



# **Giáo trình Microsoft Access 2003**

# GIỚI THIỆU VỀ MICROSOFT ACCESS

- Giới thiệu về Access 2003
- Khái niệm về cơ sở dữ liệu
- Xác lập môi trường làm việc

## 1.1 GIỚI THIỆU MS - ACCESS 2003

### 1.1.1 Nguồn gốc và công dụng

™ Microsoft Access 2003 là chương trình của bộ ứng dụng văn phòng Microsoft Office 2003 chạy trên môi trường Windows. Đây là phần mềm thuộc hệ Quản Trị Cơ Sở Dữ Liệu Quan hệ (Relational Database Management System – RDBMS) giúp quản lý, bảo trì và khai thác dữ liệu lưu trữ trên máy tính.

™ Một cơ sở dữ liệu kiểu quan hệ không cần phải lưu trữ các dữ liệu có liên quan với nhau nhiều lần trong các bảng dữ liệu khác nhau.

### 1.1.2 Khởi động và thoát khỏi ACCESS

™ Khởi động Access:  
Dùng Start Menu/ShortCut hay dùng lệnh RUN (tập khởi động của Access là MSACCESS.EXE).

™ Kết thúc làm việc với Access  
Chọn menu File – Exit hay click nút Close hay dùng phím tắt ALT+F4

### 1.1.3 TaskPane

™ Mặc định, khi khởi động Access 2003, TaskPane hiển thị ở cạnh phải màn hình.

™ Tắt mở TaskPane : Ctrl+F1 hoặc chọn menu View – TaskPane



## 1.2 KHÁI NIỆM VỀ TẬP TIN CƠ SỞ DỮ LIỆU

### 1.2.1 Tập tin Cơ sở Dữ liệu

ACCESS làm việc với tập tin CSDL (Datebase) có phần mở rộng là .mdb và có thể khái niệm như tập hợp các thông tin lên quan đến một chủ thể làm việc và được ghi lưu trong một tập tin theo định dạng của ACCESS.

### 1.2.2 Mở tập tin Cơ sở Dữ liệu

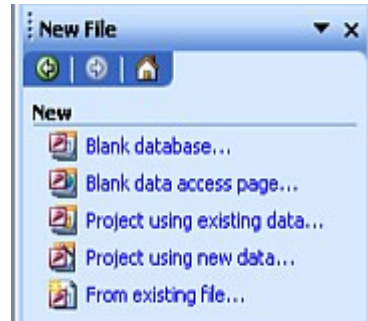
™ Mở tập tin CSDL từ TaskPane

*f* Tạo tập tin từ CSDL mới

o Click mục Create a new file để mở New File TaskPane

o Chọn cách tạo tập tin CSDL mới

- Blank Database: tập CSDL rỗng
- Blank Data access page : Trang dữ liệu Access rỗng
- From existing file : Tạo tập CSDL mới bằng cách sao chép tập CSDL đã có trên Disk
- Project ... : Để án Access (loại File mới của Access .adp dùng để làm việc với SQL Server Database)



*f* Mở tập tin CSDL đã có sẵn (Open an Existing Database):

o Chọn tên tập CSDL cần mở trên phần OPEN (nếu có hiển thị)

o Hay click More... : mở hộp thoại Open và chọn tập CSDL cần mở

™ Mở tập tin CSDL từ Menubar

Chọn Menu FILE - NEW / OPEN (phím tắt : Ctrl+N / Ctrl+O) và tiếp tục các bước tương tự như nêu trên.

### GHI CHÚ

Các tập tin CSDL được thiết kế với ACCESS 2000 trở lên mới có thể mở với ACCESS 2003. Nếu mở tập CSDL thiết kế với ACCESS 97 trong ACCESS 2003, cần phải convert sang của ACCESS 2000 hay ACCESS 2002-2003.

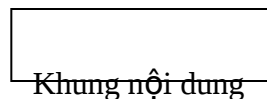
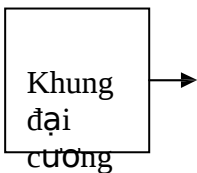
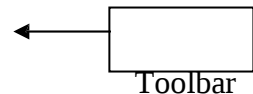
### 1.2.3 Các đối tượng của CSDL

ACCESS làm việc với tập tin CSDL (.MDB) và gồm 7  
loại  
đối tượng

- TABLES: (Bảng Dữ Liệu) Loại đối tượng cơ bản và quan trọng nhất của CSDL dùng để thiết kế các dữ liệu cơ sở, ghi các biến động cần quản lý, khai thác.
- QUERIES: (Bảng Truy Vấn) Công cụ truy vấn thông tin và thực hiện các thao tác trên dữ liệu của TABLE . Query được dùng làm nền tảng để làm các báo biểu có đặc tính định dạng cao hơn Table.
- FORMS: (Biểu Mẫu) Dùng để thiết kế màn hình nhập liệu một cách sinh động hoặc điều khiển hoạt động của chương trình ứng dụng.
- REPORTS: (Báo Biểu) Là kết quả đầu ra của quá trình khai thác dữ liệu có nguồn gốc từ các Table hay Query.
- PAGES: (Trang) các trang dữ liệu Access thiết kế theo dạng Web.
- MACROS: (Tập Lệnh) Công cụ của Access giúp tạo các hành động đơn giản khi xây dựng ứng dụng mà không cần dùng ngôn ngữ lập trình.
- MODULES: (Đơn Thể) Dùng viết các dòng lệnh cho ứng dụng theo ngôn ngữ Visual Basic. Đây là công cụ lập trình chuyên nghiệp của Access.

#### 1.2.4 Cửa sổ DATABASE

- Khi một CSDL được mở, cửa sổ Database của CSDL (còn gọi là Database Container hay Database Window) được hiển thị và gồm



- Thanh tiêu đề  
Gồm bên trái là <Tên CSDL> DATABASE (Kiểu định dạng) và bên phải là các nút điều khiển cửa sổ (phóng to, thu nhỏ, đóng cửa sổ).
- Thanh dụng cụ  
Là thanh dụng cụ của Cửa sổ Database : Hiệu lực với các mục chọn trong cửa sổ này và thường gồm Open (mở trong chế độ làm việc), Design (Mở trong chế độ thiết kế), New (mở mới), Close (đóng cửa sổ Database), các Icon chuyển đổi chế độ hiển thị trong khung nội dung.
- Khung Đại cương gồm nút Objects, Groups và biểu tượng Favorites
  - *Objects*: Dùng để mở hay thu gọn danh mục các loại đối tượng của Database
  - *Groups* : Dùng mở hay thu gọn danh mục làm việc với nhóm
  - *Favorites*: Mở danh mục các đối tượng trong thư mục Favorites.
- Khung nội dung
  - Phía trên là các shortcut để tạo mới đối tượng thuộc loại đối tượng được chọn trong khung đại cương.
  - Phía dưới là danh mục các đối tượng thuộc loại được chọn bên khung Đại cương.
- Đóng cửa sổ CSDL là đóng tập CSDL đang mở.

### 1.3 XÁC LẬP MÔI TRƯỜNG LÀM VIỆC

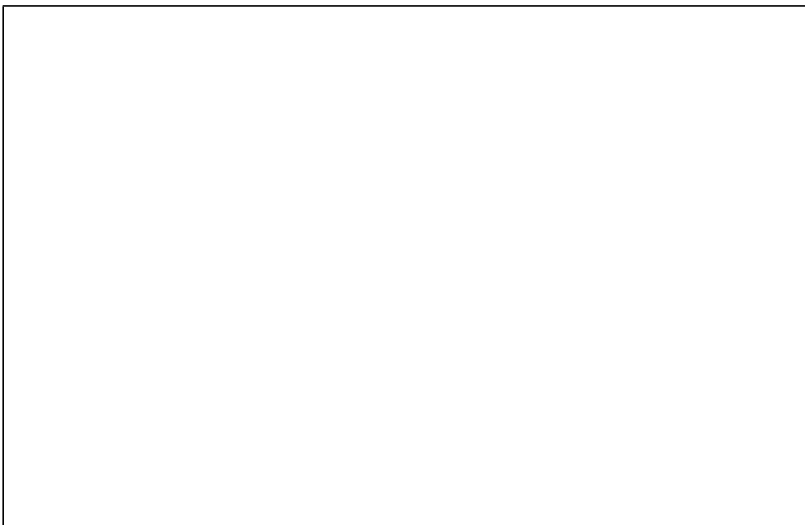
#### 1.3.1 Xác lập môi trường hệ thống

- Xác lập môi trường hệ thống để định dạngthức hiển về ngày tháng, dữ liệu kiểu số (dùng chung trong môi trường Windows).
- Mở hộp thoại REGIONAL AND LANGUAGE OPTIONS
- Xác lập khu vực (Regional Options)
  - Mặc định khu vực được xác lập là English (United States).  
Nếu chọn mục FRENCH (Standard) thì các xác lập về hiển thị số và ngày đã phù hợp với dạng thức thường dùng tại Việt Nam.
  - Nếu chọn xác lập khu vực là English (United States) và muốn hiển thị dạng thức dữ liệu số và ngày

theo quy ước được dùng tại nước ta thì phải xác lập lại phần định dạng số, định dạng tiền tệ và định dạng ngày.

### 1.3.2 Xác lập môi trường ACCESS

- Mở hộp thoại xác lập : chọn Menubar TOOLS - OPTIONS - thực hiện xác lập.
- Các Phiếu của hộp thoại gồm
  - View: Hiển thị thanh tình trạng, hộp thoại khởi động, ...v.v.
  - General: Định lề trang, thư mục làm việc mặc định .
  - Edit/Find: Chọn cách tìm kiếm, xác định khi thay đổi Record – Action queries hay xóa đối tượng CSDL.
  - Keyboard: Xử lý di chuyển cursor khi dùng phím ENTER, mũi tên.
  - DataSheet: Định dạng cho DATASHEET (Font, hình dạng, ...v.v.)
  - Forms/Reports: Sử dụng khuôn dạng, ...v.v.
  - Advanced: Các thiết lập liên quan đến nhiều người dùng.
  - Tables/Query: Các thiết lập liên quan Bảng, Queries Default size, Type, ...v.v.)
- Các xác lập thường dùng
  - Phiếu General
    - Print Margin: Xác lập lề trang in
    - Use fuor-digit year formatting: Xác lập định dạng 4 ký số năm
    - Compact on close: Nén khi đóng CSDL
    - Default Database Folder: Thư mục mặc định của CSDL
  - Phiếu DataSheet



- Default Font: Xác lập Font mặc định.
- Default color: Màu mặc định cho ký tự, nền đường lưới.



- Default Gridline Showing: Xác lập mặc định hiển thị đường lưới.
- Default Column Width: Xác lập chiều rộng mặc định của cột
- Default Cell Effect: Xác lập dạng hiển thị Ô

□ Phiếu Find / Edit



- Default Find / Replace: Xác lập việc tìm và thay
- Confirm: Xác lập yêu cầu xác nhận khi xóa CSDL, thực hiện văn tin hành động, thay đổi mẫu tin.

## **BÀI TẬP:**

1. Khởi động Access và quan sát màn hình làm việc của Access. Tìm hiểu các chức năng của hệ thống menu trong Access.
2. Tạo một tập tin CSDL trong Access, đặt tên cho tập tin này là  
QLHS.MDB
3. Đóng tập tin QLHS.MDB lại. Đổi tên tập tin này thành  
QLSV.MDB
4. Tạo folder BT\_Access trên ổ D. Tạo tập tin CSDL mới đặt tên  
THUVIEN.MDB để trong folder BT\_Access.
5. Thiết lập môi trường làm việc: Font chữ tiếng Việt, bảng mã Unicode, kiểu gõ VNI (hoặc Telex). Qui định cách nhập dữ liệu thời gian dạng ngày/tháng/năm (hiện 4 số).
6. Thoát khỏi Access. Chép tập tin QLSV.MDB vào folder BT\_Access.
7. Khởi động Access, mở lại tập tin QLSV.MDB và xem kích thước của tập tin.

## CHƯƠNG 2

# TẠO CSDL VÀ BẢNG

- Những giai đoạn thiết kế ứng dụng
- Tạo tập tin CSDL
- Khái niệm về Bảng
- Thiết kế cấu trúc Bảng
- Nhập liệu vào Bảng
- Các thuộc tính thường dùng
- Thiết lập Lookup

### 2.1 NHỮNG GIAI ĐOẠN THIẾT KẾ ỨNG DỤNG CSDL

Khi thiết kế một ứng dụng CSDL (gồm CSDL và các thao tác khai thác), cần tiến hành các bước sau:

#### 2.1.1 Xác định mục đích của CSDL

CSDL dùng để làm gì: xác định những chủ thể dữ liệu cần thiết (Bảng dữ liệu – Table) và dữ liệu cần lưu trữ (các Field của mỗi Bảng).

#### 2.1.2 Xác định các bảng dữ liệu cần thiết

Xem xét các thông tin muốn lấy từ CSDL và phân chia thông tin đó thành những chủ thể cơ bản. Mỗi chủ thể được tổ chức trong Table riêng.

#### 2.1.3 Tạo lập các Table

™ Xác định vùng dữ liệu (Field) của mỗi Bảng.

™ Mỗi vùng phải có quan hệ trực tiếp với chủ thể dữ kiện. Nếu mỗi vùng nào đó của Record lại mô tả một chủ thể dữ liệu của Table khác thì vùng này cũng được định nghĩa trong Table này để làm cơ sở thiết lập mối quan hệ giữa các Table sau này.

™ Không ghi những dữ liệu phải tính toán vào Table.

™ Lưu trữ các dữ kiện theo các thành phần luận lý nhỏ nhất.

#### 2.1.4 Thiết lập mối quan hệ giữa các Bảng

Xem xét sự liên quan dữ liệu giữa các Bảng và thiết lập quan hệ trên cơ sở các Field làm tiêu chuẩn quan hệ. Field làm tiêu chuẩn quan hệ giữa các Bảng gọi là mục Khóa (Key).

#### 2.1.5 Thiết kế các công cụ khai thác dữ liệu

Thiết kế màn hình xuất nhập dữ liệu dùng để nhập liệu, hiển thị thông tin và kết xuất ra máy in.

### 2.1.6 Ấn định các thao tác xử lý cho người sử dụng

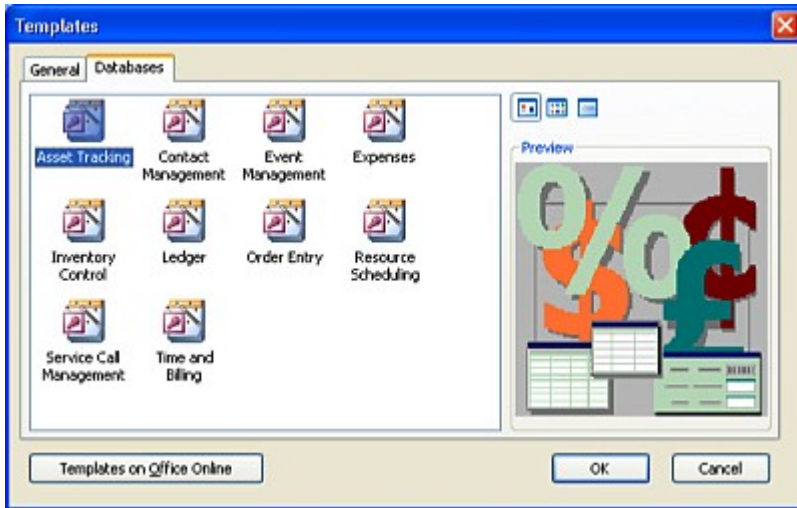
Thiết kế các công cụ điều khiển hoạt động để tạo tiện nghi cho người sử dụng khi thao tác với ứng dụng.

## 2.2 TẠO CƠ SỞ DỮ LIỆU

Khi làm việc với Access, mọi việc chỉ có thể bắt đầu khi có một tập tin đã được tạo và được mở.

Có 2 cách tạo tập tin CSDL: dùng DATABASE WIZARD để tạo Database với các thành phần kèm theo hoặc tạo một Database rỗng để chủ động thiết kế các thành phần theo yêu cầu riêng của người sử dụng.

### 2.1.1 Dùng Database Wizard



- ™ Mở hộp thoại Templates
  - f Trên TaskPane, chọn mục New File hay Ctrl+N
  - f Tại phần **Templates**, click liên kết **On Computer** để mở hộp thoại Templates. Nếu muốn dùng Templates trên mạng thì click mục **Templated on Office online**.
- ™ Trên hộp thoại Templates
  - f Chọn phiếu Databases
  - f Chọn chủ đề rồi click OK
- ™ Thực hiện các bước theo hướng dẫn của Wizard

### 2.1.2 Tạo CSDL rỗng

- ™ Có thể tạo CSDL rỗng để tự thiết kế các đối tượng Tables, Reports, ... v.v theo nhu cầu riêng.
- ™ Quy trình
  - f Trên TaskPane, chọn mục New File hay Ctrl+N
  - f Tại phần **New**, click liên kết **Blank Database**
  - f Nhập tên tập CSDL trong hộp thoại File New Database rồi click nút **Create**.
- ™ CSDL mới được tạo và hiển thị Cửa sổ Database. Muốn thao tác với loại đối tượng nào thì chọn mục loại đối tượng ấy ở khung đại cương (khung trái).

---

GHI CHÚ: Khi tạo CSDL theo cách này, ban đầu không có Bảng dữ liệu nào cả. Do vậy phải tạo Bảng dữ liệu trước rồi mới thao tác với các đối tượng khác.

---

## 2.3 KHÁI NIỆM VỀ BẢNG (TABLE)

### 2.3.1 Vai trò của Bảng

Bảng là đối tượng cơ bản và quan trọng nhất của CSDL trong Microsoft Access, dùng để ghi nhận và khai thác những thông tin muốn quản lý (dữ liệu cơ sở).

### 2.3.2 Cấu trúc của Bảng

#### 2.3.2.1 *Khái quát*

- ™ Một Bảng được tổ chức thành cột (Field) và dòng (Record - tin)
- ™ Mỗi cột ứng với một kiểu dữ liệu cần lưu trữ.
- ™ Mỗi dòng là một tin (Record), là tập hợp các Field và chức năng thông tin liên quan đến đối tượng cụ thể.
- ™ Tạo Bảng: thiết kế cấu trúc trước và sau đó, nhập các tin.
- ™ *Thiết kế cấu trúc cho Bảng là tạo phần Header chứa tên các Field và gán thuộc tính cho mỗi Field.*

#### 2.3.2.2 Tên Field

- ™ Dài tối đa 64 ký tự kể cả và gồm ký tự chữ cái, ký tự số, ký tự trắng, các ký tự đặc biệt để đặt tên cho Field. Không được dùng dấu chấm (.), dấu chấm than (!) và dấu ngoặc vuông([])
- ™ Không thể bắt đầu bằng ký tự trắng

---

™ Tên Field phải duy nhất (không trùng nhau) và không trùng với tên hàm, từ khóa của Access.

™ Access cho phép dùng ký tự trắng và ký tự đặc biệt để đặt tên Field nhưng không khuyến khích sử dụng các ký tự này để tiện lợi khi dùng biểu thức và lập trình.

### 2.3.2.3 Kiểu dữ liệu của Field

#### ™ Chọn kiểu dữ liệu

f Mặc nhiên, Access gán kiểu Text cho Field tạo mới. Hãy xác lập kiểu dữ liệu cho Field bằng cách chọn kiểu dữ liệu trong hộp danh sách đổ xuống ở cột Data Type trong màn hình Design View.

f Khi xác lập kiểu cho Field, cần chỉ định loại dữ liệu (Text, số, ...v.v), kích thước dành cho Field (Field Size), chỉ mục và các xác lập khác (tùy thuộc vào kiểu dữ liệu của Field)

#### ™ Các kiểu dữ liệu

##### f **Text** (văn bản)

Chứa văn bản gồm ký tự chữ cái, số, khoảng trắng và các ký tự khác. Dài tối đa 255 ký tự (mỗi byte một ký tự).

##### f **Memo** (ký ức)

Giống như Field kiểu Text nhưng có chiều dài tối đa đến

65535 ký tự.

##### f **Number** (số)

Chứa các giá trị số. Chỉ chứa các ký số từ 0 đến 9, dấu phân cách thập phân, dấu âm (-). Chiều dài tối đa có thể là

1,2,4,8,12 bytes. Các loại giá trị lưu trong Field này thường dùng là

- *Byte*: GỒM các giá trị nguyên (không có phần thập phân) từ trị 0 đến 255 và chiếm 1 byte.
- *Integer*: GỒM các giá trị nguyên (không có phần thập phân) từ trị -32,768 đến 32,767 và chiếm 2 bytes.
- *Long Integer*: GỒM các giá trị nguyên (không có phần thập phân) từ trị -2,147,843,648 đến 2,147,843,647 và chiếm 4 bytes. Còn gọi là số nguyên dài.
- *Single*: GỒM các giá trị từ -3.4E38 đến 3.4E38 và chiếm 4 bytes. Còn gọi là số tinh xác đơn (tinh xác 7 chữ số thập phân).
- *Double*: GỒM các giá trị từ -1.79E308 đến 1.79E308 và chiếm 8 bytes. Còn gọi là số tinh xác kép (tinh xác 15 chữ số thập phân)

*f* **Currency (tiền tệ)**

Gồm các giá trị từ -922,337,203,685,477.5808 đến 922,337,203,685,477.5807 và chiếm 8 bytes. (Tinh xác đến 15 chữ số phần nguyên và 4 chữ số thập phân)



*f* **Date/Time** (ngày/giờ)

Chứa trị thời gian từ năm 100 đến năm 9999, chiếm 8 bytes.

*f* **AutoNumber** (số điền tự động)

Gồm các giá trị kiểu số và Access tự động gán cho mỗi Record một trị tương ứng (khởi đầu là 1,2,... và tăng dần). Trị đếm này cứ tiếp tục tăng và không trùng lặp ngay cả với các trị của Record đã được xóa. Kích thước 4 bytes.

*f* **Yes/No**

Chứa các trị Logic và chỉ nhận hai giá trị là Yes/No. Chiều dài là 1 bit.

*f* **OLE Object**

Chứa các đối tượng do các ứng dụng khác trong Windows cung cấp hỗ trợ liên kết (Object Linking) và nhúng (Embedding). Dùng kiểu này để lưu hình ảnh, âm thanh bằng tính hay văn bản (tạo bởi trình xử lý khác). Có thể chứa đến 1 GB.

*f* **Hyperlink**

Chứa các siêu liên kết.

*f* **Lookup Wizard**

Đây không phải là thuộc tính đúng nghĩa, chỉ dùng thiết kế một danh sách chọn cho Field chỉ định theo hướng dẫn của Wizard.

### 2.3.3 Các mục khóa (key)

#### 2.3.3.1 Công dụng của Mục khóa

*f* Một hay nhiều Field được chỉ định làm mục khóa để nhận diện các mẫu tin của một Bảng hay dùng để thiết lập mối quan hệ giữa các Table.

*f* Access phân biệt hai loại mục khóa : khóa chính và khóa ngoại.

#### 2.3.3.2 Mục khóa chính (Primary Key)

*f* Là mục khóa thiết lập trên một hay nhiều Field để nhận diện duy nhất các mẫu tin của một Bảng và có thể dùng lập mối quan hệ với các Bảng khác.

*f* Trị của Field mục khóa chính *duy nhất và không là trị*

*NULL*

*f* Nên tạo Field mục khóa chính với chiều dài nhỏ nhất có thể được ví có ảnh hưởng tốc độ truy xuất.

#### 2.3.3.3 Mục khóa ngoại (Foreign Key)

- f Là mục khóa thiết lập trên một hay nhiều Field của Bảng và dùng tham chiếu đến một hay nhiều Field là khóa chính trong Table khác.
- f Một Foreign Key chỉ ra mối quan hệ của Bảng này với các Bảng chứa khóa chính tương ứng.
- f Kiểu dữ liệu trong Field được chỉ định là Foreign Key và Primary Key phải cùng kiểu và chiều dài với nhau (không nhất thiết cùng tên).

### 2.3.4 Chế độ làm việc với Bảng

Có hai chế độ làm việc với Bảng

- f Với **Datasheet view** (chế độ bảng biểu): nhập, xóa, hiệu chỉnh Record.
- f Với **Datasheet Design** (chế độ thiết kế): thiết lập hay hiệu chỉnh cấu trúc Bảng.

## 2.4 THIẾT KẾ CẤU TRÚC BẢNG

### 2.4.1 Các cách thực hiện

- ™ Từ cửa sổ Database - chọn mục Tables trong khung đại cương và thực hiện một trong các thao tác sau.
- ™ Click mục New trên Toolbar của cửa sổ Database, hộp thoại New Tables được hiển thị - Chọn một trong các mục sau.
  - f Database View: Tạo Bảng theo chế độ Datasheet View (tự nhập dữ liệu)
  - f Design View: Tạo Bảng theo chế độ tự thiết kế.
  - f Table Wizard: Tạo Bảng theo hướng dẫn của Wizard.
  - f Import Table: Nhập Bảng từ một CSDL khác.
  - f Link Table: Nối kết Bảng.



™      Hoặc click shortcut liên quan trong khung nội dung  
f      Create Table in Design View: Tạo Bảng trong khung  
Design  
View

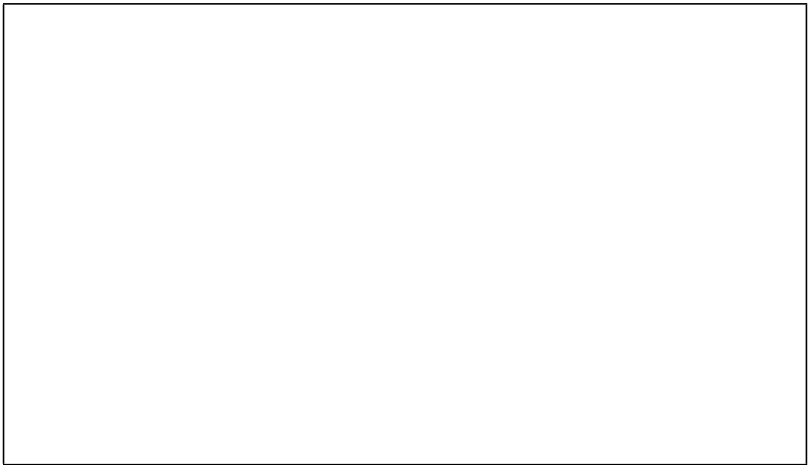
- f* Create Table by using Wizard: **Tạo Bảng với Table Wizard**
- f* Create Table by entering Data: **Tạo Bảng bằng cách nhập trực tiếp.**

#### 2.4.2 Tạo Bảng bằng Wizard

Chọn mục Table Wizard từ hộp thoại New Table và thực hiện các bước của Wizard hay shortcut Create Table by using Wizard.

<sup>TM</sup> **Bước 1: Chọn loại Bảng, Bảng mẫu và Field mẫu**

- f* Chọn loại Bảng (Table Categories): Business hay Personal.
- f* Chọn Bảng mẫu trong khung Sample Tables.
- f* Chọn các Field trong Sample Fields để chuyển sang khung Fields in my new table (Double click hay các nút dấu >, >>, <, <<).
- f* Đổi tên Field đã chọn: Click nút Rename để đổi tên các Fields đã chọn. Làm xong click Next để qua bước kế tiếp.



<sup>TM</sup> **Bước 2: Đặt tên cho Bảng và thiết lập Primary Key**

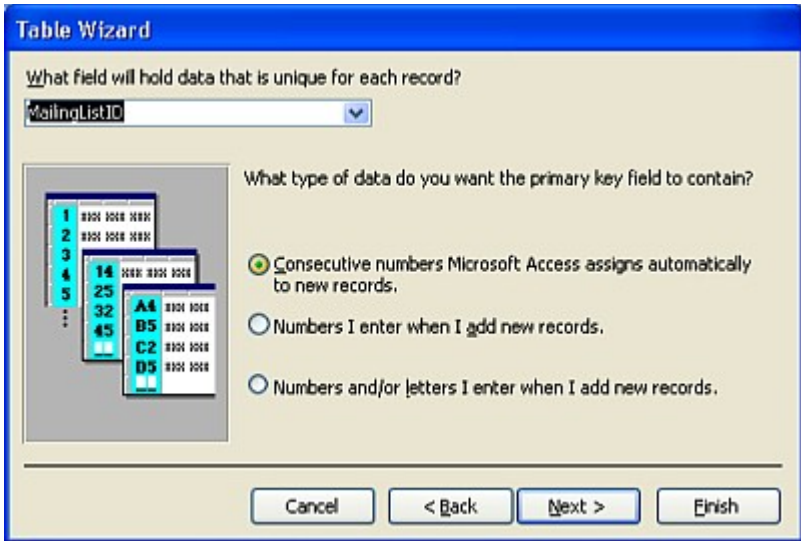
- f* Đặt tên Bảng (khung What do you want to name your Table).
- f* Xác định Wizard đặt Primary Key cho Bảng hay người sử dụng tự đặt.

- Wizard tự thiết lập: chọn mục Yes, set a primary key for me
- Nếu tự đặt Primary Key: chọn mục No, I'll set the primary key, một hộp thoại được hiển thị và chọn một trong

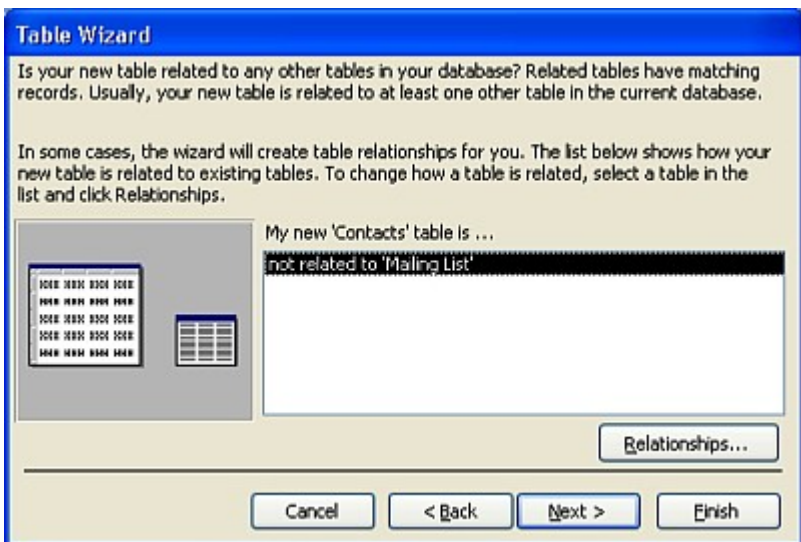
các mục.

- o Consecutive Number Microsoft Access assigns automatically to new records: Access tự động điền trị số cho record mới (tương tự chọn primary key field là autonumber)

- o Number I enter when I add new records: người sử dụng tự điền trị cho Field và là trị số kiểu.
- o Number and/or letters I enter when I add new records: người sử dụng tự điền trị cho Field và là trị kiểu số hay ký tự.
- o Khai báo xong, click Next để qua bước kế tiếp.



™ **Bước 3:** Tạo mối quan hệ giữa Bảng mới với các Bảng hiện có của CSDL

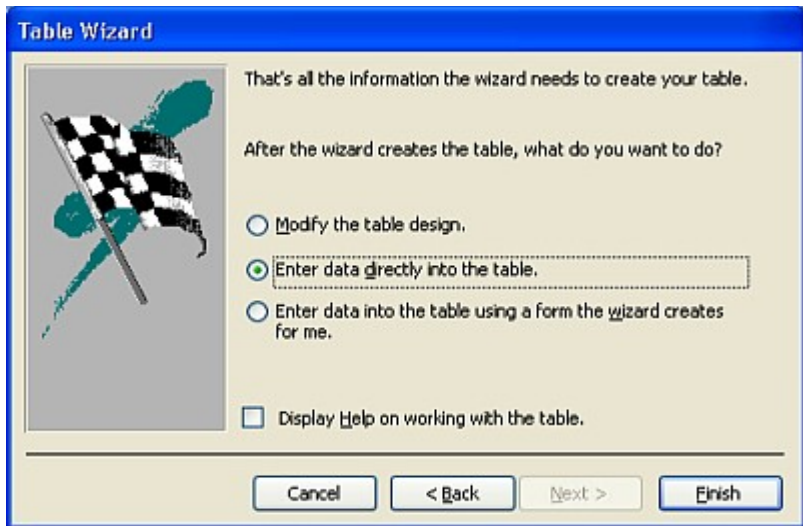


f Nếu Bảng đang tạo là Bảng đầu tiên của CSDL thì Access bỏ

qua bước này.

f Ngược lại nếu CSDL đã có Bảng thì Access phỏng đoán rằng Bảng mới có thể có quan hệ đến các Bảng hiện hành và hiển thị bước này. Nếu muốn tạo các mối quan hệ giữa Bảng mới tạo với các Bảng đã có thì chọn tên Bảng cần khai báo quan hệ với Bảng đang tạo rồi click vào nút RELATIONSHIPS trên hộp thoại để khai báo quan hệ.

™ **Bước 4:** Kết thúc, chọn một trong các mục sau



f Modify the Table design (hiệu chỉnh cấu trúc)

f Enter DATA directly into the Table (nhập trực tiếp DATA)

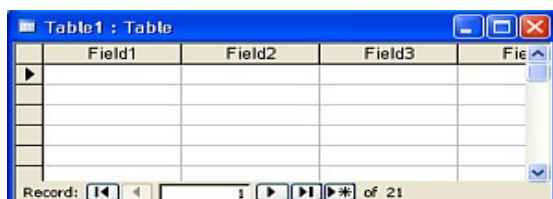
f Enter DATA into the Table using a form the Wizard creates for me (Nhập DATA qua FORM)

f Mục Display Help on working with the Table: hiển thị Help khi làm việc với Bảng

f Click FINISH để kết thúc giai đoạn thiết kế. Tiếp theo hộp thoại của Bảng vừa tạo sẽ hiện ra để nhập liệu (theo hình thức trực tiếp hay dưới hình thức của một Biểu mẫu).

### 2.4.3 Tạo Bảng bằng cách nhập trực tiếp dữ liệu (dùng Datasheet View)

™ Chọn mục Datasheet View từ hộp thoại New Table hay chọn shortcut



Create      new  
Table by entering data.



- ™ Một Bảng hiện ra với 10 cột 22 dòng. Các cột có tên là Field 1, Field 2, .... Field 10.
- ™ Nhập trực tiếp dữ liệu vào Bảng.
- ™ Đổi tên Field
  - f Click phải tên cột - chọn Rename Column trên Shortcut Menu.
  - f Hay đúng vào một ô bất kỳ trên cột muốn đổi tên và chọn menu Format - Rename Column.
- ™ Xóa các cột: chọn cột muốn xóa và chọn Delete trên shortcut Menu hay chọn menu Edit – Delete Column.
- ™ Thêm cột: chọn cột nơi muốn thêm cột mới và chọn Insert Column trên shortcut Menu hay chọn menu Insert – Column.
- ™ Thiết kế xong, đóng Bảng: Access sẽ yêu cầu đặt tên cho Bảng mới (và xác định Primary Key nếu chưa có) và các cột thừa sẽ tự động được xóa.

#### 2.4.4 Tự thiết kế Bảng bằng Design View

Chọn mục Design View từ hộp thoại New Table hay chọn shortcut Create new Table in Design View.  
Màn hình Table Design View gồm 2 phần:

Phần thiết kế

Phần thuộc tính

##### 2.4.4.1 Phần thiết kế cấu trúc Bảng

Gồm cột Field Name, Data Type và Description.

- f Field Name (Tên Field): tối đa 64 ký tự và bắt đầu bằng ký

tự

chữ cái hay số.

*f* Data Type (Kiểu dữ liệu): chọn kiểu dữ liệu trong danh sách

(đã nêu ở mục trên)

*f* Description (Diễn giải): tùy chọn. Dùng ghi các chú thích liên quan đến Field.

#### 2.4.4.2 Phần thuộc tính

Gồm phiếu General và Lookup.

*f* Phiếu General quy định các thuộc tính Field.

*f* Phiếu Lookup tạo danh sách chọn cho Field.

---

Nhấn phím F6 để chuyển đổi qua lại giữa phần thiết kế cấu trúc và khai báo thuộc tính.

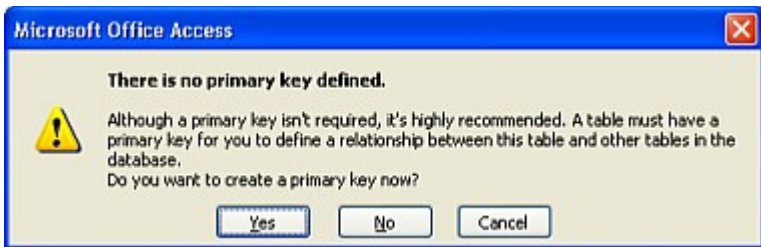
---

#### 2.4.4.3 Xác lập Primary Key

*f* Việc xác lập Primary Key tuy không bắt buộc nhưng nên thực hiện.

*f* Nếu kết thúc thiết kế Bảng mà không xác lập Primary Key thì

Access sẽ thực hiện thông báo nhắc nhở



- Chọn Yes Æ lập Primary Key
- Chọn No Æ không tạo Primary Key
- Chọn Cancel Æ trở lại màn hình thiết kế và không tạo

Primary Key.

*f* Chỉ được xác lập một Primary Key trong một Bảng. Cách xác lập:

*f* Chọn Field muốn xác lập

- Xác lập Primary Key cho một Field: chọn hay đứng tại

Field muốn xác lập Primary Key.

- Xác lập cho Primary Key gồm nhiều Field: phối hợp với phím Shift hay Ctrl để chọn các Field muốn xác lập làm Primary Key.

*f* Click biểu tượng chìa khóa trên Toolbar

*f* Hay từ menu EDIT - Primary Key

*f* Hay click phải và chọn Primary Key trên shortcut Menu.

#### 2.4.4.4 *Ghi lưu*

Thiết kế xong, ghi lưu Bảng và đóng hộp thoại thiết kế.  
Nếu chưa ghi lưu, Access sẽ nhắc ghi lưu Bảng.

## 2.5 NHẬP LIỆU VÀO BẢNG

### 2.5.1 Khái quát

Có thể nhập, hiệu chỉnh dữ liệu trong Bảng theo nhiều cách:

- ™ Nhập, hiệu chỉnh trực tiếp khi làm việc với Bảng.
- ™ Hoặc khi làm việc với Query's Datasheet.
- ™ Hoặc khi làm việc với biểu mẫu.

### 2.5.2 Trực tiếp nhập dữ liệu vào Bảng

Mở Bảng cần nhập liệu

- ™ Từ Database Window, chọn mục loại Tables bên khung đại cương và chọn tên Bảng cần nhập liệu rồi click Open trên Menu của hộp thoại này.
- ™ Hay double click lên tên Bảng.  
Bảng được hiển thị dạng Datasheet View.

#### 2.5.2.1 Cách nhập

*f* Với các Field thường

- Nhập, hiệu chỉnh bình thường. Nhập record mới vào dòng trống kế tiếp tin cuối.
- Hoặc chọn Menu RECORD – DATA ENTRY. Trên Bảng chỉ hiển thị Record trắng để nhập liệu.
- Muốn hiển thị lại các Record của Bảng: chọn menu

RECORD – REMOVE FILTER/SORT

*f* Với Field kiểu AutoNumber: Access sẽ tự nhập.

*f* Với Field kiểu OLE Object

- Đứng tại Field của record muốn đưa 1 Object vào: chọn menu Insert – Object (chọn Create New hay From File)
- Hay Copy vào Clipboard hình ảnh rồi Paste vào Field OLE.
- Hiệu chỉnh một Object: đứng trong Field có Object muốn hiệu chỉnh và double click, ứng dụng liên quan được mở để hiệu chỉnh.

2.5.2.2 Record Selector: để biểu thị trạng thái của tin. Nếu

- f* Có hình tam giác: tin hiện hành.
- f* Có hình cây bút: tin đang được hiệu chỉnh hay nhập liệu.
- f* Có hình hoa thị: tin trống liền kế tin cuối cùng.

	MAMH	HK	MA
	026	1	304051
	026	1	304051
	694	1	309061
	694	1	309061
	694	1	309061

	MAHS	MAMH	HK
	226061148	002	2
	226061153	002	2
		002	2

### 2.5.2.3 Các phím tắt thường dùng

Phím tắt	Ý nghĩa
Tab/Shift+Tab	Đến Field kế tiếp / Field trước
Home/End	Đến Field đầu / cuối của Record hiện hành.
Ctrl+PgUp/PgDn	Đến Record trước / sau (khi dùng Biểu mẫu)
Ctrl+Home/End	Đến Field đầu của Record đầu/ Field cuối của
Shift+F2	Zoom Field có chứa dữ liệu Text lớn hay Memo
F2	Chuyển sang chế độ làm việc với Field

## 2.6 CÁC THUỘC TÍNH SỬ DỤNG

### 2.6.1 Khái quát

™ Property (thuộc tính) là các tính chất gán cho mỗi đối tượng liên quan. Các tính chất này có thể do Access tự gán khi đối tượng được tạo hay do người sử dụng gán cho đối tượng. Thuộc tính của Field là các tính chất liên quan đến Field. Thuộc tính của Bảng là các tính chất liên quan đến Bảng.

™ Property Sheet hay Property Window (Cửa sổ thuộc tính) dùng thao tác với các thuộc tính của đối tượng.

™ Mở cửa sổ thuộc tính: click biểu tượng Properties trên Toolbar hay chọn menu VIEW – Properties để mở cửa sổ thuộc tính của Bảng.

™ Thao tác với thuộc tính của Field: Khai báo trong phần

Properties của cửa sổ thiết kế Bảng (Design View).

### 2.6.2 Các thuộc tính thường dùng

#### 2.6.2.1 Thuộc tính Caption

Là biểu thức chuỗi dài tối đa 2.048 ký tự. Đối với Form và

Report, nếu Caption dài hơn Title Bar thì sẽ bị cắt bớt. Công dụng:

Caption của Field	Tiêu đề cột cho Field (Datasheet của Table/Query)
Caption của Form / Report	Làm tiêu đề cho Form / Report
Caption của Control và Label	Text xuất hiện trong Control, Label.

#### 2.6.2.2 Thuộc tính FieldSize

Được dùng để định kích thước tối đa cho một Field. Cần chỉ định Fieldsize vừa đủ với yêu cầu dữ liệu để có tốc độ xử lý nhanh và tiết kiệm vùng nhớ.

#### 2.6.2.3 Thuộc tính Format

Dùng để ấn định dạng hiển thị của dữ liệu. Thuộc tính này

sử  
dụng các xác lập khác nhau cho những kiểu dữ liệu khác nhau.

◆ Ký tự định dạng dùng chung

Dùng ký hiệu sau để xây dựng bất kỳ định dạng kiểu dữ liệu nào

Ký hiệu	Mô tả
(Space)	Hiển thị khoảng trắng như các ký tự
“ABC”	Hiển thị bất cứ gì trong ngoặc kép như các ký tự
!	Bắt buộc canh trái thay vì canh phải
*	Điền các khoảng trắng khả dụng với các ký tự kế tiếp
\	Hiển thị ký tự kế tiếp như ký tự bình thường
[Color]	Màu (Black, Blue, Green, Red, Cyan, Magenta, Yellow, White).

◆ Custom Format dùng riêng cho Text và Memo

Các phần của định dạng

Phần	Mô tả
Phần thứ 1	Định dạng cho Field với văn bản
Phần thứ 2	Định dạng cho trị NULL hay chuỗi rỗng
2 phần cách nhau bởi dấu chấm phẩy. Ví dụ: @;”None”	

Các ký hiệu định dạng

Ký hiệu	Mô tả
@	Bắt buộc là khoảng trắng hay một ký tự
&	Không bắt buộc là khoảng trắng hay một ký tự
<	Đổi tất cả ký tự thành chữ thường
>	Đổi tất cả ký tự thành chữ hoa

◆ Custom Format dùng riêng cho Logical Data

Các phần của định dạng

Phần	Mô tả
Phần thứ 1	Phần này không tác dụng nhưng phải gõ dấu chấm
Phần thứ 2	Văn bản hiển thị thay cho trị True, Yes, On.
Phần thứ 3	Văn bản hiển thị thay cho trị False, No, Off.

Ví dụ: ;”Always” [Blue]; “Never”[Red]

Lưu ý: để xác lập định dạng có hiệu lực: chọn mục TextBox của phiếu

Lookup (phần Properties của cửa sổ thiết kế).

◆ Custom Format dùng riêng cho Number

Các phần của định dạng

Phần	Mô tả
Phần thứ 1	Định dạng số dương
Phần thứ 2	Định dạng số âm
Phần thứ 3	Định dạng số zero
Phần thứ 4	Định dạng các trị NULL



### Các ký tự định dạng

Ký hiệu	Mô tả
. (chấm) , (phẩy)	Dấu phân cách thập phân và phân cách hàng ngàn (xác lập trong Control Panel)
0	Ký số giữ vị trí. Hiển thị ký số hay số 0
#	Ký số giữ vị trí. Hiển thị ký số hay không có gì cả
\$	Hiển thị ký tự "\$"
%	Phần trăm. Giá trị được nhân 100 và ký hiệu % nối vào
E- hay e- E+ hay e+	Ký hiệu khoa học

#### 2.6.2.4 Thuộc tính Input Mask

Dùng để định dạng dữ liệu nhập. Thuộc tính này sử dụng các xác lập khác nhau cho những kiểu dữ liệu khác nhau.

##### ◆ Các phần của định dạng

Phần	Mô tả
Phần thứ 1	Chỉ định Input Mask cho field (Ví dụ: !(999)999-9999)
Phần thứ 2	Có 2 trị: nếu là 0, ghi dữ liệu và cả các ký tự định dạng. Nếu là 1 hay để trống, chỉ những dữ liệu nhập vào
Phần thứ 3	Chỉ định ký tự hiển thị thay cho khoảng trắng (có thể

##### ◆ Các ký hiệu dùng trong Input Mask

Ký hiệu	Mô tả
0	Ký tự số (0-9) và bắt buộc. Không được dùng dấu cộng
9	Ký tự số hay khoảng trắng (không bắt buộc), không cho
#	Ký tự số hay khoảng trắng, cho dùng dấu +,- (mục không
L	Ký tự chữ (A-Z) và bắt buộc.
?	Ký tự chữ (A-Z) và không bắt buộc.
A	Mẫu tự hay ký số (mục bắt buộc).
a	Mẫu tự hay ký số (mục tùy ý)
&	Ký tự bất kỳ hay khoảng trắng (mục bắt buộc).
C	Ký tự bất kỳ hay khoảng trắng (mục tùy ý).
.,:;- /	Dấu phân cách thập phân, hàng ngàn, ngày giờ (theo Control Panel)
<	Chuyển tất cả thành chữ thường.

>	Chuyển tất cả thành chữ hoa.
!	Canh trái. Có thể kèm ký tự này với bất kỳ Input Mask nào.
\	Ký tự theo sau hiển thị như một ký tự bình thường. (Ví

**GHI CHÚ:** Đặt thuộc tính Input Mask là từ “Password” sẽ tạo ra một Control bảo mật dữ liệu nhập vào (ký tự nhập vào được lưu bình thường nhưng hiển thị thành dấu hoa thị(\*))

◆ Vài kiểu Input Mask thường dùng

Input Mask	Ví dụ
(000) 000-0000	(206) 555-0248
(999) 999-9999	(206) 555-0248 hay ( ) 555-0248
(000) AAA-AAAA	(206) 555-TELE
#999	-20
>L????L?000LO	GREENGR339M3 hay MAY R 452B7
>LOL OLO	T2F 8M4
00000-9999	98115- hay 98115-3007
>L<???????????????	Maria hay Bendan
SSN 000-00-0000	SSN 555-55-5555
>LL00000-0000	DB51392-0493

**GHI CHÚ:** Khi đặt Input Mask và Format cho cùng một dữ liệu, Format sẽ có hiệu lực cho hiển thị dữ liệu và Input Mask có hiệu lực khi nhập liệu.

2.6.2.5 Thuộc tính Decimal Places

- ◆ Dùng để chỉ định số ký số cho phần thập phân và chỉ ảnh

hưởng

lên phần hiển thị chứ không thay đổi trị của dữ liệu.

- ◆ Chỉ định thuộc tính này bị vô hiệu nếu thuộc tính Format được xác lập hay là General Number

Xác lập	Mô tả
Auto	(Mặc định) Các số hiển thị theo chỉ thị của thuộc tính Format
0-15	Chỉ định ký số thập phân

2.6.2.6 Thuộc tính Required

- ◆ Công dụng

Xác lập	Mô tả
Yes	Field bắt buộc phải có một giá trị
No	(Mặc định) Field không bắt buộc phải có một giá trị

- ◆ Dùng hai thuộc tính Required và ZeroLength để phân biệt giữa thông tin chưa tồn tại (lưu như chuỗi rỗng) và thông tin tồn tại nhưng chưa biết (trị Null). Ví dụ: phân biệt khách hàng có điện thoại nhưng chưa biết số và không có điện thoại.
- ◆ Nếu đặt AllowZeroLength là Yes: cho phép nhập chuỗi rỗng vào Field bắt chấp xác lập thuộc tính Required.
- ◆ Nếu đặt Required là Yes và AllowZeroLength là No: không cho phép nhập chuỗi rỗng vào Field.
- ◆ Có thể dùng Input Mask khi nhập liệu vào Field để phân biệt cách hiển thị trị Null và chuỗi rỗng (Ví dụ: dùng “None” khi nhập chuỗi rỗng)
- ◆ Để làm có hiệu lực một mối quan hệ giữa các Bảng quan hệ mà ta không cho phép trị Null
  - Đặt thuộc tính Required của Foreign Key Field của Bảng quan hệ là Yes.
  - Nếu Foreign Key là một phần của Primary Key của Bảng con thì điều này không còn cần thiết vì Primary Key không chấp nhận trị Null.

Kết quả khi phối hợp 2 thuộc tính Required và AllowZeroLength

Required	AllowZeroLength	Thao tác	Trị ghi lưu
No	No	Nhấn Enter, Spacebar Nhập 1 chuỗi rỗng	Null Không cho phép
No	Yes	Nhấn Enter, Spacebar Nhập 1 chuỗi rỗng	Null Chuỗi rỗng
Yes	No	Nhấn Enter, Spacebar Nhập 1 chuỗi rỗng	Không cho phép Không cho phép
Yes	Yes	Nhấn Enter Nhấn Spacebar Nhập 1 chuỗi rỗng	Không cho phép Chuỗi rỗng Chuỗi rỗng

#### 2.6.2.7 Thuộc tính Validation Rule – Validation Text

- ◆ Validation Rule

Quy tắc kiểm tra hợp lệ dữ liệu nhập vào. Quy tắc được xác lập là biểu thức bất kỳ và dài tối đa 2.048 ký tự.

- 
- ◆ Phân biệt ValidationRule cấp Field và cấp mẫu tin  
vào Validation Rule cấp Field quy định quy tắc nhập liệu Field đó.
    - Thiết lập biểu thức điều kiện trong Properties của Field
    - Biểu thức điều kiện không tham chiếu đến Field khác
    - Được thi hành khi kết thúc nhập liệu một Field

- f Validation Rule cấp mẫu tin quy định quy tắc nhập liệu cho toàn mẫu tin (đây là thuộc tính của Bảng)
  - Thiết lập biểu thức điều kiện trong Properties của Bảng (click phải vào vùng trống của Bảng thiết kế và chọn Properties).
  - Biểu thức điều kiện có thể tham chiếu đến Field khác.
  - ValidationRule được thi hành khi kết thúc nhập 1 tin.

◆ Quy định thuộc tính ValidationRule

f Là biểu thức điều kiện, có thể chứa các phép toán so sánh, lý như < > <= >= <> (so sánh), And Or Not (luận lý), Like (chuỗi)

f Nhấn Shift+F2 để mở rộng khung ValidationRule (của sổ Zoom) hay click nút 3 chấm để mở hộp thoại Expression Builder và thiết lập các biểu thức.

f Đối với quy tắc dữ liệu của Field và Record

*Biểu thức không thể chứa User Defined Function, Aggregate Function, các tham chiếu.*

◆ Validation Text

Chuỗi ký tự thông báo khi dữ liệu nhập không đúng với quy tắc kiểm tra. Chuỗi ký tự dài tối đa 255 ký tự.

◆ Vài ví dụ

<>0	Trị nhập vào phải khác không
>1000 Or Is Null	Trị nhập vào phải lớn hơn 100 hay rỗng
Like "A????"	Trị nhập vào phải có 5 ký tự và bắt đầu là chữ A
>= #1/1/96#And <#1/1/97#	Trị nhập vào phải là ngày trong năm 1996

2.6.2.8 Thuộc tính Indexed

- ◆ Công dụng: dùng để tạo chỉ mục cho Field nhằm tăng tốc độ truy vấn trên Field, thực hiện việc sắp xếp, lọc dữ liệu.
- ◆ Xác lập thuộc tính Index cho Field đơn

Xác lập	Mô tả
No	(Mặc định) không có chỉ mục
Yes (Duplicates OK)	Chỉ mục được phép trùng
Yes (No Duplicates)	Chỉ mục duy nhất, không được phép trùng

- ◆ Xác lập một thuộc tính Index cho nhiều Field

f Mở Bảng trong chế độ Design View

f Click nút Indexes hay chọn menu View – Indexes để mở cửa sổ Indexes.

- f* Trong dòng trống đầu tiên của Index Name: nhập tên cho Index
- f* Trong cột Field Name, click chọn Field đầu tiên cho Index từ trong danh sách đổ xuống (DropDown List).
- f* Trong dòng kế tiếp của cột Field Name, chọn Field thứ 2 cho Index (bỏ trống cột Index Name của dòng này). Lập lại bước này cho đến khi đã chọn tất cả các Field muốn bao gồm trong Index này.
- f* Có thể dùng tối đa 10 Field của Index Field phối hợp.



- f* Nếu Primary Key của một Bảng là Field đơn, thuộc tính Index sẽ được gán cho (No Duplicates).
- f* Không thể đặt Index cho Field có kiểu là Hyperlink, Memo hay OLE Object.

## 2.7 THIẾT LẬP LOOKUP

### 2.7.1 Giới thiệu

- ™ Có thể quy định dữ liệu nhập của một Field phải là một trong các trị định trước.
- ™ Danh sách các trị định trước có thể được tạo trực tiếp hay trích từ một Bảng hay Query khác.
- ™ Để tiện lợi cho người dùng, có thể khai báo danh sách các trị định trước và người dùng chỉ cần chọn trị thích hợp – Đây chính là tạo Lookup cho Field.
- ™ Cách thiết lập Lookup
  - f* Dùng Lookup Wizard
  - f* Hay khai báo trực tiếp trong phần Properties của Field

### 2.7.2 Tạo Lookup với Lookup Wizard

#### 2.7.2.1 Quy trình

Trong cột Data Type, xác lập kiểu cho Field trước rồi chọn

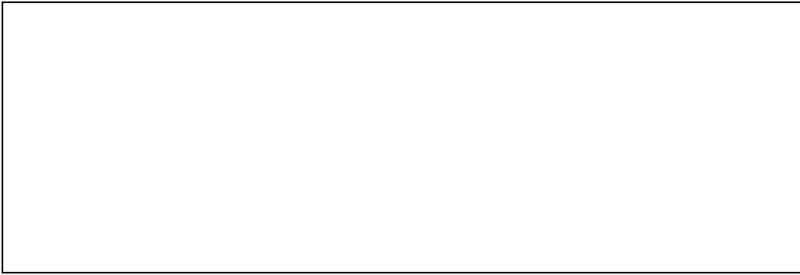
mục

Lookup Wizard cũng trong danh sách trái xuống trong cột Data

Type để khởi động Lookup Wizard và thực hiện các bước theo hướng dẫn.

### 2.7.2.2 Các bước của Lookup Wizard

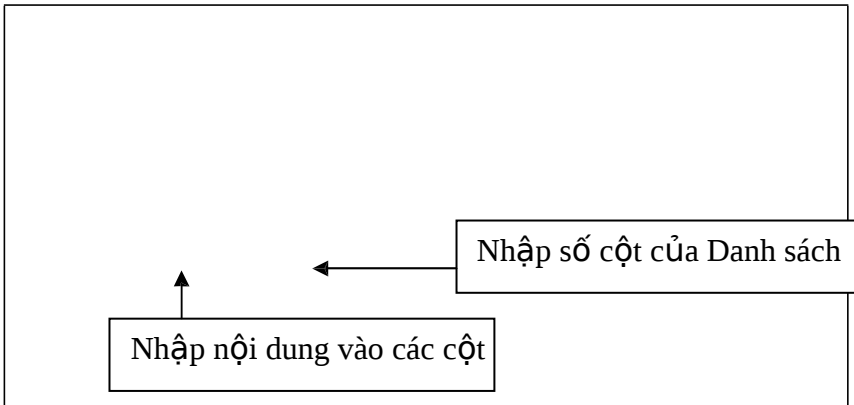
- ◆ Chọn cách khai báo danh sách trị định trước



Gồm chọn trị từ Bảng/Truy vấn hoặc tự nhập danh sách trị.

- ◆ Khai báo danh sách trị

f Nếu chọn cách tự nhập danh sách trị, màn hình như sau



- Nhập số cột của Danh sách
- Nhập trị vào từng cột của Danh sách (mỗi dòng là 1 mục chọn)
- Click Next: chọn cột chứa trị nhập vào Field nếu mục được chọn

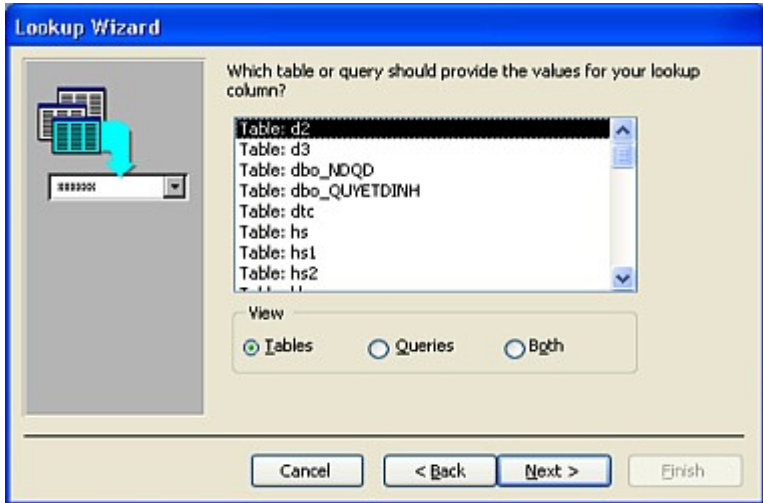


- Click Next: đặt nhãn cho Lookup, click Finish để kết thúc.

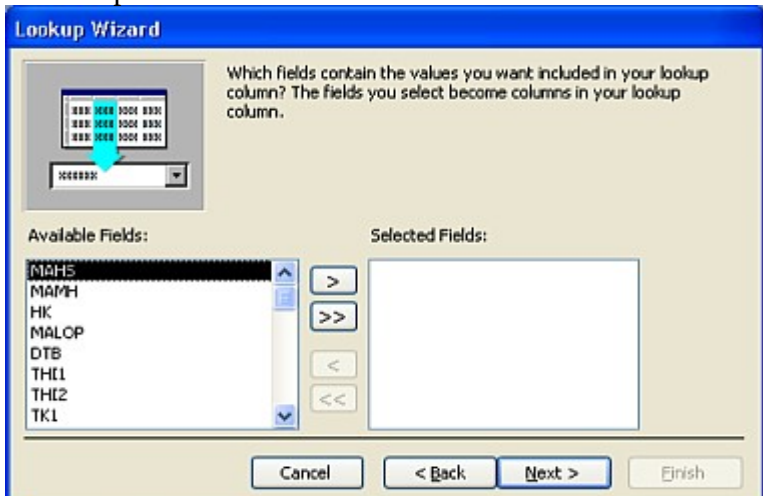
- Kết quả như sau

CONGNHAN : Table			
	MACN	HOTEN	PX
▶ +	001	Nguyen Thuy Van	
+ 002		Le Hai Viet	1 May
+ 003		Tran Binh	2 Cát
+ 004		Phan Van Thanh	3 Ráp
+ 005		Tran Minh Hoa	?

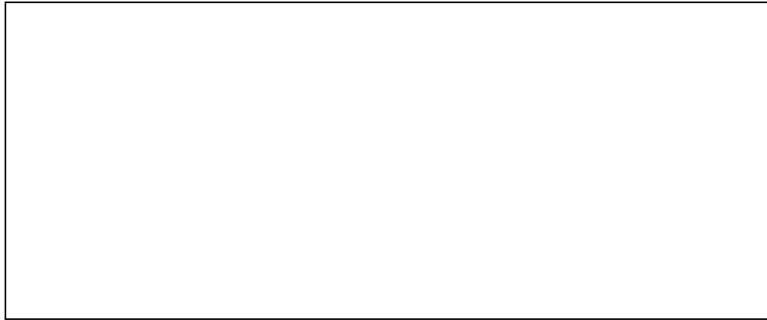
- f Nếu chọn lấy danh sách từ Bảng hay Truy vấn thì màn hình như sau



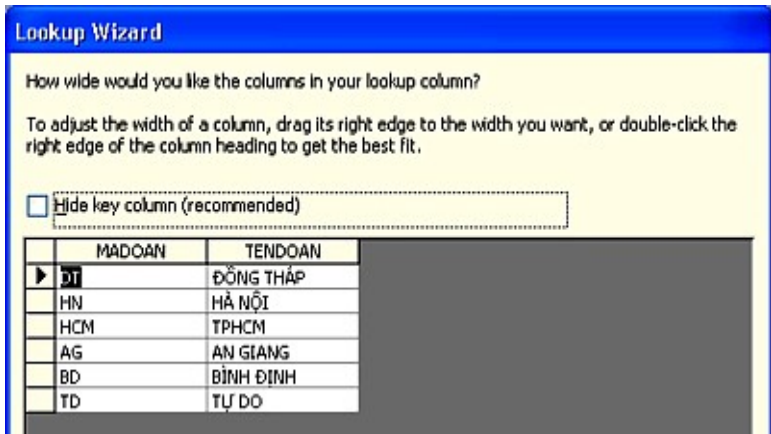
- Chọn loại đối tượng nguồn cần hiển thị (Bảng hay Truy vấn hay cả hai) và chọn tên đối tượng nguồn
- Click Next để qua bước kế tiếp: chọn Field hiển thị trong Lookup



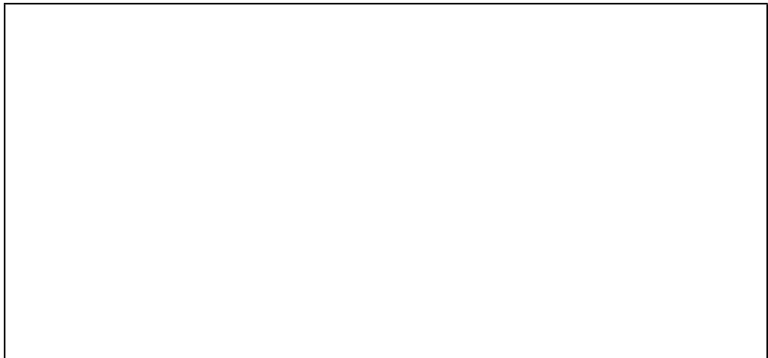
- Click Next qua bước kế tiếp: khai báo xếp thứ tự mẫu tin



- Click Next để qua bước kế tiếp: khai báo bề rộng cột danh sách



- Nếu đánh dấu chọn mục Hide key Column thì cột khóa sẽ được ẩn khi hiển thị danh sách.
- Click để qua bước kế tiếp: chọn Field của danh sách chứa trị nhập vào Field khi một mục được chọn.



