TRƯỜNG CAO ĐẲNG NGHỀ CÔNG NGHIỆP HÀ NỘI

Hoàng Anh Thơ (chủ biên)

GIÁO TRÌNH

LẬP TRÌNH QUẢN LÝ

(Lưu hành nội bộ)

Hà Nội năm 2013

LẬP TRÌNH QUẢN LÝ

MỤC LỤC

Bài mở đầu: GIỚI THIỆU CHUNG

Thời gian: 1 giờ

1. Giới thiệu Access

Microsoft Access là hệ quản trị cơ sở dữ liệu trên môi trường Windows, trong đó có sẵn các công cụ hữu hiệu và tiện lợi để tự động thiết lập chương trình cho hầu hết các bài toán thường gặp trong quản lý, thống kê, kế toán. Với Access, người dùng không phải viết từng câu lệnh cụ thể như trong Pascal, C hay Foxpro mà chỉ cần tổ chức dữ liệu và thiết kế các yêu cầu, công việc cần giải quyết.

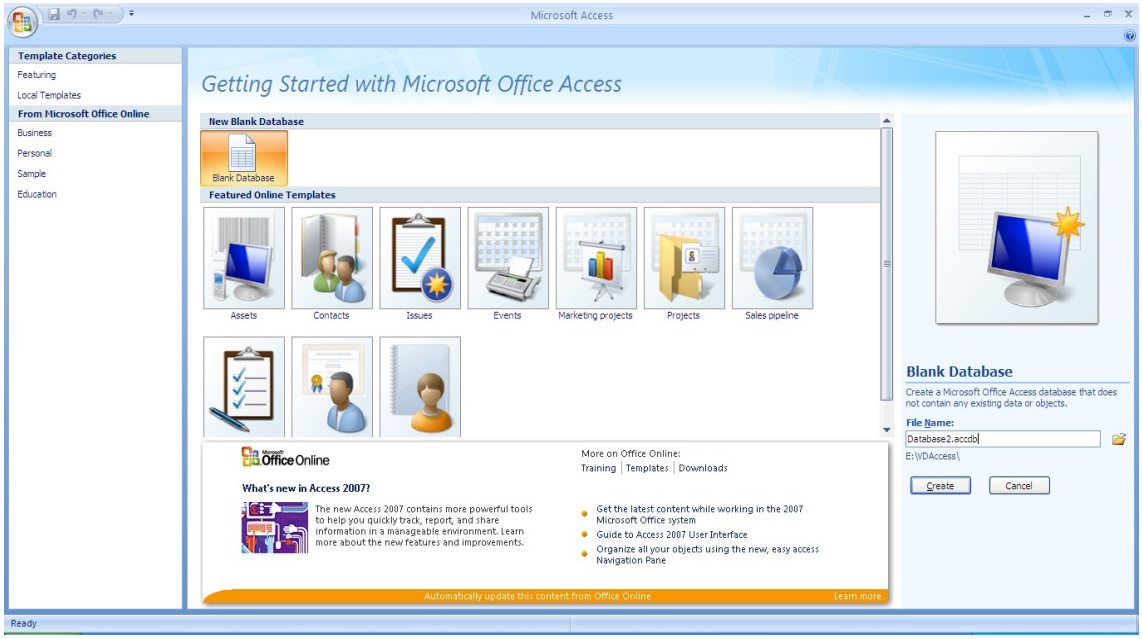
Hiện nay các phiên bản Access thường sử dụng là: Access 97 trong bộ Microsoft Office 97, Access 2000 trong bộ Microsoft Office 2000, Access 2003 trong bộ Microsoft Office 2003, Access 2007 trong bộ Microsoft Office 2007, Access 2010 trong bộ Microsoft Office 2010. Trong đề cương này sẽ hướng dẫn sử dụng Access 2007.

Sáu đối tượng công cụ mà Access cung cấp là: Bảng (Table), Truy vấn (Query), mẫu biểu (Form), báo biểu (Report), Macro và Module.

2. Khởi động

Chương trình Access được xây dựng và thực hiện trong môi trường Access. Thực hiện khởi động chương trình Access 2007 như sau:

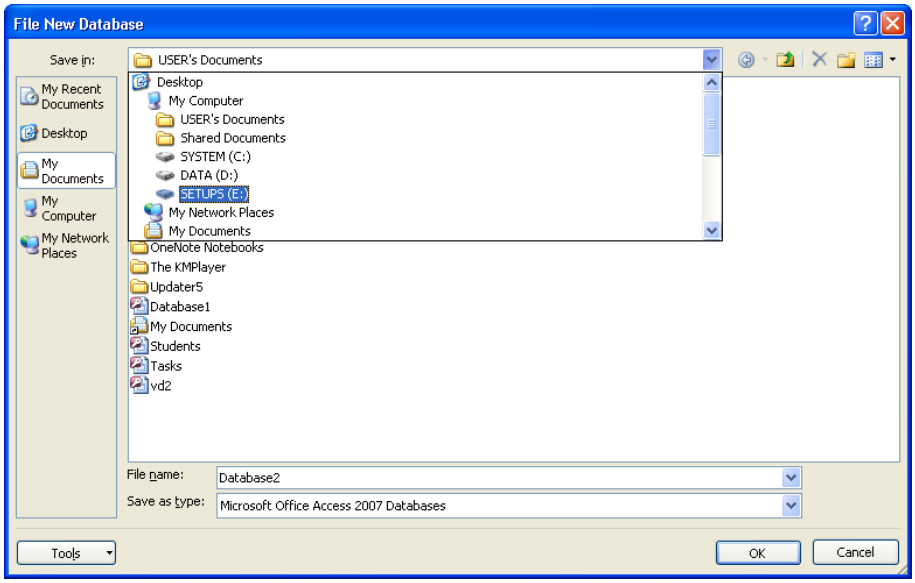
Click chọn **Start → Programs** (hoặc **All Programs**)→ **Microsoft Office → Microsoft Office Access 2007 →** xuất hiện cửa sổ Microsoft Access



3. Tạo mới tệp Access

Sau khi khởi động xong chương trình Access, trong cửa sổ Micosoft Access thực hiện tạo mới tệp Access theo trình tự sau:

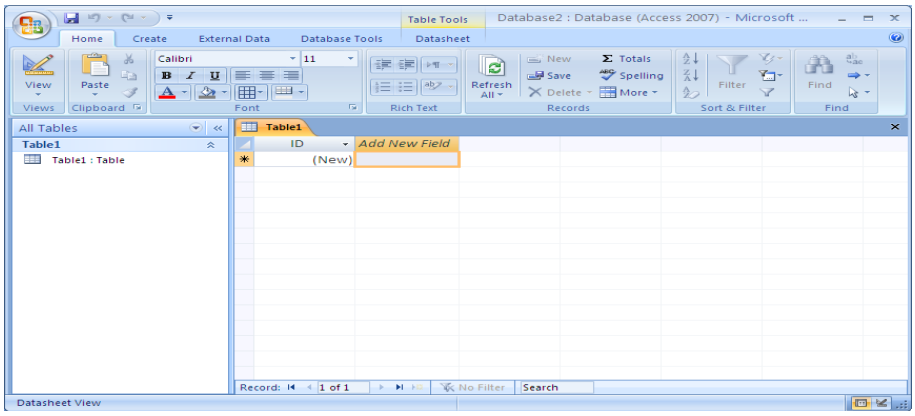
1. Click chọn **Blank Database** →Click chọn vào biểu trượng Folder →xuất hiện hộp thoại → Chọn thư mục lưu chương trình tại hộp **Save in** →Đặt tên tệp chương trình trong hộp **File Name** → **OK**



2. Click chọn **Create** →xuất hiện cửa sổ Database. Đây chính là cửa sổ quan trọng của Access để thực hiện các thao tác với cơ sở dữ liệu.

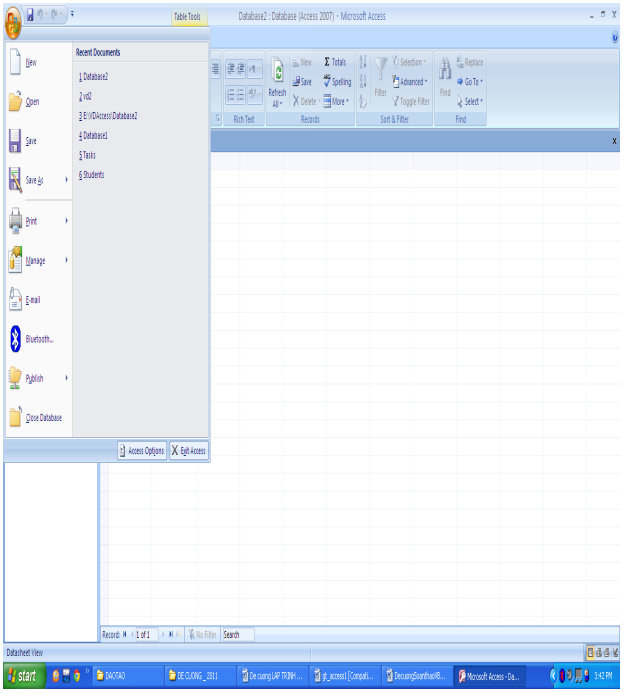
** Nếu đang ở trong cửa sổ Database muốn tạo tệp Access mới thì Click chọn vào nút **Office Button** → **New** → xuất hiện cửa sổ Microsoft Access → thực hiện hai bước tạo tệp mới như trên.*

4. Môi trường làm việc



Môi trường làm việc chính của Access là trên cửa sổ Database gồm các thành phần sau:

- ***Thanh tiêu đề (Title bar)***: nằm ở trên cùng chứa tên cơ sở dữ liệu (Database), tên chương trình (Microsoft Access) và tên đối tượng đang làm việc (Table, Query, Form, Report…).
- ***Thanh thực đơn (Menu bar)***: chứa tập hợp các lệnh làm việc của Microsoft Access. Khi click chuột vào tên nhóm (Home, Create, Database Tools …) thì một danh sách các lệnh trong nhóm sẽ hiện ra ngay tại khung hiển thị các tuỳ chọn của Menu để chọn một lệnh cần thực hiện.
- ***Khung chứa các đối tượng (Navigation Pane)***: ở phía bên trái màn hình, chứa tập các đối tượng đã được tạo gồm: Table, Query, Form, Report …
- ***Cửa sổ của đối tượng***: khi chọn làm việc với đối tượng nào thì sẽ xuất hiện cửa sổ thiết kế và hiển thị kết quả đối tượng đó.
- ***Office Button***: là nút lệnh nằm phía trên bên trái màn hình, chứa các lệnh cơ bản của chương trình như New, Open, Save, Print …

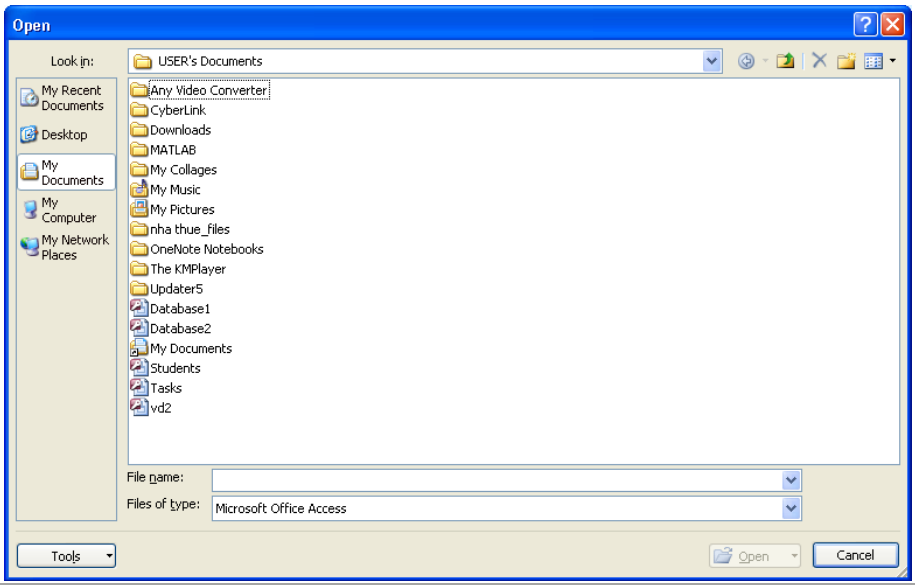


5. Mở tệp đã tồn tại

Sau khi khởi động xong chương trình Access, trong cửa sổ Micosoft Access có khung cửa sổ bên phải là **Open Recent Database** chứa tên các tệp cơ sở dữ liệu đã được mở gần đây nhất. Nếu muốn mở tệp nào thì click vào tên tệp đó.

Nếu muốn mở tệp không có trong danh sách trên thì thực hiện click vào lệnh **More** →xuất hiện cửa sổ Open → chọn thư mục chứa tệp trong hộp **Look in** → chọn tên tệp cần mở → **Open** (hoặc click đúp vào tên tệp cần mở)

Nếu đang ở trong cửa sổ Database mà muốn mở tệp khác thì thực hiện click vào nút lệnh **Office Button** → **Open** → xuất hiện cửa sổ Open → chọn thư mục chứa tệp và tên tập cần mở.



6. Thoát khỏi Access

**** Đóng cơ sở dữ liệu đang làm việc***:

Click vào nút lệnh **Office Button → Close Database**

**** Thoát khỏi môi trường Access***:

Click vào nút lệnh **Office Button → Exit Access**

Hoặc: Click vào biểu tượng ☒ (**Close**) ở góc trên bên phải màn hình

Bài 1: XÂY DỰNG CƠ SỞ DỮ LIỆU

Thời gian: 13 giờ

1. Các khái niệm về CSDL Access

1.1. CSDL Access

Chương trình Access gọi là một Database (CSDL). Trong các ngôn ngữ truyền thống như C, Pascal, Foxpro, một hệ Chương trình gồm các tệp Chương trình và các tệp dữ liệu được tổ chức một cách riêng biệt. Nhưng trong Access toàn bộ Chương trình và dữ liệu được chứa trong một tệp duy nhất (*.**mdb** hoặc *.**accdb**). Thuật ngữ hệ Chương trình hay CSDL được hiểu là tổ hợp bao gồm cả Chương trình và dữ liệu. Để ngắn gọn có thể gọi Chương trình thay cho thuật ngữ hệ Chương trình.

Như vậy khi nói đến Chương trình hay hệ chương trình hay CSDL thì cùng có nghĩa đó là một hệ phần mềm gồm cả Chương trình và dữ liệu do Access tạo ra.

1.2. Bảng dữ liệu

Bảng là nơi chứa dữ liệu của một đối tượng nào đó. Bảng có cấu trúc dòng và cột. Các cột mô tả một đặc điểm của dữ liệu gọi là trường, các dòng thể hiện đầy đủ thông tin của một dữ liệu gọi là bản ghi.

Một cơ sở dữ liệu (CSDL) thường gồm nhiều bảng. Một bảng gồm nhiều trường có các kiểu khác nhau như: Text, Number, Date/Time…

Các bảng trong một CSDL thường có quan hệ với nhau.

1.3. Liên kết các bảng dữ liệu

Liên kết các bảng dữ liệu nhằm mục đích thao tác được với dữ liệu từ nhiều bảng cùng một lúc mà vẫn đảm bảo toàn vẹn dữ liệu trong các phép thêm, xoá, sửa các bản ghi.

Liên kết các bảng dữ liệu chính là thiết lập quan hệ giữa các bảng được thực hiện dựa trên các trường quan hệ có cùng tên và cùng kiểu.

Có hai kiểu quan hệ phổ biến là quan hệ 1-1 và quan hệ 1-nhiều.

- ***Quan hệ 1-1***: là quan hệ giữa các bảng mà mỗi bản ghi trong bảng này sẽ chỉ quan hệ với một và chỉ một bản ghi trong bảng kia.

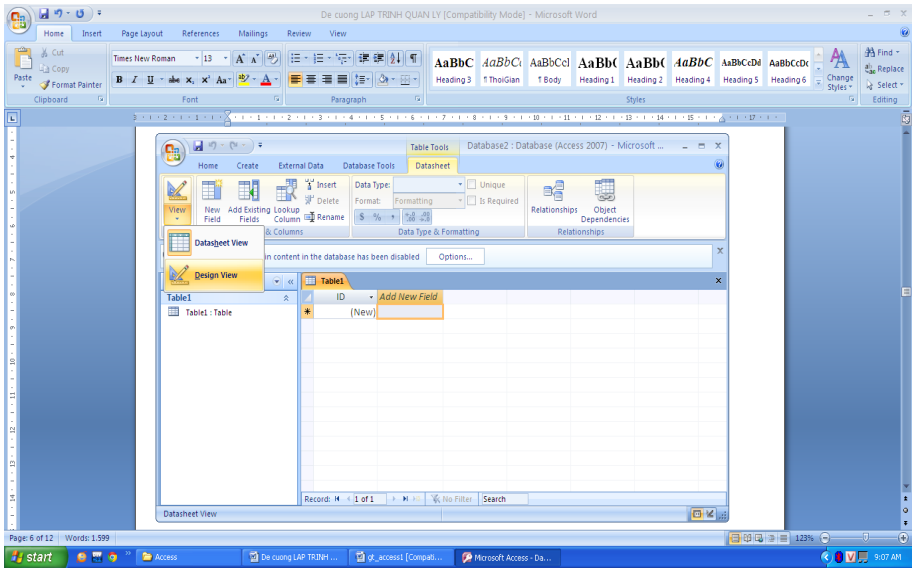
- ***Quan hệ 1-nhiều***: là quan hệ giữa các bảng mà mỗi bản ghi trong bảng 1 có quan hệ với nhiều bản ghi trong bảng nhiều, ngược lại mỗi bản ghi trong bảng nhiều chỉ được phép quan hệ với một và chỉ một bản ghi trong bảng 1.

