>	Form con- một form nằm trong một form khác. Khi đó form chứa sub-form gọi là form mẹ (form cha) thuật ngữ tiếng Anh là Master form.
<b>Table wizard</b>	Một trình điều khiển giúp đơn giản tạo một bảng dữ liệu
<b>Update</b>	Chỉ hành động cập nhật dữ liệu các bản ghi trong cơ sở dữ liệu
<b>Validation</b>	Một khái niệm chỉ sự đúng đắn của dữ liệu
<b>Where</b>	Một mệnh đề trong câu lệnh SQL để thiết lập điều kiện xử lý dữ liệu
<b>Wizard</b>	Là trình điều khiển giúp người lập trình đơn giản trong việc xây dựng ứng dụng. Có rất nhiều trình Wizard trong Acces như: Form wizard, report wizard, table wizard, database wizard, control wizard,...

# TÀI LIỆU THAM KHẢO

1. Microsoft Access 2000 Bible Quick Start; Cary N.Prague, Michael R. Iruin; Pulished by Hungry Minds, Inc, 2002.
2. Microsoft Access 2000 with VBA – Advanced; Al Napier, Phil Judd, H. Albert Napier, Philip J. Judd; Pulished by Kris Oxford, 2000.
3. Mastering Microsoft Office 2000 Professional Edition; Gini Courter and Annette Marquis; Pulished by Oxford Express, 1999.
4. Tài liệu từ Mạng Giáo dục Việt Nam  
<http://www.edu.net.vn/Default.aspx?tabindex=1&tabid=6&mid=18>
5. Thông tin sản phẩm và các tài liệu học tập liên quan tại Website của Công ty Microsoft <http://office.microsoft.com/en-ca/FX010857911033.aspx>.

# MỤC LỤC

<b>GIỚI THIỆU .....</b>	<b>1</b>
<b>BÀI MỞ ĐẦU .....</b>	<b>3</b>
1. Giới thiệu Access 2000 .....	4
2. Khởi động.....	5
3. Tạo mới tệp Access.....	6
4. Môi trường làm việc.....	8
5. Mở tệp đã tồn tại .....	8
6. Thoát khỏi Access.....	9
<b>CHƯƠNG 1: XÂY DỰNG CƠ SỞ DỮ LIỆU .....</b>	<b>10</b>
1. Các khái niệm về CSDL Access .....	11
1.1 CSDL Access .....	11
1.2 Bảng dữ liệu .....	12
1.3 Liên kết các bảng dữ liệu .....	15
2. Xây dựng cấu trúc bảng .....	17
3. Thiết lập quan hệ.....	23
4. Nhập dữ liệu.....	26
4.1 Cách nhập dữ liệu.....	26
4.2 Một số thao tác xử lý dữ liệu trên bảng.....	28
<i>a. Xoá bản ghi.....</i>	<i>28</i>
<i>b. Sắp xếp dữ liệu.....</i>	<i>29</i>
<i>c. Lọc dữ liệu.....</i>	<i>29</i>
5. Thuộc tính LOOKUP .....	30
6. Quy trình xây dựng CSDL Access .....	35
Bài tập .....	36
<b>CHƯƠNG 2: TRUY VẤN DỮ LIỆU .....</b>	<b>40</b>
1. SELECT queries .....	41
1.1 Cách tạo.....	41
1.2 Lọc dữ liệu .....	46
2. TOTAL queries .....	50
3. CROSSTAB queries .....	52
4. MAKE TABLE queries .....	55
5. DELETE queries .....	58
6. UPDATE queries .....	59
Bài tập .....	62

<b>CHƯƠNG 3: THIẾT KẾ GIAO DIỆN.....</b>	<b>66</b>
1. Khái niệm Forms.....	67
2. Sử dụng FORM WIZARD.....	67
3. Sử dụng FORM DESIGN VIEW.....	72
3.1 Thiết kế Form nhập dữ liệu đơn giản.....	72
3.2 Tinh chỉnh cấu trúc Form.....	76
4. Kỹ thuật Sub-form.....	81
Bài tập.....	93
<b>CHƯƠNG 4: THIẾT KẾ BÁO CÁO.....</b>	<b>98</b>
1. Các khái niệm về Report.....	99
1.1 Cấu trúc Report.....	99
1.2 Môi trường làm việc.....	100
2. Sử dụng Report wizard.....	101
3. Thiết kế report.....	107
4. Report chứa tham số.....	115
Bài tập.....	121
<b>CHƯƠNG 5: LẬP TRÌNH VBA CĂN BẢN.....</b>	<b>123</b>
1. Môi trường lập trình VBA.....	124
2. Các kiểu dữ liệu và khai báo.....	126
2.1 Các kiểu dữ liệu cơ bản.....	126
2.2 Biến và cách sử dụng biến.....	128
a. <i>Biến – khai báo biến đơn giản</i> .....	128
b. <i>Phạm vi biến</i> .....	130
2.3 Hằng và cách sử dụng hằng.....	131
a. <i>Khai báo hằng</i> .....	131
b. <i>Phạm vi hằng</i> .....	132
3. Các cấu trúc lệnh VBA.....	132
3.1 Cấu trúc IF... END IF.....	133
3.2 Cấu trúc SELECT CASE .. END SELECT.....	134
3.3 Cấu trúc FOR ... NEXT.....	137
3.4 Cấu trúc WHILE ... WEND.....	139
3.5 Lệnh DoCmd.....	140
4. Chương trình con.....	143
4.1 Chương trình con dạng hàm.....	144
4.2 Chương trình con dạng thủ tục.....	148
5. Kỹ thuật xử lý lỗi.....	150
5.1 Xử lý lỗi.....	150
5.2 Bẫy lỗi.....	154
6. Một số ví dụ.....	156

<b>CHƯƠNG 6: LẬP TRÌNH CƠ SỞ DỮ LIỆU.....</b>	<b>162</b>
1. Kỹ thuật DAO .....	163
1.1 Lớp đối tượng DAO .....	164
1.2 Đối tượng Database.....	166
1.3 Đối tượng RecordSet.....	167
1.4 Đối tượng QueryDef .....	172
1.5 Đối tượng TableDef .....	175
1.6 Đối tượng Relation.....	178
2. Bài toán đặt lọc dữ liệu .....	178
<b>CHƯƠNG 7: MENU &amp; TOOLBAR .....</b>	<b>183</b>
1. Tạo Menu .....	184
2. Gắn kết Menu, Toolbar .....	190
3. Tạo form chính.....	190
Bài tập .....	193
<b>THUẬT NGỮ TIN HỌC .....</b>	<b>202</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>204</b>