- Click Next để qua bước kế tiếp: đặt nhãn cho Lookup và click Finish để kết thúc.

### 2.7.3 Tự thiết kế Lookup

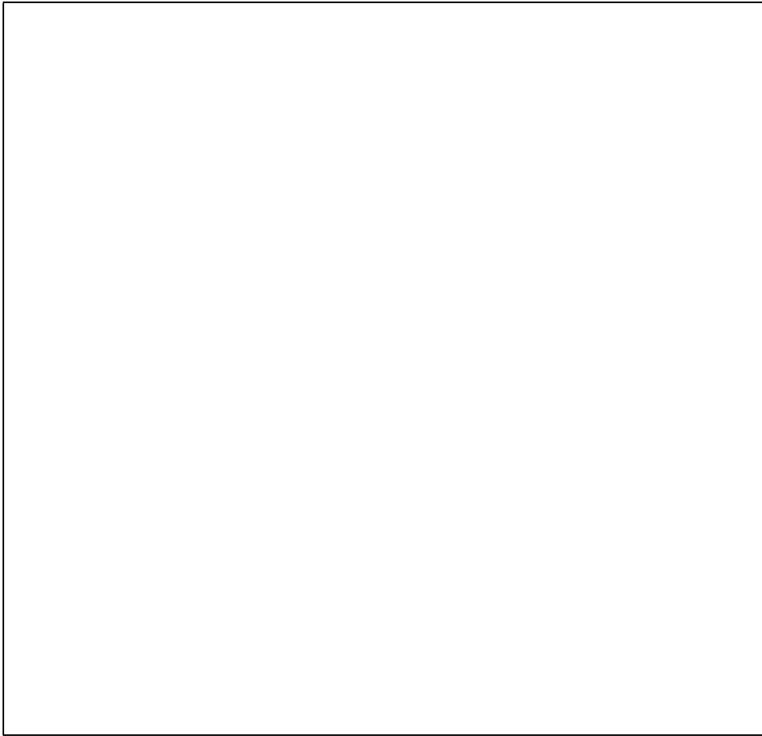
™ Trong cửa sổ thiết kế Bảng, đứng tại tên Field (đã khai báo

Field Type) cần tạo Lookup.

™ Chọn phiếu Lookup trên phần Properties và khai báo

Mục	Giá trị	Ý nghĩa
Display Control	Combo Box, List Box, Check Box hay Text Box	Dạng hiển thị là Combo Box, List Box, Text Box hay Check Box. Chọn Combo Box
Row Source Type	Table/Query hay Value List	Nguồn cung cấp dữ liệu cho Lookup Table/Query: nguồn là Bảng hay Query Value List: nguồn là trị quy định trực tiếp.
Row Source	SELECT... hay (danh mục quy định)	Nguồn dữ liệu (Lệnh SQL hay tên Bảng/Query) hoặc trị nhập trực tiếp Nếu là trị nhập trực tiếp thì nhập trị theo từng dòng và cách nhau bởi dấu chấm phẩy
Bound Column	Một số nguyên dương	Chỉ cột của danh sách ràng buộc với Field
Column Count	Một số nguyên dương	Số cột của Combo Box hay List Box
Column heads	Yes/No	Có Tiêu đề cột hay không
Column Widths	n1; n2	Độ rộng của các cột (nếu 0: không hiển thị cột) phân cách bởi dấu chấm phẩy.
List Row	Một số dương	Số dòng của danh sách
List Width	Một số dương	Chiều rộng danh sách (tính chung các cột)
Limit to List	Yes/No	Cho phép (Yes)/ không cho phép (No) nhập vào Field của Bảng một trị ngoài danh sách của

™ Ví dụ một mẫu khai báo



- Loại nguồn dữ liệu là Table / Query
- Nguồn dữ liệu là Bảng LOP
- Cột ràng buộc là cột 1 của Bảng LOP
- Danh sách có 2 cột
- Hiển thị danh sách với 3 dòng
- Không cho phép nhập ngoài danh sách của Lookup

**BÀI TẬP:**

1. Tạo tập tin CSDL để quản lý các công trình xây dựng, đặt tên

QLCT.MDB

2. Trong tập tin QLCT.MDB hãy tạo các table sau đây:

Table CONGTRINH có cấu trúc sau: đặt khóa chính là MACT

Field Name	Data Type	Field size	Yêu cầu
<b><u>MACT</u></b>	Text	5	Mã công trình, khóa chính. Gồm 2 ký tự CT và 3 ký số (chữ CT có lưu trong field). Input Mask: "CT"000;0
TENCT	Text	50	Tên công trình
NGAYKC	Datetime		Ngày khởi công, nhỏ hơn ngày hiện tại Validation Rule: < Date() Validation Text: Ngày khởi công phải nhỏ hơn ngày hiện tại

Table VATTU có cấu trúc: đặt khóa chính là MAVT

Field Name	Data Type	Field size	Yêu cầu
<b><u>MAVT</u></b>	Text	5	Mã vật tư, khóa chính. Gồm 2 ký tự hoa và 3 ký số. Input Mask: >LL000
TENVT	Text	20	Tên vật tư, nhập 1 chữ đầu hoa, còn lại là chữ thường. Input Mask: >L<????????????????????
DVT	Text	10	Đơn vị tính, dùng combo box. Chọn trang Lookup: Display control: là combo box Row source type: Value list Row source: bao;thùng;kg;tấn;mét;m3

Table XUAT\_VT có cấu trúc: đặt khóa chính là SOPHIEU

Field Name	Data Type	Field size	Yêu cầu
<b><u>SOPHIEU</u></b>	Text	4	Số phiếu xuất, gồm 1 ký tự X và 3 ký số (chữ X có lưu trong field). Input Mask: \X000;0

NGAY	Datetime		Ngày xuất nhỏ hơn hay bằng ngày hiện tại. Validation Rule: <= Date() Validation text: Ngày xuất phải nhỏ hơn hay bằng ngày hiện
MACT	Text	5	Mã công trình, tạo Combo box chọn từ Table CONGTRINH. Bấm trang Lookup Display control: Combo box Row source type: Table/Query Row source: CONGTRINH Bound column: 1 Column count: 2 Column widths: 1;2 List width: 3

Table CHITIET\_XUAT có cấu trúc: khóa chính là SOPHIEU và MAVT

Field Name	Data Type	Field size	Yêu cầu
<b><u>SOPHIEU</u></b>	Text	4	Số phiếu
<b><u>MAVT</u></b>	Text	5	Mã vật tư dùng combo box chọn từ Table VATTU. Bấm trang Lookup Display control: Combo box Row source type: Table/Query Row source: VATTU Bound column: 1 Column count: 2 Column widths: 1; 1.5 List width: 2.5
SOLUONG	Number	Long Integer	Số lượng, định dạng nhóm 3 số lại và phải nhập lớn hơn 0 Format: #,### Validation Rule: > 0
DONGIA	Number	Long Integer	Đơn giá, định dạng nhóm 3 số lại có ký hiệu tiền VN và phải nhập lớn hơn 0 Format: #,### “đ” Validation Rule: > 0

3. Liên kết các Table và nhập dữ liệu sau

Table VATTU

MAVT	TENVT	DVT
CA001	Cát	m <sup>3</sup>
DA002	Đá xanh	tấn
SA003	Sắt cuộn	kg
TL004	Tôn lạnh	mét
XM005	Xi măng	bao

Table CONGTRINH

MACT	TENCT	NGAYKC
CT001	Nhà nghỉ công đoàn	01/10/05
CT002	Nhà trẻ Hoa mai	16/12/05
CT003	Công viên Tháng 8	22/01/06
CT004	Nhà máy Thắng lợi	01/02/06
CT005	Trường Kim đồng	10/02/06

Table XUAT\_VT

SOPHIEU	NGAY	MACT
X001	01/11/05	CT001
X002	05/12/05	CT001
X003	06/02/06	CT001
X004	21/12/05	CT003
X005	02/03/06	CT003
X006	22/03/06	CT004

Table CHITIET\_XUAT

SOPHIEU	MAVT	SOLUONG	DONGIA
X001	CA001	12	12000
X001	DA002	7	314000
X001	SA003	400	4500
X002	SA003	100	5300
X002	XM005	100	43000
X003	CA001	70	15000
X003	SA003	120	4000
X004	TL004	750	78000
X004	XM005	150	45000
X005	DA002	9	270000
X006	XM005	90	45000
X006	CA001	20	18000



## CHƯƠNG 3

**THAO TÁC TRÊN BẢNG**

- Hiệu đính Bảng
- Khai thác thông tin của Bảng

**3.1 HIỆU ĐÍNH BẢNG**

Hiệu đính một Bảng để thay đổi cấu trúc Bảng, hoặc thay đổi nội dung các Record hoặc thay đổi cách trình bày.

**3.1.1 Thay đổi cấu trúc****3.1.1.1 *Thay đổi thuộc tính, thêm bớt Field***

Mở CSDL liên quan, chọn và mở Bảng cần hiệu chỉnh trong chế độ thiết kế (DESIGN VIEW)

*f* Thay đổi thuộc tính:

Chọn Field và hiệu chỉnh trong Property Sheet. Nếu Field đó đang giữ vai trò quan hệ (Relation Field) với một Bảng khác thì có thể không hiệu chỉnh được.

*f* Bỏ Field cũ:

Chọn Field và chọn menu EDIT (hay Shortcut Menu) chọn

DELETE ROW

*f* Thêm Field mới:

Chọn dòng tại vị trí muốn thêm vào Field mới và chọn menu

INSERT – ROW hay Shortcut Menu – INSERT ROW

*f* Sửa đổi cấu trúc:

Chọn dòng liên quan

Hiệu chỉnh cấu trúc (chỉ thực hiện hiệu chỉnh khi thật cần thiết

để tránh bị mất dữ liệu).

**3.1.1.2 *Nếu chỉ thay đổi tên Field***

*f* Mở Bảng ở chế độ DataSheet – chọn cột liên quan

*f* Chọn menu Format – Rename column

*f* hay Menu shortcut – Rename column

**3.1.2 Hiệu chỉnh tin**

<sup>™</sup> Sửa nội dung Record:

Chọn Record liên quan và sửa chữa tùy ý. Không thể hiệu

*chính*

*AutoNumber.*

™ Hủy bỏ Record:

Chọn tin, chọn menu EDIT (hay Shortcut Menu) – DELETE RECORD hay nhấn phím DELETE

™ Thêm Record mới:

Nhập vào dòng cuối cùng hay chọn menu Insert – New Record (các Record sẽ được sắp xếp lại khi Bảng được mở lần sau).

™ Chọn toàn bộ các Record:

Nhấn Ctrl + A hay chọn menu EDIT – Select ALL

™ Sao chép, cắt, dán Record:

Chọn Record – dùng menu EDIT hay các tổ hợp phím tắt (Ctrl+C, Ctrl+X, Ctrl+V)

### 3.1.3 Thay đổi cách trình bày

#### 3.1.3.1 Định dạng

Mở Bảng trong chế độ DataSheet – chọn menu FORMAT - chọn lệnh liên quan

f Mục DataSheet

- CELL EFFECT:

Trình bày các ô số liệu theo dạng phẳng (Flat), nhô lên (Raised) hay lõm xuống (Sunken).

- GRIDLINES SHOW:

Hiển thị đường lưới ngang dọc hay không (chỉ hiệu lực khi

Cell Effect là Flat)

- BACKGROUND COLOR: Xác lập màu nền.

- GRIDLINE COLOR: Xác lập màu đường lưới.

- BORDERS AND LINE STYLES: Xác lập kiểu đường viền và đường lưới.

f Mục FONT: Thiết lập Font chữ cho Bảng.

f Mục ROW HEIGHT và mục COLUMN WIDTH

Ấn định độ cao của dòng và độ rộng của cột chỉ định (có thể

dùng Mouse để hiệu chỉnh)

f Mục RENAME

COLUMN Dùng để đặt

tên lại cho cột

f Mục HIDE COLUMN và UNHIDE

COLUMN Che / hiển thị lại cột chỉ định

f Mục FREEZE COLUMN và UNFREEZE COLUMNS

Cố định / bỏ cố định cột chỉ định. Cột được cố định phải đặt phía trái Bảng.

#### 3.1.3.2 Di chuyển cột

Chọn cột và nhấn giữ nút trái Mouse (có hình chữ nhật) – rê đến vị trí mới. Khi rê, có vệt đứng để xác định vị trí cột

### 3.1.3.3 Sao chép

- Thao tác Copy và Paste / Paste Special cũng tương tự như các

ứng dụng khác của bộ Office.

- Ngoài ra còn có chức năng Paste Append dùng để dán nối đuôi từ Clipboard vào Record hiện hành. *Thao tác* : chọn menu EDIT – PASTE APPEND.

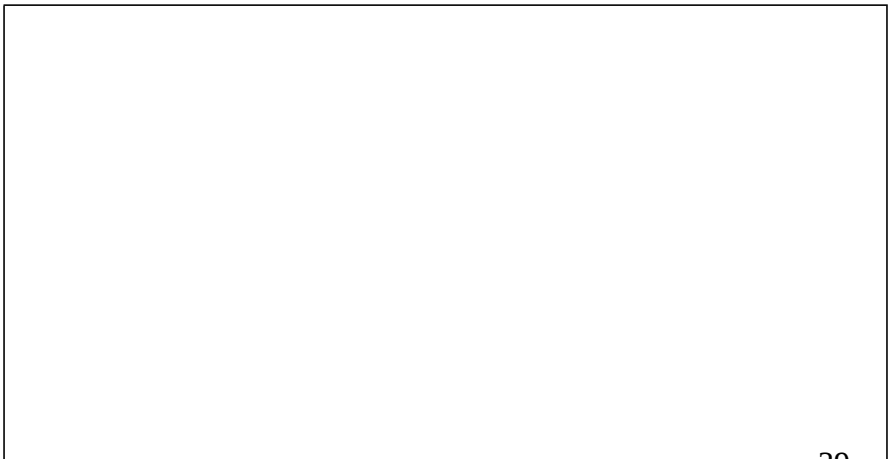
## 3.2 KHAI THÁC THÔNG TIN CỦA BẢNG

### 3.2.1 Tìm thay

- Đặt điểm nháy vào trong cột cần tìm thay.
- Chọn menu EDIT – FIND hay REPLACE (phím tắt Ctrl+F/Ctrl+)
  - Nhập nội dung cần tìm/thay vào khung Find What (và khung Replace With).
  - Chỉ định các mục chọn
    - Match Case: Chính xác với mẫu chọn (phân biệt chữ thường chữ hoa)
    - Match Wole Field: Mẫu chọn phải chính xác toàn Field
    - Search Only Current Field: Chỉ tìm trong Field hiện hành
    - Search Field As Formatted: Tìm Field như định dạng
  - Click vào một trong các nút Ctrl để thực hiện
    - Find Next : Tìm tiếp
    - Replace : Thay thế
    - Replace All : Thay thế tất cả

### 3.2.2 Sắp xếp (SORT)

- Sắp xếp là hiển thị các mẫu tin theo một trật tự nào đó



- Sắp xếp theo từng cột
  - Đứng trong (hoặc chọn) cột cần sắp xếp

- Click nút SORT ASCENDING / DESCENDING trên thanh dụng cụ
- Hay chọn menu RECORD - SORT - ASCENDING / DESCENDING
- Nếu cần sắp xếp theo nhiều cột
  - Di chuyển các cột liên quan đến các vị trí cạnh nhau (thứ tự ưu tiên cột khóa được tính từ bên trái).
  - Sau đó chọn các cột này rồi thực hiện SORT tương tự như nêu trên

### 3.2.3 Lọc dữ liệu

Lọc là hiển thị các mẫu tin theo một tiêu chuẩn nào đó. Trình tự thực hiện: chọn menu RECORDS - FILTER - chọn mục FILTER thích hợp.

#### 3.2.3.1 Filter by Selection

- Lọc theo trị đang được chọn nào đó.
  - Tiêu chuẩn lọc là trị của một Field của một Record: Đứng trong Field của Record có trị muốn làm tiêu chuẩn lọc – click nút Filter by Selection trên Toolbar hay chọn menu RECORD, chọn FILTER BY SELECTION.
  - Tiêu chuẩn lọc là một phần trị của một Field của Record: Chọn phần dùng làm tiêu chuẩn lọc rồi thực hiện lọc như nêu trên.
- Không dùng được biểu thức trong phương thức lọc này.

#### 3.2.3.2 Filter Excluding Selection

Tương tự như Filter by Selection nhưng chỉ những Record không đáp ứng điều kiện lọc mới được lọc.

#### 3.2.3.3 Filter by Form

- Cách này được dùng để đặt các điều kiện lọc cho một Field cụ thể.
  - Thiết lập điều kiện lọc
    - Click biểu tượng Filter by Form hoặc chọn menu RECORD - FILTER - FILTER BY FORM
    - Chọn Phiếu Look For và nhập điều kiện lọc vào Field liên quan trong hộp thoại Filter by Form.
    - Chọn Phiếu OR và nhập bổ sung điều kiện lọc vào Field liên quan
  - Cách nhập điều kiện lọc
    - γ Cách 1 : Khai báo biểu thức lọc
      - o Điều kiện lọc là biểu thức chuỗi

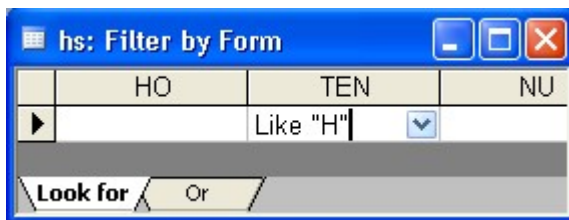
Nhập trực tiếp chuỗi → Access sẽ tự động đưa vào

khóa

Like và đặt chuỗi trong dấu nháy kép. Có thể dùng ký tự

đại diện.

- o Điều kiện lọc là biểu thức số : nhập bình thường
- o Điều kiện lọc là dữ liệu ngày
  - Nhập theo quy ước Control Panel và đặt trong dấu rào (#)
  - Có thể nhập trực tiếp và Access tự động đưa vào cặp dấu rào.
- o Dùng các tác tử
  - số học: cộng (+), trừ (-), nhân (\*), chia (/), chia nguyên (\)
  - nhóm: trong khoảng (Between ...And), In (trong)
  - Logic: And, Or, Not
  - Trị Null: Is Null, Not Is Null, Is Not Null
  - so sánh: lớn / nhỏ hơn (>,<), lớn / nhỏ hơn hay bằng (>=,<=), khác (<>), bằng (=).

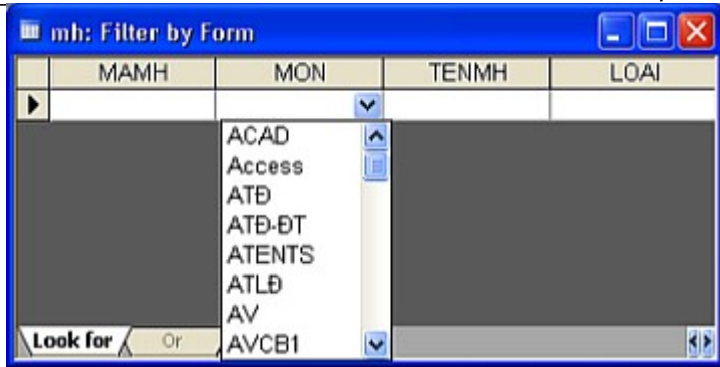


Ví dụ : Like "H\*" (có thể chỉ nhập H\*, Access sẽ viết lại đầy đủ

biểu thức) hay >= #05/01/97#

y Cách 2 : Click nút Dropdown Menu để chọn điều kiện lọc.





□ Thực hiện lọc

Click nút Apply Filter (hình cái phễu) trên Toolbar hay chọn menu RECORD - APPLY FILTER / SORT

### 3.2.3.4 Advanced Filter / Sort

- Dùng để thiết lập các điều kiện lọc một cách chi tiết hơn.



- Thiết lập điều kiện lọc
  - Click nút Filter by Form hoặc chọn menu RECORDS - FILTER - ADVANCED FILTER / SORT
  - Nhập điều kiện lọc vào hộp thoại Advanced Filter / Sort: chọn tên Field, nhập điều kiện lọc trong dòng Criteria, bổ sung điều kiện lọc trong dòng Or
  - Các điều kiện lọc trên cùng dòng có nghĩa là And.
- Thực hiện lọc : Click nút Apply Filter (hình cái phễu) trên Toolbar hay Menu RECORD - APPLY FILTER / SORT.

### 3.2.3.5 Bỏ lọc

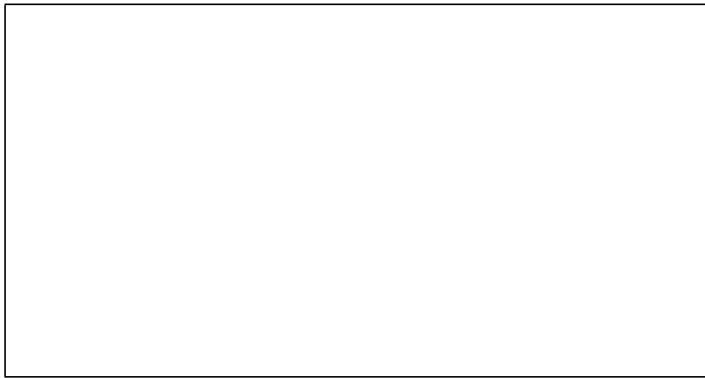
Click nút Remove Filter (hình phễu) hay chọn menu RECORD và chọn REMOVE FILTER.

### 3.2.4 Sao chép Bảng

- Sao chép Bảng trong cùng tập CSDL
  - Cách 1
    - Chọn Bảng rồi Ctrl+R
    - Bảng sao chép được tạo với tên là Copy of ...: Đặt lại tên Bảng cho phù hợp.
    - Bảng sao chép gồm cấu trúc và dữ liệu bảng như nguồn.
  - Cách 2
    - Chọn Bảng muốn sao chép trong hộp thoại Database và Copy vào Clipboard (Ctrl+C hay chọn menu EDIT - COPY)
    - Dán Bảng từ Clipboard (Ctrl+V hay chọn menu EDIT -

PASE)

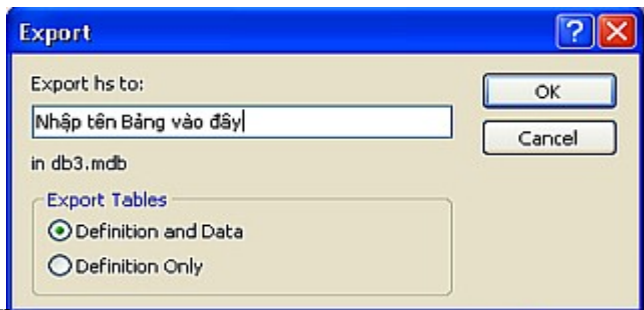
- Chọn mục thích hợp trong hộp thoại Pase.



- o Structure Only : Chỉ chép cấu trúc.
- o Structure and Data : Chép cấu trúc và dữ liệu.
- o Append Data to Existing Bảng : Ghi nối đuôi dữ liệu vào Bảng đích được chọn (phải chọn Bảng).
- ▢ Sao chép Bảng qua tập CSDL khác
  - ▢ Cách 1:
    - Copy Bảng nguồn vào ClipBoard
    - Mở tập CSDL đích và dán (chọn cách dán như nêu trên)
  - ▢ Cách 2: Mở cả hai tập CSDL nguồn và đích rồi rê Bảng từ CSDL nguồn qua CSDL đích

### 3.2.5 Export Bảng sang CSDL khác

- ▢ Chọn Bảng muốn Export
- ▢ Chọn menu FILE - EXPORT
- ▢ Trong hộp thoại Export To, chọn tập tin CSDL đích.
- ▢ Trong hộp



Export, chọn cách Export

- ▢ Definition and Data : Chép cả cấu trúc và Dữ liệu của Bảng nguồn.
- ▢ Definition Only : Chỉ chép cấu trúc Bảng nguồn.

### 3.2.6 Đổi tên Bảng

Tại cửa sổ Database, chọn Bảng cần đổi tên và thực hiện một trong các cách sau : chọn menu EDIT - RENAME hay nhấn phím tắt F2 hay chọn RENAME trên Shortcut Menu và nhập tên mới cho Bảng.

**BÀI TẬP:**

1. Tạo tập tin CSDL để quản lý cửa hàng mua bán văn phòng phẩm  
đặt tên là VPP.MDB

2. Trong tập tin VPP.MDB hãy tạo các table sau đây

Table KHACH\_HANG: khóa chính là MAKH

Field Name	Data Type	Field size	Field Properties
<u>MAKH</u>	Text	4	Gồm 1 ký tự K và 3 ký số (chữ K có lưu trong field). Input Mask: \K000;0
TENKH	Text	40	Có nhập hiện màu xanh, nếu không nhập thì hiện chữ “khách vắng lai” màu đỏ Format: @[Blue;]”Khách vắng lai”[Red]
DIACHI	Text	50	
DIENTHOAI	Text	10	Input Mask: 000-0000

**Dữ liệu nhập**

MAKH	TENKH	DIACHI	DIENTHOAI
K001	Cty TNHH Minh Đức	66 Trần Quang Khải, Q3	8327890
K002	Cty Du lịch Đồng Tháp	365 Điện Biên Phủ, Q10	8453612
K003	Khách sạn Hương Sen	155 Nguyễn Đình Chiểu, Q3	8384908
K004	Xí nghiệp Điện Biên	81 Lạc Long Quân, TB	8456432
K005	Khách sạn Hải Dương	60 Bạch Đằng, Q1	8917645
K006	Xí nghiệp Đồng Tâm	212 Hai Bà Trưng, Q1	8154981
K007	Xí nghiệp giày Đồng Đa	181 Trần Quang Diệu, Q1	8651299
K008	Cty TNHH Hoàn Cầu	254 An Dương Vương, Q5	
K009	Xí nghiệp Dược phẩm 2/9	314 CMT8, Q3	
K010	Cty giấy dép Vina	489 Hùng Vương, Q5	8235471

Table MAT\_HANG: khóa chính là MAMH

Field Name	Data Type	Field size	Field Properties
<u>MAMH</u>	Text	3	Gồm 1 ký tự V và 2 ký số (chữ V có lưu trong field). Input Mask: \V00;0
TENMH	Text	30	Tên mặt hàng
DVT	Text	10	Đơn vị tính, nhập 1 chữ đầu hoa, còn lại là chữ thường. Input Mask: >L<????????? Dùng combo box liệt kê các giá trị
DONGIA	Number	Long Integer	Format: #,### Validation Rule: >= 5000 Validation text: Đơn giá từ 5000 trở lên

Table HOADON: khóa chính là SOHD

Field Name	Data Type	Field size	Field Properties
<u>SOHD</u>	Text	3	Input Mask: 000
MAKH	Text	4	Là combo box chọn từ table KHACH_HANG Display control: Combo box Row source type: Table/Query Row source: KHACH_HANG Bound column: 1 Column count: 2 Column widths: 1;2 List width: 3
NGAYDAT	Date/Time		nhỏ hơn hay bằng ngày hiện tại
NGAYGIAO	Date/Time		Ngày giao hàng

Chọn menu View → Properties → Validation Rule: gõ vào  
[NGAYDAT]<=[NGAYGIAO]

Table CHITIET\_HD: khóa chính là SOHD và MAMH

Field Name	Data Type	Field Size	Ghi chú
SOHD	Text	3	Số hóa đơn
MAMH	Text	3	Mã mặt hàng
SOLUONG	Number	Long Integer	Số lượng, lớn hơn 0

- Thiết lập mối quan hệ giữa các Table, sau đó nhập dữ liệu sau vào các table MAT\_HANG, HOADON, CHITiet\_HD

Table MAT\_HANG

MAMH	TENMH	DVT	DONGIA
V01	Bút chì	Cây	13000
V02	Bút lông viết bảng	Hộp	55000
V03	Giấy vi tính A4	Gam	36000
V04	Đĩa mềm 1,4MB Maxell	Hộp	45000
V05	Bút bi Thiên Long	Hộp	13000
V06	Thước vẽ	Cây	8500
V07	Kẹp giấy	Hộp	7000
V08	Bìa hồ sơ	Cái	28000
V09	Băng keo 2 mặt	Cuộn	8500
V10	Bút xóa	Cây	15000

Table HOADON

SOHD	MAKH	NGAYDAT	NGAYGIAO
001	K006	1/14/2006	1/15/2006
002	K007	1/17/2006	1/21/2006
003	K001	1/28/2006	1/30/2006
004	K009	2/13/2006	2/13/2006
005	K002	2/19/2006	2/26/2006
006	K006	2/19/2006	2/22/2006

Table CHITiet\_HD

SOHD	MAHH	SOLUONG
001	V02	40
001	V04	50
001	V06	40
001	V01	100
002	V05	80
002	V01	50
003	V04	100
003	V03	20
003	V07	90

SOHD	MAHH	SOLUONG
004	V01	150
004	V05	180
004	V10	50
005	V08	200
005	V03	60
006	V03	200
006	V04	60
006	V05	50
006	V06	20

- Copy table MAT\_HANG thành table HANGHOA.
- Export table HANGHOA ra file Excel. Xóa table HANGHOA.
- Đóng Access, mở Excel và xem tập tin HANGHOA.XLS

## CHƯƠNG 4

## MỐI QUAN HỆ GIỮA CÁC BẢNG

- Khái quát
- Khai báo quan hệ

## 4.1 KHÁI QUÁT

4.1.1 Mối quan hệ (Relationships)

- MS-ACCESS là hệ quản trị CSDL quan hệ (Relational Database Management System) nên có thể sử dụng dữ kiện lấy từ nhiều Bảng khác nhau nếu các Bảng này có mối quan hệ với nhau. Muốn vậy, phải khai báo mối quan hệ giữa các Bảng liên quan.
- Tác vụ khai thác: MS-ACCESS dựa vào mối quan hệ đã được định nghĩa giữa các Bảng để thực hiện các tác vụ sau
  - Tạo liên kết trong các Bảng văn tin (Query).
  - Hiện thị các Mẫu tin có mối quan hệ trong biểu mẫu con (SubForm) hay biểu báo con (SubReport).
  - Bắt buộc quá trình khai thác dữ kiện phải tôn trọng tính toàn vẹn tham chiếu dữ kiện (Referential Integrity) để bảo vệ sự quan hệ giữa dữ kiện của các Bảng.

4.1.2 Quy định mối quan hệ

Quy định mối quan hệ giữa các Bảng bằng cách khai báo một hay nhiều Field của Bảng này mà Field đó chứa trị giống với trị trong những Record quan hệ ở Bảng kia.

- Những Field đối chiếu trong các Bảng có quan hệ với nhau thường có tên giống nhau, tuy nhiên có thể không cùng tên nhưng phải cùng kiểu dữ liệu và chứa cùng loại thông tin.
- Các Field thuộc kiểu number phải được xác lập thuộc tính FieldSize giống nhau.
  - Ngoại lệ : có thể liên kết một Field kiểu AutoNumber với một Field kiểu Number với thuộc tính FieldSize của chúng là Long Integer hay ReplicationID
  - Trong nhiều trường hợp, định nghĩa sự quan hệ bằng liên hệ mục khóa chính của một Bảng với một Field đối



chiều của một Bảng khác.

#### 4.1.3 Các loại quan hệ

- Một -Nhiều (One – To – Many): mỗi Record của một Bảng này quan hệ với nhiều Record của Bảng kia.
- Một - Một (One –To – One): mỗi Record của một Bảng này quan hệ với nhiều Record của Bảng kia.
- Nhiều - Nhiều (Many – To – Many): không thể thiết lập trực tiếp mối quan hệ nhiều - nhiều, cần có Bảng trung gian để thiết lập quan hệ một - nhiều.

## 4.2 KHAI BÁO QUAN HỆ GIỮA CÁC BẢNG

### 4.2.1 Khai báo quan hệ

Đóng tất cả các Bảng đang mở. *Không tạo hay hiệu chỉnh các quan hệ giữa các Bảng đang mở.* Muốn quy định mối quan hệ, phải mở cửa sổ Relationships.

#### 4.2.1.1 Hiển thị cửa sổ Relationships

- ◆ Đang đứng tại cửa sổ Database, Click nút Relationships

trên

Toolbar hay chọn menu Tools – Relationships để hiển thị Relationships. Cửa sổ này trình bày sự quan hệ giữa các Bảng

- ◆ Nếu là lần đầu tiên khai báo quan hệ thì cửa sổ

Relationships

trống và hộp thoại SHOW TABLE được hiển thị.

- ◆ Nếu trước đó đã được định nghĩa hay xem và điều chỉnh khai

báo

quan hệ thì Access sẽ hiển thị tính trạng cuối cùng của cửa

sổ Relationships.

- ◆ Muốn hiển thị tất cả các quan hệ trong cửa sổ

Relationships:

Click phải vào vùng trống của cửa sổ Relationships và

chọn SHOW ALL

#### 4.2.1.2 Đưa Bảng hay Query vào cửa sổ Relationships

- ◆ Mở hộp thoại SHOW TABLE (nếu chưa mở)
  - Click phải vùng trống cửa sổ Relationships và chọn SHOW TABLE
  - hay chọn menu VIEW – SHOW TABLE
  - HAY Click nút SHOW TABLE trên Toolbar.
- ◆ Chọn phiếu liên quan để hiển thị dữ liệu cần khai báo quan

hệ:

Table (hiển thị các Bảng), Queries (hiển thị các Queries) hay

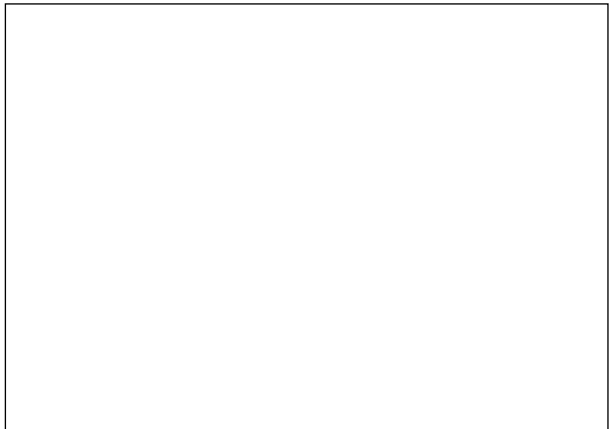
- Both (hay cả Bảng và Query)
- ◆ Đưa Bảng / Query vào cửa sổ quan hệ
    - Double Click từng Bảng / Query tham gia vào mối quan hệ.
    - Hoặc phối hợp với phím Shift hay Ctrl để chọn nhiều Bảng rồi
      - Click ADD.
      - Click Close để đóng hộp thoại SHOW TABLE.

- Hoặc nếu nhìn thấy cửa sổ Database và cửa sổ Relationships: drag Bảng / Query tham gia quan hệ từ cửa sổ Database vào cửa sổ Relationships.
- ◆ Làm ẩn Bảng / Query trong cửa sổ Relationships chỉ có nghĩa không hiển thị bảng này và các mối quan hệ của nó với các Bảng khác trong cửa sổ Relationships (không xoá mối quan hệ đã được khai báo).
  - Chọn Table / Query muốn làm ẩn
  - Chọn Hide Table từ Shortcut Menu hay từ menu Relationships hay nhấn phím Delete.

#### 4.2.1.3 Quy định mối quan hệ giữa các Bảng trong cửa sổ Relationships

- ◆ Drag Field từ Bảng chính đến Field của Bảng quan hệ. Hầu hết trường hợp là Drag từ Field mục khóa chính (Primary Key) của Bảng chính đến Field mục khóa ngoại (Foreign Key) của Bảng quan hệ.

- ◆ Hộp thoại Relationships hiển thị Field mục khóa chính thuộc Bảng chính (Primary Table), Field đối chiếu thuộc Bảng quan hệ (Related Table). Click Create để tạo mối quan hệ.



- ◆ Hộp thoại có ba phần

- Phần trên trái gồm 2 cột

- y Ghi tên Bảng / Query chính và Bảng / Query quan hệ
- y Ghi tên các Field được khai báo tham gia mối quan hệ.

- Phần dưới trái là các mục chọn để xác lập về toàn vẹn tham chiếu cho 2 Bảng quan hệ.

- y Enforce Referential Integrity: Nếu đánh dấu chọn - thiết lập tính toàn vẹn tham chiếu cho 2 Bảng quan hệ.
- y Cascade Delete Related Records

Nếu có thiết lập tính toàn vẹn tham chiếu và đánh dấu chọn mục này - cho phép Access tự động lập lại (Cascade) tác vụ xóa các Record có quan hệ (Xóa Record trong Bảng chính - xóa các Record quan hệ trong Bảng quan hệ).

## y Cascade Update Related Field

Nếu có thiết lập tính toàn vẹn tham chiếu và đánh dấu chọn mục này - cho phép Access tự động lập lại (Cascade) tác vụ cập nhật các Field có quan hệ (thay đổi trị mục khóa chính - mục khóa Field liên hệ cũng được thay đổi).

- Các nút lệnh (phần bên phải)

y Create : Ghi lưu các khai báo trong hộp thoại nghĩa là thiết lập quan hệ hai Bảng / Query

y Join Type (kiểu kết nối) : Khai báo trong hộp thoại JOIN PROPERTIES thuộc tính quan hệ đang khai báo. Mặc định là kiểu kết nối số 1 (inner join - liên kết nội)



y Create New : Khai báo mối quan hệ mới

- Nếu muốn tạo mối quan hệ của một Bảng với chính nó, thêm

Bảng đó vào cửa sổ Relationships một lần nữa. Trường hợp này xảy ra khi cần thực hiện truy tìm dữ liệu trong cùng một Bảng.

Ví dụ : trong Bảng tên Employees, một mối quan hệ được định nghĩa giữa hai Fields tên EmployeeID và ReportTo, để cho tên Field ReportTo chỉ hiển thị dữ liệu của một employee từ một EmployeeID tương ứng (Nói khác, một nhân viên chịu sự quản lý của một người và người này cũng là nhân viên Công ty).

#### 4.2.1.4 Vấn đề toàn vẹn tham chiếu

- Khái niệm

để Toàn vẹn tham chiếu là hệ thống các quy tắc được sử dụng để bảo đảm mối quan hệ giữa các mẫu tin trong các Bảng / Query quan hệ.

- Khi toàn vẹn tham chiếu được thiết lập thì không thể ngẫu nhiên xóa hay thay đổi các dữ liệu có quan hệ.

- Điều kiện thiết lập toàn vẹn tham chiếu

đầy Field tương ứng từ Bảng chính là *khóa chính hay chỉ mục*

*nhất.*

- Field quan hệ có cùng kiểu dữ liệu với Field khóa chính (trừ ngoại lệ đã nêu ở phần trên).
- Cả 2 Table đều phải thuộc cùng một CSDL (Database) Access.

- ◆ Các quy tắc cần tôn trọng khi đã lập toàn vẹn tham chiếu:  
trị Khi nhập một trị vào Field tương ứng của Table quan hệ thì  
này phải đã có trong Field mục khóa chính của Table  
chính.
- Không thể xoá mẫu tin trong Table chính nếu trong  
Table quan hệ tồn tại mẫu tin có quan hệ với mẫu tin muốn  
xóa.

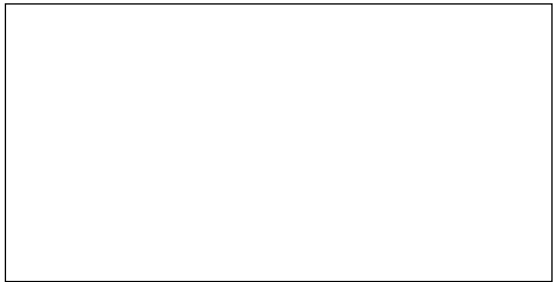
#### 4.2.2 Hiệu chỉnh mối quan hệ đã được định nghĩa

##### 4.2.2.1 Hiệu chỉnh mối quan hệ đã được định nghĩa giữa hai Table

- ◆ Mở hộp thoại Edit Relationships

Click phải vào đường biểu diễn mối quan hệ giữa hai Table

Query và chọn Edit Relationships hoặc chọn menu Relationships – Edit Relationships hoặc Double Click vào đường biểu diễn.



- ◆ Thực hiện hiệu chỉnh trong hộp thoại.

##### 4.2.2.1 Hủy bỏ một quan hệ đã khai báo

Click phải vào đường biểu diễn mối quan hệ giữa 2 Table chọn lệnh Delete trên Shortcut Menu hay chọn menu EDIT - DELETE hay Click chọn đường biểu diễn mối quan hệ cần hủy bỏ và nhấn phím DELETE.

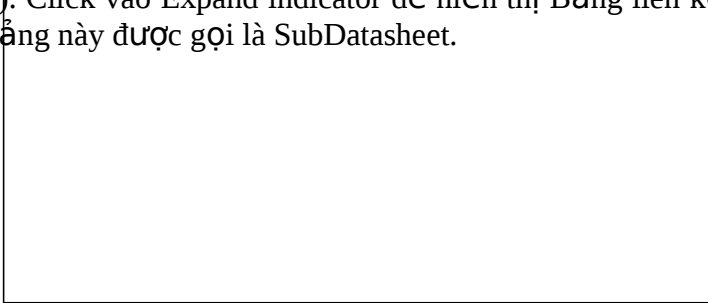
#### 4.2.3 Hiển thị các quan hệ đã định nghĩa

Muốn hiển thị các quan hệ đã khai báo

- Mở hộp thoại Relationships
- Hiển thị tất cả các quan hệ : lệnh SHOW ALL

#### 4.2.4 SubDatasheet

Khai mỗi quan hệ giữa các Bảng được khai báo, nếu mở Bảng bên 1, ACCESS hiển thị ký hiệu dấu cộng(+ : Expand indicator). Click vào Expand indicator để hiển thị Bảng liên kết bên nhiều. Bảng này được gọi là SubDatasheet.





**BÀI TẬP:**

1. Tạo tập tin CSDL để quản lý quản lý chấm công tính lương cho công nhân đặt tên là QLCN.MDB
2. Trong tập tin QLCN.MDB hãy tạo các table sau đây  
Table SANPHAM có cấu trúc: khóa chính là MASP

Field Name	Data Type	Field size	Yêu cầu
<b><u>MASP</u></b>	Text	4	Qui định nhập 1 chữ 3 số, chữ viết hoa.
TENSP	Text	30	Tên sản phẩm, chữ đầu viết hoa, còn lại thường
DONGIA	Number	Long Integer	Đơn giá sản phẩm, từ 1000 đến 50000

Dữ liệu nhập:

MASP	TENSP	DONGIA
A001	Ao sômi nữ	15000
A002	Ao sômi nam	25000
A003	Ao khoác	35000
B004	Bao tay	5000
N005	Nón	7000
Q006	Quần tây nữ	25000
Q007	Quần tây nam	35000

Table CONGNHAN có cấu trúc: khóa chính là MACN

Field Name	Data Type	Field size	Yêu cầu
<b><u>MACN</u></b>	Text	3	Mã công nhân, qui định nhập 3 số
HOTEN	Text	40	HỌ tên công nhân
PX	Number	Byte	Phân xưởng, chỉ được nhập từ 1 đến 10

Table CHAMCONG có cấu trúc: khóa chính là MACC

Field Name	Data Type	Field size	Yêu cầu
<b><u>MACC</u></b>	Number	Long Integer	Mã chấm công
NGAY	Datetime		Ngày chấm công, nhỏ hơn hoặc bằng ngày hiện tại
MACN	Text	3	Combo box chọn từ Table CONGNHAN

Table CHITIET có cấu trúc: khóa chính là MACC và MASP

Field Name	Data Type	Field size	Yêu cầu
MACC	Number	Long Integer	Mã chấm công
MASP	Text	4	Combo box chọn từ Table SANPHAM
SOSP	Number	Integer	Số sản phẩm, lớn hơn không
SOPP	Number	Integer	Số phế phẩm, nhỏ hơn số sản phẩm Chọn menu View → Properties → Validation Rule: [SOPP] < [SOSP] Validation text: Số phế phẩm phải nhỏ hơn số sản

## 3. Liên kết các Table và nhập dữ liệu sau

Table CONGNHAN

MACN	HOTEN	PX
001	Nguyen Thuy Van	1
002	Le Hai Viet	1
003	Tran Binh	2
004	Phan Van Thanh	2
005	Tran Vinh Huy	2
006	Nguyen Ngoc Han	3
007	Tran Van An	3

Table CHAMCONG

MACC	NGAY	MACN
1	14/09/05	001
2	14/09/05	002
3	14/09/05	003
4	25/09/05	004
5	25/09/05	005
6	11/10/05	003
7	11/10/05	004
8	22/10/05	005
9	22/10/05	001
10	22/10/05	006

Table CHITIET

MACC	MASP	SOSP	SOPP
1	A001	25	1
1	N005	40	0
2	A001	30	0
2	Q006	15	0
3	Q007	60	2
4	B004	80	5
4	A002	20	0
4	A003	50	2
5	Q006	70	0
6	Q007	120	4
7	A002	70	0
7	Q007	50	1
8	B004	40	0
8	Q007	20	2
8	N005	15	0
9	A003	45	4
10	N005	30	0
10	Q006	20	0

## CHƯƠNG 5

## TOÁN TỬ - BIỂU THỨC - HÀM

- Toán tử
- Biểu thức
- Hàm

## 5.1 TOÁN TỬ

5.1.1 Toán tử số học

Toán tử	Phép toán	Cú pháp	Công dụng
+	Cộng	Kết quả = số 1 + số 2	Cộng hai giá trị số. Cũng có thể dùng để nối hai chuỗi ký tự
-	Trừ	Kết quả = số 1 - số 2	Trừ hai giá trị số. Có thể dùng để biểu diễn một số âm
*	Nhân	Kết quả = số 1 * số 2	Nhân hai giá trị số
/	Chia	Kết quả = số 1 / số 2	Chia hai số có số lẻ thập phân
\	Chia nguyên	Kết quả = số 1 \ số 2	Chia hai số nguyên và lấy phần nguyên
^	Lũy thừa	Kết quả = số ^ số mũ	Lũy thừa một số. Ví dụ: 3^2 Æ 9
Mod	Phần dư phép chia	Kết quả = số 1 MOD số 2	Trả về phần dư một phép chia chẵn. Ví dụ: 23 MOD 7 Æ 2

5.1.2 Toán tử so sánh

Trả về trị True / False. Các toán hạng phải cùng kiểu dữ liệu

Toán tử	Phép toán	Cú pháp
=	So sánh bằng	Toán hạng 1 = Toán hạng 2
>	So sánh lớn hơn	Toán hạng 1 > Toán hạng 2
<	So sánh nhỏ hơn	Toán hạng 1 < Toán hạng 2
>=	So sánh lớn hơn hay bằng hơn	Toán hạng 1 >= Toán hạng 2
<=	So sánh nhỏ hơn hay bằng hơn	Toán hạng 1 <= Toán hạng 2
<>	So sánh khác nhau	Toán hạng 1 <> Toán hạng 2

### 5.1.3 Toán tử chuỗi

Toán tử	Phép toán	Cú pháp
&	Ghép chuỗi	Chuỗi 1 & Chuỗi 2 Nếu toán hạng không phải là chuỗi sẽ mặc nhiên xem như chuỗi và kết quả nhận là một chuỗi. Nếu cả hai toán hạng là Null trả trị Null Nếu chỉ một toán hạng là Null xem như một
Like	So sánh chuỗi	Biểu thức Like khuôn mẫu Được dùng các ký tự đại diện của DOS trong biểu.

- Ví dụ về toán tử Like
  - Nhập biểu thức sau vào dòng Criteria của lưới QBE khi thiết kế

Query:

Like [Enter the First Letters of the Name]&”\*” : nhập các ký tự đầu của tên

Like “P[A-F]###” : ký tự đầu là P, ký tự thứ hai trong khoảng A đến F và 3 ký tự còn lại là chữ số.

- Khi tham chiếu đến một điều khiển, hãy bao các danh hiệu điều khiển trang dấu nháy kép và các vạch thẳng đứng:

Like “[Forms]![In don dat hang]![Ma don hang]”

Nếu không thêm các dấu vạch thẳng đứng, toán tử Like trả về điều khiển thay vì trả về trị của điều khiển.

- Nếu sự so khớp thành công, trị TRUE (-1) trả về, ngược lại là trị FALSE (0). Nếu biểu thức có trị Null thì kết quả cũng là Null.

### 5.1.4 Toán tử Logic

Toán tử	Phép toán và cú pháp
And	Hội (Conjunction) Biểu thức 1 AND Biểu thức 2
Eqv	Tương đương (Equivalent) Biểu thức 1 EQV Biểu thức 2 Nếu biểu thức là Null, kết quả cũng là Null
NOT	Phủ định
Or	Phép tuyển Logic

---

Xor	Phép tuyển loại trừ
Imp	Bao hàm (Implication)

5.1.5 Toán tử khác

Toán tử	Ý nghĩa
!	Là toán tử danh hiệu. Theo sau là tên để mục do người dùng định nghĩa (như biểu mẫu, báo cáo, các điều khiển và các Field) Ví dụ: Forms![San pham]![Gia don vi]
.(dot)	Là toán tử danh hiệu. Thường được theo sau bởi một tên tính chất. Ví dụ: Forms![Customers].Visible
[]	Dấu bao tên các đề mục
Between...And...	Xác định giá trị một biểu thức có nằm trong vùng giá trị được chỉ định hay không Ví dụ: Lọc các Record có trị của Field soluong trong khoảng 50 đến 100. Tại dòng Criteria của Field này trong QBE ghi như sau: Between 50 And
In	Xác định giá trị một biểu thức có bằng với một trong nhiều trị trong một danh sách được chỉ định hay không Ví dụ: Lọc các Record có trị của Field CountryID là các mã US, UK, FR, JP. Tại dòng Criteria của Field này trong QBE ghi như sau: In("US", "UK", "JP", "FR") =IIF([Ship Region] In ("Avon", "Glos", "Sorff"), "Local", "Non Local")
Is	Dùng từ khóa NULL để xác định biểu thức có trị Null hay không =IIF(Forms![My Form]![My Control] Is Null, "No Entry", Forms![My Form]![My Control])
*?	Ký tự đại diện của DOS (thường dùng với Like) dấu * thay thế một nhóm ký tự bất kỳ dấu ? thay thế một ký tự bất kỳ

5.2 **BIỂU THỨC**5.2.1 Khái quát

Là thành phần cơ bản của nhiều phép toán của Access, dùng để chỉ định tiêu chuẩn cho các vấn tin, lọc và các macros.

Ví dụ: dùng biểu thức để tạo một Calculated

Field: ThanhTien:

[Dongia]\*[Soluong]

---

Thue: [Dongia]\*[Soluong]\*10/100



### 5.2.2 Thành phần của biểu thức

Là tổ hợp gồm các toán tử, danh hiệu, các hàm, các trị literal và các hằng

Phần tử	Ví dụ	Mô tả
Toán tử	=,<,&,And,Or,Like,...	Chỉ định một phép toán
Danh hiệu	Forms![Orders]![OrderdsId].Visible	Dành cho trị một Field, điều khiển hay tính chất
Hàm	Date, Sum, ...	Trả về trị dựa trên kết quả tính toán của hàm
Literal	100, #01/01/1999#	Biểu diễn một giá trị như đã viết
Hằng	True, False, Null, Yes, No	Biểu diễn một trị không đổi

## 5.3 HÀM

### 5.3.1 Các hàm thường dùng

- ▣ **IIF** (điều kiện, trị1, trị2) : kiểm tra điều kiện, nếu đúng trả ra trị 1, còn sai trả ra trị 2.  
Ví dụ: IIF([HTTT]=1,"Trả hết","Trả chậm")
- ▣ **SUM** (exp) : Tính tổng giá trị trong 1 Field số (dùng cho Form, Query, Report), với exp là tên Field cần tính tổng.
- ▣ **AVG** (exp) : Tính trung bình các giá trị trong 1 Field Số (dùng cho Form, Query, Report), với exp là tên Field cần tính trung bình
- ▣ **MAX** (exp), **MIN** (exp) : Lấy giá trị lớn nhất, nhỏ nhất trong 1 Field số (dùng cho Form, Query, Report), với exp là tên Field cần tính Max, Min.
- ▣ **COUNT** (exp) : Đếm các Record (khác rỗng) của Field trong exp.

### 5.3.2 Các hàm thời gian

- ▣ **DATE** () : trả về ngày tháng năm của hệ thống
- ▣ **NOW** () : trả về giờ phút giây và ngày tháng năm của hệ thống
- ▣ **TIME** () : trả về giờ phút giây của hệ thống
- ▣ **DAY** (date) : trả ra 1 giá trị từ 1 .. 31 là ngày của date
- ▣ **MONTH** (date) : trả ra 1 giá trị từ 1 .. 12 là tháng của date

- **YEAR** (date) : trả ra 1 giá trị từ 100 .. 9999 là năm của date
- **WEEKDAY** (date) : Trả ra 1 giá trị từ 1 đến 7 là Thứ của date, với 1 là Sunday, 2 là Monday, ... , 7 là Saturday
- **DATEADD** (chuỗi, số, ngày) : trả về thời điểm là một ngày nào đó sau khi đã cộng thêm khoảng thời gian vào. Trong đó chuỗi qui định khoảng thời gian cộng là loại gì (d là ngày, m là tháng, y là năm)
- **HOUR** (date) : Trả ra 1 giá trị từ 0 .. 23 là Giờ của date
- **MINUTE** (date) : Trả ra 1 giá trị từ 0 .. 59 là Phút của date
- **SECOND** (date) : Trả ra 1 giá trị từ 0 .. 59 là Giây của date

### 5.3.3 Các hàm chuỗi

- **LEN** (chuỗi) : trả ra độ dài của chuỗi (số ký tự có trong chuỗi)
- **CHR** (số) : trả ra 1 ký tự có mã ASCII bằng số  
Ví dụ: Chr(65) trả ra ký tự “A”
- **INSTR** (start, S1, S2) : Trả ra 1 số là vị trí xuất hiện đầu tiên của chuỗi S2 trong chuỗi S1, với start là vị trí bắt đầu tìm trong S1, S1 là chuỗi cần tìm, S2 là chuỗi tìm trong S1.  
Ví dụ: Instr(5, ”con gà con cần con gà”, ”con”) sẽ trả về số 8 là vị trí xuất hiện của “con” trong chuỗi.
- **LCASE** [\$] (chuỗi) : đổi chuỗi thành chuỗi mới có các ký tự là chữ thường.
- **UCASE** [\$](chuỗi) : đổi chuỗi thành chuỗi mới có các ký tự là chữ in.
- **LEFT** [\$](chuỗi, N) : Cắt N ký tự bên trái của chuỗi.
- **RIGHT** [\$](chuỗi, N) : Cắt N ký tự bên phải của chuỗi.
- **MID** [\$](chuỗi, vị trí, số) : Cắt số ký tự trong chuỗi tính từ vị trí trở đi. Nếu không có số thì cắt từ vị trí cho đến hết chuỗi.
- **LTRIM** | **RTRIM** | **TRIM** [\$] (chuỗi) : cắt các khoảng trống bên trái, bên phải, cả 2 bên của chuỗi.
- **SPACE** (N) : trả về 1 chuỗi có N khoảng trống
- **STRCOMP** (S1, S2) : trả ra giá trị -1 (true) hoặc 0 (false) phản ánh kết quả so sánh 2 chuỗi giống nhau (-1 là S1 khác S2, và 0 là S1 giống S2).

Ví dụ : a = Strcomp("abc","xyz") thì a có giá trị = -1

- ▣ **STRING** (N , ký tự) : trả ra 1 chuỗi gồm N ký tự .

#### 5.3.4 Các hàm số học

- ▣ **ABS** (số) : trả ra giá trị tuyệt đối của số.
- ▣ **ASC** (chuỗi) : Trả ra mã ASCII của ký tự đầu tiên trong chuỗi.
- ▣ **SIN , COS , ATN , TAN** (số) : Trả ra Sin , Cos , Arctan , Tan của số theo Radiant.
- ▣ **EXP**(số) : Trả ra e lũy thừa số
- ▣ **INT, FIX** (số) : Trả về phần nguyên của số. Nếu số < 0 (âm) thì FIX trả về số nguyên lớn hơn hoặc bằng số , còn INT trả về số nguyên có giá trị nhỏ hơn hoặc bằng số.
- ▣ **SQR** (số) : trả ra căn bậc 2 của số.
- ▣ **VAL** (chuỗi) : đổi chuỗi thành số

#### 5.3.5 Các hàm cơ sở dữ liệu

- ▣ **DAVG** (exp, domain, [criteria]) : tính trung bình các giá trị của các record trong *domain* thỏa theo *criteria*.

exp : chuỗi chứa tên field cần tính

domain : chuỗi chứa tên table hoặc query cần

tính criteria : chuỗi chứa 1 điều kiện.

Ví dụ: Davg("[SOLG]", "[CT BAN LE]", "[DGL] >=1000") là tính trung bình Field SOLG trên bảng CT BAN LE theo điều kiện có DGL >= 1000.

- ▣ **DMAX , DMIN** (exp , domain , [criteria]) : lấy giá trị lớn nhất, nhỏ nhất trong exp của domain.

Ví dụ: Dmax("[SOLG]", "[CT BAN SI]", "[MAHG] = '001'") tức là lấy giá trị lớn nhất trong field SOLG của bảng CT BAN SI theo điều kiện MAHG là 001.

- ▣ **DFIRST, DLAST** (exp , domain , [criteria]) : lấy giá trị đầu tiên, cuối cùng trong exp của domain thỏa theo criteria.
- ▣ **DCOUNT** (exp , domain , [criteria]) : đếm các record của exp trong domain thỏa criteria.
- ▣ **DSUM** (exp , domain , [criteria]) : tính tổng các record của exp trong domain thỏa criteria.
- ▣ **DLOOKUP** (exp, domain, [criteria]) : Trả về giá trị của exp trong domain thỏa criteria.

**BÀI TẬP:**

1. Cho biết kết quả của các hàm sau đây:

- LEN("Microsoft-Access")
- LEFT("Tin học",3)
- RIGHT("Cao Thắng",5)
- MID("Microsoft-Access",5,3)
- DATE()
- MONTH(DATE())
- YEAR(DATE())
- DAY(DATE())
- WEEKDAY(DATE())
- SQRT(5)
- INT(17.5)
- FIX(17.5)
- INT(-20.4)

2. Cho biết ý nghĩa của các biểu thức sau, với [HO], [TEN], [NGAYSINH], [PHAINU], [DIEM] là các field trong table HOCSINH:

- YEAR(DATE()) – YEAR([NGAYSINH])
- UCASE([HO] & " " & [TEN])
- IIF(Weekday(Date())=1,"Chủ nhật","Thứ " & Weekday(Date()))
- IIF([PHAINU]=Yes, "Nữ", "Nam")
- IIF([DIEM]<5,"Yếu",IIF([DIEM]<7,"TB",IIF([DIEM]<9,"Khá", "Giỏi")))
- IIF(MONTH([NGAYSINH])=MONTH(DATE()),"Sinh nhật"," ")

## CHƯƠNG 6

**TRUY VẤN (QUERY)**

- Khái quát về Query
- Thiết kế query
- Các chế độ hiển thị Query
- Thao tác với Query
- Query Wizard

**6.1 KHÁI QUÁT VỀ QUERY****6.1.1 Query là gì**

- ▣ Query là công cụ của Access dùng để khai thác dữ liệu trên một hay nhiều Bảng / Query khác bằng cách thực hiện các yêu cầu truy vấn.
- ▣ Người sử dụng nêu các câu hỏi (yêu cầu truy vấn) và Access sẽ thực hiện việc truy tìm dữ liệu theo yêu cầu trên những Bảng / Query chỉ định và tổng hợp dữ liệu để cho các thông tin cần thiết trên một Bảng gọi là Bảng dữ liệu động (Dynamic Set hay DynaSet)
- ▣ Query cũng được dùng như một nguồn dữ liệu cho các công cụ khác như FORM, REPORT hay QUERY khác.
- ▣ Muốn thực hiện Query, phải có Database với một hay nhiều Bảng.
- ▣ Nếu thực hiện hiệu chỉnh trên Query thì số liệu trên các Bảng cũng được sửa đổi theo.

**6.1.2 Các trường hợp sử dụng Query**

Thường sử dụng Query trong các trường hợp sau:

- ▣ Lựa chọn các Field cần thiết (không cần bao gồm tất cả Field của Bảng)
- ▣ Lựa chọn những tin theo tiêu chuẩn nào đó. Xếp thứ tự các tin
- ▣ Tham khảo dữ liệu của nhiều Bảng khác nhau.
- ▣ Thực hiện các phép tính.
- ▣ Tạo dữ liệu cơ sở cho Form, Report hay Query khác.

**6.1.3 Các loại Query**

Có thể tạm phân loại các Query thành các loại như sau :

Select

Query, Action Query, Crosstab Query và một số loại khác.

### 6.1.3.1 Select Query (Lựa chọn)

Sử dụng khá phổ biến. Loại Query cho phép

- Vắn tin trên nhiều Bảng dữ liệu
- Tạo ra Field tính toán (Calculated Field)
- Kết nhóm dữ liệu
- Chọn Field cần hiển thị hoặc không hiển thị dữ liệu.