2. Xây dựng cấu trúc bảng

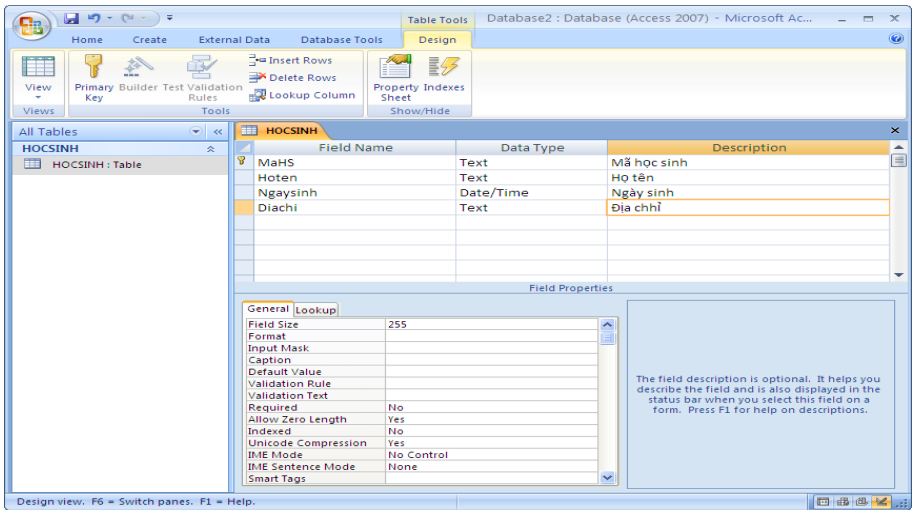
2.1. Tạo bảng

2.1.1. Các thao tác thực hiện tạo bảng

Trong cửa sổ Database, trên thanh thực đơn chọn lệnh **Creat → Table**



Trong mục **View** → chọn **Design View** → xuất hiện hộp thoại **Save As** → đặt tên bảng → xuất hiện cửa sổ thiết kế cấu trúc bảng



- ***Field Name***: đặt tên trường (bắt đầu bằng chữ cái, không quá 64 kí tự, không nên có dấu cách và các ký tự đặc biệt)

- ***Data Type***: chọn kiểu dữ liệu
- ***Description***: mô tả trường nếu muốn (không bắt buộc). Mục đích để giải thích cho rõ hơn một trường nào đó. Văn bản mô tả sẽ được hiển thị khi nhập số liệu cho các trường.
- ***Field Properties***: thuộc tính của trường. Tương ứng với một trường ở khung bên trên sẽ có các thuộc tính của trường đó ở khung bên dưới. Mỗi thuộc tính nằm trên mộl dòng. Giá trị của mỗi thuộc tính lúc đầu hoặc chưa dùng (bỏ trống) hoặc giá trị mặc định. Khi đặt giá trị của thuộc tính có thể gõ trực tiếp từ bàn phím (như thuộc tính FieldSize của trường Text) hoặc có thể chọn từ một danh sách của Combo Box (như thuộc tính FieldSize của trường Number).

Sau khi hoàn chỉnh thiết kế, phải thực hiện lưu cấu trúc bảng như sau:

Click chọn nút lệnh **Office Button → Save** (hoặc **Ctrl +S**)

2.1.2. Các kiểu dữ liệu

Access gồm các kiểu dữ liệu sau:

Tên kiểu	***Ý nghĩa***	***Độ lớn***
· Text	Ký tự	255 Byte
· Memo	Ký tự văn bản	64000 Byte
· Number	Số nguyên, thực	1 , 2, 4 hoặc 8 Byte
· Date/time	Ngày tháng/giờ	8 Byte
· Currency	Tiền tệ	8 Byte
· AutoNumber	Kiểu số	8 Byte
· Yes/No	Kiểu logic (Yes/No)	1 Bit
· OLE OObject	Đối tượng (ảnh)	1 Giga Byte
· Hyperlink	Ký tự hoặc kết hợp ký tự và số	
· Lookup Wizard	Cho phép chọn giá trị từ bảng khác	

2.1.3. Các thuộc tính của trường

Mỗi trường có một kiểu dữ liệu khác nhau, tương ứng với mỗi kiểu dữ liệu lại có các thuộc tính của trường đó khác nhau. Nhưng phổ biến là các thuộc tính sau:

Tên thuộc tính	*Ý nghĩa*
· Field Size	Độ dài của trường Text, hoặc kiểu của trường
· Format	Dạng hiển thị dữ liệu kiểu ngày và số.
· DecimalPlaces	Số chữ số thập phân trong kiểu number và currency.
· InputMask	Mặt nạ nhập, quy định khuôn dạng nhập liệu
· Caption	Đặt nhãn cho trường. Nhãn sẽ được hiển thị khi nhập liệu thay vì tên trường (nhãn mặc định).
· Default Value	Xác định giá trị mặc định của trường.
· Validation Rule	Quy tắc dữ liệu hợp lệ. Dữ liệu phải thoả mãn quy tắc này mới được nhập.
· Required	Không chấp nhận giá trị rỗng. Cần phải nhập một dữ liệu cho trường.
· AllowZeroLength	Chấp nhận chuỗi rỗng trong trường Text, Memo.
· Indexed	Tạo chỉ mục để tăng tốc độ tìm kiếm trên trường này

2.2. Đặt khóa chính

2.2.1. Ý nghĩa

Khoá chính là một hoặc nhiều trường xác định duy nhất một bản ghi.

Access tự động tạo chỉ mục (Index) trên khoá nhằm tăng tốc độ truy vấn và các thao tác khác. Khi hiển thị dữ liệu (dạng bảng hay mẫu biểu), các bản ghi sẽ được trình bầy theo thứ tự khoá chính. Khi nhập dữ liệu, Access kiểm tra sự trùng nhau trên khoá chính.

Access dùng khoá chính để tạo sự liên kết giữa các bảng. Khi thiết kế cấu trúc bảng thường có trường ID là trường mặc định làm khóa chính.

2.2.2. Thực hiện đặt khóa chính

Trong cửa sổ thiết kế bảng, chọn trường làm khóa chính → trên thanh công cụ Design, bấm chọn lệnh ***Primary Key*** (trường làm khóa chính sẽ có biểu tượng chìa khóa bên cạnh tên trường)

Nếu muốn thay đổi trường làm khóa chính thì thực hiện chọn lại trường mới → bấm chọn lệnh ***Primary Key***

Vì khóa chính là không bắt buộc nên nếu muốn xóa khóa chính thì thực hiện chọn lại trường khóa chính muốn xóa → bấm chọn lệnh ***Primary Key*** (biểu tượng chìa khóa mất đi nghĩa là đã xóa được khóa chính)

2.3.Thay đổi cấu trúc bảng

Trong cửa sổ Database, click đúp vào tên bảng cần thay đổi cấu trúc →trong mục **View** → chọn **Design View** → xuất hiện cửa sổ thiết kế cấu trúc bảng → thực hiện các thao tác thay đổi cấu trúc bảng:

- ***Chèn thêm trường mới***: chọn vị trí cần chèn → chọn lệnh **Insert Rows** trên thanh công cụ Design → nhập tên trường mới vào vị trí vừa chèn.
- ***Xóa trường***: chọn dòng chứa trường cần xóa → chọn lệnh **Delete Rows** trên thanh công cụ Design
- ***Thay đổi vị trí của trường***: chọn trường cần thay đổi vị trí, bấm giữ chuột di chyển trường đó lên vị trí mới.
- ***Thay đổi trường***: chọn trường cần thay đổi, sửa tên trường tại cột Field Name, thay đổi kiểu trường tại cột Data Type, thay đổi thuộc tính trường tại khung Field Properties bên dưới.

Sau khi thay đổi xong thiết kế, phải thực hiện lưu cấu trúc bảng mới bằng cách: Click chọn nút lệnh **Office Button → Save** (hoặc **Ctrl +S**)

3. Thiết lập quan hệ

3.1. Ý nghĩa

Access dùng quan hệ để đảm bảo những ràng buộc toàn vẹn giữa các bảng liên quan trong các phép thêm, sửa xoá bản ghi. Trong một CSDL muốn thiết lập được quan hệ thì phải có ít nhất hai bảng (hoặc truy vấn) có các trường liên quan đến nhau.

Nguyên tắc đặt quan hệ là chỉ định một hoặc một nhóm trường chứa cùng giá trị trong các bản ghi có liên quan. Thường đăt quan hệ giữa khoá chính của

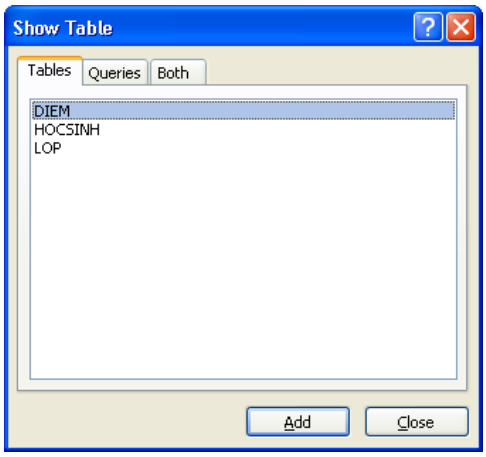
một bảng với trường nào đó của bảng khác (bảng này gọi là bảng quan hệ), các trường này thường cùng tên, cùng kiểu.

Trong một CSDL thường thiết lập hai kiểu quan hệ là 1-1 và 1-nhiều (1-n).

- ***Quan hệ 1-1***: Các trường sử dụng để tạo quan hệ trong hai bảng đều là khoá chính.
- ***Quan hệ 1-n***: trường liên kết dùng trong bảng chính phải là khoá chính, còn trường trong bảng quan hệ không phải là khoá chính của bảng đó (khoá ngoại).

3.2. Thực hiện thiết lập quan hệ

Trong cửa sổ Database, trên thanh thực đơn chọn lệnh **Database Tools** → **RelationShips** → xuất hiện hộp thoại Show Table

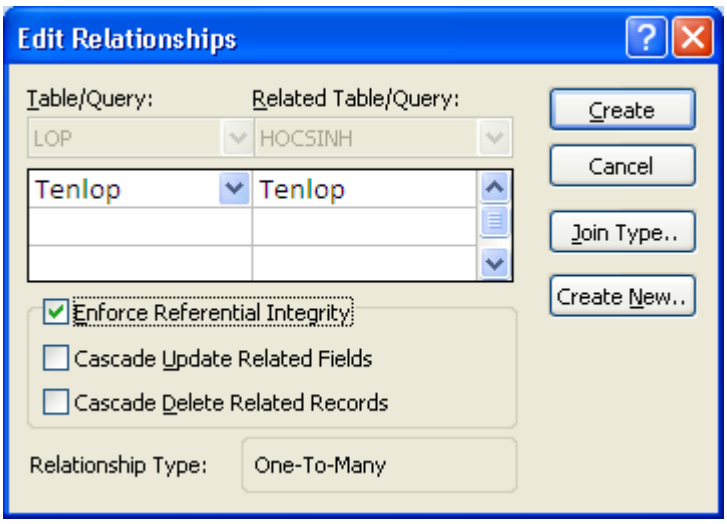


- ***Tables***: hiển thị các bảng
- ***Queries***: hiển thị các truy vấn
- ***Both***: hiển thị cả bảng và truy vấn

-

-

- Chọn các bảng và truy vấn để đưa vào quan hệ (sử dụng các phím Ctrl hoặc Shift để chọn nhiều bảng hoặc truy vấn) → **Add** → **Close** đóng hộp thoại Show Table → xuất hiện cửa sổ RelationShips
- Chọn một trường từ bảng chính (Primary table) và kéo sang trường tương ứng của bảng quan hệ → xuất hiện hộp thoại Edit RelationShips



- Click chọn ***Enforce Referential Integnty***: Đảm bào việc nhập đúng (chỉ nhập được các bản ghi trên bảng quan hệ khi giá trị trường dùng làm khoá liên kết đã có trên bảng chính)

  - ***Cascade Update Related fields***: Khi sửa giá trị trường khoá một bản ghi trong bảng chính, giá trị các bản ghi tương ứng trong bảng quan hệ sẽ bị sửa theo
  - ***Cascade Delete Related fields***: Khi xoá một bản ghi trong bảng chính, các bản ghi tương ứng trong bảng quan hệ sẽ bị xoá.

→ **Create**: thực hiện tạo quan hệ. Khi đó sẽ có đường thẳng nối giữa hai trường biểu diễn quan hệ vừa tạo, lúc này kiểu quan hệ mặc định theo trường chọn để tạo quan hệ.

- Sau khi thiết lập xong quan hệ phải thực hiện lưu lại như sau:

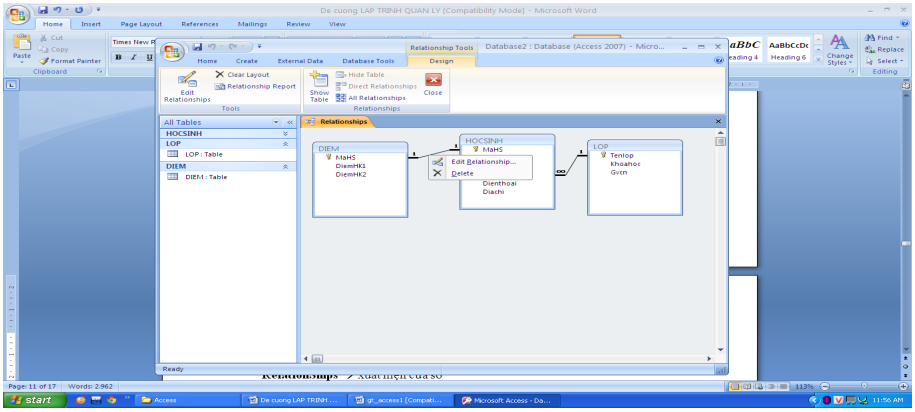
Click chọn nút lệnh **Office Button → Save** (hoặc **Ctrl +S**)

** **Chú ý**: Chỉ thực hiện thiết lập được quan hệ cho các bảng chưa có dữ liệu. Bảng đã nhập dữ liệu rồi thì không thực hiện thiết lập quan hệ được.*

3.3. Chỉnh sửa quan hệ

Trong cửa sổ Database, trên thanh thực đơn chọn lệnh **Database Tools → RelationShips** → xuất hiện cửa sổ RelationShips →Click phải chuột vào đường quan hệ cần chỉnh sửa → xuất hiện menu

- ***Edit Relationship***… : thực hiện chỉnh sửa quan hệ
- ***Delete***: xóa quan hệ



4. Nhập dữ liệu

4.1. Cách nhập dữ liệu

Trong cửa sổ Database, click đúp vào tên bảng cần nhập dữ liệu ở khung chứa các đối tượng (Navigation Pane) phía bên trái màn hình → xuất hiện cửa sổ nhập liệu của bảng (Datasheet)

Nhập dữ liệu hoặc sửa dữ liệu đã nhập trên các dòng. Kiểu dữ liệu nhập vào phụ thuộc vào kiểu trường tương ứng đã đặt.

Định dạng dữ liệu (kiểu chữ, cỡ chữ, màu chữ,…) và Datasheet bằng cách sử dụng các công cụ thuộc thực đơn **Home.**

* ***Chú ý***:

- Đối với các bảng có quan hệ phải nhập dữ liệu cho bảng chính trước rồi nhập được dữ liệu cho bảng quan hệ. Dữ liệu của trường tham gia quan hệ ở bảng chính phải tồn tại trước thì mới nhập được dữ liệu đó ở bảng quan hệ. Không cho phép nhập dữ liệu vào trường khóa của bảng quan hệ có giá trị khóa mới, không có ở bảng chính.

- Từ bảng chính có thể thực hiện nhập dữ liệu cho các bảng quan hệ. Không thực hiện ngược lại được.