### 6.1.3.2 CrossTab Query (Tham khảo chéo)

- Dùng để tổng hợp dữ liệu theo chủng loại (tổng số, số lượng, trị trung bình,...) theo dạng bảng tính (có tiêu đề cột dòng và kết quả tổng hợp).
- Loại này tỏ ra hữu ích trong việc phân tích dữ liệu và tạo các biểu đồ dựa trên tổng giá trị của các Field số của nhiều tin.

### 6.1.3.3 Action Query (hành động)

Loại Query này cho phép thực hiện những thay đổi với nhiều tin trong cùng một tác vụ, tạo các Bảng mới, hiệu chỉnh hay xóa tin. Tùy theo công dụng, các Action Query gồm:

- Delete Query: Xóa nhóm tin từ một hay nhiều Bảng.
- Update Query: Cập nhật nhóm tin cho một / nhiều Bảng.
- Append Query: Nối thêm dữ liệu từ các tin của một hay nhiều

Bảng vào phần cuối của một Bảng.

- Make-Table Query: tạo một Bảng từ một phần hay tất cả

liệu trong một hay nhiều Bảng khác.

### 6.1.4 Query có sử dụng tham số (Parameters)

- Nếu có một Query dùng vào bất kỳ mục đích nào (như Select, Update, Delete,...) mà khi thực cần phải nhận các trị tham số để tạo ra kết quả thì Query đó có sử dụng tham số.
- Các tham số dùng cho Parameter Query được xem như những trị chưa thể xác định vào lúc thiết kế và được ghi dưới dạng một tên biến (khai báo tại dòng Criteria của Field liên quan).

Ví dụ: muốn tạo báo cáo trong khoảng thời gian nào đó, ghi ở

dòng Criteria dưới cột NGAY của Bảng Query như sau

: Between [so lieu tu ngay :] And [Den ngay :]

Các tổ hợp [so lieu tu ngay :] và [Den ngay :] là những tên biến dùng tạo tham số cho Query.

### 6.1.5 Một số Query đặc biệt khác

- Union Query: Nhóm nối đuôi các dữ liệu từ hai hay nhiều Bảng thành một danh sách để xem chung.



- PassThrough Query: Dùng gửi lệnh trực tiếp đến một CSDL hệ ngôn ngữ vấn tin như Microsoft SQL Server hay Sybase SQL Server.
- Data-Definition Query: Dùng ngôn ngữ SQL để tạo hay sửa đổi cấu trúc một Bảng trong Database.
- SubQuery: Gồm một phát biểu SQL SELECT ở trong một Select Query hay Action Query khác.
- Find Duplicate Query: Tìm trong Bảng các mẫu tin trùng lặp trên giá trị của một hay nhiều Field.
- Find UnMatched Query: Tìm các mẫu tin trong một Bảng dữ liệu không có quan hệ với một mẫu tin trong Bảng quan hệ.

## 6.2 THIẾT KẾ QUERY

Có hai cách thiết kế Query

### 6.2.1 Dùng Query Wizard

- Query Wizard dùng thiết kế nhanh một Query.
- Phương pháp này có hạn chế vì chỉ thiết kế được một số Query.
- Tùy theo loại Query cần thiết kế (Simple Select, CrosstTab,...) thực hiện theo các bước hướng dẫn của WIZARD.

### 6.2.2 Thiết kế Query với Design View

Đây là trường hợp Tự thiết kế Query mà không dùng Wizard (còn gọi là thiết kế From Scratch) thông qua cửa sổ Design View. Các bước khái quát để tự tạo một Query:

- Khởi động Design View
- Chọn các Bảng, Query tham gia yêu cầu truy vấn.
- Khai báo mối liên hệ giữa các Bảng (nếu trước đó chưa khai báo), khai báo các tiêu chuẩn lọc (nếu cần)
- Chạy thử và ghi lưu.

---

**GHI CHÚ** : Có thể phối hợp hai phương pháp trên bằng cách tạo Query với WIZARD trước rồi hiệu chỉnh lại bằng Design View.

---

## 6.3 CÁC CHẾ ĐỘ HIỂN THỊ QUERY

Cửa sổ Query có 4 chế độ hiển thị.

### 6.3.1 Datasheet View

- Là dạng hiển thị chuẩn khi chạy Query (như dạng hiển thị của Bảng) gồm dòng đầu tiên là tên các Field và các dòng còn lại là các mẫu tin.
- Thao tác

- Trong cửa sổ Database, chọn Query cần hiển thị rồi Click

Open.

- Hoặc nếu đang hiển thị Query ở dạng Design View hay SQL View thì chọn lệnh View - Datasheet View trên Menubar hay trên Shortcut Menu.

- ▣ Trong khi thiết kế Query, có thể dùng Datasheet View để xem thử kết quả thiết kế (Chọn lệnh View - Datasheet View trên Menubar hay trên Shortcut Menu).

### 6.3.2 Design View

Hiển thị Query theo dạng thiết kế. Dùng để thiết kế hay hiệu chỉnh Query.

#### 6.3.2.1 Thao tác

- ◆ Trong cửa sổ Database, chọn Query cần hiển thị và Click

Design View trên Toolbar của cửa sổ hay trên Shortcut Menu.

- ◆ Hoặc nếu đang hiển thị Query ở dạng Datasheet View hay

SQL View thì chọn lệnh View - Design View trên Menubar hay trên Shortcut Menu.

#### 6.3.2.2 Cửa sổ Design View

Có hai khung gồm

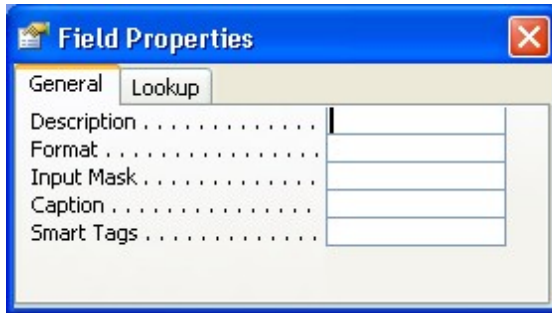
Khung chứa dữ liệu nguồn

Khung lưới QBE

- ◆ Khung trên : Khung chứa dữ liệu nguồn. Hiển thị các Bảng tham gia Query và mối quan hệ giữa chúng (nếu có).

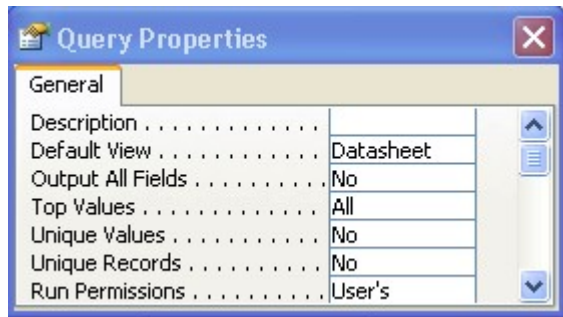
- 
- ◆ Khung dưới : gọi là lưới QBE (Query By Example) dùng để nhập yêu cầu vấn tin cho Query
- 6.3.2.3 Thuộc tính Field của Query
- ◆ Khai báo thuộc tính Field trong cửa sổ thuộc tính của Field.
  - ◆ Mở cửa sổ thuộc tính của Field

- Click phải vào Field liên quan trên lưới QBE và chọn Properties.
- Hay đứng tại Field liên quan và chọn menu View – Properties.
- Hay đứng tại Field liên quan và nhấn ALT+ENTER
- ◆ Các thuộc tính
  - Description: Chứa chuỗi ký tự mô tả Field
  - Format: Quy định dạng thức hiển thị nội dung Field.
  - Input Mask: Quy định dạng thức nhập liệu vào Field
  - Caption: Chứa chuỗi ký tự làm nhãn cho Field.



#### 6.3.2.4 Thuộc tính của Query

- ◆ Khai báo các thuộc tính cho Query trong cửa sổ thuộc tính của Query.
- ◆ Mở cửa sổ thuộc tính của Query
  - Không đứng tại Field nào của Query: chọn menu View và chọn Properties hay nhấn ALT+ENTER
  - Hay Click phải vào vùng không thuộc Field nào cả và chọn Properties trên Shortcut Menu.
  - ◆ Các thuộc tính
    - Description : Mô tả Query
    - Default View : Chế độ hiển thị mặc định
    - Output All Fields : Hiển thị tất cả Field
    - Top Values : Hiển thị các trị đầu tiên
    - Unique Values : Hiển thị tin mà tất cả trị các Field của Query là duy nhất



- Unique records : Không hiển thị tin trùng lặp

### 6.3.3 SQL View

- ▢ Hiển thị thiết kế Query dưới dạng dòng lệnh theo ngôn ngữ SQL. Các dòng lệnh này do Access tạo ra khi thiết kế Query hay do người dùng khai báo.
- ▢ Thao tác: Mở Query trong chế độ Datasheet View hay Design View và chọn lệnh View-SQL View trên Menubar hay trên Shortcut Menu.

### 6.3.4 Print PreView

- ▢ Là chế độ xem trước khi in (Kết quả hiển thị giống như khi in).  
Chợ Query cần hiển thị trong cửa sổ Database hay mở Query theo chế độ Datasheet View.
- ▢ Chọn lệnh Print Preview từ menu VIEW hay Shortcut Menu.

## 6.4 THAO TÁC VỚI QUERY

### 6.4.1 Ghi lưu với Query

- ▢ Như mọi đối tượng khác, Query sau khi hoàn tất thiết kế hay hiệu chỉnh phải được ghi lưu.
- ▢ Ghi lưu : FILE - SAVE hay Ctrl+S hay Click biểu tượng ghi lưu và đặt tên cho Query trong hộp thoại Save As.
- ▢ Tương tự như Bảng, Query là một thành phần của CSDL nên sau khi được thiết kế và ghi lưu, Query được đặt trong ngăn QUERIES của Database Container.

### 6.4.2 Thao tác với Query

- ▢ Có thể thực hiện tác vụ sao chép, sửa chữa, xóa ...v.v. trên  
Query (tương tự thao tác với Bảng).
- ▢ Sửa đổi thiết kế trong QUERY : Chọn tên Query cần hiệu chỉnh rồi click biểu tượng Design trên cửa sổ Database.
- ▢ Thi hành một Query : Chọn tên Query liên quan - Open (hay Double Click) hay trong cửa sổ Design View, click biểu tượng RUN trên thanh dụng cụ
- ▢ Query có thể được kích hoạt thông qua lệnh Macro.
- ▢ Trong môi trường nhiều người sử dụng
  - Muốn xem Query với kết quả mới nhất : Shift + F9 (Record - Refresh).
  - Nếu đang hiệu chỉnh trong Query Design lại có người khác hiệu chỉnh Query Design (trong môi trường nhiều người dùng), Access sẽ thông báo cho biết và chọn một trong hai cách : hoặc chấp nhận những sửa đổi ấy hoặc lưu thiết kế

với bảng tên mới.

## 6.5 QUERY WIZARD

### 6.5.1 Giới thiệu

Query Wizard là công cụ giúp tiến hành nhanh việc thiết kế một

Query và thường được đổi với những loại Query sau

- Simple Query : Sselect Query đơn giản
- Crosstab Query : Vấn tin tham khảo chéo.
- Find Duplicate Query : Vấn tin tìm tin trùng lặp.
- Find Unmached Query : Vấn tin tìm các tin trong một Bảng dữ liệu không có quan hệ với một Record trang Bảng quan hệ (Ví dụ tìm những khách hàng chưa có đơn đặt hàng)

### 6.5.2 Các bước chung tạo Query bằng Query Wizard

- Bước 1: Mở CSDL liên quan.
- Bước 2: Trong cửa sổ Database
  - Chọn loại đối tượng QUERIES
  - Hoặc chọn menu View - Database Object - Queries
- Bước 3: Mở hộp thoại New Query.
  - Click nút New trong Database Container
  - Hoặc chọn menu INSERT - Query
  - Hoặc Click nút Dropdown cạnh biểu tượng New Object : <đối tượng> trên Toolbar - chọn Query trong Menu Dropdown.
- Bước 4: Chọn loại Query muốn tạo trong hộp thoại New Query.

### 6.5.3 Tạo Query với Simple Query Wizard

#### 6.5.3.1 Khởi động Simple Query Wizard

Chọn mục Simple Query Wizard trong hộp thoại New Query

#### 6.5.3.2 Các bước của Wizard

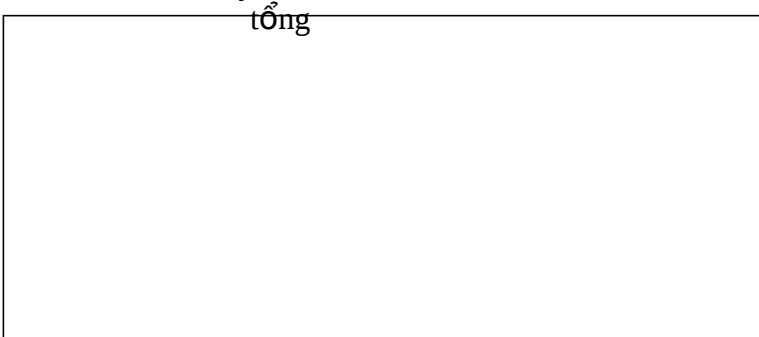
- ◆ Chọn Bảng / Query và Field tham gia Query



- Chọn Bảng hay Query trong khung Table / Queries



- Chọn các Field tham gia trong khung Available Field và chuyển sang (click nút > bên cạnh) khung Selected Field.
  - Nếu chọn tất cả Field thì click nút >>
  - Click nút < hay << để chuyển các Field trong khung Selected Field trở lại khung Available Fields.
  - Thực hiện tương tự trên với các Bảng / Query khác nếu có nhiều Bảng / Query tham gia.
  - Thực hiện xong, click NEXT để qua bước tiếp theo.
  - Nếu 2 Bảng chưa khai báo quan hệ nên Access hiển thị thông báo : Click OK và thực hiện quy trình khai báo quan hệ giữa các Bảng tham gia Query.
- ♦ Chọn loại Query chi tiết hay Query tính



Chọn một trong hai mục

- Nếu chọn DETAIL : Các mẫu tin thỏa điều kiện đều được hiển thị.
  - Nếu chọn SUMMARY
    - y Click nút Summary Options để mở hộp thoại Summary Options
    - y Chỉ định phép toán cho Summary (SUM, AVG, MIN, MAX).
  - Thực hiện xong bấm NEXT.
- ♦ Đặt tiêu đề cho Query chọn cách kết thúc
- Đặt TITLE cho Query trong khung *What title do you want for your query*
  - Và chọn một trong hai mục để kết thúc quy trình tạo Query
    - y MỞ QUERY khi WIZARD tạo xong (Open the Query to view information): Query được hiển thị theo dạng DataSheet View.
    - y hay hiệu chỉnh cấu trúc Query (Modify this Query Design): cửa sổ Design View được hiển thị để thực hiện các hiệu chỉnh.

**BÀI TẬP:**

1. Mở tập tin VPP.MDB ra và chọn trang Queries, thực hiện các truy vấn sau:
  - Liệt kê các khách hàng ở Q1. { **like\*q1** }
  - Liệt kê các khách hàng không có số điện thoại { **is null** }.
  - Liệt kê các mặt hàng có đơn giá dưới 100000 { **<10000** }
  - Liệt kê các mặt hàng bút có đơn vị tính là cây. { **like cay , like but** }
  - Liệt kê các mặt hàng đã được đặt trong tháng 1/2006 { **between 01/01/2006 and 31/01/2006 - like\*/01/2006** }
  - Liệt kê các hóa đơn đã đặt và giao trong vòng 3 ngày { **[ngaydat] - [ngay giao]<=3** }
2. Mở tập tin QLCT.MDB ra và chọn trang Queries, thực hiện các truy vấn sau:
  - Liệt kê các công trình có tên bắt đầu là chữ N và khởi công năm 2006. { **like N\* - like\*/\*/2006** }
  - Liệt kê các vật tư có chữ A trong mã vật tư { **like \*A\*** }
  - Liệt kê các công trình đã khởi công từ 3 năm trở lên { **(date()- [ngayKC])/365>=3** }
  - Liệt kê MAVT, TENVT, SOLUONG, DONGIA mà có giá trị từ 50000 đến 100000.
3. Mở tập tin QLCN.MDB ra và chọn trang Queries, thực hiện các truy vấn sau:
  - Liệt kê các công nhân có họ “Tran” ở phân xưởng 2.
  - Liệt kê các sản phẩm không phải là áo hoặc quần.
  - Liệt kê các công nhân có chữ lót là “Van” nhưng không phải tên “Van”
  - Liệt kê các sản phẩm sắp theo đơn giá giảm dần mà từ 20000 trở lên.

## CHƯƠNG 7

**SELECT QUERY**

- Thiết kế Select Query
- Khai báo quan hệ
- Tiêu chuẩn lựa chọn
- Calculated Field
- Query Tổng cộng – Nhóm
- Summary Query
- Các vấn đề bổ sung

**7.1 THIẾT KẾ SELECT QUERY**

Select Query được dùng khá nhiều để lọc, tổng hợp dữ liệu từ một đến nhiều Bảng / Query khác và làm dữ liệu nguồn cho Biểu mẫu, Báo biểu. Sử dụng Query Design View để tạo Select Query như sau:

**7.1.1 Mở cửa sổ Design View**

- Mở CSDL liên quan - mở Phiếu Query - chọn mục Design View trong hộp thoại New Query.
- Cửa sổ QUERY DESIGN VIEW được hiển thị với tên trên Title gồm <TÊN QUERY>:SELECT QUERY (Tên Query mặc định là QUERY#).

**7.1.2 Cửa sổ QUERY DESIGN VIEW**

Cửa sổ thiết kế Query có 2 khung: *Khung trên* (chứa các Bảng dữ kiện được chọn tham gia vào Query) và *Khung dưới* (Lưới QBE (Query By Example) chứa các Field được đưa vào Query). Các dòng của lưới QBE gồm:

- Dòng Field: ghi tên Field được đưa vào lưới
- Dòng Table: ghi tên Bảng/Query của Field chỉ định tại dòng Field  
(đánh dấu chọn mục Table Names trên Menu VIEW)
- Dòng Sort: dùng để sắp xếp các mẫu tin
- Dòng Show: chứa Check Box xác định hiển thị / không hiển thị  
Field liên quan
- Dòng Criteria: dùng khai báo tiêu chuẩn lọc
- Dòng OR: dùng khai báo bổ sung điều kiện lọc
- Và tùy loại Query, một số dòng khác sẽ được hiển thị.

**7.1.3 Đưa dữ liệu nguồn vào cửa sổ thiết kế**



- Từ hộp thoại SHOW TABLE, chọn Tên Bảng/Query liên quan và click ADD hoặc double click tên Bảng/Query để đưa thêm vào cửa sổ thiết kế.
- Chỉ đưa vào cửa sổ thiết kế các Field của Bảng/Query cần sử

dụng

- ▣ Đưa Bảng dữ kiện ra khỏi Query
  - Chọn biểu tượng Bảng muốn đưa ra khỏi cửa sổ Design ở phần trên cửa sổ thiết kế rồi nhấn phím Delete
  - Hay chọn Remove Table trên Shortcut Menu
- ▣ Khai báo quan hệ giữa các Bảng/Query
  - Khi đưa vào cửa sổ thiết kế hơn một Bảng/Query, quan hệ giữa các dữ liệu nguồn sẽ tự động hiển thị nếu quan hệ đã được tự khai báo trước đó.
    - Ngược lại có thể khai báo quan hệ ngay trong cửa sổ Query.

Cách khai báo cũng tương tự như khai báo khi thiết kế Bảng.

#### 7.1.4 Đưa các Field của các Bảng/Queries tham gia Query vào lưới QBE

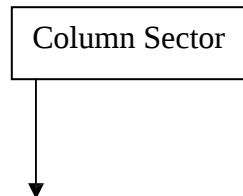
##### 7.1.4.1 Đưa Field dữ liệu vào lưới QBE

- ◆ Khái quát
  - đều Field nào của Bảng dữ kiện cần sử dụng trong Query đưa vào lưới QBE
  - Một Field được đưa vào lưới có thể không được hiển thị.
- ◆ Cách chung
  - Lần lượt chọn Field từ các Bảng ở khung trên cửa sổ thiết kế Query và kéo vào lưới QBE hoặc double click tên Field.
- ◆ Thao tác nhanh
  - Đưa tất cả Field vào lưới
    - y Chọn tất cả Field của Bảng dữ kiện: double click tiêu đề biểu tượng Bảng/Query
    - y Rê vào lưới QBE: tất cả Field sẽ được đưa vào lưới
  - Đưa một Field vào lưới
    - y Double click một Field, Field ấy được đưa vào lưới
    - y Hay rê thả vào lưới
    - y Hay tại cột tương ứng, mở và chọn tên Bảng/Query ở

- dòng Table, chọn tên Field ở dòng Field
- Đưa nhiều Field vào lưới cùng lúc
  - y Chọn các Field muốn đưa vào lưới
  - y rê thả vào lưới
- rê /double click dấu hoa thị (\*)
  - y Biểu thị chọn tất cả Field nhưng trong lưới chỉ có một cột chứa dấu hoa thị và tên Bảng (Ví dụ: HOCSINH.\*)

- y Dùng phương pháp chọn dấu hoa thị có lợi điểm là không cần sửa đổi Query khi thay đổi Field của Bảng cơ sở (kể cả thêm Field mới, Query tự động được cập nhật).

#### 7.1.4.2 Sắp xếp thứ tự, chèn, xóa, thay đổi bề rộng các Field trong QUERY



- ◆ Thay đổi vị trí Field: click Column Selector để chọn Field muốn thay đổi vị trí rồi kéo đến vị trí mới.
- ◆ Thêm Field mới: double click Field muốn thêm trên Bảng thích ứng rồi điều chỉnh vị trí (như đã nêu trên)
- ◆ Xóa Field: chọn Field cần xóa - nhấn phím Delete hay chọn lệnh Delete từ Menu Edit.
- ◆ Thay đổi bề rộng Field: kéo hay double click bờ phải Column Selector của Field tương ứng.

#### 7.1.4.3 Khai báo thứ tự sắp xếp các tin

Đứng trong dòng SORT của Field cần xếp thứ tự - click Dropdown và chọn cách sắp xếp (Not Sorted: không xếp thứ tự, Ascending: tăng dần, Descending: giảm dần)

#### 7.1.5 Chỉ định Field được hiển thị

Đánh dấu chọn hay bỏ chọn Checkbox tại dòng Show của Field liên quan (Đánh dấu chọn: hiển thị Field)

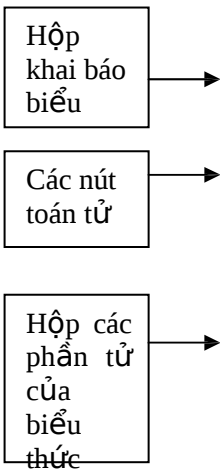
#### 7.1.6 Khai báo tiêu chuẩn vấn tin (Criteria)

##### 7.1.6.1 Khái quát

- ♦ Khai báo tiêu chuẩn vấn tin để chỉ định phạm vi hiển thị các mẫu tin. Chỉ những mẫu tin thỏa các tiêu chuẩn chỉ định mới được hiển thị.
- ♦ Tiêu chuẩn vấn tin được khai báo trên dòng Criteria của Field tương ứng trong lưới QBE.
- ♦ Có hai cách khai báo tiêu chuẩn vấn tin: nhập trực tiếp vào dòng Criteria hay dùng Expression Builder.

7.1.6.2 Khai báo bằng Expression Builder

- ♦ Mở hộp thoại Expression Builder
  - Đứng trong dòng Criteria của Field cần khai báo
  - Click biểu tượng Build trên Toolbar hay lệnh Build trên Shortcut Menu
  - Nếu CSDL có nhiều Builder, Access hiển thị hộp thoại CHOOSE BUILDER để chọn Expression Builder.
- ♦ Hộp thoại Expression Builder
  - Hộp khai báo biểu thức (trên)
  - Các nút toán tử (giữa)
  - Hộp các phần tử của biểu thức (dưới) gồm 3 khung



- y Khung trái: chứa các Bảng, Folders chứa các hàm, các biểu thức thường dùng
- y Khung giữa: liệt kê các phần tử hạng mục của thành phần đang chọn bên trái.
- y Khung phải: liệt kê các trị của mục chọn ở trái và



giữa. muốn đưa các trị ở khung này vào hộp biểu thức:  
double click trị liên quan.

7.1.6.3 Trực tiếp khai báo tiêu chuẩn trong QBE

- ◆ Nhập trực tiếp biểu thức tiêu chuẩn vào dòng Criteria.
- ◆ Nhấn Shift+F2 để mở cửa sổ Zoom cho dễ nhập.
- ◆ Nhập biểu thức tiêu chuẩn
  - *Dữ kiện kiểu số học* (Number, Currency, AutoNumber):  
nhập bình thường. Không được sử dụng dấu phân cách hàng ngàn và dấu tiền tệ.
  - *Dữ kiện kiểu ngày/giờ*: phải theo Control Panel và có thể không dùng ký tự rào # (Ví dụ: <16/1/94 hay <#16/1/94# hay 16 January 1994)
  - *Dữ kiện kiểu Text*: bao/không bao dấu nháy kép cũng được.
  - *Dữ kiện kiểu Logic YES/NO*: nếu là YES: nhập Yes, TRUE, ON hoặc -1. Nếu là NO: nhập NO, FALSE, OFF hoặc 0

### 7.1.7 Khai báo thuộc tính

Tùy nhu cầu thiết kế, có thể khai báo thuộc tính cho Field hay cho Query.

### 7.1.8 Kết thúc

- Trong quá trình thiết kế có thể chạy thử để xem kết quả (click biểu tượng RUN trên Toolbar hay chọn menu Query và chọn RUN hay click biểu tượng Datasheet View trên Toolbar)
- Kết thúc: phải ghi lưu Query

## 7.2 KHAI BÁO QUAN HỆ TRONG QUERY

### 7.2.1 Khái quát

- Query lấy dữ liệu từ nhiều Bảng/Query khác và có thể quan hệ giữa các dữ liệu nguồn chưa được khai báo.
- Khi các Bảng chưa được khai báo quan hệ lúc thiết kế cấu trúc, Access cũng tự thiết lập liên hệ giữa chúng khi thiết kế Query nếu:
  - Giữa chúng có một Field mà tên và kiểu dữ liệu hoàn toàn giống nhau
  - Và nếu một trong hai Field là khóa chính (Primary Key)  
Quan hệ này không có tính toàn vẹn tham chiếu (đường nối không có dấu 1- ∞)
- Nếu Access không thể tự động thiết lập quan hệ thì người sử dụng phải khai báo trong cửa sổ thiết kế Query.

### 7.2.2 Thiết lập quan hệ giữa các Bảng/Query khi thiết kế Query



Rê Field thuộc Bảng/Query thứ nhất vào Field liên kết tương ứng trong Bảng/Query thứ hai. Quan hệ được thiết lập không có tính toàn vẹn tham chiếu (đường nối không có dấu 1-∞)

□ Các cách liên kết

- ◆ Liên kết chặt hoặc tương đương (Inner-Join hay Equi-Join)

Là kiểu liên kết trong đó chỉ đưa vào Query những mẫu tin của 2 Bảng/Query có trị vùng liên kết bằng nhau.

- ◆ Liên kết không chặt (Outer

Join)

Là kiểu liên kết trong đó những mẫu tin của Bảng/Query thứ nhất được đưa vào Query ngay cả khi nó không có mẫu tin đối chiếu ở Bảng/Query thứ hai và những mẫu tin của Bảng/Query thứ hai chỉ được đưa vào Query khi chúng có mẫu tin đối chiếu ở Bảng thứ nhất. Kiểu liên kết này có 2 loại.

□ LEFT

OUTER JOIN

Là kiểu liên kết trong đó tất cả mẫu tin thuộc Bảng/Query được khai báo ở phần bên trái (LEFT JOIN) được đưa vào Query. Trong cửa sổ Query, kiểu liên kết này được biểu thị bằng đường nối mũi tên ở bên trái.

□ RIGHT

OUTER JOIN

Là kiểu liên kết trong đó tất cả mẫu tin thuộc Bảng/Query được khai báo ở phần bên phải (RIGHT JOIN) được đưa vào Query. Trong cửa sổ Query, kiểu liên kết này được biểu thị bằng đường nối mũi tên về bên phải.

- ◆ Tự liên kết (Self-

Join)

Là kiểu liên kết một Bảng với chính nó (đưa Bảng này vào Query hai lần). Một mẫu tin trong Bảng được liên kết với những mẫu tin khác của chính Bảng đó. Với liên kết này, tất cả những Record trong Bảng mà Field liên kết có trị bằng nhau đều được đưa vào Query.

### 7.2.3 Thêm Bảng trung gian vào Query

---

Khi đưa vào Query hai Bảng/Query nhưng chúng lại không có Field liên quan để tạo mối liên kết, trong trường hợp này, phải đưa thêm Bảng trung gian vào Query rồi thiết lập mối liên kết giữa chúng.

#### 7.2.4 Tìm kiếm thông tin tự động (AutoLookup)

- ▢ Giả sử một khách hàng quen thuộc (đã đăng ký) mua hàng của một công ty, khi nhập dữ liệu vào, không cần nhập chi tiết về khách hàng vì thông tin này đã được lưu trong tập tin KHACHHANG.
- ▢ Với cách thiết kế Query trong đó sử dụng công cụ AutoLookup, chỉ cần nhập mã số khách hàng, các chi tiết liên quan đến khách hàng này sẽ được Access tìm kiếm.

- Tác vụ AutoLookup chỉ làm việc với những Query trong đó hai Bảng cơ sở có quan hệ kiểu One-To-Many và Field liên kết thuộc Bảng dữ kiện phía One là *Field chỉ mục duy nhất hay Primary key*.

### 7.3 SỬ DỤNG TIÊU CHUẨN LỰA CHỌN TRONG QUERY

#### 7.3.1 Sử dụng ký tự đại diện

- Dùng ký tự đại diện để trích chọn các Field có trị là chuỗi ký tự hay Date/Time.
- Tuy nhiên, đối với dữ liệu Date/Time thì nên dùng hàm DatePart trong tiêu chuẩn thì tốc độ xử lý sẽ nhanh hơn.
- Vài ví dụ

Biểu thức tiêu chuẩn	Access hiển thị	Tìm những trị
Sm?th	Like “Sm?th”	Smith, Smyth ...
L*ng	Like “L*ng”	Ling, Levering, Laka Travel and Trading, ...
*th	Like “*th”	128 <sup>th</sup> , Perth, ...
*/10/93	Like “*/10/93”	tất cả ngày của tháng 10/93

#### 7.3.2 Trích những mẫu tin không khớp với một trị khai báo

- Dùng toán tử NOT trong biểu thức tiêu chuẩn.
- Ví dụ:  
 NOT “H\*” hay NOT LIKE “H\*” : những mẫu tin có ký tự đầu không phải là H  
 NOT 2 : những mẫu tin có trị không phải là 2

#### 7.3.3 Biểu thức tiêu chuẩn khai báo nhiều điều kiện

- *Nhiều điều kiện đối với một Field*  
 Sử dụng các toán tử And, Or để thiết lập nhiều điều kiện cho một Field.  
 Ví dụ: Like “The\*” And Like “Box\*” hay  
 “Davolio” Or “Buchanan”
- *Nhiều điều kiện đối với nhiều Field*  
 Lần lượt khai báo các tiêu chuẩn cho các Field và theo tiêu chuẩn sau:  
 - Các tiêu chuẩn trên cùng dòng: điều kiện AND  
 - Các tiêu chuẩn trên khác dòng: điều kiện OR

#### 7.3.4 Trích mẫu tin có hoặc không có dữ kiện

☐ Dùng biểu thức NOT NULL hay IS NOT NULL cho Criteria để trích mẫu tin của Field có dữ kiện.

☐ Dùng biểu thức NULL hay IS NULL cho Criteria để trích mẫu tin của Field không có dữ kiện.

7.3.5 Trích mẫu tin ứng với ngày hiện hành của máy hay 1 danh sách

☐ Nhập hàm DATE() vào dòng Criteria để trích những Record ứng với ngày hiện hành của máy tính.

☐ Dùng toán tử IN (tên1, tên2,...) để trích một trị trong danh sách.

Danh sách được ghi trong cặp ngoặc đơn của toán tử IN và phân cách nhau bởi dấu phẩy.

Ví dụ: In (“France”, “Germany”, “Japan”)

7.3.6 Khai báo các phép tính trong biểu thức tiêu chuẩn

☐ Có thể khai báo các phép tính trong Field tiêu chuẩn, đặc biệt là với Field Date/Time. Ví dụ: Between DATE() And DATE()-5

☐ Trong biểu thức tiêu chuẩn, muốn tham chiếu đến trị của một

Field khác thì rào tên Field bằng dấu ngoặc vuông

[]. Ví dụ: <=[ORDER DATE] + 30

7.3.7 Ví dụ minh họa một số cách khai báo

☐ Dùng trị TEXT là tiêu chuẩn

Field	Biểu thức	Diễn giải
ShipCity	“London” “London”Or “Paris”	Hiển thị đơn đặt hàng gửi đến “London” Dùng toán tử Or hiển thị đơn đặt hàng gửi đến “London” hay “Paris”
ShipeedDate	#2/2/07# Between #1/5/07# And #1/10/07#	Hiển thị đơn đặt hàng ngày 2/2/07 Dùng toán tử Between...And... để hiển thị các đơn đặt hàng từ ngày 1/5/07 đến

ShipCountry	In (“Canada”, “UK”)  Not “USA”	Dùng toán tử IN để hiển thị đơn đặt hàng gửi đến “Canada” hay “UK” Dùng toán tử NOT để hiển thị đơn đặt hàng gửi đến nơi không phải là USA.
-------------	--------------------------------------	--



CompanyName	>="N"  Len([CompanyName])> Val(30)	Hiển thị đơn đặt hàng gửi đến công ty có tên bắt đầu từ chữ N đến Z Dùng hàm Len để hiển thị những đơn đặt hàng của công ty có tên dài hơn 30 ký tự
OrderID	Right([OrderID],2)="99"	Dùng hàm Right để hiển thị các đơn đặt hàng có trị của OrderID kết thúc là 99

▣ Dùng trị DATE là tiêu chuẩn

Field	Biểu thức	Diễn giải
RequiredDate	Between Date() And DateAdd ("m",3,Date())	Hiển thị đơn đặt hàng yêu cầu giao hàng từ ngày hôm nay đến 3 tháng tới
OrderDate	<Date() - 30  Year([OrderDate])=2006 DatePart("q",[OrderDate])=4	Hiển thị đơn đặt hàng trước 30 ngày so với ngày hiện tại. Hiển thị đơn đặt hàng 2006 Hiển thị đơn đặt hàng thuộc quý 4

▣ Dùng trị trống (Null hay chuỗi rỗng) là tiêu chuẩn

Field	Biểu thức	Diễn giải
ShipRegion	Is Null  Is Not Null	Hiển thị đơn đặt hàng của Record có Field ShipRegion bỏ trống Hiển thị đơn đặt hàng của Record có Field ShipRegion không bỏ trống (có trị)
Fax	""	Hiển thị đơn đặt hàng của Record của khách hàng không có Fax

▣ Dùng một phần trị của Field là tiêu chuẩn

Field	Biểu thức	Diễn giải
ShipName	Like "S*" Like "*Import"  Like "[A-D]*"  Like "*ar*"	Hiển thị đơn đặt hàng có tên bắt đầu là S Hiển thị đơn đặt hàng có tên kết thúc là Import Hiển thị đơn đặt hàng có tên bắt đầu từ chữ A đến D

## 7.4 TẠO FIELD TÍNH TOÁN (CALCULATED) TRÊN QUERY

### 7.4.1 Khái quát

- Có thể tạo Field tính toán trên Query bằng cách kết hợp 2 hay nhiều Field khác của Bảng cơ sở.
- Access tự động đặt tên cho các Field tính toán là Expr1, Expr2 và sau đó người dùng có thể đặt lại bằng tên có ý nghĩa hơn.

### 7.4.2 Các bước khai báo

- Chọn một Field trống trên QBE và định vị dấu nháy trong dòng Field rồi nhập vào biểu thức tính toán cần thiết.
- Nếu biểu thức dài thì nhấn tổ hợp phím Shift+F2 để mở cửa sổ Zoom.
- Nhấn Enter để kết thúc biểu thức. Access tự đặt thêm tên mặc nhiên vào trước biểu thức và phân cách bởi dấu 2 chấm.

Ví dụ: tạo Field tính toán hiển thị HỌ Tên của Bảng

HOCSINH Expr1:[HOLOT] & " " & [TENHS]

- Sửa lại tên Field cho phù hợp (phần

Expr1) HOTEN: [HOLOT] & " " &

[TENHS]

### 7.4.3 Hiển thị một phần dữ kiện kiểu TEXT, kiểu ngày

Có thể tạo một Field tính toán trong đó.

- Sử dụng hàm LEFT(Text,n), RIGHT(Text,n), MID(Text,n1,n2) để trích một phần dữ liệu kiểu TEXT.
- Sử dụng hàm DATEPART(DatePart,Date) để trích một phần dữ liệu kiểu ngày, trong đó tham số datepart là ký hiệu tắt chỉ thành phần thuộc ngày ("d": ngày, "m": tháng, "yyyy": năm (4 ký số), "q": quý (Quarter), "ww": tuần (week)).

Ví dụ: trị ngày của OrderDate là 3-Jun-93, biểu thức Datepart("m",[OrderDate]) cho kết quả là 6 (tháng trong năm).

### 7.4.4 Hiển thị có điều kiện

Sử dụng hàm IIF() để định điều kiện hiển thị cho

Field. Cú pháp hàm: IIF(điều kiện, trị đúng, trị sai).

Ví dụ: Tạo Calculated Field để hiển thị số lượng nhập biết rằng trong Bảng cơ sở chỉ có một Field SOLUONG ghi

chung số lượng nhập xuất và dùng MSPHIEU (ký tự N hay X) để phân biệt nhập xuất. Calculated Field như sau:  
SLNHAP : IIF([MSPhieu]="N", [SOLUONG],0)

### 7.4.5 Trích chọn trị duy nhất

Mặc nhiên, Query sẽ hiển thị những Record thỏa tiêu chuẩn đã khai báo ở dòng Criteria. Muốn chỉ hiển thị Record duy nhất khi có nhiều Record thỏa tiêu chuẩn: khai báo thuộc tính Unique Value của Query là YES.

## 7.5 QUERY TÍNH TỔNG CỘNG NHÓM

### 7.5.1 Kết nhóm dữ liệu

- Có thể hiển thị các mẫu tin trong Query theo nhóm: kết nhóm dữ liệu.
- Thao tác
  - Mở Query trong Design View, đưa các Bảng liên quan vào Query. Đưa các Field liên quan vào lưới QBE.
  - Sắp xếp các Field cần kết nhóm theo thứ tự từ trái sang phải. Field kết nhóm cao nhất được đặt ở ngoài cùng bên trái lưới QBE.
  - Đưa dòng Totals vào lưới QBE: chọn menu VIEW – Totals hay click biểu tượng Total trên Toolbar  $\Sigma$
  - Chọn mục GROUP BY trên dòng Total của Field cần kết nhóm.

### 7.5.2 Tính tổng

- Có thể thực hiện các phép tính Tổng trên nhóm dữ liệu nào đó bằng cách sử dụng Query tổng cộng (Totals Query).
- Trong Query này, khai báo hiển thị những trị tổng cộng và Query này còn được gọi là Aggregate Query (Query tính gộp).

#### 7.5.2.1 Các hàm có thể sử dụng trong Query tổng cộng

Tên hàm	Ý nghĩa
SUM	Tính tổng của một Field số học
AVG	Tính trung bình của một Field số học
MIN/MAX	Tính trị cực đại/cực tiểu của một Field số học
COUNT	Đếm số lượng mẫu tin có dữ kiện trong Field
FIRST/LAST	Trị của Field thuộc mẫu tin đầu tiên/cuối cùng của Bảng/Query

#### 7.5.2.2 Tính tổng đối với tất cả Record của Bảng

- ◆ Tạo Query mà dữ liệu cơ sở là Bảng/Query muốn tính tổng và đưa dòng Totals vào lưới QBE.
- ◆ Trong mỗi dòng Total của mỗi Field, chọn phép tính (SUM, AGV, ...). Vì muốn tính tổng của tất cả các Record nên không

khai báo GROUP BY (TỔng theo nhóm).

- ◆ Trong bảng kết quả Query, Access hiển thị những tiêu đề để nhận biết. Bảng Query tổng cộng chẳng hạn như AvgOfUnit, SumOfPrice, ... và người sử dụng có thể thay đổi lại các tiêu đề đó.

### 7.5.3 Khai báo tiêu chuẩn lựa chọn trong Query tổng cộng nhóm

- Có thể khai báo tiêu chuẩn lựa chọn trong Query tổng cộng nhóm và như vậy cũng có nghĩa là đã giới hạn mẫu tin đã tổng kết được hiển thị thuộc mỗi nhóm.  
Ví dụ: Tạo bảng Query tổng cộng theo nhóm mặt hàng: khai báo tiêu chuẩn cho Field mặt hàng (Field tính tổng là Field số lượng).
- Khai báo tiêu chuẩn tổng cộng: khai báo tiêu chuẩn cho Field làm tiêu chuẩn lựa chọn (nhưng không là Field dùng để tính tổng).
- Khi chạy Query, Access sẽ thực hiện đối với tất cả các mẫu tin của Bảng lựa chọn rồi lựa chọn các Record theo tiêu chuẩn chỉ định. Nói cách khác, tổng cộng trước rồi mới lựa chọn Record kết quả.

### 7.5.4 Khai báo tiêu chuẩn giới hạn Record trước khi tính tổng

- Với phương thức này, Access thực hiện việc lựa chọn trước rồi mới tính tổng cộng.
- Khai báo tiêu chuẩn
  - Phải khai báo tiêu chuẩn cho Field làm tiêu chuẩn chọn loại
  - Và khai báo tiêu chuẩn cho cả Field dùng để tính toán
  - Ví dụ như khai báo tiêu chuẩn cho Field mặt hàng và khai báo tiêu chuẩn cho Field số lượng (Field tính tổng).

### 7.5.5 Khai báo biểu thức tổng cộng - Mục chọn Expression

Khi thiết lập Query tính tổng, tại dòng Total của Field muốn tính tổng, phép tính thường dùng là các hàm đơn giản có sẵn. Trường hợp muốn tự khai báo một biểu thức khác biệt với những mục chọn tính toán sẵn có tại mục Total thì làm như sau:

- Mở Query trong Design View và hiển thị dòng Total trên QBE.
- Ở dòng Total của Field muốn lập biểu thức, chọn mục  
EXPRESSION.
- Tại dòng FIELD của Field cần lập biểu thức: nhập biểu thức hay dùng công cụ Expression Builder để khai báo biểu

thức.

GHI CHÚ

- 9 trong 12 mục chọn của dòng Total là các hàm Aggregate (hàm tính gộp). Các hàm này không bao gồm các Field chứa trị NULL. Để đếm trị NULL, phải chuyển chúng thành zero (hàm NZ()).
- Các tùy chọn còn lại của dòng Total:

Chọn	Để
GROUP BY	Định nghĩa các nhóm muốn thực hiện tính toán. Ví dụ: để trình bày tổng số bán theo LOAIHH, chọn Group By Field LOAIHH.
EXPRESSION	Tạo một Calculated Field có đính kèm hàm Aggregate trong biểu thức của nó. Thông thường tạo một Calculated
WHERE	Chỉ định tiêu chuẩn cho một Field mà không dùng định nghĩa nhóm. Nếu chỉ định tùy chọn này cho một Field, Access sẽ ẩn nó trong kết quả của Query (bỏ chọn checkbox SHOW).

## 7.6 SUMMARY SELECT QUERY

### 7.6.1 Khái quát

- Khi có nhu cầu xem kết quả tổng hợp mà không quan tâm đến các Record chi tiết: Thu Chi trong ngày (không quan tâm thu chi về vấn đề gì). Đây là lĩnh vực thích hợp cho Summary Select Query.
- Ví dụ Bảng PHATSINH lưu các hoạt động phát sinh hàng ngày gồm xuất nhập vật tư và các hoạt động thu chi.  
Yêu cầu lập một Query hiển thị số liệu thu chi trong ngày (không hiển thị các lý do thu chi).

### 7.6.2 Tạo Summary Select Query đơn giản

- Tạo mới Select Query (Design View). Chọn Bảng tham gia (ví dụ Bảng PHATSINH).
- Chọn các Field đưa vào QBE (ví dụ Field NGAYCT, LOAICT, TIEN)
- Đưa dòng TOTAL vào lưới QBE.
- Trên dòng Totals: chọn GROUP BY cho NGAYCT (và xếp thứ tự Ascending), SUM cho TIEN và WHERE cho LOAICT.
- Dòng Criteria của LOAICT nhập: "T" Or "C" (chứng từ là thu hay chi)

### 7.6.3 Summary Select có sử dụng Calculated Field

Yêu cầu lập một Query hiển thị số liệu thu chi trong ngày theo hai cột riêng: cột Thu và cột Chi.

- Tạo mới Select Query (Design View). Chọn Bảng tham gia (PHATSINH).
- Chọn các Field đưa vào QBE (NGAYCT, LOAICT, TIEN).
- Đưa dòng TOTAL vào lưới QBE



- Trên dòng Totals: chọn Group BY cho NGÀYCT (và xếp thứ tự Ascending) và WHERE cho LOAICT.
- Dòng Criteria của LOAICT nhập: "T" Or "C" (chúng từ là thu hay chi)
- Tạo 2 Calculated Field THU và CHI  
 THU:SUM(IIF([LoaiCT]="T",[Tien],0)) và  
 CHI:SUM(IIF([LoaiCT]="C",[Tien],0))
- Chọn Expression trên dòng Total cho 2 Calculated Field THU, CHI.

## 7.7 CÁC VẤN ĐỀ BỔ SUNG

### 7.7.1 Thiết kế Query để hiệu chỉnh dữ liệu cơ sở

- Khi thiết kế Query dựa trên một hay hai Bảng có quan hệ 1-1 thì tất cả các Field khai báo trong Query đều có thể hiệu chỉnh được. Nhưng nếu thiết kế Query trên cơ sở dữ liệu từ nhiều Bảng và một số Bảng có quan hệ 1-nhiều thì vấn đề có khó khăn hơn.
- Khi thiết kế một Query dựa trên dữ liệu của nhiều Bảng, để có thể cập nhật dữ liệu trên Query, phải đảm bảo rằng các Bảng thuộc quan hệ MỘT phải có Field liên kết là Field Primary Key hay chỉ mục duy nhất.

#### 7.7.1.1 *Field nào có thể hiệu chỉnh*

Loại Query hay Field	Phép hiệu chỉnh
Query với một Bảng cơ sở	Cho phép
Query với dữ liệu là những Bảng có quan hệ 1-1	Cho phép
Query với dữ liệu là những Bảng có quan hệ 1-nhiều	Thông thường cho phép
Crosstab Query	Không
Query có tính toán tổng cộng	Không
Query có thuộc tính Unique Value là YES	Không
Union Query	Không
Pass-Through Query	Không
Các Calculated Field của Query	Không
Field có thuộc tính Read-Only	Không
Query có Attached Table	Không
Field thuộc Record đã bị xóa hay Khóa bởi USER khác	Không (nhiều USERS)
Query hay Bảng không cho phép thực hiện tác vụ INSERT, REPLACE và DELETE	Không

### 7.7.1.2 Hiệu chỉnh trong Query có dữ liệu ở 2 bảng quan hệ 1-∞

#### ◆ Phạm vi hiệu chỉnh

**Được** hiệu chỉnh mọi Field trừ Field liên kết với dữ liệu Bảng

phía quan hệ MỘT. Có hai trường hợp cần lưu ý như sau:

- Có thể hiệu chỉnh dữ liệu của Field liên kết phía quan hệ MỘT thuộc khai báo quan hệ không chặt (OUTER) khi trị Field liên kết đó không có Record đối chiếu trong Bảng phía nhiều.
- Có thể hiệu chỉnh dữ liệu của Field liên kết quan hệ MỘT thuộc khai báo liên kết trong đó có thuộc tính toàn vẹn tham chiếu với mục chọn CASCADE UPDATE RELATED FIELDS.
- ◆ Thiết kế Query, Biểu mẫu cho/không hiệu chỉnh dữ liệu
  - Muốn một số Field trong biểu mẫu không được hiệu chỉnh Lập thuộc tính LOCKED của các Control ứng với Field đó là YES.
  - Không muốn cho phép hiệu chỉnh bất cứ Field nào Thuộc tính AllowEditing: Unavailable và thuộc tính DefaultEditing: Read-Only (hoặc khai báo thuộc tính AllowUpdating là No Tables)
  - Nếu muốn sử dụng công cụ AutoLookup
    - y Phải đưa các Field thuộc Bảng phía quan hệ NHIỀU vào QUERY. Đối với biểu mẫu thì phải dùng Field liên kết ở Bảng phía nhiều như một Field cho phép điều chỉnh.
    - y Có thể dùng List Box hay Combo Box để liệt kê những trị có thể lựa chọn cho một Field nào đó trong biểu mẫu để thuận tiện nhập dữ kiện.

**BÀI TẬP:**

1. Mở tập tin QLCN.MDB ra và chọn trang Queries, thực hiện các truy vấn sau:
  - Liệt kê MASP, TENSP, tổng số sản phẩm đã thực hiện nhiều nhất trong tháng 9/2005.
  - Liệt kê MASP, TENSP mà không có bị phế phẩm.
  - Liệt kê MACN, HOTEN, PX, tổng tiền lương (tính bằng công thức: số sản phẩm nhân đơn giá trừ số phế phẩm nhân 1/2 đơn giá)
  - Liệt kê MASP, TENSP, DONGIA, tổng số sản phẩm đã thực hiện, thành tiền mà có đơn giá trên 20 ngàn và tổng số sản phẩm thực hiện từ 100 trở lên.
  - Liệt kê HOTEN công nhân, TENSP, tổng số sản phẩm mà làm được nhiều nhất của mỗi công nhân.
2. Mở tập tin QLCT.MDB ra và chọn trang Queries, thực hiện các truy vấn sau:
  - Liệt kê MACT, TENCT, NGAYKC, tổng thành tiền vật tư đã xuất trong tháng 3/06.
  - Liệt kê MAVT, TENVT, tổng thành tiền vật tư đã xuất nhiều nhất.
  - Liệt kê phiếu xuất vừa có xuất vật tư cát vừa có xuất vật tư xi măng
  - Liệt kê TENCT, TENVT, tổng thành tiền vật tư đã xuất nhiều nhất cho mỗi công trình.
  - Liệt kê các công trình chưa có xuất vật tư lần nào.
3. Mở tập tin VPP.MDB ra và chọn trang Queries, thực hiện các truy vấn sau:
  - Liệt kê các mặt hàng có tổng số lượng bán trên 100
  - Liệt kê các hóa đơn có tổng trị giá từ 1 triệu trở lên
  - Liệt kê mặt hàng bán ít nhất
  - Liệt kê khách hàng với tổng giá trị hóa đơn mua hàng nhiều nhất
  - Liệt kê mặt hàng chưa từng bán.
  - Liệt kê khách hàng chưa từng mua hàng lần nào.

## CHƯƠNG 8

## CÁC QUERY KHÁC

- Parameter Query
- Crosstab Query
- Action Query
- Find Duplicates – UnmMatched

**8.1 PARAMETER QUERY****8.1.1 Giới thiệu**

- Khi thiết kế Query, có thể hiển thị các mẫu tin theo một hay nhiều điều kiện nào đó bằng cách nhập trực tiếp điều kiện lọc vào mục Criteria của Field liên quan. Tuy nhiên, phương thức này không phù hợp vì các điều kiện lọc thường thay đổi theo nhu cầu sử dụng và do người sử dụng nhập vào cho mỗi lần chạy. Để giải quyết vấn đề này: sử dụng Parameter Query.
- Một Query được nhận tham số nhập từ bàn phím để định tiêu chuẩn lọc các mẫu tin cho mỗi lần chạy được gọi là Parameter Query.
- Trong hộp thoại Enter Parameter Value, chỉ có thể nhập vào một tiêu chuẩn và không phải là một biểu thức.

**8.1.2 Tạo Parameter Query**

- Mở Query ở chế độ Design View.
- Tại dòng Criteria của Field muốn dùng làm tham số
  - Nhập chuỗi ký tự thông báo trong dấu ngoặc vuông (Thông báo này sẽ được hiển thị trong hộp thoại ENTER PARAMETER VALUE khi chạy Query).
  - Chuỗi ký tự không được trùng với tên Field.

- Có thể dùng Query với thông báo yêu cầu đánh vào vài ký tự đầu tiên của trị muốn tìm.

Ví dụ khai báo: LIKE [Nhập các ký tự đầu Tên HS] &"\*"

- Chuỗi ký tự thông báo được xem như một biến chứa tham số truyền vào cho Query.

### 8.1.3 Hiển thị thông báo với nhiều tham số

Có thể tạo ra Parameter Query với nhiều tham số : một tham số cho mỗi Field hay nhiều tham số cho một Field (dùng tác tử Between ... And ... để biểu thị cho dãy trị).

Ví dụ: **Between [Nhap ngay bat dau] And [Nhap ngay ket thuc]**

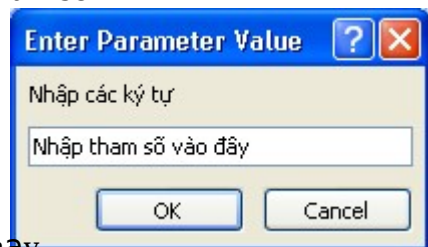
### 8.1.4 Khai báo dữ kiện của tham số

- Có khi đã khai báo tham số nhưng Query hoạt động không chính xác vì chưa khai báo kiểu dữ kiện của tham số.
- Để có thể khai báo kiểu của tham số truyền cho Query.
  - Thiết lập Parameter Query theo các bước trên.
  - Chọn lệnh QUERY – PARAMETER. Hộp thoại Parameter Query hiển thị.
  - Nhập chuỗi tên các tham số vào cột Parameter, khai báo kiểu tương ứng ở cột Data Type.
  - Tiếp tục khai báo các thông số theo đúng tên và thứ tự đã khai báo trong QBE cho đến khi hoàn tất (*từ trái sang phải*).



### 8.1.5 Thực hiện Query có dùng tham số

- Khi thực hiện Query có dùng tham số, ACCESS hiển thị hộp thoại Parameter Value để người dùng nhập trị cho tham số. Chuỗi ký tự trong cặp ngoặc vuông (tên biến) chính là chuỗi ký tự hướng dẫn nhập tham số
- Nếu thiết kế Parameter Query với yêu cầu trả lời nhiều tiêu chuẩn cho mỗi lần chạy thì các khung đối thoại cho từng tiêu chuẩn đã khai báo lần lượt được hiển thị.
- Theo ví dụ trên, khi chạy



Query, hiển thị Parameter hình bên.

### 8.1.6 Hiển thị thông báo nhận trị tiêu chuẩn (Criteria) từ 1 biểu mẫu

- Ví dụ phải thiết kế một biểu mẫu trong đó hiển thị thông báo yêu cầu đánh vào một tên Nhân Viên rồi tên nhân viên đó được sử dụng như một trị tiêu chuẩn. Chạy Query lấy dữ kiện từ một biểu mẫu làm tiêu chuẩn được gọi với thuật ngữ là QUERY BY FORM.
- Để tạo Query by Form, phải tạo các thành phần sau
  - Tạo biểu mẫu trong đó khai báo Macro hay thủ tục xử lý sự kiện (Event Procedure) để chạy bằng Query sau khi đánh vào biểu mẫu trị sử dụng làm tiêu chuẩn lựa chọn Record.
  - Bảng Query.
- Ví dụ: Giả sử tên biểu mẫu là CUSTOMER INFO, tại mục APTER UPDATE của EVENT PROPERTY, Click nút 3 chấm để mở lệnh khai báo Macro thực hiện tác vụ mở Query (giả sử tên là QUERY3. Trên Field tương ứng của QUERY dòng Criteria, chọn biểu tượng 3 chấm (Build) trên thanh công cụ để mở Expression Builder để nhập biểu thức tiêu chuẩn như sau : [FORMS]![tên biểu mẫu]!ID.

Thay vì dùng Expression Builder để khai báo, có thể nhập trực tiếp như sau : [Forms]![Tên biểu mẫu]![Tên ô điều khiển]

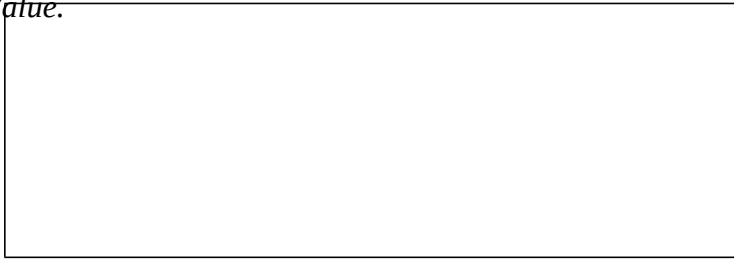
## 8.2 CROSSTAB QUERY (THAM KHẢO CHÉO)

Tạo Crosstab Query để tổng kết dữ kiện theo một dạng tương tự Bảng tính. Query này có thể biểu diễn số lượng lớn dữ kiện theo dạng thức dễ đọc và dễ so sánh. Ngoài ra có thể sử dụng làm dữ kiện cơ sở cho các báo biểu.

### 8.2.1 Tạo Crosstab Query không dùng Wizard

- Tạo Query bằng Design View
  - Tạo một Query bằng Design View.
  - Bật Table Name và Total
    - γ Chọn menu VIEW - TABLES NAMES: hiển thị dòng Table trong QBE
    - γ Chọn menu VIEW - TOTALS để hiển thị dòng Total
  - (Access tự gán thuộc tính Group By)
  - Xác định loại Query
    - Chọn menu QUERY - CROSSTAB QUERY (xác định kiểu Query đang thiết kế)
- Khai báo tiêu đề dòng, cột và tổng hợp trên dòng CrossTab
  - Tại Field làm tiêu đề dòng của Query
    - γ Tại dòng Crosstab, chọn mục ROW HEADING

- γ Tại dòng Total, phải khai báo là Group By
- Tại Field làm tiêu đề cột của Query
  - γ Tại dòng Crosstab, chọn mục COLUMN HEADING
  - γ Tại dòng Total, phải khai báo là Group By
  - γ Chỉ được phép chọn một Field có dòng CrossTab là Column Heading.
- Tại Field muốn là số liệu tổng kết
  - γ Tại dòng CrossTab chọn mục Value.
  - γ Tại dòng Total của Field này phải khai báo phép tính tổng kết như là Sum, Count, ...v.v.
  - γ Chỉ được phép chọn một Field có dòng CrossTab là Value.



## 8.2.2 Tạo CrossTab-Query bằng Wizard

### 8.2.2.1 Mở CrossTab-Query Wizard

- ◆ Chọn CrossTab-Query trong hộp thoại NEW QUERY.
- ◆ Hoặc click nút NEW OBJECT trên Toolbar, chọn Query rồi chọn CrossTab-Query trong hộp thoại NEW QUERY.

### 8.2.2.2 Các bước của Wizard

- ◆ Bước 1: chọn Bảng/Query tham gia.
- ◆ Bước 2: chọn Field làm trị cho ROW HEADING
  - Được chọn tối đa 3 Field
  - Chọn tên Field trong phần Available
    - γ Click nút > để chuyển qua khung Selected Field.
    - γ Nếu làm sai: click nút < để chuyển ngược lại.
  - Tên của Field được chọn sẽ hiển thị trên Sample.
- ◆ Bước 3: chọn Column Heading
  - Chỉ được chọn một Field
  - Tên của Field được chọn sẽ hiển thị trên Sample
  - Cách chọn: tương tự như trên.
- ◆ Bước 4:



- Chọn tên Field làm trị tính tổng (chỉ được chọn một Field)
- Chọn tên Field trên khung Field.

- Chọn phép tính trên khung Function
  - ◆ Bước 5: đặt tên Query chỉ định và chọn cách kết thúc
- Mở Query để xem (chọn View Query) hay vào chế độ Design để hiệu chỉnh.

### 8.2.3 Các khai báo bổ sung

Dù tạo CrossTab Query bằng Wizard hay Query Design, có thể thực hiện các khai báo bổ sung gồm:

#### 8.2.3.1 Khai báo tiêu đề cột (Column Heading)

- ◆ Khi chạy CrossTab Query, Access sắp xếp kết quả văn tin tùy thuộc trị của Field chọn làm tiêu đề cột và theo:
  - Thứ tự tiêu đề cột đã khai báo
  - Hoặc theo thứ tự ABC
  - Hoặc theo thứ tự số
- ◆ Dùng thuộc tính Column Heading để khai báo lại thứ tự tiêu

đề

cột và theo quy trình sau

- Thiết kế CrossTab Query theo các bước nêu trên.
- Mở cửa sổ thuộc tính của Query.
- Tại mục thuộc tính ColumnHeadings, nhập vào danh sách chuỗi ký tự dùng làm thứ tự tiêu đề cột. Ví dụ: “VIET NAM”; “UNITED KINGDOM”; “CANADA”. Dấu phân cách giữa các chuỗi ký tự là List Separator quy định trong Control Panel và tiêu đề phải chính xác với các trị dùng làm tiêu đề cột.

#### 8.2.3.2 Khai báo tiêu đề cột có chiều dài cố định

- ◆ Để cải thiện tốc độ thực hiện Query bằng cách khai báo các

tiêu

đề có chiều dài cố định và chỉ định cột nào được hiển thị trong Query.

- ◆ Cú pháp: ghi tiếp trên dòng Field dùng là tiêu đề cột chuỗi ký tự: `FORMAT([tên Field],”chuỗi định dạng”)`

## 8.3 ACTION QUERY

Action Query (Query hành động) được dùng để thực hiện một công việc nào đó như tạo Bảng mới, cập nhật mẫu tin, ....