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

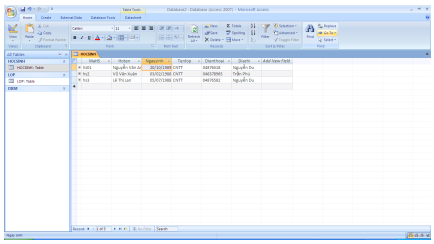
4.2.1. Xoá bản ghi

Click phải chuột vào đầu dòng bản ghi cần xóa → xuất hiện menu → chọn **Delete Record**

4.2.2. Sắp xếp dữ liệu

- Chọn trường khóa để sắp xếp dữ liệu

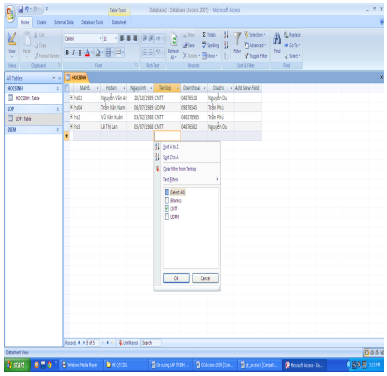
- Vào thực đơn **Home** → chọn nhóm công cụ **Sort & Filter**



- A→Z: Ascending: sắp xếp tăng dần
- Z→A: Descending: sắp xếp giảm dần
- Clear All Sort: hủy bỏ lệnh sắp xếp

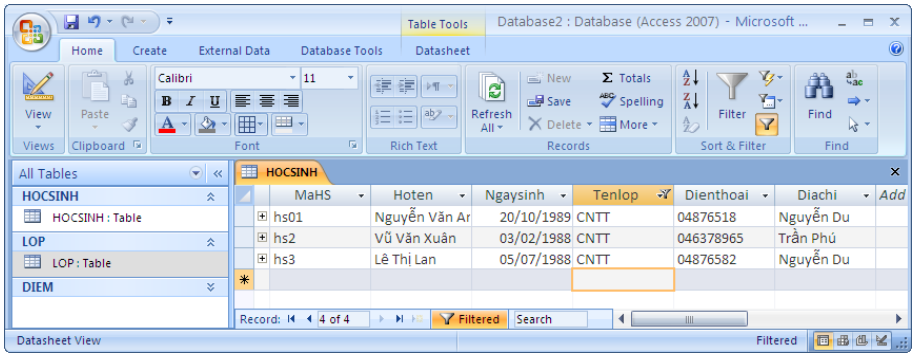
4.2.3. Lọc dữ liệu

- Chọn trường cần lọc dữ liệu



- Vào thực đơn **Home** → chọn nhóm công cụ **Sort & Filter** → click chọn công cụ **Filter** → xuất hiện hộp thoại:

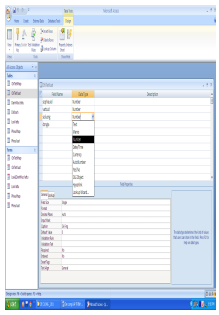
Trong mục **Text Filters**, chọn dữ liệu cần lọc, bỏ chọn các dữ liệu không cần → OK → Kết quả hiện thị cac bản ghi thỏa mãn điều kiện lọc.



- Muốn hủy bỏ lọc dữ liệu click lại vào công cụ **Filter**→ chọn **Clear Filter**

5. Thuộc tính LOOKUP

5.1. Khái niệm về Combo Box



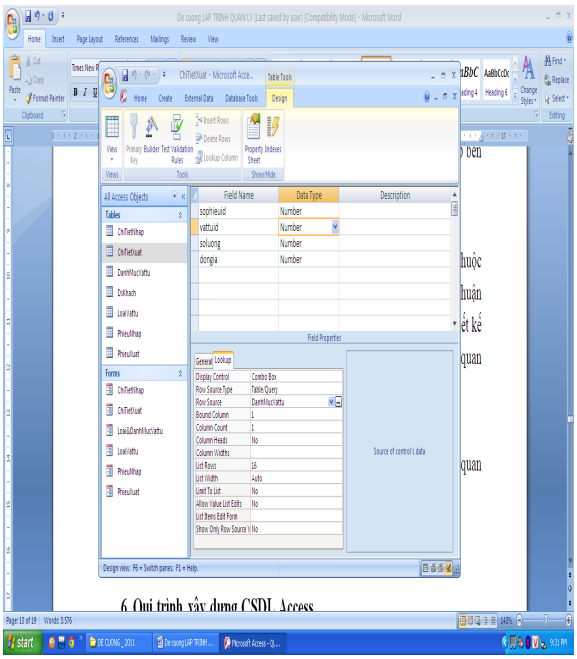
Combo Box là một công cụ tiện lợi dùng để chọn một giá trị trong danh sách cho trước. Combo Box rất thường được sử dụng trong Access để tổ chức nhập liệu và chọn giá trị cho các thuộc tính.

Mỗi Combo Box chứa một danh sách giá trị. Muốn hiển thị danh sách này phải click chuột tại vị trí có biểu tượng hình mũi tên bên phải.

Khi chọn một giá trị trong danh sách thì giá trị này sẽ xuất hiện tại hộp bên trên của Combo Box.

5.2. Tạo Combo Box nhập dữ liệu bằng thuộc tính Lookup

Tại bảng quan hệ (bảng phụ), dữ liệu của trường tham gia quan hệ phụ thuộc vào trường tham gia quan hệ ở bảng chính. Để đảm bảo tính toàn vẹn này và thuận tiện trong việc nhập dữ liệu thì tại trường tham gia quan hệ của bảng phụ thiết kế dạng Combo Box. Dữ liệu trong Combo Box này chính là dữ liệu của trường quan hệ tại bảng chính. Sử dụng thuộc tính Loopup của trường để thiết kế.



Thực hiện thiết kế như sau:

- Trong cửa sổ thiết kế cấu trúc bảng, đặt con trỏ tại trường tham gia quan hệ, chọn kiểu dữ liệu phù hợp
- Tại khung **Field Properties**, thiết lập các thuộc tính sau:
  - ***Display Control***: chọn Combo Box
  - ***Row Source Type***: Table/ Query
  - ***Row Source***: chọn bảng chính hoặc truy vấn chứa dữ liệu nhập

Bài 2: TRUY VẤN DỮ LIỆU

Thời gian : 13 giờ

1. SELECT queries

Select Query là truy vấn chọn, loại thông dụng nhất có các khả năng như:

- Chọn bảng, query khác làm nguồn dữ liệu

- Chọn các trường hiển thị

- Thêm các trường mới là kết quả thực hiện các phép tính trên các trường của bảng nguồn

- Đưa các điều kiện tìm kiếm, lựa chọn

- Đưa các trường dùng để sắp xếp

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

1.1. Cách thực hiện

1.1.1. Các bước thao tác

* Trên menu Create chọn biểu tượng Query Design

* Chọn các bảng/ truy vấn nguồn sau đó nhấn Add/Close (hoặc kích đúp vào các bảng/truy vấn).

* Chọn các trường từ các bảng /truy vấn nguồn để đưa vào truy vấn mới bằng cách kích đúp chuột vào từng trường hoặc kéo các trường xuống đặt vào dòng Field.

* Chọn kiểu sắp xếp dữ liệu cho các trường tại dòng Sort

* Bỏ dấu tích tại dòng Show dưới các trường không muốn hiển thị dữ liệu.

* Đưa điều kiện vào dòng Criteria để trích các bản ghi theo yêu cầu của bài toán. Nếu không đưa điều kiện để chọn lọc thì kết quả của truy vấn bao gồm tất cả các bản ghi từ các bảng/truy vấn nguồn.

* Đóng và lưu truy vấn.

Chú ý

- Có thể thêm các trường mới vào truy vấn theo công thức như sau:

[Tên trường mới]:[Biểu thức tính toán]

- Đổi tên trường khi hiển thị dữ liệu theo công thức như sau:

[Tên mới]: [tên trường cũ]

1.1.2. Chỉnh sửa và thực thi truy vấn

a. Sửa truy vấn

Phải chuột vào truy vấn → chọn Design view → Sử dụng các công cụ trong menu Design để thực hiện chỉnh sửa:

- Thêm, bớt các bảng, truy vấn nguồn
- Thêm, bớt các trường đưa vào truy vấn
- Đổi thứ tự các trường
- Sửa, thêm, xóa các điều kiện lọc
- Sửa, thêm, xóa các trường dùng để sắp xếp

b. Thực thi truy vấn

- Cách 1: Mở truy vấn ở dạng Design, chọn biểu tượng Run trên thanh công cụ
- Cách 2: Kích đúp chuột vào tên truy vấn
- Cách 3: Chọn biểu tượng DataSheet View trên thanh công cụ

1.2. Lọc dữ liệu

1.2.1. Mục đích

Lọc dữ liệu là lập tiêu chuẩn lựa chọn (điều kiện tìm kiếm) để chỉ hiện những thông tin cần quan tâm theo đúng yêu cầu trong bảng kết quả.

1.2.2. Cách thực hiện

Gõ trực tiếp một biểu thức điều kiện vào ô Criteria của các trường cần đặt điều kiện. Các điều kiện đồng thời thỏa mãn thì đặt cùng dòng Criteria, các điều kiện có phép hoặc thì đặt trên các dòng OR.

1.2.3. Các điều kiện trong truy vấn

a. Các phép toán số học

- Các phép tính toán: +, -, *, /
- So sánh: = , <>, >, >=, <, <=

b. Các hàm Logic

- Hàm AND (và): Biểu thức1 AND biểu thức 2

- Hàm OR (hoặc): Biểu thức1 OR biểu thức 2

- Hàm NOT (phủ định): NOT Biểu thức

- Hàm IIF(Điều kiện, Giá trị 1, Giá trị 2)

Trong đó: Điều kiện là biểu thức Logic

Giá trị 1, Giá trị 2: Là biểu thức số hoặc chuỗi

Nếu điều kiện đúng hàm nhận giá trị 1 ngược lại nhận giá trị 2. Có thể kết hợp nhiều hàm IIF lồng nhau.

c. Phép toán, hàm với chuỗi ký tự

- Phép LIKE kết hợp với các ký tự thay thế '?' (thay thế một ký tự) hoặc '*' (thay thế một nhóm ký tự)

*VD: Lọc danh sách những người có họ "Nguyễn" → Like "Nguyễn *"*

- Phép & dùng để nối 2 chuỗi

- Left(String,n) trích ra n ký tự bên trái của chuỗi String

- Right(String,n) trích ra n ký tự bên phải của chuỗi String.

- Mid(String,i,n) trích ra n ký tự bắt đầu từ ký tự thứ i của chuỗi String.

- Str(number) đổi số sang chuỗi.

d. Hàm ngày tháng

- Date() Cho ngày hiện tại của hệ thống

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

- Day([trường ngày tháng]): Cho kết quả là ngày

- Month([trường ngày tháng]): Cho kết quả là tháng

- Year([trường ngày tháng]): Cho kết quả là năm

- Weekday([trường ngày tháng]): Cho kết quả là tuần

VD: tìm mặt hàng nhập từ ngày 22/11/2004 đến ngày 22/12/2004 viết như sau: Between #22/11/2005# and #22/112006#

- DateAdd([kí tự],[số],[trường ngày tháng]): Phép cộng ngày tháng, cho kết quả trả về kiểu DateTime

+ Kí tự = "d": cộng thêm ngày

+ Kí tự = "m": cộng thêm tháng

+ Kí tự = "yyyy": cộng thêm năm

VD:

DateAdd("d",3,date()): cộng thêm 3 ngày vào ngày hiện tại

DateAdd("m",2,Date()):cộng thêm 2 tháng so với tháng hiện tại

DateAdd("yyyy",1,Date()): cộng thêm 1 năm so với năm hiện tại

e. Các phép toán khác

- BETWEEN ... AND ...: Dùng để biểu thị một khoảng giá trị

  VD: tìm mặt hàng nhập từ ngày 22/11/2004 đến ngày 22/12/2004 viết như sau: Between #22/11/2005# and #22/112006#

- IN (Giá trị 1, giá trị 2,...): Đưa ra các bản ghi có giá trị trùng với giá trị trong hàm IN. Tương tự phép OR.
- Round([giá trị],n): hàm làm tròn giá trị đến n số trước và sau dấu phẩy. *(n>0 làm tròn sau dấu phẩy, n<0: làm tròn trước dấu phẩy)*
- Int([giá trị]): hàm lấy phần nguyên của X.
- Is null: hàm điều kiện để đưa ra những bản ghi có giá trị của trường là rỗng.

2. TOTAL queries

2.1. Công dụng

Total queries là truy vấn tính toán trên nhóm. Đầu tiên là thực hiện phân nhóm các bản ghi theo điều kiện, sau đó thực hiện các phép tính trên từng nhóm.

2.2. Cách thực hiện

- Tạo truy vấn Select query

- Chọn Menu Design -> chọn biểu tượng Total.

- Chọn trường phân nhóm (Group by trên dòng Total)

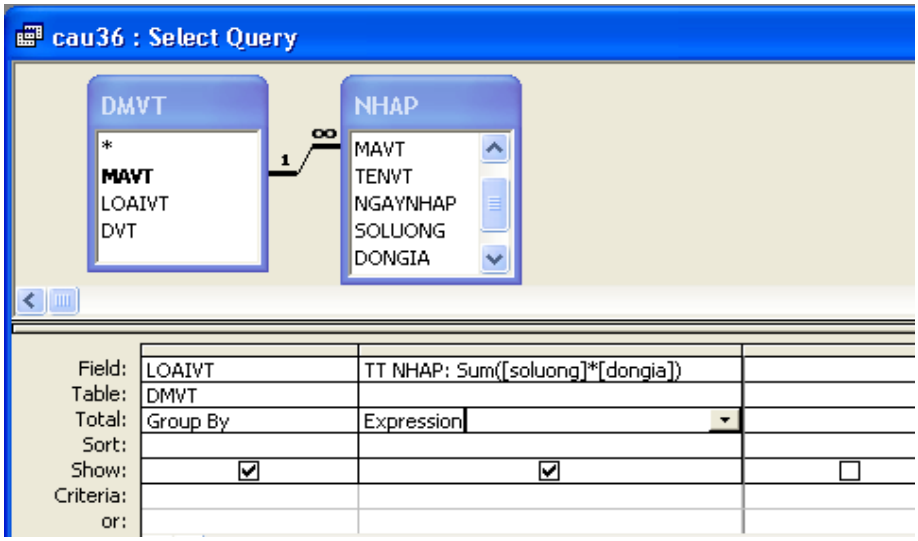
- Chọn trường điều kiện, tiêu chuẩn tham gia phân nhóm và tính tổng (Where trên dòng Total, đặt điều kiện trên dòng Criteria)

- Chọn hàm tính toán. Trong đó có các hàm sau:

Tên Hàm	***ý nghĩa***
Sum	Tính tổng các giá trị trên trường kiểu Number
Avg	Tính giá trị trung bình của trường kiểu

	Number
Min	Tính giá trị nhỏ nhất của trường kiểu Number
Max	Tính giá trị lớn nhất của trường kiểu Number
Count	Đếm giá trị khác rỗng của trường
First	Cho giá trị của bản ghi đầu tiên trong nhóm
Last	Cho giá trị của bản ghi cuối cùng trong nhóm
Group	Dùng để nhóm các bản ghi
Where	Dùng cho các trường có điều kiện

* ***Ví dụ***: Đưa ra tổng tiền nhập của từng loại vật tư.



3. CROSSTAB queries

3.1. Công dụng

Truy vấn Crosstab được dùng để tóm lược dữ liệu và trình bày kết quả theo dạng cô đọng như một bảng tính toán.

3.2. Cách thực hiện

- Tạo truy vấn Select query

- Chọn biểu tượng CrossTab query trên thanh công cụ của Menu Design.

- Đưa các trường cần thể hiện trong bảng vào dòng Field

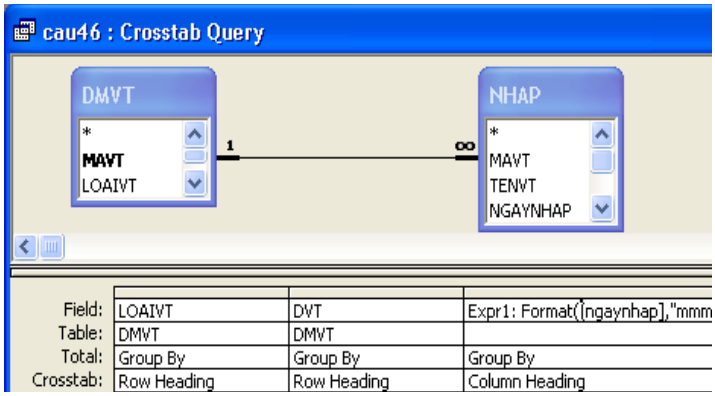
- Chọn hàm cần tính cho trường vào dòng Total

- Chọn kiểu thể hiện dữ liệu cho trường vào dòng CrossTab.

(***Chú ý**: có thể chọn nhiều trường thể hiện trong Row Heading nhưng chỉ được chọn 1 trường Column Heading và 1 trường Value. Nếu không muốn hiện dữ liệu của trường thì chọn Not Show)*

- Chọn kiểu sắp xếp tại dòng Sort và đưa điều kiện vào dòng Criteria.
- Đóng và lưu truy vấn.

* ***Ví dụ:*** Đưa ra bảng tổng hợp tổng tiền các loại vật tư theo tháng.



Sau khi chạy truy vấn được bảng tổng hợp dữ liệu sau:

cau46 : Crosstab Query

LOAIVT	DVT	December	February	January
Bánh	Hộp			1080000
Bia	Thùng	2900000		
Kủo	Gói		0	
Nước ngọt	Thùng		660000	
Sữa	Hộp	1160000	0	800000
Thuốc lá	Tút	408000		800000

4. MAKE TABLE queries

4.1. Công dụng

Make Table queries là truy vấn tạo bảng. Kết quả mỗi lần thực hiện chạy truy vấn (run) sẽ được lưu thành một bảng mới.

4.2. Cách thực hiện

- Tạo truy vấn Select query

- Chọn biểu tượng Make-Table Query trên thanh công cụ của Menu Design xuất hiện hộp hội thoại có các lựa chọn sau:

+ Current Database: Chọn CSDL hiện tại

+ Another Database: Chọn CSDL khác

- Đặt tên bảng mới vào Table Name, nhấn Ok

- Đóng và lưu truy vấn.

(sau khi chạy truy vấn, Access sẽ tạo ra một bảng mới)

5. DELETE queries

5.1. Công dụng

Delete queries là truy vấn xóa dữ liệu. Mỗi lần chạy truy vấn sẽ thực hiện xóa dữ liệu trên bảng được chọn theo điều kiện.

5.2. Cách thực hiện

- Tạo truy vấn Select query

- Chọn biểu tượng Delete query trên thanh công cụ của Menu Design

- Chọn Where trên dòng Delete và nhập điều kiện xoá vào dòng Criteria. *(Nếu không có điều kiện thì các bản ghi sẽ bị xoá hết).*

- Đóng, lưu và chạy truy vấn.

(sau khi chạy truy vấn, dữ liệu trong bảng được chọn để tạo truy vấn sẽ bị xóa theo điều kiện đã đặt)

6. UPDATE queries

6.1. Công dụng

Update queries là truy vấn cập nhật dữ liệu. Mỗi lần chạy truy vấn sẽ thực hiện cập nhật dữ liệu thỏa mãn điều kiện vào bảng được chọn.

6.2. Cách thực hiện

- Tạo truy vấn Select query

- Chọn biểu tượng Update query trên thanh công cụ của Menu Design

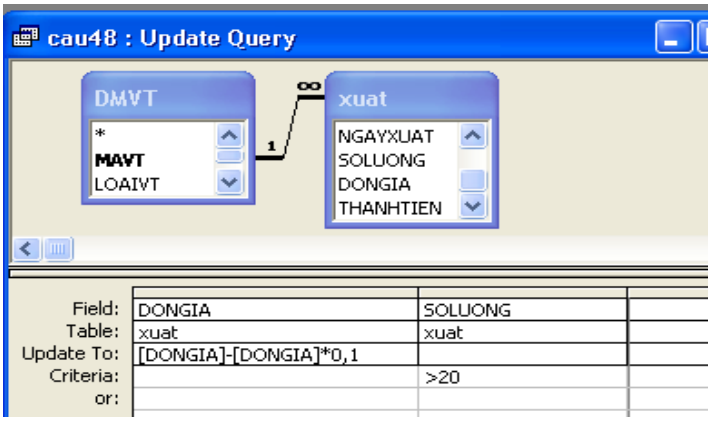
- Đưa trường cần cập nhật dữ liệu vào dòng Field.

- Nhập công thức cần cập nhật cho trường vào dòng Update.

- Đóng, lưu và chạy truy vấn

(khi chạy truy vấn máy sẽ hỏi bạn có cập nhật hay không? nếu muốn cập nhật chọn Yes, không muốn chọn No)

* ***Ví dụ*** : Cập nhật lại cho trường đơn giá với điều kiện giảm 10% đơn giá với những vật tư có số lượng >20:



Bài 3: THIẾT KẾ GIAO DIỆN

Thời gian : 13 giờ

1. Khái niệm Forms

Form hay còn gọi là mẫu biểu được sử dụng cho phép người sử dụng nhập thay đổi, trình bày dữ liệu và sử dụng form như giao diện ứng dụng. Việc trình bày dữ liệu thông qua form đã trở nên phổ đối với người thiết kế CSDL biến.

**** Cấu trúc của mẫu biểu***

Mỗi mẫu biểu gồm 5 thành phần sau:

- **Đầu biểu** (Form Header): Dùng để trình bày tiêu đề của form, hướng dẫn sử dụng các nút lệnh, các thông tin cố định của mẫu biểu. *(Có thể không in đầu biểu bằng cách đặt thuộc tính Display When: Screen Only. Có thể không cho hiển thị bằng thuộc tính Visible:No)*

- **Đầu trang** (Page Header): Dùng để trình bày các tiêu đề, chú thích trên đầu trang in.

- **Thân biểu** (Detail): Dùng để trình bày dữ liệu với nhiều hình thức bằng các ô điều khiển: Hộp văn bản (Text Box), Nhãn (Label), Nút lệnh (Command Button), Hộp lựa chọn (Combo Box), Hộp danh sách (List Box)...

- **Cuối trang** (Page Footer): Dùng để trình bày ngày tháng năm, số trang, tổng hợp dữ liệu bên dưới của trang in.

- **Cuối biểu** (Form Footer): giống đầu biểu

2. Sử dụng FORM WIZARD

* Chọn menu Create, chọn biểu tượng More Forms -> Form Wizard -> Xuất hiện hợp thoại -> thực hiện các bước tạo theo hướng dẫn:

- Chọn bảng hoặc truy vấn ở hộp Table/Query.

- Kích đúp vào các trường trong hộp thoại Avaiable Fields để đưa sang hộp thoại Select Fields *(Có thể lấy thêm các trường ở bảng hoặc truy vấn khác trong hộp thoại Table/ Query)* -> chọn Next.

- Chọn kiểu thể hiện dữ liệu của Form -> chọn Next

- Chọn nền Form -> chọn Next

- Đặt tiêu đề cho Form, chọn mở Form ở chế độ: Form to View hoặc Form's Design -> chọn Finish.

- Đóng và lưu Form.

3. Sử dụng FORM DESIGN VIEW

3.1. Cách thực hiện

* Chọn nhãn Form Design trên thanh công cụ của Menu Create -> xuất hiện Form thiết kế

- Chọn biểu tượng Add Eixsting Field trên thanh công cụ của menu Design để chọn bảng và trường cần đưa vào trong biểu mẫu.

- Kích đúp vào các trường trong bảng Field List để đưa các trường vào trong thân Form.

3.2. Tinh chỉnh cấu trúc Form

3.2.1. Thêm một Field vào mẫu biểu

Để thêm một trường vào mẫu biểu ta chọn field trong hộp Field List kéo vào trong Form hoặc kích đúp vào tên trường.

3.2.2. Chọn, di chuyển đối tượng

Để chọn một đối tượng trên Form ta chỉ cần chọn bằng cách kích chuột vào đối tượng cần chọn. Sau khi chọn đối tượng ta có thể di chuyển điều chỉnh hay thay đổi các thuộc tính của chúng. Có thể chọn một lúc nhiều đối tượng bằng cách nhấn Shift hay Ctrl trong lúc dùng chuột để chọn hoặc có thể quét một vùng các đối tượng trên Form. Ngoài ra có thể chọn các đối tượng bằng cách bấm chuột và di trên thước dọc và ngang.

Để di chuyển đối tượng có Lable đính kèm thì khi chọn chúng con chuột có hình bàn tay ta có thể di chuyển chúng đến vị trí khác. Di chuyển nhiều đối tượng thì chọn sau đó kéo chuột đến vị trí khác.

3.2.3. Canh lề

Có thể canh lề cho các đối tượng trên Form bằng cách chọn chúng và gán giá trị cho thuộc tính Left, Top của chúng. Hoặc nhấn phải chuột vào trường chọn Align -> chọn một trong các loại canh lề.

3.2.4. Điều chỉnh kích thước đối tượng

Để điều chỉnh kích thước đối tượng ta phải chọn đối tượng sau đó sử dụng chuột chọn các điểm đen trên 4 cạnh của đối tượng và điều chỉnh theo chiều mũi tên.

Ngoài ra ta có thể chọn các phương thức điều chỉnh có sẵn của Access bằng cách bấm phải chuột trên đối tượng chọn Size có các lựa chọn sau:

+ *To fit*: Kích thước của đối tượng được chọn bằng với chiều cao của các ký tự trong đối tượng.

+ *To Grid*: Kích thước đối tượng được chọn gắn với kích thước của đường lưới.

+ *To Tallest*: Kích thước đối tượng có chiều cao bằng chiều cao lớn nhất của một trong các đối tượng được chọn.

+ *To Shortest*: Kích thước đối tượng có chiều cao bằng chiều cao nhỏ nhất của một trong các đối tượng được chọn.

+ *To Widest*: Kích thước đối tượng có chiều rộng bằng chiều rộng lớn nhất của một trong các đối tượng được chọn.

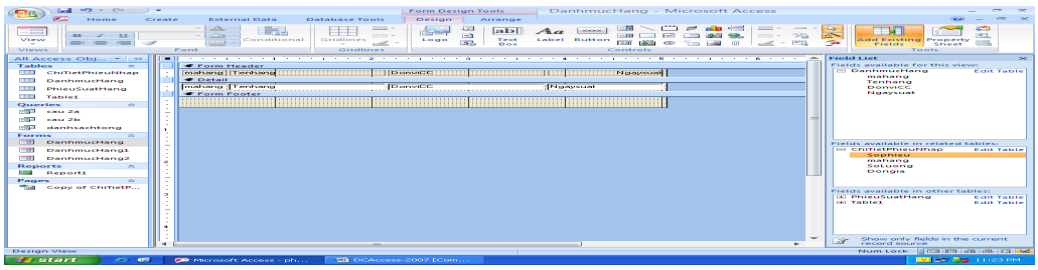
+ *To Narrowest*: Kích thước đối tượng có chiều rộng bằng chiều rộng nhỏ nhất của một trong các đối tượng được chọn.

3.2.5. Điều khiển khoảng cách giữa các đối tượng

Access sẽ điều chỉnh tất cả các khoảng cách giữa các đối tượng bằng cách chọn thực đơn Format/ Vertical Spacing (để điều chỉnh theo chiều dọc), chọn Format/Horizonal (để điều chỉnh theo chiều ngang).

3.3. Các ô điều khiển trên hộp Toolbox.

Các ô điều khiển này nằm trên thanh công cụ của thực đơn Design để thiết kế trên Form hoặc Report. Tuỳ từng ứng dụng mà nhà thiết kế có thể chọn các ô điều khiển sao cho phù hợp.



3.3.1. Lable

Dùng để nhập văn bản vào trong Form, Report. Dùng trong các tiêu đề cột, dữ liệu, nhóm tiêu đề. Ô điều khiển này không ràng buộc vào trường dữ liệu. Muốn gõ văn bản chỉ việc kích chuột vào ô Lable rồi kích ra Form, Report sau đó nhập văn bản vào.

3.3.2. TextBox

Gồm 2 phần dùng để trình bày dữ liệu của trường hay của biểu thức, chúng có thể dùng trên nhiều vùng của Report

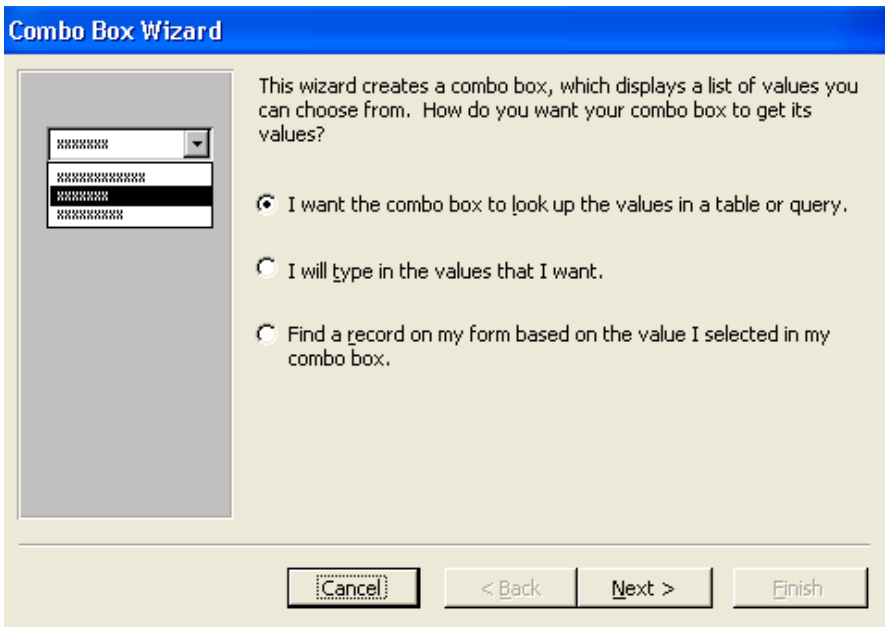
Phần đầu: để thể hiện tiêu đề của cột hay trường

Phần cuối: có thể ràng buộc vào một trường nào đó hoặc không hoặc chứa một biểu thức tính toán. Khi chạy Form hay Report dữ liệu sẽ thể hiện tại đây.

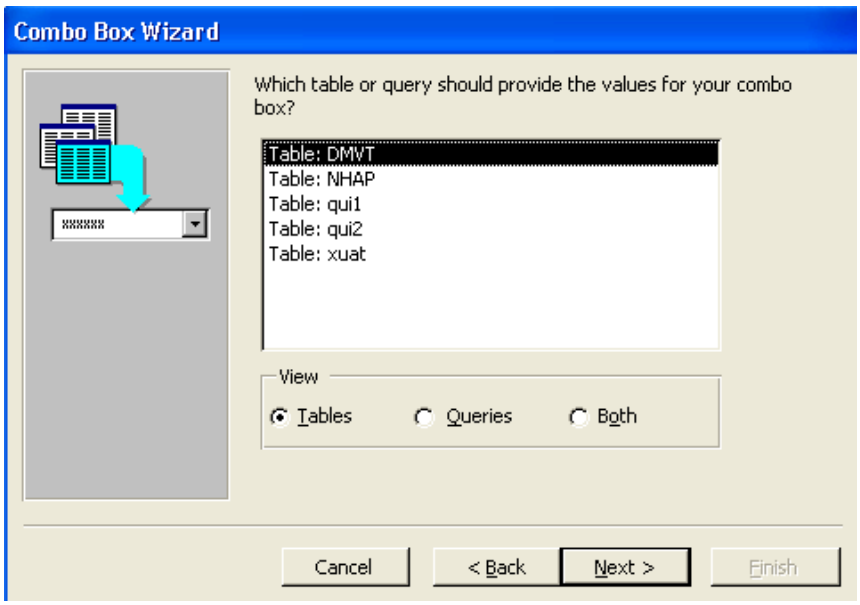
3.3.3. ComboBox

Dùng để liệt kê dữ liệu cho phép người dùng chọn duy nhất một giá trị phù hợp theo yêu cầu để nhập vào trong bảng. Dữ liệu trong combo box được lấy từ các bảng, truy vấn hay Lookup Cách tạo như sau:

Chọn nút ComboBox kích vào trong Form -> xuất hiện bảng chọn sau:

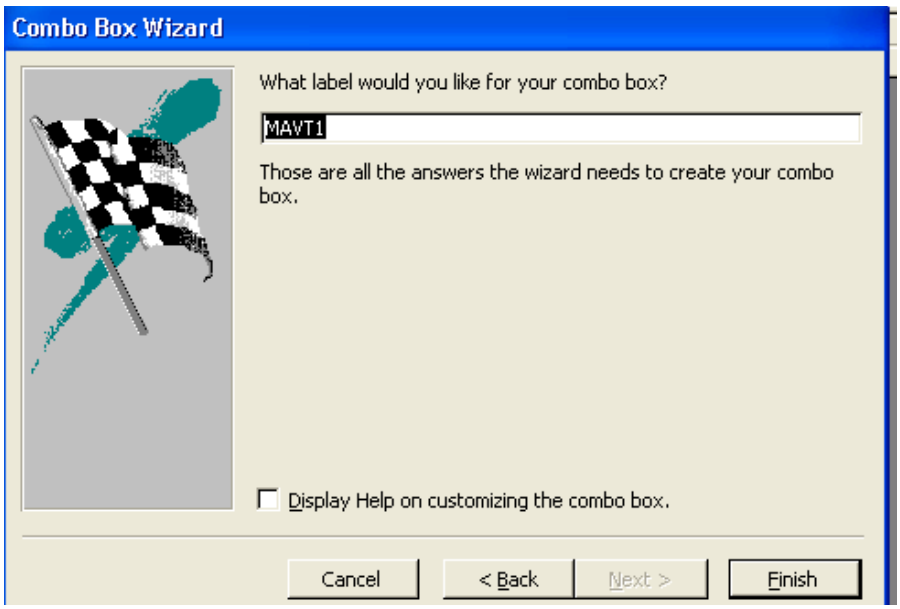


Tại đây lựa chọn mục đầu tiên (lấy giá trị từ bảng hoặc truy vấn) nhấn Next. Xuất hiện bảng cho phép chọn một bảng hay một truy vấn để lấy giá trị.



Chọn bảng xong nhấn Next. Xuất hiện hộp hội thoại tiếp theo cho phép chọn trường cần lấy dữ liệu. Muốn lấy trường nào thì kích đúp vào trường đó. Chọn xong nhấn Next sẽ xuất hiện hộp hội thoại hiển thị danh sách các giá trị của trường vừa chọn. Nhấn tiếp Next để chọn trường sẽ được nhận giá trị từ danh sách vừa chọn tại mục "Store that value in this field".

Chọn xong kích Next để gõ tên cho combo box rồi kích Finish để kết thúc.



3.3.4. ListBox

Dùng để liệt kê dữ liệu theo danh sách cho phép người sử dụng chọn một hoặc nhiều giá trị cùng một lúc. Cách thiết kế giống như Combo box.