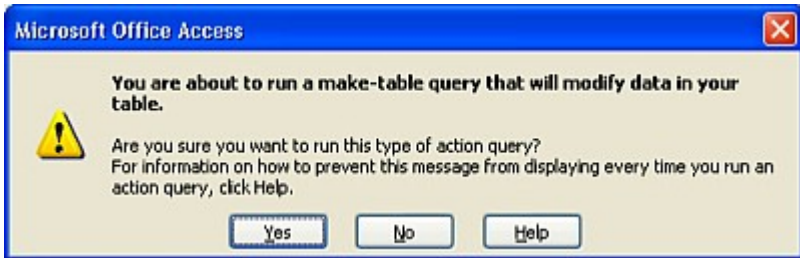
Access cung cấp 4 loại Action Query gồm Make Table Query, Delete Query, Append Query và Update Query.

### 8.3.1 Make Table Query

▣ Công dụng:

Dùng tạo **Bảng mới từ dữ kiện của các Bảng khác**. Cách này có lợi ích là tạo các báo biểu trình bày số liệu phát sinh trong một thời điểm nào đó, tạo bản lưu cho Bảng, ....v.v

- Quy trình
  - Tạo Query mới (Design View), chọn các Bảng và các Field tương ứng vào khung thiết kế (là dữ liệu nguồn dùng tạo Bảng mới).
  - Chọn menu Query – Make Table Query
  - Đặt tên Bảng mới trong khung Table Name của hộp thoại Make Table.
  - Khai báo CSDL nơi lưu Bảng mới: Current Database (CSDL hiện hành) hay Another Database (chỉ định tên Database mới)
  - Hoàn tất: ghi lưu Query.
  - Khi chạy Make Table Query: Access sẽ thông báo để yêu cầu người sử dụng xác nhận việc thực hiện.



**GHI CHÚ:** Tắt các hộp thoại cảnh báo khi thực hiện Action Query.  
 Chọn menu Tools – Options - Phiếu Edit/Find - Tại 118/uc

Confirm: BỎ chọn mục Action Queries.

### 8.3.2 Delete Query

- Công dụng: dùng để xóa các mẫu tin trong Bảng
- Quy trình
  - Tạo Query mới (Design View), chọn các Bảng tham gia (là Bảng mà các mẫu tin sẽ bị xóa bởi Query đang tạo)
  - Chọn menu Query – Delete Query
  - Chọn các Field sau đây vào QBE.
    - γ Biểu tượng Field hoa thị (\*) của Bảng chứa các mẫu tin muốn xóa. Chữ FROM hiển thị ở dòng DELETE ứng với cột chứa biểu tượng (\*)
    - γ Những Field của các Bảng đã chọn để đưa vào Query dùng làm tiêu chuẩn xóa mẫu tin của Bảng chứa dấu hoa thị. Chữ WHERE sẽ hiển thị ở dòng DELETE của Field này (Ví dụ: Field MAKHO của Bảng PHATSINH). Nhập tiêu chuẩn lọc mẫu tin xóa tại dòng CRITERIA (Ví dụ: MaKho là LB).
  - Khi chạy Query Delete, Access sẽ thông báo cảnh báo không thể phục hồi được.
  - Khi xóa bằng Delete Query, chỉ các mẫu tin bị xóa, Bảng vẫn còn (rỗng) trong CSDL. Muốn xóa Bảng phải thực hiện trong cửa sổ Database.

### 8.3.3 Append Query

- Công dụng: Dùng nối vào một Bảng các mẫu tin lấy từ một hay nhiều Bảng khác.
  - Quy trình
    - ◆ Tạo Query mới (Design View), chọn Bảng tham gia (là Bảng có các mẫu tin sẽ nối vào Bảng chỉ định)
    - ◆ Chọn menu Query – Append Query : hiển thị hộp thoại
- APPEND
- Khai báo CSDL đích lưu Bảng trong khung File Name. Khai báo tên Bảng trên khung Table Name (CSDL và Bảng đích đã phải tồn tại).
  - Chọn OK để đóng hộp thoại.
  - ◆ Trong cửa sổ Append Query
    - Không có dòng SHOW. Có dòng APPEND TO trong lưới QBE
    - Đưa vào lưới QBE những Field muốn ghi nối đuôi (Append)
    - Nếu Field của Bảng đích trùng tên với Field của Bảng cơ sở, những tên đó được tự động hiển thị ở

Append của Field.

- Muốn chọn Field đích theo ý muốn: click vào dòng này để chọn.

- Nếu chọn dấu hoa thị, tương ứng với việc chọn tất cả các

Field.  
 ♦ Ví dụ minh họa:

Trên cơ sở Bảng PHATSINH, có nhu cầu vào mỗi kỳ thời gian nào đó trích các mẫu tin thuộc chứng từ THU CHI sang Bảng riêng gọi là THUCHI và các mẫu tin thuộc chứng từ nhập xuất sang Bảng riêng gọi là NHAPXUAT (lần trích đầu tiên có thể dùng copy nhưng các lần sau phải dùng Append Query để ghi nối đuôi vào Bảng tương ứng). Trình tự thực hiện Append Query như sau:

- Trước hết, sao chép cấu trúc Bảng PHATSINH thành Bảng mới là THUCHI. Hiệu chỉnh lại cấu trúc cho phù hợp (bỏ các Field không liên quan - Ví dụ: Field MAVT trong Bảng THUCHI) để chuẩn bị cho lần trích đầu tiên.
- Tạo mới một Query trong cửa sổ Design View. Chọn Bảng PHATSINH làm Bảng tham gia.
- Chọn menu QUERY – APPEND QUERY để hiển thị hộp APPEND
  - y Nhập tên Bảng đích (nhận dữ liệu trích). Ví dụ: Bảng THUCHI
  - y Chọn CSDL chứa Bảng đích: Current Database (CSDL hiện hành). Thực hiện xong chọn OK.
- Chọn các Field của Bảng (PHATSINH) muốn trích ghi nối đuôi vào Bảng đích (THUCHI) đưa vào lưới. Nếu tên các Field của hai Bảng (PHATSINH và THUCHI) giống nhau, Access sẽ tự động đưa vào dòng Append To, ngược lại, mở Field tương ứng tại dòng Append To của Bảng đích để chọn tên Field nhận dữ liệu ghi nối đuôi.
- Nhập các tiêu chuẩn lọc vào dòng Criteria của Field tương ứng (Ví dụ: #<=08/05/97# tại Field NGÀY và “C” Or “N” tại Field LOAI).
- Thực hiện xong ghi lưu

Query.  
 ♦ Có thể dùng Calculated Field để làm tiêu chuẩn lọc cho Query. Ví dụ: mở thêm hai Field THANG và NAM như sau:

- Cột THANG: Month(NGAY)



tại dòng Criteria ghi: VAL([Thang nao])

- Cột NAM: Year(NGAY)

tại dòng Criteria ghi: VAL([Nam nao])

- ◆ Nếu không dùng Calculated Field thì tạo dòng Criteria của  
Field

ngày có thể ghi như sau: Month([NGAY])=[Thang nao]  
And  
Year([NGAY])=[Nam nao]

- ◆ Khi chạy Append Query, Access sẽ thông báo và chờ xác nhận của người sử dụng để thực hiện.

### 8.3.4 Update Query

□ Công dụng: dùng để cập nhật các Record trong Bảng.

□ Quy trình

- ◆ Tạo Query mới, chọn Bảng tham gia (có mẫu tin sẽ được

cập

nhật)

- ◆ Chọn menu Query – Update Query: trong lưới QBE có thay đổi
  - Không có dòng SORT và dòng SHOW
  - Có thêm dòng UPDATE TO
- ◆ Đưa các *Field muốn cập nhật* vào lưới QBE và những

*Field*

*dùng làm tiêu chuẩn* (Ví dụ: LOAI, SOLG, DGIA, TIEN).

- ◆ Khai báo tiêu chuẩn lọc tại dòng *Criteria* của Field tương

ứng

và khai báo tại dòng *Update To* biểu thức theo đó Access cập nhật Field tương ứng. Ví dụ: khai báo như sau

- Field DGIA, Update to [DGIA]\*1.1
- Field TIEN, Update to [SOLG]\*[DGIA]\*1.1
- Field LOAI, Criteria “N”
- ◆ Ghi lưu UPDATE QUERY. Khi chạy Query, Access sẽ thông báo: *không thể phục hồi các Record đã được cập nhật.*

## 8.4 FIND DUPLICATES QUERY – FIND UNMATCHED QUERY

### 8.4.1 Find Duplicates Query

□ Khái quát

- Trong thực tế, khi thao tác với Access, có thể xảy ra trường hợp được nhập nhiều lần cùng một mẫu tin (do nhầm lẫn chưa nhập). Để có thể phát hiện trường hợp này, sử dụng Find Duplicates Query.
- Đây cũng là một Select Query nhưng do tính phức tạp khi thiết kế nên cần sử dụng Wizard.
- Khởi động FIND DUPLICATES QUERY WIZARD: chọn Find Duplicates Query Wizard trong hộp thoại New Query.

□ Các bước của Wizard

- Bước 1: chọn Bảng/Query tham gia Query
- Bước 2: chọn Field kiểm tra trùng lặp
- Bước 3: hiển thị thêm các Field bổ sung
- Bước 4: đặt tên cho Query và ghi lưu

## 8.4.2 Find UnMatched Query Wizard

- Khái quát

- Trong quá trình nhập liệu, có thể có trường hợp không khớp dữ liệu giữa hai Field liên kết của hai Bảng. Trong trường hợp này dùng Find UnMatched Query để kiểm tra sự không khớp này.
- Khởi động Find UnMatched Query Wizard: chọn Find UnMatched Query Wizard trong hộp thoại New Query.

□ Các bước của Wizard

- Bước 1: chọn Bảng /Query cần kiểm tra so khớp
- Bước 2: chọn Bảng/Query là chuẩn so khớp
- Bước 3: chọn Field so khớp giữa hai Bảng
- Bước 4: chọn Field hiển thị
- Bước 5: đặt tên cho Query

**BÀI TẬP:**

- Mở tập tin QLCN.MDB ra và chọn trang Queries, thực hiện các truy vấn sau:
  - Liệt kê các sản phẩm có cùng đơn giá.
  - Liệt kê MACN, HOTEN, PX của phân xưởng bất kỳ nhập vào.
  - Thực hiện crosstab query liệt kê ra HOTEN, PX của công nhân, và các cột ứng với các tên sản phẩm trong đó chứa tổng số sản phẩm mà công nhân đã làm.
  - Dùng make table query để tạo ra table tên LUONG\_09 gồm các field MACN, HOTEN, PX, MASP, TENS, DONGIA, SOSP, SOPP trong đó chứa dữ liệu có ngày chấm công thuộc tháng 9/05.
- Tạo tập tin CSDL để quản lý điểm của học sinh đặt tên là QLD.MDB và tạo trong đó các table sau đây:

Table GIAOVIEN có cấu trúc:

Field Name	Data Type	Field size	Giải thích
<u>MAGV</u>	Text	3	Qui định nhập 3 số, khóa chính.
TENGV	Text	40	Tên giáo viên
MON	Text	10	Môn học mà giáo viên dạy

Table LOPHOC có cấu trúc:

Field Name	Data Type	Field size	Giải thích
<u>LOP</u>	Text	4	Qui định nhập 2 số, 1 chữ hoa và 1 số. Đây là khóa chính.
BUOI	Text	10	Buổi học: sáng, chiều hoặc tối
GVCN	Text	3	Combo box chọn từ Table GIAOVIEN

Table HOCSINH có cấu trúc: khóa chính là MAHS

Field Name	Data Type	Field size	Giải thích
------------	-----------	------------	------------

<u>MAHS</u>	Text	7	Qui định nhập 2 số, 1 chữ, 1 số, dấu _ và 2 số
HO	Text	40	Họ và chữ lót
TEN	Text	10	Tên học sinh
PHAI	Yes/No		Phái nam là Yes, phái nữ là No
LOP	Text	4	Combo box chọn từ Table LOPHOC

Table TONGKET có cấu trúc sau: khóa chính là MAHS và MON

Field Name	Data Type	Field size	Giải thích
MAHS	Text	7	Combo box chọn từ Table HOCSINH
MON	Text	10	Môn học
HK1	Number	Double	Điểm học kỳ 1, qui định nhập 0 đến 10, định dạng lấy một số lẻ.
HK2	Number	Double	Điểm học kỳ 2, qui định nhập 0 đến 10, định dạng lấy một số lẻ.

3. Liên kết các table và nhập dữ liệu sau:

Table GIAOVIEN

MAGV	TENGV	MON
001	Nguyễn Văn An	Anh văn
002	Trần Thanh Vân	Toán
003	Lê Mỹ Hân	Sử
004	Phan Thế Hùng	Sinh
005	Lâm Thị Uyên	Văn
006	Nguyễn Việt	Hóa
007	Lê Văn Quốc	Lý
008	Trần Văn Hải	Địa

Table LOPHOC

LOP	BUOI	GVCN
10A1	Sáng	001
10A2	Sáng	006
10A3	Sáng	004
11B4	Chiều	001
11B5	Chiều	004
12C6	Sáng	007
12C7	Sáng	003
12C8	Tối	008

Table HOCSINH

MAHS	HO	TEN	PHAI	LOP
10A1_01	Lê Văn	Anh		
10A1_02	Trần Văn	Bảo	➤	
10A1_03	Lê Mỹ	Hạnh		
10A2_04	Lâm Thy	Linh		
10A2_05	Nguyễn	Bình	➤	
11B4_06	Trần	Trung	➤	
11B4_07	Phan Thúy	Vân		
11B5_08	Đình Thái	Hà		
11B5_09	Lý Văn	Tuấn	➤	
12C6_10	Lê Thanh	Hằng		
12C6_11	Phan Việt	Hà	➤	

Tạo query Update field LOP với nội dung là 4 ký tự đầu của field

MAHS trong HOCSINH

Table DIEMDANH có cấu trúc sau: khóa chính là MAHS và NGÀY

Field Name	Data Type	Field size	Giải thích
MAHS	Text	7	Combo box chọn từ Table HOCSINH
NGAY	Date/Time		Ngày nghỉ, không được nhập lớn hơn ngày hiện tại, định dạng hiện ngày/tháng
PHEP	Yes/No		Nghỉ có phép là Yes, không phép là No

Table TONGKET

MAHS	MON	HK1	HK2
10A1_01	Toán	7.5	6.7
10A1_01	Văn	6.1	6.8
10A1_01	Sinh	8.6	8.1
10A1_02	Toán	5.4	4.3
10A1_02	Anh văn	4.1	5.0
10A1_02	Hóa	6.2	4.0
10A1_03	Toán	8.0	8.4
10A1_03	Sử	8.2	8.3
11B4_06	Văn	7.0	7.5
11B4_06	Lý	8.2	7.4
11B4_06	Hóa	9.1	8.0
11B4_07	Anh văn	7.6	7.1
11B4_07	Toán	5.2	6.0
11B5_08	Sinh	6.5	5.0
11B5_08	Sử	8.1	9.0
11B5_09	Hóa	7.5	8.6

Table DIEMDANH

MAHS	NGAY	PHEP
10A1_01	15/02	☑
10A1_02	15/02	
10A2_04	16/02	
10A2_04	18/02	
10A1_01	21/02	☑
11B4_06	21/02	
10A2_04	21/02	
11B4_06	22/02	
11B5_08	22/02	☑
10A2_05	25/02	☑
11B5_08	25/02	
10A1_01	27/02	☑
10A1_02	13/03	
11B4_06	13/03	☑
10A1_01	15/03	☑
12C6_10	15/03	☑

#### 4. Thực hiện các truy vấn sau

- Query 1: Liệt kê MAHS, HO, TEN, LOP, TENGV chủ nhiệm lớp và học buổi “Chiều”
- Query 2: Liệt kê MAHS, HO, TEN, LOP, MON, điểm TB cả năm của một môn học bất kỳ nhập vào.
- Query 3: Liệt kê MAHS, HO, TEN, LOP, điểm tổng kết (là trung bình cộng của điểm cả năm các môn học cộng lại) theo từng học sinh.
- Query 4: Liệt kê các học sinh chưa có nhập điểm tổng kết các môn học.



- Query 5: Liệt kê lớp, tổng số học sinh, tổng số học sinh nam, tổng số học sinh nữ.
- Query 6: dùng make table query tạo table mới tên HS\_KG gồm MAHS, HO, TEN, MON, điểm cả năm của học sinh mà đạt từ 7 điểm trở lên.
- Query 7: Dựa vào query 3, liệt kê tất cả thông tin kèm theo cột xếp loại với nội dung là Yếu nếu điểm tổng kết dưới 5, là Trung bình nếu điểm tổng kết từ 5 đến dưới 7, là Khá nếu điểm tổng kết từ 7 đến dưới 9, ngược lại từ 9 đến 10 là Giỏi.
- Query 8: Liệt kê các giáo viên không chủ nhiệm lớp nào.
- Query 9: Liệt kê MAHS, HO, TEN, LOP, tổng số ngày nghỉ trong một tháng bất kỳ nhập vào.
- Query 10: Liệt kê MAHS, HO, TEN, số ngày nghỉ có phép, số ngày nghỉ không phép.
- Query 11: Liệt kê lớp có tổng số ngày nghỉ của học sinh nhiều nhất.
- Query 12: dùng make table query tạo ra table mới tên NGHI\_K10 gồm MAHS, HO, TEN, LOP, NGÀY, PHEP của học sinh thuộc khối lớp 10.
- Query 13: Dựa vào query 10, liệt kê tất cả thông tin kèm theo cột xếp loại đạo đức với nội dung là D nếu nghỉ không phép từ 3 ngày trở lên, là C nếu không phép từ 1 đến 2 ngày, là B nếu nghỉ có phép từ 3 ngày trở lên, ngược lại các điều trên thì là A.
- Query 14: dùng append query để thêm vào các field trong table NGHI\_K10 phần điểm danh nghỉ học của các học sinh thuộc khối lớp 11.
- Query 15: dùng delete query để xóa các record trong table HS\_KG mà có điểm cả năm là dưới 8.

## CHƯƠNG 9

**BIỂU MẪU**

- Khái quát về Biểu mẫu
- Thiết kế với

Wizard

- Tự thiết kế Biểu mẫu
- Thao tác trên cửa sổ thiết kế
- Các thuộc tính thường dùng
- Tạo đối tượng của Biểu mẫu
- Các bổ sung

**9.1 KHÁI QUÁT VỀ BIỂU MẪU****9.1.1 Giới thiệu**

- Biểu mẫu (FORM) là cửa sổ dùng làm màn hình hiển thị thông tin từ các Bảng/Query, mẫu nhập liệu, hộp thoại để nhận chỉ thị từ người sử dụng.
- Đối tượng (Object) là thuật ngữ chỉ các Điều khiển (Control) và các thành phần khác trên Biểu mẫu. Bản thân Biểu mẫu cũng là một đối tượng. Trong Biểu mẫu, ngoài các Object dùng thể hiện dữ liệu còn có các Graphic Object dùng để trang trí (Line, Object Frame,...) hay giúp thao tác thuận tiện hơn.
- Dữ liệu nguồn của Biểu mẫu (Record Source) là Bảng hay Query hay lệnh SQL chứa các mẫu tin được hiển thị hay xử lý trên Biểu mẫu đó.

**9.1.2 Các loại Biểu mẫu thông dụng**

- Biểu mẫu dữ liệu dùng hiển thị và xử lý dữ liệu
- Biểu mẫu hộp thoại dùng hiển thị các điều khiển để chọn công việc.

**9.1.3 Thiết kế Biểu mẫu**

Có hai cách tạo Biểu mẫu: sử dụng Wizard (Form Wizard hay AutoForm) hoặc tự thiết kế thông qua cửa sổ Design View.

**9.1.4 Hiển thị Biểu mẫu**

- Các chế độ hiển thị Biểu mẫu
  - Design View: chế độ thiết kế
  - Form View : chế độ hiển thị chuẩn của Form đang hoạt động.
  - Datasheet View: hiển thị theo dạng thức của Bảng.



Chọn menu View - chọn chế độ hiển thị thích hợp hay click biểu tượng trên Toolbar.

### 9.1.5 Mở một Biểu mẫu đã thiết kế

Từ cửa sổ Database, chọn mục loại Forms rồi chọn tên Form cần mở và:

- Click nút Design: mở Biểu mẫu ở chế độ thiết kế
- Click nút Open: mở Biểu mẫu ở chế độ hoạt động (Form View).

## 9.2 THIẾT KẾ BIỂU MẪU VỚI WIZARD

Access cung cấp công cụ Wizard để giúp tạo nhanh một số loại Biểu mẫu.

### 9.2.1 Sử dụng AUTOFORM

#### 9.2.1.1 Các loại AutoForm

- ◆ Columnar

Là loại Biểu mẫu dạng một cột. Chỉ hiển thị một mẫu tin trên

Biểu mẫu. Mỗi Field trên 1 dòng với nhãn ghi tên Field bên trái.

- ◆ Tabular

Là loại Biểu mẫu dạng nhiều cột. Mỗi cột là một Field và mỗi dòng là một mẫu tin.

- ◆ Datasheet

Là loại Biểu mẫu có dạng thể hiện như Datasheet View của

Bảng/Query (cột là Field, dòng là record).

#### 9.2.1.2 Các bước thực hiện

- ◆ Mở hộp thoại NEW FORM: trong cửa sổ Database, chọn mục

loại FORMS rồi click nút NEW.

- ◆ Chọn loại AutoForm: chọn một trong các loại AutoForm.

- ◆ Chọn dữ liệu nguồn: click nút Dropdown mở danh sách và chọn Bảng/Query tham gia Biểu mẫu.

- ◆ Kết thúc: click OK. Wizard sẽ tạo một AutoForm theo dạng

chỉ

định và hiển thị kết quả theo chế độ FORM VIEW.

- ◆ Ghi lưu: ghi lưu Biểu mẫu và sẽ được đặt trong mục

loại

FORMS của cửa sổ Database.

#### 9.2.1.3 Dùng nhanh chức năng AutoForm

Ngoài trình tự nêu trên, có thể dùng nhanh chức năng AutoForm.

- 
- ◆ Trong cửa sổ Database, chọn mục loại Tables hay Queries, chọn Bảng hay Query làm dữ liệu cơ sở cho Biểu mẫu.
  - ◆ Click nút New Object trên Toolbar, chọn AutoForm (hay chọn menu INSERT – AUTOFORM).
  - ◆ Kết quả

- Một Biểu mẫu dạng một cột (Columnar) được tạo và hiển thị ở chế độ Form View.
- Nếu dữ liệu cơ sở của AutoForm là Bảng có khai báo quan hệ và là bên 1 thì Access sẽ tạo một MainForm-SubForm.

GHI CHÚ: Nếu dùng Query là dữ liệu cơ sở thì phải dùng Select Query.

Nếu không dùng đúng loại thì Access sẽ báo lỗi.

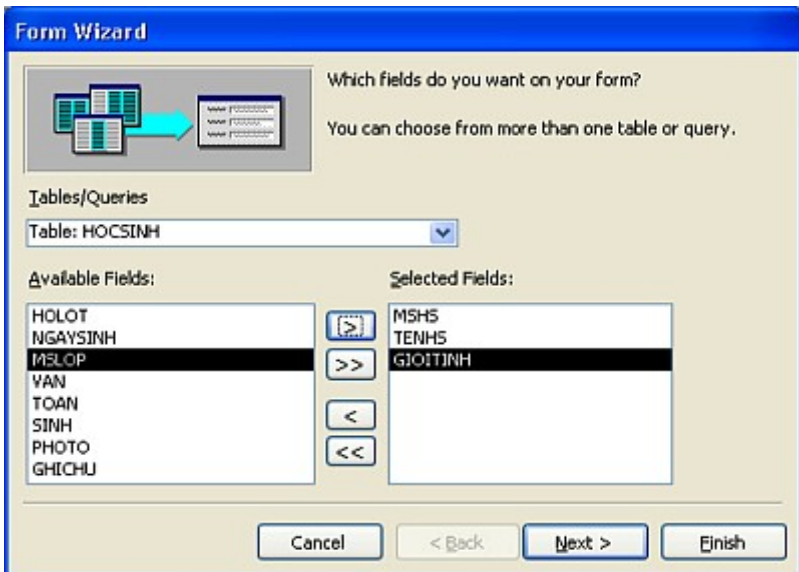
## 9.2.2 Sử dụng Form Wizard

### 9.2.2.1 Khởi động Form Wizard

- ◆ Chọn mục loại FORMS trong cửa sổ Database rồi click New.
- ◆ Trong hộp thoại New Form, chọn Form Wizard hay click Shortcut Create Form by using Wizard.
- ◆ Có thể chỉ định hay không chỉ định Bảng/Query tham gia.
- ◆ Thực hiện các bước theo hướng dẫn của Wizard.

### 9.2.2.2 Các bước của Wizard

- ◆ **BƯỚC 1:** Chọn các Field tham gia vào Biểu mẫu Access cho phép chọn nhiều Field từ nhiều Bảng/Query khác nhau.



- Chọn Bảng/Query từ danh sách trong khung TABLES/QUERIES
- Chọn Field trong khung Available Fields và chuyển qua khung Selected Fields.
- Lần lượt thực hiện như trên với các Bảng/Query chỉ định

- cần tham gia Form.
- Click NEXT để qua bước 2.

- ◆ **BƯỚC 2:** Chọn cách hiển thị dữ liệu
  - Hiển thị bước này khi có dữ liệu nguồn gồm Bảng/Query và có khai báo quan hệ.
  - Chọn cách hiển thị dữ liệu: Single Form (Biểu mẫu đơn), Main/SubForm (Biểu mẫu chính/phụ) hay Linked (liên kết). Click NEXT: qua bước 3.
- ◆ **BƯỚC 3:** Chọn cách trình bày Biểu mẫu (Layout):

gồm

Columnar, Tabular, Datasheet, Justified, ...v.v.

- ◆ **BƯỚC 4:** Chọn kiểu của Biểu mẫu (Style).
- ◆ **BƯỚC 5:** Chọn Tiêu đề (Title) cho Biểu mẫu và cách kết thúc.
  - Open the Form to View or Enter Information: mở Form để xem hay nhập thông tin.
  - Modify the Form's Design: hiệu chỉnh thiết kế Biểu mẫu. Và click Finish để kết thúc.

### 9.2.3 Chart Wizard và Pivot Table Wizard

Ngoài ra, Access còn cung cấp công cụ để tạo Biểu mẫu biểu đồ (Chart Wizard - biểu diễn sự biến thiên của dữ liệu số) và Biểu mẫu Pivot Table (Pivot Table Wizard - công cụ tương tự CrossTab Query).

## 9.3 TỰ THIẾT KẾ BIỂU MẪU

### 9.3.1 Khái quát

#### 9.3.1.1 Công cụ Form Design View

Form Design View là cửa sổ thiết kế Biểu mẫu dùng để hiệu chỉnh Biểu mẫu đã được tạo trước đó (tự tạo hay dùng Wizard) hoặc tự thiết kế Biểu mẫu từ đầu. Thực tế, thường dùng Wizard để thiết kế Biểu mẫu sau đó dùng Design View để hiệu chỉnh, bổ sung.

#### 9.3.1.2 Các phần của Biểu mẫu

- ◆ **Detail (Chi tiết)**
  - Là phần quan trọng nhất, bắt buộc phải có trên Biểu mẫu, dùng để thể hiện chi tiết của Biểu mẫu.
- ◆ **Form Header/ Form Footer**
  - Là phần tùy chọn.
  - Các điều khiển đặt trong Form Header luôn hiển thị ở phần đỉnh của Biểu mẫu.
  - Các điều khiển đặt trong Form Footer luôn hiển thị ở phần chân của Biểu mẫu.
  - Không bị che bởi phần Detail. Nếu phần Detail lớn hơn



phạm vi hiển thị thì có thanh trượt để cuộn màn hình và không chiếm diện tích của Form Header/Footer.

- Thường dùng thiết kế các Control để hiển thị Tên đơn vị, Nhân Biểu mẫu, ngày thiết kế, tác giả, các nút lệnh, ... v.v

- ◆ Page Header / Page Footer

Là phần tùy chọn và xuất hiện ở đầu cuối trang chỉ định (hiệu lực khi in).

### 9.3.1.3 Quy trình thiết kế Biểu mẫu với Design View

- ◆ Mở cửa sổ thiết kế (Design View) và hiển thị các vùng thiết kế

(Detail và các Header/Footer)

- ◆ Khai báo dữ liệu nguồn
- ◆ Đưa các Control vào Biểu mẫu và xác định các thuộc tính
- ◆ Chạy thử và ghi lưu

### 9.3.2 Cửa sổ thiết kế Biểu mẫu

#### 9.3.2.1 Mở cửa sổ thiết kế

Từ cửa sổ Database chọn mục loại FORMS.

- ◆ Cách 1: chọn Design View trong hộp thoại New Form.
- ◆ Cách 2: click Shortcut Create Form in Design View.

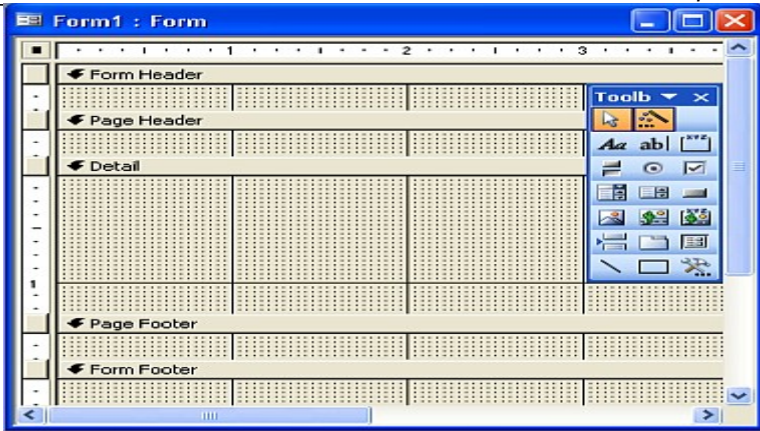
Cửa sổ Design View hiển thị một Biểu mẫu trắng (rỗng)

#### 9.3.2.2 Thành phần của cửa sổ thiết kế

- ◆ Thước (Ruler)

Thước ngang và dọc ở trên và trái cửa sổ được chia vạch theo inch hay centimeter theo quy định của Control Panel.

- Tắt mở thước: chọn menu VIEW - RULER
- ◆ Các vùng thiết kế: tương ứng với các phần của Biểu mẫu
  - Detail: luôn hiển thị khi mở Form Design View.
  - Tắt mở Form Header/Footer và Page header/Footer: Chọn menu VIEW – Form Header/Footer hay Page Header/Footer
- ◆ Toolbox
  - Gồm các dụng cụ dùng để thiết kế các đối tượng trên Biểu mẫu.
  - ◆ Cửa sổ thuộc tính
    - Dùng xác định thuộc tính cho Biểu mẫu và các điều khiển



9.3.3 Khai báo dữ liệu nguồn của Biểu mẫu

□ Chỉ khai báo dữ liệu nguồn cho Biểu mẫu dữ liệu

□ Cách khai báo

◆ Chọn

Bảng/Query làm  
dữ liệu nguồn  
trong cửa sổ New  
Form

◆ Hay tại thuộc tính  
Record Source  
của Biểu mẫu

- Click

Dropdown và  
chọn

Bảng/Query làm  
dữ liệu nguồn  
trong danh sách.

- Hoặc nhập dòng lệnh SQL để tạo Nguồn mẫu tin (hay click nút Build để thiết kế Query và Access sẽ điền lệnh SQL tương ứng vào thuộc tính)











9.3.4 Tạo các đối tượng trên Biểu mẫu




9.3.4.1 ToolBox

◆ Hiện thị Toolbox

Chọn menu View – Toolbox hay click biểu tượng Toolbox trên Toolbar.

◆ Các công cụ trên Toolbox

Dụng cụ	Công dụng
	Object Selector – Dùng chọn đối tượng
	Control Ward - Trợ giúp thiết kế điều khiển
	Label - Tạo điều khiển chứa chuỗi ký tự làm nhãn chú
	TextBox - Tạo điều khiển để hiển thị, nhập hay hiệu chỉnh
	Option Group - Tạo nhóm nút chọn, hộp kiểm hay nút bật tắt
	Toggle Button - Tạo nút bật tắt
	Option Button - Tạo nút chọn
	Check Box - Tạo hộp kiểm
	Combo Box - Tạo hộp danh sách trải xuống
	List Box - Tạo hộp danh sách

	Command Button - Tạo nút lệnh
	Image - Tạo điều khiển để hiển thị hình ảnh
	Unbound Object Frame - Tạo khung chức đối tượng không ràng buộc
	Bound Object Frame - Tạo khung chứa đối tượng ràng buộc
	Page Break - Tạo điều khiển ngắt trang
	Tab Control - Tạo điều khiển Phiếu (Tab)
	SubForm/SubReport - Tạo điều khiển chứa Biểu mẫu/Báo
	Line - Tạo đường kẻ
	Rectangle - Tạo khung viền chữ nhật
	More Controls – Chèn đối tượng ActiveX

#### 9.3.4.2 Phân loại điều khiển

- ◆ Điều khiển ràng buộc (Bound Control)
  - là loại Điều khiển mà dữ liệu chứa trong điều khiển là dữ liệu của một Field dữ liệu nguồn.
  - Dùng để hiển thị, hiệu chỉnh, nhập mới cho một Field của dữ liệu nguồn. Đây cũng là loại điều khiển thông dụng của Biểu mẫu.
- ◆ Điều khiển không ràng buộc
  - Là loại điều khiển không lấy dữ liệu từ nguồn nào cả.
  - Thường dùng để chứa các thông tin không cần lưu trữ vào dữ liệu nguồn hay thông tin luôn hiển thị trên Biểu mẫu.
- ◆ Điều khiển tính toán (Calculated Control)
  - là loại điều khiển mà dữ liệu là biểu thức mà Toán hạng là một hay nhiều Field của dữ liệu nguồn.

#### 9.3.4.3 Cách tạo điều khiển

Bật Control Wizard nếu muốn được Access hỗ trợ điều khiển. Cách chung để tạo một điều khiển trên Biểu mẫu:

- ◆ Điều khiển không ràng buộc
  - Click dụng cụ tương ứng trên Toolbox của Điều khiển muốn tạo và vẽ lên Biểu mẫu.
  - Khi tạo một điều khiển trên Biểu mẫu, Access thường tạo kèm theo Label để ghi chú giải cho điều khiển.
- ◆ Điều khiển ràng buộc

- Click dụng cụ tương ứng trên Toolbox của điều khiển muốn tạo và vẽ lên Biểu mẫu rồi khai báo thuộc tính Control Source là tên Field liên quan.
- Hoặc kích hoạt Control Wizard rồi chọn điều khiển tương ứng trên Toolbox và vẽ lên Biểu mẫu. Control Wizard hướng dẫn các bước để tạo điều khiển.
- Nếu muốn tạo Textbox để hiển thị dữ liệu Field thì mở Field List (chọn menu View – Field List hay click biểu tượng Field List trên Toolbar) và rê Field liên quan vào Biểu mẫu (Access sẽ chèn cả Label và Textbox).
- ◆ Điều khiển tính toán
  - Tạo Textbox trên Biểu mẫu.
  - Khai báo biểu thức tính toán cho thuộc tính Control Source của điều khiển.
  - Tên của điều khiển (thuộc tính Name) phải khác với tên Field dùng trong biểu thức.

#### 9.3.4.4 Khai báo biểu thức cho Điều khiển tính toán

- ◆ Biểu thức bắt đầu bởi dấu bằng (=)
- ◆ Tên Field được bao bởi cặp ngoặc vuông.
- ◆ Trục kiện là chuỗi ký tự được bao bằng dấu nháy kép.
- ◆ Hoặc dùng Expression Builder để tạo biểu thức.

#### 9.3.5 Xác lập thuộc tính cho đối tượng

- ▢ Chọn đối tượng và xác định thuộc tính trong cửa sổ thuộc tính.
- ▢ Có thể xác định thuộc tính khi Biểu mẫu ở chế độ Design View hay ở chế độ Form View.

#### 9.3.6 Hiển thị Biểu mẫu và ghi lưu

- ▢ Hiển thị Biểu mẫu khi đang thiết kế
  - Click biểu tượng View để hiển thị dạng Form View (biểu tượng sẽ chuyển sang là biểu tượng thiết kế).
  - Hoặc chọn menu View – Form View hay Datasheet View.
- ▢ Ghi lưu và đặt tên cho Form (theo quy ước đã nêu)

## 9.4 THAO TÁC TRÊN CỬA SỔ THIẾT KẾ

### 9.4.1 Thao tác với đối tượng

Muốn thao tác với một hay nhiều đối tượng: chọn đối tượng ấy, dấu quai bao hiện lên đối tượng được chọn.

#### 9.4.1.1 Cách chọn đối tượng

- ◆ Chọn một đối tượng trên Biểu mẫu nào: click đối tượng ấy.
- ◆ Bỏ chọn: click vùng trống của Biểu mẫu.
- ◆ Chọn nhiều đối tượng gần nhau:

Dùng mouse vẽ khung bao quanh các đối tượng ấy.

- ◆ Chọn nhiều đối tượng không gần nhau:  
Nhấn giữ phím SHIFT và lần lượt click các đối tượng cần chọn.
- ◆ Chọn một phần của đối tượng:  
Các đối tượng khi được tạo thường có Label kèm theo. Muốn thao tác riêng với Label hay đối tượng, click vào thành phần ấy của đối tượng.
- ◆ Dùng thước để chọn:  
Nhấn mouse và rê trên thước dọc/ngang để chọn các đối tượng. Các đối tượng nằm trên đường thẳng tương ứng hay trong vùng rê của thước sẽ được chọn.
- ◆ Chọn tất cả các đối tượng trên Biểu mẫu  
Chọn menu EDIT – SELECT ALL hay Ctrl + A
- ◆ Chọn Biểu mẫu  
Click nút vuông đen tại giao điểm hai thước ngang dọc của cửa sổ thiết kế

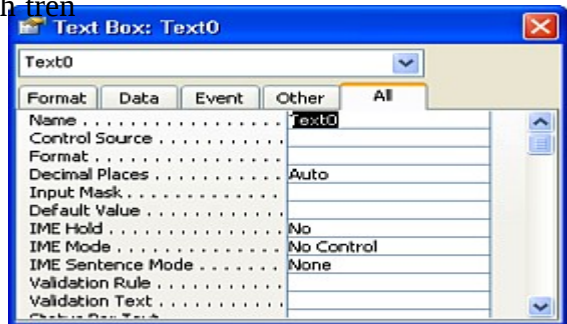
#### 9.4.1.2. Xác lập thuộc tính cho đối tượng

- ◆ Mọi đối tượng đều được gán thuộc tính theo mặc định.

Muốn

thay đổi thuộc tính nào thì xác định lại (gán trị mới) cho thuộc tính đó.

- ◆ Mở cửa sổ thuộc tính
  - Double click lên đối tượng (trong Design View)
  - Hay click phải đối tượng và chọn Properties
  - Hay chọn đối tượng (có thể chọn nhiều đối tượng), rồi chọn menu View - Properties (hay Atl+Enter hay click biểu tượng thuộc tính trên thanh công cụ)
- ◆ Cửa sổ thuộc tính
  - Tên đối tượng hiện trên Title Bar của cửa sổ thuộc tính và hiển thị các thuộc tính liên quan đến đối tượng đó.
  - Nếu chọn nhiều đối tượng thì Title Bar hiển thị Multiple Selection và hiển thị các thuộc tính chung của đối



tượng

- Khi cửa sổ thuộc tính đang mở, nếu chọn đối tượng khác, cửa sổ thuộc tính hiển thị các thuộc tính liên quan đến đối tượng đang được chọn.
- ◆ Xác lập thuộc tính



- Cửa sổ thuộc tính có 5 phiếu
  - y Phiếu Format gồm các thuộc tính liên quan đến định dạng
  - y Phiếu Data gồm các thuộc tính liên quan đến dữ liệu nguồn của đối tượng
  - y Phiếu Event gồm các thuộc tính liên quan đến các sự kiện xảy ra cho đối tượng
  - y Phiếu Other gồm các thuộc tính còn lại khác
  - y Phiếu All hiển thị tất cả thuộc tính của đối tượng
- Mỗi dòng trên cửa sổ thuộc tính là một thuộc tính gồm bên trái là tên thuộc tính và bên phải là trị của thuộc tính
- Xác lập trị mới cho thuộc tính:  
Trên phần bên trái của tên thuộc tính liên quan: nhập/chọn trị trong danh sách/hay click nút 3 chấm (Build) xác định trị mới.

#### 9.4.1.3 Di chuyển, hiệu chỉnh kích thước, xóa, cắt sao chép đối tượng

- ◆ Di chuyển đối tượng  
Chọn đối tượng và rê đến vị trí mới hay dùng phím mũi tên.
  - ◆ Di chuyển một phần đối tượng  
Rê góc trên trái phần liên quan.
  - ◆ Hiệu chỉnh kích thước  
Rê nút bao quanh của đối tượng.
  - ◆ Xóa đối tượng  
Chọn đối tượng, nhấn phím Delete hay chọn menu EDIT DELETE
  - ◆ Sao chép, cắt dán đối tượng  
Chọn đối tượng và sử dụng chức năng Copy, Cut, Paste.
- #### 9.4.1.4 Nhập chuỗi ký tự hay công thức vào đối tượng
- ◆ Click phần nội dung của đối tượng (hay Label của đối tượng) và nhập chuỗi ký tự hay công thức.
  - ◆ Công thức nhập vào phải khởi đầu bởi dấu bằng (=)

#### 9.4.1.5 Đóng hàng, nhóm các đối tượng

- Chọn các đối tượng liên quan, chọn menu FORMAT và chọn lệnh thích hợp hay click biểu tượng tương ứng trên thanh định dạng (Formatting Toolbar)
- ◆ Snap To Grid: bật tắt chế độ bắt dính vào lưới
  - ◆ Align: đóng hàng

- 
- ◆ Horizontal/Vertical Spacing: hiệu chỉnh khoảng cách ngang/đọc giữa các đối tượng
  - ◆ Group/UnGroup: nhóm/rã nhóm đối tượng
  - ◆ Bring To Front/Send To Back: xếp đối tượng trước sau
- 9.4.2 Thao tác với các phần (Section) của Biểu mẫu

- Tăng giảm bề cao của Section  
Rê biên dưới của Section (trỏ mouse thành hình mũi tên hai đầu)
- Chọn riêng một Section  
Click vào Section Selector hay click ngay lên thanh tiêu đề của Section đó. Không thể chọn cùng lúc nhiều Section
- Chọn cả Form  
Click vào Form Selector (nút vuông tại giao điểm hai thước ngang dọc)
- Tắt mở Section  
Chọn menu VIEW – click vào tên Section liên quan.

## 9.5 CÁC THUỘC TÍNH CƠ BẢN

### 9.5.1 Thuộc tính của đối tượng

- ◆ Name : Tên của đối tượng
- ◆ Control Source : Nguồn dữ liệu của đối tượng
- ◆ Caption : Nhan của đối tượng
- ◆ Format : Định dạng dữ liệu khi hiển thị
- ◆ Font ... : Các thuộc tính dùng xác định Font cho đối tượng
- ◆ Back Style : Kiểu nền của đối tượng
- ◆ Back Color : Màu nền đối tượng
- ◆ Special Effects : Hiệu ứng nền (Raised, Sunken, Shadow...)
- ◆ Text Align : Canh lề dữ liệu trong đối tượng
- ◆ Decimal Places : Định chữ số thập phân hiển thị trong đối tượng
- ◆ Input Mask : Định khuôn mẫu nhập liệu
- ◆ DefaultValue : Trị mặc định của đối tượng
- ◆ Validation Rule : Ấn định quy tắc kiểm chính
- ◆ Validation Text : Thông báo khi quy tắc kiểm chính bị vi phạm
- ◆ Visible : Hiển thị hay không hiển thị đối tượng
- ◆ Display When : Tình huống hiển thị đối tượng
  - Always: luôn luôn hiển thị
  - Print Only: Chỉ hiển thị khi in ra giấy
  - Screen Only: Chỉ hiển thị trên màn hình
- ◆ Enable : Cho/không cho xâm nhập đối tượng để nhập,

sửa.

- ◆ Locked : Khóa đối tượng không cho hiệu chỉnh

### 9.5.2 Thuộc tính của Biểu mẫu

- ◆ Allow Filters : Cho phép/không cho phép sử dụng lọc dữ liệu
- ◆ Allow Edits : Cho phép/không cho phép hiệu chỉnh dữ liệu
- ◆ Allow Deletions : Cho phép/không cho phép xóa dữ liệu
- ◆ Allow Additions : Cho phép/không cho phép lọc dữ liệu
- ◆ Autoresize : Tự động hiệu chỉnh kích thước để hiện đủ tin

---

◆ Autocenter : Tự động canh giữa khi mở biểu thức

- ◆ Modal : Xác lập trị Yes thì không thể chuyển qua cửa sổ khác khi Biểu mẫu chưa đóng.
- ◆ Pop Up : Xác lập trị Yes khi Biểu mẫu nằm trên các cửa sổ, Biểu mẫu khác
- ◆ Close : Có/không có nút Close
- ◆ Control Box : Hiện/không hiện Control Box
- ◆ Caption : Nhân
- ◆ Default View : Chế độ hiển thị mặc định (Single Form: hiển thị một Record, Continuous Form: Các Record được hiển thị theo chiều ngang và kế tiếp nhau, Datasheet: hiển thị dạng Datasheet)
- ◆ Data Entry : Cho phép nhập thêm dữ liệu mới.
- ◆ Dividing Lines : Đường phân cách giữa các thành phần khi xem Biểu mẫu ở chế độ Form View.
- ◆ Min Max Buttons : Có/không có nút Max/Min
- ◆ Navigation Buttons : Hiện/không hiện các nút di chuyển qua các mẫu tin.
- ◆ Picture : Các thuộc tính Picture của Biểu mẫu dùng xác lập ảnh nền cho Biểu mẫu.
- ◆ Record Source : Nguồn dữ liệu của Form
- ◆ Record Selectors : Hiện/không hiện dấu chỉ định Record hiện hành
- ◆ Scroll Bars : Hiện/không hiện các thanh trượt (Neither, Horizontal Only, Vertical Only, Both)
- ◆ View Allowed : Xác định cho phép xem ở chế độ nào (Form, Datasheet và Both)

## 9.6 CÁC ĐỐI TƯỢNG TRÊN BIỂU MẪU:

### 9.6.1 Nhãn (Label)

- Khi muốn tạo những dòng chữ như tiêu đề, chú giải hoặc thông tin khác trong Biểu mẫu thì nên sử dụng điều khiển nhãn.
- Khi dùng dụng cụ để tạo ô nhãn thì có thể tạo ở vị trí bất kỳ của Biểu mẫu.
- Điều khiển Nhãn luôn là Unbound Control và tĩnh vì dữ liệu nhập vào điều khiển không thay đổi được (trừ phi hiệu chỉnh Biểu mẫu).

- Nhập/thay đổi nội dung vào ô nhãn: Nhập/ hiệu chỉnh thuộc tính

Caption của điều khiển.

#### 9.6.2 Text Box

- Text Box có thể là điều khiển ràng buộc, không ràng buộc hay tính toán.

- Thường được dùng để hiển thị nội dung Field của dữ liệu nguồn hay biểu thức (Điều khiển ràng buộc hay tính toán)

### 9.6.3 List Box – Combo Box

#### 9.6.3.1 Khái quát

- ◆ List Box và Combo Box dùng để hiển thị một dds các mục tùy chọn và người dùng chỉ cần chọn mục cần thiết từ danh sách này để tránh sai lầm khi nhập dữ liệu.
- ◆ Danh sách gồm nhiều dòng và từ một đến nhiều cột, mỗi

dòng

là một mục tùy chọn.

- ◆ List Box là hiển thị các mục tùy chọn. Combo Box mỗi lần hiển

thị một mục được chọn và chỉ khi nào click Dropdown Button thì danh sách mới hiển thị đầy đủ để chọn. Combo Box giúp tiết kiệm diện tích thiết kế.

#### 9.6.3.2 Các loại danh sách trong List Box/Combo Box

List Box/Combo Box có 3 loại tùy thuộc vào cách dùng nguồn dữ liệu để tạo ra các mục tùy chọn gồm

- ◆ Dữ kiện lấy từ Bảng hay Query
- ◆ Dữ kiện tự nhập trực tiếp khi tạo List Box/Combo Box (Value

List)

- ◆ Dữ kiện tạo ra từ các cách khác.

#### 9.6.3.3 Tạo List Box/Combo Box với Wizard

- ◆ Bật công cụ Control Wizard
- ◆ Chọn dụng cụ List Box hay Combo Box và vẽ lên Biểu mẫu.
- ◆ Khi trả mouse, Wizard hiển thị các bước như sau
  - Chọn cách tạo danh sách các mục tùy chọn



- Tùy theo mục chọn (tự tạo danh sách trị hay lấy từ Table/Query), các bước tiếp theo tương tự như tạo Lookup cho Field.

- ◆ Tạo xong List Box/ Combo Box, mở cửa sổ thuộc tính và xác

---

lập bổ sung

- Để chỉ định tại cột thứ mấy của danh sách được cập nhật vào

Field, khai báo tại thuộc tính Bound Column.



- Để chỉ định Field nhận giá trị được chọn, khai báo tên Field tại thuộc tính Control Source của điều khiển. Nếu không cập nhật vào Field nào thì để trống thuộc tính này.
- Các thuộc tính khác cũng tương tự như Lookup của Field.

9.6.3.4 Tạo List Box/ Combo Box không dùng Wizard

- ◆ Tất công cụ Control Wizard.
- ◆ Chọn dụng cụ List Box/ Combo Box và vẽ lên Biểu mẫu.
- ◆ Khai báo các thuộc tính cho điều khiển.

Thuộc tính	Khai báo
Contro Source	Field nào nhận giá trị chọn từ Combo/ List Box
Row Source Type	Bảng/Query (lấy trị từ Bảng/Query) hay Value List (tự nhập trị)
Row Source	Tên dữ liệu nguồn hay Value1; Value2;...(nếu Value là chuỗi ký tự thì đặt trong dấu nháy kép).
Column Count	Số Field hay cột (từ 1 trở lên)
Column Width	Độ rộng cột mỗi cột (cách nhau bởi dấu chấm)
Bound Column	Cột ràng buộc nếu gắn với Control Source
Limit To List	YES (bắt buộc chọn trong danh sách)/NO (ngược lại)

9.6.4 Check Box, Option Box, Toggle Box

□ Có thể tạo các Check Box (hộp kiểm), Option Box (nút chọn), Toggle Box (nút bật tắt) để là ô điều khiển nhập dữ kiện hoặc hiển thị dữ kiện YES/NO. Sự khác biệt giữa các Control này là ở hình thức hiển thị nên có thể tùy chọn theo yêu cầu hay thị hiếu.

□ Thao tác:

Chọn dụng cụ liên quan và vẽ lên Biểu mẫu để tạo điều khiển, hiệu chỉnh Label (hoặc bỏ Label).

□ Với những ô điều khiển này, trị được gán là Yes/True khi được chọn và No. Access gán trị -1 cho True và 0 cho False

□ Thay đổi hình thức nút bật tắt

Có thể ghi dòng chữ hay hình ảnh làm nhãn lên bề mặt nút bật tắt.

- Ghi dòng chữ làm nhãn: nhập chuỗi ký tự vào thuộc tính Caption.

- Dùng hình ảnh làm nhãn: click vào nút Build của thuộc tính Picture và chọn hình trong hộp thoại Picture Builder.

9.6.5 Option Group

□ Tạo nhóm nhiệm ý để trình bày tập hợp các nút chọn nào đó.

□ Nhóm nhiệm ý gồm 1 khung chữ nhật, nhãn và các nút nhiệm ý.

□ Các nút nhiệm ý có thể là Option Button, Check Box, hay

Toggle

Button.

☐ Thao tác:

Chọn dụng cụ Option Group rồi vẽ trên Biểu mẫu để tạo điều khiển rồi thực hiện theo hướng dẫn của Wizard (bật công cụ Control Wizard)

- Bước 1: nhập nhãn cho mỗi mục chọn.
- Bước 2: chỉ định mục chọn mặc nhiên cho nhóm
- Bước 3: chỉ định trị cho mỗi mục chọn.
- Bước 4: chỉ định sử dụng trị được chọn.
- Bước 5: chỉ định loại Option.
- Bước 6: đặt tên cho Nhóm (Caption), Finish để kết thúc.

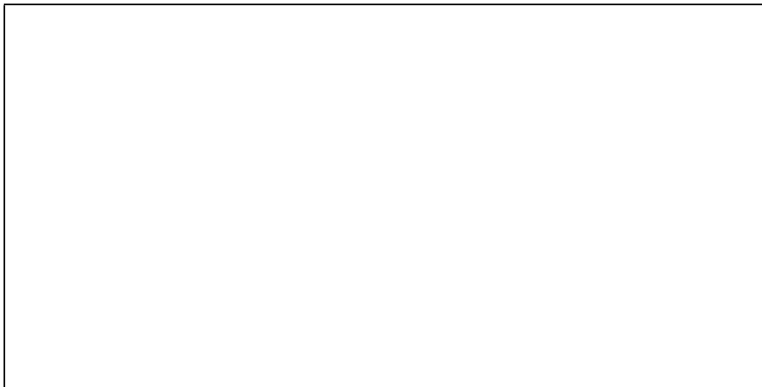
#### 9.6.6 Command Button (nút lệnh)

□ Tạo nút lệnh trong Form để khi sử dụng Access sẽ thi hành một tác vụ hay một dãy tác vụ.

□ Thao tác:

Tạo điều khiển Command Button và thực hiện theo hướng dẫn của Wizard.

- Chọn loại tác vụ trên khung Categories cho Command button (Ví dụ: Record Navigation) và chọn tác vụ trên khung Action (Ví dụ: Find Next)



- Chọn hình (Picture) hay chữ (Text) làm nhãn cho Command button

- Đặt tên cho Command và click Finish.

### 9.6.7 Page Break

- Khi dữ liệu trình bày trên Biểu mẫu nhiều hơn màn hình, có thể dùng điều khiển Page Break để tạo dấu ngắt trang một trang màn hình mới trên Biểu mẫu. Khi in Biểu mẫu, dấu ngắt sẽ bắt đầu trang mới.
- Thao tác
  - Chọn dụng cụ Page Break và click vào vị trí muốn ngắt trên Biểu mẫu (chế độ thiết kế). Dấu ngắt trang sẽ được điền vào phía bên trái của Biểu mẫu.
  - Di chuyển, xóa: chọn dấu ngắt – Rê để di chuyển / nhấn phím Delete để xóa.

### 9.6.8 Tab Control

- Dùng Tab Control để phân chia nội dung hiển thị của Biểu mẫu theo từng Tab (Phiếu)
- Tạo Tab Control:
  - Chọn dụng cụ Form Tab và vẽ trên Form để định khung cho Tab Control.
  - Mở Field List để hiển thị các Field cần đưa vào các Tab.
  - Click chọn Tab (để hiển thị On Top)
  - Chọn các Field liên quan và rê thả vào tab.
- Thêm, xóa Tab: Chọn Tab - click phải - chọn Delete (xóa), Insert (chèn thêm Tab).

### 9.6.9 Đối tượng Rectangle và Line

- Đưa vào Biểu mẫu các hình chữ nhật, đường kẻ (line) để tạo chú ý các thông tin cần chú ý hay để trang trí.
- Thao tác:
  - Chọn công cụ Rectangle / Line và vẽ trên Biểu mẫu.
  - Nếu đối tượng che khuất các Control khác, chọn đối tượng và chọn menu FORMAT – Send To Back để chuyển đối tượng ra phía sau. Ngược lại muốn đối tượng nằm trên thì chọn Bring To Front.

## 9.7 CÁC BỔ SUNG

### 9.7.1 Hiển thị thông tin của Điều khiển trên Status Bar

- Có thể hiển thị thông tin cho một ô điều khiển lên Status Bar khi ô điều khiển này được chọn để hướng dẫn người sử dụng.
- Thao tác: chọn ô điều khiển muốn đặt thông tin và nhập

dòng thông tin (chuỗi ký tự) cho thuộc tính Status Bar Text.

- Dòng thông tin dài tối đa 255 ký tự nhưng khả năng hiển thị tùy thuộc vào loại System Font của Windows và cỡ lớn của cửa sổ Access.

### 9.7.2 Thay đổi thứ tự Tab

□ Khi thiết kế Biểu mẫu, mỗi điều khiển thiết kế được gắn một số thứ tự Tab (thứ tự chuyển từ điều khiển này đến điều khiển khác khi nhấn phím Tab). Thông thường, thứ tự này không phù hợp vì quá trình thiết kế có thêm bớt các điều khiển.

□ Có thể thiết lập lại trật tự Tab theo ý riêng

- ◆ Mở Biểu mẫu ở chế độ thiết kế.
- ◆ Chọn menu VIEW – TAB ORDER - hiển thị hộp thoại Tab

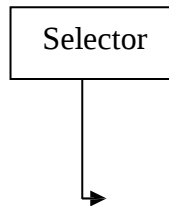
Order.

- ◆ Khung trái hộp thoại gồm 3 mục chọn

- Form Header
- Detail
- Form Footer

Chọn một mục để chỉ định để hiển thị các điều khiển của phần nào trên Biểu mẫu.

- ◆ Khung Phải là CUSTOM ORDER: liệt kê các Field của phần chỉ định và theo thứ tự hiện tại.



- Click Selector để Field muốn thay đổi thứ tự Tab
- rê Selector của Field đang chọn đến vị trí mới
- Nếu chọn mục Auto Order, Access lập thứ tự lần lượt từ trái sang phải rồi từ trên xuống

□ Nếu muốn thao tác với Control nhưng không muốn đưa vào trật tự khi nhấn Tab:

chỉ định NO cho thuộc tính Tab Stop.

□ Thiết lập cách xử lý khi nhấn Tab ở Field cuối của một mẫu tin:



chỉ định  
thuộc tính Cycle của Biểu  
mẫu.

◆ All Records: Đi tiếp đến Field đầu của mẫu tin kế

tiếp

◆ Current Record: Trở lại Field đầu của mẫu tin hiện

hành

### 9.7.3 Điều khiển chứa dữ kiện có nhiều

#### dòng

- Sử dụng điều khiển để hiển thị dữ kiện. Tùy thuộc nội dung dữ kiện mà thiết kế điều khiển đủ lớn để chứa dữ kiện (chiều ngang và chiều cao). Đây là điều khiển chứa dữ kiện có nhiều dòng.
- Trường hợp ô điều khiển không đủ chứa dữ kiện
  - Khi xem có thể dùng phím mũi tên để điều khiển trôi xuống.
  - Hoặc khai báo thuộc tính Scrollbar là Vertical (thanh trượt dọc).

### 9.7.4 Chuyển đổi loại điều

#### kiển

- Có thể chuyển đổi một điều khiển từ loại này sang loại khác (Ví dụ: từ Option Box sang Check Box)
- Thao tác: chọn điều khiển cần chuyển đổi và chọn menu FORMAT – CHANGE TO, chọn loại điều khiển phù hợp (mục nào bị Disable là mục không phù hợp cho chuyển đổi).
- Sau khi chuyển đổi: các tham chiếu vẫn không thay đổi.

### 9.7.5 Sử dụng biểu thức trong Biểu

#### mẫu

Biểu thức là sự kết hợp giữa các toán hạng (hằng, tên Field, tên ô điều khiển, hàm) với các toán tử để tính một giá trị kết quả.

Hàm là một thủ tục mà khi thi hành thì cho một giá trị kết quả tại vị trí quy định (vị trí tham chiếu hàm – Function Reference). Ví dụ hàm NOW() cho kết quả là ngày giờ hiện hành.

- Quy định biểu thức
  - Khi thiết kế Biểu mẫu, thường phải sử dụng biểu thức trong một điều khiển Text Box loại tính toán (Calculated Control) để tính một giá trị mà trong Bảng hay Query không có. Cũng có thể quy định biểu thức trong một ô điều khiển loại khác miễn là ô điều khiển đó có mục thuộc tính Control Source. *Nói cách khác một ô điều khiển có thuộc tính Control Source thì có thể xem là ô thuộc loại Calculated Control.*
  - Cách nhập biểu thức vào Calculated Control là Text Box (đã nêu ở phần trước)
- Quy định trị mặc nhiên cho một điều khiển là ngày/giờ hiện



hàng Nhập vào Control hàm DATE(): ngày hiện hành, NOW(): ngày và giờ hiện hành của máy tính. Ví dụ: =DATE()

#### 9.7.6 Quy định trị số trang của Biểu mẫu

□ Tạo trực tiếp

- Tạo Text Box trên vùng Page Header / Footer của Form
- Mở cửa sổ Property của Control này rồi quy định thuộc tính Control Source là =[PAGE] (dấu bằng khởi đầu).
- *Biểu thức=[Page] không có cặp dấu ngoặc đơn kèm theo vì đây không phải là Hàm của Access*

- Dùng lệnh Insert trên Menu Bar
- INSERT - PAGE NUMBER - chọn các mục liên quan trong hộp thoại.
- Access tự động tạo Page Header / Footer và chèn một TextBox chứa mã số trang.
- Nếu chọn TextBox này và xem dòng Control Source sẽ thấy biểu thức tương ứng.

#### 9.7.7 Nối chuỗi ký tự

- Muốn giá trị của nhiều vùng ký tự hiển thị trong một ô điều khiển
  - quy định biểu thức kết nối chuỗi.

- Ví dụ: nhập vào Control Source của TextBox biểu thức sau đây:
  - =[HOLOT] & " " & [TENHS] để hiển thị Họ lót và Tên của học sinh.

- Sử dụng ký tự nối chuỗi
  - Nếu dùng ký tự &: Access sẽ xem vùng ký tự có trị NULL như một chuỗi có chiều là không. Nên dùng ký tự này.
  - Nếu dùng ký tự +: Access sẽ xem vùng ký tự có trị NULL thì trị NULL trả về (chứ không xem như một chuỗi ký tự có chiều dài là không)

#### 9.7.8 Sử dụng hàm ISNULL và IIF

- Cú pháp: =IIF(<btL>,<kết quả đúng>,<kết quả sai>)
- Ví dụ:
  - Khi dùng biểu thức để tính các phép tính trên hai giá trị và nếu gặp trường hợp một trong hai tác tử là giá trị NULL thì kết quả là NULL. Có thể sử dụng hàm IIF để kiểm tra giá trị NULL trong một biểu thức.
  - Giả định: có biểu thức =[nợ]-[tiền trả] (tính số tiền còn nợ).

Nếu Field [tiền trả] có trị NULL thì biểu thức trên sẽ có trị NULL. Để kiểm tra trị NULL của biểu thức, sử dụng hàm IIF trong biểu thức:

=IIF(IsNull([nợ]-[Tiền trả]), "Xem lại số liệu",[nợ]-[Tiền trả])

- Hoặc sử dụng hàm NZ() để chuyển đổi trị Null thành không.

### 9.7.9 Tính tổng của một nhóm mẫu tin

□ Hàm SUM dùng để tính tổng một nhóm Record

□ Cú pháp =SUM(<biểu thức>)

Trong đó <biểu thức> có thể là một Field (thuộc Bảng hay Query) hay một biểu thức *nhưng không được dùng tên điều kiện làm đối số cho hàm SUM*

#### 9.7.10 Biểu thức dùng tra cứu (Lookup) một trị trong Bảng/Query

□ Khi thiết kế Biểu mẫu, có thể có nhu cầu hiển thị thuộc Bảng/Query không phải là Bảng/Query cơ sở của Biểu mẫu. Trường hợp này, sử dụng hàm DLOOKUP của Access.

□ Cú pháp

DLOOKUP(<biểu thức>,<Tên Bảng/Query>[,<tiêu chuẩn>])

- Biểu Thức: Tên Field dùng để tra cứu

- Tên Bảng/Query: Tên Bảng/Query chứa Field muốn tra cứu

- Tiêu chuẩn: Tiêu chuẩn lựa chọn dữ liệu. Tham số này là tùy chọn (có thể không quy định) và cách dùng tương tự như trong tác vụ vấn tin (Query) nhằm mục đích giới hạn dữ liệu cần tra cứu. Ví dụ:

=DLOOKUP("[HoTen]","[NV]","[MaNV]=Forms![SP]![MaNV]")

Biểu thức này thực hiện việc tìm trong Bảng NV và cho giá trị

kết

quả là họ tên của nhân viên có mã NV (tên Field MaNV) bằng với trị ô điều khiển MaNV trong Biểu mẫu SP.

#### 9.7.11 Thêm hình ảnh, đồ thị vào Biểu mẫu

□ Chỉ nên đưa hình ảnh, đồ thị hay những đối tượng khác được tạo từ những phần mềm có tiêu chuẩn OLE.

□ Đối tượng chèn có thể thuộc diện liên kết (Linking) hay nhúng

(Embed)

□ Dùng UnBound Object Frame hay Bound Object Frame để chèn

đối tượng:

- Dùng UnBound Object Frame khi đối tượng chèn không có mối liên kết dữ liệu (Ví dụ: Logo): chọn dụng cụ UnBound Object Frame – Access của cửa sổ INSERT OBJECT - chọn Create New hay From File để tạo / chọn đối tượng.

- Dùng Bound Object Frame khi đối tượng thuộc một Field của dữ liệu nguồn: chỉ định Field liên quan tại thuộc tính Control Source.

□ Ngoài ra, còn dùng dụng cụ IMAGE để chèn một hình ảnh vào

Biểu mẫu.

## BÀI TẬP:

1. Mở tập tin QLD.MDB, chọn trang Forms và tạo Form sau đây cho table HOCSINH

CẬP NHẬT HỌC SINH		
Mã học sinh	<input type="text"/>	<input type="button" value="ĐẦU"/>
Họ	<input type="text"/>	<input type="button" value="CUỐI"/>
Tên	<input type="text"/>	<input type="button" value="KẾ"/>
Phái	<input type="text" value="♂"/>	<input type="button" value="TRƯỚC"/>
Nam	<input type="text" value="↓"/>	<input type="button" value="ĐÓNG"/>
Lớp		

2. Tạo Form cho table GIAOVIEN bằng công cụ Auto Form theo dạng Tabular.
3. Tạo Form cho table TONGKET bằng công cụ Auto Form theo dạng Datasheet.
4. Tạo Form cho table LOPHOC bằng công cụ Auto Form theo dạng Columnar. Đóng tập tin QLD.MDB
5. Tạo mới tập tin CSDL để quản lý cửa hàng kinh doanh các loại gạo đặt tên là KDGAO.MDB trong đó tạo các table sau:

KHACHHANG (**MAKH**, TENKH, DIACHI, DIENTHOAI)

LOAIGAO (**MAGAO**, TENGAO, DACDIEM, GIABAN)

HOADON (**SOHD**, NGAYBAN, MAKH, QUATANG)

CTHD (**SOHD, MAGAO**, SOKG, GIAM)

*Ý nghĩa của các table:*

- Table KHACHHANG: lưu danh sách các khách hàng mua gạo. Mỗi khách hàng có các thông tin: MAKH (mã khách hàng), TENKH (tên khách hàng), DIACHI (địa chỉ), DIENTHOAI (số điện thoại của khách hàng).
- Table LOAIGAO: lưu danh mục các loại gạo, gồm: MAGAO (mã gạo), TENGAO (tên gạo), DACDIEM (đặc điểm của gạo), GIABAN (giá bán 1 kg).
- Table HOADON: lưu các hóa đơn bán hàng, gồm SOHD (số hóa đơn), NGAYBAN (ngày bán), MAKH (mã khách hàng mua gạo),

QUATANG (quà tặng nếu có hiện màu xanh, nếu không có thì hiện chữ không có màu đỏ).

- Table CTHD: lưu nội dung chi tiết của hóa đơn, gồm: SOHD (số hóa đơn), MAGAO (mã gạo), SOKG (số kg gạo), GIAM (tiền giảm cho mỗi kg gạo).

Tạo các table, các field có gạch dưới là khóa của table

- Dựa vào dữ liệu mẫu hãy tự chọn kiểu dữ liệu phù hợp cho các field trong table. Qui định các tính chất Format, Input Mask cần thiết.

y Tạo combo box trong table HOADON tại field QUATANG

gồm các giá trị sau: đường, bột ngọt, tằm, nếp.

y Tạo combo box trong table HOADON tại field MAKH tham chiếu đến field MAKH trong table KHACHHANG (combo box hiện 2 cột MAKH và TENKH).

y Tạo combo box trong table CTHD tại field MAGAO tham chiếu đến field MAGAO trong table LOAIGAO (combo box hiện 2 cột MAGAO và TENGAO).

- Cài đặt các ràng buộc: NGÀYBAN nhỏ hơn hay bằng ngày hiện hành, GIABAN từ 4000 trở lên, SOKG phải là số dương, GIAM từ 0 đến 2000.

- Lập mối liên kết giữa các table và nhập dữ liệu mẫu sau đây

KHACHHANG : Table				
	MAKH	TENKH	DIACHI	DIENHONAI
	K001	Nhà hàng Văn Anh	327 Nguyễn Văn Trỗi, PN.	8794342
	K002	Cơm tấm Thuận Kiều	134 Đào Duy Từ, Q10	7563014
	K003	Trần Bình	Chợ Lách, Long An.	
	K004	Phan Văn Thanh	69 Tôn Thất Tùng, Q1, TPHCM	8326751
	K005	Quán cơm Trí Dũng	471 Cách Mạng Tháng 8, Q3	
	K006	Nguyễn Ngọc Hải	162 Nguyễn Du, Q1.	0906789123

**LOAIGAO : Table**

MAGAO	TENGAO	DACDIEM	GIABAN
G01	Tài nguyên	Xốp, ngọt	6,000
G02	Tám thơm	Thơm, dẻo	6,500
G03	Nàng hương	Thơm, dẻo	8,000
G04	Tám xoan	Dẻo, ngọt	7,000
G05	Thơm Thái lan	Thơm, dẻo	8,000
G06	Nếp	Dẻo, ngọt	7,000
G07	Tấm	Xốp, mềm	5,000

HOADON : Table				
	SOHD	NGAYBAN	MAKH	QUATANG
	H001	14/08/06	K002	Không có
	H002	26/08/06	K005	đường
	H003	13/09/06	K001	bột ngọt
	H004	15/09/06	K006	Không có
	H005	20/09/06	K004	nếp
	H006	28/09/06	K001	Không có
	H007	30/10/06	K005	đường

CTHD : Table				
	SOHD	MAGAO	SOKG	GIAM
	H001	G01	200	300
	H001	G03	50	500
	H002	G01	20	0
	H002	G06	10	0
	H003	G05	40	0
	H004	G01	50	0
	H004	G03	150	1000
	H004	G04	30	0
	H005	G02	80	200
	H006	G03	10	0
	H006	G04	30	0
	H006	G07	20	0
	H007	G06	120	0

6. Tạo các truy vấn sau:

- Liệt kê 2 loại gạo được bán nhiều nhất của tháng bất kỳ nhập vào.
- Liệt kê các loại gạo chưa bán trong tháng 8.
- Liệt kê các loại gạo mà có cùng đặc điểm và giá bán
- Liệt kê mã gạo, tên gạo, tổng số kg bán trong tháng 8, tổng số kg bán trong tháng 9.
- Liệt kê các hóa đơn, tên khách hàng mà không có quà tặng nhưng có giảm giá.
- Liệt kê tháng, tên gạo mà có tổng số kg gạo bán nhiều nhất trong mỗi tháng.
- Liệt kê mã khách hàng, tên khách hàng, tổng thành tiền (đã trừ tiền giảm), tên các loại gạo (mỗi loại gạo là một cột và trong các cột này chứa tổng số kg của loại gạo)
- Liệt kê các hóa đơn, ngày bán mà có bán đồng thời 2 loại Tài



nguyên và Nàng hương.

## CHƯƠNG 10

# MAINFORM - SUBFORM

- Khái quát về Main / Subform
- Thiết kế Main / Subform với Wizard
- Tự thiết kế Main / Subform
- Tạo biểu mẫu có 2 cấp Subform
- Định dạng có điều kiện

### 10.1 KHÁI QUÁT VỀ MAIN FORM / SUBFORM

#### 10.1.1 Main Form – Subform

- Trong việc thiết kế Form, có thể thiết kế Form trong Form tức là có một Form chính (Main Form) hiển thị các thông tin đồng thời chứa một Form khác gọi là Form phụ (SubForm). Mỗi form sử dụng dữ liệu nguồn khác nhau, hiển thị đồng thời và có thể thao tác trên cả hai FORM.
- Những mẫu tin dùng trong MainForm và SubForm thường có mối quan hệ với nhau
  - Quan hệ một - một : Một mẫu tin trong Main Form sẽ ứng với một mẫu tin trong SubForm.
  - Quan hệ một - nhiều : Một mẫu tin trong Main Form sẽ ứng với nhiều mẫu tin trong SubForm.

#### 10.1.2 Dạng thức của Main Form / SubForm

- Main Form thường có dạng thức một cột (mỗi lần chỉ hiển thị một mẫu tin)
- SubForm có được thiết kế theo dạng DataSheet View hay Form View. Dạng DataSheet View thường được sử dụng.

#### 10.1.3 Khi nào dùng Main Form / SubForm

- Sử dụng Main Form - SubForm khi muốn dùng một Bảng / Query làm nguồn dữ liệu cho MainForm và một Bảng / Query khác để trình bày mối quan hệ giữa các Record của các Bảng / Query này. Thực tế, Main Form – SubForm được dùng thường xuyên khi trình bày dữ liệu trên Form.
- Khi dùng Main Form - SubForm cần chú ý các vấn đề sau:
  - Có các Bảng/Query có quan hệ theo kiểu một - nhiều : Main Form sẽ sử dụng Bảng/Query bên một và SubForm sử dụng Bảng/Query bên nhiều.
  - Kiểm tra xem các Bảng/Query sử dụng có các Field quan

hệ

hay không. Access sẽ dùng Field đối chiếu (Matching Field hay

Linking Field) các Bảng/Query để xác định những Record được hiển thị trong Main Form và SubForm.

- Nếu sử dụng Field đối chiếu không phải là Primary key hay

Foreign key thì phải lập chỉ mục cho Field này.

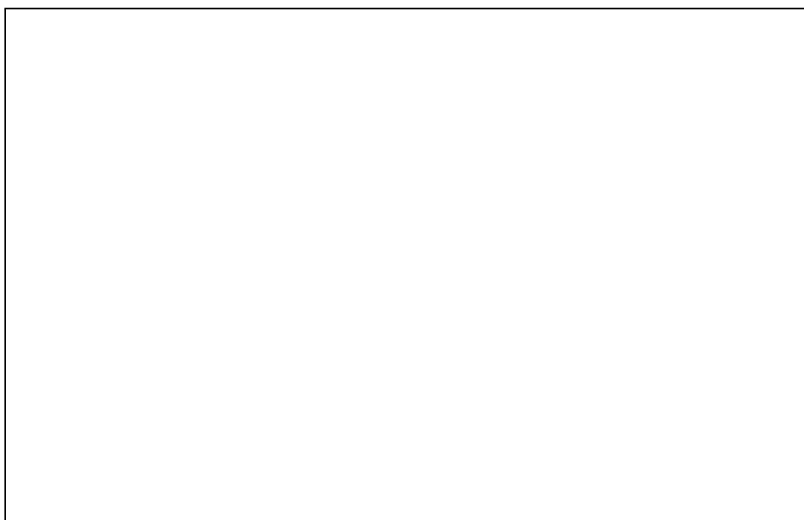
## 10.2 TẠO MAIN FORM - SUBFORM BẰNG WIZARD

### 10.2.1 Khởi động Form Wizard:

- Yêu cầu : Hai dữ liệu nguồn đã khai báo quan hệ.
- Khởi động Form Wizard
  - Mở CSDL và chọn mục loại FORM trên cửa sổ Database, chọn mục FORM WIZARD trên hộp thoại NEW FORM.
  - Thực hiện các bước theo hướng dẫn của Wizard

### 10.2.2 Các bước của Wizard

- Chọn các Bảng/Query và Field tham gia Biểu mẫu

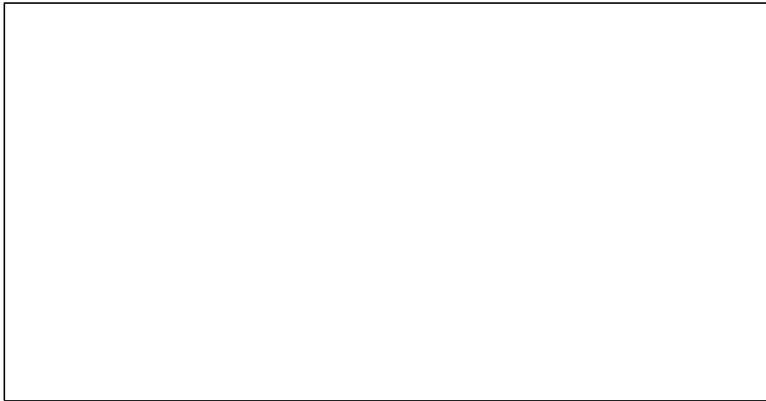


- Chọn Bảng/Query trong khung Bảng/Query, danh sách các Field hiển thị trong khung Available Fields, chọn các Fields sẽ đưa vào Form và chuyển qua khung Selected Fields.
- Lần lượt chọn tiếp Bảng/Query khác chuyển các Field cần thiết vào khung Selected Fields như bước trên.
- Click Next để qua bước kế tiếp.

- Chọn cách hiển thị Form
  - Tại mục How do you want to view your Data?(khung trái).Chọn dữ liệu nguồn cho Main Form (là bên 1 của quan hệ đã khai báo).

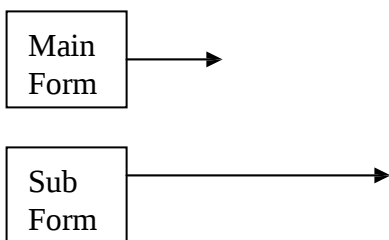
- Rồi chọn mục Form with SubForm(s)

- Khung mẫu hiển thị khái quát Main Form (ở phía trên) và SubForm (ở phía dưới).
- Click Next để qua bước kế tiếp.



- Chọn hình thức hiển thị cho SubForm (Layout)
- Chọn kiểu cho biểu mẫu
- Đặt tiêu đề cho Main Form – SubForm và kết thúc thiết kế

- Đặt tiêu đề cho Main Form trong khung FORM.
- Đặt tiêu đề cho SubForm trong khung SUBFORM..
- Chọn một trong hai để kết thúc thiết kế: Mở Biểu mẫu để xem hay nhập thông tin hay mở cửa sổ Design để hiệu chỉnh Biểu mẫu và Click Finish để kết thúc Wizard.
- Access sẽ ghi lưu Main Form và SubForm thành hai Biểu mẫu riêng biệt trong mục loại Form của cửa sổ Database.



## 10.3 TỰ THIẾT KẾ MAINFORM / SUBFORM

### 10.3.1 Cách chung

- Thiết kế Main Form/SubForm không dùng Wizard thì phải tạo MainForm riêng và SubForm riêng, sau đó chèn SubForm vào MainForm
- Khi tạo MainForm, nhớ dành chỗ trước cho SubForm.
- Trong đa số trường hợp, Access sẽ tự động liên kết SubForm với MainForm. Nếu Access không tự động liên kết được thì người sử dụng phải thực hiện việc này.

### 10.3.2 Thiết kế MainForm

MainForm thường được thiết kế theo dạng Biểu mẫu một cột: mỗi lần chỉ hiển thị một mẫu tin.

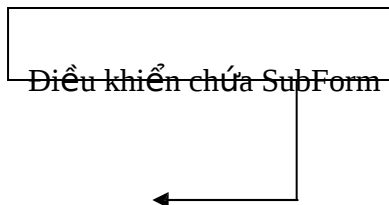
### 10.3.3 Thiết kế SubForm

- Thiết kế SubForm  
SubForm thường được thiết kế theo dạng Biểu mẫu Bảng: Hiển thị nhiều mẫu tin cùng lúc và mỗi dòng là một mẫu tin.
- Xác lập thuộc tính Default View của SubForm là Continuous hay DataSheet.
- Hiệu chỉnh kích thước SubForm cho thích hợp với kích thước vùng trống của Main Form nơi chứa SubForm.
- Kết thúc : Ghi lưu SubForm như các biểu mẫu khác.

### 10.3.4 Chèn SubForm vào Main Form

- Mở Main Form trong chế độ Design View
- Nhấn phím F11 để chuyển qua cửa sổ Database
  - Rê SubForm và thả vào MainForm tại vùng trống đã dành sẵn.
  - Một ô điều khiển tự động được tạo và chứa SubForm.

- Thực hiện các hiệu chỉnh vị trí, kích thước của Điều khiển chứa SubForm



---

**GHI CHÚ:** Nếu mở riêng lẻ SubForm và hiệu chỉnh rồi ghi lưu, Main

Form sẽ được cập nhật theo tình trạng mới của SubForm.

---

#### 10.3.5 Liên kết Main Form với SubForm

- Trong nhiều trường hợp, khi nhúng SubForm vào MainForm thì

Access sẽ thiết lập sự liên kết giữa chúng theo các điều kiện sau:

- Main Form và SubForm đều dùng dữ liệu nguồn là Bảng và quan hệ giữa hai Bảng này đã được thiết lập.
- Main Form dùng dữ liệu nguồn từ Bảng có là khóa chính (Primary key) và SubForm dùng dữ liệu nguồn từ Bảng cũng chứa Field cùng tên với Field khóa chính của bảng cơ sở của Main Form trên làm khóa cơ bản (có cùng kiểu dữ liệu hay tương thích).
- Access dùng 2 Field có tên giống nhau của 2 Bảng làm liên kết.

- Trường hợp sự liên kết không thiết lập được hay muốn quy định Field khác làm mối liên kết thì phải quy định lại tại thuộc tính LinkMasterFields và LinkChildFields của SubForm.

- Click chọn SubForm và mở cửa sổ thuộc tính
- Thuộc tính LinkChildFields: chỉ định Field liên kết của SubForm. Nếu nhiều Field thì cách nhau bởi dấu chấm phẩy.
- Thuộc tính LinkMasterFields: chỉ định Field liên kết của Main Form .



**Subform/Subreport: HOCSINH Subform** [X]

HOCSINH Subform [v]

Format    Data    Event    Other    **All**

Name . . . . .	HOCSINH Subform	[^]
Source Object . . . . .	HOCSINH Subform	[v]
Link Child Fields . . . . .	LOP	
Link Master Fields . . . . .	LOP	
Status Bar Text . . . . .		[v]

## 10.4 TẠO BIỂU MẪU CÓ HAI CẤP SUBFORM

### 10.4.1 Khái niệm

- ▢ Thông thường, Main Form / SubForm biểu thị mối quan hệ giữa hai dữ liệu nguồn (quan hệ Một - Nhiều). SubForm loại này là SubForm cấp 1.
- ▢ Một số trường hợp cần thiết kể Biểu mẫu biểu thị mối Quan hệ 2 cấp giữa 3 dữ liệu cơ sở.
  - Dữ liệu của Main Form quan hệ một - nhiều với dữ liệu của SubForm cấp 1.
  - Dữ liệu của SubForm cấp 1 có quan hệ với Dữ liệu của SubForm cấp 2.
- ▢ Ví dụ: Thiết kế Biểu mẫu biểu thị các mối quan hệ như sau
  - Main Form hiển thị Công trình (Bảng CONGTRINH) và SubForm cấp 1 hiển thị phiếu xuất vật tư cho công trình chỉ định ở Main Form (Bảng XUAT\_VT). Field quan hệ là MACT (mã công trình).
  - SubForm cấp 2 hiển thị các chi tiết của phiếu xuất vật tư (Bảng CHITIET\_XUAT) liên quan đến phiếu xuất có Mã số phiếu xuất chỉ định tại SubForm cấp 1. Field quan hệ là SOPHIEU.

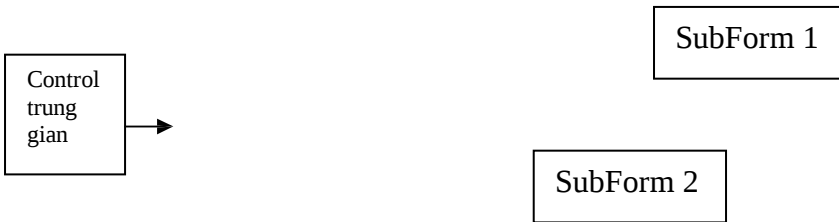
### 10.4.2 Các bước thiết kế

- ▢ Bước 1: Tạo và ghi lưu Main Form và SubForm cấp 1
  - Tạo Main Form và SubForm cấp 1 như quy trình đã trình bày trên.
  - Ví dụ:  
Tạo Main Form tên MAIN - CONGTRINH chứa dữ liệu của Bảng CONGTRINH và SubForm cấp 1 tên SUB1 – XUAT\_VT chứa dữ liệu của Bảng XUAT\_VT .
- ▢ Bước 2: Tạo SubForm cấp 2
  - Tạo SubForm cấp 2 riêng lẻ như tạo một Form thông thường
  - Ví dụ:  
Tạo Form tên SUB2 – CHITIET\_XUAT chứa dữ liệu từ Bảng CHITIET\_XUAT.
- ▢ Bước 3: Nhúng SubForm cấp 2 vào Main/SubForm
  - Mở biểu mẫu MAIN - CONGTRINH trong cửa sổ thiết kế (SubForm cấp 1 là SUB1 – XUAT\_VT cũng được mở theo và hiệu chỉnh cho có chỗ trống để chứa SubForm cấp 2.
  - Nhấn phím F11 để trở về cửa sổ Database.

- Chọn và kéo Form hóa đơn thả vào Main Form tại vị trí dự kiến.

▣ BƯỚC 4: Thiết lập một Control làm trung gian liên kết giữa 2 SubForm

- Dùng ToolBox tạo một UnBound Control (TextBox)
- BỎ nhãn của Control này (Label do Access đưa vào khi tạo TextBox).
- ĐỔI thuộc tính Name của Control (đang là TextBox#) thành Field quan hệ của SubForm 1 với SubForm 2. Theo ví dụ là SOPHIEU
- Nhập công thức vào Control trung gian theo cú pháp sau:  
[Tên SubForm 1].Form![Tên Field liên kết của SubForm 1] Theo ví dụ là : [SUB1 – XUAT\_VT].Form![SOPHIEU]
- Đặt thuộc tính VISIBLE của Control là NO



□ **Bước 5** : Thiết lập liên kết giữa 2 SubForm

- Chọn SubForm 2 - mở cửa sổ Properties.
- Đặt thuộc tính LinkMasterField là tên của Control trung gian Theo ví dụ là SOPHIEU
- Đặt thuộc tính LinkChildField là tên Field liên kết với SubForm cấp 1 Theo ví dụ: LinkMasterField là SOPHIEU

Subform/Subreport: CHITIET Subform

CHITIET Subform

Format	Data	Event	Other	All
Name . . . . .				CHITIET SU
Source Object . . . . .				CHITIET Su
Link Child Fields . . . . .				MACC
Link Master Fields . . . . .				[CHAMCON
Status Bar Text . . . . .				
Visible . . . . .				Yes
Display When . . . . .				Always
Enabled . . . . .				Yes

## □ KẾT QUẢ NHƯ SAU

- Chọn một công trình trên MainForm, SubForm1 hiển thị các phiếu xuất vật tư cho công trình này.
- Chọn một phiếu xuất vật tư trên SubForm 1, SubForm 2 hiển thị

chi tiết liên quan đến phiếu xuất vật tư trên SubForm

1.

**THEO DÕI CÔNG TRÌNH**

Mã công trình:

Tên công trình:

Ngày khởi công:

	SOPHIEU	NGAY	MACT
▶	X001	01/11/2004	CT001
	X002	05/12/2004	CT001
	X003	06/02/2005	CT001

Record:  of 3

**CHI TIẾT XUẤT**

	SOPHIEU	MAVT	TENVT	DVT	SOLUONG	DONGIA	THANHTIEN
▶	X001	CA001	Cát	m3	12	12,000	144,000
	X001	DA002	Đá xanh	tấn	7	314,000	2,198,000
	X001	SA003	Sắt cuộn	kg	400	4,500	1,800,000

Record:  of 3

**TỔNG CỘNG**

## 10.5 ĐỊNH DẠNG CÓ ĐIỀU KIỆN (CONDITIONAL FORMATTING)

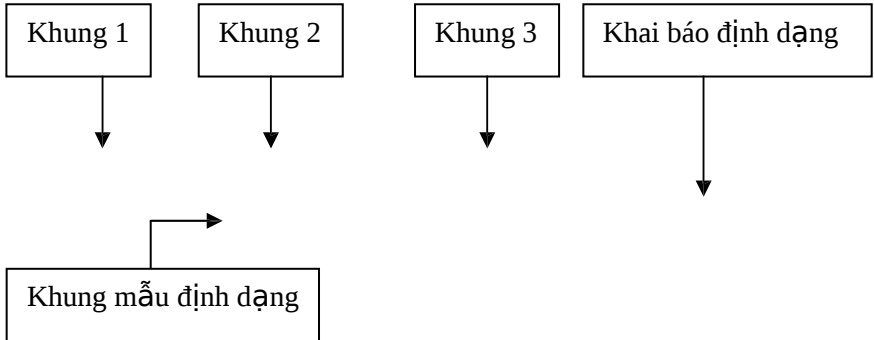
### 10.5.1 Khái quát

- Định dạng có điều kiện cho một hay nhiều đối tượng trên Biểu mẫu là dữ liệu hiển thị của đối tượng được định dạng theo xác lập khi đối tượng hay dữ liệu của đối tượng thỏa điều kiện chỉ định (Chủ yếu là các Text Box mà dữ liệu là Field của dữ liệu nguồn)
- Định dạng có điều kiện cho đối tượng có thể xác lập theo trị của đối tượng, theo kết quả trả về của một biểu thức hay khi đối tượng nhận focus.
- Có thể khai báo định dạng có điều kiện khi mở biểu mẫu ở chế độ Form View hay ở chế độ thiết kế. Tuy nhiên, việc khai báo ở chế độ thiết kế sẽ thuận lợi hơn.
- Được khai báo tối đa 3 điều kiện định dạng cho cùng 1 phạm vi.

### 10.5.2 Khai báo định dạng có điều kiện

- Mở biểu mẫu ở chế độ thiết kế và chọn đối tượng xác lập định dạng.
  - Chọn một hay nhiều đối tượng muốn xác lập định dạng.
  - Nếu là biểu mẫu Tabular, có thể chọn toàn dòng (phần Detail)

- Mở hộp thoại định dạng có điều kiện
- Chon menu FORMAT - CONDITIONAL FORMATTING
- Khai báo định dạng
  - ◆ Khai báo định dạng dựa theo trị của đối tượng

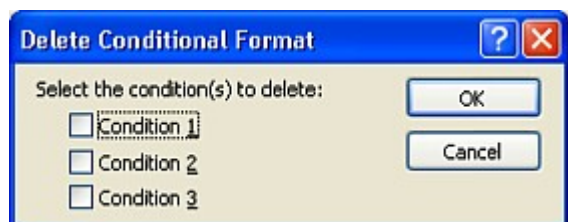


- **Bước 1:** Chọn cách định dạng - Chọn mục Field Value is trong khung 1.
- **Bước 2:** Chọn phép toán trong khung 2 (gồm trong khoảng, ngoài khoảng, bằng,...)
- **Bước 3:** Nhập trị làm tiêu chuẩn trong khung 3. Nếu chọn phép toán là Between hay Not Between thì khung 3 sẽ chia thành 2 khung con để nhận 2 trị giới hạn.
- **Bước 4:** Khai báo định dạng (click vào các biểu tượng định dạng bên phải khung mẫu (gồm đậm, nghiêng, gạch chân, màu chữ, màu nền, kiểu nền)

Khi chưa khai báo định dạng, khuôn mẫu hiện chữ *No Format Set*, sau khi khai báo định dạng, khung mẫu hiển thị theo định dạng khai báo.

- **Bước 5:** Chọn cách kết thúc
  - y Nếu không khai báo điều kiện nào khác : Click OK
  - y Nếu còn điều kiện khai báo thêm : Click ADD và lặp lại các bước trên cho điều kiện mới này (tối đa 3 điều kiện)
- Xóa định dạng có điều kiện:
  - y Chọn lại phạm vi đã khai báo và mở lại hộp thoại

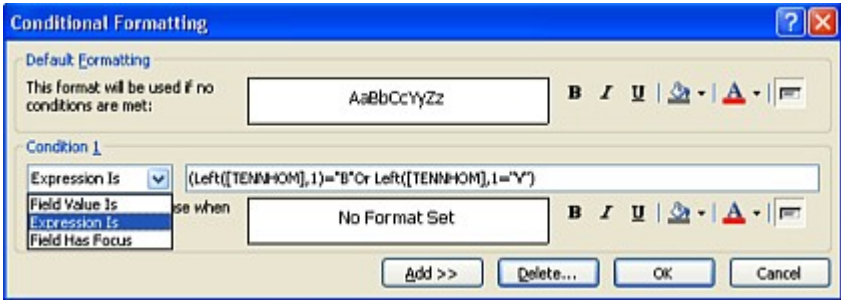
Conditional Formatting  
y Click nút DELETE và chọn điều



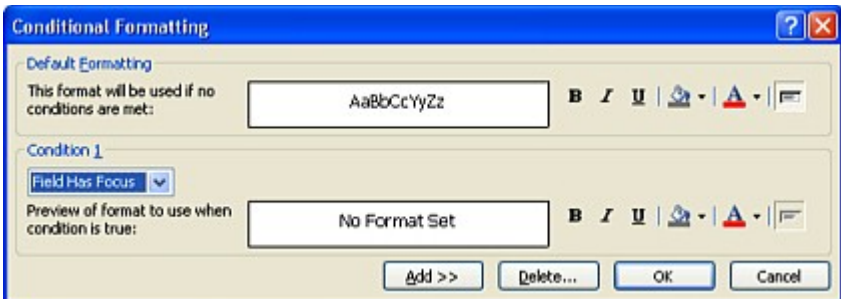


kiện  
muỐn  
xóa rồi click OK.

- ◆ Khai báo định dạng theo kết quả trả về của một biểu thức
  - Chọn mục Expression is trong khung 1
  - Nhập biểu thức trong khung 2 (không có khung 3)
  - Các bước khác cũng như trên.
  - Cách khai báo biểu thức
    - y Là biểu thức Logic trả về trị True / False
    - y Có thể tham chiếu đến nhiều Field trong biểu mẫu



- ◆ Khai báo định dạng cho Field nhận tiêu điểm
  - Khai báo theo cách này, đối tượng được định dạng khi nhận Focus
  - Quy trình:
    - y Chọn mục Field Has Focus trong khung 1
    - y Và khai báo định dạng



## BÀI TẬP:

1. Mở tập tin KDGAO.MDB, chọn trang Forms và tạo form sau (dùng hai table HOADON và CTHD):

SOHD	MAGAO	TENGAO	GIABAN	SOKG	GIAM	THANHTIEN
H001	G01	Tài nguyên	6,000	200	300	1,140,000
H001	G03	Nàng hương	8,000	50	500	375,000

2. Tạo mới tập tin CSDL để quản lý đơn đặt hàng các loại cửa đặt tên là KDCUA.MDB trong đó tạo các table sau:

KHACH\_HANG (**MAKH**, TENKH, DIACHI, DIENTHOAI)

LOAI\_CUA (**MALC**, TENCUA, GIABAN)

DDH (**MADDH**, NGÀYDAT, MAKH, THOIHAN, NGÀYGIAO)

CT\_DDH (**MADDH**, **MALC**, NGANG, CAO, SL)

*Ý nghĩa của các table:*

- Table KHACH\_HANG: lưu danh sách các khách hàng đặt mua cửa. Mỗi khách hàng có các thông tin: MAKH (mã khách hàng), TENKH (tên khách hàng), DIACHI (địa chỉ), DIENTHOAI (số điện thoại của khách hàng).
- Table LOAI\_CUA: lưu danh sách các loại cửa, gồm: MALC (mã loại cửa), TENCUA (tên loại cửa), GIABAN (giá bán 1 mét vuông cửa).
- Table DDH: lưu các đơn đặt hàng của khách hàng, gồm MADDH (mã số đơn đặt hàng), NGÀYDAT (ngày đặt hàng), MAKH (mã khách hàng mua cửa), THOIHAN (thời hạn thực hiện, tính theo ngày), NGÀYGIAO (ngày giao thực tế, nếu hàng chưa giao thì chứa giá trị rỗng).
- Table CT\_DDH: lưu nội dung chi tiết của đơn đặt hàng, gồm: MADDH (mã số đơn đặt hàng), MALC (mã loại cửa),

NGANG (chiều ngang cửa), CAO (chiều cao cửa), SL (số lượng cánh cửa).

Tạo các table, các field có gạch dưới là khóa của table

- Dựa vào dữ liệu mẫu thí sinh tự chọn kiểu dữ liệu phù hợp cho các field trong table. Qui định các tính chất Format, Input Mask cần thiết.
  - y Tạo combo box trong table DDH tại field MAKH tham chiếu đến field MAKH trong table KHACH\_HANG (combo box hiện 2 cột MAKH và TENKH).
  - y Tạo combo box trong table CT\_DDH tại field MALC tham chiếu đến field MALC trong table LOAI\_CUA (combo box hiện 2 cột MALC và TENCUA).
- Cài đặt các ràng buộc: NGAYDAT nhỏ hơn hay bằng ngày hiện hành, THOIHAN từ một tuần đến một tháng, NGAYGIAO hoặc là chưa có, nếu có thì phải lớn hơn NGAYDAT, GIABAN từ 100000 trở lên, các field NGANG, CAO, SL là số dương.
- Lập mối liên kết giữa các table và nhập dữ liệu mẫu sau đây:

	<b>MAKH</b>	<b>TENKH</b>	<b>DIACHI</b>	<b>DIENTHOAI</b>
	K001	Nguyen Thuy Van	327 Nguyễn Văn Trỗi, PN	8794342
	K002	Le Hai Viet	134 Đào Duy Từ, Q5	7563014
	K003	Tran Binh	805 Xô Viết Nghệ Tĩnh, BT	9637522
	K004	Phan Van Thanh	69 Tôn Thất Tùng, Q1	8326751
▶	K005	Tran Vinh Huy	471 Cách Mạng Tháng 8, Q3	
	K006	Nguyen Ngoc Han	562 Điện Biên Phủ, Q3	0906789123

	<b>MALC</b>	<b>TENCUA</b>	<b>GIABAN</b>
	C01	Kính	250,000
	C02	Nhựa	180,000
	C03	Nhôm	350,000
	C04	Gỗ	500,000
	C05	Sắt	300,000

**DDH : Table**

	<b>MADDH</b>	<b>NGAYDAT</b>	<b>MAKH</b>	<b>THOIHAN</b>	<b>NGAYGIAO</b>
	D001	14/08/06	K002	10	25/08/06
	D002	26/08/06	K005	15	
	D003	13/09/06	K001	15	30/09/06
	D004	15/09/06	K006	7	21/09/06
	D005	20/09/06	K004	10	
	D006	28/09/06	K001	30	

CT_DDHH : Table					
	MADDH	MALC	NGANG	CAO	SL
	D001	C01	0.8	2	2
	D001	C04	1.6	1	4
	D002	C03	1.2	2.2	1
	D002	C05	1	2.4	2
	D003	C02	0.8	2.1	5
	D004	C01	2	1	3
	D004	C04	0.5	1	12
	D004	C05	0.9	2.2	2
	D005	C04	0.7	2.5	4
	D006	C03	2.4	1.2	6
	D006	C04	0.8	2	3
	D006	C05	0.6	1.2	8

### 3. Tạo các truy vấn sau:

- Cho biết hai loại cửa được đặt mua nhiều nhất trong tháng 09/06
- Liệt kê các khách hàng mà họ tên có chữ lót là “Van” nhưng không phải tên “Van”
- Liệt kê các đơn đặt hàng đã quá thời hạn nhưng chưa giao hàng.
- Liệt kê các đơn đặt hàng có đặt đồng thời 2 loại cửa Sắt và Nhôm.
- Liệt kê các đơn đặt hàng, ngày đặt mà có cùng thời hạn thực hiện.
- Liệt kê mã loại cửa, tên cửa, tổng số mét vuông đặt trong tháng 8, tổng số mét vuông đặt trong tháng 9.
- Liệt kê mã khách hàng, tên khách hàng, tên cửa, tổng tiền cửa mà có trị giá cao nhất của mỗi khách hàng đó.
- Liệt kê MAKH, TENKH, tổng số tiền, tên các loại cửa (mỗi loại cửa là 1 cột, trong các cột này chứa tổng mét vuông của loại cửa).

### 4. Tạo form sau đây (dùng table DDH và CT\_DDHH):

**Don dat hang**

MADDH

NGAYDAT

MAKH

THOIHAN

NGAYGIAD

CHI TIET DON DAT HANG

	MALC	TENCUA	GIABAN	NGANG	CAO	SL	THANHTIEN
▶	C01	Kính	2.50,000	0.8	2	2	800,000
	C04	Gỗ	500,000	1.6	1	4	3,200,000
*							

TONG CONG

Record:       of 6



5. Mở tập tin QLCT.MDB, chọn trang Forms và tạo form sau (dùng table CONGTRINH, XUAT\_VT và CHITIET\_XUAT)

**THEO DÕI CÔNG TRÌNH**

Mã công trình:

Tên công trình:

Ngày khởi công:

SOPHIEU	NGAY	MACT
X001	01/11/2004	CT001
X002	05/12/2004	CT001
X003	06/02/2005	CT001

Record:  of 3

**CHI TIẾT XUẤT**

SOPHIEU	MAVT	TENVT	DVT	SOLUONG	DONGIA	THANHTIEN
X001	CA001	Cát	m3	12	12,000	144,000
X001	DA002	Đá xanh	tấn	7	314,000	2,198,000
X001	SA003	Sắt cuộn	kg	400	4,500	1,800,000

Record:  of 3

**TỔNG CỘNG**

6. Mở tập tin VPP.MDB, chọn trang Forms và tạo form sau (dùng table KHACH\_HANG, HOADON, CHITIET\_HD)

MAKH:

TENKH:

DIACHI:

DIENHVAI:

**HOA\_DON**

SOHD	NGAYDAT	NGAYGIAO
003	28/1/2006	30/1/2006
006	19/2/2006	22/2/2006

Record:  of 3

**CHITIET\_HD**

MAMH	TENMH	DVT	SOLUONG	DONGIA	THANHTIEN
V03	Giấy vi tính A4	Gam	20	36,000	720,000
V04	Đĩa mềm 1,4MB Maxell	Hộp	100	45,000	4,500,000
V07	Kẹp giấy	Hộp	90	7,000	630,000

Record:  of 3

**TỔNG CỘNG:**

## CHƯƠNG 11

# BÁO BIỂU (REPORT)

- Khái quát về Báo biểu
- Thiết kế Báo biểu với Wizard
- Tự thiết kế Báo biểu

### 11.1 GIỚI THIỆU

#### 11.1.1 Công dụng của Báo biểu (Report)

Tuy có thể in trực tiếp các Bảng/Query/Form nhưng Báo biểu là phương thức trình bày bản in tương đối tốt nhất. Trong Báo biểu, có thể tổ chức thông tin, quy định dạng thức phù hợp với yêu cầu người sử dụng.

#### 11.1.2 Các loại Báo biểu

Có thể phân loại các Báo biểu như sau

- ▣ Column Report: thể hiện các Field của mẫu tin từ trên xuống (cột)
- ▣ Tabular Report: mỗi mẫu tin hiển thị một dòng ngang, các Field xếp theo từng cột.
- ▣ Group/Totals Report: Báo biểu kết số cho một hay nhiều chỉ tiêu số liệu theo từng đối tượng (tổng cộng nhóm).
- ▣ Mailing Label Report: thường dùng để in các nhãn rời cho một mẫu tin.
- ▣ Summary Report: Báo biểu cần lấy kết số cho một hay nhiều chỉ tiêu phân biệt (tương tự loại Group/Totals Report) nhưng không có phần chi tiết (tổng kết).
- ▣ Main/Sub Report: trong Báo biểu có Báo biểu phụ khác.
- ▣ Chart Report: Báo biểu biểu đồ với sự hỗ trợ của MS Graph.

#### 11.1.3 Công việc chuẩn bị

- ▣ Chuẩn bị dữ liệu nguồn của Báo biểu
- ▣ Và xác lập trang in (chọn menu FILE – PAGE SEUP)

#### 11.1.4 Cách tạo Report

Tạo Báo biểu bằng Report Wizard hay tự thiết kế thông qua cửa sổ thiết kế (Design View). Thực tế thường kết hợp hai phương pháp này.

#### 11.1.5 Hiển thị và ghi lưu Report

- ▣ Sau khi hoàn tất thiết kế, cần ghi lưu Báo biểu. Báo biểu được lưu trữ trong mục loại REPORTS của cửa sổ Database và là thành phần của CSDL.

▫ Các chế độ hiển thị:

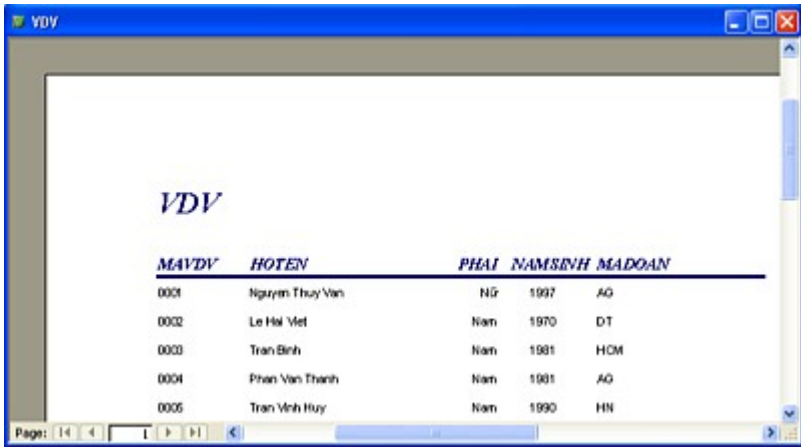
- Design View: chế độ thiết kế để hiệu chỉnh.
- Print Preview: để xem trước kết quả thiết kế.
- Sample View: xem thử.

## 11.2 TẠO REPORT BẰNG WIZARD

Khởi động Report Wizard để bắt đầu tạo Báo biểu: từ cửa sổ Database, chọn mục loại Report, click nút NEW để mở hộp thoại NEW REPORT. Các quy trình cũng tương tự như tạo Biểu mẫu.

### 11.2.1 Dùng Auto Report

- Bước 1: từ cửa sổ NEW REPORT, chọn dữ liệu nguồn cho Report trong khung “Choose the Table or Query where the Object’s data comes from”
- Bước 2: chọn mục AutoReport: Column hay mục AutoReport: Tabular.  
Một Báo biểu được tạo và hiển thị theo chế độ Sample View.

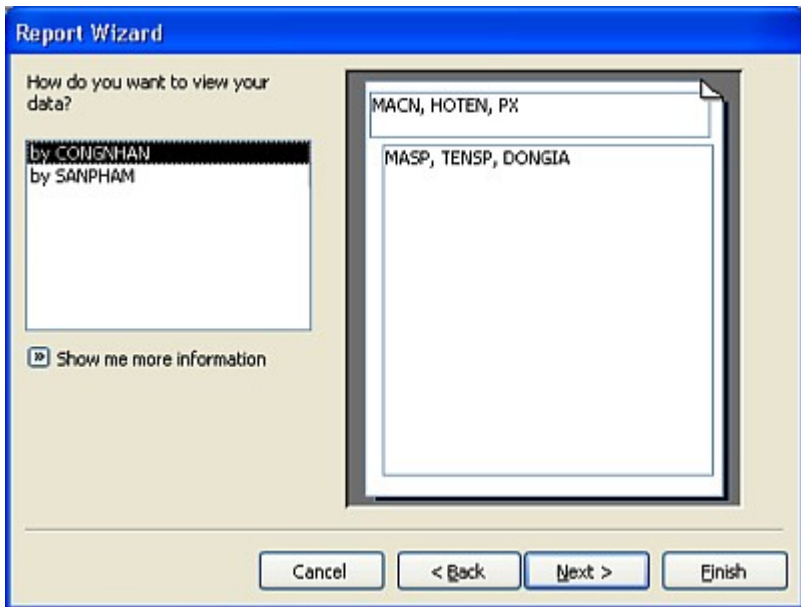


<i>MAVDV</i>	<i>HOTEN</i>	<i>PHAI</i>	<i>NAM SINH</i>	<i>MADOAN</i>
0001	Nguyen Thuy Van	NỮ	1997	AG
0002	Le Hai Viet	Nam	1970	DT
0003	Tran Binh	Nam	1981	HCM
0004	Phan Van Thanh	Nam	1981	AG
0005	Tran Vinh Huy	Nam	1990	HN

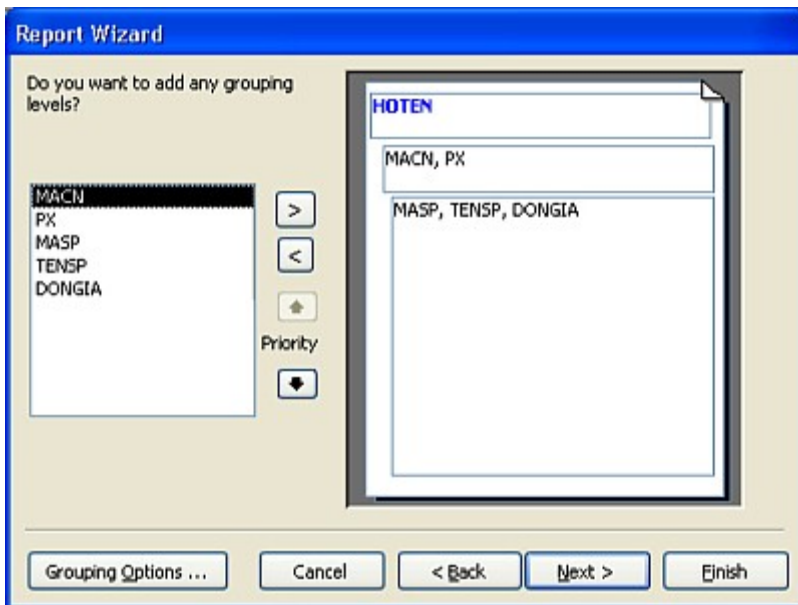
### 11.2.2 Dùng Report Wizard

- Bước 1: khởi động Wizard và chọn dữ liệu nguồn của Báo biểu
  - Từ cửa sổ NEW REPORT, chọn REPORT WIZARD hay click shortcut Create Report by using Wizard
  - Chọn Bảng/Query trong khung Tables/Queries
  - Chọn các Field của Bảng/Query được chọn trong khung Available Fields và chuyển qua khung Selected Fields.
- Bước 2: chọn cách thiết kế Báo biểu

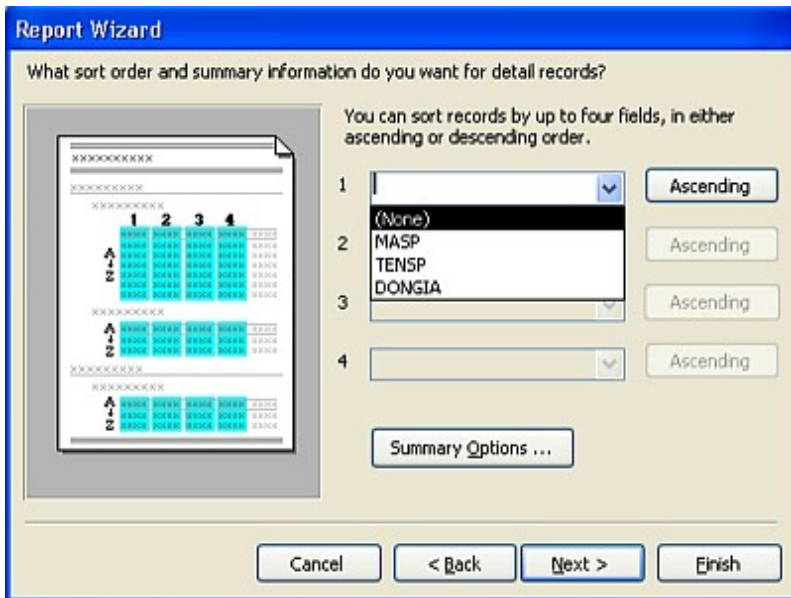
Chỉ hiển thị bước này khi dữ liệu nguồn là một Query được tạo từ nhiều Bảng/Query khác.



- **BƯỚC 3:** khai báo nhóm (tùy chọn)  
 Nếu cần, khai báo các nhóm (Group) cho Báo biểu
- Chọn Field kết nhóm dữ liệu bên khung trái
  - Click dấu > để chuyển qua mẫu
  - Và click nút Group Options để xác lập bổ sung.



- **BƯỚC 4:** Sắp xếp dữ liệu và tạo tổng kết  
 Click nút Summary Options để chọn các mục cho Summary.



- BƯỚC 5: chọn cách trình bày cho bản in (Layout).
- BƯỚC 6: chọn kiểu cho Report (Style)
- BƯỚC 7: đặt tên và kết thúc.

## 11.3 TỰ THIẾT KẾ BÁO BIỂU

### 11.3.1 Cấu trúc của Báo biểu

Các phần (Section) của Report cũng tương tự như của Biểu mẫu và có thêm phần kết nhóm (Group Header/Footer). Sơ đồ tổng quát như hình bên.

- Tiêu đề Báo biểu

- ◆ *Report Header*

(Tiêu đề đầu của

Báo biểu):

Được in một

lần duy

nhất trên

- Report header
  - Page header
    - Group Header cấp 1
      - .....
      - Group Header cấp n
        - Detail
        - Group Header cấp n
          - .....
          - Group Footer cấp 1
  - Page Footer
  - Report Footer

trang đầu tiên của Báo biểu. Thường được dùng để in Tên Công ty, Logo, ngày tháng của báo cáo và trang trí.

◆ *Report Footer* (Tiêu đề cuối của Báo biểu):

Được in một lần duy nhất trên trang cuối của Báo biểu.

Thường được dùng để in những giá trị tổng hợp hay thông tin tổng kết.

□

Tiêu đề trang



- ◆ *Page Header* (Tiêu đề đầu

trang)

ĐƯỢC in ở đầu mỗi trang của Báo biểu. Thường dùng để  
những nội dung lặp lại cho mỗi trang, tiêu đề cột, ...v.v

- ◆ *Page Footer* (Tiêu đề cuối

trang)

ĐƯỢC in ở cuối mỗi trang của Báo biểu. Thường dùng để  
những giá trị tổng hợp.

- Tiêu đề nhóm:

Sử dụng Group Header/Footer khi có nhu cầu kết nhóm dữ  
liệu, ví dụ như in danh sách học viên theo khóa, trong mỗi khóa  
chia theo lớp.

- ◆ *Group*

*Header*

ĐƯỢC in ở đầu mỗi nhóm trước các mẫu tin chi tiết đầu tiên  
của nhóm. Thường phần này được dùng in tên nhóm.

- ◆ *Group*

*Footer*

ĐƯỢC in ở cuối nhóm sau dòng in của các mẫu tin chi tiết  
cùng của nhóm. Thường phần này được dùng để in các giá  
trị  
tổng kết.

- Phần chi tiết (Detail) là phần chủ yếu của Báo biểu dùng  
in dữ

liệu các mẫu tin của dữ liệu  
nguồn.

Mở Report Header/Footer, Page Header/Footer: tương tự  
như  
thao tác với Biểu  
mẫu.

### 11.3.2 Thiết kế Báo biểu

#### 11.3.2.1 Mở cửa sổ thiết kế Báo biểu (Report Design View)

- ◆ Trước khi thiết kế Phải chuẩn bị dữ liệu cơ sở cho Báo

biểu

- ◆ Khởi động Report Design

## View

- Từ cửa sổ Database, chọn mục loại REPORTS và click mục Design View từ hộp thoại New Report.
- Hay click shortcut Create Report in Design View.
- ◆ Khai báo dữ liệu nguồn (trường hợp chưa chỉ định dữ

## liệu

nguồn khi mở cửa sổ thiết kế)

- Chọn đối tượng Báo biểu (click Selector tại giao điểm 2 thước)
- Khai báo dữ liệu nguồn tại thuộc tính Record Source của Báo biểu.

### 11.3.2.2 Hiệu chỉnh bề rộng của

#### Report

- ◆ Trước hết phải hiệu chỉnh trong Page

## Setup

- Phiếu Margin
  - y Xác lập lề trang (Top, Bottom, Left, Right).
  - y Nếu đánh dấu chọn DATA ONLY thì phần Page Header/Footer, Report Header/Footer sẽ không được hiển thị và in ra.

- Phiếu Page: chỉ định kích thước giấy và hướng in.
- Phiếu Column: dùng khi in dạng Mailing Report
  - y Column Munber: chỉ định số cột sắp xếp trên trang giấy.
  - y Row Spacing: khoảng cách dòng giữa các mẫu in
  - y Column Spacing: khoảng cách cột giữa các mẫu in
  - y Column Size: kích thước cột (Width – Height)
  - y Column Layout: chỉ định từ trái sang phải rồi xuống hàng dưới (Across, then Down) hay in từ trên xuống rồi sang cột phải (Down, then Across)
- ◆ Sau đó hiệu chỉnh vùng thiết kế (tương tự thao tác với

Biểu  
mẫu)

#### 11.3.2.3 Tạo các Điều khiển trong Báo biểu

- ◆ Các ô điều khiển trong Báo biểu có thể là Điều khiển ràng buộc, không ràng buộc hay tính toán.
- ◆ Cách tạo, sắp xếp, hiệu chỉnh, định dạng và khai báo thuộc tính:

tương tự như thao tác với Biểu mẫu.

#### 11.3.2.4 Thuộc tính Can Grow và Can Shrink

- ◆ Can Grow (Trị Yes/No)  
Dùng xác lập ô điều khiển có thể tự điều chỉnh kích thước chiều dọc thích hợp với dữ liệu chứa trong điều khiển.
- ◆ Can Shrink (Trị Yes/No)  
Khi trị chứa trong TextBox là một chuỗi rỗng, Access sẽ trống ô điều khiển nên Báo biểu sẽ có những khoảng trống.  
- Xác lập Can Shrink để ngăn các khoảng trống thừa theo chiều dọc nêu trên của Báo biểu.

#### 11.3.2.5 Đánh số thứ tự trong Báo biểu

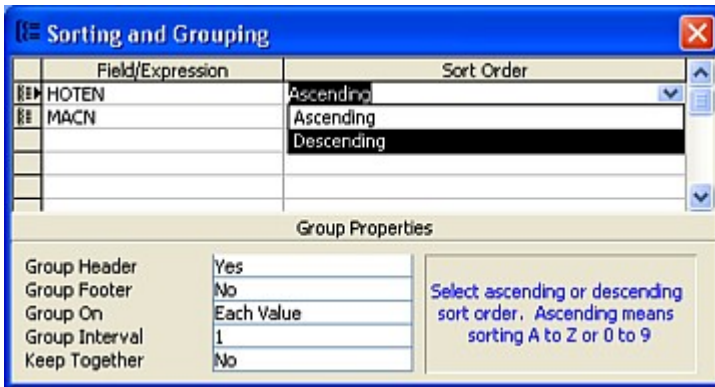
- ◆ Tạo một TextBox tại phần Detail hay đầu nhóm
- ◆ Gán trị cho thuộc tính Control Source là =1 (bằng 1)
- ◆ Xác lập thuộc tính Running Sum: Over Group (đánh số thứ tự cho từng nhóm), Over All (đánh số thứ tự cho tất cả dữ liệu)

## 11.4 XẾP THỨ TỰ VÀ KẾT NHÓM

### 11.4.1 Xếp thứ tự

- Xếp thứ tự để trình bày các mẫu tin trong Report theo một thứ tự nào đó.
- Thao tác
  - ◆ Mở hộp thoại Sorting và Grouping
  - Chọn menu VIEW – SORTING AND GROUPING hay click biểu tượng Sorting And Grouping trên Toolbar.
  - ◆ Xác lập:

- FIELD/EXPRESSION: quy định Field/biểu thức dùng để sắp xếp các mẫu tin.
- SORT ORDER: thứ tự sắp xếp (Ascending, Descending)



## 11.4.2 Kết lập nhóm (Grouping)

### 11.4.2.1 Khái quát

- ♦ Khi trình bày trang in, ngoài việc xếp thứ tự các mẫu tin, thường còn phải kết nhóm (xếp các mẫu tin theo từng nhóm dữ kiện).
- ♦ Access cho phép lập 10 cấp nhóm. Khi thiết lập từ hai cấp

trở

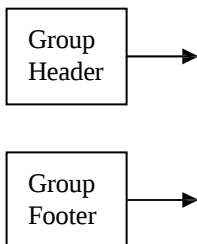
lên, Access sẽ sắp xếp lồng nhóm theo thứ tự từ ngoài vào trong (cấp nhóm 1 ở ngoài cùng).

### 11.4.2.2 Thêm – loại bỏ nhóm

- ♦ Khi xếp thứ tự các mẫu tin, có thể tạo nhóm trong Report bằng cách khai báo thêm phần Group Header/Footer trong cửa sổ thiết kế.
- ♦ Thêm/bỏ nhóm
  - Mở hộp thoại SORTING AND GROUPING
  - FIELD / EXPRESSION
    - γ Chọn Field làm chuẩn kết nhóm
    - γ Nếu có lồng cấp thì chọn Field kết nhóm ở dòng kế tiếp
  - Quy định thuộc tính cho Group Header và Group Footer
    - γ Xác lập cho từng nhóm: YES (tạo nhóm), NO (gỡ bỏ nhóm)
    - γ Access sẽ Bỏ bổ sung hay gỡ bỏ phần Group

Header/ Footer trên cửa sổ thiết kế.

- ♦ Trong phần Group Header, có thể thêm Text box để hiển thị tên nhóm trong phần Group Footer, có thể thêm Text box để hiển thị kết quả tổng hợp nhóm.



#### 11.4.2.3 Quy định phạm vi lập nhóm

- ♦ Mở hộp thoại SORTING AND GROUPING
- ♦ Chỉ định phạm vi lập nhóm tại dòng GROUP ON của phần Property.
  - Nếu Field nhóm thuộc kiểu Text
    - γ EACH VALUE: những mẫu tin có trị Field bằng nhau
    - γ PREFIX CHACRATERS: những mẫu tin có ký tự đầu giống nhau
  - Nếu Field nhóm thuộc kiểu ngày
    - γ EACH VALUE :những record có trị Field bằng nhau
    - γ YEAR : những record cùng năm
    - γ MONTH : những record cùng tháng
    - γ DAY : những record cùng ngày
    - γ WEEK : những record cùng tuần
    - γ QRT : những record cùng quý
    - γ HOUR : những record cùng giờ
    - γ MINUTE : những record cùng phút
  - Nếu Field nhóm thuộc kiểu Number, Currency, AutoNumber:
    - γ EACH VALUE : những record có trị Field bằng nhau
    - γ INTERVAL : những record có trị của Field lập nhóm trong khoảng quy định.

#### 11.4.2.4 Thay đổi thứ tự nhóm



Thay đổi lại thứ tự nhóm trong hộp thoại SORTING AND GROUPING: chọn nhóm muốn thay đổi thứ tự và rê Selector (trước tên nhóm).

#### 11.4.2.5 Xác lập in cho nhóm

- ◆ Mỗi nhóm trong Báo biểu bao gồm Tiêu đề đầu nhóm

(Group

Header), các nhóm con (Nested Groups), phần chi tiết (Detail) và Tiêu đề cuối nhóm (Group Footer).

- ◆ Xác lập in nhóm cho Báo biểu

Mở Báo biểu trong cửa sổ thiết kế và mở hộp thoại

SORTING AND GROUPING.

- Chọn Field nhóm muốn xác lập phạm vi in và xác lập thuộc tính Keep together trong phần Group Properties

γ No: in bình thường và không ngắt trang theo nhóm

γ Whole Group: in tất cả nhóm trên cùng một trang

γ With First Detail: in Tiêu đề đầu nhóm và chi tiết đầu tiên, các dòng chi tiết còn lại được in ở trang sau.

#### 11.4.2.6 Sang trang cho mỗi nhóm

- ◆ Đây là trường hợp muốn mỗi nhóm được in trên trang riêng

- ◆ Xác lập:

- Mở cửa sổ thuộc tính của phần Tiêu đề cuối nhóm

- Xác lập thuộc tính Force New Page là After Section

#### 11.4.3 Main Report/ Sub Report

□ Đây là trường hợp thiết kế Báo biểu chính và có chứa Báo biểu phụ.