3.3.5. Command Buttom

Thiết lập các hành động như mở bảng, chạy truy vấn, thêm bản ghi, ...Có thể tạo Command Button bằng Wizard như sau: Bật nút điều khiển Contrl trên hộp Toolbox, chọn nút Command rồi nhấn lên trên Form xuất hiện hộp hội thoại: Trong đó liệt kê các nhóm hành động.

a. Record Nevigation

+ *Find Next*: Tìm kiếm mẩu tin kế tiếp
+ *Find Recorf*: Mở cửa sổ tìm kiếm bản ghi bất kỳ
+ *Go First Record*: Về bản ghi đầu tiên
+ *Go Next Record*: Về bản ghi kế tiếp
+ *Go Last Record*: Về bản ghi cuối cùng
+ *Go Previous Record*: Về bản ghi trước đó

b. Record Operations

+ *Add New Record*: Thêm một bản ghi
+ *Delete Record*: Xoá bản ghi hiện thời
+ *Duplicate Record*: Thêm bản ghi có giá trị giống bản ghi hiện thời
+ *Print Record*: In bản ghi hiện thời
+ *Save Record*: Lưu bản ghi hiện thời
+ *Undo Record*: Phục hồi bản ghi hiện thời

c. Form Operation

+ *Close Form*: Đóng Form hiện thời
+ *Open Form*: Mở một Form
+ *Print a Form*: In màn hình Form
+ *Print Current Form*: In Form hiện thời
+ *Fefresh Form Data*: Làm tươi dữ liệu trên Form

d. Report Operation

+ *Mail Report*: Gửi Report bằng mail
+ *Preview*: Hiển thị báo cáo trước khi in
+ *Print Report*: In báo cáo
+ *Send Report to File*: Lưu báo cáo ra dạng File

e. Application

+ *Quit Application*: Thoát chương trình

+ *Run Application*: Thực thi một chương trình ứng dụng

+ *Run MS Excel*: Mở Microsoft Excel

+ *Run MS Word*: Mở Microsoft Word

+ *Run Notepad*: Mở Notepad

f. Miscellaneous

+ *Run Macro*: Thực thi một Macro

+ *Run Query*: Thực thi một truy vấn

+ *Print Table*: In bảng dữ liệu

Sau khi chọn xong các chức năng tương ứng trên, tiếp tục chọn Next sẽ xuất hiện bảng hội thoại cho phép chọn hinhd dạng hiển thị trên nút là Text hay Picture. Nếu là chọn Text thì phải gõ tên cho nút lệnh, nếu là Picture thì có thể tìm hình ảnh bằng cách chọn Browse để tìm ảnh.

3.3.6. Image

Điều khiển này không có ràng buộc với trường nào. Thường dùng để chèn ảnh vào trong Form hay Report. Thực hiện bằng cách chọn ô điều khiển vẽ lên trên Form sau đó Access yêu cầu chọn tệp ảnh rồi nhấn OK.

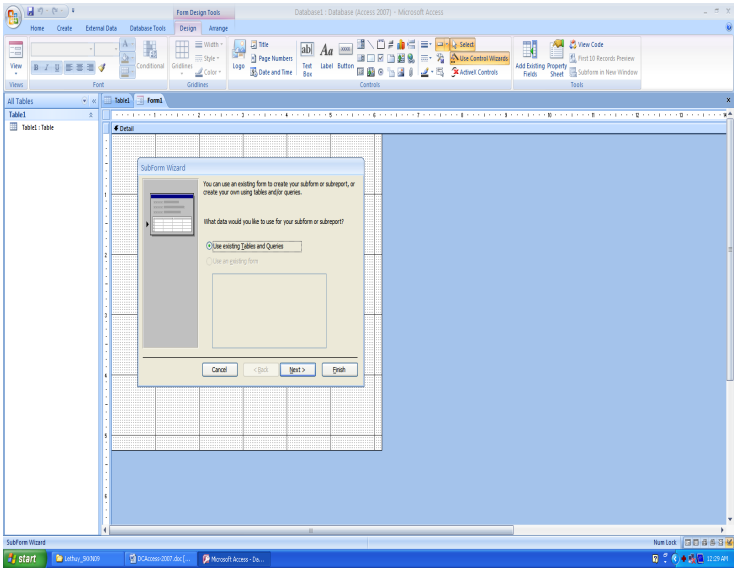
4. Kỹ thuật Sub-form

4.1. Công dụng

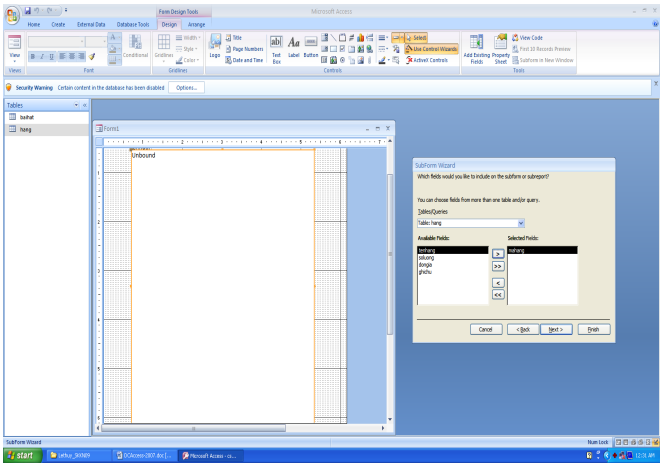
Kỹ thuật Sub-Form là thực hiện chèn thêm một Form phụ vào trong Form hoặc Report chính có quan hệ 1-n trong CSDL nhằm trình bầy dữ liệu chi tiết hơn.

4.2. Cách thực hiện

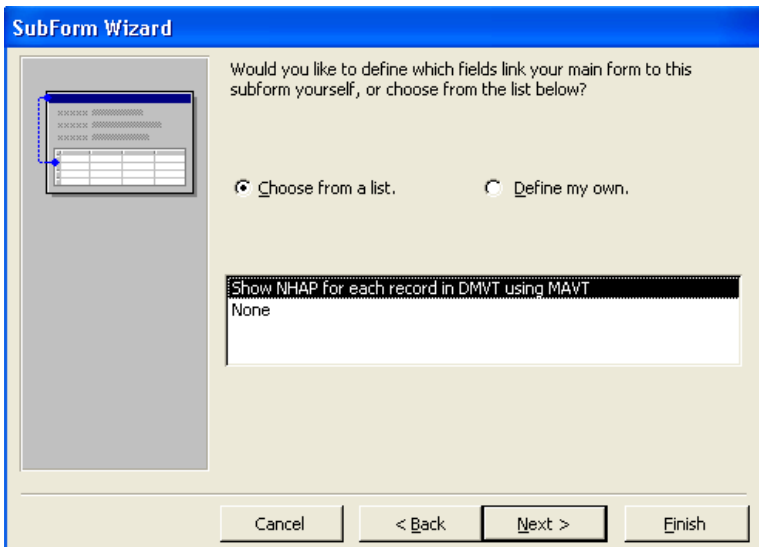
Chọn ô điều khiển Subform vẽ lên Form chính xuất hiện hộp hội thoại sau:



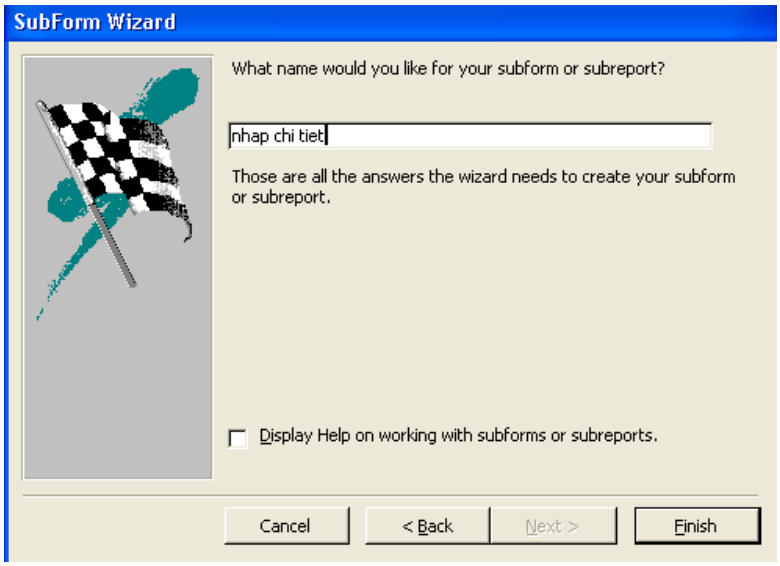
Trên hộp hội thoại này chọn Use existing Table and Queries rồi nhấn Next.



Tại hộp Table/Queries chọn bảng hoặc truy vấn, kích đúp vào các trường cần đưa vào trong Subform rồi nhấn Next.



Tại cửa sổ này lựa chọn giống như hình trên rồi nhấn Next.



Nhập tiêu đề cho Subform sau đó nhấn Finish để hoàn thành.

Bài 4 : THIẾT KẾ BÁO CÁO

Thời gian : 12 giờ

1. Các khái niệm về Report

Report hay còn gọi là báo biểu là một kiểu biểu mẫu đặc biệt được thiết kế để in ấn, trong báo biểu, Access tổ hợp dữ liệu trong bảng và truy vấn để có thể in theo những yêu cầu cụ thể.

1.1. Cấu trúc Report

Mỗi báo biểu gồm 5 thành phần sau:

- Page Header: Phần này chứa thông tin tựa đề của các trang báo cáo

- Page Footer: Chứa thông tin như số trang, ngày in…

- Detail: Là phần chính của báo cáo dùng để trình bày dữ liệu chi tiết của bảng hay truy vấn

- Report Header: Chứa thông tin chung về báo cáo

- Report Footer: Giống Report Header

1.2. Môi trường làm việc

Báo biểu trong Access có đầy đủ tính năng cho phép người sử dụng trình bày dữ liệu dưới nhiều hình thức khác nhau:

- Dạng chi tiết: dùng khi trình bày thông tin chi tiết về một đối tượng nào đó như thông tin về nhân viên, chi tiết hóa đơn…

- Dạng tổng hợp: dùng khi thống kê dữ liệu theo nhóm (theo field nào đó) như tổng hợp doanh thu bán hàng theo từng quý từng năm….

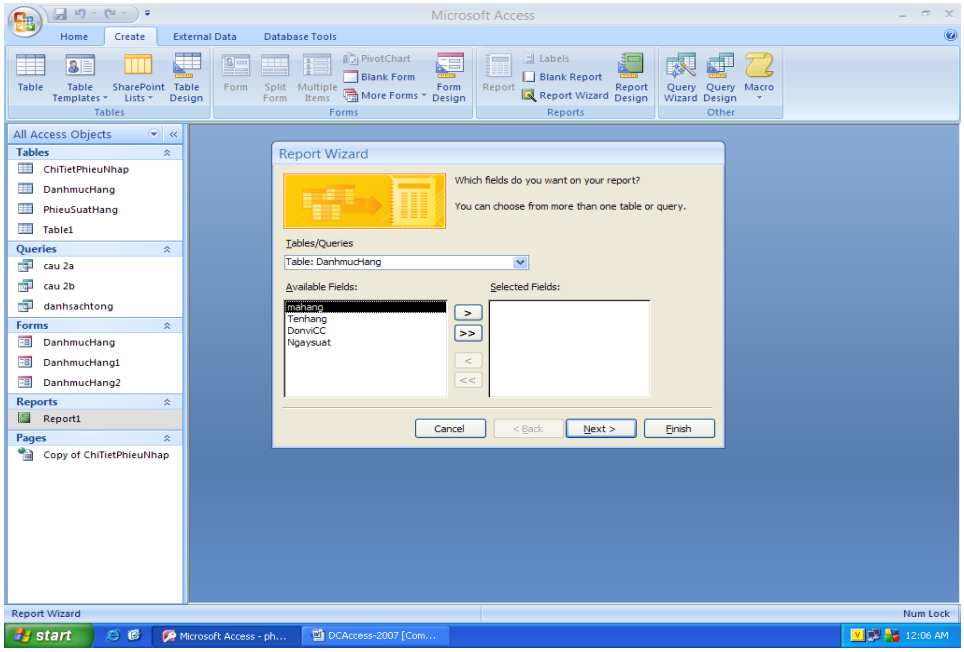
- Dạng đồ thị: dùng khi cần thể hiện dữ liệu dưới dạng đồ thị như thể hiện quá trình kinh doanh của công ty qua doanh thu, tiến độ công việc …

- Dạng form: khi muốn có báo cáo giống như form thì trong Access cung cấp các công cụ đồ họa để thiết kế các báo cáo như các đường kẻ, khung, hình ảnh.

- Dạng lable: khi cần thiết kế nhiều nhãn cùng kích thước trên một trang như phong bì để gửi cho khách hàng…

2. Sử dụng Report wizard

- Chọn nhãn Report wizard trên thanh công cụ của menu Create xuất hiện giao diện như sau:



- Chọn New/ Report Wizard, chọn bảng hoặc truy vấn ở hộp thoại bên dưới. Nhấn Ok.

- Kích đúp vào các trường trong hộp thoại Avaiable Fields để đưa sang hộp thoại Select Fields. Có thể lấy thêm các trường ở bảng hoặc truy vấn khác ở hộp danh sách Table/ Query, chọn Next.

- Chọn cách hiển thị dữ liệu của SubReport, chọn Next.

- Chọn trường cần nhóm dữ liệu bằng cách kích đúp vào tên trường ở hộp thoại bên trái, chọn nhãn Grouping Option, chọn kiểu nhóm trong hộp thoại Grouping Intervals, chọn Next

- Chọn các trường cần sắp xếp theo thứ tự tăng dần hoặc giảm dần, chọn nut Summery option, chọn hàm cần tính toán cho các trường số,chọn OK/Next.

- Chọn kiểu hiện thị các nhóm dữ liệu và hướng giấy ngang hoặc dọc, chọn Next.

- Chọn màu và kiểu chữ cho các tiêu đề, chọn Next.

- Đặt tiêu đề cho Report, chọn kiểu hiển thị mở Report ở chế độ View hoặc chế độ Design, chọn Finish.

- Đóng và lưu Report.

3. Thiết kế Report

3.1. Các bước thực hiện

- Chọn biểu tượng Report Design trên thanh công cụ của menu Create.

- Mở bảng Add Exitsting Field

- Để mở danh sách các trường bằng cách vào View/ Field List.

- Kích đúp vào các trường trong bảng Field List để đưa các trường vào trong thân Report.

Chú ý: *ở cách tạo này chỉ lấy dữ liệu từ một bảng hoặc một truy vấn, muốn lấy thêm dữ liệu từ bảng hoặc truy vấn có quan hệ với nó thì sử dụng điều khiển SubForm/SubReport trên hộp Toolbox và đưa vào trong phần thân Report. Cách làm giống như thêm một Form phụ vào Form chính.*

3.2. Sắp xếp và phân nhóm

+ Tại cửa sổ thiết kế chọn chọn biểu tượng Group & Sort

+ Chọn vào Add a Group hoặc Add a sort để nhóm hoặc sắp xếp.

3.3. Chỉnh sửa báo biểu

- Phải chuột vào Report cần sửa, kích nhãn Design. Có thể thêm hoặc bỏ đầu trang, cuối trang, đầu Report, cuối Report bằng cách vào menu View, chọn Page Header/Footer hoặc Report Header/Footer.

- Chọn Font chữ, kích cỡ, màu cho các nhãn và text box

- Có thể đưa thêm các điều khiển text box để tính toán và tổng hợp dữ liệu.
- Đánh số thứ tự, số trang và ngắt trang.

Chú ý: *Nếu muốn xoá một điều khiển nào đó ta kích chọn điều khiển đó, chọn Delete. Có thể thay đổi tên các nhãn, tiêu đề bằng cách đưa trỏ chuột vào sửa.*

Bài 5: LẬP TRÌNH VBA CĂN BẢN

Thời gian: 12 giờ

1. Môi trường lập trình VBA

1.1. Giới thiệu về Visual Basic

1.1.1. Khái niệm dự án

- Trong Visual Basic mọi thành phần như chương trình (modul), giao diện (các form *.frm) … của một ứng dụng đều được gói gọn vào một Project gọi là dự án.

-Dự án (Project) là sự tập hợp các files sử dụng để tạo một trình ứng dụng, và khi lưu dự án luôn có một file chính có phần mở rộng là **vbp**.

- Khi viết một chương trình có nghĩa là triển khai một dự án (Project).

- Khi thiết kế, các chương trình và giao diện (form) của cùng một dự án được quản lý trong một Project. Chính vì vậy mà VB có khả năng đóng gói chương trình dịch ra file exe.

1.1.2. Khởi động Visual Basic 6.0

+ Cách 1: Start→ Programs→ MicrosoftVisual Studio 6.0→ Microsoft VisualBasic 6.0

+ Cách 2: Nhấp đúp chuột vào biểu tượng Microsoft Visual Basic 6.0

Sau khi khởi động xuất hiện cửa sổ New Project:

- Chọn New →Standard EXE: Mở 1 Project mới.

- Chọn Existing: Mở 1 Project đã lưu trên đĩa.

- Chọn Recent: Mở 1 Project trong các Project đã được mở gần đây nhất.

1.1.3. Tạo, mở, ghi một file Project

(Thực hiện khi đang ở trong VB)

- Tạo một file project mới: File → New project → Standard EXE

- Mở một file project đã có : File → Open → chọn đường dẫn và chọn file.

- Ghi một file project:

File → Save project → chọn đường dẫn, gõ tên file .

1.1.4. Tạo, mở và ghi một Form

(Khi mở một Project mới thì VB sẽ tự động có một Form mới. Thực hiện các lệnh sau khi muốn thêm các Form cho Project)

- Tạo một Form mới: Project → Add Form → Form → Open

hoặc phải chuột vào tên Project trên cửa sổ Project chọn Add Form.

- Mở một Form đã có : Kích đúp vào tên Form trên cửa sổ Project
- Ghi một Form: File → Save (tên form chọn ghi)…

1.2. Các thành phần trên cửa sổ IDE

Cửa sổ quản lý dự án (Project Explorer) giúp chúng ta quản lý và định hướng nhiều đề án. VB6 cho phép tổ chức nhiều đề án trong một nhóm. Project Explorer có cấu trúc cây phân cấp.

Cửa sổ thuộc tính:Mỗi thuộc tính có một hoặc nhiều giá trị. Cửa sổ thuộc tính giúp bạn xem và sửa các thuộc tính của điều khiển ActiveX.

Cửa sổ Layout: cho phép qui định vị trí của các biểu mẫu (Form)

Thanh menu: là các bộ chức năng trong môi trương tích hợp(IDE) của VB

Thanh công cụ: gồm có thanh công cụ chuẩn, soạn thảo, gỡ lỗi chương trình (debug),…

Hộp công cụ: là bảng chứa các điều khiển dùng để thiết kế giao diện và các chức năng của chương trình.

1.3. Các bước lập trình

1.3.1. Thiết kế giao diện

- Tạo một Form mới
- Đưa các điều khiển vào Form (Textbox, Label, CommandButton, …)
- Cố định các điều khiển trên Form: Format → Lock Controls

- Thiết lập thuộc tính cho các điều khiển (Name, Caption, Text, Font, …)

1.3.2. Viết Code cho chương trình

- Bấm đúp vào đối tượng điều khiển để viết sự kiện tương ứng cho đối tượng đó. Tên của đối tượng là gì thì sự kiện sẽ có tên tương ứng.

1.3.3. Hoàn thiện chương trình

- Lưu Project và Form

- Thiết lập Form chạy mặc định (Nếu trong Project chỉ có một Form thì không cần thực hiện bước này)

- Chạy chương trình (F5)

- Dịch chương trình thành file .exe

1.4. Dịch và chạy chương trình

1.4.1. Thiết lập Form chạy mặc định

(Form chạy mặc định là form sẽ được hiển thị khi thực hiện dịch và chạy chương trình. Trong một Project có thể có nhiều form, do đó khi làm làm việc với form nào thì cần thiết lập cho form đó chạy mặc định.)

Project → Project properties → General → Startup object → Tên Form.

1.4.2. Chạy chương trình

VB cho phép chạy thử từng bước một, sau khi thiết kế và viết lệnh có thể thực hiện chạy thử với từng điều khiển mà không cần phải đợi đến khi hoàn chỉnh toàn bộ chương trình

- **Run → Start** hoặc nhấn nút **F5** hoặc bấm vào biểu tượng (►)

1.4.3. Dịch chương trình

Chương trình sau khi đã hoàn thành có thể dịch ra đuôi exe để chạy và sao chép, di chuyển nhằm giấu mã nguồn:

- File → Make Project.exe

2. Các kiểu dữ liệu và khai báo

2.1. Các kiểu dữ liệu cơ bản

Kiểu dữ liệu là một tập hợp các giá trị mà một biến của kiểu có thể nhận và một tập hợp các phép toán có thể áp dụng trên các giá trị đó.

2.1.1. Dữ liệu kiểu số

a. Kiểu

Byte: kiểu số nguyên (0 → 255)

Integer: kiểu số nguyên (-32768 → 32767)

Long: kiểu số nguyên (-2147483648 → 2147483647)

Single: kiểu số thực (-3.403*1038 → 3.403*1038)

Double: kiểu số thực (1.7977*10308 → 1.7977*10308)

Currency: kiểu tiền tệ (-922*1012→ 922*1012 -1)

b. Các phép toán

+, -, *, /

\: phép chia lấy phần nguyên

Mod: phép chia lấy phần dư

^ : lũy thừa *(2^3 = 8)*

c. Một số hàm thông dụng

Abs(N): trả về giá trị tuyệt đối của số N

Int(N): trả về giá trị phần nguyên của số N

Sqr(N): tính căn bậc hai của số N

Round(N,m): làm tròn số N với m số phần thập phân

IsNumeric(N): trả về giá trị logic là True hoặc False cho biết giá trị N có phải là kiểu số không.

2.1.2. Dữ liệu kiểu chuỗi ký tự (String)

a. Kiểu

Mặc định các ký tự hiển thị hoặc được nhập vào các điều khiển trên giao diện là dữ liệu kiểu chuỗi kí tự. Có hai đặc tả chuỗi ký tự theo cú pháp như sau:

String * <n>: xác định một chuỗi ký tự có độ dài cố định là n ký tự.

String: không xác định chiều dài tối đa của chuỗi

b. Các phép toán

Phép **+** hoặc **&**: thực hiện ghép chuỗi

c. Một số hàm thông dụng

Trim(X): cắt bỏ các ký tự trắng ở hai đầu chuỗi X.

Len(X): trả về một giá trị số là độ dài của chuỗi X.

Left(X,n): trả về một chuỗi con được trích từ chuỗi gốc lấy n ký tự bên trái.

Right(X,n): trả về một chuỗi con được trích từ chuỗi gốc lấy n ký tự bên phải.

Mid(X,m,n): trả về một chuỗi con được trích từ chuỗi gốc lấy n ký tự tính từ ký tự thứ m sang bên phải.

InStr(n,X,Y): trả về một giá trị số là vị trí tìm thấy chuỗi con Y trong chuỗi gốc X từ vị trí n. Nếu không tìm thấy thì giá trị trả về là 0. Thực hiện tìm từ trái sang phải.

InStrRev(X,Y,n): trả về một giá trị số là vị trí tìm thấy chuỗi con Y trong chuỗi gốc X từ vị trí n. Nếu không tìm thấy thì giá trị trả về là 0. Thực hiện tìm từ phải sang trái.

Replace(X,y,z): trả về một chuỗi ký tự mới bằng cách thay thế chuỗi y bằng chuỗi z trong chuỗi gốc X.

- *Ví dụ: Replace("Hoa hồng đỏ","hồng","phượng")="Hoa phượng đỏ"*

StrReverse(X): đảo ngược các ký tự trong chuỗi X

UCase(X): đổi tất cả các ký tự trong chuỗi X thành ký tự hoa

Lcase(X): đổi tất các ký tự trong chuỗi X thành ký tự thường

ASC(kí tự): trả về giá trị mã ASCII của ký tự

Chr(n): trả về ký tự tương ứng với mã ASCII n

2.1.3. Dữ liệu kiểu ngày tháng (Date)

a. Kiểu

Dữ liệu kiểu ngày tháng (Date) là kiểu mà các biến của nó chứa giá trị ngày tháng. Để cho VB biết dữ liệu là kiểu Date cần đặt dữ liệu đó giữa hai dấu # (hoặc “”). *(Dữ liệu kiểu ngày tháng sẽ phụ thuộc vào định dạng của máy là 'dd/mm/yy' hay 'mm/dd/yy')*

b. Một số hàm thông dụng

Isdate(D): trả về giá trị logic là True hoặc False cho biết chuỗi D có phải là kiểu ngày tháng

Day(D): trả về giá trị ngày của dữ liệu kiểu ngày tháng D

Month(D): trả về giá trị tháng của dữ liệu kiểu ngày tháng D

Year(D): trả về giá trị năm của dữ liệu kiểu ngày tháng D

Weekday(D): trả về giá trị thứ trong tuần của dữ liệu kiểu ngày tháng D

Date: trả về giá trị 'ngày/tháng/năm' hiện tại của hệ thống.

Now: trả về giá trị 'ngày/tháng/năm giờ : phút : giây' hiện tại của hệ thống.

**** Chú ý**: Khi nhập dữ liệu kiểu ngày tháng vào textbox thì phải sử dụng dấu phân cách '/', '-' hoặc 'dấu cách'.*

2.1.4. Dữ liệu kiểu logic (Boolean)

a. Kiểu

Dữ liệu kiểu logic là dữ liệu chỉ trả về một trong hai giá trị là True/False. Dữ liệu kiểu này thường dùng trong các phép toán kiểm tra.

b. Các phép toán

And: phép và

Or: phép hoặc

Not: phép phủ định

2.2. Biến và cách khai báo biến

2.2.1. Khái niệm

Biến là vùng lưu trữ được đặt tên để chứa dữ liệu tạm thời trong quá trình tính toán, so sánh và các công việc khác. Giá trị của biến luôn được thay đổi trong cả quá trình thực hiện chạy chương trình. Biến có 2 đặc điểm sau:

Mỗi biến có một tên.

Mỗi biến có thể chứa duy nhất một kiểu dữ liệu.

2.2.2. Khai báo biến

Cách 1: **[Global| Public| Private]Dim** <tên biến> **[As** <kiểu dữ liệu> **]**

Cách 2: **Dim** <tên biến> [**As** <kiểu dữ liệu>]

Trong đó:

<tên biến>: được đặt theo quy tắc đặt. Có thể khai báo nhiều biến trên một dòng, phân cách nhau bởi dấu phẩy (,).

[Global]: khai báo biến toàn cục *(biến được dùng chung cho cả dự án, nó thường được khai báo trong một module)*

[Public]: khai báo biến dùng chung *(dùng cho một biểu mẫu, các biến này được khai báo ở vùng General, vùng đầu tiên trên cửa sổ lệnh của biểu mẫu)*

[Private]: khai báo biến cục bộ *(dùng trong thủ tục/ hàm, các biến này được khai báo sau dòng lệnh Sub hoặc Function)*

Ví dụ: Dim a As Integer, b As Double

VB cho phép không cần phải khai báo một biến trước khi sử dụng nhưng sẽ gây một số sai sót, chẳng hạn khi gõ nhầm tên biến, VB sẽ hiểu đó là một biến mới dẫn đến kết quả chương trình sai mà rất khó phát hiện. Do đó nên quy định rằng VB sẽ báo lỗi khi gặp biến chưa được khai báo bằng dòng lệnh: ***Option Explicit*** *đặt ở vị trí trên cùng của cửa sổ soạn thảo mã lệnh, trước khi viết các lệnh khác*

2.3. Hằng và cách khai báo hằng

2.3.1. Khái niệm

Hằng số là giá trị dữ liệu xác định cụ thể, không thay đổi trong suốt quá trình thực hiện chương trình.

2.3.2. Khai báo hằng

Cách 1: [**Public|Private**]**Const**<tên hằng>[**As**<kiểu dữ liệu>]=<biểu thức>

Cách 2: **Const** <tên hằng> = <biểu thức>

Trong đó:

<tên hằng> được đặt theo quy tắc đặt tên. Tất cả các hằng có thể khai báo trên cùng một dòng, phân cách nhau bởi dấu phẩy(,)

[*Public*]: khai báo hằng dùng chung (sử dụng cho một mẫu biểu)

[*Private*]: khai báo hằng cục bộ *(chỉ sử dụng trong thủ tục, modul)*

- Ví dụ:

Public Const Pi As Double = 3.14

Const g = 9.8

3. Các cấu trúc lệnh VBA

3.1. Cấu trúc rẽ nhánh

3.1.1. Câu lệnh điều kiện đơn

a. Cú pháp

Cp1: **If** <điều kiện> **Then** <lệnh>

Cp2: **If** <điều kiện> **Then**

<khối lệnh>

End If

Trong đó:

<điều kiện>: là biểu thức logic trả về giá trị True hoặc False. *Biểu thức logic thường chứa các phép toán so sánh và phép toán logic.*

<khối lệnh>: có thể là một lệnh hoặc nhiều lệnh.

b. Ý nghĩa

Khi gặp cấu trúc lệnh IF...THEN máy sẽ kiểm tra điều kiện. Nếu điều kiện đúng (True) thì sẽ thực hiện lệnh sau Then. Ngược lại, nếu điều kiện sai (False) thì sẽ bỏ qua lệnh này không thực hiện gì cả. *(lệnh chỉ được thực hiện khi điều kiện đúng)*

3.1.2. Câu lệnh điều kiện rẽ nhánh

a. Cú pháp

Cp1: **If** <điều kiện> **Then**

<khối lệnh 1>

Else

<khối lệnh 2>

End If

Cp2: **If** <điều kiện 1> **Then**

<khối lệnh 1>

ElseIf <điều kiện 2> **Then**

<khối lệnh 2>

… …

ElseIf <điều kiện n> **Then**

<khối lệnh n>

Else

<khối lệnh n+1>

End If

b. Ý nghĩa

(1) Khi gặp cấu trúc lệnh IF … THEN … ELSE máy sẽ kiểm tra điều kiện. Nếu điều kiện đúng (True) thì sẽ thực hiện khối lệnh 1 sau Then. Ngược lại, nếu điều kiện sai (False) thì sẽ thực hiện khối lệnh 2 sau Else. *(Như vậy tùy theo điều kiện đúng sai mà thực hiện một trong hai lệnh 1 hoặc 2)*

(2) Khi gặp cấu trúc lệnh IF … THEN .. ELSEIF máy sẽ kiểm tra điều kiện 1 trước tiên. Nếu điều kiện 1 đúng thì thực hiện khối lệnh 1. Nếu điều kiện 1 sai thì kiểm tra điều kiện 2. Nếu điều kiện 2 đúng thì thực hiện khối lệnh 2, ngược lại nếu điều kiện 2 sai thì tiếp tục kiểm tra các điều kiện tiếp theo cho đến điều kiện thứ n. Nếu tất cả các điều kiện kiểm tra đều không đúng thì thực hiện khối lệnh <n+1> sau Else. *(Như vậy điều kiện nào đúng thì thực hiện khối lệnh tương ứng, nếu tất cả đều sai thì thực hiện khối lệnh cuối cùng sau Else)*

3.1.3. Lệnh lựa chọn

Trong trường hợp có quá nhiều các điều kiện cần phải kiểm tra, nếu ta dùng cấu trúc rẽ nhánh If…Then thì đoạn lệnh không được trong sáng, khó kiểm tra, sửa đổi khi có sai sót. Ngược lại với cấu trúc Select…Case, biểu thức điều kiện sẽ được tính toán một lần vào đầu cấu trúc, sau đó VB sẽ so sánh kết quả với từng trường hợp (Case). Nếu bằng nó thi hành khối lệnh trong trường hợp (Case) đó

a. Cú pháp

Select Case <biểu thức điều kiện>

Case <ds giá trị 1>

<khối lệnh 1>

Case <ds giá trị 2 >

<khối lệnh 2>

… …

Case <ds giá trị n>

<khối lệnh n>

[Case Else

<khối lệnh n+1>**]**

End Select

Trong đó:

***<biểu thức điều kiện>*:** giá trị trả về phải là dữ liệu kiểu số nguyên (*Byte, Integer*) hoặc chuỗi kí tự (*String*)

***<ds giá trị>*:** có thể một giá trị, nếu nhiều giá trị thì các giá trị này phân cách nhau bởi dấu phẩy (,)

***<khối lệnh>*:** có thể là không. Một hoặc nhiều dòng lệnh

[*CaseElse*]: có thể có hoặc không

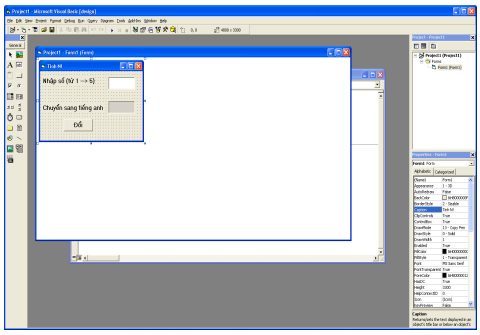
b. Ý nghĩa

Với câu lệnh này chương trình sẽ thực hiện tính toán và kiểm tra biểu thức điều kiện trước, sau đó so sánh với từng giá trị ở từng trường hợp. Nếu kết quả

của biểu thức đúng với giá trị của trường hợp nào thì thực hiện khối lệnh của trường hợp đó. Nếu có nhiều trường hợp cùng thỏa mãn biểu thức điều kiện thì khối lệnh của trường hợp đầu tiên sẽ được thực hiện, ngược lại nếu không có trường hợp nào đúng thì khối lệnh sau CaseElse sẽ được thực hiện.

- Ví dụ

Viết chương trình nhập vào một số, thực hiện đổi số sang tiếng Anh



Mã lệnh:

```
Private Sub CmdDoi_Click()
    Select Case CInt(txtSo.Text)
        Case 1
            txtThongbao.Text = “One”
        Case 2
            txtThongbao.Text = “Two”
        Case 3
            txtThongbao.Text = “Three”
        Case 4
            txtThongbao.Text = “Four”
        Case 5
            txtThongbao.Text = “Five”
        Case Else
            txtThongbao.Text = “ Số nhập vào không thỏa mãn”
    End Select
End Sub
```

3.2. Cấu trúc lặp For

3.2.1. Khái niệm

Cấu trúc lặp For là cấu trúc lệnh lặp xác định, biết trước số lần lặp bằng cách dùng biến đếm tăng dần hoặc giảm dần để xác định số lần lặp đó.

3.2.2. Cấu trúc

a. Cú pháp

For <biến đếm> = <giá trị 1> **To** <giá trị 2> [**Step** <bước nhảy>]

<khối lệnh>

Next

Trong đó:

<biến đếm>: là giá trị có kiểu dữ liệu số (Integer, Single,…)

<giá trị 1>: là giá trị được gán đầu tiên cho biến đếm, phải có cùng kiểu dữ liệu với biến đếm

<giá trị 2>: là giá trị được gán cuối cùng cho biến đếm, có cùng kiểu dữ liệu với biến đếm

<bước nhảy>: có thể là số âm hoặc số dương có cùng kiểu dữ liệu với biến đếm. Nếu bước đếm là số dương thì giá trị 1 < giá trị 2, ngược lại nếu biến đếm là số âm thì giá trị 1> giá trị 2. Khi lệnh Step không được chỉ ra thì VB mặc định bước nhảy là 1.