□ Quy trình cũng tương tự như thao tác với Biểu mẫu.

## BÀI TẬP:

- Mở tập tin VPP.MDB, chọn trang Reports và tạo report theo mẫu:

### HOA DON

SOHD	001	MAKH	K006
NGAYDAT	14/1/2006	TENKH	Xí nghiệp Đồng Tâm
NGAYGIAO	15/1/2006	DIACHI	212 Hai Bà Trưng, Q1
		DIENTHOAI	8154981

STT	MAMH	TENMH	DVT	DONGIA	SOLUONG	THANHTIEN
1	V04	Đĩa mềm 1,4MB Maxell	Hộp	45,000	50	2,250,000
2	V06	Thước vẽ	Cây	8,500	40	340,000
3	V01	Bút chì	Cây	13,000	100	1,300,000
4	V02	Bút lông viết bằng	Hộp	55,000	40	2,200,000
<i>TONG CONG</i>						6,090,000

- Mở tập tin QLCN.MDB, chọn trang Reports và tạo report sau:

### BANG CHAM CONG

#### Phân xưởng: 1

1 Mã CN: 001

Họ tên: Nguyễn Thụy Vân

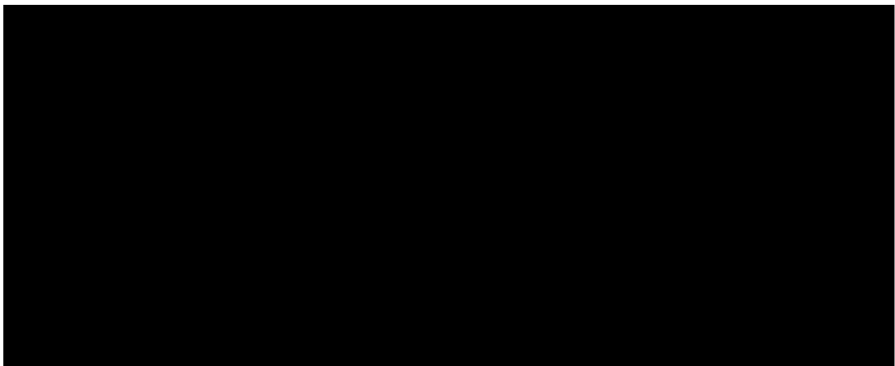
NGAY	MASP	TENSP	DONGIA	SOSP	SOPP	THANHTIEN
14/09/05	N005	Nón	7,000	40	0	280,000
14/09/05	A001	Áo sơ mi nữ	15,000	25	1	367,500
22/10/05	A003	Áo khoác	35,000	45	4	1,505,000
Cộng				110	5	2,152,500

2 Mã CN: 002

Họ tên: Lê Hai Việt

NGAY	MASP	TENSP	DONGIA	SOSP	SOPP	THANHTIEN
14/09/05	A001	Áo sơ mi nữ	15,000	30	0	450,000

- Mở tập tin QLD.MDB, chọn trang Reports và tạo report sau:



## CHƯƠNG 12

# MACRO

- Khái quát về Macro
- Tạo Macro đơn giản
- Các Action của Macro
- Macro điều kiện
- Macro nhóm
- Macro trong Form/ Report
- Các bổ sung

### 12.1 GIỚI THIỆU

#### 12.1.1 Khái niệm về Macro

- Macro là một lệnh hay một nhóm lệnh quy định sẵn được gán cho một tên để mỗi khi được gọi, các lệnh ấy sẽ tự động thực hiện.
- ACCESS cung cấp nhiều hành động Macro cho phép tự động hóa chương trình ứng dụng.
- Macro đặt biệt hữu ích khi tạo những chương trình ứng dụng nhỏ.

#### 12.1.2 Công dụng

Bất cứ tác vụ nào phải thực hiện thường xuyên thì nên tạo một Macro. Dùng Macro để nâng cao tính hiệu quả và Macro luôn thực hiện theo cách đã chỉ định không bao giờ sai lệch.

Có thể sử dụng Macro để:

- Mở Bảng, Truy vấn, Biểu mẫu, Báo biểu trong khung nhìn (View Mode) bất kỳ và đóng các đối tượng này.
- Định giá trị cho các điều khiển của Biểu mẫu, Báo biểu.
- Cũng có thể mô phỏng hành động bàn phím (cung cấp đầu vào hộp thoại hệ thống)
- Thi hành hàm Visual Basic hay Macro khác. Thi hành lệnh trên Menu của Access. Tạo Menu tùy biến.
- Khởi động / thoát chương trình ứng dụng. Và các công dụng khác.

## 12.2 TẠO MACRO ĐƠN GIẢN

### 12.2.1 Cửa sổ Macro

#### 12.2.1.1 Mở cửa sổ Macro

- ◆ Mở cửa sổ để thiết lập Macro

mới :

Chọn mục loại Macro trong cửa sổ Database và click nút NEW trên cửa sổ Database.

- ◆ Mở cửa sổ Macro để hiệu chỉnh Macro đã thiết lập trước đó. Chọn mục loại Macro trong cửa sổ Database, chọn tên Macro cần hiệu chỉnh và click nút DESIGN.

#### 12.2.1.2 Đóng cửa sổ Macro

- ◆ Click nút Close Title Bar
- ◆ Hoặc chọn menu FILE – CLOSE

#### 12.2.1.3 Thành phần cửa sổ Macro

- ◆ Khung trên : Dùng để khai báo các hành động Macro
  - Cột Action: Ghi hành động Macro
  - Cột Condition: Ghi điều kiện Macro
  - Cột Macro Names: Ghi tên Macro (dùng cho Macro nhóm)
  - Cột Comment: Ghi ghi chú hành động Macro tương ứng (mục này không ghi cũng được)

Mặc định, cửa sổ Macro chỉ có cột Action và Comment.

Muốn mở thêm cột Condition và Macro Name thì:

- Chọn menu View - Macro Names hay Conditions.
- Hoặc click biểu tượng trên Toolbar (Macro Names, Conditions).
- ◆ Khung dưới

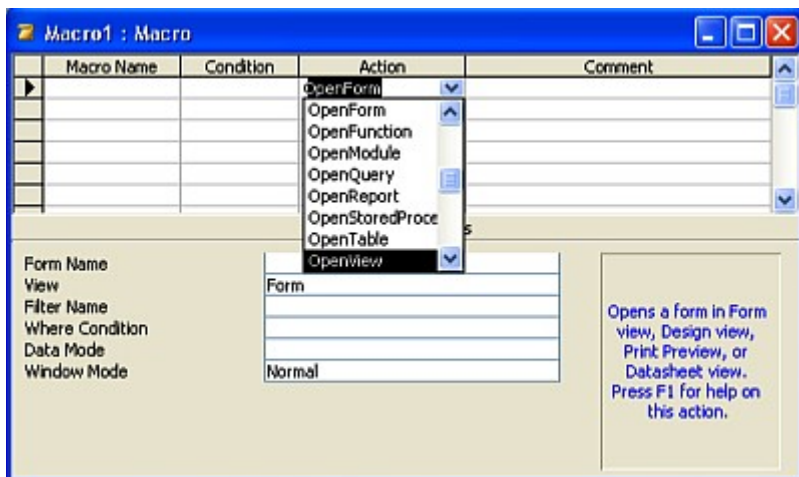
Dùng khai báo các đối số cho hành động Macro. Một hành động (Action) được khai báo, có thể không có đối số hoặc có một /

nhiều đối số.

- ◆ Di chuyển qua lại giữa khung trên và khung dưới: *Nhấn*

*phím*

*F6.*



### 12.2.2 Khai báo hành động Macro

### 12.2.2.1 Nguyên tắc chung

- ◆ Một Macro có thể có một hay nhiều hành động (ACTION).
- ◆ Mỗi hành động được khai báo trên một dòng.
- ◆ Các dòng hành động có thể thiết kế liên tục từ trên xuống hay có những dòng trống giữa các dòng khai báo hành động.
- ◆ Các hành động thực hiện trước được khai báo trước (ở dòng trên), hành động thực hiện sau được khai báo sau (ở dòng dưới).

### 12.2.2.2 Cách khai báo một hành động cho Macro

Gồm khai báo loại hành động và có thể khai báo đối số.

- ◆ Khai báo Action
  - Click vào một dòng trống và click nút Dropdown để mở danh sách hành động Macro (hoặc ALT + phím mũi tên xuống hoặc nhấn phím F4 để mở danh sách các hàng động).
  - Click chọn tên hành động thích hợp (hay dùng phím mũi tên để di chuyển đến hành động thích hợp và ENTER), tên hành động vừa chọn sẽ được ghi vào dòng tương ứng.
  - Cũng có thể nhập trực tiếp hành động vào dòng tương ứng trong cột Action.
  - Nhập chú thích vào cột COMMENT (nếu cần).
- ◆ Khai báo các đối số của Action
  - Tùy theo hành động khai báo, có thể phải khai báo các đối số cần thiết trong khung dưới.
  - Cách khai báo
    - y Nhập trực tiếp nội dung đối số vào các mục liên quan trong phần Action Arguments.
    - y Hoặc click nút Dropdown để chọn đối số trong danh sách.
    - y Hoặc click nút build để mở hộp thoại Expression Builder và thiết lập biểu thức.  
*Nếu không có nút Build hay nút Dropdown - loại Action được chọn không có tác vụ tương ứng.*
  - Tham chiếu đến một điều khiển trong Biểu mẫu hay Báo biểu
    - y Dùng cú pháp sau :  
Forms![tên mẫu biểu]![Tên ô điều khiển] Reports![tên báo biểu]![Tên ô điều khiển]
    - y Khi Macro trong Biểu mẫu hay Báo biểu tham chiếu đến một điều khiển trong chính Biểu mẫu hay Báo biểu

đó : chỉ cần ghi [tên điều khiển] trong quy định tham chiếu  
(tham chiếu tương đối)

#### 12.2.2.3 Chọn dòng hành động đã khai báo

- ◆ Chọn một dòng: Click vào phần Selector ở đầu dòng.



- ♦ Chọn nhiều dòng: Click chọn dòng đầu, nhấn giữ Shift và click chọn dòng cuối hoặc dùng thao tác quét chuột.

#### 12.2.2.4 Chèn dòng trống giữa hai dòng hành động

- ♦ Chọn dòng hành động nơi muốn chèn dòng trống.
- ♦ Click phải và chọn lệnh INSERT ROW hoặc chọn menu

INSERT - ROW

#### 12.2.2.5 Xóa hành động Macro đã khai báo

- ♦ Chọn một hay nhiều dòng hành động muốn xóa.
- ♦ Click phải và chọn lệnh Delete Row.
- ♦ Hoặc chọn menu EDIT - Delete.
- ♦ Hoặc chọn chuỗi ký tự hành động và nhấn phím Delete.

#### 12.2.3 Ghi lưu Macro

- Click biểu tượng Save (đĩa mềm) trên Toolbar (hoặc chọn menu FILE - SAVE) rồi nhập tên Macro vào khung Macro Name của hộp thoại Save As.



- Hay ghi lưu lại với tên mới
- ♦ Chọn menu FILE - SAVE AS
- ♦ Nhập tên mới vào khung Save Macro to <tên Macro mới>.
- ♦ Chọn mục Macro trong khung As.




---

**Ghi chú:** Nếu đặt tên Macro là AUTOEXEC, Access sẽ chạy Macro này mỗi khi mở CSDL. Nếu không muốn chạy AUTOEXEC Macro: Nhấn giữ SHIFT khi mở CSDL.

---

#### 12.2.4 Xóa một Macro

Thực hiện trong cửa sổ Database tương tự như xóa đối tượng khác của CSDL (Bảng, Query, Form, Report,...v.v).

#### 12.2.5 Chạy Macro

Có nhiều cách chạy một

Macro

### 12.2.5.1 Từ cửa sổ Database

Chọn mục loại Macros và chọn tên Macro muốn thực hiện rồi click nút RUN hoặc Dbclick tên Macro.

### 12.2.5.2 Từ cửa sổ thiết kế Macro

- ♦ Chạy tất cả các hành động Macro

Chọn menu RUN - RUN hay click biểu tượng RUN trên Toolbar.

- ♦ Hay chạy từng bước (từng hành động Macro):

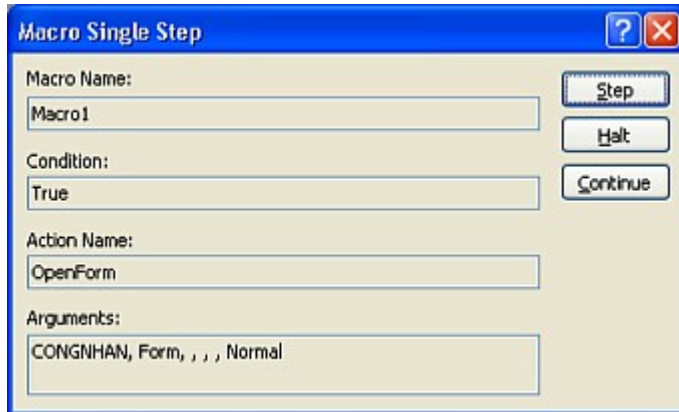
Dùng cách này để chạy thử Macro khi thiết kế. Cách thức như sau :

- Trong cửa sổ thiết kế Macro : click biểu tượng Single Step hay chọn menu RUN - SINGLE STEP.

- Sau đó click nút RUN (hay từ Menu RUN → chọn lệnh RUN) để chạy Action đầu tiên Macro. Hộp thoại MACRO SINGLE

STEP được hiển thị, click một trong các nút sau đây:

- y STEP: thực hiện hành động Macro đang hiển thị trong hộp thoại
- y HALT: ngưng hành động Macro
- y CONTINUE: chạy các hành động trong Macro một cách liên tục (thay vì chạy từng bước).



### 12.2.5.3 Trường hợp gặp lỗi

Hộp thoại MACRO FAILED hiển thị (tương tự như hộp thoại

SINGLE STEP), click nút HALT để ngưng Macro.

## 12.3 HÀNH ĐỘNG VÀ ĐỐI SỐ CỦA MACRO

### 12.3.1 ADD MENU

- Tạo thêm một PullDown Menu vào Menu Bar cho một Form

hay

Report.

□

Đối số:

- ◆ Menu Name: Tên mục chọn ngang muốn thêm vào Menu Bar
- ◆ Menu Macro Name: Tên Macro chứa các lệnh cấu tạo trên PullDown Menu này.
- ◆ Status Bar Text: Thông báo trên Status Bar khi chọn Menu này.

### 12.3.2 APPLY FILTER

- Cho phép lọc dữ liệu khi xử lý Bảng dữ liệu, Biểu mẫu hay Báo biểu.
- Đối số:
  - ◆ Filter Name: Tên Query dùng đọc dữ liệu đang xử lý.
  - ◆ Where Condition: Điều kiện lọc như điều kiện Where trong SQL.

### 12.3.3 BEEP

Phát tiếng BEEP. Không đối số.

### 12.3.4 CANCEL EVENT

Hủy bỏ một biến cố đang thực hiện. Không đối số.

### 12.3.5 Hành động CLOSE :

- Đóng một đối tượng.
- Đối số:
  - ◆ Object Type: Loại đối tượng muốn đóng. *Nếu để trống có nghĩa là cửa sổ hiện hành.*
  - ◆ Object Name: Tên của đối tượng muốn đóng. *Nếu mục Object Type được để trống thì mục này cũng để trống.*
  - ◆ Save: Lưu lại khi đóng. Chọn một trong ba mục sau :
    - Prompt : Nhắc khi lưu
    - Yes : Lưu khi đóng đối tượng
    - No : Không lưu khi đóng đối tượng.

### 12.3.6 COPYOBJECT

- Sao chép đối tượng.
- Đối số:
  - ◆ Destination Database: Tên tập tin CSDL (MDB) nhận sao chép. Để trống nếu sao chép đến CSDL hiện hành.
  - ◆ New Name: Tên của đối tượng muốn Copy. *Để trống mục này nếu copy vào Database khác với Database hiện hành để giữ*

*nguyên tên cũ của đối tượng.*

- ◆ **Source Object Type:** Loại đối tượng nguồn (Bảng, Form,

*Query,...v.v). Để trống mục này nếu copy đối tượng đang chọn trong cửa sổ CSDL.*

- ◆ **Source Object Name:** Tên của đối tượng nguồn. Nếu để trống  
*mục Source Object Type thì để trống mục này.*

### 12.3.7 DELETEOBJECT

- Xóa bỏ đối tượng.
- Đối số:
  - ◆ Object Type: Loại đối tượng muốn xóa. *Nếu để trống có nghĩa là cửa sổ hiện hành.*
  - ◆ Object Name: Tên đối tượng muốn xóa. *Nếu mục Object Type được để trống thì mục này cũng để trống.*

### 12.3.8 ECHO

- Cho phép ẩn hay hiện lại kết quả thực hiện Macro trên Status bar.
- Các đối số:
  - ◆ Echo On: Yes (hiển thị kết quả Macro khi nó được chạy).  
No  
(tắt hiển thị).
  - ◆ Status Bar Text: Nhập dòng văn bản hiển thị khi chạy Macro và mục ECHO ON được chọn NO.

### 12.3.9 FINDNEXT

Tìm mẫu tin kế tiếp thỏa điều kiện trong hành động FindRecord.

### 12.3.10 FINDRECORD

- Tìm mẫu tin đầu tiên thỏa điều kiện chỉ định trong đối số.
- Đối số:
  - ◆ Find What: Nội dung dữ liệu muốn tìm kiếm (Một giá trị hay một biểu thức. Nếu là biểu thức thì phải bắt đầu bởi dấu bằng).
  - ◆ Match: Cách so sánh khi tìm kiếm.
    - Any Part of Field : Giá trị tìm kiếm là phần bất kỳ của Field.
    - Whole Field : Giá trị tìm kiếm bằng giá trị của Field.
    - Start Of Field : Giá trị tìm kiếm là phần đầu tiên của Field
    - ◆ Match Case: Phân biệt chữ thường chữ hoa (Yes) hay không  
(No)
  - ◆ Search: Quy định cách tìm kiếm.
    - All: Toàn bộ Field
    - Up: Tìm từ vị trí mẫu tin hiện hành ngược trở lên mẫu tin

đầu tiên.

- Down: Tìm từ vị trí mẫu tin hiện hành xuôi xuống đến mẫu tin cuối cùng.
- ◆ Search As Formatted: Việc tìm kiếm có theo đúng định dạng

hay không.

- ◆ Only Current Field: Quy định việc tìm kiếm thực hiện trên Field hiện hành (Yes) hay tất cả các Field (No).
- ◆ Find First: Quy định việc tìm kiếm bắt đầu thực hiện tại mẫu tin đầu tiên (Yes) hay mẫu tin hiện tại (No).

#### 12.3.11 GOTOCONTROL

- ▢ Di chuyển điểm chèn đến Control chỉ định trên Form.



- Đối số:
- ◆ Control Name : Tên của điều khiển

#### 12.3.12 GOTOPAGE

- Di chuyển điểm chèn đến Control đầu tiên của trang chỉ định trong Form có nhiều trang (Form có dùng Page Break).

- Đối số:
- ◆ Page Number: Số trang chuyển đến. Nếu để trống là trang hiện

- hành
- ◆ Right : Khoảng cách so với lề trái
- ◆ Down : Khoảng cách so với lề trên.

#### 12.3.13 GOTORECOND

- Chuyển đến mẫu tin chỉ định.

- Đối số:
- ◆ Object Type : Kiểu đối tượng. Để trống nếu là đối tượng

hiện

hành.

- ◆ Object Name : Tên đối tượng
- ◆ Record : Mẫu tin chuyển đến
  - Next, Previous, First, Last, Goto : chuyển đến mẫu tin chỉ định trong Offset
  - New (đến mẫu tin trống cuối dữ liệu).
- ◆ Offset : Nhập một số nguyên n hay biểu thức trả về

một

số nguyên. Nếu Record chọn là GOTO : chuyển đến mẫu tin số mẫu tin là số n ghi trong Offset.

#### 12.3.14 HOURLASS

- Thay đổi dạng thức trở chuột thành Hour Glass khi Macro đang chạy.

- Đối số HourGlassOn: Bật (Yes) / tắt (No) chuyển đổi.

#### 12.3.15 MAXIMIZE / MINIMIZE

Phóng to / thu nhỏ tối đa cửa sổ hiện hành.

#### 12.3.16 MOVESIZE

- Di chuyển hay thay đổi kích thước cửa sổ hiện hành.

- Đối số:

- ◆ Right / Down : Nhập vị trí mới của cửa sổ
- ◆ Width / Height : Bề rộng / cao của cửa sổ.

#### 12.3.17 MSGBOX

- Hiển thị hộp thoại thông báo.
- Đối số:
  - ◆ Message : Nội dung thông báo
  - ◆ Beep     : Phát tiếng Beep (Yes) hay không (No)

- ◆ Type : Loại hộp thông báo (Critical, Warning ?, Warning !, Information).
- ◆ Title : Tiêu đề hộp thoại

### 12.3.18 OPENFORM

- Mở biểu mẫu
- Đối số:
  - ◆ Form Name: Tên biểu mẫu
  - ◆ Form View: Chế độ hiển thị (Form, Design, Print Preview, DataSheet).
  - ◆ FilterName: Tên Query dùng lọc dữ liệu trong Form.
  - ◆ Where Condition: Điều kiện lọc dữ liệu (như Where trong SQL).
  - ◆ Data Mode: Mode dữ liệu (Add, Edit, Read Only).
  - ◆ Window Mode: Chế độ cửa sổ gồm Normal, Hidden, Icon, Dialog (không làm việc với cửa sổ khác nếu chưa đóng lại).

### 12.3.19 OPENMODULE

- Mở Module ở chế độ thiết kế để xem lệnh của hàm hay thủ tục chỉ định.
- Đối số:
  - ◆ Module Name : Tên Module
  - ◆ Procedure Name: Thủ tục / Hàm cần mở

### 12.3.20 OPENQUERY

- Mở hay thực hiện một Query
- Đối số:
  - ◆ Query Name: Tên Query
  - ◆ View: Chế độ hiển thị (DataSheet, Design, Print Preview).
  - ◆ Data Mode: Chế độ dữ liệu (Add, Edit, Read Only).

### 12.3.21 OPENREPORT

- Mở một Báo biểu.
- Đối số:
  - ◆ Report Name : Tên Report
  - ◆ View : Chế độ hiển thị (Print Preview, Print).
  - ◆ Filter Name : Tên Query dùng lọc dữ liệu.
  - ◆ Where Condition : Điều kiện lọc

### 12.3.22 OPENTABLE

- Mở một Bảng
- Đối số:
  - ◆ TableName: Tên Bảng

- ◆ View: Chế độ hiển thị (DataSheet, Design, Print Preview).
- ◆ Data Mode: Chế độ dữ liệu (Add, Edit, Read Only).

### 12.3.23 OUTPUTTO

□ Đưa dữ liệu Access ra ngoài dưới dạng Bảng tính Excel, tài liệu

Word hay trang Web (HTML).

□ Đối số:

- ◆ Object Type: Kiểu đối tượng (Bảng, Query,..)
- ◆ Object Name: Tên đối tượng
- ◆ Output Format: Dạng thức kết xuất (Excel, Word).
- ◆ Auto Start: Tự động mở ứng dụng khi Macro kết thúc.
- ◆ Template File: Tên tập tạm dùng trong quá trình kết xuất dữ liệu.

### 12.3.24 PRINTOUT

□ In đối tượng hiện tại ra máy in.

□ Đối số:

- ◆ Print Range: Phạm vi in (All, Selection, Pages)
- ◆ Page From: Trang bắt đầu
- ◆ Page To: In đến trang
- ◆ Print Quality: Chỉ định chất lượng in.
- ◆ Copies: Chỉ định số bản in
- ◆ Collate Copies: Có sắp xếp, đối chiếu những bản in khi in nhiều bản.

### 12.3.25 QUIT

□ Thoát khỏi ACCESS.

□ Có 1 đối số là Options gồm:

- ◆ Prompt : Nhắc lưu các đối tượng khi thoát.
- ◆ Save All : Tự động lưu khi thoát
- ◆ Exit : Thoát nhưng không lưu các đối tượng.

### 12.3.26 RENAME

□ Ý nghĩa : Đổi tên một đối tượng

□ Đối số:

- ◆ New Name : Tên mới của đối tượng
- ◆ Object Type : Loại đối tượng
- ◆ Old Name : Tên cũ của đối tượng.

### 12.3.27 REPAINTOBJECT

□ Cập nhật màn hình một đối tượng sau quá trình thực hiện các hiệu chỉnh (kể cả việc tính toán lại các điều khiển).

□ Đối số:

- ◆ Object Type : Loại đối tượng

◆ Object Name : Tên đối tượng  
12.3.28 REQUERY

- Cập nhật lại dữ liệu cho một điều khiển trong đối tượng hiện hành.
- Đối số Control Name : Tên điều khiển muốn cập nhật dữ liệu.

### 12.3.29 RESTORE

Phục hồi cửa sổ hiện hành (khi đang cực đại hay cực tiểu).

### 12.3.30 RUNAPP

- Chạy một trình ứng dụng trong môi trường Windows
- Đối số Command line: Tên tập tin chương trình khởi động ứng dụng.

### 12.3.31 RUNCODE

- Thực hiện một thủ tục hàm có trong module của CSDL
- Đối số:
  - ◆ Function Name : Tên hàm / thủ tục. Các đối số nếu có đặt trong cặp ngoặc đơn.

### 12.3.32 RUNCOMMAND

- Thực hiện một số lệnh do ACCESS cung cấp
- Đối số Command: Tên lệnh

### 12.3.33 RUNMACRO

- Chạy một Macro
- Đối số:
  - ◆ Macro Name : Tên Macro
  - ◆ Repeat Count : Số lần Macro lặp lại.
  - ◆ Repeat Expression : Biểu thức chỉ định điều kiện lặp lại. Nếu 2 tham số Repeat được để trống, Macro được thực hiện một lần.

### 12.3.34 RUNSQL

- Thực hiện một câu lệnh SQL Make-Table, Delete, Update, Append.
- Đối số:
  - ◆ SQL Statement: Câu lệnh SQL
  - ◆ Use Transaction: Có sử dụng cơ chế “transaction” hay không.

### 12.3.35 SAVE

- Lưu một đối tượng
- Đối số:
  - ◆ Object Type : Loại đối tượng
  - ◆ Object Name : Tên đối tượng

### 12.3.36 SELECTOBJECT

□ Chọn đối tượng cần xử lý trong CSDL

□ Đối số:

- ◆ Object Type : Loại đối tượng
- ◆ Object Name : Tên đối tượng



- ◆ In Database Window : Xác định đối tượng sau khi được chọn có nằm trong cửa sổ Database hay không.

### 12.3.37 SENDKEYS

- Ghi ra màn hình các phím nhấn.
- Đối số:
  - ◆ KeyStrokes : Các phím nhấn (dùng Visual Basic).
  - ◆ Wait : Có ngưng lại chờ khi phím được nhấn hay không.

### 12.3.38 SETVALUE

- Gán giá trị cho Field, Control, Properties trên Form hay Report.
- Đối số:
  - ◆ Item : Tên Field, Control, Properties muốn gán giá trị
  - ◆ Expression : Giá trị hay biểu thức trả về giá trị gán.
- sau Lưu ý: khi chỉ tên Field, Control, Properties, thì dùng cú pháp
  - ◆ Trong Form: [Forms]![Tên Form]![Tên Field / tên đối tượng]
  - ◆ Trong Report: [Reports]![TênReport]![Tên Field của đối tượng]
  - ◆ Chỉ Properties của một đối tượng:

[Forms/Reports]![Tên Form/Report]![Tên đối tượng].[Tên thuộc tính]

### 12.3.39 SETWARNING

- Bật tắt chế độ thông báo của ACCESS.
- Đối số Warning On : Bật tắt chế độ thông báo

### 12.3.40 SHOWALLRECORDS

Hủy bỏ chế độ lọc dữ liệu - Hiển thị tất cả mẫu tin

### 12.3.41 SHOWTOOLBAR

- Hiện / ẩn Toolbar
- Đối số:
  - ◆ Toolbar Name : Tên Toolbar
  - ◆ Show : Bật / tắt hiển thị.

### 12.3.42 STOPALLMACRO

Ngừng tất cả Macro đang chạy.

### 12.3.43 STOPMACRO

Ngừng Macro đang chạy.

### 12.3.44 TransferDatabase

- Nhập dữ liệu từ table của tập tin CSDL khác vào tập tin CSDL hiện hành hoặc liên kết table của tập tin CSDL khác hoặc xuất dữ liệu từ tập tin CSDL hiện hành ra tập tin CSDL khác.
- Đối số:
  - ◆ Transfer type: Import (nhập)/Export (xuất)/Link (liên kết)

- ◆ Database type: loại tập tin CSDL (Access, ODBC Database, ...)
- ◆ Database name: tên với đường dẫn đầy đủ đến tập tin CSDL

- ◆ Object type: loại đối tượng sử dụng (table, query, ...)
- ◆ Source: tên dữ liệu nguồn
- ◆ Destination: tên dữ liệu đích
- ◆ Structure only: No/Yes (lấy/không lấy dữ liệu)

#### 12.3.45 TransferSpreadsheet

- Nhập hoặc liên kết dữ liệu từ tập tin dạng bảng tính vào tập tin CSDL hiện hành hoặc xuất dữ liệu từ tập tin CSDL hiện hành ra tập tin dạng bảng tính:
  - ◆ Transfer type: Import (nhập)/Export (xuất)/Link (liên kết)
  - ◆ Spreadsheet type: loại tập tin bảng tính (Excel, Lotus, ...)
  - ◆ Table name: tên table để nhập / xuất / liên kết
  - ◆ File name: tên với đường dẫn đầy đủ đến tập tin bảng tính
  - ◆ Has Field names: No/Yes (không/có sử dụng dòng đầu tiên của dữ liệu làm tên field).
  - ◆ Range: địa chỉ vùng dữ liệu trong bảng tính (ví dụ A1:E25)

## 12.4 MACRO ĐIỀU KIỆN

### 12.4.1 Khái niệm

Có thể chỉ định điều kiện cho một hành động của Macro.  
Hành

động này chỉ được thực hiện khi biểu thức điều kiện có trị TRUE.

### 12.4.2 Thiết lập điều kiện cho hành động Macro

- Thêm cột Condition vào cửa sổ thiết kế
    - ◆ Chọn menu VIEW - CONDITIONS hoặc click biểu tượng
- Conditions trên thanh dụng cụ.
- ◆ Cột Conditions được thêm vào cửa sổ thiết kế bên trái
- cột

Action.

□ Khai báo điều kiện cho hành động Macro:

- ◆ Tại dòng hành động Macro tương ứng, nhập vào cột Conditions biểu thức điều kiện cho hành động.
  - ◆ Biểu thức điều kiện phải là biểu thức logic : trả về trị

TRUE /

FALSE.

- ◆ Nếu dùng cùng một điều kiện cho nhiều hành động xếp liên tục:

Chỉ cần khai báo điều kiện cho hành động đầu và dùng 3 chấm (...) để thay thế cho điều kiện của các hành động kế tiếp.

- Cách thực hiện Macro điều kiện  
 Khi điều kiện có trị TRUE, hành động được thực hiện, ngược lại hành động sẽ bỏ qua (và chuyển qua hành động kế tiếp - nếu có).

## 12.5 MACRO NHÓM

### 12.5.1 Khái niệm

- Có thể thiết kế và ghi lưu Macro riêng lẻ nhưng cách này có phần khó quản lý vì có quá nhiều tên Macro trong mục loại Macros ở cửa sổ chương trình.
- Để gọn hơn nên tổ chức Macro theo nhóm. Với một Macro nhóm:

- ◆ Tên nhóm Macro hiển thị trong mục loại Macro của cửa sổ

Database, không hiển thị tên từng Macro trong nhóm.

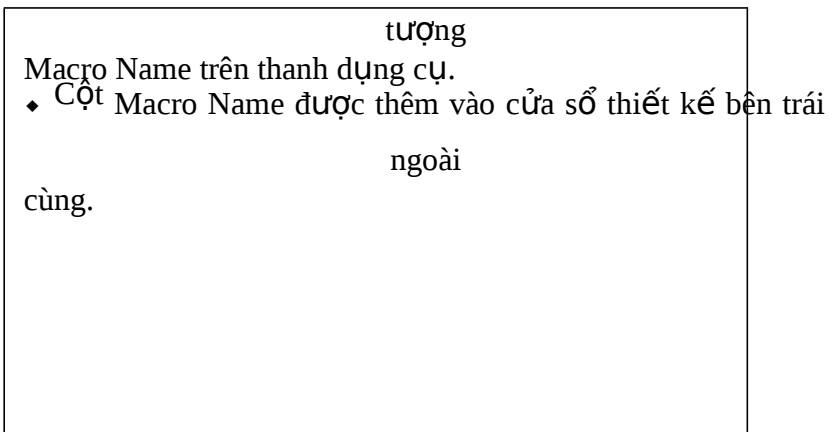
- ◆ Khi thực hiện, chỉ định tên Macro cụ thể trong nhóm.

nếu

không chỉ định thì chỉ Macro đầu tiên được thực hiện.

### 12.5.2 Thiết kế Macro nhóm

- Thêm cột Macro Name vào cửa sổ thiết kế
  - ◆ Chọn menu VIEW - MACRO NAMES hoặc click biểu



- Khai báo các Macro trong nhóm:
  - ◆ Khai báo tên cho mỗi Macro của nhóm trong cột Macro Name.

- ◆ Khai báo các hành động và điều kiện cho mỗi Macro.
  - ◆ Các hành động của một Macro kết thúc khi gặp tên một Macro khác hay kết thúc danh sách hành động trong Macro nhóm.
- Ghi lưu Macro nhóm
- ◆ Thực hiện ghi lưu như ghi lưu một Macro đơn. Tên Macro nhóm hiển thị trong đối tượng Macro của cửa sổ Database.
  - ◆ Tên các Macro trong nhóm không được hiển thị trong cửa sổ này.
- Thực hiện Macro trong nhóm:
- ◆ Chọn tên Macro nhóm và click biểu tượng RUN trong cửa sổ Database hay click phải tên Macro nhóm và chọn lệnh Run
- Chỉ Macro đầu tiên trong nhóm được thực hiện.
- ◆ Muốn thực một Macro cụ thể trong Macro nhóm
    - Chọn tên Macro nhóm liên quan và chọn menu TOOLS - MACRO - RUN MACRO : cửa sổ Run Macro được mở.
    - Click nút Dropdown và chọn tên Macro muốn thực hiện.
- Tên  
Macro gồm tên nhóm và tên Macro (phân cách bởi dấu chấm).
- Tên nhóm Macro.tên Macro

### 12.5.3 Macro AUTOKEYS

- Có thể gán một hay nhiều hành động Macro cho phím bấm. Đây là công dụng của Macro nhóm Autokeys.
- Tạo Macro AUTOKEYS
- ◆ Trong cửa sổ Database, chọn nhóm mục loại Macros
  - ◆ Click NEW để mở cửa sổ thiết kế Macro, hiển thị cột Macro
- Name và cột Condition.
- ◆ Tại cột Macro Name, nhập tên phím bấm hay tổ hợp phím bấm
- (Ví dụ gõ {DEL} chỉ phím Delete, ^A chỉ tổ hợp phím Ctrl+A)
  - ◆ Tại cột Condition, nhập biểu thức điều kiện (nếu cần)
  - ◆ Tại cột Action, chọn tên hành động
  - ◆ Khai báo các đối số cần thiết

□ Ghi lưu : AUTOKEYS

□ Cách ghi phím trong cột Macro Name

Ký hiệu	Ý nghĩa	Ví dụ
^	Phím Ctrl	^A : Ctrl+A

{F1}	Phím F1	
+	Phím Shift	+A : Shift+A ^+A : Ctrl+Shift+A
{DELETE}hay{DEL}	Phím Delete	
+{DEL}	Phím Shift+Del	
^{DEL}	Phím Ctrl+Del	



- Ví dụ:
  - ◆ Gán phím Ctrl+Shift+O để mở Biểu mẫu DANH MỤC LỚP
  - ◆ Gán phím Ctrl+C để đóng Biểu mẫu hiện hành

## 12.6 SỬ DỤNG MACRO VÀO FORM / REPORT

### 12.6.1 Gọi một đối tượng trong

#### Macro

Để gọi một đối tượng của một Biểu mẫu hoặc Báo biểu cần xử lý trong Macro, thì theo quy ước sau đây:

Biểu mẫu : Forms![Tên Form]![Tên đối tượng]

Báo biểu : Reports![Tên Report]![Tên đối tượng]

### 12.6.2 Những tính chất của một biến

#### CỔ

#### 12.6.2.1 Khái niệm biến cố

Biến cố (Event) là những sự kiện xảy ra trên Biểu mẫu hay Báo biểu như nhấn phím, nhấp chuột, di chuyển chuột.

#### 12.6.2.2 Các tính chất

Các tính chất cần sử dụng trong hộp thoại Properties của mục

Event của một đối tượng.

- ◆ Before Update: Quy định Macro hay thủ tục phải thi hành

trước

khi đối tượng được cập nhật.

- ◆ After Update: Quy định Macro hay thủ tục phải thi hành sau khi đối tượng được cập nhật.
- ◆ On Click: Quy định Macro hay thủ tục phải thi hành khi click chuột tại đối tượng.
- ◆ On Dbl Click: Quy định Macro hay thủ tục phải thi hành khi Double click chuột tại đối tượng.
- ◆ On Mouse Down: Quy định Macro hay thủ tục phải thi hành khi nhấn giữ chuột tại đối tượng.
- ◆ On Mouse Up: Quy định Macro hay thủ tục phải thi hành khi nhả chuột tại đối tượng.
- ◆ On Mouse Move: Quy định Macro hay thủ tục phải thi hành khi di chuyển chuột tại đối tượng.
- ◆ On Key Press: Quy định Macro hay thủ tục phải thi hành khi nhấn một phím tại đối tượng.
- ◆ On Key Down: Quy định Macro hay thủ tục phải thi hành khi nhấn giữ một phím tại đối tượng.
- ◆ On Key Up: Quy định Macro hay thủ tục phải thi hành khi

nhả

một phím tại đối tượng.

- ◆ On Enter: Quy định Macro hay thủ tục phải thi hành khi

điểm

nháy (Cursor) vào đối tượng.

- ◆ On Exit: Quy định Macro hay thủ tục phải thi hành khi

điểm

nháy (Cursor) rời khỏi đối tượng.

### 12.6.3 Tạo Menu gắn cho Form / Report

#### 12.6.3.1 Tạo Macro nhóm cho PullDown Menu

- ◆ Tạo Macro nhóm trong đó tên mỗi Macro sẽ là tên mục lệnh của PullDown Menu nằm trên Menu muốn tạo
- ◆ Cần bao nhiêu PullDown Menu thì tạo bấy nhiêu Macro nhóm.

#### 12.6.3.2 Tạo Macro cho Menubar của Form / Report

Tạo một Macro dùng làm Menu Bar cho Form / Report.

Cách tạo như sau :

- ◆ Cột Action : Chọn mục ADD MENU.
- ◆ Khung Action Argument :
  - Menu Name: Nhập tên sẽ hiển thị làm tên của Menu PullDown
  - Menu Macro Name: Tên nhóm Macro dùng làm PullDown Menu.
  - Status Bar Text: Thông báo hiển thị khi đứng tại PullDown Menu.
  - ◆ Cần bao nhiêu PullDown Menu thì tạo bấy nhiêu dòng

Action như trên. Mỗi dòng tương ứng với một Macro nhóm.

#### 12.6.3.3 Gắn hệ thống Menu vào Form / Report

- ◆ Mở Form / Report ở chế độ Design View rồi mở

Form

Properties.

- ◆ Tại thuộc tính Menu Bar: Nhập tên Macro tạo ở bước hai trên (Macro dùng cho Menu Bar).

### 12.6.4 Gắn Macro vào các biến cố trên Form / Report

#### 12.6.4.1 Khi đã có Macro

- ◆ Thường thì một Macro sẽ được thực hiện theo một biến cố nào đó xảy ra trên Form / Report.
- ◆ Muốn chạy một Macro theo một biến cố nào trên Form thì chọn thuộc tính biến cố đó và nhập tên Macro tương ứng.

#### 12.6.4.2 Khi chưa có Macro

- ◆ Đây là trường hợp muốn thực hiện một hàng động tại một biến cố nào đó nhưng hành động này chưa được tạo bằng Macro.
- ◆ Cách giải quyết
  - Tự tạo một Macro cho hành động dự kiến rồi trở lại Form và gắn Macro vào biến cố tương ứng (như trường hợp trên).
  - Hoặc chọn thuộc tính biến cố tương ứng - click nút build (nút có 3 chấm), chọn mục Macro Builder để mở cửa sổ thiết kế macro và tạo Macro.

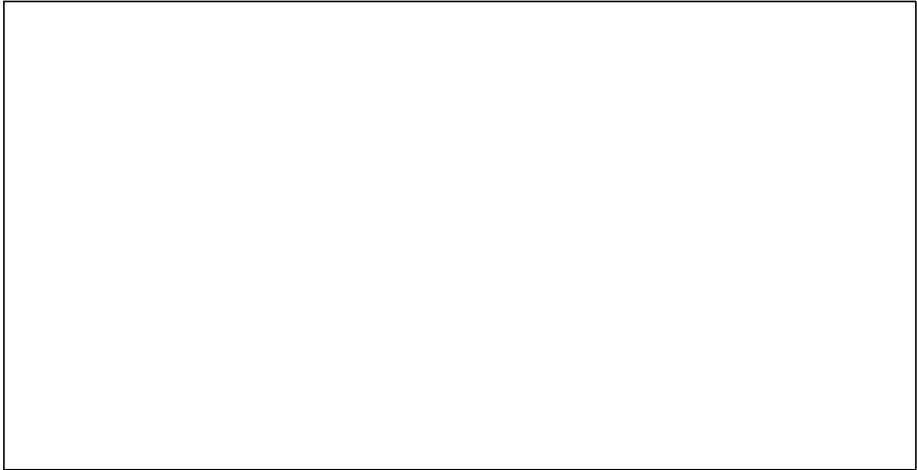
## 12.7 TỰ ĐỘNG MỞ BIỂU MẪU KHI MỞ CSDL

Ngoài việc tạo một Macro với tên là AUTOEXEC để thực hiện các công việc khi khởi động CSDL.

Có thể chỉ định mở một FORM bằng cách xác lập cấu hình Access.

### 12.7.1 Mở hộp thoại Statup

Mở CSDL - chọn menu Tool - Startup...



### 12.7.2 Các xác lập

- ☐ Application Title: Tiêu đề của ứng dụng khi mở tập CSDL (dùng để thay tiêu đề của Access).
- ☐ Application Icon: Chọn tập tin biểu tượng cho ứng dụng.
- ☐ Display Form/Page: Chọn tên Form muốn hiển thị khi mở CSDL.
- ☐ Display Database Window: Hiển thị hay ẩn cửa sổ CSDL.
- ☐ Display Status Bar: Hiển thị hay ẩn thanh trạng thái.

## BÀI TẬP:

1. Mở tập tin QLCN.MDB, chọn trang Forms và tạo form sau:

**Theo dõi chấm công**

Mã công nhân:

Họ tên:

Phân xưởng:

	MAcc	NGAY
▶	1	14/09/05
	9	22/10/05
*		

Record:  of 2

**Chi tiết chấm công**

	MASP	TENSP	DONGIA	SOSP	SOPP	THANHTIEN
▶	A001	Áo sơ mi nữ	15,000	25	1	367,500
	N005	Nón	7,000	40	0	280,000
*						

Tổng cộng:

Tìm công nhân:

- Tạo macro kiểm tra MACN không được trùng. Nếu trùng thì hiện thông báo “Mã công nhân này đã có” và ngăn con trỏ đi đến field kế.
- Tạo macro gắn vào ô textbox Tìm công nhân, nếu sau khi gõ mã công nhân vào mà tìm thấy thì hiện thông tin của công nhân này ra.
- Tạo macro nhóm gắn vào các nút di chuyển trong form.
- Tạo macro gắn vào nút lệnh In, khi bấm vào nút này thì in ra report

### BANG CHAM CONG

2. Mở tập tin QLD.MDB, chọn trang Forms và tạo form sau:

### KẾT QUẢ HỌC TẬP

Lớp   
 Buổi   
 Giáo viên CN

In Bảng điểm

Danh sách Học sinh

	MAHS	HO	TEN
▶	10A1_01	Lê Văn	An
	10A1_02	Trần Văn	Bảo
	10A1_03	Lê Mỹ	Hạnh
*			

Tổng kết điểm

	MON	HK1	HK2	Cả năm
▶	Sinh	8.6	8.1	8.4
	Toán	7.5	6.7	7.1
	Văn	6.1	6.8	6.5
*				



Điểm tổng kết

Xếp loại

3. Chọn trang Reports, tạo report theo mẫu sau:

### BẢNG ĐIỂM

**Lớp:** 10A1

**Giáo viên CN:** Nguyễn Văn An

1 Mã HS: 10A1\_01

Họ tên: Lê Văn An

STT	MÔN HỌC	HOC KY 1	HOC KY 2	CA NAM	XEP LOAI
1	Văn	6.1	6.8	6.5	Trung bình
2	Sinh	8.6	8.1	8.4	Giỏi
3	Toán	7.5	6.7	7.1	Khá
Điểm Tổng kết				7.3	LÊN LỚP

2 Mã HS: 10A1\_02

Họ tên: Trần Văn Bảo

STT	MÔN HỌC	HOC KY 1	HOC KY 2	CA NAM	XEP LOAI
1	Toán	5.4	4.3	4.9	Yếu

4. Sau đó tạo macro gắn vào nút In bảng điểm để in ra report

BANGDIEM đã tạo.

5. Chọn trang Forms, tạo form theo mẫu:

THEO DÕI ĐIỂM DANH

Lớp:

Buổi:

Giáo viên CN:

Tất cả các lớp

In Tổng kết chuyên cần

Danh sách Học sinh

	MAHS	HO	TEN
▶	11B5_08	Đình Thái	Hà
	11B5_09	Lý Văn	Tuấn
	11B5_10	Nguyễn Anh	Kiệt
*			

Điểm danh vắng

	NGAY	PHEP
	22/3/2006	<input type="checkbox"/>
	25/3/2006	<input type="checkbox"/>
✎	13/4/2006	<input checked="" type="checkbox"/>
*		<input type="checkbox"/>

Nghỉ có phép:

Nghỉ không phép:

6. Tạo macro điều kiện gắn vào nút lệnh In Tổng kết chuyên cần, dựa vào check box Tất cả các lớp nếu có chọn check box thì in ra report XEP LOAI REN LUYEN cho tất cả các lớp, ngược lại nếu không chọn check box thì chỉ in cho lớp hiện hành.



## CHƯƠNG 13

# ĐƠN THỂ VÀ THỦ TỤC

- Giới thiệu
- Đơn thể - Thủ tục
- Lập trình với Access Basic
- Các cấu trúc điều khiển

### 13.1 GIỚI THIỆU

#### 13.1.1 Ngôn ngữ lập trình Basic

Basic là ngôn ngữ thảo chương theo kiểu điều khiển bởi sự kiện (event driven programming language) kết hợp với kiểu thảo chương có cấu trúc (structured programming language), và được sử dụng trong hệ quản trị cơ sở dữ liệu Microsoft Access.

#### 13.1.2 Công dụng của việc sử dụng Basic

- Thuận lợi trong việc bảo trì các hệ thống ứng dụng cài đặt trong môi trường Access.
- Xây dựng các hàm tự tạo với các tính năng xử lý chuyên dụng phù hợp cho từng hệ thống ứng dụng.
- Thay đổi hình thức của những thông báo lỗi (error message) do hệ thống phát ra.
- Xây dựng và xử lý các đối tượng trong hệ thống ứng dụng.
- Thực hiện các hành động ở cấp hệ thống như gọi thi hành các hàm trong thư viện liên kết động của Windows (Dynamic Link Libraries – DLL).
- Xử lý nhiều mẫu tin cùng một lúc.
- Chuyển giao các đối mục (arguments) cho chương trình làm tăng sự linh hoạt trong xử lý và điều khiển của chương trình đó.
- Sử dụng ngôn ngữ Basic kết hợp với Macros sẽ tăng hiệu quả các xử lý.

### 13.2 THỦ TỤC VÀ ĐƠN THỂ

#### 13.2.1 Thủ tục (Procedure)

Các chương trình Basic được viết thành những đơn vị riêng lẻ gọi là thủ tục. Trong mỗi thủ tục sẽ chứa các lệnh Basic để thực hiện một xử lý nhất định hoặc tính toán một giá trị. Có

hai loại thủ tục :

### 13.2.1.1 Hàm (Function) :

Là loại thủ tục trả về một giá trị kết quả duy nhất tại vị trí tham chiếu đến hàm đó, vì vậy có thể sử dụng hàm trong các biểu thức. Hàm có thể nhận các đối mục để sử dụng trong chương trình. Tổ chức của hàm như sau:

```
Function Tên_hàm(danh_sách_đối_mục) As kiểu_dữ_liệu
    Lệnh
    .....
    Tên_hàm = biểu_thức
End function
```

### 13.2.1.2 Thường trình phụ (Subroutine)

:

Là loại thủ tục thực hiện các tác vụ nhưng không trả lại một giá trị kết quả nào, vì vậy không thể sử dụng trong các biểu thức. Tương tự như hàm, thường trình phụ cũng nhận các đối mục. Tổ chức của thường trình phụ như sau:

```
Sub Tên_thường_trình_phụ
    (danh_sách_đối_mục) Lệnh
    .....
End Sub
```

Trong các Form và Report, thường phải viết các thường trình phụ và qui định tại các mục tính chất sự kiện của Form hay Report hoặc các ô điều khiển chứa trong đó, cho nên các thường trình phụ này còn được gọi là thủ tục biến cố (Event Procedure). Khi một sự kiện xảy ra trên Form hoặc Report tại một ô điều khiển nào đó thì sẽ gọi thi hành Event procedure đã được qui định tại mục tính chất sự kiện tương ứng.

### 13.2.2 Đơn thể (Module)

Các chương trình Basic được lưu trữ trong các đơn thể, đây là một phương thức tổ chức của những thủ tục. Trong một module sẽ gồm :

- Phần khai báo (Declarations): nằm đầu module, chứa các khai báo biến, khai báo hằng dùng chung và một số lệnh khai báo khác. Trong phần này thường có sẵn lệnh Option Compare Database và Option Explicit.

- Các thủ tục chứa trong đơn thể, phân cách nhau bằng 1 đường kẻ ngang.

```

Option Compare Database
Option Explicit

Public Sub Tinh_DT()
Dim D As Double, R As Double
Dim DT As Double
D = Val(InputBox("Chieu Dai:"))
R = Val(InputBox("Chieu Rong:"))
DT = D * R
MsgBox "Dien tich:" & DT
End Sub

Public Function Tinh_CV()
Const PI = 3.14
Dim BK As Double
Dim CV As Double

```

Đơn thể được chia làm hai loại :

### 13.2.2.1 Đơn thể của mẫu biểu, báo biểu (Form/Report Module) :

Đối với mỗi mẫu biểu hoặc báo biểu trong cơ sở dữ liệu đều được tạo sẵn một đơn thể để lưu trữ các thủ tục biến cố (Event Procedure) và những thủ tục khác dành riêng cho mẫu biểu, báo biểu đó.

Đơn thể của mẫu biểu, báo biểu là một bộ phận thuộc về mẫu biểu, báo biểu đó. Nếu copy mẫu biểu hoặc báo biểu vào một cơ sở dữ liệu khác thì đơn thể cũng được chép theo. Tương tự nếu delete mẫu biểu, báo biểu thì đơn thể thuộc mẫu biểu, báo biểu cũng bị xóa theo.

Các thủ tục trong đơn thể của mẫu biểu, báo biểu chỉ có hiệu lực trong phạm vi của mẫu biểu, báo biểu chứa nó. Các mẫu biểu, báo biểu khác không thể gọi thi hành những thủ tục này.

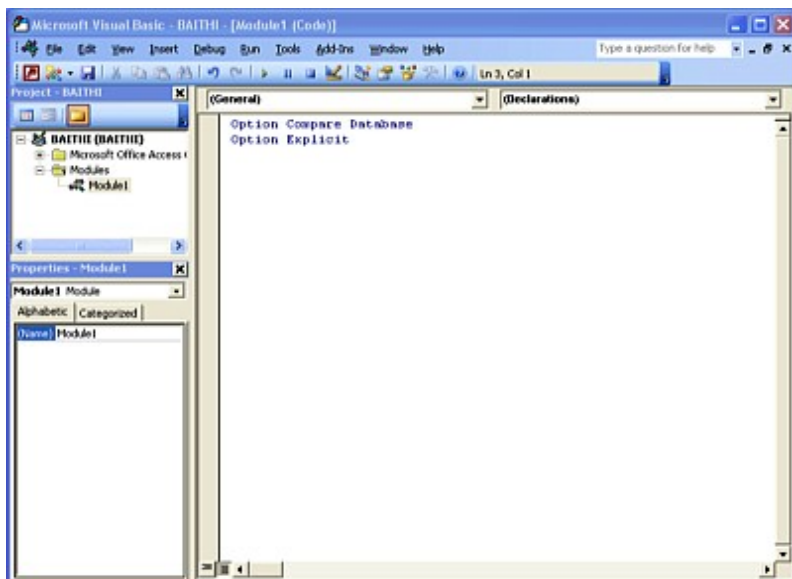
### 13.2.2.2 Đơn thể dùng chung (Global module) :

Là những đơn thể được tạo riêng biệt trong trang Modules của cửa sổ Database. Các đơn thể này chứa những thủ tục có thể gọi thi hành từ bất kỳ tình huống hay đối tượng nào có trong cơ sở dữ liệu đang mở.

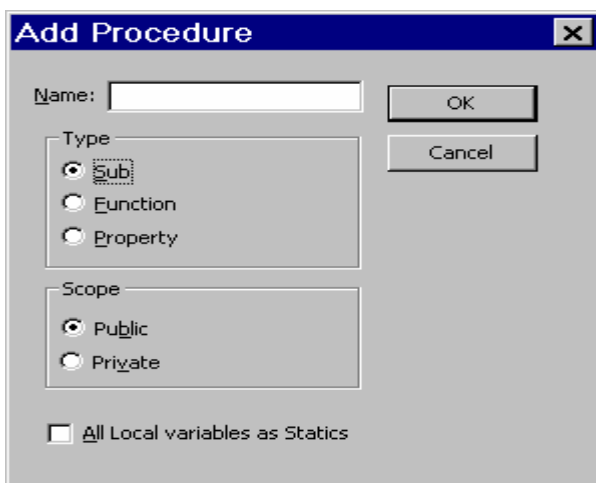
### 13.2.3 Tạo thủ tục

#### 13.2.3.1 Thủ tục dùng chung :

Chọn trang Modules trong cửa sổ Database, sau đó chọn một đơn thể rồi bấm nút Design. Nếu muốn tạo một đơn thể mới chọn nút New.




Để tạo một thủ tục trong cửa sổ đơn thể đang mở, chọn menu Insert, rồi chọn Procedure, sau đó gõ tên thủ tục vào.

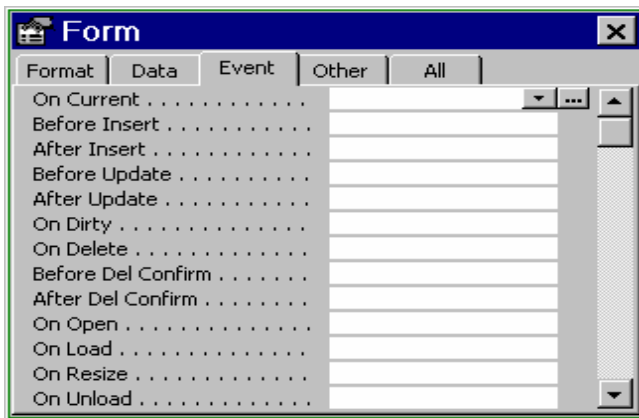


Tên đặt cho thủ tục phải theo qui tắc sau :

- Tên thủ tục bắt đầu bằng ký tự chữ (A - Z)
- Các ký tự còn lại có thể là chữ, số, dấu gạch dưới.
- Không dài quá 40 ký tự.
- Không được trùng với từ khóa dành riêng của Basic.
- Tên đặt cho thủ tục phải duy nhất, không được trùng tên với thủ tục khác trong bất kỳ đơn thể nào thuộc cơ sở dữ liệu, trừ khi thủ tục được khai báo riêng cho đơn thể với từ khóa Private đặt ở trước thủ tục thì có thể trùng tên.

### 13.2.3.2 Thủ tục biến cố (Event Procedure) :

Chọn Form hoặc Report muốn tạo thủ tục biến cố trong đó rồi mở ra ở chế độ Design. Chọn một đối tượng rồi bật ra Properties, sau đó rồi chọn trong trang Event một sự kiện để gắn thủ tục xử lý và bấm nút  chọn Code Builder, rồi gõ nội dung thủ tục vào.



Ngoài ra có thể chọn menu View, rồi chọn Code (hoặc chọn biểu tượng Code trên Toolbar) sẽ xuất hiện cửa sổ Module của Form, Report. Tại ComboBox Object, chọn đối tượng và tại ComboBox Procedure chọn sự kiện mà sẽ gắn thủ tục xử lý, sau đó gõ nội dung thủ tục vào.



### 13.2.4 Soạn thảo các thủ tục

#### 13.2.4.1 Các qui định khi soạn thảo thủ tục :

- Mỗi lệnh đặt riêng từng dòng. Nếu lệnh dài có thể để nhiều dòng, trước khi ngắt câu xuống dòng thì thêm dấu gạch dưới vào cuối dòng.
- Trong thủ tục có thể thêm những dòng chú thích để giải nghĩa chương trình. Mỗi dòng chú thích bắt đầu bằng dấu nháy đơn ‘

#### 13.2.4.2 Các thao tác khi soạn thảo thủ tục :

- Ctrl + Y : cắt dòng hiện tại và chép vào vùng ký ức clipboard.
- Ctrl + V : dán nội dung trong clipboard ra tại vị trí điểm nháy (cursor).
- Ctrl + ↑ : dời điểm nháy đến thủ tục đứng trước.
- Ctrl + ↓ : dời điểm nháy đến thủ tục kế tiếp.

đã F1 : hiện cửa sổ Help đối với từ khóa hoặc đoạn văn bản được đánh dấu khối mà điểm nháy đang đứng tại đó.

- F2 : liệt kê tất cả những đơn thể đã nạp vào bộ nhớ và những thủ tục thuộc đơn thể đang chọn.
- Shift + F2 : chuyển đến thủ tục mà điểm nháy đang đứng ngay tại lệnh gọi thủ tục đó.
- Đánh dấu khối các đoạn văn bản rồi nhấn Tab sẽ dời sang phải

0,5 inch hoặc nhấn Shift+Tab sẽ dời sang trái 0,5 inch.

- Khi soạn thảo các thủ tục trong cửa sổ Module có thể phân chia cửa sổ này để nhìn thấy cùng một lúc hai thủ tục khác nhau trong một đơn thể, bằng cách chọn menu Window, rồi chọn Split. Nếu không muốn phân chia cửa sổ nữa thì chọn lại menu Window, Split lần nữa.

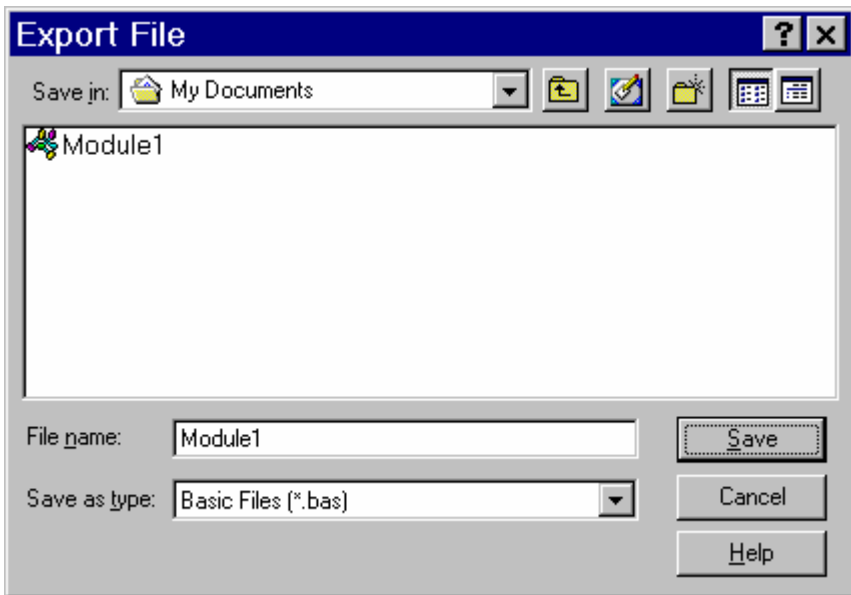
#### 13.2.4.3 Đọc và lưu đơn thể như tập tin văn bản (Text File).

Khi làm việc với cửa sổ Module, có thể đọc một tập tin văn bản vào đơn thể đang soạn thảo hoặc cũng có thể lưu đơn thể này lên đĩa thành một tập tin văn bản. Điều này cho phép dùng một phần mềm soạn thảo văn bản khác để soạn các thủ tục Basic hoặc chuyển các thủ tục Basic cho những phần mềm khác.

- Đọc tập tin văn bản đưa vào đơn thể: đầu tiên mở cửa sổ Module rồi đặt điểm nháy tại vị trí muốn chèn tập tin văn bản vào, sau đó chọn menu Insert, File và chọn tên của tập tin văn bản.

- Lưu đơn thể thành tập tin văn bản: mở cửa sổ Module, chọn menu File, chọn Export File rồi đặt tên cho tập tin văn bản.





#### 13.2.4.4 Sao chép các thủ tục trong các đơn thể:

Có thể sử dụng clipboard để sao chép (Copy) hoặc cắt (Cut), dán (Paste) các thủ tục từ đơn thể này sang đơn thể khác, nhưng không được phép có hai thủ tục tên giống nhau trong các đơn thể dùng chung của một cơ sở dữ liệu.