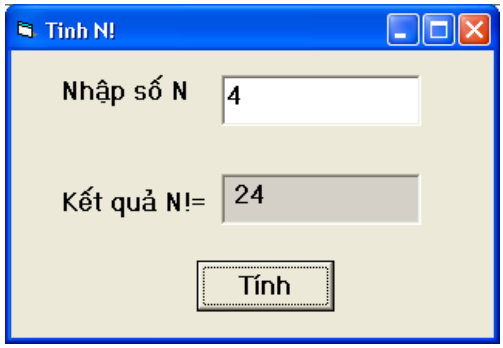
b. Ý nghĩa

Đầu tiên chương trình sẽ gán <biến đếm> = <giá trị 1> sau đó thực hiện <khối lệnh>. Khi gặp lệnh Next, chương trình sẽ tăng <biến đếm> lên một đơn vị hoặc tăng (giảm) theo bước nhảy Step (nếu có) rồi thực hiện lại khối lệnh. Cứ như vậy khối lệnh được thực hiện lặp đi lặp lại cho đến khi biến đếm vượt quá <giá trị 2> thì kết thúc.

***- Ví dụ**: Tính N!*

```
Private Sub CmdTinh_Click()

    Dim i, kq As Integer
     Kq =1
          For i=1 To Cint(txtso.Text)
               kq=kq*i
          Next
      txtKetqua.Text = Str(kq)
  End Sub
```



3.3. Cấu trúc lặp While

3.3.1. Khái niệm:

Cấu trúc lặp While là cấu trúc lặp không xác định, có số lần lặp không đặt trước mà sẽ được quyết định bởi một biểu thức điều kiện.

3.3.2. Cấu trúc Do While … Loop

a. Cú pháp

Do While <điều kiện>

<khối lệnh>

Loop

Trong đó:

<điều kiện>: là biểu thức logic trả về giá trị True hoặc False. *Biểu thức logic thường chứa các phép toán so sánh và phép toán logic.*

<khối lệnh>: có thể là một lệnh hoặc nhiều lệnh.

b. Ý nghĩa

Khi gặp cấu trúc lệnh DO WHILE … LOOP, chương trình sẽ thực hiện kiểm tra điều kiện, nếu điều kiện đúng sẽ thực hiện khối lệnh rồi quay lên kiểm tra lại điều kiện. Khối lệnh sẽ được thực hiện lặp cho đến khi điều kiện sai thì dừng. Biểu thức điều kiện được kiểm tra trước khi thi hành khối lệnh, do đó nếu trong lần kiểm tra đầu tiên mà điều kiện sai thì khối lệnh sẽ không được thực hiện một lần nào cả.

3.3.3. Cấu trúc Do … Loop While

a. Cú pháp

Do

<khối lệnh>

Loop While <điều kiện>

b. Ý nghĩa

Khi gặp cấu trúc lệnh DO… LOOP WHILE, chương trình sẽ thực hiện khối lệnh trước rồi mới kiểm tra điều kiện. Nếu điều kiện đúng sẽ thực hiện lặp lại khối lệnh cho đến khi điều kiện sai thì dừng. Do biểu thức điều kiện được kiểm tra sau nên dù điều kiện sai ngay từ đầu thì khối lệnh cũng sẽ được thực hiện ít nhất một lần.

4. Chương trình con

Chương trình con là những đoạn chương trình (module) được viết để giải quyết một công việc hoặc một phần công việc nào đó. Trong Visual Basic, chương trình con có hai dạng là hàm (Function) và thủ tục (Sub).

Mục đích sử dụng chương trình con:

Chia nhỏ chương trình lớn thành nhiều phần logic tránh viết lặp đi lặp lại, tránh rườm rà

Dễ dàng kiểm tra xác định tính đúng đắn của thuật toán, giúp gỡ rối, gỡ lỗi chương trình một cách dễ dàng

Có thể được sử dụng lại trong nhiều ứng dụng khác

4.1. Chương trình con dạng hàm

4.1.1. Khái niệm

Hàm là một chương trình con chứa các lệnh để thực hiện một hoặc một số công việc nào đó mà kết quả trả về là một giá trị cụ thể và giá trị đó được gán vào tên hàm.

4.1.2. Khai báo hàm

Function <Tên hàm>[(tham số)] **As** <kiểu dữ liệu>

<khối lệnh>

<Tên hàm> = <kết quả trả về>

End Function

Trong đó:

<Tên hàm>: theo quy tắc đặt tên

[<tham số>]: có thể có hay không

<kiểu dữ liệu>: là kiểu dữ liệu của kết quả do hàmtrả về.

4.1.3. Gọi hàm thực thi

Khi gọi hàm để thực thi ta nhận được một kết quả có kiểu dữ liệu chính là kiểu trả về của hàm. Do đó lời gọi hàm phải là thành phần của một biểu thức.

Cú pháp gọi hàm thực thi:

<Đối tượng> = <Tên hàm>[(Tham số thực tế)]

4.2. Chương trình con dạng thủ tục

4.2.1. Khái niệm

Thủ tục là một chương trình con chứa các lệnh để thực hiện một hoặc một số công việc nào đó mà không trả về một giá trị cụ thể.

4.2.2. Khai báo thủ tục

Sub <Tên thủ tục> [(<tham số>]

<khối lệnh>

End Sub

Trong đó:

<***Tên thủ tục***>: theo quy tắc đặt tên

[<***tham số***>]: có thể có hoặc không.

4.2.3. Gọi thủ thục

Để gọi thủ tục sử dụng một trong hai cách sau:

<Tên thủ tục> [<Tham số thực tế>]

Call <Tên thủ tục> ([<Tham số thực tế>])

Ví dụ về chương trình con

Viết chương trình tính giai thừa của số N được nhập vào từ bàn phím.

- Thủ tục tính N!

```
Dim i, kq As Integer
Sub TTGiaithua(n As Integer)
        kq = 1
        For i = 1 To n
           kq = kq * i
        Next
End Sub
```

- Gọi thủ tục

```
Private Sub cmdtinh_Click()
        Dim a As Integer
        a = CInt(txtso.Text)
        TTGiaithua(a)
        txtketqua.Text = Str(kq)
End Sub
```

- Hàm tính N!

```
Dim i, kq As Integer
Function HamGiaithua(n As Integer) As
Integer
        kq = 1
        For i = 1 To n
           kq = kq * i
        Next
        HamGiaithua = kq
End Function
```

- Gọi hàm

```
Private Sub cmdtinh_Click()
        Dim a As Integer
        a = CInt(txtso.Text)
        txtketqua.Text =
        Str(HamGiaithua(a))
End Sub
```

5 Kỹ thuật xử lý lỗi

5.1. Xử lý lỗi

Không một chương tình nào là không có lỗi. Tuy nhiên, giảm khả năng lỗi đến mức tối thiểu là có thể làm được

5.1.1.Một số giải pháp giảm lỗi

Thiết kế cẩn thận, đặt tên đối tượng theo quy tắc gợi nhớ.

Ghi ra các vấn đề quan trọng và cách giải quyết cho từng phần.

Thực hiện đúng các bước lập trình.

Ghi ra từng thủ tục và mục đích của nó.

Chú thích rõ ràng trong chương trình

Chạy thử chương trình từng bước

Một trong những nguyên nhân gây lỗi là gõ sai tên biến hoặc nhầm lẫn điều khiển. Dùng "Option Explicit" để tránh trường hợp này.

5.1.2. Đối tượng Err

Là đối tượng do Visual basic cung cấp sẵn để thông báo các

lỗi gặp trong chương trình. Đối tượng này có một số thuộc tính cơ bản sau:

Thuộc tính	Giải thích
Number	Số hiệu lỗi
Description	Chú thích tóm tắt về lỗi
Source	Tên đối tượng gây ra lỗi

5.1.3. Một số kỹ thuật xử lý lỗi

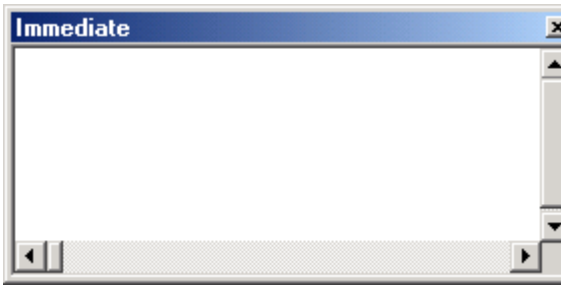
a. Dừng chương trình

Có thể tạm dừng chương trình bằng cách chọn Break từ menu Run hoặc nhấn trên thanh công cụ.

Hoặc nhấn trên tổ hợp phím Ctrl-Break.

Hoặc có thể đặt dòng lệnh Stop trong chương trình.

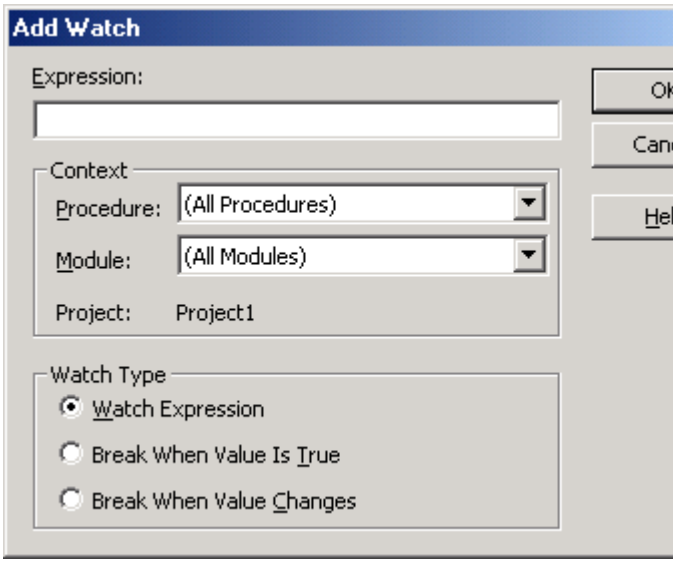
b. Cửa sổ Immediate



Cửa sổ này cho phép ta xem các giá trị của các biến trong form khi chạy gỡ lỗi. Từ đó phát hiện ra các đối tượng gây lỗi.

c. Cửa sổ Watch

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



d. Đi qua từng dòng chương trình

Sử dụng thanh Debug

Thứ tự nút bấm từ trái sang phải như sau:

Start: thi hành chương trình

Break: tạm dừng chương trình

End: Kết thúc chương trình

BreakPoint: Điểm đánh dấu dòng lệnh để tạm dừng chương trình. *(Nút này được sử dụng để bật tắt chế độ*

breakpoint. Khi có lỗi xảy ra và ta chưa khoanh được khu vực nghi ngờ, thì Breakpoint là giải pháp tốt nhất để cô lập vùng chương trình bị lỗi)

Step Into: Nếu dòng lệnh hiện hành đang gọi một thủ tục, nhấn F8 sẽ nhảy vào bên trong thủ tục.

Step Over: Nếu dòng lệnh hiện hành đang gọi một thủ tục, nhấn Shipt-F8 sẽ chạy qua thủ tục.

Step Out: Nếu điểm dừng đang ở trong một thủ tục, nhấn Ctrl-Shift-F8 sẽ chạy hết thủ tục và dừng ở dòng kế tiếp sau lệnh gọi thủ tục

5.2. Bẫy lỗi

Lệnh On Error

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

On Error GoTo <Nhãn>: sử dụng thuật bẫy lỗi để có thể kiểm soát được lỗi. <Nhãn> là tên thủ tục xử lý lỗi được gọi đến.

On Error Goto 0: tắt xử lý lỗi.

On Error Resume next: bỏ qua lỗi, trả chương trình về dòng lệnh ngay sau dòng lệnh sinh lỗi.

Bài 6: LẬP TRÌNH CƠ SỞ DỮ LIỆU

Thời gian: 14 giờ

1. Đối tượng dữ liệu ADO (ActiveX Data Object)

1.1. Thư viện ADODB

ADO là công nghệ truy cập cơ sở dữ liệu hướng đối tượng (***ActivateX Data Object***). Hiện nay Microsoft xem ADO là kỹ thuật cơ bản để truy cập cơ sở dữ liệu bởi vì nó được cung cấp dưới dạng thư viện ActivateX Server có thể dùng thuận tiện trong ứng dụng Visual Basic. ADO được cài đặt như một phần của Visual Basic 6.0.

Để có thể lập trình với thư viện ADO, phải thiết lập tham chiếu đến thư viện này trong ứng dụng Visual Basic như sau:

Trong cửa sổ VB:

→ chọn **Project →References** → xuất hiện hộp thoại **References**

→ Tích chọn: ☑ **Microsoft ActivateX Data Objects 2.8 Library**

→ **OK**

1.1.1. Tập hợp đối tượng Connections

a. Chức năng

Thiết lập kết nối đến nguồn dữ liệu (kết nối CSDL thật sự)

b. Khai báo:

Cách1:

[Global | Public] Dim <tên biến> **As New ADODB.Connection**

Cách 2:

[Global | Public] Dim <tên biến> **As ADODB.Connection**

Set <tên biến> **= New ADODB.Connection**

c. Thuộc tính:

Connection String: chuỗi kết nối

= "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=<đường dẫn>/<tên CSDL.mdb>" → *Kết nối đến cơ sở dữ liệu Access 2003*

= "Provider=Microsoft.ACE.OLEDB.12.0; Data Source=<đường dẫn>/<tên CSDL.accdb>" → *Kết nối*

đến cơ sở dữ liệu Access 2007:

Cursor Location: xác định vị trí con trỏ *(hỗ trợ duyệt qua từng bản ghi tại một thời điểm và cho phép điều khiển cách quản lý của một Recordset*)

= adUserClient*: con trỏ phía Client*

= adUserServer*: con trỏ phía Server*

State: trạng thái kết nối *(trong trường hợp dùng đi dùng lại kết nối nhiều lần cần kiểm tra trạng thái kết nối vì sẽ không thể thực hiện mở được kết nối đang mở và cũng không thể thực hiện đóng được kết nối đang đóng)*

= adStateOpen*: trạng thái mở*

= adStateClosed*: trạng thái đóng*

d. Phương thức:

Open <Connection String>: mở kết nối với các thống số được chỉ định bởi chuỗi Connection String.

Excute <Command Text>: thực thi một truy vấn SQL với CSDL đang được mở bởi Connection. <*Command Text*> *là lệnh truy vấn được thực thi.*

Close: đóng kết nối

1.1.2.Tập hợp đối tượng Recordsets

a. Chức năng

Là tập bản ghi được sử dụng để thao tác truy cập thông tin

b. Khai báo

Cách 1:

[**Global | Public**] **Dim** <tên biến> **As New ADODB.RecordSet**

Cách 2:

[**Global | Public**] **Dim** <tên biến> **As ADODB.RecordSet**

Set <tên biến> = **New ADODB.RecordSet**

c. Thuộc tính

CursorType: xác định loại con trỏ được trả về từ cơ sở dữ liệu.

LockType: xác định cách thức khóa mẩu tin trong Recordset.

Source: xác định câu lệnh truy vấn để lấy dữ liệu

*(ví dụ: Rs.Source = "SELECT * FROM HangHoa")*

ActiveConnection: chỉ định đối tượng Connection mà qua đó Recordset sẽ lấy dữ liệu. *(ví dụ: Set rs.ActiveConnection = cnn)*

RecordCount: xác định tổng số bản ghi *(đối với con trỏ phía Client)*

BOF (= True hoặc False): xác định vị trí đầu tập bản ghi

EOF(= True hoặc False): xác định vị trí cuối tập bản ghi *(khi vị trí đầu tập bản ghi trùng vị trí cuối tập bản ghi tức là tập bản ghi đó rỗng)*

Fields(<tên trường>): xác định trường cần thao tác

d. Phương thức

Open: mở tập bản ghi

Close: đóng tập bản ghi

AddNew: thêm bản ghi mới

Update: cập nhật bản ghi vào CSDL

Delete: xoá bản ghi

ReQuery: thực hiện lại truy vấn

Move First: về bản ghi đầu tiên

Move Previous: về bản ghi trước đó

Move Next: đến bản ghi tiếp theo

Move Last: đến bản ghi cuối cùng

Move <N>: đến bản ghi thứ <N>

- Chú ý:

Nếu xác lập các thuộc tính cho đối tượng Connection và RecordSet thì phải xác lập thuộc tính trước khi thực thi các phương thức.

1.1.3. Tập hợp đối tượng Commands

a. Chức năng

Là tập đối tượng được sử dụng khi muốn thi hành các thủ tục lưu trữ sẵn hay những câu truy vấn có tham số.

b. Khai báo

Cách 1:

[**Global | Public**] **Dim** <tên biến> **As New ADODB.Command**

Cách 2:

[**Global | Public**] **Dim** <tên biến> **As ADODB. Command**

Set <tên biến> = **New ADODB. Command**

c. Thuộc tính

ActiveConnection: tên của đối tượng Connection chứa đối tượng Command

CommandText: chuỗi chứa văn bản lệnh cần thực thi.

CommandTimeout: số giây ADO cần đợi để lấy kết quả của việc thực thi đối tượng Command trước khi gây ra lỗi *(mặc định 30 giây)*.

CommandType: loại lệnh được thực thi bằng đối tượng Command.

Name: chuỗi biểu thị tên đối tượng Command.