- Sao chép thủ tục:
  - Mở cửa sổ Module có chứa thủ tục cần chép.
  - Đánh dấu khối toàn bộ nội dung thủ tục (kể cả dòng Sub và End Sub hoặc Function và End Function).
  - Chọn menu Edit, Copy (hoặc nhấn Ctrl+C) thì thủ tục đã đánh dấu sẽ được chép vào Clipboard.
- Cắt thủ tục dời đi:
  - Mở cửa sổ Module có chứa thủ tục cần dời đi.
  - Đánh dấu khối toàn bộ nội dung thủ tục (kể cả dòng Sub và End Sub hoặc Function và End Function).
  - Chọn menu Edit, Cut (hoặc nhấn Ctrl+X) thì thủ tục đã đánh dấu được chuyển vào Clipboard và trên cửa sổ Module thủ tục này được xóa đi.
- Dán thủ tục trong Clipboard vào đơn thể:
  - Mở đơn thể mà muốn dán thủ tục vào.

- Dời dấu nháy đến một dòng trống tại nơi nào mà sẽ dán thủ tục vào.
- Chọn menu Edit, Paste (hoặc nhấn Ctrl+V)

### 13.2.5 Các chức năng trên thanh công cụ của cửa sổ Module



Biểu tượng	Ý nghĩa
View Microsoft Access	Trở lại cửa sổ của Microsoft Access
Insert Module	Tạo 1 module mới. Nhấp vào mũi tên xuống bên cạnh nút này để tạo một class module hoặc chèn một
Save	Lưu các sửa đổi trong module hiện hành.
Cut	Cắt đoạn văn bản đã chọn trong module và đưa vào Clipboard.
Copy	Chép đoạn văn bản đã chọn trong module vào Clipboard.
Paste	Dán nội dung trong Clipboard vào module tại vị trí điểm nháy.
Find	Tìm kiếm 1 từ hoặc cụm từ trong module.
Undo	Hủy bỏ thao tác chuột hoặc bàn phím mới thực hiện nếu được.
Redo	Hủy bỏ Undo mới thực hiện nếu được.
Run Sub/UserForm	Bắt đầu thi hành thủ tục hiện tại hoặc tiếp tục thi hành 1 thủ tục sau khi tạm dừng thi hành bởi 1 điều kiện
Break	Kết thúc việc thi hành 1 thủ tục.
Reset	Kết thúc việc thi hành 1 thủ tục và khởi tạo lại mọi biến theo giá trị ngầm
Design Mode	Chế độ thiết kế.
Project Explorer	Mở cửa sổ Project.
Properties Window	Mở cửa sổ Properties
Object Browser	Mở cửa sổ Object Browser.
ToolBox	Xuất hiện hộp công cụ.

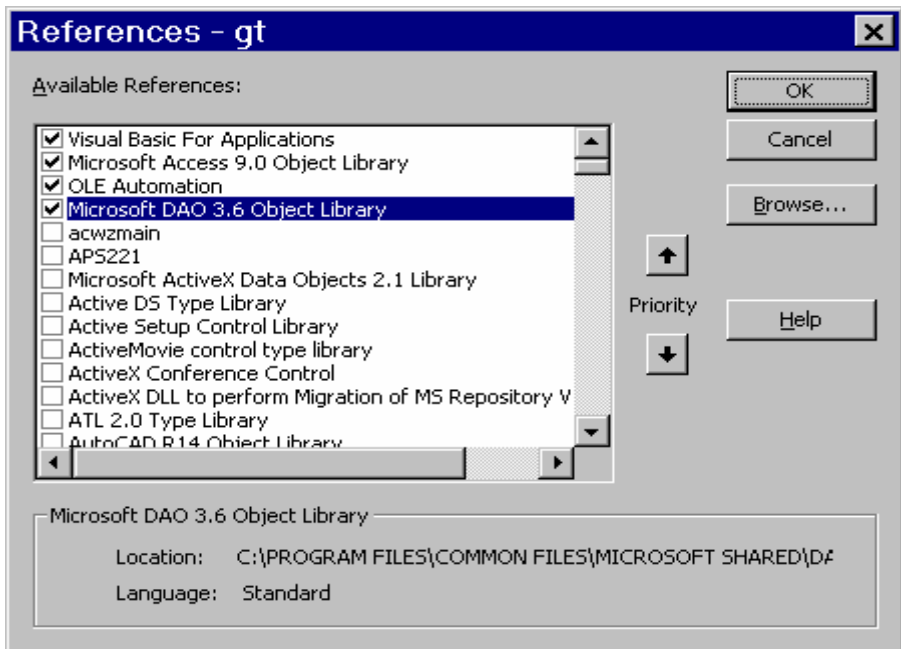
### 13.2.6 Các chức năng trong menu Debug

Chức năng	Ý nghĩa
Compile	Biên dịch các thủ tục trong module
Step Into (F8)	Thi hành thủ tục theo từng bước

Step Over (Shift+F8)	Thi hành từng bước nhưng không chuyển đến thủ tục được gọi trong dòng lệnh
Step Out (Ctrl+Shift+F8)	Thực hiện hết các lệnh còn lại trong thủ tục, đồng thời trở về chế độ thi hành từng bước tại thời điểm sau khi thủ tục hiện hành được gọi
Run To Cursor (Ctrl+F8)	Thực hiện các lệnh từ đầu thủ tục cho đến vị trí điểm nháy đang đứng thì chuyển sang thi hành từng
Add Watch	Tạo thêm biểu thức bổ sung vào cửa sổ Watch
Edit Watch	Sửa đổi biểu thức trong cửa sổ Watch
Quick Watch	Hiển thị cửa sổ chứa biểu thức ngay tại vị trí điểm nháy đang đứng (nếu có)
Toggle Breakpoint	Tạo hoặc xóa điểm dừng tại dòng lệnh có điểm nháy đang đứng
Clear All Breakpoints	Xóa tất cả các điểm dừng trong thủ tục

### 13.2.7 Thiết lập tham chiếu đến các thư viện trong Basic

Để sử dụng các đối tượng trong các thư viện liên quan với Basic, thì thiết lập tham chiếu đến các thư viện này bằng cách sau đây:



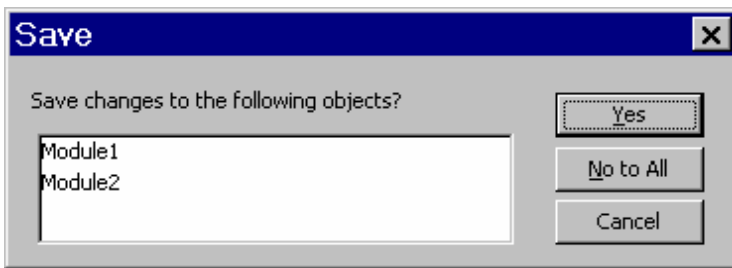
- Mở cửa sổ Module (New hoặc Design)
- Chọn menu Tools, References
- Khi xuất hiện cửa sổ References thì dùng chuột đánh dấu chọn thư viện cần tham chiếu, hoặc bỏ đánh dấu chọn đối với các thư viện không tham chiếu nữa.
- Sau đó chọn nút OK để xác nhận các tham chiếu đã chọn.

### 13.2.8 Lưu đơn thể

Để lưu các đơn thể trong tập tin cơ sở dữ liệu Access từ cửa sổ

tạo Module, thực hiện như sau:

- Chọn menu File, Save hoặc biểu tượng Save trên thanh công cụ của cửa sổ Module
- Khi xuất hiện hộp thoại dưới đây thì bấm chọn các đơn thể muốn lưu và bỏ chọn các đơn thể không cần lưu rồi nhấn nút Yes. Nếu không muốn lưu đơn thể nào cả thì nhấn nút No to All.



### 13.2.9 Sử dụng thủ tục

#### 13.2.9.1 Dạng biểu thức:

Dùng để gọi thi hành các hàm và có sử dụng giá trị trả về của hàm.

```
Tên_thủ_tục( )
Tên_thủ_tục(danh_sách_đối_mục)
```

#### 13.2.9.2 Dạng lệnh:

Dùng để gọi thi hành các thường trình con hoặc các hàm mà không sử dụng giá trị trả về của hàm.

Call Tên\_thủ\_tục

Call Tên\_thủ\_tục (danh\_sách\_đổi\_mục)

## 13.3 CÁC THÀNH PHẦN CƠ BẢN TRONG NGÔN NGỮ BASIC

### 13.3.1 Kiểu dữ liệu (Data type)

#### 13.3.1.1 Các kiểu dữ liệu cơ sở:

Tên kiểu	Kiểu Field tương Ứng	Phạm vi
Boolean	Yes/No	True, False
Byte	Byte	0 → 255
Integer	Integer	-32768 → 32767
Long	Long Integer, AutoNumber	-2147483648 → 2147483647
Single	Single	-3.402823E+38 → -1.401298E-45 và 1.401298E-45 → 3.402823E+38
Double	Double	-1.79769313486232E+308 → -4.94065645841247E-324 và 4.94065645841247E-324 → 1.79769313486232E+308
Currency	Currency	-922337203685477.5808 → 922337203685477.5807
String	Text hoặc Memo	Chiều dài từ 0 đến 65500 ký tự
Date	Date/ Time	Từ 01/01/100 → 31/12/9999
Variant	Kiểu bất kỳ (Date, String, Double)	Tương tự như các kiểu Date, Double, String ở trên

#### 13.3.1.2 Kiểu dữ liệu tự định nghĩa:

Là kiểu dữ liệu do người sử dụng tự tạo, trong đó bao gồm các kiểu dữ liệu cơ bản của Access. Cú pháp khai báo kiểu dữ liệu như sau:

```
Type Tên kiểu định nghĩa
    Tên_field_1 As
    Kiểu_dữ_liệu Tên_field_2 As
    Kiểu_dữ_liệu
    ...
End Type
```

Kiểu dữ liệu tự định nghĩa phải khai báo trong phần Declarations của module. Để sử dụng các thành phần (field) trong kiểu tự định nghĩa thì ghi như sau:

Tênbiến.Tênfield



## 13.3.2 Biến (Variable)

### 13.3.2.1 Định nghĩa:

Là đại lượng có giá trị thay đổi tùy thuộc các xử lý. Biến để lưu trữ tạm thời các kết quả khi thực hiện các phép tính trong chương trình, các kết quả được so sánh với nhau hoặc lại sử dụng trong các phép tính khác.

Biến cũng tương tự như các field của Table, phải có 1 tên gọi phân biệt, để biểu thị cho giá trị đang chứa trong biến và 1 kiểu dữ liệu để xác định loại dữ kiện nào có thể chứa trong biến.

Tên đặt cho biến theo quy định sau đây:

- Ký tự bắt đầu là chữ (A – Z)
- Các ký tự còn lại có thể là chữ (A - Z), số (0 - 9), dấu gạch nối (\_)
- Không được trùng với các từ khóa dành riêng của Basic.

---

Lưu ý: dòng lệnh **Option Explicit** trong phần Declaration Section của module để kiểm tra các biến sử dụng trong thủ tục đã có khai báo trước đó hay không. Khi biên dịch chương trình sẽ phát hiện và xuất hiện thông báo lỗi nếu biến sử dụng chưa mà được khai báo.

---

### 13.3.2.2 Phạm vi sử dụng biến:

Tùy thuộc vào từ khóa khai báo biến và vị trí khai báo mà phạm vi sử dụng biến trong các thủ tục sẽ thuộc vào 1 trong 3 trường hợp sau:

Loại	Từ khóa khai báo	Vị trí khai báo	Phạm vi sử dụng
Biến cục bộ	Dim	Trong các thủ tục	Bên trong thủ tục có chứa lệnh khai báo
Biến thuộc đơn thể	Dim	Trong Declaration Section của module	Trong bất kỳ các thủ tục thuộc module có chứa lệnh khai báo
Biến toàn cục	Global hoặc Public	Trong Declaration Section của module	Trong bất kỳ thủ tục nào thuộc bất kỳ module nào của hệ thống ứng dụng

### 13.3.2.3 *Biến tĩnh:*

Thông thường khi 1 thủ tục chấm dứt thi hành thì giá trị các biến cục bộ được khai báo trong thủ tục này mất đi và lần gọi thi hành sau đó của thủ tục các biến cục bộ sẽ được khởi tạo lại.

Nếu muốn giữ lại giá trị của biến cục bộ trong 1 thủ tục sau khi thủ tục này chấm dứt thi hành thì khai báo biến đó là biến tĩnh theo cú pháp sau:

```
Static tên_biến
```

Ví dụ: Dim I As Integer  
 Static S As Double  
 Public B As String

### 13.3.3 Hằng và hằng ký hiệu

Hằng là đại lượng có giá trị cụ thể và không đổi trong chương trình. Cách biểu diễn hằng tùy thuộc vào kiểu dữ liệu của hằng. Nếu hằng được sử dụng nhiều lần tại nhiều nơi khác nhau thì nên khai báo hằng ký hiệu để biểu thị cho giá trị hằng đó. Hằng ký hiệu sau khi khai báo có thể sử dụng trong các biểu thức nhưng không được phép thay đổi giá trị của hằng ký hiệu hoặc gán cho hằng ký hiệu 1 giá trị khác.

Phạm vi sử dụng hằng tùy thuộc vào từ khóa khai báo và giá trị khai báo như sau:

Loại	Từ khóa khai báo	Vị trí khai báo	Phạm vi sử dụng
Hằng cục bộ	Const	Trong các thủ tục	Bên trong thủ tục có chứa lệnh khai báo
Hằng thuộc đơn thể	Const	Trong Declaration Section của module	Trong bất kỳ các thủ tục thuộc module có chứa lệnh khai báo
Hằng toàn cục	Global Const hoặc Public Const	Trong Declaration Section của module	Trong bất kỳ thủ tục nào thuộc bất kỳ module nào của hệ thống ứng

Ví dụ:

```
Const CS = "Trung Tâm Máy tính và Điện tử"
Const PI = 3.14159
Const NDN = #1/1/01#
```

C  
o  
n  
s  
t

S  
A  
I

=

F  
a  
l  
s  
e

### 13.3.4 Mảng (array)

#### 13.3.4.1 Định nghĩa:

Mảng là một dãy các phần tử liên tiếp nhau có cùng một tên gọi và dùng chỉ số thứ tự để chỉ định một phần tử nào đó trong mảng.

Mỗi phần tử được sử dụng giống như một biến. Các phần tử có thể có cùng kiểu dữ liệu hoặc khác kiểu nếu mảng được khai báo là Variant.

#### 13.3.4.2 Khai báo mảng:

Loại	Từ khóa khai báo	Vị trí khai báo	Phạm vi sử dụng
Mảng cục bộ	Static	Trong các thủ tục	Bên trong thủ tục có chứa lệnh khai báo
Mảng thuộc đơn thể	Dim	Trong Declaration Section của module	Trong bất kỳ các thủ tục thuộc module có chứa lệnh khai báo
Mảng toàn cục	Global Dim hoặc Public Dim	Trong Declaration Section của module	Trong bất kỳ thủ tục nào thuộc bất kỳ module nào của hệ thống ứng

#### 13.3.4.3 Phân loại:

□ **Mảng một chiều**: Khi khai báo chỉ chứa một kích thước và khi truy xuất một phần tử sẽ gồm tên mảng kèm một chỉ số thứ tự của phần tử đó.

Ví dụ:       Public A(10) As Integer  
              Dim B(5) As Double  
              Static C(20) As Long

□ **Mảng nhiều chiều**: Khi khai báo sẽ chứa nhiều kích thước (tối đa là 60) và khi truy xuất một phần tử thì bao gồm tên mảng kèm theo các chỉ số thứ tự phân cách nhau bởi dấu phẩy đặt trong ngoặc đơn.

Ví dụ:       Static M(5,5) As Integer  
              Dim N(3, 4, 2) As Single

**Lưu ý**: Theo mặc định phần tử đầu tiên trong mảng có chỉ số là 0. Tuy nhiên có thể quy định lại là 1 bằng cách thêm dòng lệnh sau vào phần Declaration Section của đơn thể có chứa khai báo mảng:

Option Base 1

Một cách khác là ghi rõ kích thước mảng bắt đầu từ giá trị nào đến giá trị nào bằng từ khóa **To** như sau:

```
Ví dụ:      Static A(1 To 5) As Integer
           Static B(1 To 10, 1 To 4) As Double
```

#### 13.3.4.4 Mảng động (Dynamic Arrays)

Là mảng mà có thể điều khiển thay đổi kích thước của mảng ngay trong khi chương trình thi hành. Để tạo mảng động thực hiện như sau:

- Khai báo mảng bằng lệnh Dim, Global hoặc Public trong phần Declaration Section của đơn thể với kích thước để trống.
- Trong chương trình mỗi khi muốn mảng có kích thước nào đó thì dùng lệnh Redim để quy định.

Ví dụ: Khai báo mảng động trong phần Declaration Section của đơn thể:

```
Dim M( ) As Integer
```

Trong thủ tục của đơn thể, quy định kích thước mảng bằng lệnh

#### **Redim**

```
Sub Tinh_toan( )
...
Redim M(5,10)
...
End Sub
```

Lưu ý: Mỗi khi thi hành lệnh Redim, giá trị hiện tại trong tất cả các phần tử của mảng sẽ bị mất đi và Access Basic sẽ khởi tạo giá trị mặc nhiên cho các phần tử ứng với kích thước mới (nếu mảng kiểu Variant thì có phần tử có giá trị rỗng, mảng kiểu số có giá trị bằng 0 và mảng kiểu chuỗi sẽ có giá trị là ""). Nếu muốn giữ lại giá trị cũ của các phần tử trong mảng khi thực hiện lệnh Redim thì phải có thêm từ khóa **Preserve** trong lệnh.

Ví dụ: Redim Preserve M(5, 10)

## 13.4 CẤU TRÚC ĐIỀU KHIỂN

### 13.4.1 Cấu trúc chọn lựa theo điều

## kiện

Là cấu trúc khi thi hành sẽ cho kiểm định một điều kiện và tùy thuộc kết quả kiểm định sẽ thực hiện các lệnh khác nhau. Có các cấu trúc sau:



### 13.4.1.1 Cấu trúc If... Then

- Dạng 1: ghi trên một dòng:

```
If điều_kiện Then [ lệnh ] [ Else lệnh_khác ]
```

Ví dụ: Dim SL As Integer, DG As Double, TT As Double

```
SL = Val(InputBox (“Nhập số  
lượng:”)) DG = Val(InputBox (“Nhập  
đơn giá:”)) TT = SL * DG  
If TT >= 200000 Then TT = TT * 0.95
```

- Dạng 2: ghi trên nhiều dòng thành một khối lệnh:

```
If điều_kiện Then  
    lệnh  
    ...  
End If
```

Ví dụ:

```
Const PI = 3.14159  
Dim ban_kinh As Double  
ban_kinh = Val(InputBox (“Nhập bán kính:”))  
If ban_kinh > 0 then  
    Dim dientich As Double  
    Dim chuvi As Double  
    Dientich = ban_kinh*ban_kinh*PI  
    Chuvi = ban_kinh*2*PI  
End If
```

### 13.4.1.2 Cấu trúc If... Then ... Else

```
If điều_kiện_1 Then  
    [ Khối_lệnh_1 ]  
[ ElseIf điều_kiện_2 Then  
    [ Khối_lệnh_2 ] ]  
...  
[ Else  
    [ Khối_lệnh_n ] ]  
End If
```

Lưu ý: số dòng lệnh **ElseIf** dùng trong cấu trúc là tùy ý, còn

dòng lệnh **Else** thì chỉ có một và có thể dùng hoặc không.

Ví dụ: Const LCB = 2000000  
 Dim luong As Double  
 Dim thue As Double  
 Luong = Val(InputBox("Mức lương tháng:")) If luong <= LCB then  
     thue = 0  
 ElseIf luong <= 3000000 then  
     thue = (luong - LCB)\*10/100  
 ElseIf luong <= 5000000 then  
     thue = (luong - LCB)\*20/100  
 ElseIf luong <= 10000000 then  
     thue = (luong - LCB)\*40/100  
 Else  
     thue = (luong - LCB)\*60/100  
 End If

MsgBox "Thực lãnh:" & (luong - thue)

#### 13.4.1.3 Cấu trúc Select Case:

```

Select Case biểu_thức_kiểm_định
[ Case
    Liệt_kê_các_biểu_thức_1
    : [ Khối_lệnh_1 ]
  [ Case
    Liệt_kê_các_biểu_thức_2
    : [ Khối_lệnh_2 ]
    ...
  [ Case Else
    [ Khối_lệnh_n ]
End Select
  
```

Lưu ý: biểu\_thức\_kiểm\_định là biểu thức kiểu số hoặc biểu thức kiểu chuỗi. Khi dùng liệt\_kê\_các\_biểu\_thức phải theo quy định sau:

- Các biểu thức riêng biệt sẽ liệt kê cách nhau bởi dấu phẩy.
- Các biểu thức gồm các giá trị liên tục:  
     giá trị\_bắt\_đầu **To** giá trị\_kết\_thúc
- Các biểu thức so sánh: **Is** toán\_tử\_so\_sánh biểu\_thức

Ví dụ: Const LCB = 2000000  
 Dim luong As Double  
 Dim thue As Double  
 Luong = Val(InputBox("Mức lương

tháng.” )) Select Case luong  
Case Is <= LCB  
thue = 0

```

Case LCB To 3000000
    thue = (luong – LCB)*10/100
Case 3000000 To 5000000
    thue = (luong – LCB)*20/100
Case 5000000 To 10000000
    thue = (luong – LCB)*40/100
Case Else
    thue = (luong – LCB)*60/100
End Select
MsgBox “Thực lãnh:” & (luong – thue)

```

### 13.4.2 Cấu trúc lặp

Là cấu trúc cho phép điều khiển thi hành lặp đi lặp lại nhiều lần một hoặc nhiều câu lệnh. Số lần lặp lại tùy thuộc vào kết quả kiểm định 1 điều kiện trong cấu trúc lặp. Các cấu trúc lặp gồm có:

#### 13.4.2.1 *Cấu trúc Do ... Loop:*

□ Dạng 1: kiểm định điều kiện trước :

<pre> <b>Do Until</b>   <b>While</b> <i>Điều_kiện_kiểm_định</i>     Lệnh     .....     [Exit Do]     .....     Lệnh <b>Loop</b> </pre>
--

- Khi thi hành vòng lặp, điều kiện lặp sẽ được kiểm định trước tiên, và tùy thuộc giá trị của điều kiện mà các lệnh trong vòng lặp có được thực hiện hay không.
- Với Do Until, nếu điều kiện có giá trị là True (hoặc khác 0) thì sẽ chấm dứt vòng lặp và thi hành tiếp các lệnh sau Loop, ngược lại nếu điều kiện có giá trị False (hoặc bằng 0) thì các lệnh trong vòng lặp được thi hành sau đó quay lên kiểm định lại điều kiện lần nữa và trình tự lặp lại như trên.
- Với Do While thì kết quả ngược lại, tức là nếu điều kiện là True (hoặc khác 0) thì các lệnh trong vòng lặp được thi hành sau đó quay lên kiểm định lại điều kiện lần nữa, còn nếu điều kiện là False (hoặc bằng 0) thì kết thúc vòng lặp.
- Trong khi thực hiện các lệnh của vòng lặp, nếu gặp lệnh Exit Do thì chương trình sẽ thoát khỏi vòng lặp và thực hiện tiếp các lệnh nằm sau Loop.

```

Ví dụ: Dim x As Integer
        Dim y As Double
        x = Val (InputBox("Nhập 1
        số:")) Do While x > 0
            y = y + x
            x = x - 2
        If x < 1 Then Exit Do
    Loop
    MsgBox "Tổng =" & y

```

□ Dạng 2: kiểm định điều kiện sau:

```

Do
    Lệnh
    .....
    [Exit Do]
    .....
    Lệnh
Loop Until | While Điều_kiện_kiểm_định

```

hiện Khi thi hành vòng lặp, các lệnh trong vòng lặp sẽ được thực trước tiên rồi sau đó mới kiểm định điều kiện. Như vậy bất luận kết quả điều kiện như thế nào thì các lệnh của vòng lặp đã được làm qua.

- Sau đó vòng lặp sẽ tiếp tục thực hiện lại các lệnh hay chấm dứt là tùy thuộc vào kết quả kiểm định điều kiện, hình thức kiểm định giống như vòng lặp trong dạng 1.

```

Ví dụ: Dim a As Long, b As Long
        Do
            a = Val(InputBox("Nhập số (Ngưng thì nhập số 0) :")
            b = b + a
        Loop Until a = 0
        MsgBox "Tổng các số :" & b

```

#### 13.4.2.2 Cấu trúc While ... Wend:

```

While Điều_kiện_kiểm_định
    Lệnh
    .....
Wend

```

Vòng lặp While sẽ còn tiếp tục thi hành các lệnh trong vòng lặp khi nào điều kiện kiểm định vẫn còn là True (hay khác 0).

Ví dụ:

```

Dim n As Integer
Dim S As Double
n = Val (InputBox (“Nhập 1
số :”)) While n > 0
    S = S + 1/n
    n = n - 1
Wend
MsgBox “Tổng = “ & S

```

#### 13.4.2.3 Cấu trúc For ... Next

```

For biến_đếm = giá_trị_1 To giá_trị_2 [Step
    giá_trị_3] Lệnh
    .....
    [Exit For]
    .....
    Lệnh
Next [biến_đếm]

```

- Các *giá\_trị\_1*, *giá\_trị\_2*, và *giá\_trị\_3* trong vòng lặp đều là kiểu số.
- Đầu tiên *biến\_đếm* được gán *giá\_trị\_1* sau đó so sánh với *giá\_trị\_2*, biểu thức so sánh sẽ tùy thuộc vào *giá\_trị\_3*. Khi *giá\_trị\_3* là số dương thì biểu thức so sánh là *biến\_đếm* <= *giá\_trị\_2*, nếu giá trị biểu thức là True thì các lệnh trong vòng lặp sẽ được thi hành, sau đó *biến\_đếm* được cộng thêm *giá\_trị\_3* rồi quay lên kiểm định lại biểu thức so sánh lần nữa. Trình tự cứ lặp lại như thế cho đến khi giá trị của *biến\_đếm* lớn hơn *giá\_trị\_2* thì vòng lặp chấm dứt.
- Nếu *giá\_trị\_3* là số âm thì biểu thức so sánh sẽ là *biến\_đếm* >= *giá\_trị\_2*, và vòng lặp sẽ còn thi hành các lệnh khi nào giá trị *biến\_đếm* vẫn lớn hơn hay bằng *giá\_trị\_2*.
- Nếu không ghi **Step** *giá\_trị\_3* trong vòng lặp thì mặc định *giá\_trị\_3* bằng 1.
- Trong khi thực hiện các lệnh của vòng lặp, nếu gặp lệnh **Exit For** thì chương trình sẽ thoát khỏi vòng lặp và thực hiện tiếp các lệnh nằm sau **Next**.

Ví dụ:

```

Dim N As Long
Dim I AS Long

```



```

Dim GT AS Double
GT = 1
N = Val (InputBox (“Nhập 1 số để tính giai
thừa:”)) If N>0 Then
    For I = 1 To N
        GT = GT * I
    Next
    MsgBox “Giai thừa “ & N & “! =” & GT
Else
    MsgBox “Không tính giai thừa của số âm”
End If

```

### 13.4.3 Các lệnh khác

#### 13.4.3.1 Lệnh thoát khỏi thủ tục:

- Thoát khỏi hàm: **Exit Function**
- Thoát khỏi thường trình con: **Exit Sub**

#### 13.4.3.2 Lệnh gọi thi hành các hành động hoặc các Macros:

DoCmd.Tên_hành_động [ Danh_sách_đối_mục ]
---

- Tên\_hành\_động là tên của các lệnh Macros (chương 12)
- Danh\_sách\_đối\_mục là các giá trị cung cấp cho lệnh Macro tương ứng để thi hành, các đối mục cách nhau bởi dấu phẩy.

Ví dụ:

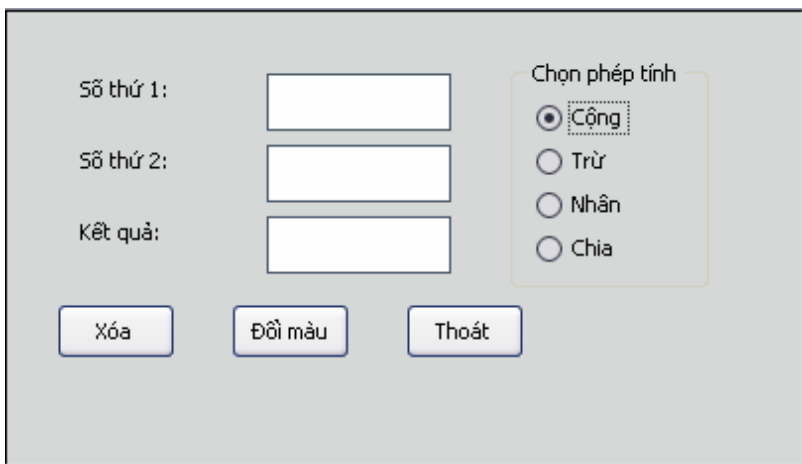
```

DoCmd.OpenForm “NHANVIEN”, acNormal, , , acFormAdd
DoCmd.OpenReport “LYLICH”, acViewPreview
DoCmd.Maximize
DoCmd.Close
DoCmd.GotoControl “MANV”

```

## BÀI TẬP:

1. Tạo một module dùng chung, trong đó viết các thủ tục sau:
  - Hàm nhận 2 đối số, trả về giá trị lớn hơn trong hai đối số.
  - Hàm nhận 2 đối số, trả về giá trị nhỏ hơn trong hai đối số.
  - Hàm nhận 3 đối số, trả về giá trị ở giữa trong hai đối số.
2. Tạo một module dùng chung, trong đó viết các thủ tục sau:
  - Hàm nhận 1 đối số kiểu ngày, trả về giá trị chuỗi là tên của thứ trong tuần ứng với đối số.
  - Hàm nhận 1 đối số kiểu chuỗi, trả về chuỗi mà ký tự đầu trong mỗi từ sẽ viết hoa còn lại các ký tự khác viết thường.
  - Hàm nhận 1 đối số là năm dương lịch, trả về chuỗi là tên năm âm lịch ứng với năm dương lịch.
  - Hàm nhận 1 đối số kiểu ngày, trả về giá trị số cho biết số ngày trong tháng ứng với đối số đã nhận.
3. Tạo Form sau đây và trong đó viết các thủ tục xử lý vào các nút lệnh:



The image shows a graphical user interface for a calculator. It consists of three input fields on the left, each with a label: 'Số thứ 1:', 'Số thứ 2:', and 'Kết quả:'. To the right of these fields is a section titled 'Chọn phép tính' (Choose operation) containing four radio buttons: 'Cộng' (Add), 'Trừ' (Subtract), 'Nhân' (Multiply), and 'Chia' (Divide). The 'Cộng' option is currently selected. Below the input fields and the operation selection are three buttons: 'Xóa' (Clear), 'Đổi màu' (Change color), and 'Thoát' (Exit).

- Sau khi nhập 2 số thì bấm chọn phép toán nào sẽ hiện ra kết quả tương ứng.
- Nếu bấm nút Xóa thì các ô text box sẽ được xóa trắng
- Nếu bấm nút Đổi màu thì thay đổi màu nền của Form.

- Nếu bấm nút Thoát thì đóng Form lại.

## CHƯƠNG 14

# ĐỐI TƯỢNG TRUY XUẤT DỮ LIỆU

- Giới thiệu
- Sử dụng Cấu trúc Điều khiển với đối tượng
- Xử lý dữ liệu với đối tượng DAO

### 14.1 GIỚI THIỆU

#### 14.1.1 Đối tượng truy xuất dữ liệu (Data access object - DAO)

Đây là lớp đối tượng thuộc Microsoft Jet Database Engine, một bộ phận của Microsoft Access, được tham chiếu qua thư viện Microsoft DAO 3.6 Object Library (tập tin Dao360.dll), cung cấp các công cụ để sử dụng lập trình, tạo và xử lý các thành phần trong những cơ sở dữ liệu ứng dụng bằng Microsoft Access.

Trong lớp đối tượng truy xuất dữ liệu này có một số đối tượng cho phép khai báo biến để biểu thị cho đối tượng đó trong các thủ tục. Dưới đây là bảng liệt kê các đối tượng truy xuất dữ liệu:

Đối tượng	Ý nghĩa	Biến biểu thị
Container	Đối tượng chứa thông tin về những đối tượng khác đã định nghĩa trong MS Access hoặc Jet Database Engine	Có
Containers	Tập hợp những Containers	Không
Database	Cơ sở dữ liệu đang mở	Có
Databases	Tập hợp những cơ sở dữ liệu đang mở	Không
DBEngine	Microsoft Jet Database Engine	Không
Document	Đối tượng chứa thông tin về những đối tượng khác trong cơ sở dữ liệu	Có
Documents	Tập hợp những Documents	Không
Field	Vùng thuộc bảng dữ liệu, bảng truy vấn,	Có
Fields	Tập hợp những vùng thuộc bảng dữ liệu, bảng truy vấn, Recordset, chỉ mục	Không

Group	Nhóm người sử dụng Jet Database Engine có đăng ký thuộc nhóm làm việc hiện hành	Có
-------	---	----

Groups	Tập hợp những Groups	Không
Index	Bảng chỉ mục	Có
Indexes	Tập hợp những bảng chỉ mục	Không
Parameter	Tham số của truy vấn	Có
Parameters	Tập hợp những tham số của truy vấn	Không
Property	Tính chất của một đối tượng	Có
Properties	Tập hợp những tính chất của một đối tượng	Không
QueryDef	Qui định truy vấn đã lưu trữ trong một cơ sở dữ liệu	Có
QueryDefs	Tập hợp những qui định truy vấn đã lưu trữ trong một cơ sở dữ liệu	Không
Recordset	Tập hợp mẫu tin được định nghĩa bởi một bảng dữ liệu hoặc bảng truy vấn	Có
Recordsets	Tập hợp những Recordset	Không
Relation	Quan hệ giữa hai field của bảng dữ liệu hoặc bảng truy vấn	Có
Relations	Tập hợp các Relations	Không
TableDef	Bảng dữ liệu đã lưu trữ trong một cơ sở dữ liệu	Có
TableDefs	Tập hợp những bảng dữ liệu đã lưu trữ trong một cơ sở dữ liệu	Không
User	Người sử dụng Jet Database Engine có đăng ký thuộc nhóm làm việc hiện hành	Có
Users	Tập hợp những người sử dụng Jet Database Engine có đăng ký thuộc nhóm làm việc hiện hành	Không
WorkSpace	Phiên sử dụng Jet Database Engine hiện hành	Có
WorkSpaces	Tập hợp những phiên sử dụng Jet Database Engine hiện hành	Không

#### 14.1.2 Đối tượng của Microsoft Access

Gồm các đối tượng được định nghĩa sẵn bởi MS Access, trong đó có một số cho phép khai báo biến biểu thị.

Đối tượng	Ý nghĩa	Biến biểu thị
Application	Toàn bộ hệ thống ứng dụng MS Access hiện hành	Không
Control	Ô điều khiển thuộc mẫu biểu / báo biểu	Có

Form	Mẫu biểu đang mở kể cả mẫu biểu con	Có
Forms	Tập hợp các mẫu biểu đang mở	Không
Module	Đơn thể của mẫu biểu hay báo biểu	Không
Report	Báo biểu đang mở kể cả báo biểu con	Có
Reports	Tập hợp các báo biểu đang mở	Không
Screen	Màn hình	Không
Section	Một phần của mẫu biểu hay báo biểu	Không

### 14.1.3 Khai báo và gán biến biểu thị đối tượng

#### 14.1.3.1 Khai báo biến biểu thị đối tượng:

<b>Dim</b> Tên_biến As Kiểu_đối_tượng
<b>Dim</b> Tên_biến As DAO.Kiểu_đối_tượng

Ví dụ: Dim DB As DAO.Database  
Dim MyForm As Form

#### 14.1.3.2 Gán giá trị cho biến biểu thị đối tượng:

Cho phép liên kết một biến đối tượng với một đối tượng đã có trong cơ sở dữ liệu khi viết các thủ tục.

Set Tên_biến = biểu_thức
-----------------------------

Ví dụ:

```
Dim F_NV As Form
Dim C_hoten As Control
Set F_NV = Forms!Nhan_vien
Set C_hoten = F_NV!Hoten
Dim Db As DAO.Database
Dim Rs As DAO.Recordset
Set Db = CurrentDb
Set Rs = Db.OpenRecordset("NHAN_VIEN", dbOpenTable)
```

### 14.1.4 Sử dụng đối tượng và biến biểu thị đối tượng

#### 14.1.4.1 Tính chất (Properties) của đối tượng:

Tất cả các đối tượng (ngoại trừ Database) đều có các mục tính chất do MS Access định nghĩa sẵn, tùy thuộc mỗi loại đối tượng mà các tính chất sẽ thay đổi khác nhau. Đa số các tính chất có thể gán giá trị thay đổi tương ứng với kiểu dữ liệu của tính chất đó trong chương trình Basic trừ một số tính chất chỉ đọc thì

không cho gán. Khi tham chiếu tính chất đối tượng sẽ theo cú pháp sau:

Đối\_tượng.Tính\_chất



```

Ví dụ: Dim F_NV As Form
        Dim C_hoten As Control, TC
        Set F_NV = Forms!Nhan_vien
        Set C_hoten = F_NV!Hoten
        TC = F_NV.Visible
        C_hoten.Visible = False

```

#### 14.1.4.2 Thành phần (Members) của đối tượng:

Một số đối tượng của MS Access có thể chứa những đối tượng khác do người sử dụng định nghĩa (Ví dụ như mẫu biểu hay báo biểu chứa các ô điều khiển, bảng dữ liệu hay bảng truy vấn thì chứa các field). Những đối tượng được chứa trong một đối tượng khác gọi là thành phần của đối tượng chứa và được tham chiếu theo cú pháp:

```
Đối_tượng!Thành_phần
```

Ví dụ:

```

Dim Db As DAO.Database
Dim Rs As DAO.Recordset
Set Db = CurrentDb
Set Rs = Db.OpenRecordset("NHAN_VIEN", dbOpenTable)
Dim C_ht, C_ns
C_ht = Rs!Hoten           ' Hoten và Ngaysinh là hai field
C_ns = Rs!Ngaysinh       ' trong bảng NHAN_VIEN

```

Lưu ý: Ta có thể tham chiếu đến đối tượng mà tên của đối tượng biểu thị bởi biến, hoặc tham chiếu đến các ô điều khiển trong mẫu biểu, báo biểu bằng số thứ tự của ô điều khiển.

Ví dụ:

```

Dim F_Name As String
Dim Frm As Form
Dim I As Integer
F_Name = InputBox("Tên mẫu
biểu:") DoCmd.OpenForm F_Name
Set Frm = Forms(F_Name) ' dùng biến chứa tên
form
For I = 0 To Frm.Count - 1
    Frm(I).ForeColor = 255 ' dùng chỉ số thứ tự để tham chiếu
Next

```

#### 14.1.4.3 Phương thức (Methods) của đối tượng:

Phương thức hay còn gọi là hành vi của đối tượng là một lệnh xử lý hoặc một hàm đặc biệt tác động lên một đối tượng nào đó.

Tùy thuộc mỗi loại đối tượng mà MS Access định sẵn các phương thức khác nhau cho từng đối tượng. Để thực hiện phương thức của đối tượng thì sử dụng cú pháp sau:

```
Đối_tượng.Phương_thức [(danh_sách_đối_mục)]
```

Danh sách đối mục có thể có hoặc không tùy thuộc từng phương thức của đối tượng và các đối mục phân cách nhau bằng dấu phẩy.

Ví dụ:

```
Dim Db As DAO.Database
Dim Rs As DAO.Recordset
Dim TimNV As String
Set Db = CurrentDb
' Dùng phương thức OpenRecordset của đối tượng
Database Set Rs = Db.OpenRecordset("NHAN_VIEN",
dbOpenTable) TimNV = InputBox("Tìm nhân viên tên:")
Rs.MoveFirst
Do Until Rs.EOF
    If Rs!Hoten = TimNV Then Exit Do
    Rs.MoveNext
Loop
If Rs.EOF Then
    MsgBox "Không có nhân viên này"
Else
    MsgBox "HỌ tên:" & Rs!Hoten & _
        " – Ngày sinh:" & Rs!Ngaysinh & _
        " – Địa chỉ:" & Rs!Diachi & _
        " – Mức lương:" & Rs!
        Luong
End If
' Dùng phương thức Close của đối tượng Recordset và Database
Rs.Close
Db.Close
```

## 14.2 SỬ DỤNG CẤU TRÚC ĐIỀU KHIỂN VỚI BIẾN ĐỐI TƯỢNG

Đối với một số đối tượng của MS Access và đối tượng truy xuất dữ liệu có thể sử dụng những cấu trúc điều khiển đã trình bày trong chương 13. Ngoài ra còn có những cấu trúc dùng riêng cho các đối tượng này.

### 14.2.1 Cấu trúc If TypeOf ... Is ... Then

```
If TypeOf biến_đối_tượng Is hằng_nội_tại Then
    Lệnh_1
[ ElseIf TypeOf biến_đối_tượng Is hằng_nội_tại Then
    Lệnh_2 ]
.....
[ Else
    Lệnh_n ]
End If
```

Cấu trúc này kiểm định đối tượng mà biến biểu thị thuộc loại nào với hằng nội tại là các hằng ký hiệu đã định nghĩa sẵn trong Access để qui định về loại đối tượng. Hằng nội tại biểu thị cho các đối tượng thuộc loại ô điều khiển trong mẫu biểu hoặc báo biểu gồm có: CheckBox, CommandButton, ComboBox, ListBox, OptionGroup, OptionButton, ToggleButton, TextBox, Label, Image, ObjectFrame ...

Ví dụ:

```
Dim F_Name As String
Dim Frm As Form
Dim I As Integer
F_Name = InputBox("Tên mẫu
biểu:") DoCmd.OpenForm F_Name
Set Frm = Forms(F_Name)
For I = 0 To Frm.Count - 1
    If TypeOf Frm(I) Is Label Then
        Frm(I).ForeColor = 255
    ElseIf TypeOf Frm(I) Is TextBox Then
        Frm(I).ForeColor = 0
    ElseIf TypeOf Frm(I) Is CommandButton Then
        Frm(I).ForeColor = 128
    End If
Next
```

### 14.2.2 Cấu trúc With ... End With

```
With đối_tượng
    Lệnh
    .....
End With
```

Cấu trúc With cho phép thực hiện các lệnh đối với một biến đối tượng nào đó hoặc biến có kiểu do người sử dụng tự định nghĩa. Các lệnh trong cấu trúc With sẽ tham chiếu trực tiếp đến các tính chất của đối tượng, vì vậy cấu trúc này thường được dùng để thay đổi giá trị của các tính chất trong đối tượng.

Ví dụ:

```
Dim F_NV As Form, I As Integer
Set F_NV = Forms!Nhan_vien
For I = 0 To Frm.Count - 1
    If TypeOf Frm(I) Is TextBox Then
        With Frm(I)
            .BackColor = 1
            .ForeColor = 128
            .BorderStyle = 1
            .Height = 200
        End With
    End If
Next
```

#### 14.2.3 Cấu trúc For Each ... Next

```
For Each phần_tử In nhóm
    Lệnh
    .....
    [ Exit For ]
    Lệnh
Next [ phần_tử ]
```

Số lần vòng lặp For thực hiện các lệnh tùy thuộc vào số phần tử trong nhóm. Phần tử là biến đối tượng chứa trong một đối tượng khác, còn nhóm có thể là mảng hoặc biến đối tượng bao gồm nhiều đối tượng thành phần.

Ví dụ:

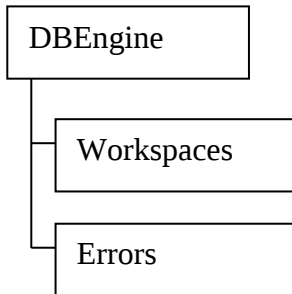
```
Dim frm As Form, Ctl As Control
Set frm = Forms!Nhan_vien
For Each Ctl In frm.Controls
    With Ctl
        .FontName = "VNI-Times"
        .FontSize = 12
        .ForeColor = 128
        .FontBold = True
    End With
```

Next

## 14.3 XỬ LÝ DỮ LIỆU VỚI CÁC ĐỐI TƯỢNG DAO

### 14.3.1 Đối tượng DBEngine

Là đối tượng có cấp cao nhất trong mô hình đối tượng DAO.



Để tham chiếu các tập hợp thuộc DBEngine hoặc các tính chất, phương thức thì dùng:

```
DBEngine.tập_hợp | tính_chất | phương_thức
```

#### 14.3.1.1 Tập hợp:

- **Workspaces:** để tham chiếu đến vùng làm việc được chỉ định bằng chỉ số thứ tự.
- **Errors:** dùng để xem xét chi tiết các lỗi về truy xuất dữ liệu.

#### 14.3.1.2 Tính chất:

- **DefaultType:** trả về giá trị cho biết kiểu mặc định của vùng làm việc hoặc qui định kiểu cho vùng làm việc sẽ tạo sau đó.
- **DefaultUser:** qui định tên người sử dụng để tạo vùng làm việc mặc định.
- **DefaultPassword:** qui định mật mã sử dụng để tạo vùng làm việc mặc định.

#### 14.3.1.3 Phương thức:

- **CreateWorkspace:** tạo một đối tượng vùng làm việc mới.

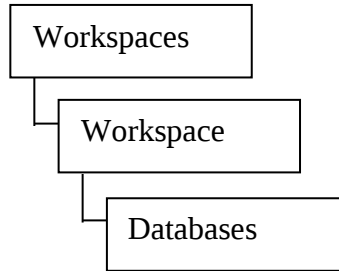
```
CreateWorkspace(tên_workspace, user, password, loại)
```

Loại	Ý nghĩa
dbUseJet	Tạo 1 đối tượng Workspace với Microsoft Jet database engine
DbUseODBC	Tạo 1 đối tượng Workspace nối kết với ODBC data source

Ví dụ:

```
Dim wrk As Workspace
Set wrk = DBEngine.WorkSpaces(0) ' vùng làm việc hiện hành
Dim wrkJet As Workspace
DBEngine.DefaultType = dbUseJet
Set wrkJet = CreateWorkspace("JetWorkspace", "admin", "")
```

### 14.3.2 Đối tượng Workspace



#### 14.3.2.1 Tập hợp:

- **Databases:** để tham chiếu đến cơ sở dữ liệu được chỉ định bằng chỉ số thứ tự.
- **Users:** tập hợp các người sử dụng trong vùng làm việc.

#### 14.3.2.2 Tính chất:

- **Type:** loại của vùng làm việc.
- **Name:** tên vùng làm việc .

#### 14.3.2.3 Phương thức:

- **CreateDatabase:** tạo đối tượng cơ sở dữ liệu, lưu trên đĩa và trả về đối tượng cơ sở dữ liệu đang mở trong vùng làm việc (chỉ dùng với vùng làm việc Microsoft Jet).

```
CreateDatabase(Tên_database, connect, option)
```

- **OpenDatabase:** mở một cơ sở dữ liệu chỉ định trong một đối tượng vùng làm việc.

```
OpenDatabase(Tên_dbs, option, read_only, connect)
```

- **Close:** đóng vùng làm việc.

Ví dụ:

```
Dim wr As Workspace
```



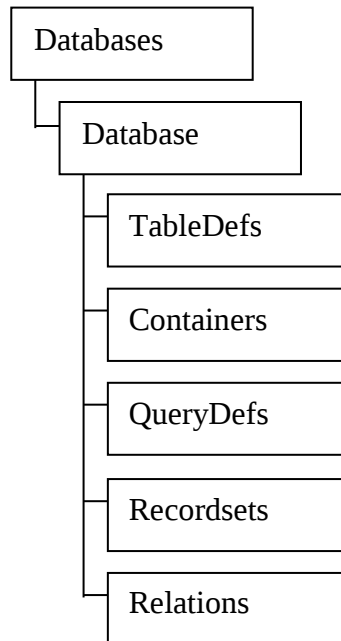
```
Dim db As DAO.Database
Set wr = DBEngine.WorkSpaces(0) ' vùng làm việc hiện hành
Set db=wr.CreateDatabase("DL.mdb", dbLangGeneral, dbEncrypt)
db.Close
```

```

Dim wrkJet As Workspace
Dim dbs As DAO.Database
DBEngine.DefaultType = dbUseJet
Set wrkJet = CreateWorkspace("JetWorkspace", "admin", "")
Set dbs = wrkJet.OpenDatabase("Northwind.mdb", True)
dbs.Close
wrkJet.Close

```

### 14.3.3 Đối tượng Database



#### 14.3.3.1 *Tính chất:*

- Name: tên cơ sở dữ liệu
- Querydefs: tập hợp các qui định truy vấn
- Tabledefs: tập hợp các bảng dữ liệu
- Relations: tập hợp các liên kết.

Tương ứng với mỗi loại tập hợp trên đều có thể tham chiếu đến từng phần tử của tập hợp bằng cách thêm chỉ số thứ tự vào. Ngoài ra các tập hợp còn có các tính chất và phương thức sau đây:

- y Count: tính chất trả về số đối tượng phần tử có trong tập hợp.
- y Append: phương thức thêm 1 đối tượng DAO vào tập hợp.
- y Delete: phương thức xóa đối tượng đã lưu trữ trong cơ sở dữ liệu

liệu từ tập hợp.

#### 14.3.3.2 Phương thức:

- CreateQueryDef: tạo 1 đối tượng qui định truy vấn mới.

<b>CreateQueryDef</b> ( <i>tên_querydef</i> , <i>lệnh_sql</i> )
---

Ví dụ:

```
Dim db As DAO.Database
Dim qd As QueryDef
Set db=DBEngine.WorkSpaces(0).OpenDatabase("QLNS.mdb")
With db
    Set qd= .CreateQueryDef("DSNV", "SELECT * FROM NV")
    DoCmd.OpenReport "DSNV"
    .QueryDefs.Delete qd.Name
    .Close
End With
```

- CreateTableDef: tạo đối tượng TableDef mới (dùng với Microsoft Jet Workspace).

**CreateTableDef** (*name, attributes, source, connect*)

Ví dụ:

```
Dim db As DAO.Database
Dim tdfNew As TableDef
Set db=DBEngine.WorkSpaces(0).OpenDatabase("QLNS.mdb")
Set tdfNew = db.CreateTableDef("Lylich")
With tdfNew
    .Fields.Append .CreateField("MaNV", dbText,5)
    .Fields.Append .CreateField("Hoten", dbText,30)
    .Fields.Append .CreateField("Phai", dbBoolean)
    .Fields.Append .CreateField("Ngaysinh", dbDate)
    .Fields.Append .CreateField("Diachi", dbText,40)
    .Fields.Append .CreateField("Luong", dbLong)
    .Fields.Append .CreateField("Ghichu", dbMemo)
End With
db.TableDefs.Append tdfNew
db.Close
```

- CreateRelation: tạo 1 đối tượng Relation mới (dùng với Microsoft Jet Workspace).

**CreateRelation** (*name, table, foreigntable, attributes*)

Ví dụ:

```
Dim dbs As DAO.Database
Dim taNV As TableDef
Dim taNew As TableDef
Dim idx As Index, relNew As Relation
Set dbs = DBEngine.WorkSpaces(0).OpenDatabase("QLNS.mdb")
```

With dbs

```
Set taNV = .TableDefs!Nhanvien
taNV.Fields.Append taNV.CreateField("MAPB", dbText, 2)
Set taNew = .CreateTableDef("PhongBan")
With taNew
    .Fields.Append .CreateField("MAPB", dbText, 2)
    .Fields.Append .CreateField("TenPB", dbText, 20)
    Set idx = .CreateIndex("MAPBIndex")
    idx.Fields.Append idx.CreateField("MAPB")
    idx.Unique = True
    .Indexes.Append idx
End With
.TableDefs.Append taNew
Set relNew = .CreateRelation("NV_PB", taNew.Name, _
    taNV.Name, dbRelationUpdateCascade)
relNew.Fields.Append relNew.CreateField("MAPB")
relNew.Fields!MAPB.ForeignName = "MAPB"
.Relations.Append relNew
‘ Xóa liên kết, field và Table
.Relations.Delete relNew.Name
.TableDefs.Delete taNew.Name
taNV.Fields.Delete "MAPB"
.Close
```

End With

- OpenRecordset: tạo 1 đối tượng recordset mới và thêm vào tập hợp recordsets.

**OpenRecordset** (tên\_source, type, options, lockedits)

<i>Type</i>	Qui định
dbOpenTable	Table (giá trị ngầm định cho nguồn TableDef cục bộ)
dbOpenDynaset	Dynaset (giá trị ngầm định cho Database, QueryDef, Recordset hoặc một TableDef đính kèm)
dbOpenSnapshot	Snapshot (không cập nhật được)
dbOpenForwardOnly	Tương tự như Snapshot nhưng bạn chỉ có thể dời tới qua Recordset (là ngầm định cho các đối tượng ODBDirect Recordset)
dbOpenDynamic	Con trỏ động (chỉ các đối tượng ODBDirect Recordset)

<b>Options</b>	<b>Qui định</b>
dbDenyWrite	Ngăn không cho người khác thay đổi các mẫu tin trong (các) bảng cơ sở khi Recordset đang mở
dbDenyRead	Ngăn không cho người khác đọc các mẫu tin trong bảng cơ sở khi Recordset đang mở. Tùy chọn này chỉ áp dụng cho đối tượng Recordset kiểu Table, nó phải được cho các mục tiêu điều hành chỉ
dbReadOnly	Không cho phép cập nhật các khoản tin trong bảng. Việc truy cập chỉ ở mức chỉ đọc sẽ gia tăng tốc độ của một số tác
dbAppendOnly	Chỉ cho phép thêm các mẫu tin mới. (Chỉ áp dụng cho các đối tượng Recordset kiểu
dbInconsistent	Bạn có thể cập nhật phía một trong mỗi
dbConsistent	quan hệ một-đối-nhiều. (Chỉ áp dụng không có thể cập nhật phía một trong mỗi
dbForwardOnly	quan hệ một-đối-nhiều, là giá trị ngầm định. (Chỉ áp dụng cho các đối tượng
dbSQLPassThrough	Tạo một đối tượng Recordset có kiểu
dbSeeChanges	Chỉ định một bộ hỏi thông tuyến SQL đối lại một cơ sở dữ liệu ODBC dùng Jet,
dbRunAsync	Phát sinh một lỗi thời gian thực hiện, khi người dùng cố thay đổi dữ liệu mà người dùng khác đang hiệu đính. (Chỉ Recordset
dbExecDirect	Thi hành bộ hỏi dị bộ, cho phép hủy bỏ bộ
	Chạy một bộ hỏi không có các tham số trực tiếp, thay vì tạo một điều lệnh

<b>Lockedit</b>	Qui định
dbReadOnly	Không cho phép cập nhật Recordset
dbPessimistic	Dùng tính năng khóa bi quan (chốt khóa trang hiệu chỉnh trong khi hiệu chỉnh)
dbOptimistic	Dùng tính năng khóa lạc quan (chỉ chốt khóa trang hiệu chỉnh khi áp dụng các đợt hiệu chỉnh)
dbOptimisticValue	Dùng tính năng khóa lạc quan dựa trên các giá trị hàng (chỉ ODBCDirect)
dbOptimisticBatch	Dùng tính năng khóa lạc quan cho các đợt cập nhật lô (chỉ ODBCDirect)

Ví dụ:

‘ Tạo biến Recordset loại Table ứng với bảng dữ liệu

Dim Db As DAO.Database

Dim Rst As DAO.Recordset

Set Db = DBEngine.Workspaces(0).Databases(0)

Set Rst = Db.OpenRecordset(“Nhanvien”, dbOpenTable)

‘ Tạo biến Recordset loại Dynaset ứng với bảng truy vấn

Dim Db As DAO.Database

Dim Rsd As DAO.Recordset

Dim R\_sql As DAO.Recordset

Dim SQLs As String

Set Db = DBEngine.Workspaces(0).Databases(0)

Set Rsd = Db.OpenRecordset(“Chamcong”, dbOpenDynaset)

SQLs = “SELECT \* FROM Nhanvien WHERE MaPB = ‘KT’ “

Set R\_sql = Db. OpenRecordset(SQL\_text)

‘ Tạo biến Recordset trên cơ sở dữ liệu của mẫu biểu

Dim Rsf As DAO.Recordset

Set Rsf = Forms!Nhan\_vien.RecordsetClone

□ Close: đóng các biến đối tượng

Database. Ví dụ:

Dim wk As Workspace

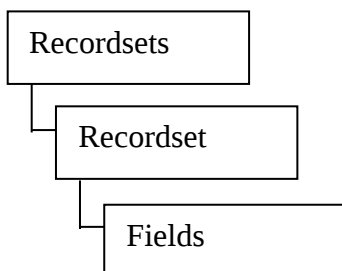
Dim db As DAO.Database

Set wk = DBEngine.WorkSpaces(0)

Set db = wk.CreateDatabase(“DB.mdb”,dbLangGeneral,dbEncrypt)

dbNew.Close

#### 14.3.4 Đối tượng Recordset



##### 14.3.4.1 *Tính chất:*

- Name: tên đối tượng Recordset
- Index: trả về tên của chỉ mục hiện hành hoặc qui định một chỉ mục có sẵn để thay đổi thứ tự các mẫu tin trong recordset.
- Filter: qui định biểu thức để lọc lại các mẫu tin trong đối tượng Recordset loại Dynaset, snapshot hoặc forward only.
- RecordCount: trả về tổng số mẫu tin trong Recordset.
- Fields: tập hợp của các Field.
- Bookmark: là chuỗi ký tự được tạo ngẫu nhiên để nhận diện từng mẫu tin trong bảng dữ liệu. Giá trị của bookmark luôn luôn chỉ đến mẫu tin hiện hành.
- BOF: trả về giá trị True nếu vị trí mẫu tin hiện hành đứng trước mẫu tin đầu tiên, ngược lại trả về False nếu vị trí mẫu tin hiện hành đứng sau mẫu tin đầu tiên.
- EOF: trả về giá trị True nếu vị trí mẫu tin hiện hành đứng sau mẫu tin cuối cùng, ngược lại trả về False nếu vị trí mẫu tin hiện hành đứng trước mẫu tin cuối cùng.
- NoMatch: xác định kết quả tìm kiếm mẫu tin sau khi sử dụng phương thức Seek hoặc Find. Trả về True nếu không tìm thấy được mẫu tin.

##### 14.3.4.2 *Phương thức:*

- Close: đóng đối tượng Recordset.
- MoveFirst: dời về mẫu tin đầu tiên và mẫu tin này trở thành mẫu tin hiện hành.
- MoveLast: dời về mẫu tin cuối cùng và mẫu tin này trở thành mẫu tin hiện hành.
- MoveNext: dời đến mẫu tin kế tiếp và mẫu tin này trở thành mẫu tin hiện hành.
- MovePrevious: dời về mẫu tin đứng ngay trước và mẫu tin này trở thành mẫu tin hiện hành.



- FindFirst: tìm đến mẫu tin đầu tiên trong nhóm những mẫu tin đáp ứng điều kiện qui định của đối tượng Recordset loại

Dynaset hoặc Snapshot (chỉ dùng với Microsoft Jet workspaces).

- FindLast: tìm đến mẫu tin cuối cùng trong nhóm những mẫu tin đáp ứng điều kiện qui định của đối tượng Recordset loại Dynaset hoặc Snapshot (chỉ dùng với Microsoft Jet workspaces).
- FindNext: tìm đến mẫu tin kế tiếp mẫu tin hiện tại trong nhóm những mẫu tin đáp ứng điều kiện qui định của đối tượng Recordset loại Dynaset hoặc Snapshot (chỉ dùng với Microsoft Jet workspaces).
- FindPrevious: tìm đến mẫu tin ngay trước mẫu tin hiện tại trong nhóm những mẫu tin đáp ứng điều kiện qui định của đối tượng Recordset loại Dynaset hoặc Snapshot (chỉ dùng với Microsoft Jet workspaces).
- AddNew: thêm một mẫu tin mới vào Recordset.
- Update: lưu nội dung mẫu tin trong copy buffer vào Recordset.
- Seek: định vị một mẫu tin trong Recordset thuộc loại Table và Access Basic sẽ dùng chỉ mục hiện tại của bảng dữ liệu để so khớp mẫu tin. Chỉ mục này đã được định nghĩa trước và qui định hiện hành bằng tính chất Index của bảng dữ liệu.  
recordset.Seek toán\_tử\_so\_sánh , key1, key2,...key13
- Edit: chép mẫu tin hiện hành trong đối tượng Recordset vào copy buffer.
- Delete: xóa mẫu tin hiện hành trong đối tượng Recordset.
- Requery: cập nhật lại dữ liệu trong đối tượng Recordset bằng cách thi hành lại truy vấn trên đối tượng mà Recordset là cơ sở.

Ví dụ: Dùng FindFirst và FindNext để tìm mẫu tin

```
Dim Db As DAO.Database, Rcd As DAO.Recordset
```

```
Dim Quidinh As String, Dienthoai As String
```

```
Set Db = DBEngine.Workspaces(0).Databases(0)
```

```
Set Rcd = Db.OpenRecordset("Khach_hang", dbOpenDynaset)
```

```
Quidinh = " Dienthoai Is Null"
```

```
Rcd.FindFirst Quidinh
```

```
Do Until Rcd.NoMatch
```

```
MsgBox Rcd!TenKH & " chưa có số điện thoại"
```

```
" Dienthoai = InputBox("Nhập số điện thoại :")
```

```
Rcd.Edit
```

```
Rcd!Phone = Dienthoai
```

```
Rcd.Update
```

```
Rcd.FindNext Quidinh
```

```
Loop
Rcd.Close
Db.Close
```

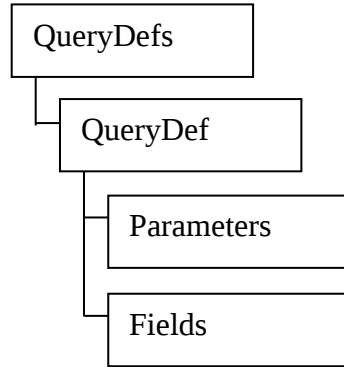
Ví dụ: Dùng Seek để tìm mẫu tin

```
Dim Db As DAO.Database, Rcd As DAO.Recordset
Dim Tim_MaNV As String
Dim varBookmark As Variant
Set Db = DBEngine.Workspaces(0).Databases(0)
Set Rcd = Db.OpenRecordset("Nhanvien", dbOpenTable)
With Rcd
    .Index = "PrimaryKey"
    Tim_MaNV = InputBox("Tìm mã nhân viên :")
    If Tim_MaNV <> "" Then
        VarBookmark = .Bookmark
        .Seek "=", Tim_MaNV
        If .NoMatch Then
            MsgBox "Không có nhân viên này"
            .Bookmark = varBookmark
        End If
    End If
End With
Rcd.Close
Db.Close
```

Ví dụ: Xoá các mẫu tin trong Table Nhanvien

```
Dim Db As DAO.Database
Dim Rcd As DAO.Recordset
Set Db = DBEngine.Workspaces(0).Databases(0)
Set Rcd = Db.OpenRecordset("Nhanvien", dbOpenTable)
Rcd.MoveFirst
Do Until Rcd.EOF
    If Rcd!Tinh_trang = "Nghỉ luôn" Then
        Rcd.Delete
    End If
    Rcd.MoveNext
Loop
Rcd.Close
Db.Close
```

### 14.3.5 Đối tượng QueryDef



#### 14.3.5.1 *Tính chất:*

- Fields: tập hợp các fields của đối tượng QueryDef.
- Parameters: tập hợp các tham số truy vấn của đối tượng QueryDef.
- Name: tên của đối tượng QueryDef.
- SQL: qui định hoặc trả về câu lệnh SQL mà sẽ thi hành bởi đối tượng QueryDef.