State: xác định đối tượng Command được mở hay đóng.

d. Các tập hợp

Parameters: chứa các đối tượng Parameter

Properties: thuộc tính của đối tượng Command.

e. Các phương thức

CreateParameter: tạo đối tượng Parameter mới cho tập hợp Parameters của đối tượng Command.

Execute: thực thi lệnh được chứa trong thuộc tính CommandText.

1.2 Truy cập cơ sở dữ liệu với ADODB

1.2.1. Mở và đóng kết nối CSDL

Để đưa ra các yêu cầu đến nguồn dữ liệu sử dụng ADO, cần mở kết nối đến nguồn dữ liệu đó bằng phương thức Open của đối tượng Connection.

a. Mở kết nối CSDL

CP1: <Biến kết nối>**.Open** <Chuỗi kết nối>

Cp2: <Biến kết nối>**.Connection String =** <Chuỗi kết nối>

<Biến kết nối>**.Open**

Khi đã hoàn thành tất cả các thao tác liên quan đến nối kết này, cần phải đóng và huỷ nối kết một cách tường minh thông qua phương thức Close của đối tượng Connection. Đóng kết nối một cách tường minh sẽ đảm bảo rằng tất cả các tài nguyên liên quan đến nối kết này trên Server cũng như Client đều được giải phóng một cách hợp lý.

b. Đóng kết nối CSDL

CP: <Biến kết nối>**.Close** '→Đóng kết nối

Set <Biến kết nối> **= Nothing**

'→Huỷ kết nối

***Chú ý**

Mở và đóng kết nối CSDL nên viết thành thủ tục trong Module để sử dụng chung cho cả chương trình. Biến kết nối, chuỗi kết nối nên khai báo toàn cục (Global).

Ví dụ:

Viết thủ tục Mở kết nối CSDL và Đóng, huỷ kết nối CSDL trong Module

```
Option Explicit
Global StrCnn As String
Global Cnn As New ADODB.Connection

Sub MoKetnoi()
    StrCnn = "Provider=Microsoft.ACE.OLEDB.12.0; " _
        & " Data Source=" & App.Path & "\QLHang.accdb"
    If Cnn.State = adStateOpen Then Cnn.Close

    Cnn.CursorLocation = adUseClient
    Cnn.Open StrCnn
End Sub

Sub DongHuyKetnoi()
    Cnn.Close
    Set Cnn = Nothing
End Sub
```

1.2.2. Thao tác với CSDL

a. Thực thi các câu truy vấn

Các câu truy vấn hành động (Insert, Update, Delete) được thực hiện nhờ phương thức Execute của đối tượng Connection; ngoài ra phương thức này cũng có thể được sử dụng để thực thi các thủ tục lưu trữ sẵn trong cơ sở dữ liệu hay các câu SELECT.

- Cú pháp:

CP1: Không có giá trị trả về

<Biến kết nối>.**Execute** <câu lệnh truy vấn hành động>

CP2: Có giá trị trả về

Set<Biến tập bản ghi>=<Biến kết nối>.**Execute** (<lệnh lấy dữ liệu>)

Trong đó:

<***Biến kết nối***>: là đối tượng kết nối CSDL Connection.

<***Biến tập bản ghi***>: là đối tượng Recordset lưu kết quả trả về của phương thức Execute

<***câu lệnh truy vấn hành động***>: thường là các lệnh Insert, Update, Delete

<***câu lệnh lấy dữ liệu***>: thường là lệnh Select.

b. Ngắt kết nối RecordSet

Để ngắt kết nối với Server trong ADO, ta quy định thuộc tính ActiveConnection của đối tượng Recordset là Nothing. Client sẽ tiếp tục làm việc với dữ liệu thậm chí khi nó không kết nối với server.

- Cú pháp:

<Biến RecordSet>.**Close** '→Ngắt kết nối RecordSet

Set <Biến RecordSet> = **Nothing** '→Huỷ kết nối RecordSet

2 Bài toán đặt lọc dữ liệu

2.1. Câu lệnh SELECT

a. Chức năng

Đưa ra các thông tin từ cơ sở dữ liệu theo các trường hoặc theo điều kiện tìm kiếm hay thống kê

b. Cú pháp:

Select <DS trường> **From**<Bảng> [**Where** <Điều kiện lọc>][**Order by** <DS trường>]

Trong đó:

Select < *Ds trường*>: là danh sách các trường được đưa ra, phân cách nhau bởi dấu ",". *Nếu lấy tất cả các trường trong bảng CSDL thì dùng ký tự "*" thay thế. Muốn hiển thị n số bản ghi đầu:* ***Top <n>****, hiển thị n số bản ghi cuối:* ***Top <n> percent*** *Khi muốn đưa ra trường kết xuất từ những trường đã có thì thực hiện theo cú pháp: <biểu thức>* ***As*** *<tên trường kết xuất>*

From <*Bảng*>: chỉ ra tên bảng CSDL cần truy vấn thông tin.

Where<*Điều kiện lọc*>: tiêu chuẩn cần lọc thông tin theo một điều kiện nào đó.

Order by <*DS trường*>: Sắp xếp thông tin hiển thị, thứ tự ưu tiên theo danh sách trường. *Sắp xếp tăng dần: **ASC** , sắp xếp giảm dần: **DESC**.*

Ví dụ:

1. *Lọc đưa ra dữ liệu trong bảng Hàng Hoá với điều kiện số lượng hàng >100 và sắp xếp tăng dần theo đơn giá.*

Select * ***From*** HangHoa ***Where*** Soluong>100 ***Orderby*** Dongia ***ASC***

2. *Lọc đưa ra dữ liệu gồm: Mahang, Tenhang, Soluong, Dongia, Thanhtien trong bảng Hàng Hóa với tên hàng là Sữa*

Select *Mahang, Tenhang, Soluong, Dongia, Soluong*Dongia* ***As*** *Thanhtien* ***From*** HangHoa ***Where*** *Tenhang='Sữa'*

2.2. Hiển thị dữ liệu trên DataGrid

a. Lấy điều khiểnDataGrid

Trong cửa sổ chương trình Visual Basic

Vào Project → Components (Ctr+T) → xuất hiện hộp Componets

Tích chọn: ☑ **Microsoft DataGrid Controls 6.0 (OLEDB)**

b. Các thuộc tính cơ bản

Name: tên của điều khiển

DataSource: nguồn dữ liệu hiển thị

AllowAddNew (***T/F***): cho phép hoặc không cho phép thêm bản ghimới

AllArrows (***T/F***): có hoặc không có con trỏ di chuyển giữa các bản ghi

AllowDelete (***T/F***): cho phép hoặc không cho phép xoá bản ghi

AllowUpdate (***T/F***): cho phép hoặc không cho phép sửa bản ghi

Caption: tiêu đề của điều khiển

c. Gán dữ liệu vào DataGrid

Set <tên DataGrid>.**DataSource** = <tên RecordSet>

Ví dụ: Đưa ra danh sách hàng hóa có số lượng <=20.

```
Private Sub Form_Load()
Dim Rs As New ADODB.Recordset
Dim sqlSelect As String
    sqlSelect = "Select * From HangHoa Where soluong<=20"
```

```
        Cnn.CursorLocation = adUseClient
    MoKetnoi                'Gọi thủ tục Mở kết nối
        Set Rs = Cnn.Execute(sqlSelect)
            If Rs.EOF And Rs.BOF Then
                MsgBox "Khong co ban ghi", vbInformation, "QL Hang Hoa"
            Else
                Set DGHang.DataSource = Rs
            End If
    End Sub
```

Bài 7: MENU VÀ TOOLBAR

Thời gian 12 giờ

1. Tạo Menu

1.1. Khái niệm

Menu là một loại điều khiển mà qua đó người sử dụng có thể lựa chọn các mục từ một danh sách cho trước. *(Giao diện của menu phụ thuộc vào Windows còn lập trình chỉ quản lý phần xử lý các sự kiện mà thôi).*

Có 2 loại menu thường gặp sau:

Drop-Down Menu (menu thả xuống): là menu chính của một chương trình nằm phía dưới thanh tiêu đề.

Pop-Up Menu (menu hiện ra): thường hiển thị ở vị trí mà được ấn nút phải chuột. *(Đây là dạng menu shortcut chỉ tồn tại khi có menu chính).*

1.2. Các thuộc tính của Menu

Thuộc tính của menu không thuộc cửa sổ Properties như các điều khiển khác mà đặt trong trình soạn thảo menu (Menu Editor).

Khi tạo menu cần chú ý thiết lập những thuộc tính sau:

Caption: chuỗi ký tự hiển thị trên menu. *(Tên đề mục của menu, dùng dấu & phía trước từ được chọn làm shortcut của menu)*

Name: tên của menu theo quy tắc đặt tên. *(Mỗi mục có một tên riêng, nhưng nên bắt đầu bằng mnu+chức năng của menu)*

Shortcut: chọn phím tắt.

Checked: hiển thị dấu tích (✓) bên cạnh. (thuộc tính này không được gán

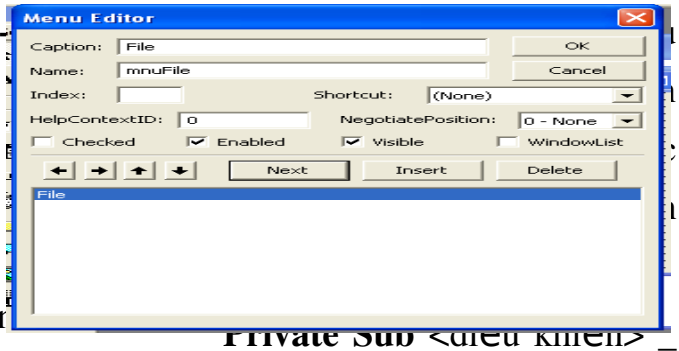
cho những mục menu đang chứa menu con)

Enable: làm mờ menu. *(chức năng của menu không được thực hiện)*

Visible: hiển thị menu

1.3. Tạo Drop-Down Menu

Mở cửa sổ soạn thảo menu: **Tool → Menu Editor** (Ctrl +E) → xuất hiện cửa sổ



- Thiết lập các thuộc tính cho từng đề mục của menu
- Dùng các phím mũi tên (←,→,↑,↓) để di chuyển vị trí đề mục menu lên xuống, vào ra theo từng mức
- **Next**: thực hiện tạo đề mục menu mới sau khi đã xong một đề mục
- **Insert**: chèn thêm đề mục
- **Delete**: xoá đi một đề mục
- Tạo đường gạch ngang giữa các đề mục menu: thuộc tính Caption = '-', thuộc tính Name đặt tên bất kì.
- **OK**: hoàn thành và kết thúc việc tạo menu.

1.4. Tạo Pop-Up Menu

Pop-Up menu thực ra là một dạng Shortcut linh động của Drop-Down Menu, và chỉ thực hiện được cho những đề mục menu có menu con.

Private Sub <điều khiển>_ <sự kiện> **(Button As Integer, Shift As Integer, X As Single, Y As Single)**

If Button = <nút bấm chuột> **Then**

PopupMenu <tên menu>, <vị trí hiển thị>

End If

End Sub

Trong đó:

<nút bấm chuột>= **vbRightButton**: bấm nút chuột phải

= vbLeftButton: bấm nút chuột trái

***<vị trí hiển thị>*=**

vbPopupMenuLeftAlign: hiển thị phía trái vị trí mũi tên trỏ chuột

=

vbPopupMenuRightAlign: hiển thị phía phải vị trí mũi tên

=

vbPopupMenuCenterAlign: hiển thị giữa vị trí mũi tên trỏ chuột

<tên menu>: là tên của Drop-Down Menu đã được đặt trong trình soạn thảo menu

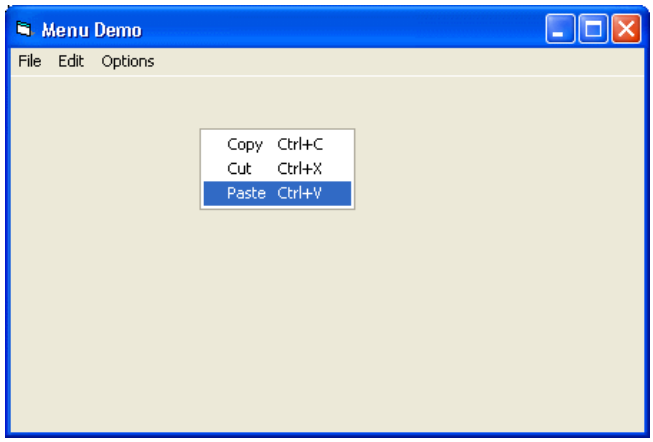
***Ví dụ**

- Tạo Drop-Down menu

- Tạo Pop-Up menu

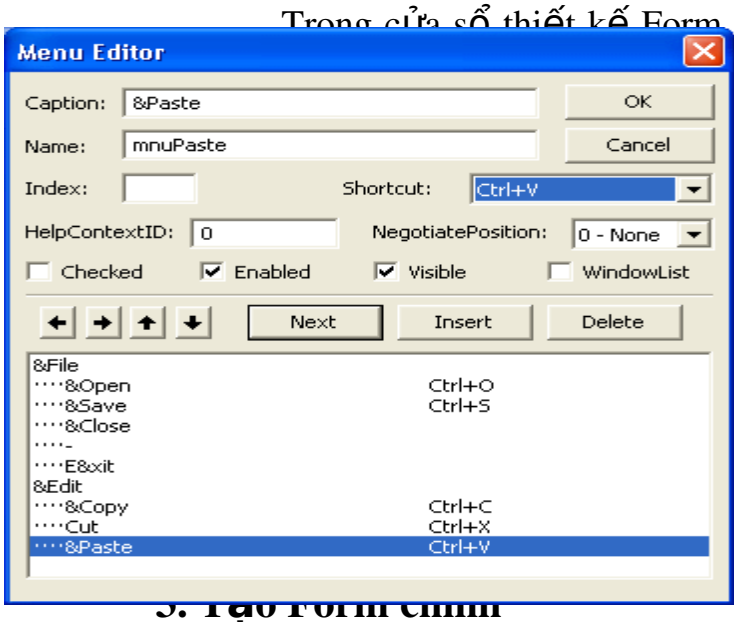
```
Private Sub Form_MouseDown(Button As Integer, …)
    If Button = vbRightButton Then
        PopupMenu mnuEdit
    End If
End Sub
```

- Chạy thử chương trình



2. Gắn kết Menu

Trong cửa sổ thiết kế Form



3. Tạo Form chính

3.1. Giới thiệu

Form chính là giao diện hiển thị lên đầu tiên khi khởi động một chương trình quản lý. Form chính thường ghi các thông tin sau:

- Tên chương trình

- Tên cơ quan quản lý, sử dụng chương trình
- Tên công ty, nhóm người lập trình
- Thông tin hướng dẫn sử dụng
- …

Tuy nhiên chức năng chính của Form chính là chứa hệ thống menu liên kết đến các Form thành phần và menu đăng nhập hệ thống.

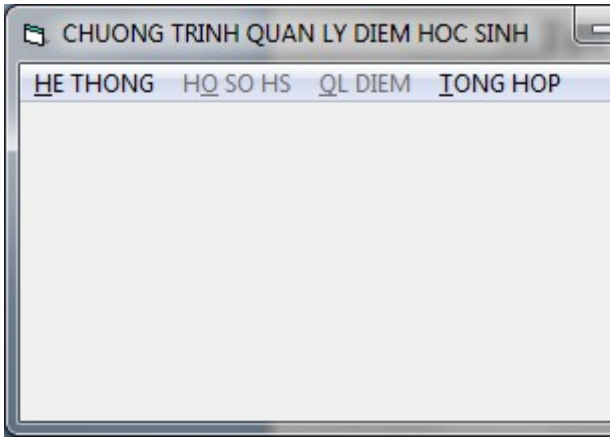
3.2. Các bước thực hiện

- Tạo Form

- Tạo các Lable (nhãn) ghi các thông tin cần thiết

- Tạo hệ thống menu liên kết đến các Form thành phần

- Tạo hệ thống đăng nhập



3.3. Tạo Form đăng nhập

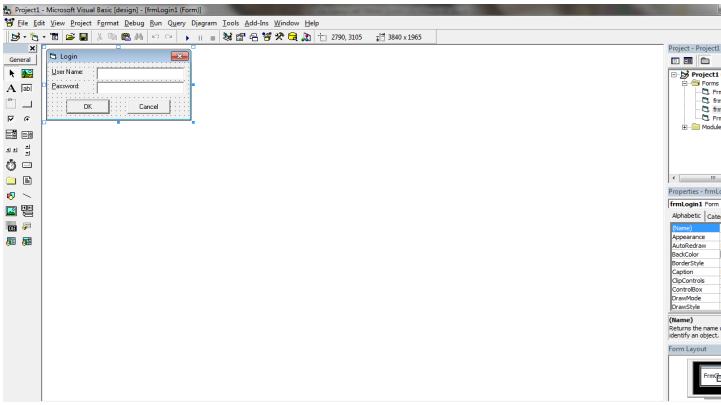
a. Tác dụng

- Bảo mật thông tin: chỉ cho phép những người có quyền mới được phép truy nhập hệ thống.

- Phân quyền: giới hạn người được phép nhập, sửa, xóa thông tin hệ thông tin hệ thống.

b. Các bước thực hiện

Trong cửa sổ Microsoft Visual Basic, chọn lệnh **Project → Add Form →** xuất hiện hộp thoại → chọn **Login Dialog → Open →** xuất hiện giao diện Form Login như sau:



Thực hiện lập trình mã lệnh cho nút lệnh OK.