#### 14.3.5.2 *Phương thức:*

- OpenRecordset: mở một recordset loại dynaset hoặc snapshot từ bảng truy vấn đã có sẵn.
- Execute: thực hiện lệnh truy vấn quy định bởi tính chất SQL.
- Close: đóng đối tượng QueryDef.

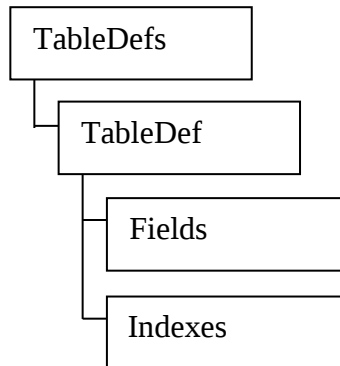
Ví dụ:

```
Dim Rcs As DAO.Recordset
Dim Qr As QueryDef
Dim Db As DAO.Database
Set Db = DBEngine.Workspaces(0).Databases(0)
Set Qr = Db.QueryDefs("Danh_sach_NVKT")
Set Rcs = Qr.OpenRecordset()
Dim MyQuery As QueryDef
Dim SQL_text As String
SQL_text = "SELECT * FROM Nhanvien WHERE MaPB='KD' "
Set MyQuery = Db.CreateQueryDef("Nhanvien_KD", SQL_text)
MyQuery.SQL = "UPDATE Luong SET Luong!Heso = 1.92 " _
    & " WHERE Mucluong!Hesoluong = 1.86 "
MyQuery.Execute
MyQuery.Close
Db.QueryDefs.Delete "Nhanvien_KD"
```

Ví dụ:

```
Dim dbs As DAO.Database
Dim qdf As QueryDef
Dim rst As DAO.Recordset
Dim prmBegin As Parameter, prmEnd As Parameter
Set dbs = OpenDatabase("QLNS.mdb")
Set qdf = dbs.CreateQueryDef("", _
    "PARAMETERS dteBegin DateTime, dteEnd DateTime; " & _
    "SELECT * FROM Nhanvien " & _
    "WHERE Ngayvaolam BETWEEN " & _
    "[dteBegin] AND [dteEnd] ORDER BY MaNV")
Set prmBegin = qdf.Parameters!dteBegin
Set prmEnd = qdf.Parameters!dteEnd
prmBegin = #1/1/95#
prmEnd = #1/1/99#
Set rst = qdf.OpenRecordset(dbOpenForwardOnly)
.....
rst.Close
dbs.Close
```

### 14.3.6 Đối tượng TableDef



#### 14.3.6.1 *Tính chất:*

- RecordCount: tổng số records trong bảng dữ liệu.
- Fields: tập hợp các fields trong bảng dữ liệu.
- Indexes: tập hợp các chỉ mục trong bảng dữ liệu.
- ValidationRule: qui định biểu thức kiểm tra trên một field nào đó khi thêm field vào bảng dữ liệu.
- ValidationText: qui định thông báo lỗi xuất hiện nếu dữ liệu nhập vào field không phù hợp với điều kiện kiểm tra.
- Name: tên của bảng dữ liệu.

Ví dụ:

```
Dim dbs As DAO.Database
Dim fld As Field
Set dbs = OpenDatabase("QLNS.mdb")
Set fld = dbs.TableDefs!Nhanvien.CreateField("Heso", dbDouble)
fld.ValidationRule = "BETWEEN 1.86 AND 7"
fld.ValidationText = "Hệ số lương từ 1.86 đến 7"
dbs.TableDefs!Nhanvien.Fields.Append fld
dbs.Close
```

#### 14.3.6.2 Phương thức:

- CreateField: tạo một field mới cho đối tượng TableDef.  
Set field = object.CreateField (name, type, size)
- CreateIndex: tạo một chỉ mục mới cho đối tượng TableDef.  
Set index = tabledef.CreateIndex(name)
- OpenRecordset: mở recordset loại table từ bảng dữ liệu có sẵn.  
Set recordset = object.OpenRecordset(type,options,lockedit)
- RefreshLink: cập nhật thông tin kết nối cho bảng dữ liệu link. Ví dụ:

```
Dim dbs As DAO.Database
Dim tdf As TableDef
Dim idxPB As Index
Dim idxHT As Index
Set dbs = OpenDatabase("QLNS.mdb")
Set tdf = dbs!Nhanvien
With tdf
    Set idxPB = .CreateIndex("PB")
    With idxPB
        .Fields.Append .CreateField("MAPB")
        .Fields.Append .CreateField("Ten")
        .Fields.Append .CreateField("Ho")
    End With
    .Indexes.Append idxPB
    Set idxHT = .CreateIndex
    With idxHT
        .Name = "Ten_Ho"
        .Fields.Append .CreateField("Ten")
    End With
End With
```

```

.Fields.Append .CreateField("Ho")
End With
.Indexes.Append idxHT
.Indexes.Refresh
.Indexes.Delete idxPB.Name
.Indexes.Delete idxHT.Name
End With
dbs.Close

```

#### 14.3.7 Những tính chất liên quan đến các đối tượng

Một số tính chất đặc biệt của các đối tượng trong MS Access có khả năng cho phép tham chiếu đến đối tượng khác có liên quan theo một phương thức nhất định. Bảng liệt kê dưới đây trình bày các tính chất đặc biệt của đối tượng đã có sẵn trong MS access:

Tính chất	Dùng với đối tượng	Tham chiếu đến
ActiveControl	Screen	Ô điều khiển nhận focus (đang được chọn).
ActiveForm	Screen	Mẫu biểu hiện hành hoặc mẫu biểu đang chứa ô điều khiển được chọn.
ActiveReport	Screen	Mẫu biểu hiện hành hoặc báo biểu đang chứa ô điều khiển được chọn.
Form	Mẫu biểu hoặc ô điều khiển chứa mẫu biểu con.	Đối với mẫu biểu: là chính mẫu biểu đó. Đối với ô điều khiển chứa mẫu biểu con: là mẫu biểu chứa trong ô điều khiển đó.
Me	Mẫu biểu hoặc báo biểu.	Mẫu biểu hoặc báo biểu.
Module	Mẫu biểu hoặc báo biểu.	Đơn thể của mẫu biểu hoặc báo biểu.
Parent	Ô điều khiển.	Mẫu biểu hoặc báo biểu chứa ô
PreviousControl	Screen	Ô điều khiển đã có focus ngay trước lúc chọn ô điều khiển hiện hành.

RecordSetClone	Mẫu biểu.	Tập hợp mẫu tin đã được dùng làm dữ kiện cơ sở của mẫu biểu (bảng dữ kiện hoặc bảng tra vấn).
Report	Báo biểu hoặc ô điều khiển chứa báo biểu con.	Đối với báo biểu: là chính báo biểu đó. Đối với ô điều khiển chứa báo biểu con: là báo biểu chứa trong ô điều khiển đó.
Section	Ô điều khiển.	Một phần của mẫu biểu hay báo biểu trong đó chứa ô điều khiển.

Ví dụ:

```
Sub ChangeFormColor(FormName As Form)
FormName.BackColor = RGB(Rnd * 256, Rnd * 256, Rnd * 256)
End Sub
```

Gọi thủ tục để thay đổi màu nền của Form  
dùng: ChangeFormColor Me

Ví dụ:

```
Sub MaNV_AfterUpdate()
Dim rst As DAO.Recordset
Set rst = Me.RecordsetClone
rst.FindFirst "[MANV] = '" & Me!MaNV & "'"
If rst.NoMatch Then
MsgBox "Không có mã nhân viên này"
Else
Me.Bookmark = rst.Bookmark
End If
```

```
rst.Close
End Sub
```

## 14.4 XỬ LÝ DỮ LIỆU THÔNG QUA SỰ KIỆN

### 14.4.1 Sự kiện về cửa sổ

Sự kiện	Đối tượng	Xảy ra khi	Được phép Cancel
---------	-----------	------------	---------------------



Close	Form Report	Khi đóng mẫu biểu hay báo biểu hoặc khi loại bỏ mẫu biểu hay báo biểu trên màn hình	Không
-------	-------------	---	-------

Load	Form	Khi một mẫu biểu được mở và những mẫu tin được hiển thị trong mẫu biểu. Sự kiện này xảy ra trước sự kiện Current nhưng theo sau sự kiện	Không
Open	Form Report	Khi mở một mẫu biểu hoặc báo biểu, nhưng trước khi hiển thị mẫu tin đầu tiên	Có
Resize	Form	Khi cỡ lớn của mẫu biểu thay đổi hoặc	Không
Unload	Form	Khi đóng mẫu biểu và những mẫu tin của nó không được nạp vào mẫu biểu, nhưng trước khi mẫu biểu bị	Có

#### 14.4.2 Sự kiện về Focus

Sự kiện	Xảy ra khi	Được phép Cancel
Activate	Khi mẫu biểu hay báo biểu trở thành một cửa sổ hoạt động	Không
Deactivate	Khi đóng mẫu biểu hay báo biểu, hoặc khi một cửa sổ của MS Access trở thành không hoạt động, nhưng trước khi một cửa sổ	Không
Enter	Trước khi một ô điều khiển nhận focus, hoặc từ một ô điều khiển khác trong cùng mẫu biểu chuyển sang hoặc khi mở mẫu biểu.	Không
Exit	Trước khi ô điều khiển mất focus và dời sang một ô điều khiển khác trong cùng mẫu biểu. Sự kiện này xảy ra trước sự kiện	Có
GotFocus	Khi một ô điều khiển hoặc một mẫu biểu không có ô điều khiển hoạt động hay cho phép chọn nào nhận được Focus. Mẫu biểu chỉ có thể nhận Focus nếu tất cả những ô điều khiển được hiển thị trên mẫu biểu đều không cho phép chọn hoặc	Không
LostFocus	Khi ô điều khiển hoặc mẫu biểu mất Focus Mẫu biểu chỉ có thể nhận Focus nếu tất cả	Không

	những ô điều khiển được hiển thị trên mẫu biểu đều không cho phép chọn hoặc không có ô điều khiển nào trong mẫu biểu.	
--	---	--

#### 14.4.3 Sự kiện về Data

Sự kiện	Xảy ra khi	Được phép Cancel
AfterDelConfirm	Sau khi người sử dụng xác nhận hoặc hủy bỏ tác vụ xóa mẫu tin.	Không
AfterInsert	Khi một mẫu tin mới được thêm vào bảng dữ liệu	Không
AfterUpdate	Sau khi một ô điều khiển hoặc mẫu tin được cập nhật với dữ liệu mới. Sự kiện này xảy ra khi ô điều khiển mất focus hoặc khi người sử dụng chọn menu Records, Save Record Sự kiện này xảy ra đối với cả những mẫu tin hiện hữu và những	Không
BeforeDelConfirm	Trước khi người sử dụng xác nhận tác vụ xóa dữ liệu. Người sử dụng thường xác nhận hoặc hủy bỏ tác vụ xóa mẫu tin bằng cách chọn OK hoặc	Có
BeforeInsert	Khi người sử dụng bấm ký tự đầu tiên vào một mẫu tin mới hoặc khi người sử dụng đặt để giá trị cho một ô điều khiển trong một mẫu tin mới, nhưng trước khi mẫu tin được	Có
BeforeUpdate	Trước khi ô điều khiển hoặc mẫu tin được cập nhật với dữ liệu mới. Sự kiện này xảy ra đối với những mẫu tin hiện hữu và cả những mẫu tin mới khi ô điều khiển hoặc mẫu tin mất focus hoặc khi người sử dụng chọn menu Records – Save	Có

Change	Khi nội dung của một ô điều khiển Textbox hoặc của một ô điều khiển Combo Box thay đổi, chẳng hạn như,	Không
--------	--	-------

	khi người sử dụng bấm ký tự đầu tiên vào ô điều khiển Textbox hoặc sửa đổi tính chất Text của ô điều khiển Textbox bằng cách dùng Macro hoặc lệnh ACCESS BASIC.	
Current	Khi bạn mở mẫu biểu lần đầu tiên hoặc khi focus rời khỏi một mẫu tin và chuyển sang mẫu tin khác. Sự kiện Current cũng xảy ra khi bạn Refresh lại dữ liệu hoặc Requery dữ liệu cơ sở của mẫu biểu. Chẳng hạn như, khi bạn chọn menu Records – Refresh, sử dụng hành động ShowAllRecords hoặc sử dụng hành động hay phương	Không
Delete	Khi bạn yêu cầu xóa mẫu tin nhưng trước khi xác nhận tác vụ.	Có
NotInList	Khi bạn bấm vào Combo Box một giá trị mà giá trị đó chưa có trong danh sách đã định tại Combo Box.	Không
Updated	Khi dữ liệu thuộc một đối tượng OLE đã bị sửa đổi.	Không

#### 14.4.4 Sự kiện về mouse

Sự kiện	Xảy ra khi	Được phép Cancel
Click	Khi người sử dụng ấn nút mouse bên trái và buông ra ngay (click) khi đầu hiệu mouse chỉ tại một ô điều khiển hoặc vùng trắng trong mẫu biểu.	Không
DblClick	Khi người sử dụng click đôi nút mouse bên trái tại một ô điều khiển hoặc một trắng trong mẫu biểu.	Có
MouseDown	Khi người sử dụng ấn nút mouse trong lúc chỉ điểm đang ở trên mẫu biểu hoặc một ô điều khiển. Việc hủy bỏ sự kiện MouseDown giúp bạn tránh được sự hiển thị menu tắt của mẫu biểu hoặc của ô điều khiển	Có

	bỏ sự kiện này bằng cách quy định 1 macro chứa hành động CancelEvent tại mục tính chất OnMouseDown của ô điều khiển hoặc mẫu biểu.	
MouseMove	Khi người sử dụng dời chỉ điểm mouse bên trên một mẫu biểu hoặc ô điều khiển.	Không
MouseUp	Khi người sử dụng buông nút mouse (sau khi đã ấn nút mouse) lúc chỉ điểm đang ở trên một mẫu biểu hoặc ô điều khiển.	Không

#### 14.4.5 Sự kiện về bàn phím

Sự kiện	Xảy ra khi	Được phép Cancel
KeyDown	<p>Khi người sử dụng ấn một phím bất kỳ trên bàn phím lúc ô điều khiển hoặc mẫu biểu nhận được focus, hoặc khi gửi một phím đến mẫu biểu hoặc một điều khiển bằng hành động hay hiệu lệnh SendKeys. Nếu phím bấm được giữ (không buông ra) thì sự kiện KeyDown sẽ lặp lại nhiều lần. Mặc dù không thể hủy bỏ sự kiện KeyDown nhưng vẫn có thể tránh không cho đối tượng nhận 1 phím bấm nào cả bằng cách quy định đối mục Keycode của thủ tục xử lý sự kiện tại tính chất KeyDown của đối tượng đó là 0.</p> <p>→ Thủ tục xử lý sự kiện KeyDown có dạng:</p> <pre> <b>Sub</b> Form_KeyDown(Keycode As Integer, Shift As Integer) ... <b>End Sub</b> </pre> <p>Hoặc</p> <pre> <b>Sub</b> Form_KeyDown(Keycode As Integer) ... <b>End Sub</b> </pre>	Không
KeyPress	Khi người sử dụng bấm một phím hoặc tổ hợp phím để biểu thị một ký tự ASCII lúc ô điều khiển hoặc mẫu biểu nhận được focus,	Có (Macro)

	<p>hoặc khi gọi một phím đến mẫu biểu hoặc một ô điều khiển bằng hành động hay hiệu lệnh SendKey.</p> <p>Nếu phím bấm được giữ (không buông ra) thì sự kiện KeyPress sẽ lặp lại nhiều lần. Có thể hủy bỏ sự kiện này bằng cách quy định 1 macro chứa hành động CancelEvent tại mục tính chất sự kiện OnKeyPress của mẫu biểu hoặc ô điều khiển. Cũng có thể tránh không cho đối tượng nhận 1 phím nào cả bằng cách quy định đối mục KeyAscii của thủ tục xử lý sự kiện KeyPress là 0.</p> <p>→ Thủ tục xử lý sự kiện KeyPress có dạng:  <b>Sub Form_KeyPress</b> (KeyAscii As Integer)  ...  <b>End Sub</b>  Hoặc  <b>Sub Ôđiềukhiển_KeyPress</b> (KeyAscii As Integer)  ...  ...</p>	
KeyUp	<p>Khi người sử dụng buông 1 phím đã bấm lúc ô điều khiển hoặc mẫu biểu nhận được focus hoặc khi gọi 1 phím đến mẫu biểu hoặc 1 ô điều khiển bằng hành động hay hiệu lệnh SendKey.</p> <p>Mặc dù không thể hủy bỏ sự kiện KeyUp nhưng vẫn có thể tránh không cho đối tượng nhận 1 phím bấm nào cả bằng cách quy định đối mục Keycode của thủ tục xử lý sự kiện tại tính chất KeyUp của đối tượng đó là 0.</p> <p>→ Thủ tục xử lý sự kiện KeyUp có dạng:  <b>Sub Form_KeyUp</b> (Keycode As Integer, Shift As Integer)  ...  <b>End Sub</b>  Hoặc  <b>Sub Ôđiềukhiển_KeyUp</b> (Keycode As Integer, Shift As Integer)  ...  ...</p>	Không

#### 14.4.6 Sự kiện về in ấn

Sự kiện	Xảy ra khi	Được phép Cancel
Format	Khi MS ACCESS xác định dữ liệu nào thuộc về thành phần của báo biểu và trước khi MS ACCESS định dạng thành phần đó để xem trước trên màn hình hoặc in ra.	Có
Print	Khi dữ liệu thuộc một thành phần của báo biểu được định dạng, nhưng trước khi in thành phần đó.	Có
Retreat	Khi MS ACCESS quay trở lại thành phần báo biểu đã định dạng trước đó trong quá trình định dạng báo biểu.	Không

#### 14.4.7 Sự kiện về sai lầm (error) và ấn định thời gian

Sự kiện	Xảy ra khi	Được phép Cancel
Error	Khi một sai lầm trong thời gian thi hành xảy ra do một hành động của người sử dụng. Sự kiện Error xảy ra đối với những sai lầm trong thời gian thi hành tạo nên bởi Microsoft Jet Database Engine, chứ không phải do ACCESS BASIC.	Có
Timer	Khi một khoảng thời gian ấn định đã trôi qua sau khi sự kiện Timer xảy ra lần cuối cùng trước đó, khoảng thời gian này được xác định bởi mục tính chất TimerInterval của mẫu biểu. Bạn có thể làm cho dữ liệu được đồng bộ hóa trong môi trường có nhiều người sử dụng bằng cách “làm mới” lại dữ liệu (Refresh Data) hoặc tái truy vấn (Requery) dữ liệu	Không

#### 14.4.8 Xử lý error khi thi hành chương trình

Khi hệ thống ứng dụng bắt đầu cài đặt sử dụng, có thể xảy ra các sai lầm mà người dùng gặp phải do những nguyên nhân bên ngoài tác động mà không thể dự đoán trước như: tập tin cần thiết vô tình bị xóa



do lầm lẫn, không còn đủ chỗ trống trên đĩa để ghi, ổ đĩa không có đĩa ... Những nguyên nhân này sẽ ảnh hưởng đến chương trình, tuy nhiên MS Access có thể phát hiện được khi hệ thống ứng dụng đang hoạt động. Để xử lý các sai lầm này phải viết thêm các lệnh điều khiển trong các thủ tục, còn gọi là lệnh để bẫy lỗi.

Mục đích của lệnh bẫy lỗi là khi sai lầm xảy ra trong lúc thi hành chương trình sẽ ngăn không cho hệ thống MS Access xử lý sai lầm đó mà thay thế bằng việc thi hành các lệnh xử lý viết thêm trong thủ tục. Tùy thuộc sai lầm xảy ra ở đâu, có thể dùng một trong hai cách sau đây:

Tình huống xảy ra sai lầm	Cách xử lý
Khi sử dụng giao diện MS Access hoặc MS Jet Database Engine	Thêm tập lệnh xử lý lỗi vào thủ tục sự kiện Error của mẫu biểu hay báo biểu.
Trong chương trình Access Basic	Thêm lệnh ON ERROR và tập lệnh xử lý lỗi vào thủ tục.

#### 14.4.8.1 Dùng thủ tục xử lý sự kiện Error

Thủ tục được qui định tại mục tính chất OnError của mẫu biểu hoặc báo biểu, khi sai lầm xảy ra thì thủ tục xử lý tại OnError sẽ thi hành.

Ví dụ:

Trong mẫu biểu dùng để nhập vào bảng dữ liệu Hoa\_don, có hai ô điều khiển nhập vào field SoHD và MaKH đòi hỏi phải nhập đầy đủ thông tin (field SoHD là khóa của bảng dữ liệu Hoa\_don và field MaKH có định nghĩa quan hệ phụ thuộc tồn tại với bảng dữ liệu Khách\_hang). Dưới đây là thủ tục xử lý tại mục tính chất OnError của mẫu biểu:

```
Sub Form_Error ( DataErr As Integer, Response As Integer)
Dim TB As String
NewLine As String
Const Err_Join = 3101      ' lỗi khi liên kết các bảng bị vi phạm
Const Err_NullKey = 3058  ' lỗi khi khóa của bảng có giá trị
Null NewLine = Chr(13) & Chr(10)
Select Case DataErr
Case Err_Join :
    TB = " Mã khách hàng trong hóa đơn không có "
    TB = TB & NewLine
```

```
TB = TB & " Hãy nhập lại Mã khách hàng  
" TB = TB & NewLine  
TB = TB & " hoặc hủy bỏ bằng cách nhấn Esc  
" MsgBox TB  
Response = DATA_ERRCONTINUE
```

Case Err\_NullKey :

```
TB = " Hóa đơn chưa có số hóa đơn hoặc bị  
trùng" TB = TB & NewLine  
TB = TB & " Hãy nhập lại số hóa đơn  
" TB = TB & NewLine  
TB = TB & " hoặc hủy bỏ bằng cách nhấn Esc  
" MsgBox TB  
SoHD.SetFocus  
Response = DATA_ERRCONTINUE
```

End Select

End Sub

#### 14.4.8.2 Dùng lệnh On Error

Để xử lý lỗi lúc thi hành chương trình Access Basic bằng cách thêm lệnh On Error và tập lệnh xử lý sai lầm vào chương trình đó. Cú pháp lệnh như sau:

```
On Error Goto nhãn_1  
.....  
nhãn_1:  
.....  
Resume nhãn_2
```

- Khi xảy ra sai lầm chương trình sẽ chuyển đến dòng lệnh có nhãn\_1 và thực hiện tập lệnh xử lý lỗi tại đó.
- Lệnh Resume nhãn\_2 sẽ cho thi hành tiếp chương trình bắt đầu tại dòng lệnh có nhãn\_2

Ví dụ:

Trong mẫu biểu nhập vào bảng dữ liệu Nhanvien có nút lệnh Xem cho phép mở một mẫu biểu khác tên là QTCT khi người sử dụng click vào. Nếu mẫu biểu QTCT không tồn tại sẽ xảy ra sai lầm. Thủ tục sau đây gắn tại mục tính chất OnClick của nút Xem:

Nhân Viên	
Mã nhân viên	NV01
Họ tên	Nguyễn Văn Huy
Phái nam	<input checked="" type="checkbox"/>
Ngày sinh	01/01/1980
Mức lương	500
Phòng ban	KT

```

Sub Xem_Click()
    On Error Goto Err_Click
    Dim frm As String frm
    = "QTCT"
    DoCmd.OpenForm frm
Exit_Xem :
    Exit Sub
Err_Click :
    MsgBox "Không có form này"
    Resume Exit_Xem
End Sub

```

## BÀI TẬP:

1. Tạo tập tin CSDL để quản lý kinh doanh các loại nước giải khát đặt tên QLKD.MDB trong đó tạo các table sau:

CUA\_HANG (**MACH**, TENCH, DIACHI, CHIETKHAU)  
NUOCGIAIKHAT (**NGK**, TENNGK, LOAI, GIABAN)  
HD\_GIAOHANG (**SOHD**, NGAYGIAO, MACH)  
CT\_GIAOHANG (**SOHD**, **NGK**, SOKET)

*Ghi chú:* các field có gạch dưới là khóa của table.

*Ý nghĩa của các table:*

- Table CUA\_HANG: lưu danh sách các cửa hàng mua nước giải khát. Mỗi cửa hàng có các thông tin: MACH (mã cửa hàng), TENCH (tên cửa hàng), DIACHI (địa chỉ), CHIETKHAU (tỷ lệ chiết khấu cho cửa hàng).
  - Table NUOCGIAIKHAT: lưu danh sách các sản phẩm giải khát, gồm: NGK (mã số nước giải khát), TENNGK (tên sản phẩm nước giải khát), LOAI (loại chai), GIABAN (giá bán 1 chai).
  - Table HD\_GIAOHANG: lưu các hóa đơn giao nước giải khát cho các cửa hàng, gồm SOHD (mã số hóa đơn giao hàng), NGAYGIAO (ngày giao hàng), MACH (mã cửa hàng mua nước giải khát).
  - Table CT\_GIAOHANG: lưu nội dung chi tiết của hóa đơn giao hàng, gồm: SOHD (mã số hóa đơn giao hàng), NGK (mã số nước giải khát), SOKET (số lượng kết hàng, mỗi kết chứa 30 chai).
2. Tạo các table và thiết lập mối liên kết giữa các table. Dựa vào dữ liệu mẫu, hãy chọn kiểu dữ liệu phù hợp cho các field trong table. Qui định các tính chất Format, Input Mask cần thiết.
    - Tạo combo box trong table HD\_GIAOHANG tại field MACH tham chiếu đến field MACH trong table CUA\_HANG (combo box hiện 2 cột MACH và TENCH).
    - Tạo combo box trong table CT\_GIAOHANG tại field NGK tham chiếu đến field NGK trong table NUOCGIAIKHAT (combo box hiện 2 cột NGK và TENNGK).
    - Cài đặt các ràng buộc: CHIETKHAU từ 0 đến 30%, NGAYGIAO nhỏ hơn hay bằng ngày hiện hành, GIABAN từ 2000 trở lên, SOKET phải là số dương.
    - Nhập dữ liệu mẫu vào các table:

	MACH	TENCH	DIACHI	CHIETKHAU
	C001	Nhà hàng Bảo Lan	327 Nguyễn Văn Trỗi, PN	20%
	C002	Cửa hàng Thuận Kiều	134 Đào Duy Từ, Q5	15%
	C003	Trung tâm DN Đức Trí	805 Xô Viết Nghệ Tĩnh, BT	10%
	C004	Công ty Hoàn Thành	69 Tôn Thất Tùng, Q1	10%
	C005	Trống Đồng	471 Cách Mạng Tháng 8, Q3	20%
	C006	Bách hóa Thịnh Phát	562 Điện Biên Phủ, Q3	25%
▶	C007	Bán lẻ		0%

Record: 7 of 7

	NGK	TENNGK	LOAI	GIABAN
	C24	Cocacola	24	3000
▶	F24	Fanta	24	3000
	P28	Pepsi	28	4000
	S28	Sprite	28	4500
	S33	Seven up	33	5000

Record: 2 of 5

	SOHD	NGAYGIAO	MACH
▶	001-09	15/09/05	C003
	002-09	15/09/05	C004
	003-09	22/09/05	C006
	004-09	27/09/05	C003
	005-10	02/10/05	C001
	006-10	08/10/05	C007
	007-10	08/10/05	C002

Record: 1 of 7

	SOHD	NGK	SOKET
▶	001-09	C24	10
	001-09	F24	5
	002-09	P28	6
	003-09	C24	15
	003-09	S28	4
	003-09	S33	2
	004-09	F24	12
	005-10	P28	8
	005-10	S28	10
	006-10	C24	1
	007-10	C24	5
	007-10	F24	7
	007-10	S28	10

Record: 1 of 13

### 3. Tạo các truy vấn sau:

- Cho biết nước giải khát đã bán nhiều nhất
- Liệt kê các nước giải khát có cùng loại và cùng đơn giá.
- Liệt kê các cửa hàng chưa mua hàng trong tháng 10/05
- Liệt kê các cửa hàng, tổng doanh số bán, tổng chiết khấu.
- Liệt kê các cửa hàng, tên nước giải khát và tổng số chai bán nhiều nhất cho mỗi cửa hàng.
- Liệt kê các số hóa đơn giao hàng đồng thời hai loại nước giải khát

Cocacola và Fanta

- Liệt kê mã cửa hàng, tên cửa hàng, tổng doanh số, tên các nước giải khát (mỗi nước giải khát là một cột và trong các cột này chứa tổng số chai của nước giải khát đó)

### 4. Tạo Form theo mẫu sau với các yêu cầu:

QUẢN LÝ KINH DOANH						
Mã cửa hàng	<input type="text" value="C001"/>	Hóa đơn giao hàng				
Tên cửa hàng	<input type="text" value="Nhà hàng Bảo Lan"/>	<input type="checkbox"/>	<b>SOHD</b>	<input type="checkbox"/>	<b>NGAYGIAO</b>	
Địa chỉ	<input type="text" value="327 Nguyễn Văn Trỗi, PN"/>	<input type="checkbox"/>	▶ 004-09	<input type="checkbox"/>	27/09/05	
Tỷ lệ chiết khấu	<input type="text" value="20%"/>	<input type="checkbox"/>	005-10	<input type="checkbox"/>	02/10/05	
Chi tiết giao hàng			<input type="checkbox"/>	*		
	<b>NGK</b>	<b>TENNGK</b>	<b>LOAI</b>	<b>SOKET</b>	<b>GIABAN</b>	<b>THANHTIEN</b>
	▶ F24	Fanta	24	12	3,000	1,080,000
	P28	Pepsi	28	8	4,000	960,000
	*					
Tổng thành tiền					<input type="text" value="2,040,000"/>	
Trừ chiết khấu					<input type="text" value="408,000"/>	
<input type="button" value="Đầu"/>	<input type="button" value="Cuối"/>	<input type="button" value="Kế"/>	<input type="button" value="Trước"/>	<input type="button" value="Nhập"/>	<input type="button" value="In"/>	<input type="button" value="Đóng"/>

- Tạo macro hoặc viết thủ tục sự kiện để kiểm tra MACH không được trùng. Nếu trùng thì hiện thông báo “Mã cửa hàng này đã có” và ngăn con trỏ đi đến field kế.
- Tạo 2 textbox hiện tổng thành tiền và tiền chiết khấu (bằng tổng thành tiền \* tỷ lệ chiết khấu)
- Khi nhấn các nút di chuyển thì record sẽ di chuyển tương ứng.  
Nút nhập sẽ cho nhập thêm record, nút đóng sẽ cho đóng form lại.
- Khi bấm nút In thì sẽ in ra report theo mẫu sau:



5. Tạo Form xử lý để in báo cáo như sau:

BAO CAO

BÁO CÁO DOANH THU

Chọn in

Theo mặt hàng

Tất cả

Chọn mặt hàng

In ra

Màn hình

Mất in

In báo cáo

Đóng

- Viết thủ tục sự kiện để gắn vào nút lệnh In báo cáo cho phép in ra report với nội dung hiển thị tùy thuộc vào sự chọn lựa trong Form (In tất cả hay chỉ in theo một mặt hàng được chọn).
- Viết thủ tục gắn vào nút Đóng để chấm dứt việc in báo cáo.

## BÀI TẬP CUỐI KHÓA

1. Tạo tập tin CSDL đặt tên QLKHO.MDB trong đó tạo các table sau:

VATTU (MAVT, TENVT, DVT, TON)

NHAPXUAT (SOCT, NGÀY, MAVT, SOLUONG)

Ý nghĩa của các table:

- Table VATTU: lưu danh sách các vật tư. Mỗi vật tư có các thông tin: MAVT (mã vật tư), TENVT (tên vật tư), DVT (đơn vị tính), TON (số lượng tồn của vật tư).
- Table NHAPXUAT: lưu các chứng từ nhập xuất vật tư. Mỗi chứng từ có: SOCT (số chứng từ, đánh số theo qui luật phiếu nhập bắt đầu là N và phiếu xuất bắt đầu là X), NGÀY (ngày nhập hoặc xuất), MAVT (mã vật tư), SOLUONG (số lượng nhập hay xuất).

2. Tạo các table, các field có gạch dưới là khóa của table

- a. Dựa vào dữ liệu mẫu thí sinh tự chọn kiểu dữ liệu phù hợp cho các field trong table. Qui định các tính chất Format, Input Mask cần thiết.
  - Tạo combo box trong table NHAPXUAT tại field MAVT tham chiếu đến field MAVT trong table VATTU (combo box hiện 3 cột MAVT, TENVT, TON).
  - Tạo combo box trong table VATTU tại field DVT tham chiếu đến các giá trị: cái; kg; lít; thùng; tấm; mét ... (combo box hiện 1 cột).
- b. Cài đặt các ràng buộc: NGÀY nhỏ hơn hay bằng ngày hiện hành, TON từ 0 trở lên, SOLUONG phải là số dương.
- c. Lập mối liên kết giữa các table và nhập dữ liệu mẫu sau:

	MAVT	TENVT	DVT	TON
	+ VT001	Gạch men	Thùng	10
	+ VT002	Gạch ống	Viên	200
	+ VT003	Khung sắt	Mét	24
	+ VT004	Sơn nước	Lít	140
	+ VT005	Xi măng	Bao	75
	+ VT006	Ván ép	Tấm	0
	+ VT007	Cây 5x10	Cây	16
▶				0



Table NHAPXUAT

SOCT	NGAY	MAVT	SOLUONG
N001	02/05/06	VT001	30
N002	14/06/06	VT004	160
N003	20/06/06	VT001	75
X001	16/05/06	VT002	12
X002	25/06/06	VT001	20
X003	12/07/06	VT005	50

3. Tạo Form (tiêu đề form hiện tên sinh viên) theo mẫu sau với các yêu cầu:
- Khi mở form thì các text box, combo box đều vô hiệu hóa. Khi nhấn nút Thêm thì mới có hiệu lực để nhập.
  - Khi chọn mã vật tư trong combo box thì tên vật tư tương ứng hiện ở textbox bên cạnh. Textbox ngày hiện sẵn ngày hiện tại.
  - Textbox số chứng từ không cho nhập mà phát sinh giá trị theo qui luật khi chọn một loại chứng từ nào đó trong option group loại chứng từ.
  - Khi nhập xong bấm nút Lưu thì tùy thuộc chứng từ là nhập hay xuất thì sẽ hiện trong danh sách Nhập hay Xuất bên dưới. (1đ)
  - Nhấp vào textbox Tổng thì tính ra số lượng tồn của vật tư căn cứ vào lượng nhập và xuất.

**NHẬP XUẤT VẬT TƯ**

Mã vật tư: VT001    Gạch men    Tổng: 85

Ngày:    Số chứng từ:    Loại chứng từ:  Nhập     Xuất

SOCT	NGÀY	MAVT	SOLUONG
N001	2/5/2006	VT001	30
N002	14/6/2006	VT004	160
N003	20/6/2006	VT001	75

SOCT	NGÀY	MAVT	SOLUONG
X001	16/5/2006	VT002	12
X002	25/6/2006	VT001	20
X003	12/7/2006	VT005	50
X004	30/7/2006	VT004	25

Thêm    Lưu    In    Thoát

4. Tạo report theo mẫu sau, khi bấm nút **In** trên form thì hiện report ở dạng preview với yêu cầu: nếu combo box mã vật tư có chứa mã vật tư nào thì chỉ in nhập xuất với đúng vật tư đó. Còn ngược lại ô này đang trống thì in nhập xuất với toàn bộ thông tin đang có



## ĐỀ THI MẪU:

Thời gian : 90 phút – Không sử dụng tài liệu

Học sinh tạo folder theo qui định sau: D:\HỌtênHS\_LỚP

Câu 1: Tạo tập tin CSDL đặt trong folder, sau đó tạo các table sau đây:

field gạch dưới là khóa chính (1đ)

a. Table LOAIHV

Field Name	Data Type	Field Size	Ghi chú
<u>MALOA</u>	Text	2	Mã loại
DIEN	Text	20	Diện học viên
GIAM	Number	Single	Tỷ lệ giảm, từ 0 – 50 %

b. Table LOP

Field Name	Data Type	Field Size	Ghi chú
<u>MALOP</u>	Text	3	Mã lớp, nhập 1 chữ và 2 số
TENLOP	Text	30	Tên lớp
NGAYKG	DateTime		Ngày khai giảng
HOCPhi	Number	Long Integer	Học phí, từ 100000 trở lên

c. Table HOCVIEN

Field Name	Data Type	Field Size	Ghi chú
<u>MAHV</u>	Text	4	Mã học viên
HO TEN	Text	30	Họ và chữ lót
PHAINU	Text	10	Tên học viên
	YesNo		Phái nữ là Yes - phái nam là No
NGAYSINH	DateTime		Ngày sinh
MALOA	Text	2	Mã loại, combo box tham chiếu đến table LOAIHV

d. Table DANGKY

Field Name	Data Type	Field Size	Ghi chú
<u>MAHV</u>	Text	4	Mã học viên, combo box chọn từ table HOCVIEN
<u>MALOP</u>	Text	3	Mã lớp, combo box chọn từ table LOP
DIEM	Number	Byte	Điểm thi : từ 0 đến 10

**Câu 2:** Thiết lập mối quan hệ giữa các Table, sau đó nhập dữ liệu (1đ)

Table LOAIHV

MALOI	DIEN	GIAM
G1	Chính sách	50 %
G2	HS vùng sâu	20 %
G3	HSSV	10 %
G4	Khác	0 %

Table LOP

MALOP	TENLOP	NGAYKG	HOCPhi
L01	Tin học văn phòng	08/04/07	120 000
L02	C++	15/04/07	200 000
L03	AutoCad	15/04/07	250 000
L04	Access	04/05/07	200 000
L05	Sửa chữa cài đặt	04/05/07	300 000
L06	Mạng máy tính	20/05/07	250 000

Table HOCVIEN

MAHV	HO	TEN	PHAINU	NGAYSINH	MALOI
HV01	Trần Phước	Huy		25/09/86	G1
HV02	Phan Bích	Vân	➤	29/04/87	G2
HV03	Hoàng Xuân	Lan	➤	21/05/88	G3
HV04	Nguyễn Minh	Vũ		17/04/87	G4
HV05	Phạm Nguyễn	Thịnh		23/12/86	G1
HV06	Trần Tấn	Vũ		11/12/85	G3
HV07	Đoàn Kim	Trang	➤	13/06/87	G4
HV08	Nguyễn Trung	Nghĩa		18/08/89	G4

Table DANGKY

MAHV	MALOP	DIEM
HV01	L01	9
HV01	L02	4
HV01	L04	8
HV02	L01	4
HV02	L05	9
HV03	L01	7
HV03	L03	4
HV04	L02	10
HV04	L03	8
HV05	L02	5
HV05	L05	4
HV06	L04	7
HV06	L06	6
HV07	L04	9
HV08	L02	8
HV08	L03	6

**Câu 3:** Thực hiện các truy vấn sau: (3đ)

- Liệt kê mã lớp, tên lớp, ngày khai giảng, có số học viên đăng ký đông nhất trong 1 tháng bất kỳ nhập vào.
- Liệt kê mã học viên, họ, tên, ngày sinh mà đăng ký học đồng thời lớp C++ và AutoCad.
- Liệt kê tên lớp, tổng số học viên, số học viên giỏi, số học viên khá, số học viên trung bình, số học viên yếu (với điểm 9-10 là giỏi, 7-8 là khá, 5-6 là trung bình, dưới 5 là yếu).

**Câu 4:** Tạo form theo mẫu sau: (3đ) với yêu cầu sau:

- tiêu đề form là họ tên học sinh làm bài thi
- Trong khi nhập kiểm tra mã lớp không được trùng. Khi nhập đăng ký thì kiểm tra mã học viên của cùng 1 lớp không được trùng.

Mã lớp	<input type="text" value="L01"/>		
Tên lớp	<input type="text" value="Tin học văn phòng"/>		
Ngày khai giảng	<input type="text" value="08/04/07"/>		
Học phí	<input type="text" value="120,000 đ"/>	Số học viên	<input type="text" value="3"/>

	MAHV	HO	TEN	NU	NGAYSINH	DIEM
▶	HV01	Trần Phước	Huy	<input type="checkbox"/>	25/09/86	9
	HV02	Phan Bích	Vân	<input checked="" type="checkbox"/>	29/04/87	4
	HV03	Hoàng Xuân	Lan	<input checked="" type="checkbox"/>	21/05/88	7
*				<input type="checkbox"/>		

Đầu	Cuối	Kế	Trước	In	Thoát	Đạt	<input type="text" value="2"/>
						Không đạt	<input type="text" value="1"/>

**Câu 5:** Tạo report theo mẫu: (2đ) Khi bấm nút In thì in bảng điểm của lớp hiện tại, ngược lại sẽ không in.

Trường CDKT Cao Thắng

## BẢNG ĐIỂM

Mã lớp: L01

Tên lớp: Tin học văn phòng

Ngày khai giảng: 25/9/86

Học phí: 120,000 đ

STT	Họ	Tên	Ngày sinh	Điểm	Xếp loại
1	Trần Phước	Huy	25/9/86	9	Giỏi
2	Hồng Xuân	Lan	21/5/88	7	Khá
3	Phan Bích	Vân	29/4/87	4	Yếu

Phụ Lục:

## Các thuộc tính của Form và Control

### I. CÁC THUỘC TÍNH CỦA FORM

**1. Trang Format:** là trang chức các thuộc tính về định

dạng. Caption: chứa một tên gọi khác của Form.

Default View: chọn một chế độ cho Form (Single Form: chế độ đơn, Continuous Forms: chế độ nối tiếp, DataSheet: chế độ bảng).

View Allowed: chọn một chế độ có thể xem Form (DataSheet: chế độ

bảng, Form: chế độ Form, Both: có thể cả hai).

Scroll Bars: chọn các dạng thanh cuộn cho Form (Neither: không có, Vertical Only: chỉ có thanh cuộn dọc, Horizontal Only: chỉ có thanh cuộn ngang, Both: có cả 2 thanh cuộn ngang dọc).

Record Selectors: (Yes / No) cho hoặc không cho phép chọn Record trên Form.

Navigation Buttons: (Yes / No) Có hoặc không có các nút thay đổi

Record trên Form (nút ở đáy Form).

Dividing Lines: (Yes / No) Có đường phân chia các thành phần trên

Form hay không.

Auto Resize: (Yes / No) Có hoặc không chế độ tự động thay đổi kích thước.

Yes: Khi mở Form, Form sẽ tự động thay đổi kích thước để hiện

đầy đủ dữ liệu của Record.

No: Form không tự động thay đổi kích thước được.

Auto Center: (Yes / No) có hoặc không tự động canh giữa Form trên Màn hình.

Border Style: chọn 1 dạng đường viền cho Form (None: không có, Thin: đường mỏng, Resize: có thể thay đổi kích thước Form, Dialog: đặt Form ở chế độ đối thoại).

Control Box: (Yes / No) có hoặc không có Control Box trên Form.

Max Min Buttons: chọn bố trí các nút phóng to, thu nhỏ trên Form (None: không có, Both Enabled: có cả 2 nút, Min Enabled: có nút thu nhỏ, Max Enabled: có nút phóng to).

Close Button: (Yes / No) có hoặc không có nút đóng Form.

Whats This Button: (Yes / No) có hoặc không có nút ? (hỏi) trên Form. Width: qui định chiều rộng của Form.

Picture: cho phép chọn 1 hình ảnh chèn vào Form.

Picture Type: chọn 1 kiểu cho hình ảnh (Embedded: Nhúng, Linked: Liên kết).



Picture Size Mode: chọn 1 chế độ kích thước cho hình ảnh (Clip: hình ảnh bị cắt, Stretch: hình ảnh tự kéo giãn cho vừa, Zoom: hình ảnh được nhìn qua khung chứa).

Picture Aligment: chọn 1 vị trí bố trí hình ảnh (Top Left: góc bên trái, Top Right: góc bên phải, Center: Ở giữa khung, Bottom Left: góc dưới trái, Bottom Right: góc dưới phải, Form Center: chính giữa Form).

Picture Tiling: (Yes / No) có hoặc không có “Hoa văn“ cho hình ảnh. Grid X: chỉ vị trí tọa độ X cho Form.

Grid Y: chỉ vị trí tọa độ Y cho Form.

Layout for Print: (Yes / No) dùng Font máy in hay sử dụng Font màn hình khi in.

Pallete Source: cho phép ta chỉ định 1 tập tin màu dùng cho các màu trên Form.

## 2. **Trang Data**: là trang chức các thuộc tính về dữ liệu.

Record Source: chứa tên Bảng / Query là nguồn dữ liệu cho Form.

Filter: là tên của bộ lọc hoặc biểu thức lọc dữ liệu cho Form. Order By: chứa tên Field sắp xếp trên Form.

Allow Filters: (Yes / No) cho hoặc không cho phép lọc dữ liệu.

Allow Edits: (Yes / No) cho hoặc không cho phép sửa dữ liệu.

Allow Deletions: (Yes / No) cho hoặc không cho phép xóa dữ liệu.

Allow Additions: (Yes / No) cho hoặc không cho phép nhập mới dữ liệu. Data Entry: (Yes / No) có hoặc không cho phép đến Record mới khi mở Form.

RecordSet Type: chọn một kiểu cho RecordSet.

Record Locks: chọn 1 cách khóa Record trên Form (Nolocks: không khoá, All Records: khóa tất cả các Record, Edited Record: chỉ khóa Record đang sửa).

## 3. **Trang Event**: là trang chức các thuộc tính về sự kiện.

On Current: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi Form trở thành hiện hành.

Before Insert: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra trước khi thêm vào một Record mới.

After Insert: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra sau khi thêm vào một Record mới.

Before Update: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra trước khi

Form cập nhật một Record vào bảng.

After Update: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra sau khi

Form cập nhật một Record vào bảng.

On Delete: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi xóa một Record.

Before Del Confirm: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra trước khi xác nhận xóa một Record.

After Del Confirm: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra sau khi xác nhận xóa một Record.

On Open: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi mở Form. On Load: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi Form đã mở và các Record được nạp.

On Unload: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi trước khi Form biến mất khỏi màn hình.

On Activate: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi Form trở thành cửa sổ hoạt động.

On Deactivate: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi cửa sổ Form không hoạt động và trước khi Form được đóng lại.

On Close: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi đóng Form.

On Got Focus: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi Form nhận con trỏ.

On Lost Focus: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi Form mất con trỏ.

On Click: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi Click chuột trong Form.

On Dbl Click: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi Double Click chuột trong Form.

On Mouse Down: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi nhấn chuột.

On Mouse Move: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi di chuyển chuột trong Form.

On Key Down: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi nhấn giữ Phím trong Form.

On Key Up: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi nhả Phím trong Form.

Key Preview: (Yes/No) qui định việc nhận xử lý sự kiện bàn phím xảy ra Yes: Form sẽ nhận sự kiện bàn phím trước khi Control nhận được sự kiện bàn phím.

No: Chỉ có Control nhận được sự kiện Bàn phím.

On Error: chứa tên Macro hoặc thủ tục cho sự kiện khi xảy ra một lỗi trên

Form (Run Time).

On Filter: chứa tên Macro hoặc thủ tục của sự kiện lúc thực hiện tạo

Filter (bộ lọc) cho Form.

On Apply Filter: chứa tên Macro hoặc thủ tục của sự kiện lúc thực hiện lọc trên Form.

Timer Interval: chứa một giá trị số tính bằng Mili giây dùng qui định chu kỳ thời gian cho sự kiện On Timer xảy ra.

On Timer: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra lúc lặp lại khoảng thời gian (Timer Interval) trong quá trình làm việc trên Form.

#### 4. **Trang Other**: là trang chức các thuộc tính khác.

Pop Up: (Yes / No) có hoặc không qui định thuộc tính Pop Up (Form có thuộc tính Pop Up sẽ được sắp trên các Form khác).

Modal: (Yes / No) có hoặc không qui định thuộc tính Modal (Form có thuộc tính Modal sẽ được giữ chặt con trỏ không thể chuyển sang các Form khác khi nó chưa được đóng lại).

Cycle: chỉ định chu trình chuyển của con trỏ.

*All Records*: con trỏ lần lượt từ Control này sang control khác và từ Record này sang Record khác.

*Current Record*: con trỏ chạy theo chu trình trong Record hiện hành

*Current Page*: con trỏ chạy theo chu trình trong trang hiện hành.

Menu Bar: chứa tên của Macro tạo Menu User, Menu này sẽ hiện khi mở Form.

Tool Bar: chứa tên của ToolBar sẽ hiện mỗi khi mở Form.

ShortCut Menu: (Yes / No) cho phép hoặc không cho phép hiện Menu tắt khi Right Click chuột.

ShortCut Menu Bar: chứa tên của Menu Short Cut (menu tắt).

Fast Laser Printing: (Yes / No) có hoặc không cho phép in nhanh không?

Yes: quá trình in diễn ra nhanh nhưng các đường thẳng và khung được thay bởi các ký tự vẽ khung thông thường.

No: không thể in nhanh, các đường thẳng và khung được in

đúng. Help File: chứa tên của File Help.

Help Context Id: chứa chỉ mục trong Help.

Tag: có thể là chuỗi chứa thông tin bất kỳ của Form (chuỗi ≤ 2048 ký tự)

Has Module: (Yes/No) Form có hay không chứa Module chương trình nào.

## II. CÁC THUỘC TÍNH CỦA CONTROL

Phần mô tả này có tính tổng quát cho nhiều loại Control khác nhau, một số thuộc tính có thể không có trong Control loại này nhưng lại có trong Control loại khác.

Name: chứa tên đặt cho Control.

Caption: chứa chuỗi hiện trên Control.

Control Source: nguồn dữ liệu (từ Field nào của Table). Column Count: là số nguyên qui định số Field. Column Header: (Yes / No) có tiêu đề Field hay không ? Column Width: qui định độ rộng Control. Format: chứa các ký tự định dạng số liệu của Control. Decimal Place: qui định dạng thập phân. Input Mask: chứa các ký tự làm khuôn nhập liệu. Default Value: chứa giá trị ngầm định. Display Type: qui định kiểu hiển thị (Icon, Content). Validation Rule: chứa biểu thức kiểm chứng dữ liệu nhập.

Validation Text: chứa chuỗi thông báo khi biểu thức kiểm chứng bị vi phạm.

Status Bar Text: chứa chuỗi nhắc nhở, chuỗi này sẽ hiện ở thanh trạng thái cuối màn hình.

Auto Tab: (Yes/No) có tự động chuyển con trỏ từ Control này sang Control khác khi nhập đủ dữ liệu hay không.

Enter Key Behavior: (Default / New Line in Field) cho phép tạo dòng mới trong control (dùng cho các Field Memo)

**Default**: khi nhập dữ liệu muốn xuống dòng để nhập dòng mới phải nhấn Ctrl+Enter.

**New Line in Field**: khi nhập dữ liệu, muốn xuống dòng để nhập dòng mới chỉ nhấn Enter.

Auto Active: (Manual / Double Click) kích hoạt khi nào ?

Allow Auto Correct: (Yes / No) cho hoặc không cho phép tự động sửa. Visible: (Yes / No) hiện hoặc ẩn Control.

Display When: (Always / Print only / Screen only) hiển thị khi nào ?

Enable: (Yes / No) cho phép sử dụng control ?

Lock: (Yes / No) khoá hay không ?

Filter Lookup: (DataBase / Never / Always) có áp dụng lọc và tìm kiếm ? Tab Stop: (Yes / No) có dừng trong Control khi nhấn phím Tab ?

Tab Index: chứa một số nguyên (tính từ 0) là thứ tự dừng của Tab. Scroll Bar: (None / Vertical) không hoặc có thanh cuộn dọc.

Can Grow: (Yes / No) có hoặc không thể phát triển kích thước. Can Shrink: (Yes / No) có hoặc không thể thu nhỏ kích thước.

Left: tọa độ góc trái.

Top: tọa độ góc trên.

Width: kích thước chiều ngang.

Height: kích thước chiều cao.

Back Color: qui định màu nền.

Special Effect: qui định các tác động khác (Flat, Raise, Sunken ...).

Border Style: qui định đường viền (Clear, Normal).



Border Color: qui định màu đường viền.

Bound Column: chứa số nguyên qui định Control được gắn với Field thứ mấy ?

Border Width: độ dày của đường viền. Border Line Style: kiểu đường viền. Fore Color: qui định màu chữ.

Font Name: qui định Font chữ.

Font Size: qui định cỡ chữ.

Font Weight: qui định nét chữ (độ dày).

Font Italic: (Yes / No) chữ nghiêng hay không ?

Font Underline: (Yes / No) có gạch chân hay không ?

Text Align: chọn các canh biên (General, Left, Right, Center).

Help Context Id: chỉ mục trợ giúp trong Help.

Update Options: chọn lựa cách cập nhật (Automatic, Manual).

OLE Type Allowed: cho phép kiểu đối tượng OLE (Link, Embedded, Either).

Row Source Type: chọn kiểu dữ liệu nguồn cho Control (Table/Query, Value List, Field List)

Row Source: chứa dòng dữ liệu nguồn, có thể là lệnh SQL, danh sách các giá trị, tên bảng nguồn.

List Row: chứa số nguyên qui định số dòng hiện tối đa trong danh sách. List Width: qui định độ rộng của danh sách.

Before Update: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra trước khi cập nhật.

After Update: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra sau khi cập nhật.

On Change: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi sửa đổi dữ liệu.

On Enter: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi nhấn Enter. On Exit: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi thoát khỏi Control.

On Got Focus: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi

Control nhận con trỏ.

On Lost Focus: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi

Control mất con trỏ.

On Click: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra khi Click chuột.

On Dbl Click: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra Double

Click chuột.

On Mouse Down: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra nhấn giữ chuột (không nhả).

On Mouse Move: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra di chuyển chuột.

On Mouse Up: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra nhả chuột.

On Key Down: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra nhấn phím lần đầu.

On Key Press: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra một phím được nhấn.

On Key Up: chứa tên Macro hoặc thủ tục cho sự kiện xảy ra nhả phím nhấn.

Picture: chứa đường dẫn, tên File hình ảnh.

Picture Type: chọn 1 kiểu cho hình ảnh (Embedded: Nhúng, Linker: Liên kết ).

Size Mode: chọn 1 chế độ kích thước cho hình ảnh (**Clip**: hình ảnh bị cắt, **Stretch**: hình ảnh tự kéo giãn cho vừa, **Zoom**: hình ảnh được nhìn qua khung chứa).

Picture Alignment: chọn 1 vị trí bố trí hình ảnh (**Top Left**: góc trên trái, **Top Right**: góc trên phải, **Center**: ở giữa khung, **Bottom Left**: góc dưới trái, **Bottom Right**: góc dưới phải).

Picture Tiling: (Yes / No) có hoặc không có “Hoa văn” cho hình ảnh.

Hyperlink Address: chứa địa chỉ của đối tượng, tài liệu, trang Web siêu liên kết.

Hyperlink SubAddress: chứa địa chỉ phụ đối tượng, tài liệu, trang Web siêu liên kết.

ControlTip Text: chứa một chuỗi (<255 ký tự) hiện trên màn hình Tip (ScreenTip) khi ta giữ chuột trong Control.

# MỤC LỤC

## Chương 1: GIỚI THIỆU MICROSOFT ACCESS

- 1.1 Giới thiệu MS-Access 2003..... Trang 1
- 1.2 Khái niệm về tập tin cơ sở dữ liệu ..... Trang 1
- 1.3 Xác lập môi trường làm việc ..... Trang 4
- Bài tập ..... Trang 7

## Chương 2: TẠO CƠ SỞ DỮ LIỆU VÀ BẢNG

- 2.1 Những giai đoạn thiết kế ứng dụng CSDL ..... Trang 8
- 2.2 Tạo cơ sở dữ liệu ..... Trang 9
- 2.3 Khái niệm về Bảng ..... Trang 10
- 2.4 Thiết kế cấu trúc Bảng ..... Trang 13
- 2.5 Nhập dữ liệu vào Bảng ..... Trang 19
- 2.6 Các thuộc tính sử dụng ..... Trang 20
- 2.7 Thiết lập Lookup..... Trang 26
- Bài tập ..... Trang 32

## Chương 3: THAO TÁC TRÊN BẢNG

- 3.1 Hiệu chỉnh Bảng ..... Trang 35
- 3.2 Khai thác thông tin của Bảng..... Trang 37
- Bài tập ..... Trang 42

## Chương 4: MỐI QUAN HỆ GIỮA CÁC BẢNG

- 4.1 Khái quát..... Trang 45
- 4.2 Khai báo quan hệ giữa các Bảng ..... Trang 46
- Bài tập ..... Trang 50

## Chương 5: TOÁN TỬ - BIỂU THỨC - HÀM

- 5.1 Toán tử ..... Trang 53
- 5.2 Biểu thức..... Trang 55

5.3 Hàm.....	Trang 56
Bài tập .....	Trang 59

## Chương 6: TRUY VẤN (QUERY)

6.1 Khái quát về Query .....	Trang 60
6.2 Thiết kế Query .....	Trang 62
6.3 Các chế độ hiển thị Query .....	Trang 62
6.4 Thao tác với Query .....	Trang 65
6.5 Query Wizard.....	Trang 66
Bài tập .....	Trang 68

## Chương 7: SELECT QUERY

7.1 Thiết kế Select Query .....	Trang 69
7.2 Khai báo quan hệ trong Query .....	Trang 73
7.3 Sử dụng tiêu chuẩn chọn lựa trong Query .....	Trang 75
7.4 Tạo Field tính toán (Calculated Field) trong Query .....	Trang 78
7.5 Query tổng cộng nhóm .....	Trang 79
7.6 Summary Select Query .....	Trang 81
7.7 Các vấn đề bổ sung .....	Trang 82
Bài tập .....	Trang 84

## Chương 8: CÁC QUERY KHÁC

8.1 Parameter Query .....	Trang 85
8.2 CrossTab Query .....	Trang 87
8.3 Action Query .....	Trang 89
8.4 Find Duplicate, Find Unmatched Query.....	Trang 93
Bài tập .....	Trang 95

## Chương 9: BIỂU MẪU (FORM)

9.1 Khái quát về biểu mẫu .....	Trang 99
9.2 Thiết kế biểu mẫu với Wizard .....	Trang 100

9.3 Tự thiết kế biểu mẫu .....	Trang 102
9.4 Thao tác trên cửa sổ thiết kế .....	Trang 106
9.5 Các thuộc tính cơ bản .....	Trang 109
9.6 Các đối tượng trên biểu mẫu.....	Trang 110
9.7 Các vấn đề bổ sung .....	Trang 114
Bài tập .....	Trang 119

## Chương 10: MAIN FORM – SUB FORM

10.1 Khái quát về Main form và Sub form .....	Trang 122
10.2 Thiết kế MainForm/SubForm với Wizard .....	Trang 123
10.3 Tự tạo MainForm/SubForm.....	Trang 125
10.4 Thiết kế Form có hai cấp SubForm .....	Trang 127
10.5 Định dạng có điều kiện .....	Trang 129
Bài tập .....	Trang 132

## Chương 11: BÁO BIỂU (REPORT)

11.1 Giới thiệu .....	Trang 136
11.2 Thiết kế báo biểu với Wizard .....	Trang 137
11.3 Tự thiết kế báo biểu .....	Trang 139
11.4 Xếp thứ tự và kết nhóm .....	Trang 141
Bài tập .....	Trang 145

## Chương 12: MACRO

12.1 Giới thiệu .....	Trang 146
12.2 Tạo Macro đơn giản.....	Trang 146
12.3 Hành động và đối số của Macro .....	Trang 150
12.4 Macro điều kiện .....	Trang 158
12.5 Macro nhóm.....	Trang 159
12.6 Sử dụng Macro vào Form/Report .....	Trang 161
12.7 Tự động mở biểu mẫu khi mở CSDL .....	Trang 163
Bài tập .....	Trang 164

## Chương 13: ĐƠN THỂ VÀ THỦ TỤC

13.1 Giới thiệu .....	Trang 166
13.2 Thủ tục và đơn thể .....	Trang 166
13.3 Các thành phần cơ bản trong ngôn ngữ Basic .....	Trang 176
13.4 Cấu trúc điều khiển .....	Trang 180
Bài tập .....	Trang 187

## Chương 14: ĐỐI TƯỢNG TRUY XUẤT DỮ LIỆU

14.1 Giới thiệu .....	Trang 188
14.2 Sử dụng cấu trúc điều khiển với biến đối tượng .....	Trang 192
14.3 Xử lý dữ liệu với các đối tượng DAO .....	Trang 195
14.4 Xử lý dữ liệu thông qua sự kiện.....	Trang 209
Bài tập .....	Trang 219
Bài tập cuối khóa .....	Trang 223
Đề thi mẫu.....	Trang 226
Phụ lục: Các thuộc tính của Form và Control.....	Trang 229
Mục lục .....	Trang 236